pitney bowes

Big Data Quality SDK

Version 12.0

Guía de SDK para Big Data Quality

Contents

1 - Introducción

In this section

Introducción

Big Data Quality SDK lo ayuda a crear, configurar y ejecutar trabajos MapReduce Trabajos Spark y funciones de Hive definidas por el usuario para operaciones de Data Quality en una plataforma Hadoop Hadoop.

Con el uso de SDK, puede crear y ejecutar trabajos directamente en una plataforma Hadoop, de este modo, se eliminan retrasos en la red y se ejecutan los procesos de Data Quality distribuidos en el clúster, lo que provoca una mejora sustancial en el rendimiento.

Los módulos compatibles con Big Data Quality SDK son:

1. Módulo Advanced Matching
2. Módulo Data Normalization
3. Módulo Universal Name
4. Módulo Universal Addressing

Uso de SDK

Este SDK actualmente se puede utilizar a través de:

1. API de Java: admite MapReduce y Spark
2. Funciones de Hive definidas por el usuario

Informes

Big Data Quality SDK proporciona la característica de *Informes* para determinados trabajos. Esta característica utiliza contadores específicos para cada trabajo admitido, lo que le permite monitorear el éxito de cruce logrado por el trabajo correspondiente. Los diversos contadores hacen seguimiento de la cantidad de registros duplicados, la cantidad de registros únicos y otros parámetros útiles de un trabajo ejecutado.

Actualmente, la característica de *Informes* es compatible con estos trabajos:

- Interflow Match
- Intraflow Match
- Transactional Match
- Open Name Parser
- Validate Address
- Validate Address Global
- Validate Address Loqate

Flujo de trabajo

Para usar el SDK, los componentes necesarios son:

instalación de Big Data Quality SDK. El archivo JAR Big Data Quality SDKdebe estar instalado en su sistema y disponible para que la aplicación lo utilice.

Aplicación cliente La aplicación de Java que debe crear para invocar y ejecutar las operaciones de calidad de los datos requeridas mediante el SDK. El archivo JAR Big Data Quality SDK que debe importar en su aplicación Java.

Plataforma Hadoop Durante la ejecución de un trabajo con Big Data Quality SDK, los datos primero se leen desde la plataforma Hadoop configurada y después del procesamiento pertinente, los datos de salida se escriben en la plataforma Hadoop.

Para esto, los detalles de acceso de la plataforma Hadoop deben estar configurados correctamente en su equipo. Para obtener más información, consulte **Información general** en la página 8.

Datos de referencia Los datos de referencia, requeridos por Big Data Quality SDK, se colocan en el clúster Hadoop.

API de Java Para usar la API de Java, puede optar por colocar los datos de referencia en cualquiera de las siguientes ubicaciones:

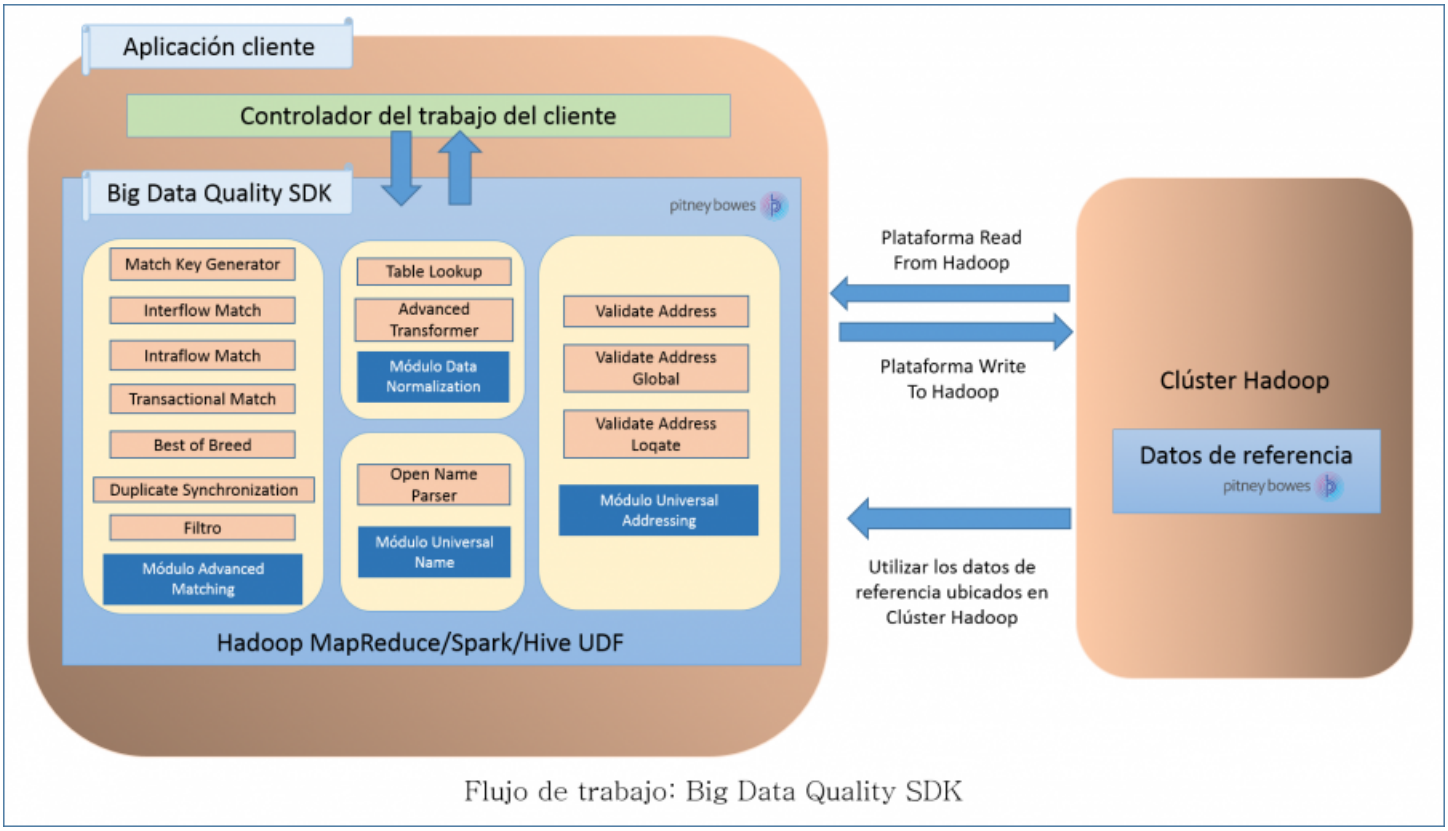
- **Nodos de datos locales**: los datos de referencia se colocan en todos los nodos de datos del clúster.

  Nota: Esta no es un método seguro.

- **Hadoop Distributed File System (HDFS)**: Los datos de referencia se colocan en un directorio HDFS. Esto asegura que sus datos están protegidos.

UDF de Hive Para usar las UDF de Hive, debe colocar los datos de referencia en cada nodo de datos local del clúster.

Nota: El SDK también permite un mejor rendimiento del *Almacenamiento en caché distribuido*.



Flujo de trabajo: Big Data Quality SDK

¿Quién debería usar SDK?

Big Data Quality SDK está diseñado para:

1. Los clientes que deseen revisar la calidad de los datos que residen en Hadoop.
2. Los desarrolladores de Hadoop familiarizados con la programación MapReduce o Spark que deseen crear una solución para un determinado caso de uso.
3. Los desarrolladores de Hadoop que deseen realizar operaciones de *limpieza de datos*, *enriquecimiento de datos*, *deduplicación de datos* y *consolidación de datos* en datos existentes.
4. Los usuarios de Hive que no están familiarizados con las complejidades de MapReduce o Spark, pero que se sienten cómodos con Hive Query Language (HQL), que sintácticamente es similar a SQL.

2 - Instalación

In this section

Requisitos del sistema

Para usar con Hadoop Distributed File System (HDFS):

1. Java JDK versión 1.7 y superior.
2. Hadoop versión 2.6 y superior.
3. Spark 2.0.1 y superior.

Para usar con Hive:

1. Hive versión 1.2.
2. Un cliente Hive a su elección. Por ejemplo, Beeline.

Nota: Puede ejecutar Spectrum™ Technology Platform solo con clústeres Hadoop.

Actualizaciones obligatorias del sistema operativo

Antes de instalar Big Data Quality SDK, asegúrese de aplicar las últimas actualizaciones del producto disponibles para su sistema operativo, especialmente las que solucionan problemas relacionados con Java.

Instalación de SDK

Información general

Use el enlace del correo electrónico de bienvenida para descargar el archivo ZIP. A typical installer ZIP file is downloaded, named like `BigDataSDK120F0101.zip`.

Extraiga el contenido del archivo ZIP descargado en el equipo para acceder al instalador, y ejecute el instalador que lo guía a través del proceso de instalación. Una vez instalada, la herramienta SDK se agrega en el sistema y se coloca en la ubicación definida.

Luego, puede importar el archivo JAR de Big Data Quality SDK en el proyecto e iniciar el acceso a las API desde su equipo.

Módulos compatibles

Big Data Quality SDK admite los módulos.

1. Módulo Advanced Matching
2. Módulo Data Normalization
3. Módulo Universal Name
4. Módulo Universal Addressing

> **Nota:** Debe iniciar el servicio Acushare antes de crear el primer trabajo *Validate Address* del Módulo Universal Addressing. Para obtener más información, consulte **Running Acushare Service** en la página 11.

Uso de SDK

El SDK actualmente se puede utilizar a través de:

1. API de Java
   - API de MapReduce
   - Spark API
2. Funciones de Hive definidas por el usuario

Installer Inclusions

El archivo ZIP de instalación de SDK contiene estos componentes:

1. `Readme.txt`
2. `sdkinst.bin`: Instalador para equipos con LINUX.
3. `sdkinst.exe`: Instalador para equipos con WINDOWS.

Instalación de SDK en Windows

Para instalar Big Data Quality SDK en un equipo con Windows, siga los pasos que figuran a continuación:

1. Descargue el archivo del instalador ZIP Big Data Quality SDK mediante las instrucciones de descarga contenidas en su correo electrónico de bienvenida o el correo electrónico de anuncio de lanzamiento.

2. Extraiga los archivos desde el archivo hasta una ubicación donde desee instalar Big Data Quality SDK.
3. Diríjase al directorio de instalación y ubique el instalador con el nombre *sdkinst.exe*.
4. Haga doble clic en `sdkinst.exe` Aparece el asistente de instalación.
5. Haga clic en **Siguiente**. Aparece la ventana **Elegir carpeta de instalación**.

   Aquí, puede especificar la carpeta donde desea instalar Big Data Quality SDK. Por ejemplo, `C:\Program Files\Pitney Bowes\Spectrum BigDataSDK\SDK`.

   a) Haga clic en el botón **Elegir** para seleccionar la carpeta requerida.
   b) Haga clic en el botón **Restaurar la carpeta predeterminada** para seleccionar la carpeta predeterminada.

   Atención: If you select a non-default folder as the installation directory, ensure that the length of the absolute installation path does not exceed 34 characters.

   The default installation path with 27 characters is admissible:

```
/root/PBSpectrum_BigDataSDK
```

6. Haga clic en **Siguiente**.

   En la pantalla **Resumen de instalación previa**, revise la información de instalación.
7. Haga clic en **Instalar** . Big Data Quality SDK se instala en su computadora.
8. Haga clic en **Listo** para finalizar el proceso de instalación.
9. Verifique que haya instalado SDK de forma correcta. Diríjase a la ubicación donde instaló el SDK, por ejemplo `C:\Program Files\Pitney Bowes\Spectrum BigDataSDK\SDK`.

Una vez que haya instalado de forma correcta SDK en el equipo, estas carpetas se agregan en el directorio de instalación:

- `API`
- `Documentation`
- `modules`
- `samples`
- `utilities`

Nota: To use the jobs of Data Normalization Module, Universal Name Module or Universal Addressing Module, you must install the respective Reference Data for each module.

Instalación de SDK en Linux

Para instalar Big Data Quality SDK con la línea de comandos en un equipo con Linux, siga los pasos que figuran a continuación:

1. Descargue Big Data Quality SDK mediante las instrucciones de descarga contenidas en su correo electrónico de bienvenida o el correo electrónico de anuncio de lanzamiento.
2. Extraiga los archivos desde el archivo hasta una ubicación en el servidor donde desea instalar Big Data Quality SDK.
3. Cambie el directorio a la ubicación.
4. Compruebe si cuenta con el permiso `execute` para ejecutar los archivos mediante el ingreso del comando:

   ```
   chmod a+x sdkinst.bin
   ```

5. Ejecute este comando:

   ```
   ./sdkinst.bin
   ```

   Siga las indicaciones en el cuadro del comando.

6. Cuando el sistema se lo solicite, proporcione el directorio donde desea instalar SDK.

   Por ejemplo, `/home/hadoop/BDQ_InstallPath`.

   Atención: If you select a non-default folder as the installation directory, ensure that the length of the absolute installation path does not exceed 34 characters.

   The default installation path with 27 characters is admissible:

   ```
   /root/PBSpectrum_BigDataSDK
   ```

   Se muestra un resumen de instalación previa.

7. Revise el resumen y presione `ENTER` para continuar con la instalación.
8. Consulte el archivo de registro de instalación para verificar que Big Data Quality SDK se haya instalado de forma correcta.
9. Cuando haya finalizado, presione `ENTER` para finalizar y salir del instalador.

Una vez que haya instalado de forma correcta SDK en el equipo, estas carpetas se agregan en el directorio de instalación:

- `API`
- `Documentation`
- `modules`
- `samples`
- `utilities`

Nota: To use the jobs of Data Normalization Module, Universal Name Module or Universal Addressing Module, you must install the respective Reference Data for each module.

Running Acushare Service

Antes de crear y ejecutar el primer trabajo *Validate Address*, debe ejecutar el servicio Acushare en cada nodo del clúster de Hadoop o Spark.

Nota: Esta es una actividad obligatoria que se debe realizar una sola vez antes de ejecutar el primer trabajo *Validate Address*.

En cada nodo del clúster:

1. Copie la secuencia de comandos de configuración de Acushare `sdkrts.bin` de la ruta de instalación de Big Data Quality SDK en cualquier ubicación del nodo.

   Atención: En el servidor de SDK, la secuencia de comandos de configuración de Acushare `sdkrts.bin` está en `<BDQ SDK_InstallPath>/SDK/utilities/dbloader/aq/runtime/bin`.

2. Inicie sesión en el nodo con los derechos de administrador o como un usuario raíz.
3. Vaya a la ruta donde copió la secuencia de comandos del instalador de Acushare `sdkrts.bin`.
4. Compruebe si cuenta con el permiso `execute` para ejecutar el archivo mediante el ingreso del comando:

   ```
   chmod a+x sdkrts.bin
   ```

5. Ejecute el archivo del instalador y siga las instrucciones:

   ```
   ./sdkrts.bin
   ```

6. Cuando el sistema se lo solicite, presione INTRO para seleccionar la ruta de tiempo de ejecución predeterminada`/root/slave_node`, o ingrese una ruta absoluta de su elección.

   Importante: La ruta de tiempo de ejecución para Acushare debe ser la misma en todos los nodos del clúster para que el trabajo *Validate Address* se ejecute.

   Nota: La ruta seleccionada debe estar presente en el nodo antes de especificarla aquí.

   El servicio Acushare se inicia automáticamente una vez que la instalación finaliza correctamente.
7. O bien, para iniciar el servicio Acushare manualmente en un nodo, vaya a `<Acushare runtime path>/runtime` y ejecute el archivo de secuencia de comandos `startrts.sh` con el argumento `<Acushare runtime path>/runtime`.

Detención del servicio Acushare	Para detener el servicio Acushare en cualquier nodo, vaya a `<Acushare runtime path>/runtime` y ejecute el archivo de secuencia de comandos `stoprts.sh` con el argumento `<Acushare runtime path>/runtime`.
Desinstalación del servicio Acushare	Para desinstalar el servicio Acushare de cualquier nodo, ejecute el archivo de secuencia de comandos `Uninstall_SDKRTS.sh` ubicado en `<Acushare runtime path>/Uninstall`.

Datos de referencia

Información general de los datos de referencia

Los datos de referencia de Pitney Bowes definen un conjunto de valores permitidos usarán otros campos de datos en su sistema para garantizar la calidad de los datos. Mejoran la validez, la precisión y la coherencia de los datos. Le permiten aprovechar mejor sus datos y obtener datos confiables del sistema Big Data.

Por ejemplo, si usa los datos de referencia con el módulo Data Normalization, puede establecer una única identidad de cliente en la empresa. Contar con información de los clientes bien definida es el primer paso para mejorar la eficiencia operativa.

Importante: Para los trabajos *Validate Address* y *Validate Address Global*, los datos de referencia se deben colocar en todos los nodos de datos del clúster de Hadoop. Para el trabajo *Validate Address Loqate*, se deben colocar en un nodo, que luego se debe montar en todos los otros nodos de datos.

Installation Directory Structure

En el directorio de instalación de SDK, el directorio `Utilities/dbloader` contiene las carpetas secundarias:

dataquality Contiene archivos JAR y secuencias de comandos para instalar los datos de referencia en:

- Módulo Data Normalization
- Módulo Universal Name

Nota: Para obtener más información, consulte **Using Reference Data: Data Normalization Module and Universal Name Module** en la página 14.

aq Contiene:

- La secuencia de comandos `scripts/server/installdb_unc.sh` para instalar los datos de referencia. You must run this script to install or extract the data.
- Carpeta `runtime` que contiene la información de configuración del servicio Acushare para el trabajo *Validate Address* del Módulo Universal Addressing.

Nota: Para obtener más información, consulte **Using Reference Data: Universal Addressing Module** en la página 14.

Using Reference Data: Data Normalization Module and Universal Name Module

To use the Reference Data for **Data Normalization Module** and **Universal Name Module** you need to run the data loader script file, for example`installdb_dnm`. La ejecución del archivo de secuencia de comandos le permite extraer datos de referencia en su equipo.

Asegúrese de que el archivo de secuencia de comandos, por ejemplo `installerdb_dnm`, y el archivo JAR estén en la misma carpeta.

1. Inicie sesión en su equipo.
2. Cambie el directorio a la ubicación donde ha instalado el SDK.

   Una vez que haya instalado Big Data Quality SDK correctamente en su equipo, debe tener el cargador de los datos de referencia en el siguiente directorio `BDQ_InstallPath/SDK/utilities/dbloader/unix/bin`.

3. Ejecute la secuencia de comandos del cargador de datos de referencia. Por ejemplo, `installdb_dnm`.
   Aparece una lista enumerada de etapas, como se ve a continuación, y se le pide que seleccione la etapa.
4. Introduzca el nombre de la etapa para la cual quiere cargar los datos.
5. Especifique la ruta adonde se extraen y se colocan los conjuntos de datos de referencia después de la descarga.

   La entrada de los datos de referencia son las tablas básicas del módulo Data Normalization, las principales bases de datos de nombre, y otras similares, requeridas para realizar los trabajos de los módulos Data Normalization y Universal Name.
6. Especifique la ruta del directorio de salida. Esta es la ruta adonde se extraerán sus datos de entrada.
7. El sistema le pregunta si desea ver el archivo de registro. Select as desired.
8. El sistema comienza a cargar los datos. Los datos se extraen en el directorio de salida especificado.

   Nota: Repita los pasos para cada etapa.

Using Reference Data: Universal Addressing Module

Para acceder a los datos de referencia y utilizarlos, primero recupere los datos de la tienda electrónica en formato ZIP.

Para *Validate Address Global* y *Validate Address Loqate*, simplemente extraiga el contenido del archivo ZIP y los datos de referencia estarán listos para su uso.

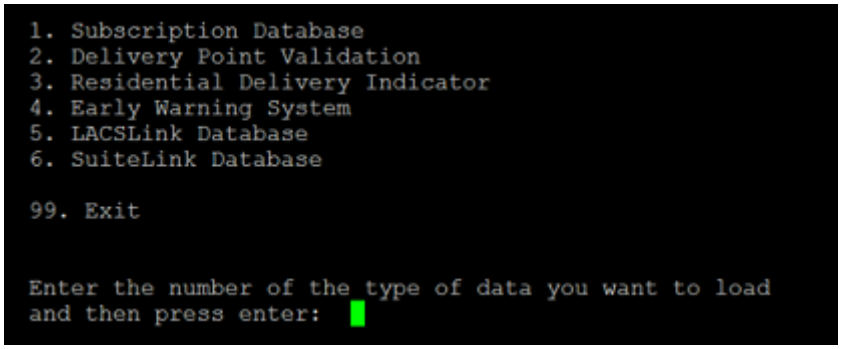
Para *Validate Address*, lleve a cabo los pasos mencionados para extraer los datos de referencia en su equipo.

> **Nota:** Asegúrese de que se haya otorgado el permiso `execute` a la carpeta `aq`.

1. Inicie sesión con los derechos de administrador o como un usuario raíz.
2. Cambie el directorio a la ubicación `<BDQ_Installation>/SDK/utilities/dbloader/aq/scripts/server`.
3. Ejecute la secuencia de comandos `installdb_unc` con el comando:

   ```
   sh installdb_unc.sh <BDQ_Installation/SDK> <Acushare runtime path>
   ```

   Este comando también verifica si el servicio Acushare está en funcionamiento. Si no es así, este comando inicia el servicio.
4. Después de ejecutar este comando, las opciones que se presentan son:
   - **US Subscription**: Press 1 to list the available types of data loading, as mentioned in the next step.
   - **Exit**: Press 99 to exit.
5. Ingrese el número específico para el tipo de datos que desea cargar.

6. Specify the path where the sourced data sets are placed.

   Los datos suministrados por la tienda electrónica están disponibles como una entrada de datos de referencia, que se requiere para realizar los trabajos del Módulo Universal Addressing. Para la ubicación del archivo de salida, el sistema muestra la ruta de salida predeterminada.
7. Se muestran la ubicación del archivo de entrada y la ubicación del archivo de salida.

   Ingrese `c` para continuar, `m` para modificar la ruta predeterminada, o `q` para salir.

```
The Residential Delivery Indicator load environment is currently set to:

Residential Delivery Indicator input file location:
Residential Delivery Indicator output file location: /root/SDK/utilities/dbloader/addressquality/s

Enter c to (c)ontinue
   or m to (m)odify
   or q to (q)uit

   ===>
```

Los datos de entrada se extraen en la ubicación designada del archivo de salida.

8. El sistema le solicita que verifique si la ubicación de su nuevo archivo RDI es correcta. Ingrese `y` o `n`.

```
Please enter full path where you would like to install
the RDI file ==> /root/out

The new RDI file location will be:  /root/out
Is this correct?

Enter (y)es to continue.
      (n)o to try again.

   ===> y

The RDI file output location /root/out does not exist.
Do you want to create it now?

Enter (y)es to create the new RDI file area.
      (n)o to exit.

   ===>
```

El sistema comienza a cargar los datos. Los datos se extraen en el directorio de salida especificado.

Nota: Repeat the steps for the type of data that you want to load.

3 - Módulos

In this section

Módulo Advanced Matching

El módulo Advanced Matching (Comparación avanzada) establece cruces de registros entre una cantidad indefinida de archivos de entrada y/ o dentro de estos archivos. También es posible utilizar el módulo Advanced Matching para establecer cruces en una amplia variedad de campos, lo que incluye los campos de nombre, dirección, nombre y dirección, o bien campos sin nombre ni dirección, como los de número de seguro social o fecha de nacimiento.

El módulo también tiene trabajos para consolidar los registros de un grupo, seleccionando un registro mejor mediante una configuración apropiada, sincronizando todos los registros de un grupo determinado, o bien, filtrando un registro concreto de un grupo de registros.

Puestos de trabajo compatibles

El módulo Advanced Matching de Big Data Quality SDK admite los trabajos:

1. Match Key Generator
2. Interflow Match
   - Generando una clave de coincidencia
   - Usando la clave de coincidencia existente mediante las opciones de Agrupar por
3. Intraflow Match
   - Generando una clave de coincidencia
   - Usando la clave de coincidencia existente mediante las opciones de Agrupar por
4. Transactional Match
   - Generando una clave de coincidencia
   - Usando la clave de coincidencia existente mediante las opciones de Agrupar por
5. Best of Breed
6. Duplicate Synchronization
7. Filtro

Nota: Mientras usa la opción Agrupar por, la clave de coincidencia ya está presente en el archivo de entrada, usando la que realizó la operación Agrupar por.

Match Key Generator

Match Key Generator (Generador de clave de cruce) crea una clave no exclusiva para cada registro, que posteriormente se pueda usar en las etapas de comparación para identificar grupos de registros potencialmente duplicados. Las claves de cruce facilitan el proceso de comparación al permitir la agrupación de registros por clave de cruce y posteriormente solo comparando los registros al interior de estos grupos.

La clave de cruce se crea por medio de reglas definidas por el usuario y se compone de los campos de entrada. Cada campo de entrada especificado cuenta con un algoritmo seleccionado que se ejecuta para el campo. A continuación, el resultado de cada algoritmo es concatenado para crear un único campo de clave de cruce.

Además de crear claves de cruce, también puede crear claves de cruce inmediato, las que posteriormente usará una etapa Intraflow Match o Interflow Match en el flujo de datos.

Puede crear varias claves de cruce y claves de cruce inmediato.

Por ejemplo, si el registro entrante es:

Primer nombre: Fred
Apellido: Mertz
Código postal: 21114-1687
Código de género: M

Y define una regla de clave de cruce que genera una clave de cruce al combinar los datos del registro, de la siguiente manera:

Campo de entrada	Posición de inicio	Longitud
Código postal	1	5
Código postal	7	4
Apellido	1	5
Nombre	1	5
Código de género	1	1

Entonces la clave será:

211141687MertzFredM

Interflow Match

Interflow Match (Cruce de interflujo) localiza los cruces (coincidencias) entre registros de datos similares de dos flujos de registros de entrada. El primer flujo de registros es una fuente de origen de registros sospechosos, mientras que el segundo es una fuente de origen de registros candidatos.

Mediante el criterio de grupo de cruce (por ejemplo, una clave de cruce), Interflow Match identifica un grupo de registros que son posibles duplicados de un registro sospechoso en particular.

Informes

El trabajo de Interflow Match le permite supervisar los resultados del trabajo. Los contadores disponibles son:

DUPLICATE_COLLECTIONS	La cantidad de colecciones duplicadas, que consiste en un registro sospechoso y sus duplicados agrupados por CollectionNumber.
EXPRESS_MATCHES	La cantidad de coincidencias inmediatas en una colección. Una coincidencia inmediata se realiza cuando un sospechoso y un candidato tienen una coincidencia exacta en el contenido de un campo designado; por lo general, en un campo ExpressMatchKey generado por Match Key Generator. Si se realiza una coincidencia inmediata, no se realizan más procesamientos para determinar si el sospechoso y el candidato son duplicados.
AVERAGE_SCORE	La media coincide con puntuación de todos los duplicados. Los valores posibles son 0-100; 0 indica una coincidencia poco satisfactoria y 100, una coincidencia exacta.
INPUT_SUSPECTS	El número de registros en el flujo de entrada que el comprobador intentó hacer coincidir con otros registros.
SUSPECTS_WITH_DUPLICATES	El número de sospechosos de entrada que se cruzan con al menos uno de los registros de candidatos.
UNIQUE_SUSPECTS	El número de sospechosos de entrada que no se cruzan con ninguno de los registros de candidatos.
SUSPECTS_WITH_CANDIDATES	El número de sospechosos de entrada que tenían por lo menos a un candidato de su grupo de coincidencia y que, por lo tanto, tenían al menos un intento de cruce.
SUSPECTS_WITHOUT_CANDIDATES	El número de sospechosos de entrada que no tenían ningún candidato en su grupo de cruce y que, por lo tanto, no tenían intentos de cruce.
TOTAL_DUPLICATE_CANDIDATES	La cantidad total de candidatos duplicados encontrados.
TOTAL_DUPLICATE_SCORE	La puntuación de cruce total de todos los duplicados.

Intraflow Match

Intraflow Match (Cruce de intraflujo) localiza los cruces (coincidencias) entre registros de datos similares en un mismo flujo de entrada. Puede crear reglas jerárquicas basadas en cualquier campo que se haya definido o creado en otras etapas en el flujo de datos.

Informes
El trabajo de Intraflow Match le permite supervisar los resultados del trabajo. Los contadores disponibles son:

INPUT_RECORDS	La cantidad de registros en la etapa de coincidencia antes de que se realice la clasificación por coincidencia.
DUPLICATE_RECORDS	El número de registros duplicados dentro de un grupo de coincidencia, que puede ser un registro sospechoso o candidato.
UNIQUE_RECORDS	La cantidad de registros sospechosos o candidatos que no coinciden con ningún otro registro de su grupo de coincidencia respectivo. Si es el único registro de un grupo de coincidencia, el sospechoso automáticamente es único.
MATCH_GROUPS	(Agrupar por) Registros agrupados por clave de coincidencia.
DUPLICATE_COLLECTIONS	La cantidad de colecciones duplicadas, que consiste en un registro sospechoso y sus duplicados agrupados por CollectionNumber.
EXPRESS_MATCHES	La cantidad de coincidencias inmediatas en una colección. Una coincidencia inmediata se realiza cuando un sospechoso y un candidato tienen una coincidencia exacta en el contenido de un campo designado; por lo general, en un campo ExpressMatchKey generado por Match Key Generator. Si se realiza una coincidencia inmediata, no se realizan más procesamientos para determinar si el sospechoso y el candidato son duplicados.
AVERAGE_SCORE	La media coincide con puntuación de todos los duplicados. Los valores posibles son 0-100; 0 indica una coincidencia poco satisfactoria y 100, una coincidencia exacta.
TOTAL_DUPLICATES	La cantidad total de candidatos duplicados encontrados.
TOTAL_SCORE	La puntuación de cruce total de todos los duplicados.

Transactional Match

Transactional Match hace coincidir los registros sospechosos con los registros candidatos de un grupo de registros para identificar duplicados. Los registros primero se agrupan por columna

seleccionada, después de lo cual se marca el primer registro como el registro sospechoso. Todos los registros restantes del grupo, denominados registros candidato, se comparan con el registro sospechoso.

Si el registro candidato es un duplicado, se le asigna un número de colección, el tipo de registro de cruce se etiqueta como duplicado y luego se copia. A los candidatos sin coincidencias en el grupo se les asigna el número de colección 0, se etiquetan como únicos y luego también se copian.

Informes

El trabajo de Transactional Match le permite supervisar los resultados del trabajo. Los contadores disponibles son:

AVERAGE_SCORE	La media coincide con puntuación de todos los duplicados. Los valores posibles son 0-100; 0 indica una coincidencia poco satisfactoria y 100, una coincidencia exacta.
INPUT_SUSPECTS	El número de registros en el flujo de entrada que el comprobador intentó hacer coincidir con otros registros.
SUSPECTS_WITH_DUPLICATES	El número de sospechosos de entrada que se cruzan con al menos uno de los registros de candidatos.
UNIQUE_SUSPECTS	El número de sospechosos de entrada que no se cruzan con ninguno de los registros de candidatos.
SUSPECTS_WITH_CANDIDATES	El número de sospechosos de entrada que tenían por lo menos a un candidato de su grupo de coincidencia y que, por lo tanto, tenían al menos un intento de cruce.
SUSPECTS_WITHOUT_CANDIDATES	El número de sospechosos de entrada que no tenían ningún candidato en su grupo de cruce y que, por lo tanto, no tenían intentos de cruce.
TOTAL_DUPLICATES_SCORE	La puntuación de cruce total de todos los duplicados.
TOTAL_DUPLICATES	La cantidad total de candidatos duplicados encontrados.

Best of Breed

Best of Breed consolida los registros duplicados. Para ello, selecciona los mejores datos de una colección de registros duplicados y crea un nuevo registro consolidado con los mejores datos. Este "súper" registro se conoce como el registro "best of breed" (el mejor de la especie). Usted debe definir las reglas a usar para seleccionar los registros que habrán de procesarse. Cuando el proceso finaliza, el registro "best of breed" se conserva en el sistema.

Duplicate Synchronization

Duplicate Synchronization (Sincronización duplicada) determina cuáles son los campos de una colección de registros que deben copiarse en los campos correspondientes de todos los registros de la colección. Usted puede especificar las reglas que deben cumplir los registros para copiar los datos del campo en los demás registros de la colección Una vez que el proceso finaliza, todos los registros de la colección se conservan.

Filtro

La etapa Filter (Filtro) conserva o elimina registros de un grupo de registros según las reglas que usted especifique.

Módulo Data Normalization

El Módulo de normalización de datos examina los términos en un registro y determina si el término está en el formato preferido.

- **Table Lookup**: esta etapa evalúa un término y lo compara con un formato previamente validado de ese término. Si el término no está en el formato adecuado, es reemplazado por su versión estándar. Las funciones de Table Lookup incluyen el cambio de palabras completas a abreviaturas y viceversa, el cambio de apodos a nombres completos o la corrección de palabras mal escritas.
- **Advanced Transformer**: esta etapa explora y divide las cadenas de datos en múltiples campos, y coloca los datos extraídos y sin extraer en un campo ya existente o nuevo.

Puestos de trabajo compatibles

El módulo Data Normalization de Big Data Quality SDK admite los trabajos:

1. Table Lookup
   - Table Lookup con opción Estandarizar
   - Table Lookup con opción Identificar
   - Table Lookup con opción Categorizar
2. Advanced Transformer
   - Advanced Transformer con opción Extracción de datos de tabla

- Advanced Transformer con opción Extracción de expresiones regulares

Table Lookup

La etapa Table Lookup estandariza los términos en función de un formato validado anteriormente para el término en cuestión y aplica la versión estándar. Para realizar esta evaluación, se realiza una búsqueda en una tabla para encontrar el término a estandarizar.

Advanced Transformer

El trabajo Advanced Transformer explora y divide las cadenas de datos en múltiples campos por medio de tablas o expresiones regulares. Esta herramienta extrae un término específico o una cantidad determinada de palabras situadas a la derecha o la izquierda de un término. Los datos extraídos y sin extraer pueden colocarse en un campo nuevo o ya existente.

Por ejemplo, supongamos que se desea extraer la información de habitación (suite) de este campo de dirección para colocarla en un campo separado.

2300 BIRCH RD STE 100

Para lograrlo, se crea una instancia de Advanced Transformer que extrae el término STE (suite) y todas las palabras a la derecha de ese término, de forma tal que el campo queda así:

2300 BIRCH RD

Módulo Universal Addressing

El módulo Universal Addressing (Direcciones universales) es un módulo que controla la calidad de las direcciones y permite estandarizarlas y validarlas, lo que mejora la capacidad de entrega de los mensajes de correo. El módulo Universal Addressing le permite asegurarse de que los datos de dirección se ajusten a los estándares de calidad establecidos por las autoridades postales. Una dirección que se ajusta a estos estándares tiene más probabilidades de entregarse con puntualidad. Además, los proveedores de servicios de correo que siguen estos estándares pueden acceder a importantes descuentos postales. Para obtener información sobre los descuentos para el sistema postal de Estados Unidos, consulte el **Domestic Mail Manual** (DMM, manual de correo nacional) de USPS disponible en www.usps.com.

> **Nota:** Para los trabajos de UAM, los datos de referencia deben estar colocados solo en nodos de datos locales en el clúster.

Puestos de trabajo compatibles

El módulo Universal Addressing de Big Data Quality SDK admite los trabajos:

1. Validate Address

   Nota: Este trabajo actualmente es compatible solo con validaciones de dirección de EE. UU.

2. Validate Address Global
3. Validate Address Loqate

Validate Address

Validate Address estandariza y valida direcciones por medio de los datos de dirección de las autoridades postales. Validate Address puede corregir la información y dar formato a la dirección aplicando el formato de preferencia de la autoridad postal correspondiente. También puede agregar la información postal faltante, como códigos postales, nombres de ciudades, estados o provincias, y más.

Validate Address también arroja indicadores de resultados referidos a los intentos de validación, como por ejemplo para señalar si Validate Address validó la dirección, cuál es el nivel de confianza respecto de la dirección devuelta, el motivo del error si la dirección no pudo validarse, etc.

Durante el proceso de comparación y estandarización de direcciones, Validate Address separa las líneas de dirección en componentes y los compara con el contenido de las bases de datos del módulo Universal Addressing. Si se encuentra una coincidencia, la dirección de entrada se *estandariza* de acuerdo con la información de la base de datos. Si no se encuentra una coincidencia con la base de datos, Validate Address *asigna formato* a las direcciones de entrada de forma opcional. El proceso de asignación de formato intenta estructurar las líneas de dirección de acuerdo con las convenciones de la autoridad postal correspondiente.

Nota: Currently, Validate Address supports only US addresses.

Informes CASS

Puede crear y ejecutar un trabajo Validate Address en el modo CASS Certified™ con Big Data Quality SDK.

Además, puede optar por generar los siguientes tipos de informes CASS:

1. Informe CASS 3553
2. Informe CASS detallado
3. Informe resumido Validate Address

Procesamiento CASS Certified

El procesamiento CASS Certified™ también genera el informe detallado CASS de USPS (USPS CASS Detailed Report), que contiene parte de la misma información que el informe 3553, pero proporciona información mucho más detallada acerca de las estadísticas DPV, LACS, y SuiteLink. El el informe detallado CASS de USPS no es obligatorio para acceder a descuentos postales y no se requiere su envío en el correo.

El informe CASS detallado se genera en tres partes, con los siguientes nombres:

1. *CASS Detail*
2. *CASS Detail 2*
3. *CASS Detail 3*

Para obtener más información sobre la configuración de CASS cuando usa SDK, consulte **Uso de un trabajo MapReduce de Validate Address** en la página 113 y **Uso de un trabajo Spark de Validate Address** en la página 115. Para obtener instrucciones sobre cómo usar los informes, consulte *Guía de Dataflow Designer*.

Informe CASS 3553

El informe USPS CASS 3553 debe entregarse a USPS junto con la pieza de correo a fin de reunir los requisitos necesarios para determinados descuentos. El informe contiene información sobre el software utilizado para el procesamiento CASS, la lista de nombres y direcciones, el archivo de salida, el proveedor de servicios de correo y otras estadísticas acerca del envío de correo. Para obtener información detallada sobre el formulario 3553 de USPS, consulte **www.usps.com**.

Para obtener instrucciones sobre cómo usar los informes, consulte *Guía de Dataflow Designer*.

Informe CASS detallado

No es necesario entregar el Informe detallado de USPS CASS al USPS para optar a ciertos descuentos. Este informe contiene información que se incluye en el informe 3553, pero proporciona mayor detalle acerca de estadísticas de DPV, LACS y SuiteLink.

Para obtener instrucciones sobre cómo usar los informes, consulte *Guía de Dataflow Designer*.

Informe resumido Validate Address

El informe resumido Validate Address muestra estadísticas acerca del trabajo, como la cantidad total de registros procesados, la cantidad de direcciones validadas, y otros datos.

Para obtener instrucciones sobre cómo usar los informes, consulte *Guía de Dataflow Designer*.

Validate Address Global

Validate Address Global ofrece funciones mejoradas de estandarización y validación para direcciones que no corresponden a Estados Unidos y Canadá. Validate Address Global también puede validar

direcciones en Estados Unidos y Canadá, pero su punto fuerte es la validación de direcciones en otros países. Si procesa un número significativo de direcciones fuera de los EE. UU. y Canadá, analice la posibilidad de usar Validate Address Global.

Validate Address Global forma parte del módulo Universal Addressing.

Validate Address Global ejecuta diversos pasos para obtener una dirección de calidad, lo que incluye análisis, validación y aplicación de formato.

Análisis, formato y estandarización de direcciones

La reestructuración de datos de direcciones asignados a campos incorrectos es una tarea compleja y dificultosa, especialmente cuando se trata de direcciones internacionales. Las personas introducen muchos datos ambiguos al ingresar direcciones en los sistemas informáticos. Los problemas incluyen elementos colocados en lugares incorrectos (como nombres personales o de empresas que aparecen en campos de direcciones de calles) o diferentes abreviaturas que no solo son específicas del idioma, sino de un país. Validate Address Global identifica las elementos de dirección en las líneas de dirección y los asigna a los campos correctos. Este es un importante paso previo a la validación real. Sin esta reestructuración, pueden generarse situaciones en las que no se generan cruces.

Los elementos de dirección correctamente identificados también son importantes en los casos en los que las direcciones deben recortarse o acortarse para cumplir con los requisitos de longitud de un campo. Si existe la información correcta en los campos adecuados, pueden aplicarse las reglas para truncar datos.

- Se analizan las líneas de dirección y se identifican los elementos de dirección individuales
- Se procesan más de 30 conjuntos de caracteres diferentes
- Se aplica el formato correspondiente a las direcciones de acuerdo con las reglas postales del país de destino
- Se estandarizan los elementos de dirección (como por ejemplo, el cambio de AVENUE a AVE)

Validación global de direcciones

La validación de direcciones es un proceso de corrección en el que los datos de dirección analizados de forma adecuada se comparan con las bases de datos de referencia suministradas por las organizaciones postales u otros proveedores de datos. Validate Address Global valida los elementos de dirección individuales para verificar si son correctos por medio de sofisticadas tecnologías de comparación, y genera resultados estandarizados y con formato aplicado sobre la base de las normas postales y las preferencias del usuario. El tipo de validación FastCompletion (Finalización rápida) puede usarse en aplicaciones de ingreso rápido de direcciones. Esta función permite ingresar datos truncados en diferentes campos de dirección y genera sugerencias sobre la base de esos datos ingresados.

En algunos casos, no es posible validar por completo una dirección. En esos casos, Validate Address Global ofrece una exclusiva función de evaluación de capacidad de entrega que clasifica las direcciones de acuerdo con la mayor o menor probabilidad de entrega.

Creación de informes de contadores

El trabajo Validate Address Global le permite monitorizar las estadísticas del trabajo una vez finalizada la ejecución. Los contadores proporcionan los informes estadísticos en todos los países admitidos en los que se ejecuta un trabajo Validate Address Global en particular.

Para obtener una lista de los países admitidos, consulte **Códigos de país ISO y compatibilidad de módulos** en la página 207.

Contadores basados en países

Estos contadores proporcionan los informes estadísticos para varios países admitidos. Cada etiqueta del contador comienza con el código del país al que el valor del contador corresponde.

Por ejemplo, estos contadores proporcionan los informes estadísticos para Estados Unidos:

1. UNITEDSTATES_STATUS_I4_COUNT
2. UNITEDSTATES_STATUS_S_COUNT
3. UNITEDSTATES_STATUS_I3_COUNT
4. UNITEDSTATES_FAILED_COUNT
5. UNITEDSTATES_STATUS_I2_COUNT
6. UNITEDSTATES_STATUS_C_COUNT
7. UNITEDSTATES_STATUS_V_COUNT

Asimismo, los mismos contadores aparecen para todos los países admitidos para los que se ejecuta el trabajo Validate Address Global.

Contadores de resumen

Los contadores de resumen proporcionan la suma total de los valores de cada tipo de contador en particular en todos los países.

Por ejemplo, el contador `SUMMARY_FAILED_COUNT` es la suma de los valores del contador `FAILED_COUNT` para todos los países admitidos en los que se ejecuta un trabajo Validate Address Global en particular.

1. SUMMARY_STATUS_I4_COUNT
2. SUMMARY_STATUS_I2_COUNT
3. SUMMARY_END_TIME
4. SUMMARY_START_TIME
5. SUMMARY_STATUS_V_COUNT
6. SUMMARY_STATUS_C_COUNT
7. SUMMARY_CHARSET
8. SUMMARY_DEFAULT_COUNTRY
9. SUMMARY_STATUS_I3_COUNT
10. SUMMARY_STATUS_S_COUNT
11. SUMMARY_FAILED_COUNT

12. COUNTRY: una lista separada por comas de los códigos de país para los que se ejecuta la validación de dirección.
13. SUMMARY_CASING: el método de mayúsculas y minúsculas de los datos de salida. Para obtener más información, consulte la sección *Opciones* de la etapa *Validate Address Global* en la *Guía de direcciones*.

Validate Address Loqate

Validate Address Loqate estandariza y valida direcciones por medio de los datos de dirección de las autoridades postales. Validate Address Loqate puede corregir la información y dar formato a la dirección aplicando el formato de preferencia de la autoridad postal correspondiente. También puede agregar la información postal que falta, como códigos postales, nombres de ciudades, estados o provincias, entre otros datos.

Validate Address Loqate también arroja indicadores de resultados referidos a los intentos de validación, como por ejemplo para señalar si Validate Address Loqate validó la dirección, cuál es el nivel de confianza respecto de la dirección devuelta, el motivo del error si la dirección no pudo validarse, etc.

Durante el proceso de comparación y estandarización de direcciones, Validate Address Loqate separa las líneas de dirección en componentes y los compara con el contenido de las bases de datos del módulo Universal Addressing. Si se encuentra una coincidencia, la dirección de entrada se estandariza de acuerdo con la información de la base de datos. Si no se encuentra una coincidencia con la base de datos, ValidateAddress Loqate asigna formato a las direcciones de entrada de forma opcional. El proceso de asignación de formato intenta estructurar las líneas de dirección de acuerdo con las convenciones de la autoridad postal correspondiente.
Validate Address Loqate forma parte del módulo Universal Addressing.

Reporting Counters

El trabajo de Intraflow Match le permite supervisar los resultados del trabajo. Los contadores disponibles son:

1. Original Postal Code Confirmed via Address Match
2. Total Records Successfully Matched
3. House Mismatch
4. Total Records for which Address Validation Attempted
5. Input Record Count
6. Number Range Mismatch
7. Total Records Valid on Input
8. No Postal Code Available
9. Total Unmatched Recorded
10. Total Corrected
11. Total Unmatched Records

12. Postal Code Corrected via Address Match
13. Standard Address Returned Successfully
14. Address Records Processed
15. Street Mismatch
16. Original Postal Code Retained
17. Records Processed by LOQATE

Módulo Universal Name

Para llevar a cabo la normalización más precisa, puede ser necesario dividir las cadenas de datos en varios campos. Big Data Quality SDK proporciona características de análisis avanzado que le permiten analizar nombres personales, nombres de empresas y muchos otros términos y abreviaturas.

Puestos de trabajo compatibles

El módulo Universal Name de Big Data Quality SDK admite el trabajo:

1. Open Name Parser

Open Name Parser

Open Name Parser divide los nombres personales, nombres de empresas y otros términos del campo de datos de nombres en las partes que los conforman. De este modo, estos elementos de nombre analizados se encuentran disponibles para otras operaciones automatizadas tales como comparación y estandarización de nombres o consolidación de nombres de múltiples registros.

Informes

Open Name Parser ofrece estadísticas resumidas sobre el trabajo, como la cantidad total de registros de entrada y la cantidad total de registros que no contienen datos de nombre, además de diversas estadísticas de análisis sintáctico.

Resultados generales

INPUT_RECORDS	La cantidad de registros en la entrada.
NO_NAME_DATA_RECORDS	La cantidad de registros en la entrada que no contienen datos de nombre para analizar.

NAMES_PARSED_OUT	La cantidad de nombres en la entrada que se analizaron.
LOWEST_NAME_PARSING_SCORE	El resultado de análisis más bajo que se da a un nombre en la entrada.
HIGHEST_NAME_PARSING_SCORE	El resultado de análisis más alto que se da a un nombre en la entrada.
AVERAGE_NAME_PARSING_SCORE	El resultado promedio del análisis entre todos los nombres analizados en la entrada.

Resultados del análisis de los nombres personales

PERSONAL_NAME_RECORDS	La cantidad de nombres personales en la entrada.
CONJOINED_NAMES_PARSED	La cantidad de nombres analizados de los registros que tenían nombres conjuntos. Por ejemplo, si la entrada tenía cinco registros con dos nombres conjuntos y siete registros con tres nombres conjuntos, el valor de contador para este campo es 31, según la ecuación: (5 x 2) + (7 x 3).
TWO_CONJOINED_NAMES_RECORDS	La cantidad de registros de entrada que contienen dos nombres conjuntos.
THREE_CONJOINED_NAMES_RECORDS	La cantidad de registros de entrada que contienen tres nombres conjuntos.
TITLE_OF_RESPECT_NAMES	La cantidad de nombres analizados que contienen un título de tratamiento.
MATURITY_SUFFIX_NAMES	La cantidad de nombres analizados que contienen un sufijo generacional.
GENERAL_SUFFIX_NAMES	La cantidad de nombres analizados que contienen un sufijo general.
ACCOUNT_DESCRIPTION_PERSONAL_NAMES	La cantidad de nombres analizados que contienen una descripción de cuenta.
TOTAL_REVERSE_ORDER_NAMES	La cantidad de nombres analizados en orden inverso, lo que deja el campo de salida `IsReverseOrder` como “Verdadero”.

Resultados del análisis de los nombres de empresas

BUSINESS_NAME_RECORDS	La cantidad de registros de entrada que contienen nombres comerciales.
FIRM_SUFFIX_NAMES	La cantidad de nombres analizados que contienen un sufijo de empresa.
ACCOUNT_DESCRIPTION_BUSINESS_NAMES	La cantidad de registros de entrada que tienen una descripción de cuenta.

TOTAL_DBA_RECORDS	La cantidad de registros de entrada que contienen conjunciones Doing Business As (DBA), lo que deja los campos de salida `isPersonal` y `isFirm` como "Verdaderos".
TOTAL_PARSED	La cantidad total de nombres analizados.
TOTAL_NAME_PARSING_SCORE	El resultado total de análisis de todos los nombres.

4 - La API Java

In this section

Introducción

Una *clase* Java es un planto técnico o prototipo que define las variables y los métodos comunes a todas las instancias de un tipo determinado. Define la implementación de un tipo de instancia particular.

Un *objeto* Java es una instancia de una clase Java. Es una instancia en tiempo real de clases Java, creada mediante la máquina virtual de Java. Una instancia de una clase, manejada mediante una variable, encapsula la información en tiempo real de la clase.

Los *métodos* de una clase definen las distintas funciones que deben realizar una clase o su objeto. Los métodos son similares a las funciones o procedimientos en lenguajes procedurales, por ejemplo C.

Los *parámetros* se utilizan para pasar la información que requiere un objeto para realizar una determinada tarea.

Los objetos de software Java interactúan y se comunican entre sí por medio de *mensajes*.

Para obtener más información sobre la tecnología Java, visite **www.oracle.com/java**.

Componentes de la API de Java de SDK

Los componentes clave para usar un trabajo Big Data Quality SDK con la API de Java son:

Archivos JAR

1. Archivos JAR de Hadoop
2. Los archivos JAR del módulo al que pertenece el trabajo Big Data Quality SDK deseado, como se indica en la tabla:

Módulo	Trabajo	Archivo JAR
Módulo Advanced Matching	Todos los trabajos AMM	*amm.core-12.0.jar*
Módulo Data Normalization	Todos los trabajos DNM	*dnm.core-12.0.jar*
Módulo Universal Addressing	Validate Address	*uam-universaladdress.core-12.0.jar*
Módulo Universal Addressing	Validate Address Global	*uam-global.core-12.0.jar*

Módulo	Trabajo	Archivo JAR
Módulo Universal Addressing	Validate Address Loqate	*uam-loqate.core-12.0.jar*
Módulo Universal Name	Todos los trabajos UNM	*unm.core-12.0.jar*

Archivos de configuración Archivos en formato XML que contienen todos los parámetros y valores necesarios para ejecutar un trabajo, como reglas de cruce, detalles de archivo de entrada, detalles de archivo de salida, detalles de configuración de MapReduce o Spark y otros datos similares.

Los archivos de configuración XML se colocan en la ubicación `<Big Data Quality bundle>\samples\configuration`.

Aplicación de cliente Java Aplicación de Java para usar la API a fin de crear y ejecutar el trabajo Big Data Quality SDK requerido provisto por su API de Java.

Plataforma Hadoop El trabajo creado accede a la plataforma configurada de Hadoop para acceder a los datos de entrada y volcar los datos de salida en un archivo.

Uso de la SDK

La SDK se puede usar para ejecutar trabajos Big Data Quality SDK utilizando uno de estos dos enfoques:

1. En una consola, ejecute directamente los archivos JAR específicos del módulo y pase los diversos archivos de propiedades de configuración en formato XML como argumentos a los comandos.

   Para trabajos MapReduce, ejecute el comando`hadoop`, en tanto que para trabajos Spark, ejecute el comando `submit-spark`.

   Para conocer los pasos, consulte **Uso de archivos de propiedades de configuración** en la página 35.

2. Cree su propio proyecto cliente Java importando el archivo JAR del módulo Big Data Quality SDK relevante, especifique todas las configuraciones requeridas del trabajo deseado dentro de su proyecto cliente y ejecútelo.

   Para conocer los pasos, consulte **Crear una aplicación Java** en la página 37.

Uso de archivos de propiedades de configuración

Asegúrese de que Big Data Quality SDK esté instalado en su equipo.

Puede ejecutar un trabajo Big Data Quality SDK usando los archivos JAR específicos del módulo y los archivos de configuración en formatos XML.

Se envían propiedades de configuración de muestra con el Big Data Quality SDK, los que se colocan en la ubicación `<Big Data Quality bundle>\samples\configuration`.

> **Nota:** Para ver una lista de los archivos JAR específicos del módulo, consulte **Componentes de la API de Java de SDK** en la página 34.

1. Para un sistema Linux, abra una indicación de comando.

   Para sistemas Windows y Unix, abra un cliente SSH, como Putty.

2. Para un trabajo *MapReduce*, use el comando`hadoop`.

   Según el trabajo que desee ejecutar:

   1. Pase el nombre del archivo JAR de ese módulo.
   2. Pase el nombre de clase de controlador `RunMRSampleJob`.
   3. Pase los diversos archivos de configuración como una lista de argumentos. Cada clave de argumentos acepta la ruta de un único archivo de propiedades de configuración, donde cada archivo contiene múltiples propiedades de configuración.

   La sintaxis del comando es:

   ```
   hadoop jar <Name of module JAR file> RunMRSampleJob [-config <Path to
   configuration file>] [-debug] [-input <Path to input configuration
   file>] [-conf <Path to MapReduce configuration file>] [-output <Path
   of output directory>]
   ```

   Por ejemplo, para un trabajo MapReduce MatchKeyGenerator:

   ```
   hadoop jar amm.core.12.0.jar RunMRSampleJob -config
   /home/hadoop/matchkey/mkgConfig.xml -input
   /home/hadoop/matchkey/inputFileConfig.xml -conf
   /home/hadoop/matchkey/mapReduceConfig.xml -output
   /home/hadoop/matchkey/outputFileConfig.xml
   ```

3. Para un trabajo *Spark*, use el comando`spark-submit`.

   Según el trabajo que desee ejecutar:

   1. Pase el nombre del archivo JAR de ese módulo.
   2. Pase el nombre de clase de controlador `RunSparkSampleJob`.
   3. Pase los diversos archivos de configuración como una lista de argumentos. Cada clave de argumentos acepta la ruta de un único archivo de propiedades de configuración, donde cada archivo contiene múltiples propiedades de configuración.

   La sintaxis del comando es:

   ```
   spark-submit --class RunSparkSampleJob <Name of module JAR file> [-config
   <Path to configuration file>] [-debug] [-input <Path to input
   ```

```
configuration file>] [-conf <Path to Spark configuration file>] [-output
<Path of output directory>]
```

Por ejemplo, para un trabajo Spark MatchKeyGenerator:

```
spark-submit --class RunSparkSampleJob amm.core.12.0.jar -config
/home/hadoop/spark/matchkey/matchKeyGeneratorConfig.xml -input
/home/hadoop/spark/matchkey/inputFileConfig.xml -output
/home/hadoop/spark/matchkey/outputFileConfig.xml
```

Nota: Para revisar una lista de claves de argumento admitidas para los comandos `hadoop` o`spark-submit`, ejecute los comandos:

```
hadoop --help
```

o bien

```
spark-submit --help
```

Crear una aplicación Java

Asegúrese de que Big Data Quality SDK esté instalado en su equipo.

Para utilizar el SDK:

1. Cree un proyecto Java para usar el SDK según sea necesario mediante uno de estos métodos:
   a) Cree un proyecto Java específico para ejecutar la operación de calidad de datos que requiere.

      Mediante este método, tendrá que crear proyectos Java independientes para cada trabajo de calidad de datos que desee ejecutar.

   b) Cree un proyecto Java común para ejecutar cualquiera de las operaciones de calidad de los datos que desee usando los argumentos de tiempo de ejecución que correspondan.

      Mediante este método, tendrá que crear solo un proyecto Java que acepte argumentos de tiempo de ejecución correspondientes a la operación de calidad de datos que desea.
2. Importe el archivo JAR específico del módulo Big Data Quality SDK en su proyecto para utilizar el SDK. Para ver una lista de los archivos JAR específicos del módulo, consulte **Componentes de la API de Java de SDK** en la página 34.
3. Importe los archivos JAR de Hadoop requeridos en el proyecto.
4. Usando las configuraciones apropiadas, cree su aplicación para ejecutar los trabajos de calidad de datos que desea.
5. Construya su proyecto utilizando cualquier herramienta de compilación, como Maven o Ant. Como resultado, se crea un archivo JAR de su proyecto.

   Por ejemplo, se crea`MatchKeyGeneratorClient-with-dependencies.jar`.
6. Coloque el archivo JAR de su proyecto en la plataforma Hadoop.

7. En la plataforma Hadoop, en un símbolo del sistema, cambie el directorio a la ruta donde colocó su archivo JAR.
8. Ejecute el JAR del proyecto mediante el comando:

```
hadoop jar <name of the JAR of your client project> <fully qualified
name of the main class>
```

Por ejemplo:

```
hadoop jar MatchKeyGeneratorClient-with-dependencies.jar
com.company.bdq.amm.mr.MatchKeyGeneratorJob
```

El trabajo deseado se crea y se ejecuta en la plataforma Hadoop.

Su aplicación Java accede a los datos de entrada desde la ruta especificada en la plataforma Hadoop, y crea y ejecuta el trabajo en dicha plataforma. El resultado del trabajo se vuelca en un archivo en la ruta de salida especificada en la plataforma Hadoop.

Entidades API comunes

ConjoinedRule

Propósito
Un tipo de regla de la consolidación, que se utiliza cuando múltiples reglas se van a unir mediante el uso de los operadores `AND` y `OR`. Una regla conjunta puede incluir reglas simples como componentes. Consulte **SimpleRule** en la página 41.

Esta clase permite la definición de reglas para los trabajos del módulo Advanced Matching y el módulo Data Normalization.

ConsolidationCondition

Propósito
Especificar las reglas de consolidación y la acción correspondiente para los trabajos del módulo Advanced Matching y el módulo Data Normalization.

ConsolidationRule

Propósito
Especificar la regla de consolidación en función de la cual debe determinarse si la acción se requiere en un registro o no.

Esta clase permite la definición de reglas de consolidación para los trabajos del módulo Advanced Matching y el módulo Data Normalization.

ConsolidationAction

Propósito
Especificar el campo que se debe copiar en otros registros en un grupo para una condición de consolidación particular.

Esta clase permite la definición de acciones de consolidación para los trabajos del módulo Advanced Matching y el módulo Data Normalization.

FilePath

Propósito
Para especificar los detalles de un archivo de texto de entrada o salida para ejecutar un trabajo.

JobConfig<T extends ProcessType>

Propósito
Una interfaz para especificar las configuraciones de Hadoop para un trabajo.

MRJobConfig

Propósito
Para especificar configuraciones de Hadoop para cualquier trabajo MapReduce.

SparkJobConfig

Propósito
Especificar configuraciones de Hadoop para cualquier trabajo Spark.

JobDetail<T extends ProcessType>

Propósito
Almacena la información básica necesaria para crear un trabajo.

JobFactory

Propósito
La interfaz básica para especificar la creación de instancias de trabajo y especificar los detalles de los trabajos por crear.

JobPath

Propósito
La clase primaria para especificar los detalles de la fuente de entrada y el destino de salida para un trabajo.

OrcFilePath

To specify the input or output paths of ORC format files to run a job.

ProcessType

Propósito
La interfaz de marcado primaria para todos los tipos de proceso compatibles, como MapReduce y Spark.

MRProcessType

Propósito
Especificar el tipo de proceso de MapReduce para trabajos.

SparkProcessType

Propósito
Especificar el tipo de proceso de Spark para los trabajos.

ReferenceDataPath

Propósito
Para especificar la ruta de los datos de referencia de un trabajo.

ReportManager

Propósito
Una interfaz para recuperar las estadísticas de informe de un trabajo.

SimpleRule

Propósito
Un tipo de regla de la consolidación. Una regla simple puede utilizarse sola y como un componente de una regla conjunta. Consulte **ConjoinedRule** en la página 38.

Excepciones

JobException

Propósito
Maneja las excepciones de trabajo específicas y muestra los mensajes pertinentes.

Trabajos del módulo Advanced Matching

Módulo común de la API

AdvanceMatchDetail<T extends ProcessType>

Propósito
Especificar los detalles de un trabajo del módulo Advanced Matching.

AdvanceMatchFactory

Propósito
Una clase de fábrica única para crear instancias de trabajo del módulo Advanced Matching.

GroupbyOption<T extends ProcessType>

Propósito
Especificar la columna en que se realizará la agrupación para un trabajo de Advanced Matching.

GroupbyMROption

Propósito
Especificar la columna en que se realizará la agrupación para un trabajo de Advanced Matching MapReduce.

GroupbySparkOption

Propósito
Especificar la columna en que se realizará la agrupación para un trabajo Spark de Advanced Matching.

MatchKeySettings

Propósito
Mantiene una `List` de las claves de comparación para un trabajo Match Key Generator.

MatchRule

Propósito
Permite la creación de reglas de cruce para los trabajos de Advanced Matching.

Esto se realiza definiendo una jerarquía de nodos primarios y secundarios. Cada nodo se asigna a uno de los campos de entrada que se cruzará.

ChildMatchRule

Propósito
Especificar un nodo secundario de una regla de cruce, que realiza un mapa para un campo o determinados algoritmos y otras propiedades.

ParentMatchRule

Propósito
Para especificar un nodo principal de una regla de cruce, que es una agrupación lógica de otros nodos principales y nodos secundarios.

Situaciones especiales

Registros con la columna Agrupar por en blanco
Todos los registros cuya columna Agrupar por esté en blanco se marcan como registros malformados y se vuelcan en archivos diferentes en la carpeta de salida de HDFS.

Estos archivos malformados se denominan de la siguiente manera:

Registros malformados en los archivos candidatos	Los registros de los archivos candidatos con la columna Agrupar por en blanco se eliminan como registros malformados y se insertan en archivos que tengan la convención de nombre `malformedRecordsCandidate-m-<5 digit numeral>`. Por ejemplo, `malformedRecordsCandidate-m-00000`,`malformedRecordsCandidate-m-00001`. Esto se aplica a los trabajos de Interflow Match.
Registros malformados en archivos sospechosos	Los registros de archivos sospechosos con la columna Agrupar por en blanco se eliminan como registros malformados y se insertan en archivos que tengan la convención de nombre `malformedRecordsSuspect-m-<5 digit numeral>`. Por ejemplo,`malformedRecordsSuspect-m-00000`,`malformedRecordsSuspect-m-00001`. Esto se aplica a los trabajos de Interflow Match.

Registros malformados en archivos de entrada	Los registros de archivos de entrada con la columna Agrupar por en blanco se eliminan como registros malformados y se insertan en archivos que tengan la convención de nombre `malformedRecords-m-<5 digit numeral>`. Por ejemplo,`malformedRecords-m-00000`,`malformedRecords-m-00001`. Esto se aplica a los trabajos de Intraflow Match, Transactional Match, Best of Breed, Duplicate Synchronization y Filter.

Contadores de registros malformados

La cantidad de registros malformados en una ejecución de trabajo se almacena en los contadores:

- MALFORMED_CANDIDATE_RECORDS
- MALFORMED_SUSPECT_RECORDS
- MALFORMED_RECORDS

> **Nota:** Puede acceder a los valores de estos contadores invocando el método `getCounters()` de la instancia `AdvanceMatchFactory`.

Match Key Generator

Información general

El trabajo de Match Key Generator le permite generar claves de cruce.

> **Nota:** Para ejecutar una clave de cruce para los datos, debe ejecutar el trabajo de Match Key Generator antes de ejecutar cualquier otro trabajo.

Entidades API

MatchKeyGeneratorDetail

Propósito
Especificar los detalles de un trabajo de Match Key Generator.

Parámetros de entrada

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.
Configuración de claves de cruce	Una combinación de las columnas y los algoritmos que se aplicarán para generar la clave de cruce que se necesita para realizar el cruce. **Nota:** Se debe especificar al menos una clave de cruce. Puede especificar más de una clave de cruce, si es necesario.
Nombre de trabajo	El nombre del trabajo.

Columnas de salida

Además de las columnas de salida, las siguientes columnas se agregan mientras se genera la salida de un trabajo Match Key Generator:

Columna	Descripción	Valor de salida
MatchKey	La clave que se genera para identificar registros.	La clave generada depende de las columnas y los algoritmos seleccionados para generar la clave de cruce. **Nota:** La cantidad de columnas de clave de cruce con nombre asignado por el usuario generado en la salida depende de la configuración del trabajo.

Uso de un trabajo MapReduce de Match Key Generator

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Match Key Generator creando una instancia de `MatchKeyGeneratorDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Especifique las configuraciones de la clave de cruce para realizar el cruce creando y configurando una instancia de `MatchKeySettings`. Para obtener más información, consulte el código de muestra correspondiente.
   b) Cree una instancia de `MatchKeyGeneratorDetail` pasando una instancia del tipo `JobConfig` y la instancia `MatchKeySettings` creada como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.
   c) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `MatchKeyGeneratorDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.
   d) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `MatchKeyGeneratorDetail`.

      Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.
   e) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `MatchKeyGeneratorDetail`.
3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `MatchKeyGeneratorDetail` como un argumento.

   El método `createJob()` crea un trabajo y devuelve una`List` de las instancias de `ControlledJob`.

4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.

Uso de un trabajo Spark de Match Key Generator

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Match Key Generator creando una instancia de `MatchKeyGeneratorDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Especifique las configuraciones de la clave de cruce para realizar el cruce creando y configurando una instancia de `MatchKeySettings`. Para obtener más información, consulte el código de muestra correspondiente.
   b) Cree una instancia de `MatchKeyGeneratorDetail` pasando una instancia del tipo `JobConfig` y la instancia `MatchKeySettings` creada como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.
   c) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `MatchKeyGeneratorDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.
   d) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `MatchKeyGeneratorDetail`.

      Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.
   e) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `MatchKeyGeneratorDetail`.
3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `MatchKeyGeneratorDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

Interflow Match

Información general

El trabajo Interflow le permite generar una clave de cruce, agrupar los registros con el uso de la clave de cruce y realizar intercruces en los registros desde diferentes fuentes de datos.

Entidades API

InterMatchDetail

Propósito
Para especificar los detalles en un trabajo Interflow Match.

InterMatchComparisonOption

Propósito
Especificar las opciones de comparación mientras se define un trabajo Interflow Match, si el registro sospechoso debe compararse con todos los registros candidatos o con cualquier registro candidato seleccionado.

Parámetros de entrada

Parámetro	Descripción
Opción Agrupar por	Para un trabajo *MapReduce*, use los argumentos: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros. **Cantidad de tareas del reductor** La cantidad de tareas del reductor requeridas para agrupar los registros. Para un trabajo *Spark*, para crear una opción Agrupar por, use los argumentos: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros.
Regla de cruce	Defina tantas reglas primarias y secundarias como sea necesario para crear un objeto `MatchRule`. Para obtener más información, consulte **MatchRule** en la página 43.

Parámetro	Descripción
Archivo candidato	*Para archivos de texto:* **Ruta de acceso al archivo** La ruta del archivo de texto candidato en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo candidato. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo candidato. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos de encabezado del archivo candidato. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo sospechoso. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoque al constructor apropiado de `FilePath`. *Para archivos de formato ORC:* **Ruta de archivo ORC** La ruta del archivo de formato ORC de entrada en la plataforma Hadoop. **Importante:** Los archivos sospechoso y candidato deben tener el mismo formato. Ambos deben ser archivos de texto o ambos deben ser archivos de formato ORC. *Parámetros comunes:* **Asignaciones de campos** Un mapa de pares de clave/valor, con los nombres de las columnas existentes como las claves y los nombres de las columnas de salida deseadas como los valores.

Parámetro	Descripción
Archivo sospechoso	*Para archivos de texto:* **Ruta de acceso al archivo** La ruta del archivo de texto sospechoso en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo sospechoso. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo sospechoso. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos de encabezado del archivo sospechoso. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo sospechoso. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoque al constructor apropiado de `FilePath`. *Para archivos de formato ORC:* **Ruta de archivo ORC** La ruta del archivo de formato ORC de entrada en la plataforma Hadoop. *Parámetros comunes:* **Asignaciones de campos** Un mapa de pares de clave/valor, con los nombres de las columnas existentes como las claves y los nombres de las columnas de salida deseadas como los valores.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.
Configuración de claves de cruce	Una combinación de las columnas y los algoritmos que se aplicarán para generar la clave de cruce necesaria para realizar el cruce. **Nota:** Especifique solo una clave de cruce. **Atención:** Establezca las configuraciones de la clave de cruce solo si desea generar una clave de cruce antes de realizar el cruce.
Nombre de trabajo	El nombre del trabajo.
Columna de cruce inmediato	El nombre de la columna que se usará para el cruce inmediato de registros.

Parámetro	Descripción
Configuración del número de colección de registros únicos en cero	Configure en `verdadero` para establecer el número de colección de registros únicos en 0 (cero).
Opción de comparación	Le permite seleccionar una de las dos opciones: • Comparar el registro sospechoso con todos los registros candidatos: especifica si los registros únicos se deben mostrar en los resultados o no. • Comparar el registro sospechoso con el registro candidato seleccionado solamente: especifica la cantidad máxima de registros duplicados que se deben buscar y devolver.
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.

Columnas de salida

Además de las columnas de salida, las siguientes columnas se agregan mientras se genera la salida de un trabajo Interflow Match:

Columna	Descripción	Valor de salida
Número de colección	Identifica una colección de registros duplicados.	Los posibles valores son `0-0-1`, `0-0-2`, etc.
Clave de comparación inmediata	Indica si la coincidencia se obtuvo utilizando la clave de cruce inmediato.	1. Para un registro de candidato duplicado coincidente que utiliza una clave de comparación inmediata, la valor de salida es `Y`. 2. Para un registro de candidato duplicado coincidente, pero que no utiliza una clave de comparación inmediata, el valor de salida está en blanco. 3. Para un regfistro de candidato único de cruce que utiliza una clave de cruce inmediato, el valor de salida es `N`. 4. Para un registro sospechoso coincidente que utiliza una clave de comparación inmediata, el valor de salida está en blanco.
Tipo de origen del interflujo	Indica si el registro de entrada es un registro sospechoso o un registro de candidato.	Los posibles valores son `S` para un registro sospechoso, y `C` para un registro de candidato.

Columna	Descripción	Valor de salida
Tipo de registro de cruce	Identifica el tipo de registro de cruce de una colección.	Los posibles valores son `S` (registro sospechoso), `D` (registro duplicado) y `U` (registro único).
Calificación de cruce	Identifica la calificación general entre dos registros.	Los posibles valores varían de 0 (cero) a 100 para los registros duplicados y únicos, donde 0 indica un cruce poco satisfactorio y 100 indica un cruce de alta calidad. **Nota:** Para los registros sospechosos, el valor es 0.

Uso de un trabajo MapReduce de Interflow Match

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Interflow Match mediante la creación de una instancia de `InterMatchDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbyMROption** en la página 42 para especificar la columna por grupo y la cantidad de reductores que se necesitan.
   b) Genere las reglas de cruce para el trabajo creando una instancia de `MatchRule`.
   c) Cree una instancia de `InterMatchDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `MatchRule` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.
   d) Establezca los detalles del archivo candidato mediante el campo `candidateFilePath` de la instancia `InterMatchDetail`.

      Para un archivo candidato de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo candidato mediante la invocación del constructor apropiado. Para un archivo candidato ORC, cree una instancia de `OrcFilePath` con la ruta del archivo candidato ORC como argumento.
   e) Establezca los detalles del archivo sospechoso mediante el campo `suspectFilePath` de la instancia `InterMatchDetail`.

      Para un archivo sospechoso de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo sospechoso mediante la invocación del constructor apropiado. Para un archivo sospechoso ORC, cree una instancia de `OrcFilePath` con la ruta del archivo sospechoso ORC como argumento.

Importante: Los archivos sospechoso y candidato deben tener el mismo formato. Ambos deben ser archivos de texto o ambos deben ser archivos de formato ORC.

f) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `InterMatchDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

g) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `InterMatchDetail`.

h) Establezca la columna Cruce inmediato con el campo `expressMatchColumn` de la instancia `InterMatchDetail`, de ser necesario.

i) Establezca la bandera `collectionNumberZerotoUniqueRecords` de la instancia `InterMatchDetail` en `verdadero` para asignar el número de colección 0 (cero) a un registro único. El valor predeterminado es `verdadero`.

Si no desea asignar el número de colección cero a registros únicos, establezca esta bandera en `falso`.

j) Establezca la opción de comparación con el uso del campo `comparisonOption` de la instancia `InterMatchDetail`. En este campo, establezca el valor solicitado utilizando la clase **InterMatchComparisonOption** en la página 49 para seleccionar una de las dos opciones:

- **Comparar el registro sospechoso con todos los registros candidatos**: especifica si los registros únicos se deben mostrar en los resultados o no.
- **Comparar el registro sospechoso con el registro candidato seleccionado solamente**: especifica la cantidad máxima de registros duplicados que se deben buscar y devolver.

k) Establezca la bandera `compressOutput` de la instancia `InterMatchDetail` en `verdadero` para comprimir la salida del trabajo.

l) Si los datos de entrada no tienen clave de cruce, debe especificar las configuraciones de clave de cruce para ejecutar, en primer lugar, el trabajo de Match Key Generator a fin de generar las claves de cruce antes de ejecutar el trabajo Interflow Match.

Para generar las claves de cruce de los datos de entrada, especifique las configuraciones de clave de cruce creando y configurando una instancia de `MatchKeySettings` para generar una clave de cruce antes de realizar el trabajo Interflow Match. Establezca esta instancia mediante el campo `matchKeySettings` de la instancia `InterMatchDetail`.

Nota: Para averiguar cómo ajustar la configuración de claves de cruce, consulte las muestras de códigos.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `AdvanceMatchFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `InterMatchDetail` como un argumento.

El método `createJob()` crea un trabajo y devuelve una`List` de las instancias de `ControlledJob`.

4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `AdvanceMatchFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Interflow Match

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Intraflow Match mediante la creación de una instancia de `InterMatchDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbySparkOption** en la página 42 para especificar la columna por grupo.

   b) Genere las reglas de cruce para el trabajo creando una instancia de `MatchRule`.
   c) Cree una instancia de `InterMatchDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `MatchRule` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

   d) Establezca los detalles del archivo candidato mediante el campo `candidateFilePath` de la instancia `InterMatchDetail`.

      Para un archivo candidato de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo candidato mediante la invocación del constructor apropiado. Para un archivo candidato ORC, cree una instancia de `OrcFilePath` con la ruta del archivo candidato ORC como argumento.

   e) Establezca los detalles del archivo sospechoso mediante el campo `suspectFilePath` de la instancia `InterMatchDetail`.

      Para un archivo sospechoso de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo sospechoso mediante la invocación del constructor apropiado. Para un archivo sospechoso ORC, cree una instancia de `OrcFilePath` con la ruta del archivo sospechoso ORC como argumento.

      Importante: Los archivos sospechoso y candidato deben tener el mismo formato. Ambos deben ser archivos de texto o ambos deben ser archivos de formato ORC.

   f) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `InterMatchDetail`.

      Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de

salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

g) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `InterMatchDetail`.

h) Establezca la columna Cruce inmediato con el campo `expressMatchColumn` de la instancia `InterMatchDetail`, de ser necesario.

i) Establezca la bandera `collectionNumberZerotoUniqueRecords` de la instancia `InterMatchDetail` en `verdadero` para asignar el número de colección 0 (cero) a un registro único. El valor predeterminado es `verdadero`.

Si no desea asignar el número de colección cero a registros únicos, establezca esta bandera en `falso`.

j) Establezca la opción de comparación con el uso del campo `comparisonOption` de la instancia `InterMatchDetail`. En este campo, establezca el valor solicitado utilizando la clase **InterMatchComparisonOption** en la página 49 para seleccionar una de las dos opciones:

- **Comparar el registro sospechoso con todos los registros candidatos**: especifica si los registros únicos se deben mostrar en los resultados o no.
- **Comparar el registro sospechoso con el registro candidato seleccionado solamente**: especifica la cantidad máxima de registros duplicados que se deben buscar y devolver.

k) Establezca la bandera `compressOutput` de la instancia `InterMatchDetail` en `verdadero` para comprimir la salida del trabajo.

l) Si los datos de entrada no tienen clave de cruce, debe especificar las configuraciones de clave de cruce para ejecutar, en primer lugar, el trabajo de Match Key Generator a fin de generar las claves de cruce antes de ejecutar el trabajo Interflow Match.

Para generar las claves de cruce de los datos de entrada, especifique las configuraciones de clave de cruce creando y configurando una instancia de `MatchKeySettings` para generar una clave de cruce antes de realizar el trabajo Interflow Match. Establezca esta instancia mediante el campo `matchKeySettings` de la instancia `InterMatchDetail`.

> **Nota:** Para averiguar cómo ajustar la configuración de claves de cruce, consulte las muestras de códigos.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `InterMatchDetail` como un argumento.
El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.
4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Intraflow Match

Información general

El trabajo Interflow le permite generar una clave de cruce, agrupar los registros que usan la clave de cruce y realizar intercruces en el registro de la misma fuente de datos.

Entidades API

IntraMatchDetail

Propósito

Para especificar los detalles de un trabajo Intraflow Match.

Parámetros de entrada

Parámetro	Descripción
Opción Agrupar por	Para un trabajo *MapReduce*, use los argumentos: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros. **Cantidad de tareas del reductor** La cantidad de tareas del reductor requeridas para agrupar los registros. Para un trabajo *Spark*, para crear una opción Agrupar por, use los argumentos: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros.
Regla de cruce	Defina tantas reglas primarias y secundarias como sea necesario para crear un objeto `MatchRule`. Para obtener más información, consulte **MatchRule** en la página 43.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.
Nombre de trabajo	El nombre del trabajo.
Columna de cruce inmediato	El nombre de la columna que se usará para el cruce inmediato de registros.
Configuración del número de colección de registros únicos en cero	Configure en `verdadero` para establecer el número de colección de registros únicos en 0 (cero).
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.

Parámetro	Descripción
Configuración de claves de cruce	Una combinación de las columnas y los algoritmos que se aplicarán para generar la clave de cruce necesaria para realizar el cruce. **Nota:** Especifique solo una clave de cruce. **Atención:** Establezca las configuraciones de la clave de cruce solo si desea generar una clave de cruce antes de realizar el cruce.

Columnas de salida

Además de las columnas de salida, las siguientes columnas se agregan mientras se genera la salida de un trabajo Intraflow Match:

Columna	Descripción	Valor de salida
Número de colección	Identifica una colección de registros duplicados.	Los posibles valores son `0-0-1`, `0-0-2`, etc.
Cruce inmediato identificado	Indica si la coincidencia se obtuvo utilizando la clave de cruce inmediato.	1. Para un registro de candidato duplicado de cruce que utiliza una clave de cruce inmediato, la valor de salida es `Y`. 2. Para un registro de candidato duplicado de cruce, pero que no utiliza una clave de cruce inmediato, el valor de salida está en blanco. 3. Para un registro de candidato único con cruce que utiliza una clave de cruce inmediato, el valor de salida está en blanco. 4. Para un registro sospechoso coincidente que utiliza una clave de cruce inmediato, el valor de salida está en blanco.
Tipo de registro de cruce	Identifica el tipo de registro de cruce de una colección.	Los posibles valores son `S` (registro sospechoso), `D` (registro duplicado) y `U` (registro único).
Calificación de cruce	Identifica la calificación general entre dos registros.	Los posibles valores varían de 0 (cero) a 100 para los registros duplicados y únicos, donde 0 indica un cruce poco satisfactorio y 100 indica un cruce de alta calidad. **Nota:** Para los registros sospechosos, el valor es 0.

Uso de un trabajo MapReduce de Intraflow Match

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Intraflow Match mediante la creación de una instancia de `IntraMatchDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbyMROption** en la página 42 para especificar la columna por grupo y la cantidad de reductores que se necesitan.
   b) Genere las reglas de cruce para el trabajo creando una instancia de `MatchRule`.
   c) Cree una instancia de `IntraMatchDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `MatchRule` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.
   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `IntraMatchDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.
   e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `IntraMatchDetail`.

      Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.
   f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `IntraMatchDetail`.
   g) Establezca la columna Cruce inmediato con el campo `expressMatchColumn` de la instancia `IntraMatchDetail`, de ser necesario.
   h) Establezca la bandera `collectionNumberZerotoUniqueRecords` de la instancia `IntraMatchDetail` en `verdadero` para asignar el número de colección 0 (cero) a un registro único. El valor predeterminado es `verdadero`.

      Si no desea asignar el número de colección cero a registros únicos, establezca esta bandera en `falso`.
   i) Establezca la bandera `compressOutput` de la instancia `IntraMatchDetail` en `verdadero` para comprimir la salida del trabajo.
   j) Si los datos de entrada no tienen clave de cruce, debe especificar las configuraciones de clave de cruce para ejecutar, en primer lugar, el trabajo de Match Key Generator a fin de generar las claves de cruce antes de ejecutar el trabajo Intraflow Match.

Para generar las claves de cruce de los datos de entrada, especifique las configuraciones de clave de cruce creando y configurando una instancia de `MatchKeySettings` para generar una clave de cruce antes de realizar el trabajo Intraflow Match. Establezca esta instancia mediante el campo `matchKeySettings` de la instancia `IntraMatchDetail`.

Nota: Para averiguar cómo ajustar la configuración de claves de cruce, consulte las muestras de códigos.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `AdvanceMatchFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `IntraMatchDetail` como un argumento.

   El método `createJob()` crea un trabajo y devuelve una`List` de las instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `AdvanceMatchFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Intraflow Match

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Intraflow Match mediante la creación de una instancia de `IntraMatchDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbySparkOption** en la página 42 para especificar la columna por grupo.
   b) Genere las reglas de cruce para el trabajo creando una instancia de `MatchRule`.
   c) Cree una instancia de `IntraMatchDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `MatchRule` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.
   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `IntraMatchDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.
   e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `IntraMatchDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `IntraMatchDetail`.

g) Establezca la columna Cruce inmediato con el campo `expressMatchColumn` de la instancia `IntraMatchDetail`, de ser necesario.

h) Establezca la bandera `collectionNumberZerotoUniqueRecords` de la instancia `IntraMatchDetail` en `verdadero` para asignar el número de colección 0 (cero) a un registro único. El valor predeterminado es `verdadero`.

Si no desea asignar el número de colección cero a registros únicos, establezca esta bandera en `falso`.

i) Establezca la bandera `compressOutput` de la instancia `IntraMatchDetail` en `verdadero` para comprimir la salida del trabajo.

j) Si los datos de entrada no tienen clave de cruce, debe especificar las configuraciones de clave de cruce para ejecutar, en primer lugar, el trabajo de Match Key Generator a fin de generar las claves de cruce antes de ejecutar el trabajo Intraflow Match.

Para generar las claves de cruce de los datos de entrada, especifique las configuraciones de clave de cruce creando y configurando una instancia de `MatchKeySettings` para generar una clave de cruce antes de realizar el trabajo Intraflow Match. Establezca esta instancia mediante el campo `matchKeySettings` de la instancia `IntraMatchDetail`.

> **Nota:** Para averiguar cómo ajustar la configuración de claves de cruce, consulte las muestras de códigos.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `IntraMatchDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Transactional Match

Información general

El trabajo Transactional Match le permite cruzar los registros sospechosos con los registros candidatos de un grupo de registros para identificar duplicados.

Entidades API

TransactionalMatchDetail

Propósito
Para especificar los detalles de un trabajo Transactional Match.

Parámetros de entrada

Parámetro	Descripción
Opción Agrupar por	Para un trabajo *MapReduce*, use los argumentos: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros. **Cantidad de tareas del reductor** La cantidad de tareas del reductor requeridas para agrupar los registros. Para un trabajo *Spark*, para crear una opción Agrupar por, use los argumentos: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros.
Regla de cruce	Defina tantas reglas primarias y secundarias como sea necesario para crear un objeto `MatchRule`. Para obtener más información, consulte **MatchRule** en la página 43.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.
Arrojar candidatos únicos	Marcar para indicar si los candidatos únicos deben arrojarse como parte de la salida.
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.
Configuración de claves de cruce	Una combinación de las columnas y los algoritmos que se aplicarán para generar la clave de cruce necesaria para realizar el cruce. **Nota:** Especifique solo una clave de cruce. **Atención:** Establezca las configuraciones de la clave de cruce solo si desea generar una clave de cruce antes de realizar el cruce.

Columnas de salida

Además de las columnas de entrada, se agregan las siguientes columnas mientras se genera la salida de un trabajo de Transactional Match:

Parámetro	Descripción	Valor de salida
Tipo de registro de cruce	Identifica el tipo de registro de cruce de una colección.	Los posibles valores son `S` (registro sospechoso), `D` (registro duplicado) y `U` (registro único).
Calificación de cruce	Identifica la calificación general entre dos registros.	Los posibles valores varían de 0 (cero) a 100 para los registros duplicados y únicos, donde 0 indica un cruce poco satisfactorio y 100 indica un cruce de alta calidad. **Nota:** Para los registros sospechosos, el valor es 0.
Tiene duplicados	Indica si los registros sospechosos tienen duplicados o no.	En el caso de los registros sospechosos, los posibles valores de salida son: • `Y` (si hay duplicados) O • `N` (si no hay duplicados) En el caso de los registros duplicados, el valor de salida es `D`. En el caso de los registros únicos, el valor de salida es `U`.

Uso de un trabajo MapReduce de Transactional Match

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Transactional Match mediante la creación de una instancia de `TransactionalMatchDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbyMROption** en la página 42 para especificar la columna por grupo y la cantidad de reductores que se necesitan.
   b) Genere las reglas de cruce para el trabajo creando una instancia de `MatchRule`.
   c) Cree una instancia de `TransactionalMatchDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `MatchRule` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.

d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `TransactionalMatchDetail`.

Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `TransactionalMatchDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `TransactionalMatchDetail`.

g) Establezca la bandera `returnUniqueCandidates` de la instancia `TransactionalMatchDetail` en `verdadero` para obtener los registros de candidato único en la salida. El valor predeterminado es `verdadero`.

h) Establezca la bandera `compressOutput` de la instancia `TransactionalMatchDetail` en `verdadero` para comprimir la salida del trabajo.

i) Si los datos de entrada no tienen clave de cruce, debe especificar las configuraciones de clave de cruce para ejecutar, en primer lugar, el trabajo de Match Key Generator a fin de generar las claves de cruce antes de ejecutar el trabajo Transactional Match.

Para generar las claves de cruce de los datos de entrada, especifique las configuraciones de clave de cruce mediante la creación y la configuración de una instancia de`MatchKeySettings` para generar una clave de cruce antes de realizar el trabajo Transactional Match. Establezca esta instancia mediante el campo `matchKeySettings` de la instancia `TransactionalMatchDetail`.

> **Nota:** Para averiguar cómo ajustar la configuración de claves de cruce, consulte las muestras de códigos.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `AdvanceMatchFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `TransactionalMatchDetail` como un argumento.

   El método `createJob()` crea un trabajo y devuelve una`List` de las instancias de `ControlledJob`.

4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `AdvanceMatchFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Transactional Match

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Transactional Match mediante la creación de una instancia de `TransactionalMatchDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbySparkOption** en la página 42 para especificar la columna por grupo.
   b) Genere las reglas de cruce para el trabajo creando una instancia de `MatchRule`.
   c) Cree una instancia de `TransactionalMatchDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `MatchRule` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.
   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `TransactionalMatchDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.
   e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `TransactionalMatchDetail`.

      Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.
   f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `TransactionalMatchDetail`.
   g) Establezca la bandera `returnUniqueCandidates` de la instancia `TransactionalMatchDetail` en `verdadero` para obtener los registros de candidato único en la salida. El valor predeterminado es `verdadero`.
   h) Establezca la bandera `compressOutput` de la instancia `TransactionalMatchDetail` en `verdadero` para comprimir la salida del trabajo.
   i) Si los datos de entrada no tienen clave de cruce, debe especificar las configuraciones de clave de cruce para ejecutar, en primer lugar, el trabajo de Match Key Generator a fin de generar las claves de cruce antes de ejecutar el trabajo Transactional Match.

      Para generar las claves de cruce de los datos de entrada, especifique las configuraciones de clave de cruce mediante la creación y la configuración de una instancia de`MatchKeySettings` para generar una clave de cruce antes de realizar el trabajo Transactional Match. Establezca

esta instancia mediante el campo `matchKeySettings` de la instancia `TransactionalMatchDetail`.

Nota: Para averiguar cómo ajustar la configuración de claves de cruce, consulte las muestras de códigos.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `TransactionalMatchDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.
4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Best of Breed

Información general

El trabajo Best of Breed consolida los registros duplicados mediante la selección de los mejores datos en una colección de registros duplicados y la creación de un registro consolidado con el uso de los mejores datos.

Entidades API

BestOfBreedConfiguration

Para especificar las reglas de la consolidación y de la plantilla para realizar el trabajo de consolidación de Best of Breed.

BestofBreedDetail

Propósito

Para especificar los detalles de un trabajo de consolidación de Best of Breed.

Parámetros de entrada

Parámetro	Descripción
Opción Agrupar por	Especifique el campo mediante el cual se creará un único registro Best of Breed combinando un grupo de registros similares. Se crea un registro Best of Breed para cada grupo de registros. Para un trabajo *MapReduce*, use los siguientes argumentos: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros. **Cantidad de tareas del reductor** La cantidad de tareas del reductor requeridas para agrupar los registros. For a *Spark* job, pass the arguments: **Columna Agrupar por** El nombre de la columna con la que agrupará los registros.
Configuración de Best of Breed	Defina las reglas de la consolidación y de la plantilla con las que se creará el registro Best of Breed para cada grupo de registros similares.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.

Columnas de salida

Además de las columnas de salida, las siguientes columnas se agregan mientras se genera la salida de un trabajo Best of Breed:

Parámetro	Descripción	Valor de salida
Tipo de registro de colección	Identifica la plantilla y los registros Best of Breed en una colección de registros duplicados.	Si se define un registro de plantilla, los valores posibles son: **Principal** Si el registro es el registro de plantilla seleccionado en una colección. **Secundario** Si el registro no es el registro de plantilla seleccionado en una colección. **BestOfBreed** Si el registro es el registro Best of Breed que se acaba de crear en la colección. Si no se define un registro de platilla, el único valor posible es **BestOfBreed**.

Nota: Otras columnas de entrada, además del **Tipo de registro de colección**, se muestran únicamente si están definidas mientras se crean las condiciones de consolidación para la configuración de Best of Breed.

Uso de un trabajo MapReduce de Best of Breed

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Best of Breed mediante la creación de una instancia de `BestofBreedDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

   Utilice una instancia de **GroupbyMROption** en la página 42 para especificar la columna por grupo y la cantidad de reductores que se necesitan.

   b) Genere las reglas de la consolidación y la plantilla para el trabajo creando una instancia de `BestOfBreedConfiguration`. Dentro de esta instancia:

      1. Defina el registro de la plantilla para la consolidación mediante el uso de una instancia de `ConsolidationCondition`, que se compone de las instancias de `ConsolidationRule`.

2. Defina las condiciones de consolidación mediante el uso de las instancias de `ConsolidationCondition` y la conexión de las condiciones con el uso de operadores lógicos.

   Cada instancia de`ConsolidationCondition` se define con el uso de una instancia `ConsolidationRule` y su instancia `ConsolidationAction` correspondiente.

   Nota: Cada instancia de`ConsolidationRule` puede definirse con el uso de una instancia única de `SimpleRule` o con el uso de una jerarquía de instancias secundarias `SimpleRule` y de instancias anidadas`ConjoinedRule` unidas con los operadores lógicos. Consulte **Enum JoinType** en la página 196 y **Enum Operation** en la página 195.

c) Cree una instancia de `BestofBreedDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `BestOfBreedConfiguration` creada anteriormente como los argumentos para su constructor.

   El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.

d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `BestofBreedDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `BestofBreedDetail`.

   Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `BestofBreedDetail`.

g) Establezca la bandera `compressOutput` de la instancia `BestofBreedDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `AdvanceMatchFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `BestofBreedDetail` como un argumento.

   El método `createJob()` crea un trabajo y devuelve una`List` de las instancias de `ControlledJob`.

4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `AdvanceMatchFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Best of Breed

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Best of Breed mediante la creación de una instancia de `BestofBreedDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbySparkOption** en la página 42 para especificar la columna por grupo.

   b) Genere las reglas de la consolidación y la plantilla para el trabajo creando una instancia de `BestOfBreedConfiguration`. Dentro de esta instancia:

      1. Defina el registro de la plantilla para la consolidación mediante el uso de una instancia de `ConsolidationCondition`, que se compone de las instancias de `ConsolidationRule`.
      2. Defina las condiciones de consolidación mediante el uso de las instancias de `ConsolidationCondition` y la conexión de las condiciones con el uso de operadores lógicos.

         Cada instancia de`ConsolidationCondition` se define con el uso de una instancia `ConsolidationRule` y su instancia `ConsolidationAction` correspondiente.

         Nota: Cada instancia de`ConsolidationRule` puede definirse con el uso de una instancia única de `SimpleRule` o con el uso de una jerarquía de instancias secundarias `SimpleRule` y de instancias anidadas`ConjoinedRule` unidas con los operadores lógicos. Consulte **Enum JoinType** en la página 196 y **Enum Operation** en la página 195.

   c) Cree una instancia de `BestofBreedDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `BestOfBreedConfiguration` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `BestofBreedDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

   e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `BestofBreedDetail`.

      Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de

salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `BestofBreedDetail`.

g) Establezca la bandera `compressOutput` de la instancia `BestofBreedDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `BestofBreedDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Duplicate Synchronization

Información general

El trabajo Duplicate Synchronization le permite determinar cuáles son los campos de una colección de registros que deben copiarse en los campos correspondientes de todos los registros de la colección.

Entidades API

DuplicateSynchronizationConfiguration

Especificar las reglas de la consolidación a fin de realizar el trabajo de consolidación de Duplicate Synchronization.

DuplicateSyncDetail

Propósito

Para especificar los detalles de un trabajo de consolidación de Duplicate Synchronization.

Parámetros de entrada

Parámetro	Descripción
Opción Agrupar por	Especifica el campo que se usará para crear grupos de registros para sincronizar. Para un trabajo *MapReduce*, use los siguientes argumentos: **Columna GroupBy** El nombre de la columna que usa los registros que se van a agrupar. **Cantidad de tareas del reductor** La cantidad de tareas del reductor necesarias para agrupar los registros. For a *Spark* job, to create a Group-By option pass the arguments: **Columna Agrupar por** El nombre de la columna que usa los registros que se van a agrupar. **Nota:** Si no hay un grupo en la salida, entonces configure este parámetro en nulo. En este caso, todos los datos se consideran como un único grupo.
Configuración de Duplicate Synchronization	Las reglas en función de las cuales los campos de un registro se copian en los otros registros de una colección.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.

Columnas de salida

Según las condiciones de consolidación definidas en el parámetro de entrada *Configuración de Duplicate Synchronization*, se pueden agregar columnas a la salida, además de las columnas de entrada, según sea necesario.

Uso de un trabajo MapReduce de Duplicate Synchronization

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Duplicate Synchronization mediante la creación de una instancia de `DuplicateSyncDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

Utilice una instancia de **GroupbyMROption** en la página 42 para especificar la columna por grupo y la cantidad de reductores que se necesitan.

b) Genere las condiciones de consolidación para el trabajo mediante la creación de una instancia de `DuplicateSynchronizationConfiguration`. Dentro de esta instancia, defina las condiciones de consolidación mediante el uso de las instancias de `ConsolidationCondition` y la conexión de las condiciones con el uso de operadores lógicos.

Cada instancia de`ConsolidationCondition` se define con el uso de una instancia `ConsolidationRule` y su instancia `ConsolidationAction` correspondiente.

> **Nota:** Cada instancia de`ConsolidationRule` puede definirse con el uso de una instancia única de `SimpleRule` o con el uso de una jerarquía de instancias secundarias `SimpleRule` y de instancias anidadas`ConjoinedRule` unidas con los operadores lógicos. Consulte **Enum JoinType** en la página 196 y **Enum Operation** en la página 195.

c) Cree una instancia de `DuplicateSyncDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `DuplicateSynchronizationConfiguration` creada anteriormente como los argumentos para su constructor.

El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.

d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `DuplicateSyncDetail`.

Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `DuplicateSyncDetail`.

For a text output file, create an instance of`FilePath` with the relevant details of the output file by invoking the appropriate constructor. For an ORC output file, create an instance of `OrcFilePath` with the path of the ORC output file as the argument.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `DuplicateSyncDetail`.

g) Establezca la bandera `compressOutput` de la instancia `DuplicateSyncDetail` en `verdadero` para comprimir la salida del trabajo.

3. Cree el trabajo usando la instancia anteriormente creada de `AdvanceMatchFactory` para invocar su método `createJob()`. Aquí, pase la instancia anterior de `DuplicateSyncDetail` como un argumento.
El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.

5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `AdvanceMatchFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Duplicate Synchronization

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Duplicate Synchronization mediante la creación de una instancia de `DuplicateSyncDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbySparkOption** en la página 42 para especificar la columna por grupo.

   b) Genere las condiciones de consolidación para el trabajo mediante la creación de una instancia de `DuplicateSynchronizationConfiguration`. Dentro de esta instancia, defina las condiciones de consolidación mediante el uso de las instancias de `ConsolidationCondition` y la conexión de las condiciones con el uso de operadores lógicos.

      Cada instancia de`ConsolidationCondition` se define con el uso de una instancia `ConsolidationRule` y su instancia `ConsolidationAction` correspondiente.

      > **Nota:** Cada instancia de`ConsolidationRule` puede definirse con el uso de una instancia única de `SimpleRule` o con el uso de una jerarquía de instancias secundarias `SimpleRule` y de instancias anidadas`ConjoinedRule` unidas con los operadores lógicos. Consulte **Enum JoinType** en la página 196 y **Enum Operation** en la página 195.

   c) Cree una instancia de `DuplicateSyncDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `DuplicateSynchronizationConfiguration` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `DuplicateSyncDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

   e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `DuplicateSyncDetail`.

For a text output file, create an instance of`FilePath` with the relevant details of the output file by invoking the appropriate constructor. For an ORC output file, create an instance of `OrcFilePath` with the path of the ORC output file as the argument.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `DuplicateSyncDetail`.

g) Establezca la bandera `compressOutput` de la instancia `DuplicateSyncDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `DuplicateSyncDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Filtro

Información general

El trabajo Filter conserva o elimina registros de un grupo de registros según las reglas que especifique.

Entidades API

FilterConfiguration

Especificar las reglas de la consolidación para realizar el trabajo de consolidación de Filter.

FilterDetail

Propósito

Para especificar los detalles de un trabajo de consolidación de Filter.

Parámetros de entrada

Parámetro	Descripción
Opción Agrupar por	Especifica el campo que se usará para crear grupos de registros para filtrar. El trabajo de Filter retiene uno o más registros de cada grupo. Para un trabajo *MapReduce*, use los siguientes argumentos: **Columna GroupBy** El nombre de la columna que usa los registros que se van a agrupar. **Cantidad de tareas del reductor** La cantidad de tareas del reductor necesarias para agrupar los registros. For a *Spark* job, to create a Group-By option pass the arguments: **Columna Agrupar por** El nombre de la columna que usa los registros que se van a agrupar. **Nota:** Si no hay un grupo en la salida, entonces configure este parámetro en nulo. En este caso, todos los datos se consideran como un único grupo.
Configuración de Filter	Define las condiciones de consolidación en función del trabajo que retiene uno o más registros de cada grupo.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.

Columnas de salida

Las columnas de salida son iguales a las columnas de entrada. No se agregan columnas adicionales en la salida.

Uso de trabajo MapReduce de Filter

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Filter mediante la creación de una instancia de `FilterDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

Utilice una instancia de **GroupbyMROption** en la página 42 para especificar la columna por grupo y la cantidad de reductores que se necesitan.

b) Genere las reglas de la consolidación para el trabajo creando una instancia de `FilterConfiguration`. Dentro de esta instancia, defina las condiciones de consolidación mediante el uso de las instancias de `ConsolidationCondition` y la conexión de las condiciones con el uso de operadores lógicos.

Cada instancia de`ConsolidationCondition` se define con el uso de una instancia `ConsolidationRule` y su instancia `ConsolidationAction` correspondiente.

> **Nota:** Cada instancia de`ConsolidationRule` puede definirse con el uso de una instancia única de `SimpleRule` o con el uso de una jerarquía de instancias secundarias `SimpleRule` y de instancias anidadas`ConjoinedRule` unidas con los operadores lógicos. Consulte **Enum JoinType** en la página 196 y **Enum Operation** en la página 195.

c) Cree una instancia de `FilterDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `FilterConfiguration` creada anteriormente como los argumentos para su constructor.

El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.

d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `FilterDetail`.

Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `FilterDetail`.

For a text output file, create an instance of`FilePath` with the relevant details of the output file by invoking the appropriate constructor. For an ORC output file, create an instance of `OrcFilePath` with the path of the ORC output file as the argument.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `FilterDetail`.

g) Establezca la bandera `compressOutput` de la instancia `FilterDetail` en `verdadero` para comprimir la salida del trabajo.

3. Cree el trabajo usando la instancia anteriormente creada de `AdvanceMatchFactory` para invocar su método `createJob()`. Aquí, pase la instancia anterior de `FilterDetail` como un argumento.
El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `AdvanceMatchFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de trabajo Filter Spark

1. Cree una instancia de `AdvanceMatchFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Filter mediante la creación de una instancia de `FilterDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Especifique la columna con la cual se agruparán los registros creando una instancia de `GroupbyOption`.

      Utilice una instancia de **GroupbySparkOption** en la página 42 para especificar la columna por grupo.

   b) Genere las reglas de la consolidación para el trabajo creando una instancia de `FilterConfiguration`. Dentro de esta instancia, defina las condiciones de consolidación mediante el uso de las instancias de `ConsolidationCondition` y la conexión de las condiciones con el uso de operadores lógicos.

      Cada instancia de`ConsolidationCondition` se define con el uso de una instancia `ConsolidationRule` y su instancia `ConsolidationAction` correspondiente.

      > **Nota:** Cada instancia de`ConsolidationRule` puede definirse con el uso de una instancia única de `SimpleRule` o con el uso de una jerarquía de instancias secundarias `SimpleRule` y de instancias anidadas`ConjoinedRule` unidas con los operadores lógicos. Consulte **Enum JoinType** en la página 196 y **Enum Operation** en la página 195.

   c) Cree una instancia de `FilterDetail` pasando una instancia del tipo `JobConfig`, la instancia `GroupbyOption` creada y la instancia `FilterConfiguration` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `FilterDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

   e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `FilterDetail`.

      For a text output file, create an instance of`FilePath` with the relevant details of the output file by invoking the appropriate constructor. For an ORC output file, create an instance of `OrcFilePath` with the path of the ORC output file as the argument.

   f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `FilterDetail`.
   g) Establezca la bandera `compressOutput` de la instancia `FilterDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`AdvanceMatchFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `FilterDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.
4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Trabajos del módulo Data Normalization

Módulo común de la API

DataNormalizationDetail<T extends ProcessType>

Propósito
Para especificar los detalles de un trabajo del módulo Data Normalization.

DataNormalizationFactory

Propósito
Una clase de fábrica única para crear instancias de trabajo del módulo Data Normalization.

Table Lookup

Información general

El trabajo Table Lookup estandariza los términos en comparación con un formato validado anteriormente de ese término y aplica la versión estándar.

Entidades API

AbstractTableLookupRule

Propósito
Para especificar la regla que se va a utilizar en Table Lookup.

Categorizar

Propósito
Especificar la regla Categorizar para un trabajo Table Lookup.

Identificar

Propósito
Para especificar la regla Identificar para un trabajo Table Lookup.

Estandarizar

Propósito
Especificar la regla Estandarizar para un trabajo Table Lookup.

TableLookupDetail

Propósito
Especificar los detalles de un trabajo Table Lookup.

TableLookupConfiguration

Propósito
Estandarizar los términos en comparación con una forma validada anterior de dicho término y aplicar la versión estandarizada a todos los registros.

Parámetros de entrada

Parámetro	Descripción
Configuración de Table Lookup	Para estandarizar los términos en comparación con un formato del término en cuestión validado anteriormente y aplicar la versión estándar a todos los registros. Las reglas pueden ser del tipo`Standardize`, `Categorize` o`Identify`.
Ruta de acceso de datos de referencia	Para especificar los detalles de la ruta de acceso de Datos de referencia.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.

Columnas de salida

Además de las columnas de entrada, se agregan las siguientes columnas mientras se genera la salida de un trabajo de Table Lookup:

Columna	Descripción	Valor de salida
Destino	Para las opciones de regla `Standardize` y `Categorize`, esta columna de salida se agrega si un nuevo nombre de columna, no presente en la entrada, se especifica como la columna de destino. El nombre de la columna será el que usted ingrese. **Nota:** Para la columna de destino, puede seleccionar una columna de origen actual o escribir un nuevo nombre.	El valor estandarizado de las columnas de origen cruzado con los datos de la tabla.
Término de estandarización identificado	Indica si el término estandarizado se identificó o no.	Los valores posibles son `Sí` y `No`.

Uso de un trabajo MapReduce de Table Lookup

1. Cree una instancia de `DataNormalizationFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Table Lookup mediante la creación de una instancia de `TableLookupDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Configure las reglas de Table Lookup mediante la creación de una instancia de`TableLookupConfiguration`. Dentro de esta instancia:

      Agregue una instancia de tipo `AbstractTableLookupRule`. Esta instancia `AbstractTableLookupRule` debe definirse con el uso de una de estas clases: `Standardize`,`Categorize` o `Identify`, que corresponde a la categoría de regla deseada de Table Lookup.
   b) Establezca los detalles de la ruta de los datos de referencia y el tipo de ubicación creando una instancia de `ReferenceDataPath`. Consulte **Enum ReferenceDataPathLocation** en la página 195.
   c) Cree una instancia de `TableLookupDetail`, mediante el paso de una instancia de tipo `JobConfig` y la `TableLookupConfiguration` y las instancias`ReferenceDataPath` creadas con anterioridad como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.
   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `TableLookupDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un

archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `TableLookupDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `TableLookupDetail`.

g) Establezca la bandera `compressOutput` de la instancia `TableLookupDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `DataNormalizationFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `TableLookupDetail` como un argumento.
El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `DataNormalizationFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Table Lookup

1. Cree una instancia de `DataNormalizationFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Table Lookup mediante la creación de una instancia de `TableLookupDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.

a) Configure las reglas de Table Lookup mediante la creación de una instancia de`TableLookupConfiguration`. Dentro de esta instancia:

Agregue una instancia de tipo `AbstractTableLookupRule`. Esta instancia `AbstractTableLookupRule` debe definirse con el uso de una de estas clases: `Standardize`,`Categorize` o `Identify`, que corresponde a la categoría de regla deseada de Table Lookup.

b) Establezca los detalles de la ruta de los datos de referencia y el tipo de ubicación creando una instancia de `ReferenceDataPath`. Consulte **Enum ReferenceDataPathLocation** en la página 195.

c) Cree una instancia de `TableLookupDetail`, mediante el paso de una instancia de tipo `JobConfig` y la`TableLookupConfiguration` y las instancias`ReferenceDataPath` creadas con anterioridad como los argumentos para su constructor.

El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `TableLookupDetail`.

Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `TableLookupDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `TableLookupDetail`.

g) Establezca la bandera `compressOutput` de la instancia `TableLookupDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`DataNormalizationFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `TableLookupDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Advanced Transformer

Información general

El trabajo Advanced Transformer explora y divide las cadenas de datos en múltiples campos por medio de tablas o expresiones regulares. Esta herramienta extrae un término específico o una cantidad determinada de palabras situadas a la derecha o la izquierda de un término.

Entidades API

AbstractAdvancedTransformerRules

Propósito

Clase principal para especificar las reglas para un trabajo Advanced Transformer.

AdvancedTransformerDetail

Propósito

Especificar los detalles de un trabajo Advanced Transformer.

AdvancedTransformerConfiguration

Propósito
Explorar y dividir las cadenas de datos en múltiples campos por medio de tablas o expresiones regulares.

RegularExpressionExtraction

Propósito
Especificar las reglas a fin de extraer datos con el uso de expresiones regulares.

RegularExpressionGroupItem

Propósito
Especificar una parte de una expresión regular principal. Cada parte de una expresión regular principal puede almacenarse en un campo de salida diferente.

TableDataExtraction

Propósito
Definir las reglas de extracción de datos de la tabla.

Parámetros de entrada

Parámetro	Descripción
Configuración de Advanced Transformer	Para explorar y dividir las cadenas de datos en múltiples campos por medio de tablas o expresiones regulares. Permite la extracción de un término específico o una cantidad determinada de palabras situadas a la derecha o la izquierda de un término. Los datos extraídos y sin extraer se colocan en un campo nuevo o ya existente. Las reglas de Advanced Transformer pueden definirse con el uso de una instancia del tipo`AdvancedTransformerConfiguration`. Esta instancia debe ser una instancia de `TableDataExtraction` o`RegularExpressionExtraction`.
Ruta de acceso de datos de referencia	Para especificar los detalles de la ruta de acceso de Datos de referencia.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.

Columnas de salida

Además de las columnas de entrada, se agregan las siguientes columnas mientras se genera la salida de un trabajo de Advanced Transformer:

Columna	Descripción	Valor de salida
Datos sin extraer	Esta columna de salida se agrega si un nuevo nombre de columna, no presente en la entrada, se especifica como la columna Datos sin extraer. El nombre de la columna será el que usted ingrese. **Nota:** Para la columna Datos sin extraer, puede seleccionar una columna de origen actual o escribir un nuevo nombre.	Los datos sin extraer para el registro correspondiente según el término especificado.

Columna	Descripción	Valor de salida
Datos extraídos	Esta columna de salida se agrega si un nuevo nombre de columna, no presente en la entrada, se especifica como la columna Datos extraídos. El nombre de la columna será el que usted ingrese. **Nota:** Para la columna Datos extraídos, puede seleccionar una columna de origen actual o escribir un nuevo nombre.	Los datos extraídos para el registro correspondiente según el término especificado.
Término de Advanced Transformer identificado	Indica si el término se identificó o no.	Los valores posibles son `Sí` y `No`.

Uso de un trabajo MapReduce de Advanced Transformer

1. Cree una instancia de `DataNormalizationFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Advanced Transformer mediante la creación de una instancia de `AdvancedTransformerDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Configure las reglas de Advanced Transformer mediante la creación de una instancia de`AdvancedTransformerConfiguration`. Dentro de esta instancia:

      Agregue una instancia de tipo`AbstractAdvancedTransformerRules`. Esta instancia`AbstractAdvancedTransformerRules` debe definirse con el uso de una de estas clases: `TableDataExtraction` o `RegularExpressionExtraction`, que corresponde a la categoría de regla deseada de Advanced Transformer.
   b) Establezca los detalles de la ruta de los datos de referencia y el tipo de ubicación creando una instancia de `ReferenceDataPath`. Consulte **Enum ReferenceDataPathLocation** en la página 195.
   c) Cree una instancia de `AdvancedTransformerDetail`, mediante el paso de una instancia de tipo `JobConfig` y la `AdvancedTransformerConfiguration` y las instancias`ReferenceDataPath` creadas con anterioridad como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.
   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `AdvancedTransformerDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `AdvancedTransformerDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `AdvancedTransformerDetail`.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `DataNormalizationFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `AdvancedTransformerDetail` como un argumento.
El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `DataNormalizationFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Advanced Transformer

1. Cree una instancia de `DataNormalizationFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Advanced Transformer mediante la creación de una instancia de `AdvancedTransformerDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Configure las reglas de Advanced Transformer mediante la creación de una instancia de`AdvancedTransformerConfiguration`. Dentro de esta instancia:

   Agregue una instancia de tipo`AbstractAdvancedTransformerRules`. Esta instancia`AbstractAdvancedTransformerRules` debe definirse con el uso de una de estas clases: `TableDataExtraction` o `RegularExpressionExtraction`, que corresponde a la categoría de regla deseada de Advanced Transformer.

   b) Establezca los detalles de la ruta de los datos de referencia y el tipo de ubicación creando una instancia de `ReferenceDataPath`. Consulte **Enum ReferenceDataPathLocation** en la página 195.
   c) Cree una instancia de `AdvancedTransformerDetail`, mediante el paso de una instancia de tipo `JobConfig` y la`AdvancedTransformerConfiguration` y las instancias`ReferenceDataPath` creadas con anterioridad como los argumentos para su constructor.

   El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `AdvancedTransformerDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un

archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `AdvancedTransformerDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `AdvancedTransformerDetail`.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`DataNormalizationFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `AdvancedTransformerDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Trabajos del módulo Universal Addressing

Módulo común de la API

UniversalAddressingDetail<T extends ProcessType>

Propósito
Para especificar los detalles de un trabajo del módulo Universal Addressing.

UniversalAddressingFactory

Propósito
Una clase de fábrica única para crear instancias de trabajo del módulo Universal Addressing.

Validate Address

Entidades API

UAMAddressingDetail<T extends ProcessType>

Propósito
Especificar los detalles de un trabajo de Validate Address Global.

UniversalAddressEngineConfiguration

Propósito
To set various configurations like the *reference data path* and *COBOL runtime path* required to create and run the Validate Address job.

These are one-time settings.

UAMAddressingFactory

Propósito
Una clase de fábrica única para crear instancias de trabajos de Validate Address Global.

Esta instancia se usa para generar contadores de informes e informes CASS.

UniversalAddressGeneralConfiguration

Propósito
Establecer configuraciones de bases de datos necesarias para crear y ejecutar el trabajo Validate Address Global.

UniversalAddressValidateInputConfiguration

Propósito
Configurar parámetros para la entrada que se creará y ejecutar el trabajo Validate Address Global. This is a rule setting, and has various options. These settings vary for every job.

Parámetros de entrada

Parámetro	Descripción
Universal Address Engine Configuration	To set various job run configurations: **1.** DPV Database Path **2.** Suite Link DB Path **3.** EWS Database Path **4.** RDI Database Path **5.** Lacs Database Path **6.** Ruta de acceso de datos de referencia **7.** COBOL Runtime Path **8.** Modules directory

Parámetro	Descripción
Universal Address Validate Input Configuration	

Parámetro	Descripción
	To configure the input settings: **1.** Output Standard Address **2.** Output Address Elements **3.** Output Postal Data **4.** Output Parsed Input **5.** Bloques de dirección de salida **6.** Salida normalizada por falta de resultados **7.** Mayúsculas y minúsculas de salida **8.** Separador de código postal de salida **9.** Generar caracteres multinacionales **10.** Ejecutar DPV **11.** Ejecutar RDI **12.** Ejecutar ESM **13.** Ejecutar ASM **14.** Ejecutar EWS **15.** Perform LACS Link **16.** Ejecutar LOT **17.** No fue posible un cruce CMRA **18.** Extract Firm **19.** Extract Urb **20.** Output Report 3553 **21.** Output Report SERP **22.** Output Report Summary **23.** Output CASS Detail **24.** Sí, se obtienen códigos de resultado de nivel de campo. **25.** Keep Multimatch **26.** Resultados máximos **27.** Standard Address Format **28.** Standard Address PMB Line **29.** City Name Format **30.** Vanity City Format Long **31.** Output Country Format **32.** País de origen **33.** Rigurosidad del cruce de datos de calles **34.** Rigurosidad del cruce de nombres de la empresa **35.** Rigurosidad del cruce de datos direccionales **36.** Lógica de dirección doble **37.** DPV Successful Status Condition **38.** Report List File Name **39.** Report List Processor Name **40.** Report List Number **41.** Report Mailer Address **42.** Report Mailer Name **43.** Report Mailer City Line **44.** Address Line Search On Fail **45.** Output Street Alias

Parámetro	Descripción
	46. Output VeriMove Block **47.** DPV Determine No Stat **48.** DPV Determine Vacancy **49.** Output Abbreviated Alias **50.** Output Preferred Alias **51.** Output Preferred City **52.** Perform Suite Link **53.** Suppress Zplus Phantom Carrier R777
Universal Address General Configuration	To set JVM configurations: **1.** DPV File Type **2.** DPV Memory Model **3.** Lacs Link Memory Model **4.** Suite Link Memory Model
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.
Comprimir el resultado	Bandera para indicar si el resultado se debe comprimir. Configure en `verdadero` para comprimir el resultado.

Parámetro	Descripción
Informes CASS	Las configuraciones para generar un informe CASS. Invoque cualquiera de los métodos sobrecargados `generateCASSReport()` mediante el uso de la instancia `UAMAddressingFactory`. Los informes CASS se generan en formato PDF. Los parámetros son los siguientes: **Contadores** — Un mapa de los contadores que se van a incluir en el informe CASS. **Nombre de trabajo** — El nombre del trabajo. Este se incluye en el nombre de archivo del informe CASS. **Ruta** — El directorio donde se aloja el informe CASS creado. Este es un valor de entrada opcional para los informes CASS. La ruta (`path`) debe estar en la ubicación del clúster o cliente según si el trabajo SDK se ejecuta en un entorno de clúster o en el equipo del cliente, respectivamente. **Nota:** Si no se especifica la ruta (`path`), el nuevo informe CASS se alojará en el directorio de trabajo actual. **Tipo de informe** — El tipo de informe CASS que se va a generar. Puede especificar uno o más valores desde **Enum UAMCASSReportType** en la página 205.

Columnas de salida

1. AdditionalInputData
2. AddressLine1
3. AddressLine2
4. AddressLine3
5. AddressLine4
6. AddressLine5
7. City
8. Country
9. FirmName
10. PostalCode
11. PostalCode.AddOn
12. PostalCode.Base
13. StateProvince
14. "USUrbanName": "",
15. AdditionalInputData
16. ApartmentLabel
17. ApartmentLabel2

18. ApartmentNumber
19. ApartmentNumber2
20. HouseNumber
21. LeadingDirectional (Elemento direccional anterior)
22. POBox
23. PrivateMailbox
24. Tipo de buzón de correo privado
25. RRHC
26. StateProvince
27. StreetName
28. StreetSuffix
29. TrailingDirectional (Elemento direccional posterior)
30. "USUrbanName": "",
31. ApartmentLabel.Input
32. ApartmentNumber.Input
33. City.Input
34. Country.Input
35. FirmName.Input
36. HouseNumber.Input
37. LeadingDirectional.Input
38. POBox.Input
39. Entrada de PostalCode
40. PrivateMailbox.Input
41. PrivateMailbox.Type.Input
42. RRHC.Input
43. StateProvince.Input
44. StreetName.Input
45. StreetSuffix.Input
46. TrailingDirectional.Input
47. USUrbanName.Input
48. PostalBarCode
49. USAltAddr
50. USBCCheckDigit
51. USCarrierRouteCode
52. USCongressionalDistrict
53. USCountyName
54. USFinanceNumber
55. USFIPSCountyNumber
56. USLACS
57. USLastLineNumber
58. AddressFormat

59. Confianza
60. CouldNotValidate
61. CountryLevel
62. MatchScore
63. MultimatchCount
64. MultipleMatches
65. ProcessedBy
66. RecordType
67. RecordType (predeterminado)
68. Estado
69. Status.Code
70. Status.Description
71. AddressRecord.Result
72. ApartmentLabel.Result
73. ApartmentNumber.Result
74. City.Result
75. Country.Result
76. FirmName.Result
77. HouseNumber.Result
78. LeadingDirectional.Result
79. POBox.Result
80. PostalCode.Result
81. PostalCodeCity.Result
82. Código fuente PostalCode:
83. PostalCode.Type
84. RRHC.Result
85. RRHC.Type
86. StateProvince.Result
87. Street.Result
88. StreetName.AbbreviatedAlias.Result
89. StreetName.Alias.Type
90. StreetName.PreferredAlias.Result
91. StreetName.Result
92. StreetSuffix.Result
93. TrailingDirectional.Result
94. USUrbanName.Result
95. USLOTCode
96. USLOTHex
97. USLOTSequence
98. USLACS.ReturnCode
99. RDI

100 DPV
101 CMRA
102 DPVFootnote
103 DPVVacant
104 DPVNoStat
105 SuiteLinkReturnCode
106 SuiteLinkMatchCode
107 SuiteLinkFidelity
108 VeriMoveDataBlock

Nota: Para las descripciones de campo, consulte el tema *Validate Address* en la *Guía de direcciones* de Spectrum™ Technology Platform.

Uso de un trabajo MapReduce de Validate Address

Atención: Before creating and running the first Validate Address job, ensure the Acushare service is running. Para obtener información sobre los pasos, consulte **Running Acushare Service** en la página 11.

1. Cree una instancia de `UAMAddressingFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Validate Address Global mediante la creación de una instancia de `UAMAddressingDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40. Para lograrlo, los pasos son:
   a) To configure the input settings for the job, create an instance of`UniversalAddressValidateInputConfiguration`.

      Set the values of the various required fields of this instance, using the enums **Enum PreferredCity** en la página 203, **Enum CasingType** en la página 202, **Enum CityNameFormat** en la página 202, **EnumOutputCountryFormat** en la página 202, **Enum StandardAddressFormat** en la página 202, **Enum StandardAddressPMBLine** en la página 203, **Enum StreetMatchingStrictness** en la página 203, **Enum FirmMatchingStrictness** en la página 203, **Enum DirectionalMatchingStrictness** en la página 203, **Enum DualAddressLogic** en la página 202, and **Enum DPVSuccessStatusCondition** en la página 204 where applicable.

      Importante: Para ejecutar Validate Address en el modo CASS Certified™, establezca los campos `outputReport3553`, `outputCASSDetail` y`outputReportSummary` de esta instancia como `true`. Los informes CASS poseen contenido que solo es válido cuando el trabajo se ejecuta en el modo CASS Certified™. Además, se generan PDF de informes en blanco.

   b) Establezca los detalles de la *ruta de los datos de referencia* creando una instancia de `LocalReferenceDataPath`.
   c) To configure the various job run settings, create an instance of`UAMUSAddressingEngineConfiguration` by passing the`LocalReferenceDataPath`

instance created above, and the *COBOL Runtime path* and *modules directory path* as `String` values, as arguments to its constructor.

Once the `UAMUSAddressingEngineConfiguration` instance is created, set the values for its various required fields.

d) To configure JVM settings, create an instance of`UniversalAddressGeneralConfiguration`.

Use the enums **Enum DPVFileType** en la página 203, **Enum DPVMemoryModel** en la página 204, **Enum LacsLinkMemoryModel** en la página 204, and **Enum SuiteLinkMemoryModel** en la página 204.

e) Cree una instancia de `UAMAddressingDetail`, mediante el paso de una instancia de tipo `JobConfig` y la `UAMUSAddressingEngineConfiguration`, `UniversalAddressGeneralConfiguration` y las instancias`UniversalAddressGeneralConfiguration` creadas con anterioridad como los argumentos para su constructor.

El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.

1. Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `UAMAddressingDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

2. Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `UAMAddressingDetail`.

   Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

3. Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `UAMAddressingDetail`.
4. Establezca la bandera `compressOutput` de la instancia `UAMAddressingDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `UAMAddressingFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `UAMAddressingDetail` como un argumento.
   El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo exitosa, use la instancia creada anteriormente de `UAMAddressingFactory` para invocar su método `getCounters()`, pasando el trabajo creado como un argumento.

Se recibe un `Map` de contadores.

6. Para generar informes CASS después de una ejecución de trabajo exitosa, use la instancia previamente creada de `UAMAddressingFactory` para invocar el método `generateCASSReport()`. Puede invocar cualquiera de las versiones sobrecargadas del método `generateCASSReport()`.

   Según qué firma de método `generateCASSReport()` se emplee, pase como argumentos el mapa (`Map`) de contadores de informes derivados del paso anterior, el nombre de trabajo (`jobName`), la ruta (`path`) donde se debe almacenar el informe CASS generado y el tipo de informe (`reportType`) requerido que se va a crear.

   La ruta (`path`) debe estar en la ubicación del clúster o cliente según si el trabajo SDK se ejecuta en un entorno de clúster o en el equipo del cliente, respectivamente.

   > **Nota:** Si no se especifica la ruta (`path`), el nuevo informe CASS se alojará en el directorio de trabajo actual.

   El parámetro `reportType` debe tener valores provenientes de **Enum UAMCASSReportType** en la página 205. Puede especificar uno o más tipos de informe en este parámetro.

Uso de un trabajo Spark de Validate Address

Atención: Before creating and running the first Validate Address job, ensure the Acushare service is running. Para obtener información sobre los pasos, consulte **Running Acushare Service** en la página 11.

1. Cree una instancia de `UAMAddressingFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Validate Address Global mediante la creación de una instancia de `UAMAddressingDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41. Para lograrlo, los pasos son:
   a) To configure the input settings for the job, create an instance of`UniversalAddressValidateInputConfiguration`.

      Set the values of the various required fields of this instance, using the enums **Enum PreferredCity** en la página 203, **Enum CasingType** en la página 202, **Enum CityNameFormat** en la página 202, **EnumOutputCountryFormat** en la página 202, **Enum StandardAddressFormat** en la página 202, **Enum StandardAddressPMBLine** en la página 203, **Enum StreetMatchingStrictness** en la página 203, **Enum FirmMatchingStrictness** en la página 203, **Enum DirectionalMatchingStrictness** en la página 203, **Enum DualAddressLogic** en la página 202, and **Enum DPVSuccessStatusCondition** en la página 204 where applicable.

      Importante: Para ejecutar Validate Address en el modo CASS Certified™, establezca los campos `outputReport3553`, `outputCASSDetail` y`outputReportSummary` de esta instancia como `true`. Los informes CASS poseen contenido que solo es válido cuando el trabajo se ejecuta en el modo CASS Certified™. Además, se generan PDF de informes en blanco.

b) Establezca los detalles de la *ruta de los datos de referencia* creando una instancia de `LocalReferenceDataPath`.

c) To configure the various job run settings, create an instance of`UAMUSAddressingEngineConfiguration` by passing the`LocalReferenceDataPath` instance created above, and the *COBOL Runtime path* and *modules directory path* as `String` values, as arguments to its constructor.

Once the `UAMUSAddressingEngineConfiguration` instance is created, set the values for its various required fields.

d) To configure JVM settings, create an instance of`UniversalAddressGeneralConfiguration`.

Use the enums **Enum DPVFileType** en la página 203, **Enum DPVMemoryModel** en la página 204, **Enum LacsLinkMemoryModel** en la página 204, and **Enum SuiteLinkMemoryModel** en la página 204.

e) Cree una instancia de `UAMAddressingDetail`, mediante el paso de una instancia de tipo `JobConfig` y la `UAMUSAddressingEngineConfiguration` y las instancias`UniversalAddressGeneralConfiguration` creadas con anterioridad como los argumentos para su constructor.

El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

1. Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `UAMAddressingDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

2. Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `UAMAddressingDetail`.

   Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

3. Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `UAMAddressingDetail`.
4. Establezca la bandera `compressOutput` de la instancia `UAMAddressingDetail` en `verdadero` para comprimir la salida del trabajo.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`UAMAddressingFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `UAMAddressingDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Para mostrar que los contadores de informes publican una ejecución de trabajo exitosa, use la instancia creada anteriormente de `UAMAddressingFactory` para invocar su método `getCounters()`, pasando el trabajo creado como un argumento.
   Se recibe un `Map` de contadores.
5. Para generar informes CASS después de una ejecución de trabajo exitosa, use la instancia previamente creada de `UAMAddressingFactory` para invocar el método `generateCASSReport()`. Puede invocar cualquiera de las versiones sobrecargadas del método `generateCASSReport()`.

   Según qué firma de método `generateCASSReport()` se emplee, pase como argumentos el mapa (`Map`) de contadores de informes derivados del paso anterior, el nombre de trabajo (`jobName`), la ruta (`path`) donde se debe almacenar el informe CASS generado y el tipo de informe (`reportType`) requerido que se va a crear.

   La ruta (`path`) debe estar en la ubicación del clúster o cliente según si el trabajo SDK se ejecuta en un entorno de clúster o en el equipo del cliente, respectivamente.

   > **Nota:** Si no se especifica la ruta (`path`), el nuevo informe CASS se alojará en el directorio de trabajo actual.

   El parámetro `reportType` debe tener valores provenientes de **Enum UAMCASSReportType** en la página 205. Puede especificar uno o más tipos de informe en este parámetro.

Validate Address Global

Entidades API

GlobalAddressingDetail<T extends ProcessType>

Propósito
Especificar los detalles de un trabajo de Validate Address Global.

GlobalAddressingEngineConfiguration

Propósito
Establecer configuraciones de bases de datos necesarias para crear y ejecutar el trabajo Validate Address Global.

GlobalAddressingFactory

Propósito
Una clase de fábrica única para crear instancias de trabajos de Validate Address Global.

GlobalAddressingGeneralConfiguration

Propósito
Establecer configuraciones de bases de datos necesarias para crear y ejecutar el trabajo Validate Address Global.

GlobalAddressingInputConfiguration

Propósito
Configurar parámetros para la entrada que se creará y ejecutar el trabajo Validate Address Global.

Parámetros de entrada

Parámetro	Descripción
Configuración del motor de Validate Address Global	Para configurar parámetros de base de datos: **1.** Tipo de base de datos **2.** Tipo de carga previa **3.** Ruta de acceso de datos de referencia **4.** Si todos los países son compatibles. Si no, lista de países compatibles
Configuración de entrada de Validate Address Global	Para configurar estos parámetros para la entrada: **1.** Tipo de estado/provincia en el resultado **2.** Alcance de comparación en proceso **3.** País forzado ISO3 en entrada **4.** País predeterminado ISO3 en entrada **5.** Delimitador de formato en entrada **6.** Delimitador de formato en resultado **7.** Incluir entradas en resultado **8.** Tipo de país en resultado **9.** Nivel de optimización de proceso **10.** Idioma preferido de resultado **11.** Modo de proceso **12.** Secuencia preferida en resultado **13.** Resultados máximos **14.** Uso de mayúscula y minúsculas para el resultado

Parámetro	Descripción
Configuración general de Validate Address Global	To set JVM configurations: **1.** Tamaño de caché **2.** Recuento máximo de subprocesos **3.** Recuento máximo de objeto de dirección **4.** Rangos para expandir **5.** Expansión de rango flexible **6.** Activar registro de transacciones **7.** Uso máximo de memoria en MB
Desbloquear código	Para desbloquear los datos en la base de datos.
Ruta de acceso de datos de referencia	Para especificar los detalles de la ruta de acceso de Datos de referencia. **Nota:** Para los trabajos de UAM, los datos de referencia deben estar colocados solo en nodos de datos locales en el clúster.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.

Columnas de salida

Datos de dirección

1. AddressBlock1-9
2. AddressLine1-6
3. AdministrativeDistrict
4. ApartmentLabel
5. ApartmentNumber
6. BlockName
7. BuildingName
8. City
9. City, AddInfo
10. City, SortingCode

11. Contact
12. Country
13. Condado
14. FirmName
15. Múltiplo inferior
16. HouseNumber
17. LastLine
18. LeadingDirectional (Elemento direccional anterior)
19. Locality
20. POBox
21. PostalCode
22. PostalCode.AddOn
23. PostalCode.Base
24. Room
25. SecondaryStreet
26. StateProvince
27. StreetName
28. StreetSuffix
29. Área secundaria de edificio
30. Suburb
31. Territory
32. TrailingDirectional (Elemento direccional posterior)

Datos de entrada originales

1. AddressLine1.Input
2. AddressLine2.Input
3. AddressLine3.Input
4. AddressLine4.Input
5. AddressLine5.Input
6. AddressLine6.Input
7. City.Input
8. StateProvince.Input
9. Entrada de PostalCode
10. Contact.Input
11. Country.Input
12. FirmName.Input
13. Street.Input
14. Number.Input
15. Building.Input
16. SubBuilding.Input
17. DeliveryService.Input

Atención: Los campos de entrada `AddressLine2.Input`,`AddressLine3.Input`, `AddressLine4.Input`,`AddressLine5.Input` y `AddressLine6.Input` están incluidos en la salida solo si el campo `resultIncludeInputs` de la clase `GlobalAddressingInputConfiguration` está configurado en `verdadero`. Además, solo aquellos campos `AddressLineX.input` están incluidos en la salida, los que forman parte de la entrada.

Códigos de resultado

1. AddressType
2. Confianza
3. CountOverflow
4. ElementInputStatus
5. ElementRelevance
6. ElementResultStatus
7. MailabilityScore
8. ModeUsed
9. MultimatchCount
10. ProcessStatus
11. Estado
12. Status.Code
13. Status.Description

> **Nota:** Para conocer las descripciones de campos, consulte el tema *Validate Address Global* de la *Guía de direcciones* de Spectrum™ Technology Platform.

Uso de un trabajo MapReduce de Validate Address Global

1. Cree una instancia de `GlobalAddressingFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Validate Address Global mediante la creación de una instancia de `GlobalAddressingDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40. Para lograrlo, los pasos son:
   a) Configure los parámetros de entrada mediante la creación de una instancia de `GlobalAddressingGeneralConfiguration`.

      Use los enums **Enum CacheSize** en la página 201, **Enum RangesToExpand** en la página 201 y **Enum FlexibleRangeExpansion** en la página 201.

   b) Establezca los detalles de la ruta de los datos de referencia creando una instancia de `LocalReferenceDataPath`.
   c) Configure los parámetros de la base de datos necesaria mediante la creación de una instancia de `GlobalAddressingEngineConfiguration` y mediante el paso de la instancia `LocalReferenceDataPath` anterior como argumento.

      1. Establezca el *tipo de carga previa* en esta instancia mediante el uso del enum **Enum PreloadingType** en la página 198.

2. Establezca el *tipo de base de datos* usando **Enum DatabaseType** en la página 197.
3. Set the supported countries using the **Enum CountryCodes** en la página 198.
4. Si todos los países son compatibles, establezca el atributo `isAllCountries` en verdadero. Además, especifique la lista de valores **Enum CountryCodes** en la página 198 separados por coma en el valor de cadena `supportedCountries`.

d) Configure los parámetros de entrada mediante la creación de una instancia de `GlobalAddressingInputConfiguration`.

Para establecer los valores de los diversos campos de esta instancia, use los enums **Enum CountryCodes** en la página 198, **EnumStateProvinceType** en la página 198, **Enum CountryType** en la página 198, **Enum PreferredScript** en la página 199, **Enum PreferredLanguage** en la página 199, **Enum Casing** en la página 199, **Enum OptimizationLevel** en la página 199, **Enum Mode** en la página 199 y **Enum MatchingScope** en la página 200 según corresponda.

e) Configure la clave de desbloqueo para los datos como valor de `String` en una `List`.

f) Cree una instancia de `GlobalAddressingDetail` mediante el paso de una instancia de tipo `JobConfig` y la `List` de valores de código de desbloqueo, la instancia `GlobalAddressingEngineConfiguration` y la instancia `GlobalAddressingInputConfiguration` creada anteriormente como los argumentos para su constructor.

El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.

1. Configure los parámetros de la base de datos mediante el establecimiento del campo `generalConfiguration` de la instancia `GlobalAddressingDetail` para la instancia `GlobalAddressingGeneralConfiguration` creada anteriormente.
2. Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `GlobalAddressingDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

3. Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `GlobalAddressingDetail`.

   Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

4. Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `GlobalAddressingDetail`.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `GlobalAddressingFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `GlobalAddressingDetail` como un argumento.
   El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `GlobalAddressingFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Validate Address Global

1. Cree una instancia de `GlobalAddressingFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Validate Address Global mediante la creación de una instancia de `GlobalAddressingDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41. Para lograrlo, los pasos son:
   a) Configure los parámetros de entrada mediante la creación de una instancia de `GlobalAddressingGeneralConfiguration`.
      Use los enums **Enum CacheSize** en la página 201, **Enum RangesToExpand** en la página 201 y **Enum FlexibleRangeExpansion** en la página 201.
   b) Establezca los detalles de la ruta de los datos de referencia creando una instancia de `LocalReferenceDataPath`.
   c) Configure los parámetros de la base de datos necesaria mediante la creación de una instancia de `GlobalAddressingEngineConfiguration` y mediante el paso de la instancia `LocalReferenceDataPath` anterior como argumento.
      1. Establezca el *tipo de carga previa* en esta instancia mediante el uso del enum **Enum PreloadingType** en la página 198.
      2. Establezca el *tipo de base de datos* usando **Enum DatabaseType** en la página 197.
      3. Set the supported countries using the **Enum CountryCodes** en la página 198.
      4. Si todos los países son compatibles, establezca el atributo `isAllCountries` en verdadero. Además, especifique la lista de valores **Enum CountryCodes** en la página 198 separados por coma en el valor de cadena `supportedCountries`.
   d) Configure los parámetros de entrada mediante la creación de una instancia de `GlobalAddressingInputConfiguration`.
      Para establecer los valores de los diversos campos de esta instancia, use los enums **Enum CountryCodes** en la página 198, **EnumStateProvinceType** en la página 198, **Enum CountryType** en la página 198, **Enum PreferredScript** en la página 199, **Enum PreferredLanguage** en la página 199, **Enum Casing** en la página 199, **Enum OptimizationLevel** en la página 199, **Enum Mode** en la página 199 y **Enum MatchingScope** en la página 200 según corresponda.
   e) Configure la clave de desbloqueo para los datos como valor de `String` en una `List`.

f) Cree una instancia de `GlobalAddressingDetail` mediante el paso de una instancia de tipo `JobConfig` y la `List` de valores de código de desbloqueo, la instancia `GlobalAddressingEngineConfiguration` y la instancia `GlobalAddressingInputConfiguration` creada anteriormente como los argumentos para su constructor.

El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

1. Configure los parámetros de la base de datos mediante el establecimiento del campo `generalConfiguration` de la instancia `GlobalAddressingDetail` para la instancia `GlobalAddressingGeneralConfiguration` creada anteriormente.
2. Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `GlobalAddressingDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

3. Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `GlobalAddressingDetail`.

   Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

4. Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `GlobalAddressingDetail`.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`GlobalAddressingFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `GlobalAddressingDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Validate Address Loqate

Entidades API

LoqateAddressingDetail<T extends ProcessType>

Propósito
Especificar los detalles de un trabajo de Validate Address Global.

LoqateAddressingEngineConfiguration

Propósito
Establecer configuraciones de bases de datos necesarias para crear y ejecutar el trabajo Validate Address Global.

LoqateAddressingFactory

Propósito
Una clase de fábrica única para crear instancias de trabajos de Validate Address Global.

LoqateAddressingGeneralConfiguration

Propósito
Establecer configuraciones de bases de datos necesarias para crear y ejecutar el trabajo Validate Address Global.

LoqateAddressingValidateConfiguration

Propósito
Configurar parámetros para la entrada que se creará y ejecutar el trabajo Validate Address Global.

Parámetros de entrada

Parámetro	Descripción
Configuración del motor de Validate Address Global	To set configurations for performing the validations: **1.** Verbose **2.** Tool Info **3.** Output Address Format **4.** Log Input **5.** {"output": [{ **6.** Log File Name **7.** Match Score Absolute Threshold **8.** Match Score Threshold Factor **9.** Postal Code Max Results **10.** Strict Reference Match

Parámetro	Descripción
Validate Address Loqate Validate Configuration	Para configurar estos parámetros para la entrada: **1.** Include Standard Address **2.** Incluir elementos de dirección cruzados **3.** Incluir elementos de dirección de entrada estandarizados **4.** Obtener bloques de datos de dirección **5.** Mayúsculas y minúsculas de salida **6.** Incluir códigos de resultado para campos individuales **7.** Obtener múltiples direcciones **8.** Failed On Multi Match Found **9.** Multiple Address Count **10.** Formato de país **11.** País predeterminado **12.** Secuencia de comandos/Alfabeto: **13.** Devolver campos de dirección geocodificados **14.** Nivel de aceptación **15.** Puntuación mínima de coincidencia **16.** Dar formato a datos mediante convenciones de AMAS **17.** Is Duplicate Handling **18.** Single Field Duplicate Handling **19.** Multi Field Duplicate Handling **20.** Non Standard Field Duplicate Handling **21.** Output Field Duplicate Handling
Configuración general de Validate Address Global	To set JVM configurations: **1.** Maximum Idle Objects **2.** Minimum Idle Objects **3.** Maximum Active Objects **4.** Maximum Wait Time **5.** Action When Exhausted **6.** Test on Borrow **7.** Test on Return **8.** Test While Idle **9.** Time Between Eviction Runs in Milliseconds **10.** Number of Tests Per Eviction Run **11.** Min Evictable Idle Time in Milliseconds
Ruta de acceso de datos de referencia	Para especificar los detalles de la ruta de acceso de Datos de referencia. **Nota:** Para los trabajos de UAM, los datos de referencia deben estar colocados solo en nodos de datos locales en el clúster.

Parámetro	Descripción
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.

Columnas de salida

1. AdditionalInputData
2. AddressLine1-4
3. City
4. Country
5. FirmName
6. PostalCode
7. PostalCode.AddOn
8. PostalCode.Base
9. StateProvince
10. AddressBlock1-9
11. ApartmentLabel
12. ApartmentNumber

13. ApartmentNumber2
14. Edificio
15. City
16. Country
17. County *
18. FirmName
19. HouseNumber
20. LeadingDirectional (Elemento direccional anterior)
21. POBox
22. PostalCode
23. Principality *
24. StateProvince
25. StreetAlias
26. StreetName
27. StreetSuffix
28. Subcity *
29. Substreet *
30. TrailingDirectional (Elemento direccional posterior)
31. ApartmentLabel.Input
32. ApartmentNumber.Input
33. City.Input
34. Country.Input
35. County.Input *
36. FirmName.Input
37. HouseNumber.Input
38. LeadingDirectional.Input
39. POBox.Input
40. Entrada de PostalCode
41. Principality.Input *
42. StateProvince.Input
43. StreetAlias.Input
44. StreetName.Input
45. StreetSuffix.Input
46. Subcity.Input *
47. Substreet.Input *
48. TrailingDirectional.Input
49. Geocode.MatchCode
50. Latitud
51. Longitud
52. SearchDistance
53. Confianza

54. CouldNotValidate
55. MatchScore
56. ProcessedBy
57. Estado
58. Status.Code
59. Status.Description
60. ApartmentLabel.Result
61. ApartmentNumber.Result
62. City.Result
63. Country.Result
64. County.Result *
65. FirmName.Result
66. HouseNumber.Result
67. LeadingDirectional.Result
68. POBox.Result
69. PostalCode.Result
70. PostalCode.Type
71. Principality.Result *
72. StateProvince.Result
73. StreetAlias.Result
74. StreetName.Result
75. StreetSuffix.Result
76. Subcity.Result *
77. Substreet.Result *
78. TrailingDirectional.Result
79. Barcode
80. DPID
81. FloorNumber
82. FloorType
83. PostalBoxNum

*Este es un subcampo y puede que no contenga datos.

Tabla 1: Códigos de cruce de centroide de código de ciudad/calle/postal

Elemento	Código de cruce
Punto de dirección	P4
Punto de dirección interpolada	I4
Centroide de calle	A4/P3

Elemento	Código de cruce
Centroide de código postal/ciudad	A3/P2/A2

Nota: Para conocer las descripciones de campos, consulte el tema *Validate Address Loqate* de la *Guía de direcciones* de Spectrum™ Technology Platform.

Uso de un trabajo MapReduce de Validate Address Loqate

1. Cree una instancia de `LoqateAddressingFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Validate Address Global mediante la creación de una instancia de `LoqateAddressingDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40. Para lograrlo, los pasos son:
   a) Configure los parámetros de entrada mediante la creación de una instancia de `LoqateAddressingGeneralConfiguration`.

      Use the enum **Enum ExhaustedAction** en la página 200.
   b) Configure the necessary database settings by creating an instance of`LoqateAddressingEngineConfiguration` and set the various fields.
   c) Configure los parámetros de entrada mediante la creación de una instancia de `LoqateAddressingValidateConfiguration`.

      Para establecer los valores de los diversos campos de esta instancia, use los enums **Enum AcceptanceLevel** en la página 200, **Enum CountryCodes** en la página 198, **Enum OutputCasing** en la página 201, **Enum CountryFormat** en la página 201, **Enum ScriptAlphabet** en la página 201, , , y según corresponda.
   d) Establezca los detalles de la ruta de los datos de referencia creando una instancia de `LocalReferenceDataPath`.
   e) Cree una instancia de `LoqateAddressingDetail` pasando una instancia del tipo `JobConfig`, la instancia `LocalReferenceDataPath` creada y la instancia `LoqateAddressingValidateConfiguration` creada anteriormente como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.

      1. Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `LoqateAddressingDetail`.

         Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

      2. Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `LoqateAddressingDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

3. Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `LoqateAddressingDetail`.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `LoqateAddressingFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `LoqateAddressingDetail` como un argumento.
   El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `LoqateAddressingFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Validate Address Loqate

1. Cree una instancia de `LoqateAddressingFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Validate Address Global mediante la creación de una instancia de `LoqateAddressingDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41. Para lograrlo, los pasos son:
   a) Configure los parámetros de entrada mediante la creación de una instancia de `LoqateAddressingGeneralConfiguration`.
      Use the enum **Enum ExhaustedAction** en la página 200.
   b) Configure the necessary database settings by creating an instance of`LoqateAddressingEngineConfiguration` and set the various fields.
   c) Configure los parámetros de entrada mediante la creación de una instancia de `LoqateAddressingValidateConfiguration`.
      Para establecer los valores de los diversos campos de esta instancia, use los enums **Enum AcceptanceLevel** en la página 200, **Enum CountryCodes** en la página 198, **Enum OutputCasing** en la página 201, **Enum CountryFormat** en la página 201, **Enum ScriptAlphabet** en la página 201, , , y según corresponda.
   d) Establezca los detalles de la ruta de los datos de referencia creando una instancia de `LocalReferenceDataPath`.
   e) Cree una instancia de `LoqateAddressingDetail` pasando una instancia del tipo `JobConfig`, la instancia `LocalReferenceDataPath` creada y la instancia `LoqateAddressingValidateConfiguration` creada anteriormente como los argumentos para su constructor.
      El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.

1. Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `LoqateAddressingDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

2. Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `LoqateAddressingDetail`.

   Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

3. Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `LoqateAddressingDetail`.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`LoqateAddressingFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `LoqateAddressingDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

Trabajos del módulo Universal Name

Módulo común de la API

UniversalNameDetail<T extends ProcessType>

Propósito
Para especificar los detalles de un trabajo del módulo Universal Name.

UniversalNameFactory

Propósito
Una clase de fábrica única para crear instancias de trabajo del módulo Universal Name.

Open Name Parser

Entidades API

OpenNameParserDetail

Propósito
Para especificar los detalles de un trabajo de Open Name Parser.

OpenNameParserConfiguration

Propósito
Dividir los nombres personales, nombres de empresas y otros términos del campo de datos `name` en las partes que los conforman.

Parámetros de entrada

Parámetro	Descripción
Configuración de Open Name Parser	Para dividir los nombres personales, nombres de empresas y otros términos del campo de datos `name` en las partes que los conforman.
Ruta de acceso de datos de referencia	Para especificar los detalles de la ruta de acceso de Datos de referencia.
Configuraciones de trabajo	Las configuraciones de Hadoop para el trabajo. Para un trabajo MapReduce, la instancia debe ser del tipo **MRJobConfig** en la página 39. Para un trabajo Spark, la instancia debe ser del tipo **SparkJobConfig** en la página 39.

Parámetro	Descripción
Archivo de entrada	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de registro** El separador de registro que se usa en el archivo de entrada. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Calificador de texto** El carácter que se utiliza para demarcar los valores de texto en un archivo delimitado. **Campos de la fila del encabezado** Una serie de campos del encabezado del archivo de entrada. **Omitir la primera fila** Bandera para indicar si se debe omitir la primera fila mientras se leen los registros del archivo de entrada. Debe configurarse como `verdadero` en caso de que la primera fila sea una fila del encabezado. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** La ruta del archivo de entrada en la plataforma Hadoop. *Common parameters:* **Asignaciones de campos** A map of key value pairs, with the existing column names as the keys and the desired output column names as the values.

Parámetro	Descripción
Archivo de salida	*For text files:* **Ruta de acceso al archivo** La ruta del archivo de entrada en la plataforma Hadoop. **Separador de campo** El separador que se usa entre dos campos consecutivos de un registro en el archivo de entrada. **Atención:** Invoke the appropriate constructor of`FilePath`. *For ORC format files:* **ORC File Path** The path of the output ORC format file on the Hadoop platform. *Common parameters:* **Sobrescribir** Bandera para indicar si el archivo de salida debe sobrescribir el archivo existente del mismo nombre. **Crear encabezado de salida** Bandera para indicar si el archivo del encabezado se debe crear en el servidor Hadoop o no.
Nombre de trabajo	El nombre del trabajo.

Columnas de salida

Además de las columnas de entrada, se agregan las siguientes columnas mientras se genera la salida de un trabajo de Open Name Parser:

	Formato	Descripción
AccountDescription	Cadena	Descripción de cuenta que es parte del nombre. Por ejemplo, en "Mary Jones Account # 12345", la descripción de cuenta es "Account#12345".
Campos relacionados con nombres de empresas		

	Formato	Descripción
FirmConjunction	Cadena	Indica que el nombre de una empresa contiene una conjunción como "d/b/a" (nombre con que hace negocios), "o/a" (nombre con que opera), y "t/a" (nombre con que comercializa).
FirmName	Cadena	El nombre de una empresa. Por ejemplo, "Pitney Bowes".
FirmSuffix	Cadena	El sufijo corporativo. Por ejemplo, "SRL" y "SA".
IsFirm	Cadena	Indica que el nombre es una empresa en vez de a una persona. Los valores son "true" (verdadero) o "false" (falso).
Campos relacionados con nombres de personas		
Conjunction	Cadena	Indica que el nombre contiene una conjunción "y", "o" o "&".
CultureCode	Cadena	Los códigos de cultura contenidos en los datos de entrada.
CultureCodeUsedToParse	Cadena	Identifica la gramática de cultura específica que se utilizó para analizar los datos. **Null (empty)** Cultura global (predeterminado). **de** Alemán **es** Español **ja** Japonés
FirstName	Cadena	El primer nombre de la persona.
GeneralSuffix	Cadena	El sufijo general o profesional de una persona. Por ejemplo, MD o PhD.
IsParsed	Cadena	Indica si se analizó un registro de salida. Los valores son "true" (verdadero) o "false" (falso).

	Formato	Descripción
IsPersonal	Cadena	Indica si el nombre es una persona en lugar de una empresa. Los valores son "true" (verdadero) o "false" (falso).
IsReverseOrder	Cadena	Indica si el nombre de la entrada está en el orden inverso. Los valores son "true" (verdadero) o "false" (falso).
LastName	Cadena	El apellido de la persona. Incluye el apellido paterno.
LeadingData	Cadena	Información que no es el nombre y que aparece antes que un nombre.
MaturitySuffix	Cadena	Un sufijo generacional de una persona. Por ejemplo, Jr. o Sr.
MiddleName	Cadena	El segundo nombre de una persona.
Name.	Cadena	El nombre personal o de empresa que se proporcionó en la entrada.
NameScore	Cadena	Indica la puntuación media de muestras conocidas y desconocidas para cada nombre. El valor de NameScore estará entre 0 y 100, como se define en la gramática de análisis. Se arroja 0 cuando no se encuentran cruces.
SecondaryLastName	Cadena	Gramática de análisis en español, el apellido de la madre de una persona.
TitleOfRespect	Cadena	Información que aparece antes que un nombre, como, por ejemplo, "Sr.", "Sra." o "Dr."
TrailingData	Cadena	Información que no es el nombre y que aparece después de un nombre.
Campos relacionados con nombres conjuntos		

	Formato	Descripción
Conjunction2	Cadena	Indica que un segundo nombre conjunto contiene una conjunción "y", "o" o "&".
Conjunction3	Cadena	Indica que un tercer nombre conjunto contiene una conjunción "y", "o" o "&".
FirmName2	Cadena	El nombre de una segunda empresa conjunta. Por ejemplo, Baltimore Gas & Electric dba Constellation Energy.
FirmSuffix2	Cadena	El sufijo de una segunda empresa conjunta.
FirstName2	Cadena	El primer nombre de un segundo nombre conjunto.
FirstName3	Cadena	El primer nombre de un tercer nombre conjunto.
GeneralSuffix2	Cadena	El sufijo general/profesional para un segundo nombre conjunto. Por ejemplo, MD o PhD.
GeneralSuffix3	Cadena	El sufijo general/profesional para un tercer nombre conjunto. Por ejemplo, MD o PhD.
IsConjoined	Cadena	Indica que el nombre de la entrada es conjunto. An example of a conjoined name is "John and Jane Smith." Values are true or false.
LastName2	Cadena	El apellido de un segundo nombre conjunto.
LastName3	Cadena	El apellido de un tercer nombre conjunto.
MaturitySuffix2	Cadena	El sufijo de madurez/generacional de un segundo nombre conjunto. Por ejemplo, Jr. o Sr.

	Formato	Descripción
MaturitySuffix3	Cadena	El sufijo de madurez/generacional de un tercer nombre conjunto. Por ejemplo, Jr. o Sr.
MiddleName2	Cadena	El segundo nombre de un segundo nombre conjunto.
MiddleName3	Cadena	El segundo nombre de un tercer nombre conjunto.
TitleOfRespect2	Cadena	Información que aparece antes de que un segundo nombre conjunto, como "Mr.", "Sra." o "Dr. "
TitleOfRespect3	Cadena	Información que aparece antes de que un tercer nombre conjunto, como "Mr.", "Sra." o "Dr. "

Uso de un trabajo MapReduce de Open Name Parser

1. Cree una instancia de `UniversalNameFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Open Name Parser mediante la creación de una instancia de `OpenNameParserDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **MRProcessType** en la página 40.
   a) Configure las reglas de Open Name Parser mediante la creación de una instancia de`OpenNameParserConfiguration`.
   b) Establezca los detalles de la ruta de los datos de referencia y el tipo de ubicación creando una instancia de `ReferenceDataPath`. Consulte **Enum ReferenceDataPathLocation** en la página 195.
   c) Cree una instancia de `OpenNameParserDetail`, mediante el paso de una instancia de tipo `JobConfig` y la `OpenNameParserConfiguration` y las instancias`ReferenceDataPath` creadas con anterioridad como los argumentos para su constructor.

      El parámetro `JobConfig` debe ser una instancia de tipo **MRJobConfig** en la página 39.
   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `OpenNameParserDetail`.

      Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.

e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `OpenNameParserDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `OpenNameParserDetail`.

3. Para crear un trabajo MapReduce, utilice la instancia anteriormente creada de `UniversalNameFactory` para invocar el método`createJob()`. Aquí, pase la instancia anterior de `OpenNameParserDetail` como un argumento.
El método `createJob()` devuelve una `List` de instancias de `ControlledJob`.
4. Ejecute el trabajo creado con el uso de una instancia de `JobControl`.
5. Para mostrar que los contadores de informes publican una ejecución de trabajo MapReduce exitosa, use la instancia creada anteriormente `UniversalNameFactory` para invocar su método `getCounters()`, mediante el paso de un trabajo creado como un argumento.

Uso de un trabajo Spark de Open Name Parser

1. Cree una instancia de `UniversalNameFactory` con su método estático `getInstance()`.
2. Proporcione los detalles de entrada y salida del trabajo Open Name Parser mediante la creación de una instancia de `OpenNameParserDetail` que especifique el `ProcessType`. La instancia debe usar el tipo **SparkProcessType** en la página 41.
   a) Configure las reglas de Open Name Parser mediante la creación de una instancia de`OpenNameParserConfiguration`.
   b) Establezca los detalles de la ruta de los datos de referencia y el tipo de ubicación creando una instancia de `ReferenceDataPath`. Consulte **Enum ReferenceDataPathLocation** en la página 195.
   c) Cree una instancia de `OpenNameParserDetail`, mediante el paso de una instancia de tipo `JobConfig` y la `OpenNameParserConfiguration` y las instancias`ReferenceDataPath` creadas con anterioridad como los argumentos para su constructor.

   El parámetro `JobConfig` debe ser una instancia de tipo **SparkJobConfig** en la página 39.
   d) Establezca los detalles del archivo de entrada mediante el campo `inputPath` de la instancia `OpenNameParserDetail`.

   Para un archivo de entrada de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de entrada mediante la invocación del constructor apropiado. Para un archivo de entrada ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de entrada ORC como argumento.
   e) Establezca los detalles del archivo de salida mediante el campo `outputPath` de la instancia `OpenNameParserDetail`.

Para un archivo de salida de texto, cree una instancia de `FilePath` con los detalles relevantes del archivo de salida mediante la invocación del constructor apropiado. Para un archivo de salida ORC, cree una instancia de `OrcFilePath` con la ruta del archivo de salida ORC como argumento.

f) Establezca el nombre del trabajo mediante el campo `jobName` de la instancia `OpenNameParserDetail`.

3. Para crear y ejecutar el trabajo Spark, utilice la instancia anteriormente creada de`UniversalNameFactory` para invocar este método`runSparkJob()`. Aquí, pase la instancia anterior de `OpenNameParserDetail` como un argumento.

   El método `runSparkJob()` ejecuta el trabajo y devuelve un`Map` de los contadores de informes del trabajo.

4. Muestre los contadores para ver las estadísticas de generación de informes para el trabajo.

5 - Funciones de Hive definidas por el usuario

In this section

Introducción

Apache Hive proporciona funciones definidas por el usuario (UDF). Una UDF puede definirse para realizar las acciones requeridas y lograr los objetivos deseados.

Big Data Quality SDK proporciona un conjunto de las funciones Hive definidas por el usuario y las funciones de agregación definidas por el usuario para ejecutar los trabajos de Data Quality enumerados.

Funciones definidas por el usuario (UDF)

Una función definida por el usuario procesa un registro a la vez.

Los trabajos basados en UDF son los que se indican a continuación:

- Match Key Generator
- Table Lookup
- Advanced Transformer
- Open Name Parser

Funciones de agregación definidas por el usuario (UDAF)

Una función de agregación definida por el usuario primero agrega registros a las colecciones en función del campo de combinación y, a continuación, procesa la colección de registros a la vez.

Los trabajos basados en UDAF son los que se indican a continuación:

- Interflow Match
- Intraflow Match
- Transactional Match
- Best of Breed
- Duplicate Synchronization
- Filtro
- Validate Address
- Validate Address Global
- Validate Address Loqate

Componentes de una UDF de Hive de **Big Data Quality SDK**

Los componentes clave necesarios para ejecutar una UDF de Hive en Big Data Quality SDK son:

Archivo JAR	El archivo JAR de Hive de Big Data Quality SDK del módulo al que pertenece la UDF de Hive de la calidad de los datos. Se debe registrar antes de usar cualquier UDF.

UDF/UDAF de trabajo	Cada trabajo de calidad de datos se brinda mediante una Función definida por el usuario (UDF) o una Función de agregación definida por el usuario (UDAF).
Alias	El alias asignado a una UDF de Hive. Esto es opcional.
Configuraciones	Las reglas especificadas en formato JSON, y otros detalles de configuración, según las cuales el trabajo se debe ejecutar.
Encabezado	Los campos de encabezado de los datos de entrada, en formato separado por comas.
Tabla de entrada	La tabla en la que se muestran los registros de entrada respectivamente para que se ejecute la UDF de Hive.
Tabla de candidato	La tabla en la que se muestran los registros del candidato para que se ejecute la UDAF de Hive en el caso de la UDF de Interflow Match.
Tabla de sospechoso	La tabla en la que se muestran los registros del sospechoso para que se ejecute la UDAF de Hive en el caso de la UDF de Interflow Match.
Hive.Map.Aggr	To turn the aggregation of data between Mapper and Reducer on or off, set this Hive environment variable to `false`. By default, `Hive.Map.Aggr = true` and the data is aggregated. Set this value to `false` for all Hive jobs in the SDK. **Nota:** This configuration is required for all UDAFs.
General Configurations	The memory configurations required to run the job. **Nota:** This configuration is required only for Universal Addressing Module Hive UDAFs.
Input Configurations	The settings for the input data. **Nota:** This configuration is required only for Universal Addressing Module Hive UDAFs.
Engine Configurations	Para definir varias configuraciones, como ajustes de bases de datos, *ruta de tiempo de ejecución COBOL*, *tipo de carga previa*, etc. **Nota:** This configuration is required only for Universal Addressing Module Hive UDAFs.
LD_LIBRARY_PATH	To set this environment variable to the paths of the various COBOL libraries required while running the Hive jobs. **Nota:** This configuration is required only for the Validate Address Hive UDAF.

Process Type	To specify the desired validation level to be used in a particular Hive job of the SDK. Currently, only address validation is supported. Set this value to `VALIDATE`. **Nota:** This configuration is required only for the Validate Address and Validate Address Loqate Hive UDAFs.
Salida	La salida de la UDF de Hive, que se puede mostrar en la consola o volcar en un archivo de salida.
Consulta	La consulta para ejecutar la UDF de Hive requerida. Para cada trabajo, puede hacer cada una de las siguientes acciones con la sintaxis de consulta correspondiente: • Mostrar la salida del trabajo en la consola • Volcar la salida del trabajo en un archivo de salida designado

Cómo usar una UDF de Hive

Para ejecutar cada trabajo basado en UDF, puede ejecutar los siguientes pasos individualmente en su cliente de Hive en una única sesión, o bien puede crear un archivo HQL que agrupe todos los pasos requeridos de forma secuencial y ejecutarlo de una sola vez.

1. En su cliente de Hive, inicie sesión en la base de datos de Hive obligatoria.
2. Registre el archivo JAR del módulo Big Data Quality SDK específico al que pertenece la UDF de Hive de Calidad de los datos.
3. In case of the Validate Address UDAF, to set the path of the COBOL libraries, set the environment variable `LD_LIBRARY_PATH` as below:

```
set mapreduce.admin.user.env =
LD_LIBRARY_PATH=/home/hduser/~/runtime/lib:
/home/hduser/~/runtime/bin:/home/hduser/~/server/modules/universaladdress/lib,
 ACU_RUNCBL_JNI_ONLOAD_DISABLE=1, G1RTS=/home/hduser/~/ ;
```

4. In case of the Validate Address Global UDAF, add the file *libAddressDoctor5.so* file as well.
5. En el caso de UDAF de Validate Address Loqate, agregue estos campos obligatorios a la memoria caché distribuida.
   - `loqate-core.car`
   - `LoqateVerificationLevel.csv`
   - `Loqate.csv`
   - `countryTables.csv`
   - `countryNameTables.csv`

6. Cree un alias de la UDF de Hive del trabajo de calidad de datos que desea ejecutar.
   Por ejemplo:

```
CREATE TEMPORARY FUNCTION matchkeygenerator as
'com.pb.bdq.amm.process.hive.matchkeygenerator.MatchKeyGeneratorUDF';
```

7. Especifique las configuraciones, como la regla de cruce, el orden de campos, la columna de cruce inmediato y otros detalles del trabajo, y asígnelas a la variable o las propiedades de configuración correspondientes.

   Nota: La regla debe tener el formato JSON.

   Por ejemplo:

```
set rule='{"matchKeys":[{"expressMatchKey":false,
"matchKeyField":"MatchKey1",
"rules":[{"algorithm":"Soundex"," field":"businessname",
"startPosition":1, "length":0,"active":true, "sortInput":null,
"removeNoiseCharacters":false}]},
{"expressMatchKey":false, "matchKeyField":"MatchKey2",
"rules":[{"algorithm":"Koeln", "field":"businessname",
"startPosition":1, "length":0, "active":true, "sortInput":null,
 "removeNoiseCharacters":false}]}]}';
```

   Nota: Asegúrese de usar las propiedades de configuración respectivas para el trabajo. Por ejemplo, `pb.bdq.match.rule`,`pb.bdq.match.express.column`,`pb.bdq.consolidation.sort.field` etc., como se indica en los archivos HQL de muestra correspondientes.

8. Especifique los campos del encabezado de los datos de entrada, en formato separado por comas, y asígnelos a una variable o propiedad de configuración.

```
set pb.bdq.match.header='businessname,recordid';
```

   Nota: Asegúrese de usar la propiedad de configuración donde sea necesario. Por ejemplo, `pb.bdq.match.header`,<codeph>pb.bdq.match.header</codeph>,`pb.bdq.consolidation.head` etc., como se indica en los archivos HQL de muestra correspondientes.

9. Switch off the aggregatiion of data between Reducer and Mapper, by seting the`Hive.Map.Aggr` environment variable configuration to `false`, as indicated in the below example:

```
set hive.map.aggr = false;
```

   Nota: This configuration is required for all UDAFs.

10. Set the general configurations for running the job as indicated in the below example:

```
set pb.bdq.uam.universaladdress.general.configuration =
{"dFileType":"SPLIT", "dMemoryModel":"MEDIUM",
"lacsLinkMemoryModel":"MEDIUM", "suiteLinkMemoryModel":"MEDIUM"};
```

Nota: This configuration is required only for Universal Addressing Module Hive UDAFs.

11. Set the input configurations for running the job as indicated in the below example:

```
set pb.bdq.uam.universaladdress.input.configuration =
{"outputStandardAddress":true, "outputPostalData":false,
"outputParsedInput":false, "outputAddressBlocks":true,
"performUSProcessing":true, "performCanadianProcessing":false,
"performInternationalProcessing":false, "outputFormattedOnFail":false,
 "outputCasing":"MIXED", "outputPostalCodeSeparator":true,
"outputMultinationalCharacters":false, "performDPV":false,
"performRDI":false, "performESM":false, "performASM":false,
"performEWS":false, "performLACSLink":false, "performLOT":false,
"failOnCMRAMatch":false, "extractFirm":false, "extractUrb":false,
"outputReport3553":false, "outputReportSERP":false,
"outputReportSummary":true, "outputCASSDetail":false,
"outputFieldLevelReturnCodes":false, "keepMultimatch":false,
"maximumResults":10,
"standardAddressFormat":"STANDARD_ADDRESS_FORMAT_COMBINED_UNIT",
"standardAddressPMBLine":"STANDARD_ADDRESS_PMB_LINE_NONE",
"cityNameFormat":"CITY_FORMAT_STANDARD", "vanityCityFormatLong":true,
 "outputCountryFormat":"ENGLISH", "homeCountry":"United States",
"streetMatchingStrictness":"MATCHING_STRICTNESS_MEDIUM",
"firmMatchingStrictness":"MATCHING_STRICTNESS_MEDIUM",
"directionalMatchingStrictness":"MATCHING_STRICTNESS_MEDIUM",
"dualAddressLogic":"DUAL_NORMAL", "dpvSuccessfulStatusCondition":"A",
 "reportListFileName":"", "reportlistProcessorName":"",
"reportlistNumber":1, "reportMailerAddress":"", "reportMailerName":"",
 "reportMailerCityLine":"", "canReportMailerCPCNumber":"",
"canReportMailerAddress":"", "canReportMailerName":"",
"canReportMailerCityLine":"", "internationalCityStreetSearching":100,
 "addressLineSearchOnFail":true, "outputStreetAlias":true,
"outputVeriMoveBlock":false, "dpvDetermineNoStat":false,
"dpvDetermineVacancy":false, "outputAbbreviatedAlias":false,
"outputPreferredAlias":false,
"outputPreferredCity":"CITY_OVERRIDE_NAME_ZIP4",
"performSuiteLink":false, "suppressZplusPhantomCarrierR777":false,
"canStandardAddressFormat":"D", "canEnglishApartmentLabel":"APT",
"canFrenchApartmentLabel":"APP", "canFrenchFormat":"C",
"canOutputCityFormat":"D", "canOutputCityAlias":true,
"canDualAddressLogic":"D", "canPreferHouseNum":false,
"canSSLVRFLG":false, "canRuralRouteFormat":"A", "canNonCivicFormat":"A",
 "canDeliveryOfficeFormat":"I", "canEnableSERP":false,
"canSwitchManagedPostalCodeConfidence":false, "stats":null,
"counts":null, "z3seg":null, "serpStats":null, "dpvSeedList":null,
"lacsSeedList":null, "zipInputSet":null, "reportName":null,
```

```
"currentUser":null, "jobName":null, "jobId":null, "jobRequest":false,
 "properties":{"DPVDetermineVacancy":"N", "DualAddressLogic":"N",
"ExtractUrb":"N", "CanFrenchFormat":"C", "AddressLineSearchOnFail":"Y",
 "OutputFieldLevelReturnCodes":"N", "OutputFormattedOnFail":"N",
"OutputStreetNameAlias":"Y", "OutputReportSERP":"N",
"OutputAddressBlocks":"Y", "ExtractFirm":"N",
"CanEnglishApartmentLabel":"APT", "OutputPreferredCity":"Z",
"FirmMatchingStrictness":"M", "CanFrenchApartmentLabel":"APP",
"KeepMultimatch":"N", "StandardAddressPMBLine":"N",
"PerformSuiteLink":"N", "CanStandardAddressFormat":"D",
"DPVSuccessfulStatusCondition":"A", "PerformLACSLink":"N",
"PerformUSProcessing":"Y", "PerformEWS":"N",
"StandardAddressFormat":"C", "SuppressZplusPhantomCarrierR777":"N",
"HomeCountry":"United States", "ReportMailerAddress":"",
"OutputReport3553":"N", "OutputVeriMoveDataBlock":"N",
"CanDeliveryOfficeFormat":"I", "OutputAbbreviatedAlias":"N",
"PerformCanadianProcessing":"N", "PerformDPV":"N",
"PerformInternationalProcessing":"N", "CanSSLVRFlg":"N",
"StreetMatchingStrictness":"M",
"InternationalCityStreetSearching":"100",
"canSwitchManagedPostalCodeConfidence":"N", "CanDualAddressLogic":"D",
 "PerformASM":"N", "OutputCasing":"M", "ReportListFileName":"",
"CanReportMailerAddress":"", "ReportMailerCityLine":"",
"CanReportMailerCPCNumber":"", "ReportListProcessorName":"",
"CanOutputCityAlias":"Y", "DirectionalMatchingStrictness":"M",
"CanRuralRouteFormat":"A", "CanOutputCityFormat":"D",
"ReportListNumber":"1", "CanReportMailerCityLine":"",
"OutputMultinationalCharacters":"N", "EnableSERP":"N",
"CanNonCivicFormat":"A", "OutputShortCityName":"S",
"OutputPostalCodeSeparator":"Y", "FailOnCMRAMatch":"N",
"PerformLOT":"N", "OutputCountryFormat":"E", "CanPreferHouseNum":"N",
 "CanReportMailerName":"", "PerformRDI":"N", "ReportMailerName":"",
"PerformESM":"N", "OutputReportSummary":"Y",
"OutputVanityCityFormatLong":"Y", "OutputPreferredAlias":"N",
"DPVDetermineNoStat":"N", "MaximumResults":"10"}}};
```

Nota: This configuration is required only for Universal Addressing Module Hive UDAFs.

12. Set the engine configurations for running the job as indicated in the below example:

```
set pb.bdq.uam.universaladdress.engine.configurations = {
"referenceData":{
"dataDir":"/home/hduser/resources/uam/universaladdress/UAM_universaladdress4.0_Feb15/",
 "referenceDataPathLocation":"LocaltoDataNodes"},
"cobolRuntimePath":"/home/hduser/tapan/addressquality/",
"modulesDir":"/home/hduser/tapan/addressquality/modules",
"dpvDbPath":null, "suiteLinkDBPath":null, "ewsDBPath":null,
"rdiDBPath":null, "lacsDBPath":null};
```

Nota: This configuration is required only for Universal Addressing Module Hive UDAFs.

13. Set the process type to indicate the desired validation level. We currently support address validation only.

    For example, in the *Validate Address* job, set the *process type* as below:

```
set pb.bdq.uam.universaladdress.process.type=VALIDATE;
```

    Nota: This configuration is required only for the Validate Address and Validate Address Loqate Hive UDAFs.

14. Para hacer el trabajo y ver los resultados en la consola, escriba la consulta como se indica en el siguiente ejemplo:

```
SELECT businessname, recordid, bar.ret["MatchKey1"] AS MatchKey1,
bar.ret["MatchKey2"] AS MatchKey2 FROM (
SELECT *, matchkeygenerator (${hiveconf:rule}, ${hiveconf:header},
businessname, recordid) AS ret FROM cust ) bar;
```

    Para hacer el trabajo y volcar los resultados en un archivo designado, escriba la consulta como se indica en el siguiente ejemplo:

```
INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/MatchKey/' row format
delimited FIELDS TERMINATED BY ',' MAP FIELDS TERMINATED BY ':'
COLLECTION ITEMS TERMINATED BY '|' LINES TERMINATED BY '\n' STORED AS
 TEXTFILE
SELECT businessname, recordid, bar.ret["MatchKey1"] AS MatchKey1,
bar.ret["MatchKey2"] AS MatchKey2 FROM (
SELECT *, matchkeygenerator (${hiveconf:rule}, ${hiveconf:header},
businessname, recordid) AS ret FROM cust ) bar;
```

    Nota: Asegúrese de usar el alias definido antes para la UDF.

Importante: For all UDAF jobs, use the respective configuration properties as variables while defining the input parameters, where indicated in the respective sample HQL files.

Por ejemplo,
`pb.bdq.match.rule`,`pb.bdq.match.express.column`,`pb.bdq.consolidation.sort.field`, etc.

Funciones del módulo Advanced Matching

Match Key Generator

Secuencia de comandos de Hive de ejemplo

```
-- Register Advance Matching Module[AMM] Hive UDF jar
ADD JAR <Directory path>/amm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.
CREATE TEMPORARY FUNCTION matchkeygenerator as
'com.pb.bdq.amm.process.hive.matchkeygenerator.MatchKeyGeneratorUDF';

-- Match Key Generator is implemented as a UDF (User Defined function).
 It processes one row at a time and generates a map of match keys for
each row.

-- Set rule and header
set rule='{"matchKeys":[{"expressMatchKey":false,
"matchKeyField":"MatchKey1",
"rules":[{"algorithm":"Soundex"," field":"businessname",
"startPosition":1, "length":0,"active":true, "sortInput":null,
"removeNoiseCharacters":false}]},
{"expressMatchKey":false, "matchKeyField":"MatchKey2",
"rules":[{"algorithm":"Koeln", "field":"businessname", "startPosition":1,
 "length":0, "active":true, "sortInput":null,
"removeNoiseCharacters":false}]}]}';

set header='businessname,recordid';

-- Execute query on the desired table to display the job output on
console. This query returns a map of key value for each row containing
 matchkeys as per rule passed.
SELECT businessname, recordid, bar.ret["MatchKey1"] AS MatchKey1,
bar.ret["MatchKey2"] AS MatchKey2 FROM (
SELECT *, matchkeygenerator (${hiveconf:rule}, ${hiveconf:header},
businessname, recordid) AS ret FROM cust ) bar;


-- Query to dump output to a directory in file system
INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/MatchKey/' row format
delimited FIELDS TERMINATED BY ',' MAP FIELDS TERMINATED BY ':'
COLLECTION ITEMS TERMINATED BY '|' LINES TERMINATED BY '\n' STORED AS
```

```
TEXTFILE
SELECT businessname, recordid, bar.ret["MatchKey1"] AS MatchKey1,
bar.ret["MatchKey2"] AS MatchKey2 FROM (
SELECT *, matchkeygenerator (${hiveconf:rule}, ${hiveconf:header},
businessname, recordid) AS ret FROM cust ) bar;

--Sample data in input table customer
--+-----------------------------------------+----------------+--+
--|           cust.businessname             | cust.recordid  |
--+-----------------------------------------+----------------+--+
--| Internal Revenue Service                | 0              |
--| Juan F Vera-Monroig                     | 1              |
--| Leonardo Pagan-Reyes                    | 2              |
--| Academia San Joaquin Colegios/Academias | 3              |
--| Nereida Portalatin-Padua                | 4              |
--+-----------------------------------------+----------------+--+


--Sample output for input query
+-----------------------------------+---------+----------+----------------+--+
|               businessname        | recordid | matchkey1 |
    matchkey2     |
+-----------------------------------+---------+----------+----------------+--+
| Internal Revenue Service          | 0        | I536      |
0627657368738    |
| Juan F Vera-Monroig               | 1        | J511      |
063376674        |
| Leonardo Pagan-Reyes              | 2        | L563      |
567214678        |
| Academia San Joaquin Colegios/Academias | 3  | A235      |
0426864645484268 |
| Nereida Portalatin-Padua          | 4        | N631      |
67217252612      |
+-----------------------------------+---------+----------+----------------+--+
```

Interflow Match

Secuencia de comandos de Hive de ejemplo

```
-- Register Advance Matching Module[AMM] Hive UDF jar
ADD JAR <Directory path>/amm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.
CREATE TEMPORARY FUNCTION rowid as
'com.pb.bdq.hive.common.RowIDGeneratorUDF';
```

```
-- This rowid is needed by Interflow Match to maintain the order of rows
 while creating groups. This is a UDF (User Defined Function) and
associates an incremental unique integer number to each row of the data.

CREATE TEMPORARY FUNCTION InterMatch as
'com.pb.bdq.amm.process.hive.interflow.InterMatchUDAF';

-- Inter Flow is implemented as a UDAF (User Defined Aggregation
function). It processes one group of rows at a time based on join field
 and generates the result for that group of rows.

-- Disable map side aggregation
set hive.map.aggr = false;

-- Set the rule using configuration property 'pb.bdq.match.rule'

set pb.bdq.match.rule={"type":"Parent",
"missingDataMethod":"IgnoreBlanks", "threshold":100.0, "weight":0,
"children":[{"type":"Child", "missingDataMethod":"IgnoreBlanks",
"threshold":80.0, "weight":0, "matchWhenNotTrue":false,
"scoringMethod":"Maximum",
"algorithms":[{"name":"EditDistance", "weight":0, "options":null},
{"name":"Metaphone", "weight":0, "options":null}],
"crossMatchField":[], "suspectField":"firstname", "candidateField":null},
{"type":"Child", "missingDataMethod":"IgnoreBlanks", "threshold":80.0,
 "weight":0,
"matchWhenNotTrue":false, "scoringMethod":"Maximum",
"algorithms":[{"name":"KeyboardDistance", "weight":0, "options":null},
{"name":"Metaphone3", "weight":0, "options":null}], "crossMatchField":[],
"suspectField":"lastname", "candidateField":null}],
"scoringMethod":"Average", "matchingMethod":"AllTrue", "name":"NameData",
 "matchWhenNotTrue":false};


-- Set the header for suspect table using configuration property
'pb.bdq.suspect.header'
set
pb.bdq.match.suspect.header=name,firstname,lastname,matchkey,middlename,recordid;

-- Set the header for candidate table using configuration property
'pb.bdq.candidate.header'
set
pb.bdq.match.candidate.header=name,firstname,lastname,matchkey,middlename,recordid;

-- Set the sorting field to the candidates unique id's alias used in
the query. This is not from the input data.
set pb.bdq.match.sort.field=c_id;

-- Set the express match column(optional)
set pb.bdq.match.express.column=matchkey;

-- Set sort field name to the alias used in the query, using
configuration property 'pb.bdq.match.inter.comparison'
```

```
set pb.bdq.match.inter.comparison=maxNumOfDuplicates,2;

-- Optionally, one can also set
'pb.bdq.match.inter.comparison=returnUniqueCandidates,true';

-- Set sort collection number option for unique records using
configuration property 'pb.bdq.match.unique.collectnumber.zero'
set pb.bdq.match.unique.collectnumber.zero=false;

-- Execute Query on the desired table. The query uses a UDF rowid, which
 must be present in the query to maintain the ordering of the data while
 reading.

SELECT lateralview.record ["MatchRecordType"],
 lateralview.record ["MatchScore"],
 lateralview.record ["HasDuplicate"],
 lateralview.record ["CollectionNumber"],
 coalesce(lateralview.record ["ExpressMatched"], ''),
 lateralview.record ["SourceType"],
 lateralview.record ["name"],
 lateralview.record ["firstname"],
 lateralview.record ["lastname"],
 lateralview.record ["matchkey"],
 lateralview.record ["middlename"],
 lateralview.record ["recordid"]
FROM (
 SELECT interMatch(s_id, s_name, s_firstname, s_lastname, s_matchkey,
s_middlename, s_recordid, c_id,c_name, c_firstname, c_lastname,
c_matchkey, c_middlename, c_recordid) AS
 OUTPUT
 FROM (
  SELECT suspects.suspect_id AS s_id,
   suspects.NAME AS s_name,
   suspects.firstname AS s_firstname,
   suspects.lastname AS s_lastname,
   suspects.matchkey AS s_matchkey,
   suspects.middlename AS s_middlename,
   suspects.recordid AS s_recordid,
   candidates.candidate_id AS c_id,
   candidates.NAME AS c_name,
   candidates.firstname AS c_firstname,
   candidates.lastname AS c_lastname,
   candidates.matchkey AS c_matchkey,
   candidates.middlename AS c_middlename,
   candidates.recordid AS c_recordid
  FROM

     (
   SELECT rowid(*) AS suspect_id
    ,*
   FROM namedataintersuspect
   ) AS suspects LEFT JOIN
   (
```

```
    SELECT rowid(*) AS candidate_id
    ,*
    FROM namedataintercandidate
   ) AS candidates
   on suspects.matchkey = candidates.matchkey

  ) AS joinrecords
 GROUP BY joinrecords.s_matchkey
 ) AS innerResult LATERAL VIEW explode(innerResult.OUTPUT) lateralview
 AS record;



-- Query to dump data to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/intermatch/output'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
collection items terminated by '||' map keys terminated by ':'
SELECT lateralview.record ["MatchRecordType"],
 lateralview.record ["MatchScore"],
 lateralview.record ["HasDuplicate"],
 lateralview.record ["CollectionNumber"],
 coalesce(lateralview.record ["ExpressMatched"], ''),
 lateralview.record ["SourceType"],
 lateralview.record ["name"],
 lateralview.record ["firstname"],
 lateralview.record ["lastname"],
 lateralview.record ["matchkey"],
 lateralview.record ["middlename"],
 lateralview.record ["recordid"]
FROM (
 SELECT interMatch(s_id, s_name, s_firstname, s_lastname, s_matchkey,
s_middlename, s_recordid, c_id,c_name, c_firstname, c_lastname,
c_matchkey, c_middlename, c_recordid) AS
 OUTPUT
 FROM (
  SELECT suspects.suspect_id AS s_id,
   suspects.NAME AS s_name,
   suspects.firstname AS s_firstname,
   suspects.lastname AS s_lastname,
   suspects.matchkey AS s_matchkey,
   suspects.middlename AS s_middlename,
   suspects.recordid AS s_recordid,
   candidates.candidate_id AS c_id,
   candidates.NAME AS c_name,
   candidates.firstname AS c_firstname,
   candidates.lastname AS c_lastname,
   candidates.matchkey AS c_matchkey,
   candidates.middlename AS c_middlename,
   candidates.recordid AS c_recordid
  FROM
```

```
    (
   SELECT rowid(*) AS suspect_id
    ,*
   FROM namedataintersuspect
   ) AS suspects LEFT JOIN
   (
    SELECT rowid(*) AS candidate_id
    ,*
    FROM namedataintercandidate
   ) AS candidates
   on suspects.matchkey = candidates.matchkey

  ) AS joinrecords
 GROUP BY joinrecords.s_matchkey
 ) AS innerResult LATERAL VIEW explode(innerResult.OUTPUT) lateralview
 AS record;


-- Sample input Suspect data

--+----------------+---------+---------------+------------+-----------+----------+
--| name             | firstname| lastname        | matchkey    |
middlename | recordid  |
--+----------------+---------+---------------+------------+-----------+----------+
--| LAURA  ABADSANTOS| LAURA    | ABADSANTOS      | L           |
       | 1         |
--+----------------+---------+---------------+------------+-----------+----------+

-- Sample input candidate data

--+----------------+---------+---------------+------------+-----------+----------+
--| name             | firstname| lastname        | matchkey    |
middlename | recordid  |
--+----------------+---------+---------------+------------+-----------+----------+
--| KATHRYN E ABATE  | KATHRYN  | ABATE           | L           |  E
       | 3         |
--| ANNA  ABAYEV     | ANNA     | ABAYEV          | L           |
       | 5         |
--+----------------+---------+---------------+------------+-----------+----------+

-- Sample output data

-+--------+-----+------+--------+--------+-----+------+-----+-----+----+-----+------+
--|MatchRecordType|MatchScore|HasDuplicate|CollectionNumber|ExpressMatched|SourceType|
 name       | firstname| lastname|matchkey|middlename| recordid  |
-+--------+-----+------+--------+--------+-----+------+-----+-----+----+-----+------+
--|S              |0         |Y           |0-0-1           |
    |S          |LAURA ABADSA| LAURA    |ABADSANTO|    L   |          |
 1         |
--|D              |80        |D           |0-0-1           |N
    |C          |KATHRYN E AB| KATHRYN  |AB       |    L   |    E     |
 3         |
--|D              |90        |D           |0-0-1           |N
```

```
    |C          |ANNA ABAYEV | ANNA     |ABAYEV    |    L    |           |
 5          |
-+----------+-----+-------+----------+---------+------+-------+-----+-----+-----+-------+
```

Intraflow Match

Secuencia de comandos de Hive de ejemplo

```
-- Register Advance Matching Module[AMM] Hive UDF jar
ADD JAR <Directory path>/amm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.
CREATE TEMPORARY FUNCTION rowid as
'com.pb.bdq.hive.common.RowIDGeneratorUDF';

-- This rowid is needed by Intraflow Match to maintain the order of rows
 while creating groups. This is a UDF (User Defined Function) and
associates an incremental unique integer number to each row of the data.

CREATE TEMPORARY FUNCTION intraMatch as
'com.pb.bdq.amm.process.hive.intraflow.IntraMatchUDAF';
-- Intra Flow is implemented as a UDAF (User Defined Aggregation
function). It processes one group of rows at a time and generates the
result for that group of rows

-- Disable map side aggregation
set hive.map.aggr = false;

-- Set the rule using configuration property 'pb.bdq.match.rule'
set pb.bdq.match.rule={"type":"Parent",
"children":[{"type":"Child", "matchWhenNotTrue":false, "threshold":80.0,
 "weight":0,
"algorithms":[{"name":"EditDistance", "weight":0, "options":null},
{"name":"Metaphone", "weight":0, "options":null}],
"scoringMethod":"Maximum", "missingDataMethod":"IgnoreBlanks",
"crossMatchField":[], "suspectField":"firstname", "candidateField":null},
{"type":"Child", "matchWhenNotTrue":false, "threshold":80.0, "weight":0,
 "algorithms":[{"name":"KeyboardDistance", "weight":0, "options":null},
{"name":"Metaphone3", "weight":0, "options":null}],
"scoringMethod":"Maximum", "missingDataMethod":"IgnoreBlanks",
"crossMatchField":[], "suspectField":"lastname", "candidateField":null}],
"matchingMethod":"AllTrue", "scoringMethod":"Average",
"missingDataMethod":"IgnoreBlanks", "name":"NameData",
"matchWhenNotTrue":false, "threshold":100,"weight":0};
```

```
-- Set header(along with id field alias used in query) using
configuration property 'pb.bdq.match.header'
set pb.bdq.match.header=firstname,lastname,matchkey,middlename,id;

-- Set the express match column (optional)
set pb.bdq.match.express.column=matchkey;

-- Set sort field name to the alias used in the query, using the
configuration property 'pb.bdq.match.sort.field'
set pb.bdq.match.sort.field=id;

-- Set sort collection number option for unique records using
configuration property 'pb.bdq.match.unique.collectnumber.zero'
set pb.bdq.match.unique.collectnumber.zero=false;

-- Execute Query on the desired table. The query uses a UDF rowid, which
 must be present in the query to maintain the ordering of the data while
 reading.
-- Intra Match returns a list of map containing <key=value> pairs. Each
 map in the list corresponds to a row in the group. The below query
explodes that list of map and fetches fields from map by keys.

SELECT innerresult.record["MatchRecordType"],
 innerresult.record["MatchScore"],
 innerresult.record["CollectionNumber"],
 innerresult.record["ExpressMatched"],
 innerresult.record["firstname"],
 innerresult.record["lastname"],
 innerresult.record["matchkey"],
 innerresult.record["middlename"]
 FROM (
  SELECT  intraMatch(
    innerRowID.firstname,
    innerRowID.lastname,
    innerRowID.matchkey,
    innerRowID.middlename,
    innerRowID.id
  ) AS matchgroup
  FROM (
   SELECT  firstname, lastname, matchkey, middlename, rowid(*)
   AS id
   FROM customer_data
   ) innerRowID
  GROUP BY matchkey
 ) AS innerResult
 LATERAL VIEW explode(innerResult.matchgroup) innerresult AS record ;

-- Query to dump output to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/IntraFlow/' ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',' collection items terminated by '||'
 map keys terminated by ':'
SELECT innerresult.record["MatchRecordType"],
```

```
 innerresult.record["MatchScore"],
 innerresult.record["CollectionNumber"],
 innerresult.record["ExpressMatched"],
 innerresult.record["firstname"],
 innerresult.record["lastname"],
 innerresult.record["matchkey"],
 innerresult.record["middlename"]
 FROM (
  SELECT  intraMatch(innerRowID.firstname,
    innerRowID.lastname,
    innerRowID.matchkey,
    innerRowID.middlename,
    innerRowID.id
  ) AS matchgroup
  FROM (
    SELECT  firstname, lastname, matchkey, middlename, rowid(*)
    AS id
    FROM customer_data
   ) innerRowID
  GROUP BY matchkey
  ) AS innerResult
 LATERAL VIEW explode(innerResult.matchgroup) innerresult AS record ;


--sample input data
--+-------------+-------------+---------------+--------------+
--| firstname   | lastname    | middlename    | matchkey     |
--+-------------+-------------+---------------+--------------+
--| Steven      | Aaen        | LYRIC         | AAE          |
--| DEBRA       | AALMO       | BOATMAN       | AAE          |

--| MARY        | AARON       | ROLLING MEADOW| AAE          |
--+-------------+-------------+---------------+--------------+

--sample output data
—+———————+——————+————————+——————+——————————+——————————+—————————+———————+
--| firstname | lastname|middlename  |
matchkey|MatchRecordType|CollectionNumber|ExpressMatched|MatchScore|
—+———————+——————+————————+——————+——————————+——————————+—————————+———————+
--| Steven    | Aaen    | LYRIC      | AAE     | S        |
0-0-1  |      Y        |     0    |
--| DEBRA     | AALMO   | BOATMAN    | AAE     |    D          |
  0-0-1  |      Y        |    100   |
--| MARY      | AARON   | ROLLING MEA| AAE     |    D           |
   0-0-1  |      Y        |    100   |
```

```
—+————————+————————+————————+————————+————————————+————————————+————————————+————————+
```

Transactional Match

Secuencia de comandos de Hive de ejemplo

```
-- Register Advance Matching Module[AMM] Hive UDF jar
ADD JAR <Directory path>/amm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.

CREATE TEMPORARY FUNCTION rowid as
'com.pb.bdq.hive.common.RowIDGeneratorUDF';

-- This rowid is needed by Transactional Match to maintain the order of
 rows while creating groups. This is a UDF (User Defined Function) and
 associates an incremental unique integer number to each row of the
data.

CREATE TEMPORARY FUNCTION transactionalMatch as
'com.pb.bdq.amm.process.hive.transactional.TransactionalMatchUDAF';

-- Transactional Match is implemented as a UDAF (User Defined Aggregation
 function). It processes one group of rows at a time and generates the
 result for that group of rows.

-- Disable map side aggregation
set hive.map.aggr = false;

-- Set the rule using configuration property 'pb.bdq.match.rule'
set pb.bdq.match.rule={"type":"Parent", "children":[{"type":"Child",
"matchWhenNotTrue":false, "threshold":80.0, "weight":0,
"algorithms":[{"name":"EditDistance", "weight":0, "options":null},
{"name":"Metaphone","weight":0,"options":null}],
"scoringMethod":"Maximum", "missingDataMethod":"IgnoreBlanks",
"crossMatchField":[], "suspectField":"firstname", "candidateField":null},
{"type":"Child", "matchWhenNotTrue":false, "threshold":80.0, "weight":0,
 "algorithms":[{"name":"KeyboardDistance","weight":0,"options":null},
{"name":"Metaphone3","weight":0,"options":null}],
"scoringMethod":"Maximum", "missingDataMethod":"IgnoreBlanks",
"crossMatchField":[], "suspectField":"lastname", "candidateField":null}],
 "matchingMethod":"AllTrue", "scoringMethod":"Average",
"missingDataMethod":"IgnoreBlanks", "name":"NameData",
"matchWhenNotTrue":false, "threshold":100, "weight":0};

-- Set header(along with id field alias used in query) using
```

```
configuration property 'pb.bdq.match.header'
set
pb.bdq.match.header=name,firstname,lastname,matchkey,middlename,recordid,id;

-- Set sort field name to the alias used in the query, using the
configuration property 'pb.bdq.match.sort.field'
set pb.bdq.match.sort.field=id;

-- Set sort collection number option for unique records using
configuration property 'pb.bdq.match.unique.candidate.return'. The
default value is false.
set pb.bdq.match.unique.candidate.return=true;

-- Execute Query on the desired table. The query uses a UDF rowid, which
 must be present in the query to maintain the ordering of the data while
 reading.
-- Transactional Match returns a list of map containing <key=value>
pairs. Each map in the list corresponds to a row in the group. The below
 query explodes that list of map and fetches fields from map by keys.

SELECT tmp2.record["MatchRecordType"],
 tmp2.record["MatchScore"],
 tmp2.record["HasDuplicate"],
 tmp2.record["name"],
 tmp2.record["firstname"],
 tmp2.record["lastname"],
 tmp2.record["matchkey"],
 tmp2.record["middlename"],
 tmp2.record["recordid"]
FROM (
 SELECT  transactionalMatch(innerRowID.name, innerRowID.firstname,
innerRowID.lastname, innerRowID.matchkey, innerRowID.middlename,
innerRowID.recordid, innerRowID.id
 ) AS matchgroup
 FROM (
  SELECT name, firstname, lastname, matchkey, middlename, recordid,
rowid(name, firstname, lastname, matchkey, middlename, recordid) AS id
 FROM customer_data
  ) innerRowID
 GROUP BY matchkey
) As innerResult
LATERAL VIEW explode(innerResult.matchgroup) tmp2 as record ;


-- Query to dump output to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/transmatch/' ROW FORMAT
 DELIMITED FIELDS TERMINATED BY ',' collection items terminated by '||'
 map keys terminated by ':'
SELECT tmp2.record["MatchRecordType"],
 tmp2.record["MatchScore"],
 tmp2.record["HasDuplicate"],
 tmp2.record["name"],
```

```
 tmp2.record["firstname"],
 tmp2.record["lastname"],
 tmp2.record["matchkey"],
 tmp2.record["middlename"],
 tmp2.record["recordid"]
FROM (
 SELECT  transactionalMatch(innerRowID.name,
   innerRowID.firstname,
   innerRowID.lastname,
   innerRowID.matchkey,
   innerRowID.middlename,
   innerRowID.recordid,
   innerRowID.id) as matchgroup
 FROM (
  SELECT name, firstname, lastname, matchkey, middlename, recordid,
rowid(name, firstname, lastname, matchkey, middlename, recordid) AS id

  FROM customer_data
  ) innerRowID
 GROUP BY matchkey ) As innerResult
LATERAL VIEW explode(innerResult.matchgroup) tmp2 as record ;



--sample input data
-+----------+-------------+-------------+-------------+-------------+-+
--| name               | firstname               | lastname
 | matchkey              | middlename              | recordid
     |
-+----------+-------------+-------------+-------------+-------------+-+
--| ZORINA  ABDOOL     | ZORINA                  | ABDOOL
 | Z                     |                         | 12
     |
--| ZULFIQAR  ALI      | ZULFIQAR                | ALI
 | Z                     |                         | 116
     |
--| ZACHARY  BENNETT   | ZACHARY                 | BENNETT
 | Z                     |                         | 515
     |
--| ZOHAR  BUERGER     | ZOHAR                   | BUERGER
 | Z                     |                         | 889
     |
-+----------+-------------+-------------+-------------+-------------+-+

--sample output data
-+----------+-------------+-------+-------+-------+-----------+-------+-------+
--|name            |firstname | lastname | matchkey | middlename |
recordid  | MatchRecordType | MatchScore | HasDuplicate |
-+----------+-------+-------+-------+-------+-------+-----------+-------+-------+
--|ZORINA  ABDOOL  |ZORINA    | ABDOOL   | Z        |           | 12
       |  S     | 0  |  Y     |
--|ZULFIQAR  ALI   |ZULFIQAR  | ALI      | Z        |           | 116
       |      D          | 90  |  D    |
```

```
--|ZACHARY  BENNETT|ZACHARY    | BENNETT   | Z         |             | 515
       |      D          | 91   |  D    |
--|ZOHAR  BUERGER  |ZOHAR      | BUERGER   | Z         |             | 889
       |  D     | 91   |  D    |
--+----------+-------+---------------+-----------+---------+---------+
```

Best of Breed

Secuencia de comandos de Hive de ejemplo

```
-- Register Advance Matching Module[AMM] Hive UDF jar
ADD JAR <Directory path>/amm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.

CREATE TEMPORARY FUNCTION rowid as
'com.pb.bdq.hive.common.RowIDGeneratorUDF';

-- This rowid is needed by Best of Breed to maintain the order of rows
 while creating groups. This is a UDF (User Defined Function) and
associates an incremental unique integer number to each row of the data.

CREATE TEMPORARY FUNCTION bestofbreed as
'com.pb.bdq.amm.process.hive.consolidation.bestofbreed.BestOfBreedUDAF';
-- Best of Breed is implemented as a UDAF (User Defined Aggregation
function). It processes one group of rows at a time and generates the
result for that group of rows.

-- Disable map side aggregation
set hive.map.aggr = false;

-- Set the rule using configuration property 'pb.bdq.consolidation.rule'

set pb.bdq.consolidation.rule={"consolidationConditions":[
{"consolidationRule":{"conditionClass":"conjoinedRule", "joinType":"AND",
"consolidationRules":[{"conditionClass":"simpleRule",
"operation":"LONGEST", "fieldName":"c5", "value":null,
"valueNumeric":true, "valueFromField":false},
{"conditionClass":"simpleRule", "operation":"IS_NOT_EMPTY",
"fieldName":"c9", "value":null, "valueNumeric":false,
"valueFromField":false}]},
"actions":[{"accumulate":false, "copyFromField":true, "sourceData":"c2",
 "destinationFieldName":"c2"},
{"accumulate":false, "copyFromField":false, "sourceData":"Admin",
"destinationFieldName":"c4"}]},
{"consolidationRule":{"conditionClass":"conjoinedRule", "joinType":"AND",
```

```
"consolidationRules":[{"conditionClass":"simpleRule",
"operation":"LONGEST", "fieldName":"c5", "value":null,
"valueNumeric":true, "valueFromField":false},
{"conditionClass":"simpleRule", "operation":"IS_NOT_EMPTY",
"fieldName":"c9", "value":null, "valueNumeric":false,
"valueFromField":false}]},
"actions":[{"accumulate":false, "copyFromField":false,
"sourceData":"Changed", "destinationFieldName":"c10"},
{"accumulate":false, "copyFromField":true, "sourceData":"c5",
"destinationFieldName":"c6"},
{"accumulate":true, "copyFromField":true, "sourceData":"c10",
"destinationFieldName":"c10"}]}],
"keepOriginalRecords":true, "buildTemplateRecord":true,
"templateRules":[{"consolidationRule":{"conditionClass":"conjoinedRule",
 "joinType":"OR",
"consolidationRules":[{"conditionClass":"simpleRule",
"operation":"CONTAINS", "fieldName":"c1", "value":"li",
"valueNumeric":false, "valueFromField":false},
{"conditionClass":"simpleRule", "operation":"LONGEST", "fieldName":"c5",
 "value":null, "valueNumeric":false, "valueFromField":false}]},
"actions":[]}]};

-- Set header (along with the id field alias used in the query) using
configuration property 'pb.bdq.consolidation.header'
set pb.bdq.consolidation.header=c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,id;

-- Set sort field name to the alias used in the query, using the
configuration property 'pb.bdq.consolidation.sort.field'
set pb.bdq.consolidation.sort.field=id;

-- Execute Query on the desired table. The query uses a UDF rowid, which
 must be present in the query to maintain the ordering of the data while
 reading.
-- Best of Breed returns a list of map containing <key=value> pairs.
Each map in the list corresponds to a row in the group. The below query
 explodes that list of map and fetches fields from map by keys.

SELECT tmp2.record["c1"],
 tmp2.record["c2"],
 tmp2.record["c3"],
 tmp2.record["c4"],
 tmp2.record["c5"],
 tmp2.record["c6"],
 tmp2.record["c7"],
 tmp2.record["c8"],
 tmp2.record["c9"],
 tmp2.record["c10"],
 tmp2.record["CollectionRecordType"]
FROM (
 SELECT  bestofbreed(innerRowID.c1,
  innerRowID.c2,
  innerRowID.c3,
  innerRowID.c4,
```

```
  innerRowID.c5,
  innerRowID.c6,
  innerRowID.c7,
  innerRowID.c8,
  innerRowID.c9,
  innerRowID.c10,
  innerRowID.id) AS matchgroup
 FROM(
  SELECT c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, rowid(*) AS id FROM
databob
  ) innerRowID
 GROUP BY c3
 ) AS innerResult
LATERAL VIEW explode(innerResult.matchgroup) tmp2 AS record ;

-- Query to dump the output to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/bestofbreed/' ROW FORMAT
 DELIMITED FIELDS TERMINATED BY ',' collection items terminated by '||'
 map keys terminated by ':'
SELECT tmp2.record["c1"],
 tmp2.record["c2"],
 tmp2.record["c3"],
 tmp2.record["c4"],
 tmp2.record["c5"],
 tmp2.record["c6"],
 tmp2.record["c7"],
 tmp2.record["c8"],
 tmp2.record["c9"],
 tmp2.record["c10"],
 tmp2.record["CollectionRecordType"]
FROM (
 SELECT bestofbreed(innerRowID.c1,
  innerRowID.c2,
  innerRowID.c3,
  innerRowID.c4,
  innerRowID.c5,
  innerRowID.c6,
  innerRowID.c7,
  innerRowID.c8,
  innerRowID.c9,
  innerRowID.c10,
  innerRowID.id) as matchgroup
 FROM(
  SELECT c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, rowid(*) AS id FROM
databob
  ) innerRowID
 GROUP BY c3
 ) AS innerResult
LATERAL VIEW explode(innerResult.matchgroup) tmp2 AS record ;


--sample input data
```

```
-+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
--| c1       |      c2  |      c3  |      c4  |      c5  |      c6  |
     c7  |      c8  | c9       | c10      |
-+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+
--| Duplicate| 87       | 1        |          |ANNA ABNEY| ANNA     |
         | ABNEY    | A        | 18       |
--| Duplicate| 77       | 1        |          |ANNA A ANN| ANDREA   |
         | ANNAKAY  | A        | 196      |
-+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+

--sample output data
-+------+------+------+------+------+------+------+------+------+-------------+
--| c1       | c2       | c3       | c4       | c5       | c6       |
c7       | c8       | c9       | c10      |CollectionRecordType|
-+------+------+------+------+------+------+------+------+------+-------------+
--| Duplicate| 87       | 1        |          |ANNA ABNEY| ANNA     |
         | ABNEY    | A        | 18       |   Primary          |
--| Duplicate| 77       | 1        |          |ANNA A ANN| ANDREA   |
 ARANOW  | ANNAKAY  | A        | 196      |   Secondary        |
--| Duplicate| 87       | 1        |          |ANNA ABNEY| ANNA     |
 ARANOW  | ABNEY    | A        | 18       |   BestOfBreed      |
-+------+------+------+------+------+------+------+------+------+-------------+
```

Duplicate Synchronization

Secuencia de comandos de Hive de ejemplo

```
-- Register Advance Matching Module[AMM] Hive UDF jar
ADD JAR <Directory path>/amm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.

CREATE TEMPORARY FUNCTION rowid as
'com.pb.bdq.hive.common.RowIDGeneratorUDF';

-- This rowid is needed by Duplicate Synchronization to maintain the
order of rows while creating groups. This is a UDF (User Defined
Function) and associates an incremental unique integer number to each
row of the data.

CREATE TEMPORARY FUNCTION dupsync as
'com.pb.bdq.amm.process.hive.consolidation.duplicatesync.DuplicateSyncUDAF';

-- Duplicate Sync is implemented as a UDAF (User Defined Aggregation
function). It processes one group of rows at a time and generates the
result for that group of rows.
```

```
-- Disable map side aggregation
set hive.map.aggr = false;

-- Set the rule using configuration property 'pb.bdq.consolidation.rule'

set pb.bdq.consolidation.rule={"consolidationConditions":
[{"consolidationRule":
{"conditionClass":"conjoinedRule", "joinType":"AND",
"consolidationRules":[{"conditionClass":"simpleRule",
"operation":"HIGHEST", "fieldName":"column2", "value":null,
"valueFromField":false, "valueNumeric":true}]},
"actions":[{"accumulate":false, "copyFromField":true,
"sourceData":"column5", "destinationFieldName":"column5"}]}]};

-- Set header (along with the id field alias used in the query) using
configuration property 'pb.bdq.consolidation.header'
set
pb.bdq.consolidation.header=column1,column2,column3,column4,column5,id;

-- Set sort field name to alias used in query using configuration
property 'pb.bdq.consolidation.sort.field'
set pb.bdq.consolidation.sort.field=id;

-- Execute Query on the desired table. The query uses a UDF rowid, which
 must be present in the query to maintain the ordering of the data while
 reading.
-- Duplicate Sync returns a list of map containing <key=value> pairs.
Each map in the list corresponds to a row in the group. The below query
 explodes that list of map and fetches fields from map by keys.

SELECT tmp2.record["column1"],
 tmp2.record["column2"],
 tmp2.record["column3"],
 tmp2.record["column4"],
 tmp2.record["column5"]
FROM (
 SELECT  dupsync (innerRowID.column1,
  innerRowID.column2,
  innerRowID.column3,
  innerRowID.column4,
  innerRowID.column5,
  innerRowID.id
  ) AS matchgroup
 FROM (
  SELECT column1, column2, column3, column4, column5, rowid(*)
  AS id
  FROM databob
  ) innerRowID
 GROUP BY column3
 ) AS innerResult
 LATERAL VIEW explode(innerResult.matchgroup) tmp2 AS record ;
```

```
-- Query to dump the output to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/dupsync/' ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',' collection items terminated by '||'
 map keys terminated by ':'
SELECT tmp2.record["column1"],
 tmp2.record["column2"],
 tmp2.record["column3"],
 tmp2.record["column4"],
 tmp2.record["column5"]
FROM (
 SELECT dupsync( innerRowID.column1,
  innerRowID.column2,
  innerRowID.column3,
  innerRowID.column4,
  innerRowID.column5,
  innerRowID.id
  ) AS matchgroup
 FROM (
  SELECT column1, column2, column3, column4, column5, rowid(*)
  AS id
  FROM databob
  ) innerRowID
 GROUP BY column3 ) AS innerResult
 LATERAL VIEW explode(innerResult.matchgroup) tmp2 AS record ;



--sample input data
--+----------+----------+----------+----------+----------+
--| column1  | column2  | column3  | column4  | column5  |
--+----------+----------+----------+----------+----------+
--| Duplicate| 87       | 1        |          |ANNA ABNEY|
--| Duplicate| 77       | 1        |          |ANNA A ANN|
--| Suspect  |          | 1        |          |ANNA A ABN|
--+----------+----------+----------+----------+----------+

--sample output data
--+----------+----------+----------+----------+----------+
--| column1  | column2  | column3  | column4  | column5  |
--+----------+----------+----------+----------+----------+
--| Duplicate| 87       | 1        |          |ANNA ABNEY|
--| Duplicate| 77       | 1        |          |ANNA A ANN|
```

```
--| Suspect  |          | 1        |          |ANNA ABNEY|
--+----------+----------+----------+----------+----------+
```

Filtro

Secuencia de comandos de Hive de ejemplo

```
-- Register Advance Matching Module[AMM] Hive UDF jar
ADD JAR <Directory path>/amm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.

CREATE TEMPORARY FUNCTION rowid as
'com.pb.bdq.hive.common.RowIDGeneratorUDF';

-- This rowid is needed by Filter to maintain the order of rows while
creating groups. This is a UDF (User Defined Function) and associates
an incremental unique integer number to each row of the data.

CREATE TEMPORARY FUNCTION filter as
'com.pb.bdq.amm.process.hive.consolidation.filter.FilterUDAF';

-- Filter is implemented as a UDAF (User Defined Aggregation function).
 It processes one group of rows at a time and generates the result for
 that group of rows.

-- Disable map side aggregation
set hive.map.aggr = false;

-- Set the rule using configuration property 'pb.bdq.consolidation.rule'
set pb.bdq.consolidation.rule={"consolidationConditions":
[{"consolidationRule":{"conditionClass":"simpleRule",
"operation":"HIGHEST", "fieldName":"column2", "value":null,
"valueFromField":false, "valueNumeric":true}, "actions":[]}],
"removeDuplicates":true};

-- Set header (along with the id field alias used in the query) using
configuration property 'pb.bdq.consolidation.header'
set
pb.bdq.consolidation.header=column1,column2,column3,column4,column5,id;

-- Set sort field name to alias used in query using configuration
property 'pb.bdq.consolidation.sort.field'
set pb.bdq.consolidation.sort.field=id;

-- Execute Query on the desired table. The query uses a UDF rowid, which
 must be present in the query to maintain the ordering of the data while
```

```
 reading.

SELECT tmp2.record["column1"],
 tmp2.record["column2"],
 tmp2.record["column3"],
 tmp2.record["column4"],
 tmp2.record["column5"]
FROM (
 SELECT  filter (innerRowID.column1,
   innerRowID.column2,
   innerRowID.column3,
   innerRowID.column4,
   innerRowID.column5,
   innerRowID.id
 ) AS matchgroup
 FROM (
  SELECT column1, column2, column3, column4, column5, rowid(*)
  AS id
  FROM data
  ) innerRowID
 GROUP BY column3
 ) AS innerResult
LATERAL VIEW explode(innerResult.matchgroup) tmp2 AS record ;

-- Query to dump the output to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/HiveUDF/filter/'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
collection items terminated by '||' map keys terminated by ':'
SELECT tmp2.record["column1"],
 tmp2.record["column2"],
 tmp2.record["column3"],
 tmp2.record["column4"],
 tmp2.record["column5"]
FROM (
 SELECT  filter (innerRowID.column1,
   innerRowID.column2,
   innerRowID.column3,
   innerRowID.column4,
   innerRowID.column5,
   innerRowID.id
 ) AS matchgroup
 FROM (
  SELECT column1, column2, column3, column4, column5, rowid(*)
  AS id
  FROM data
  ) innerRowID
 GROUP BY column3
 ) AS innerResult
LATERAL VIEW explode(innerResult.matchgroup) tmp2 AS record ;
```

```
--sample input data
--+----------+----------+----------+----------+----------+
--| column1  | column2  | column3  | column4  | column5  |
--+----------+----------+----------+----------+----------+
--| Duplicate| 80       | 98       |          | EUNICE L |
--| Suspect  |          | 98       |          | ERIC L BR|
--+----------+----------+----------+----------+----------+

--sample output data
--+----------+----------+----------+----------+----------+
--| column1  | column2  | column3  | column4  | column5  |
--+----------+----------+----------+----------+----------+
--| Suspect  |          | 98       |          | ERIC L BR|
--+----------+----------+----------+----------+----------+
```

Funciones del módulo Data Normalization

Table Lookup

Secuencia de comandos de Hive de ejemplo

```
-- Register Data Normalization Modue [dnm] BDQ Hive UDF Jar
ADD JAR <Directory path>/dnm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.
-- Table Lookup is implemented as a UDF (User Defined function). Hence
 it processes one row at a time and generates a map of key value pairs
 for each row.
CREATE TEMPORARY FUNCTION tablelookup as
'com.pb.bdq.dnm.process.hive.tablelookup.TableLookUpUDF';

-- Set rule
set rule='{"rules":[{"action":"Standardize", "source":"CityCode",
"tableName":"State Name Abbreviations", "lookupMultipleWordTerms":false,
 "lookupIndividualTermsWithinField":false, "destination":"CityCode"}]}';

-- Set Reference Directory. This must be a local path on cluster machines
 and must be present on each node of the cluster at the same path.
set refdir='/home/hadoop/reference';

-- set header
set header ='AccountDescription,Address,ApartmentNumber,CityCode';
```

```
-- Execute Query on the desired table, to display the job output on
console. This query returns a map of key value pairs containing output
 fields for each row.

SELECT bar.ret["StandardizationTermIdentified"],
 bar.ret["accountdescription"],
 bar.ret["address"],
 bar.ret["apartmentnumber"],
 bar.ret["citycode"]
FROM (
 SELECT tablelookup(${hiveconf:rule}, ${hiveconf:refdir},
${hiveconf:header}, accountdescription, address, apartmentnumber,
citycode)
 AS ret
 FROM citizen_data
 ) bar;


-- Query to dump output data to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/TableLookup/' row format
 delimited FIELDS TERMINATED BY ',' lines terminated by '\n' STORED AS
 TEXTFILE
SELECT bar.ret["StandardizationTermIdentified"],
 bar.ret["accountdescription"],
 bar.ret["address"],
 bar.ret["apartmentnumber"],
 bar.ret["citycode"]
FROM (
 SELECT tablelookup(${hiveconf:rule}, ${hiveconf:refdir},
${hiveconf:header}, accountdescription, address, apartmentnumber,
citycode)
 AS ret
 FROM citizen_data
 ) bar;

--Sample input data
--+-------------------------+---------------+----------------------+----------------+
--| citizen_data.accountdescription  | citizen_data.address  |
citizen_data.apartmentnumber  | citizen_data.citycode |
--+-------------------------+---------------+----------------------+----------------+
--|                         | 400 E M0 St Apt 1405  |
            | NY                        |
--|                                     | 190 E 72nd St         |
                        | NY                          |
--|                                     | 1381 3rd Ave Apt 4    | 4
                        | TTYYY                       |
--+-------------------------+---------------+----------------------+----------------+


--sample output data
```

```
-+--------------------+-----------------+---------------+---------------+-------+

--|StandardizationTermIdentified |  accountdescription   | address
  | apartmentnumber    | citycode|
-+--------------------+-----------------+---------------+---------------+-------+

--|    yes     |              | 400 E M0 St Apt 1405  |
       | NEW YORK |
--|    yes          |                            | 190 E 72nd St
|                       | NEW YORK |
--|    yes     |                           | 1381 3rd Ave Apt 4    | 4
                     | NEW YORK |
-+--------------------+-----------------+---------------+---------------+-------+
```

Advanced Transformer

Secuencia de comandos de Hive de ejemplo

```
-- Register Data Normalisation Module [DNM] BDQ Hive UDF Jar
ADD JAR <Directory path>/dnm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.
-- Advanced Transformer is implemented as a UDF (User Defined function).
 Hence it processes one row at a time and generates a map of key value
 pairs for each row.
CREATE TEMPORARY FUNCTION advanceTransform as
'com.pb.bdq.dnm.process.hive.advancetransformer.AdvanceTransformerUDF';

-- Set rule
set rule='{"rules":[{"extractionType":"TableData", "source":"address",
 "nonExtractedData":"address_1", "extractedData":"address_2",
"tokenizationCharacters":"", "tableName":"Street Suffix Abbreviations",
 "multipleTermLookup":false, "tokenize":true, "extract":"ExtractTerm",
 "includeTermWith":"ExtractedData", "wordsToExtract":2}]}';


-- Set Reference Directory. This must be a local path on cluster machines
 and must be present on each node of the cluster at the same path.
set refdir='/home/hadoop/reference/';

-- set header
set header ='AccountDescription,Address';

-- Execute Query on the desired table, to display the job output on
console. This query returns a map of key value pairs containing output
 fields for each row.
```

```
SELECT bar.ret["AdvancedTransformTermIdentified"],
 bar.ret["accountdescription"],
 bar.ret["address"],
 bar.ret["address_1"]
FROM (
 SELECT advanceTransform(${hiveconf:rule}, ${hiveconf:refdir},
${hiveconf:header}, accountdescription, address)
 AS ret
 FROM advxformX
 ) bar;


-- Query to dump output data to a file

INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/AdvXformer/' row format
 delimited FIELDS TERMINATED BY ',' lines terminated by '\n' STORED AS
 TEXTFILE
SELECT bar.ret["AdvancedTransformTermIdentified"],
 bar.ret["accountdescription"],
 bar.ret["address"],
 bar.ret["address_1"]
FROM (
 SELECT advanceTransform(${hiveconf:rule}, ${hiveconf:refdir},
${hiveconf:header}, accountdescription, address)
 AS ret
 FROM advxformX
 ) bar;

--sample input data
+--------------------------------+--------------------+----------------------+
| AdvancedTransformTermIdentified  | accountdescription  |      address
          |
+--------------------------------+--------------------+----------------------+
| Yes                                |            | 400 E M0 St Apt 1405
 |
| Yes                                |                     | 190 E 72nd
St          |
+--------------------------------+--------------------+----------------------+

--sample output data
+-----------------------------+-----------------+-------------------+-----------------+
| AdvancedTransformTermIdentified  | accountdescription  |      address
          |     address_1      |
+-----------------------------+-----------------+-------------------+-----------------+
| Yes                                |            | 400 E M0 St Apt 1405
 | 400 E M0 Apt 1405  |
| Yes                                |                     | 190 E 72nd
St          | 190 E 72nd          |
```

```
+-----------------------------+-----------------+-----------------+-----------------+
```

Funciones del módulo Universal Addressing

Validate Address

Atención: Before creating and running the first Validate Address job, ensure the Acushare service is running. Para obtener información sobre los pasos, consulte **Running Acushare Service** en la página 11.

Secuencia de comandos de Hive de ejemplo

```
-- Register Universal Addressing Module [UAM-Global] BDQ Hive UDAF Jar

ADD JAR <Directory
path>/uam.universaladdress.hive.${project.version}.jar;

-- Provide alias to UDAF class (optional). String in quotes represent
class names needed for this job to run.
CREATE TEMPORARY FUNCTION uamvalidation as
'com.pb.bdq.uam.process.hive.universaladdress.UAMUSAddressingUDAF';

-- set LD_LIBRARY_PATH(path to modules lib, runtime/lib and runtime/bin),
 G1RTS(path containing COBOL runtime) and ACU_RUNCBL_JNI_ONLOAD_DISABLE
 in this configuration
set mapreduce.admin.user.env =
LD_LIBRARY_PATH=/home/hduser/~/runtime/lib:
/home/hduser/~/runtime/bin:/home/hduser/~/server/modules/universaladdress/lib,
 ACU_RUNCBL_JNI_ONLOAD_DISABLE=1, G1RTS=/home/hduser/~/ ;

set hive.map.aggr = false;

-- set engine configuration
set pb.bdq.uam.universaladdress.engine.configurations={ "referenceData":{

"dataDir":"/home/hduser/resources/uam/universaladdress/UAM_universaladdress4.0_Feb15/",
 "referenceDataPathLocation":"LocaltoDataNodes"},
"cobolRuntimePath":"/home/hduser/tapan/addressquality/",
"modulesDir":"/home/hduser/tapan/addressquality/modules",
"dpvDbPath":null, "suiteLinkDBPath":null, "ewsDBPath":null,
"rdiDBPath":null, "lacsDBPath":null};
```

```
-- set input configuration
set
pb.bdq.uam.universaladdress.input.configuration={"outputStandardAddress":true,
 "outputPostalData":false, "outputParsedInput":false,
"outputAddressBlocks":true, "performUSProcessing":true,
"performCanadianProcessing":false,
"performInternationalProcessing":false, "outputFormattedOnFail":false,
 "outputCasing":"MIXED", "outputPostalCodeSeparator":true,
"outputMultinationalCharacters":false, "performDPV":false,
"performRDI":false, "performESM":false, "performASM":false,
"performEWS":false, "performLACSLink":false, "performLOT":false,
"failOnCMRAMatch":false, "extractFirm":false, "extractUrb":false,
"outputReport3553":false, "outputReportSERP":false,
"outputReportSummary":true, "outputCASSDetail":false,
"outputFieldLevelReturnCodes":false, "keepMultimatch":false,
"maximumResults":10,
"standardAddressFormat":"STANDARD_ADDRESS_FORMAT_COMBINED_UNIT",
"standardAddressPMBLine":"STANDARD_ADDRESS_PMB_LINE_NONE",
"cityNameFormat":"CITY_FORMAT_STANDARD", "vanityCityFormatLong":true,
"outputCountryFormat":"ENGLISH", "homeCountry":"United States",
"streetMatchingStrictness":"MATCHING_STRICTNESS_MEDIUM",
"firmMatchingStrictness":"MATCHING_STRICTNESS_MEDIUM",
"directionalMatchingStrictness":"MATCHING_STRICTNESS_MEDIUM",
"dualAddressLogic":"DUAL_NORMAL", "dpvSuccessfulStatusCondition":"A",
"reportListFileName":"", "reportlistProcessorName":"",
"reportlistNumber":1, "reportMailerAddress":"", "reportMailerName":"",
 "reportMailerCityLine":"", "canReportMailerCPCNumber":"",
"canReportMailerAddress":"", "canReportMailerName":"",
"canReportMailerCityLine":"", "internationalCityStreetSearching":100,
"addressLineSearchOnFail":true, "outputStreetAlias":true,
"outputVeriMoveBlock":false, "dpvDetermineNoStat":false,
"dpvDetermineVacancy":false, "outputAbbreviatedAlias":false,
"outputPreferredAlias":false,
"outputPreferredCity":"CITY_OVERRIDE_NAME_ZIP4",
"performSuiteLink":false, "suppressZplusPhantomCarrierR777":false,
"canStandardAddressFormat":"D", "canEnglishApartmentLabel":"APT",
"canFrenchApartmentLabel":"APP", "canFrenchFormat":"C",
"canOutputCityFormat":"D", "canOutputCityAlias":true,
"canDualAddressLogic":"D", "canPreferHouseNum":false,
"canSSLVRFLG":false, "canRuralRouteFormat":"A", "canNonCivicFormat":"A",
 "canDeliveryOfficeFormat":"I", "canEnableSERP":false,
"canSwitchManagedPostalCodeConfidence":false, "stats":null,
"counts":null, "z3seg":null, "serpStats":null, "dpvSeedList":null,
"lacsSeedList":null, "zipInputSet":null, "reportName":null,
"currentUser":null, "jobName":null, "jobId":null, "jobRequest":false,
"properties":{"DPVDetermineVacancy":"N", "DualAddressLogic":"N",
"ExtractUrb":"N", "CanFrenchFormat":"C", "AddressLineSearchOnFail":"Y",
 "OutputFieldLevelReturnCodes":"N", "OutputFormattedOnFail":"N",
"OutputStreetNameAlias":"Y", "OutputReportSERP":"N",
"OutputAddressBlocks":"Y", "ExtractFirm":"N",
"CanEnglishApartmentLabel":"APT", "OutputPreferredCity":"Z",
"FirmMatchingStrictness":"M", "CanFrenchApartmentLabel":"APP",
"KeepMultimatch":"N", "StandardAddressPMBLine":"N",
```

```
"PerformSuiteLink":"N", "CanStandardAddressFormat":"D",
"DPVSuccessfulStatusCondition":"A", "PerformLACSLink":"N",
"PerformUSProcessing":"Y", "PerformEWS":"N", "StandardAddressFormat":"C",
 "SuppressZplusPhantomCarrierR777":"N", "HomeCountry":"United States",
 "ReportMailerAddress":"", "OutputReport3553":"N",
"OutputVeriMoveDataBlock":"N", "CanDeliveryOfficeFormat":"I",
"OutputAbbreviatedAlias":"N", "PerformCanadianProcessing":"N",
"PerformDPV":"N", "PerformInternationalProcessing":"N",
"CanSSLVRFlg":"N", "StreetMatchingStrictness":"M",
"InternationalCityStreetSearching":"100",
"canSwitchManagedPostalCodeConfidence":"N", "CanDualAddressLogic":"D",
 "PerformASM":"N", "OutputCasing":"M", "ReportListFileName":"",
"CanReportMailerAddress":"", "ReportMailerCityLine":"",
"CanReportMailerCPCNumber":"", "ReportListProcessorName":"",
"CanOutputCityAlias":"Y", "DirectionalMatchingStrictness":"M",
"CanRuralRouteFormat":"A", "CanOutputCityFormat":"D",
"ReportListNumber":"1", "CanReportMailerCityLine":"",
"OutputMultinationalCharacters":"N", "EnableSERP":"N",
"CanNonCivicFormat":"A", "OutputShortCityName":"S",
"OutputPostalCodeSeparator":"Y", "FailOnCMRAMatch":"N", "PerformLOT":"N",
 "OutputCountryFormat":"E", "CanPreferHouseNum":"N",
"CanReportMailerName":"", "PerformRDI":"N", "ReportMailerName":"",
"PerformESM":"N", "OutputReportSummary":"Y",
"OutputVanityCityFormatLong":"Y", "OutputPreferredAlias":"N",
"DPVDetermineNoStat":"N", "MaximumResults":"10"}}};

-- set general configuration
set pb.bdq.uam.universaladdress.general.configuration =
{"dFileType":"SPLIT", "dMemoryModel":"MEDIUM",
"lacsLinkMemoryModel":"MEDIUM", "suiteLinkMemoryModel":"MEDIUM"};

-- set reference path
set pb.bdq.reference.data.local.location=/media/New
Volume/hduser/resources/uam/universaladdress/UAM_universaladdress4.0_Feb15;

-- set process type
set pb.bdq.uam.universaladdress.process.type=VALIDATE;

-- set header
set pb.bdq.header=InputKeyValue,FirmName,AddressLine1,AddressLine2,City,
StateProvince,PostalCode,Text;

-- Execute Query on the desired table, to display the job output on
console. This query returns a map of key value pairs containing output
 fields for each row.
SELECT tmp2.record["Confidence"], tmp2.record["AddressLine1"] FROM (
select uamvalidation(inputkeyvalue, firmname, addressline1, addressline2,
 city, stateprovince, postalcode, text) from uam_us) as addressgroup
LATERAL VIEW explode(addressgroup.mygp) tmp2 as record ;

-- Query to dump output data to a file
INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/GlobalAddressing/' row
format delimited FIELDS TERMINATED BY ',' lines terminated by '\n' STORED
```

```
 AS TEXTFILE
SELECT tmp2.record["Confidence"], tmp2.record["AddressLine1"] FROM (
select uamvalidation(inputkeyvalue, firmname, addressline1, addressline2,
 city, stateprovince, postalcode, text) from uam_us) as addressgroup
LATERAL VIEW explode(addressgroup.mygp) tmp2 as record ;

+------------+--------------------+----------+---------------+-------------+------------+
| address.recordid  |     address.addressline1     |  address.city   |
 address.stateprovince  | address.postalcode  | address.country  |
+------------+--------------------+----------+---------------+-------------+------------+
| 1                  | 18 Merivale St                | South Brisbane  |
 QLD                     | 4101                 | AUS              |
| 2                  | 19 Serpentine Rd              | Albany          |
 WA                      | 6330                 | AUS              |
| 3                  | 317 VICTORIA ST GR            | BRUNSWICK       |
 VIC                     | 3056                 | AUS              |
| 4                  | DUPLEX 6/16-18 O'CONNELL ST   | AINSLIE         |
 ACT                     | 2602                 | AUS              |
| 5                  | LOT 154 470 BRYGON CREEK DR   | UPPER COOMERA   |
 QLD                     | 4209                 | AUS              |
| 6                  | 16 GREENE ST                  | WARRAWONG       |
 ACT                     | 2502                 | AUS              |
| 7                  | UNIT 47/16 BLAIRMOUNT ST      | PARKINSON       |
 QLD                     | 4115                 | AUS              |
| 8                  | 13-15 FRANCESCO CRES          | BELLA VISTA     |
 NSW                     | 2153                 | AUS              |
| 9                  | 4 RYANS LANE                  | HEATHCOTE       |
 VIC                     | 3523                 | AUS              |
| 10                 | 1 CHRISTMAS LN                | NORTH POLE      |
 VIC                     | 1111                 | AUS              |
+------------+--------------------+----------+---------------+-------------+------------+



+----------+-------------+-------+-------------------------------+-------------+
|Confidence |StreetName     |HouseNumber |     AddressLine1
  | AddressType   |
+----------+-------------+-------+-------------------------------+-------------+
| 100.00   | MERIVALE      | 18    | 18 MERIVALE ST                | S
      |
| 99.42    | SERPENTINE    | 19       | 19 SERPENTINE RD E           |
 S       |
| 97.95    | VICTORIA      | 317      | 317 VICTORIA ST              |
 S       |
| 100.00   | O'CONNELL     | 16-18    | DUP 6 16-18 O'CONNELL ST     |
 S       |
| 0.00     | BRYGON CREEK  | 470      | LOT 154 470 BRYGON CREEK DR  |
 U       |
| 76.99    | GREENE        | 16        | 16 GREENE ST
| S       |
| 100.00   | BLAIRMOUNT    | 16       | U 47 16 BLAIRMOUNT ST        |
 S       |
| 100.00   | FRANCESCO     | 13-15    | 13-15 FRANCESCO CRES         |
```

```
 S       |
| 100.00   | RYANS          | 4         | 4 RYANS LANE                  |
 S       |
| 0.00     | CHRISTMAS      | 1         | 1 CHRISTMAS LN                |
 U       |
+----------+--------------+-----------+-----------------------------+--------------+
```

Validate Address Global

Secuencia de comandos de Hive de ejemplo

```
-- Register Universal Addressing Module [UAM-Global] BDQ Hive UDAF Jar

ADD JAR <Directory path>/uam.global.hive.${project.version}.jar;

ADD FILE <Directory path>/libAddressDoctor5.so;

-- Provide alias to UDAF class (optional). String in quotes represent
class names needed for this job to run.
CREATE TEMPORARY FUNCTION globalvalidation as
'com.pb.bdq.uam.process.hive.global.GlobalAddressingUDAF';

set hive.map.aggr = false;

-- set engine configuration
set pb.bdq.uam.global.engine.configurations=[{ "referenceData":
{"dataDir":"/media/New Volume/hduser/resources/uam/addressDoctor/5.8.0/",
 "referenceDataPathLocation":"LocaltoDataNodes"},
"databaseType":"BATCH_INTERACTIVE", "preloadingType":"NONE",
"allCountries":false, "supportedCountries":"CAN,USA,AUS"}];

-- set input configuration
set
pb.bdq.uam.global.input.configuration={"resultStateProvinceType":"COUNTRY_STANDARD",
 "processMatchingScope":"ALL", "processEnrichmentAMAS":false,
"inputForceCountryISO3":"AUS", "inputDefaultCountryISO3":"AUS",
"inputFormatDelimiter":"CRLF", "resultFormatDelimiter":"CRLF",
"resultIncludeInputs":false, "resultCountryType":"NAME_EN",
"processOptimizationLevel":"STANDARD",
"resultPreferredLanguage":"DATABASE", "processMode":"BATCH",
"resultPreferredScript":"DATABASE", "resultMaximumResults":1,
"resultCasing":"NATIVE",
"properties":{"Result.StateProvinceType":"COUNTRY_STANDARD",
"Process.MatchingScope":"ALL", "Process.EnrichmentAMAS":"false",
"Input.ForceCountryISO3":"AUS", "Input.FormatDelimiter":"CRLF",
"Result.FormatDelimiter":"CRLF", "Input.DefaultCountryISO3":"AUS",
"Result.IncludeInputs":"false", "Result.CountryType":"NAME_EN",
```

```
"Process.OptimizationLevel":"STANDARD",
"Result.PreferredLanguage":"DATABASE", "Process.Mode":"BATCH",
"Result.PreferredScript":"DATABASE", "Result.MaximumResults":"1",
"Result.Casing":"NATIVE", "Database.AddressGlobal":"Database"}};

-- set general configuration
set pb.bdq.uam.global.general.configuration={"cacheSize":"LARGE",
"maxThreadCount":8, "maxAddressObjectCount":8, "rangesToExpand":"NONE",
 "flexibleRangeExpansion":"ON", "enableTransactionLogging":false,
"maxMemoryUsageMB":1024};

-- set unlock codec
set pb.bdq.uam.global.unlockCode=<Insert your Unlock Code here>;

-- set header
set
pb.bdq.header=recordid,AddressLine1,City,StateProvince,PostalCode,Country;

-- Execute Query on the desired table, to display the job output on
console. This query returns a map of key value pairs containing output
 fields for each row.
SELECT tmp2.record["HouseNumber"], tmp2.record["Confidence"],
tmp2.record["AddressLine1"], tmp2.record["StreetName"],
tmp2.record["PostalCode"], tmp2.record["ElementInputStatus"],
tmp2.record["MailabilityScore"] FROM ( SELECT  globalvalidation(recordid,
 addressline1, city, stateprovince, postalcode, country) as mygp from
address) as addressgroup LATERAL VIEW explode(addressgroup.mygp) tmp2
as record ;


-- Query to dump output data to a file
INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/GlobalAddressing/' row
format delimited FIELDS TERMINATED BY ',' lines terminated by '\n' STORED
 AS TEXTFILE
SELECT tmp2.record["HouseNumber"], tmp2.record["Confidence"],
tmp2.record["AddressLine1"], tmp2.record["StreetName"],
tmp2.record["PostalCode"], tmp2.record["ElementInputStatus"],
tmp2.record["MailabilityScore"] FROM ( SELECT  globalvalidation(recordid,
 addressline1, city, stateprovince, postalcode, country) as mygp from
address) as addressgroup LATERAL VIEW explode(addressgroup.mygp) tmp2
as record ;

+------------+--------------------+-----------+---------------+-------------+------------+
| address.recordid  |     address.addressline1     |  address.city   |
 address.stateprovince  | address.postalcode  | address.country  |
+------------+--------------------+-----------+---------------+-------------+------------+
| 1                 | 18 Merivale St               | South Brisbane  |
 QLD                    | 4101                | AUS              |
| 2                 | 19 Serpentine Rd             | Albany          |
 WA                     | 6330                | AUS              |
| 3                 | 317 VICTORIA ST GR           | BRUNSWICK       |
 VIC                    | 3056                | AUS              |
| 4                 | DUPLEX 6/16-18 O'CONNELL ST  | AINSLIE         |
```

```
 ACT                    | 2602                 | AUS              |
| 5                   | LOT 154 470 BRYGON CREEK DR  | UPPER COOMERA    |
 QLD                    | 4209                 | AUS              |
| 6                   | 16 GREENE ST                 | WARRAWONG        |
 ACT                    | 2502                 | AUS              |
| 7                   | UNIT 47/16 BLAIRMOUNT ST     | PARKINSON        |
 QLD                    | 4115                 | AUS              |
| 8                   | 13-15 FRANCESCO CRES         | BELLA VISTA      |
 NSW                    | 2153                 | AUS              |
| 9                   | 4 RYANS LANE                 | HEATHCOTE        |
 VIC                    | 3523                 | AUS              |
| 10                  | 1 CHRISTMAS LN               | NORTH POLE       |
 VIC                    | 1111                 | AUS              |
+------------+--------------------+-----------+----------------+--------------+------------+



+----------+-------------+-------+-----------------------------+-------------+
|Confidence |StreetName     |HouseNumber |     AddressLine1
  | AddressType   |
+----------+-------------+-------+-----------------------------+-------------+
| 100.00   | MERIVALE      | 18    | 18 MERIVALE ST               | S
      |
| 99.42    | SERPENTINE    | 19       | 19 SERPENTINE RD E          |
 S       |
| 97.95    | VICTORIA      | 317      | 317 VICTORIA ST             |
 S       |
| 100.00   | O'CONNELL     | 16-18    | DUP 6 16-18 O'CONNELL ST    |
 S       |
| 0.00     | BRYGON CREEK  | 470      | LOT 154 470 BRYGON CREEK DR  |
 U       |
| 76.99    | GREENE        | 16        | 16 GREENE ST
| S       |
| 100.00   | BLAIRMOUNT    | 16       | U 47 16 BLAIRMOUNT ST       |
 S       |
| 100.00   | FRANCESCO     | 13-15    | 13-15 FRANCESCO CRES        |
 S       |
| 100.00   | RYANS         | 4        | 4 RYANS LANE                |
 S       |
| 0.00     | CHRISTMAS     | 1        | 1 CHRISTMAS LN              |
 U       |
```

```
+----------+-------------+-----------+---------------------------+-------------+
```

Validate Address Loqate

Secuencia de comandos de Hive de ejemplo

```
-- Register Universal Address Module [UAM] BDQ Hive Loqate UDAF Jar
ADD JAR <Directory path>/uam.loqate.hive.${project.version}.jar;

-- Provide alias to UDAF class (optional). String in quotes represent
class names needed for this job to run.
CREATE TEMPORARY FUNCTION loqatevalidation as
'com.pb.bdq.uam.process.hive.loqate.LoqateAddressingUDAF';

-- Adding required files to distributed cache.
ADD FILES <Directory Path>/loqate-core.car;
ADD FILES <Directory Path>/LoqateVerificationLevel.csv;
ADD FILES <Directory Path>/Loqate.csv;
ADD FILES <Directory Path>/countryTables.csv;
ADD FILES <Directory Path>/countryNameTables.csv;

set hive.map.aggr = false;

-- set process configuration
set pb.bdq.uam.loqate.process.configuration={"processType":"VALIDATE",
 "includeMatchedAddressElements":true,
"standardizedInputAddressElements":true, "returnAddressDataBlocks":true,
 "casing":"Mixed", "outputReportSummary":false,
"returnMultipleAddresses":false, "failedOnMultiMatchFound":false,
"countryFormat":"ENGLISH", "defaultCountry":"USA",
"scriptAlphabet":"Native", "returnGeocodedAddressFields":true,
"acceptanceLevel":"Level0", "minimumMatchScore":0,
"formatDataUsingAMASConventions":false,
"singleFieldDuplicateHandling":false,
"multiFieldDuplicateHandling":false,
"nonStandardFieldDuplicateHandling":false,
"outputFieldDuplicateHandling":false, "includeStandardAddress":true,
"duplicateHandling":false, "returnMultipleAddressCount":10};

-- set general configuration
set pb.bdq.uam.loqate.general.configuration={"maxIdle":null,
"minIdle":16, "maxActive":16, "maxWait":null, "whenExhaustedAction":null,
 "testOnBorrow":null, "testOnReturn":null, "testWhileIdle":null,
"timeBetweenEvictionRunsMillis":null, "numTestsPerEvictionRun":null,
"minEvictableIdleTimeMillis":null};

-- set engine configuration
```

```
set pb.bdq.uam.loqate.engine.configuration={"verbose":true,
"toolInfo":true, "outputAddressFormat":false, "logInput":false,
"logOutput":false, "logFileName":null, "matchScoreAbsoluteThreshold":60,
 "matchScoreThresholdFactor":95, "postalCodeMaxResults":10,
"strictReferenceMatch":false};

-- set reference directory path
set pb.bdq.referencedata.dir=/media/New
Volume/hduser/resources/uam/loqate/Linux;

-- set process type
set pb.bdq.uam.loqate.process.type=VALIDATE;

-- set input header
set pb.bdq.header='InputKeyValue,AddressLine1,AddressLine2,AddressLine3,
AddressLine4,City,StateProvince,PostalCode,Country,FirmName';

select SELECT tmp2.record["HouseNumber"], tmp2.record["Confidence"],
tmp2.record["AddressLine1"], tmp2.record["StreetName"],
tmp2.record["PostalCode"], tmp2.record["DPID"], tmp2.record["Barcode"]
 FROM ( SELECT  loqatevalidation(recordid, addressline1, city,
stateprovince, postalcode, country) as mygp from address) as <TABLE_NAME>
 LATERAL VIEW explode(addressgroup.mygp) tmp2 as record ;

-- Query to dump output data to a file
INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/loqate/' row format
delimited FIELDS TERMINATED BY ',' lines terminated by '\n' STORED AS
TEXTFILE SELECT * FROM ( SELECT tmp2.record["HouseNumber"],
tmp2.record["Confidence"], tmp2.record["AddressLine1"],
tmp2.record["StreetName"], tmp2.record["PostalCode"],
tmp2.record["DPID"], tmp2.record["Barcode"] FROM ( SELECT
loqatevalidation(recordid, addressline1, city, stateprovince, postalcode,
 country) as mygp from address) as <TABLE_NAME> LATERAL VIEW
explode(addressgroup.mygp) tmp2 as record ;



--Sample Input
+-------------+-------------------------------+-------------+-----------+-------------+
| inputkeyvalue  |            addressline1            | stateprovince
 | postalcode  |    country    |
+-------------+-------------------------------+-------------+-----------+-------------+
| 1              | 80 Quan Su                         |
 |             | Vietnam       |
| 2              | Final Av. Panteón Foro Libertador  |
 | 1010        | Venezuela     |
| 3              | P O Box 834                        |
 |             | St Vincent    |
| 4              | Colonia 2066                       |
 |             | Uruguay       |
| 5              | Ave de la Resistance BP127         |
 |             | Burkina Faso  |
| 6              | Buyuk Turon Street, 41             |
```

```
 |             | Uzbekistan    |
| 7             | Empire State Building             | NY
 | 10118       | US            |
| 8             | 3 Leontovycha St                  |
 |             | Ukraine       |
| 9             |                                   | Ceredigion
 |             | Wales         |
| 10            | 5 Main Street                     | Ballindalloch
 |             | Scotland      |
+-----------+-------------------------+-----------+---------+-----------+


-- Sample Output
+----------+--------------+--------+---+-----------------------------------+
|Match Score|StreetName     |HouseNumber |            addressline1
        |
+----------+--------------+-----------+-----------------------------------+
| 100.00   | MERIVALE      | 80    | 80 Quan Su
  |
| 100.00    | SERPENTINE    |        | Final Av. Panteón Foro Libertador
  |
| 0.00    | VICTORIA      | 0       | P O Box 834
   |
| 75.00   | O'CONNELL     | 2066    | Colonia 2066
   |
| 83.33     | BRYGON CREEK  | 470      | Ave de la Resistance BP127
      |
| 100.00    | GREENE        |         | Buyuk Turon Street, 41
     |
| 96.8254   | BLAIRMOUNT    | 41       | Empire State Building
      |
| 83.950   | FRANCESCO     | 350     | 3 Leontovycha St
    |
| 50.00   | RYANS         | 3        |
    |
| 100     | CHRISTMAS     | 5        | 5 Main Street
    |
+----------+--------------+-----------+-----------------------------------+

!quit
```

Funciones del módulo Universal Name

Open Name Parser

Secuencia de comandos de Hive de ejemplo

```
-- Register Universal Name Module [UNM] BDQ Hive UDF Jar
ADD JAR <Directory path>/unm.hive.${project.version}.jar;

-- Provide alias to UDF class (optional). String in quotes represent
class names needed for this job to run.
-- Open Name Parser is implemented as a UDF (User Defined function).
Hence it processes one row at a time and generates a map of key value
pairs for each row.
CREATE TEMPORARY FUNCTION opennameparser as
'com.pb.bdq.unm.process.hive.opennameparser.OpenNameParserUDF';

-- set rule
set rule='{"name":"name", "culture":"", "splitConjoinedNames":false,
"shortcutThreshold":0, "parseNaturalOrderPersonalNames":false,
"naturalOrderPersonalNamesPriority":1,
"parseReverseOrderPersonalNames":false,
"reverseOrderPersonalNamesPriority":2, "parseConjoinedNames":false,
"naturalOrderConjoinedPersonalNamesPriority":3,
"reverseOrderConjoinedPersonalNamesPriority":4,
"parseBusinessNames":false, "businessNamesPriority":5}';

-- Set Reference Directory. This must be a local path on cluster machines
 and must be present at the same path on each node of the cluster.
set refdir='/home/hadoop/reference/';

-- set header
set header='inputrecordid,Name,nametype';

-- Execute Query on the desired table, to display the job output on
console. This query returns a map of key value pairs containing output
 fields for each row.
select adTable.adid["Name"], adTable.adid["NameScore"],
adTable.adid["CultureCode"] from (select opennameparser(${hiveconf:rule},
 ${hiveconf:refdir}, ${hiveconf:header}, inputrecordid, name, nametype)
 as tmp1 from nameparser) as tmp LATERAL VIEW explode(tmp1) adTable AS
 adid;
```

```
-- Query to dump output data to a file
INSERT OVERWRITE LOCAL DIRECTORY '/home/hadoop/opennameparser/' row
format delimited FIELDS TERMINATED BY ',' lines terminated by '\n' STORED
 AS TEXTFILE
select adTable.adid["Name"], adTable.adid["NameScore"],
adTable.adid["CultureCode"] from (select opennameparser(${hiveconf:rule},
 ${hiveconf:refdir}, ${hiveconf:header}, inputrecordid, name, nametype)
 as tmp1 from nameparser) as tmp LATERAL VIEW explode(tmp1) adTable AS
 adid;



--sample input data
+-------------------------------+------------------------+----------------------+
| inputrecordid        | name           |      nametype          |
+-------------------------------+------------------------+----------------------+
| 1                                  |JOHN VAN DER LINDEN-JONES |
Simple Name    |
| 2                                  |RYAN JOHN SMITH           |  Simple
 Name      |
+-------------------------------+------------------------+----------------------+

--sample output data
+----------------------------+--------------+----------------------+
| Name           | NameScore    |      CultureCode    |
+----------------------------+--------------+----------------------+
| JOHN VAN DER LINDEN-JONES  |   75         |       True           |
| RYAN JOHN SMITH            |   100        |       True           |
+----------------------------+-----+--------------------------------+
```

Appendix

In this section

A - Excepciones

In this section

Mensajes de excepción

Excepciones - API Java

- `<Classname>.<Member>` está nulo o vacío.
- Los valores mín de `GroupbyMROption.numReduceTasks = 0` deberían ser 1.
- Los valores mín de `maxNumOfDuplicates` deberían ser 1.
- No hay archivos disponibles en la ruta especificada.
- No se pudo identificar el archivo de entrada como sospechoso o candidato.
- Campo ExpressMatchKey definido pero no disponible para el registro\t
- No se pudo obtener el FileName de InputSplit.
- No se pudo inicializar el motor.
- Error al procesar registros consolidados:

Excepciones: funciones de Hive definidas por el usuario

- _FUNC_ debe tener los argumentos mínimos.
- No se pudo inicializar el motor. Regla pasada: `<Rule used>`
- Tipo de argumento esperado: Cadena. Tipo de argumento recibido: `<Mismatched Type>`
- Excepción: configuración de `<Header string>` faltante.
- Error al procesar registros consolidados: `<Exception details>`
- Excepción: la columna del campo de clasificación `<column name>` falta en la configuración del trabajo.

B - Enums

In this section

Enumeraciones comunes

Enum `MatchingAlgorithm`

Paquete:`com.pb.bdq.api.matcher`

Clase: `Algorithm`

1. Sigla
2. CharacterFrequency
3. DaitchMokotoffSoundex
4. Fecha
5. DoubleMetaphone
6. EditDistance
7. EuclideanDistance
8. ExactMatch
9. Iniciales
10. JaroWinklerDistance
11. KeyboardDistance
12. Koeln
13. KullbackLeiblerDistance
14. Metaphone
15. SpanishMetaphone
16. Metaphone3
17. NGramDistance
18. NGramSimilarity
19. NumericString
20. Nysiis
21. Phonix
22. Soundex
23. Subcadena de caracteres
24. SyllableAlignment

Enum `Algorithm`

Paquete:`com.pb.bdq.api.matchkeygenerator`

Clase:`MatchKeyRule`

1. Soundex
2. Metaphone
3. SpanishMetaphone
4. DoubleMetaphone
5. Nysiis

6. Phonix
7. Metaphone3
8. Koeln
9. Consonante
10. Subcadena de caracteres

Enum `RecordSeparator`

Paquete:`com.pb.bdq.common.job`

Clase: `FilePath`

1. WINDOWS
2. LINUX
3. MACINTOSH

Enum `ReferenceDataPathLocation`

Paquete:`com.pb.bdq.common.job`

Constante Enum	Descripción
HDFS	Los datos de referencia se colocan en un directorio HDFS.
LocaltoDataNodes	Los datos de referencia se colocan en todos los nodos de datos del clúster.

Enum `Operation`

Paquete:`com.pb.bdq.api.consolidation`

1. CONTAINS
2. EL MÁS ALTO
3. EL MÁS BAJO
4. NOT_EQUAL
5. MAYOR
6. MENOR
7. IGUAL
8. GREATER_THAN_EQUAL_TO
9. LESS_THAN_EQUAL_TO
10. IS_EMPTY
11. IS_NOT_EMPTY
12. MOST_COMMON
13. EL MÁS LARGO
14. EL MÁS CORTO

Enum `MatchingMethod`

Paquete:`com.pb.bdq.api.matcher`

Clase: `ParentMatchRule`

1. AllTrue
2. AnyTrue
3. BasedOnThreshold

Enum `ScoringMethod`
Paquete:`com.pb.bdq.api.matcher`

Clase: `MatchRule`

1. Mínimo
2. Máximo
3. Promedio
4. WeightedAverage
5. VectorSummation

Enum `MissingDataMethod`
Paquete:`com.pb.bdq.api.matcher`

Clase: `MatchRule`

1. IgnoreBlanks
2. CountAs100
3. CountAs0
4. CompareBlanks

Enum `JoinType`
Paquete:`com.pb.bdq.api.consolidation`

Clase: `ConjoinedRule`

1. OR
2. AND

Enum `IncludeTerm`
Paquete:`com.pb.bdq.api.advtransformer`

Clase:`TableDataExtraction`

1. ExtractedData
2. NonExtractedData
3. TermNeither

Enum `Extract`
Paquete:`com.pb.bdq.api.advtransformer`

Clase:`TableDataExtraction`

1. ExtractTerm

2. ExtractNWordsLeft
3. ExtractNWordsRight

Enum `AdvTransformerExtractionType`
Paquete:`com.pb.bdq.api.advtransformer`

Clase:`AbstractAdvancedTransformerRules`

1. TableData
2. RegularExpression

Enum `MatchRuleType`
Paquete:`com.pb.bdq.api.matcher`

Clase: `MatchRule`

1. Elemento principal
2. Secundario

Enum `SortInput`
Paquete:`com.pb.bdq.api.matcher`

Clase: `MatchRule`

1. CARACTERES
2. TÉRMINOS

Enum `TableLookupAction`
Paquete:`com.pb.bdq.api.tablelookup`

Clase:`AbstractTableLookupRule`

1. Estandarizar
2. Categorizar
3. Identificar

Enumeraciones de Universal Addressing

Enum `DatabaseType`
Paquete:`com.pb.bdq.api.uam.global`

Clase:`GlobalAddressingEngineConfiguration`

1. BATCH_INTERACTIVE
2. FASTCOMPLETION
3. CERTIFICADO

Enum `PreloadingType`

Paquete:`com.pb.bdq.api.uam.global`

Clase:`GlobalAddressingEngineConfiguration`

1. NONE
2. FULL
3. PARCIAL

Enum `CountryCodes`

Paquete:`com.pb.bdq.api.uam`

Descripción: códigos alfabéticos asignados a todos los países admitidos.

Enum`StateProvinceType`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. COUNTRY_STANDARD
2. ABREVIATURA
3. EXTENDIDA

Enum `CountryType`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. ISO2
2. ISO3
3. ISO_NUMBER
4. NAME_CN
5. NAME_DA
6. NAME_DE
7. NAME_EN
8. NAME_ES
9. NAME_FI
10. NAME_FR
11. NAME_GR
12. NAME_HU
13. NAME_IT
14. NAME_JP
15. NAME_KR
16. NAME_NL
17. NAME_PL
18. NAME_PT
19. NAME_RU

20. NAME_SA
21. NAME_SE

Enum `PreferredScript`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. BASE DE DATOS
2. POSTAL_ADMIN_PREF
3. POSTAL_ADMIN_ALT
4. LATIN
5. LATIN_ALT
6. ASCII_SIMPLIFIED
7. ASCII_EXTENDED

Enum `PreferredLanguage`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. BASE DE DATOS
2. INGLÉS

Enum `Casing`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. NATIVO
2. SUPERIOR
3. INFERIOR
4. COMBINADO
5. SIN CAMBIO

Enum `OptimizationLevel`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. AJUSTAR
2. ESTÁNDAR
3. ANCHO

Enum `Mode`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. LOTE
2. CERTIFICADO
3. FASTCOMPLETION
4. INTERACTIVO
5. ANALIZAR

Enum `MatchingScope`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. TODOS
2. LOCALITY_LEVEL
3. STREET_LEVEL
4. DELIVERYPOINT_LEVEL

Enum `FormatDelimiter`

Paquete:`com.pb.bdq.api.uam.global`

Interfaz: `GlobalAddressingInputOption`

1. CRLF
2. LF
3. CR
4. PUNTO Y COMA
5. COMA
6. TABULACIÓN
7. BARRA VERTICAL
8. ESPACIO

Enum `ExhaustedAction`

Paquete:`com.pb.bdq.api.uam.loqate`

Clase:`LoqateAddressingGeneralConfiguration`

1. GROW
2. BLOCK
3. Con errores

Enum `AcceptanceLevel`

Paquete:`com.pb.bdq.api.uam.loqate.validate`

Clase:`LoqateAddressingValidateConfiguration`

1. Level0
2. Level1
3. Level2
4. Level3

5. Level4
6. Level5

Enum `OutputCasing`
Paquete:`com.pb.bdq.api.uam.loqate.validate`

Clase:`LoqateAddressingValidateConfiguration`

1. Combinado
2. Superior

Enum `CountryFormat`
Paquete:`com.pb.bdq.api.uam.loqate.validate`

Clase:`LoqateAddressingValidateConfiguration`

1. INGLÉS
2. ISO
3. UPU

Enum `ScriptAlphabet`
Paquete:`com.pb.bdq.api.uam.loqate.validate`

Clase:`LoqateAddressingValidateConfiguration`

1. InputScript
2. Nativo
3. Latin_English

Enum `CacheSize`
Paquete:`com.pb.bdq.api.uam.global`

Clase:`GlobalAddressingGeneralConfiguration`

1. NONE
2. PEQUEÑO
3. LARGE

Enum `RangesToExpand`
Paquete:`com.pb.bdq.api.uam.global`

Clase:`GlobalAddressingGeneralConfiguration`

1. NONE
2. ONLY_WITH_VALID_ITEMS

Enum `FlexibleRangeExpansion`
Paquete:`com.pb.bdq.api.uam.global`

Clase:`GlobalAddressingGeneralConfiguration`

1. Activado
2. Desactivado

Enum `CasingType`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. COMBINADO
2. SUPERIOR

Enum `CityNameFormat`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. CITY_FORMAT_LONG
2. CITY_FORMAT_SHORT
3. CITY_FORMAT_STANDARD

Enum `OutputCountryFormat`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. INGLÉS
2. Francés
3. Alemán
4. Español
5. ISO
6. UPU

Enum `DualAddressLogic`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. DUAL_NORMAL
2. DUAL_PO_BOX
3. DUAL_STREET

Enum `StandardAddressFormat`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. STANDARD_ADDRESS_FORMAT_COMBINED_UNIT
2. STANDARD_ADDRESS_FORMAT_SEPARATE_UNIT
3. STANDARD_ADDRESS_FORMAT_SEPARATE_DUAL

Enum `StreetMatchingStrictness`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. MATCHING_STRICTNESS_EQUAL
2. MATCHING_STRICTNESS_TIGHT
3. MATCHING_STRICTNESS_MEDIUM
4. MATCHING_STRICTNESS_LOOSE

Enum `FirmMatchingStrictness`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. MATCHING_STRICTNESS_EQUAL
2. MATCHING_STRICTNESS_TIGHT
3. MATCHING_STRICTNESS_MEDIUM
4. MATCHING_STRICTNESS_LOOSE

Enum `DirectionalMatchingStrictness`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. MATCHING_STRICTNESS_EQUAL
2. MATCHING_STRICTNESS_TIGHT
3. MATCHING_STRICTNESS_MEDIUM
4. MATCHING_STRICTNESS_LOOSE

Enum `StandardAddressPMBLine`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. STANDARD_ADDRESS_PMB_LINE_NONE
2. STANDARD_ADDRESS_PMB_LINE_1
3. STANDARD_ADDRESS_PMB_LINE_2

Enum `PreferredCity`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. CITY_OVERRIDE_NAME_ZIP4
2. CITY_USPS_STATE_FILE
3. CITY_PRIMARY_NAME

Enum `DPVFileType`
Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressGeneralConfiguration`

1. Dividir
2. FULL
3. Plano

Enum `DPVMemoryModel`

Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressGeneralConfiguration`

1. PICO
2. MICRO
3. PEQUEÑO
4. Medio
5. LARGE
6. Enorme

Enum `LacsLinkMemoryModel`

Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressGeneralConfiguration`

1. PICO
2. MICRO
3. PEQUEÑO
4. Medio
5. LARGE
6. Enorme

Enum `SuiteLinkMemoryModel`

Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressGeneralConfiguration`

1. PICO
2. MICRO
3. PEQUEÑO
4. Medio
5. LARGE
6. Enorme

Enum `DPVSuccessStatusCondition`

Paquete:`com.pb.bdq.api.universaladdress`

Clase:`UniversalAddressInputConfiguration`

1. DPV_CONDITON_FULL

2. DPV_CONDITON_PARTIAL
3. DPV_CONDITON_ALWAYS

Enum `UAMCASSReportType`

Paquete:`com.pb.bdq.uam.common`

1. CASS_3553
2. CASS_DETAIL
3. CASS_DETAIL2
4. CASS_DETAIL3

C - Códigos de país ISO y compatibilidad de módulos

In this section

Códigos de país ISO y compatibilidad de módulos

La tabla enumera los códigos ISO de dos y tres dígitos para cada país.

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Afganistán	AF	AFG
Islas Aland	AX	ALA
Albania	AL	ALB
Argelia	DZ	DZA
Samoa Americana	AS	ASM
Andorra	AD	AND
Angola	AO	AGO
Anguila	AI	AIA
Antártida	AQ	ATA
Antigua y Barbuda	AG	ATG
Argentina	AR	ARG
Armenia	AM	ARM

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Aruba	AW	ABW
Australia	AU	AUS
Austria	AT	AUT
Azerbaiyán	AZ	AZE
Bahamas	BS	BHS
Bahréin	BH	BHR
Bangladesh	BD	BGD
Barbados	BB	BRB
Bielorrusia	BY	BLR
Bélgica	BE	BEL
Belice	BZ	BLZ
Benín	BJ	BEN
Bermudas	BM	BMU
Bután	BT	BTN

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Bolivia, Estado Plurinacional de	BO	BOL
Bonaire, Saba y San Eustaquio	BQ	BES
Bosnia-Herzegovina	BA	BIH
Botsuana	BW	BWA
Isla Bouvet	BV	BVT
Brasil	BR	BRA
Territorio Oceánico Indio-Británico	IO	IOT
Brunéi	BN	BRN
Bulgaria	BG	BGR
Burkina Faso	BF	BFA
Burundi	BI	BDI
Camboya	KH	KHM
Camerún	CM	CMR
Canadá	CA	CAN

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Cabo Verde	CV	CPV
Islas Caimán	KY	CYM
República Centroafricana	CF	CAF
Chad	TD	TCD
Chile	CL	CHL
China	CN	CHN
Isla de Navidad	CX	CXR
Islas Cocos (Keeling)	CC	CCK
Colombia	CO	COL
Comoras	KM	WITH
Congo	CG	COG
Congo, República Democrática del	CD	COD
Islas Cook	CK	COK
Costa Rica	CR	CRI

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Costa de Marfil	CI	CIV
Croacia	HR	HRV
Cuba	CU	CUB
Curaçao	CW	CUW
Chipre	CY	CYP
República Checa	CZ	CZE
Dinamarca	DK	DNK
Yibuti	DJ	DJI
Dominica	DM	DMA
República Dominicana	DO	DOM
Ecuador	EC	ECU
Egipto	EG	EGY
El Salvador	SV	SLV
Guinea Ecuatorial	GQ	GNQ

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Eritrea	ER	ERI
Estonia	EE	EST
Etiopía	ET	ETH
Islas Malvinas (Falkland)	FK	FLK
Islas Feroe	FO	FRO
Islas Fiji	FJ	FJI
Finlandia	FI	FIN
Francia	FR	FRA
Guayana Francesa	GF	GUF
Polinesia Francesa	PF	PYF
Territorios Australes Franceses	TF	ATF
Gabón	GA	GAB
Gambia	GM	GMB
Georgia	GE	GEO

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Alemania	OF	DEU
Ghana	GH	GHA
Gibraltar	GI	GIB
Grecia	GR	GRC
Groenlandia	GL	GRL
Granada	GD	GRD
Guadalupe	GP	GLP
Guam	GU	GUM
Guatemala	GT	GTM
Guernsey	GG	GGY
Guinea	GN	GIN
Guinea-Bissau	GW	GNB
Guyana	GY	GUY
Haití	HT	HTI

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Islas Heard y McDonald	HM	HMD
Santa Sede (Estado de la Ciudad del Vaticano)	VA	VAT
Honduras	HN	HND
Hong Kong	HK	HKG
Hungría	HU	HUN
Islandia	IS	ISL
India	IN	IND
Indonesia	ID	IDN
República Islámica de Irán	IR	IRN
Irak	IQ	IRQ
Irlanda	IE	IRL
Isla de Man	IM	IMN
Israel	IL	ISR
Italia	IT	ITA

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Jamaica	JM	JAM
Japón	JP	JPN
Jersey	JE	JEY
Jordania	JO	JOR
Kazajistán	KZ	KAZ
Kenia	KE	KEN
Kiribati	KI	KIR
Corea, República Popular Democrática de	KP	PRK
Corea, República de	KR	KOR
Kosovo	KS	KOS
Kuwait	KW	KWT
Kirguistán	KG	KGZ
República Democrática Popular Lao	LA	LAO
Letonia	LV	LVA

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Líbano	LB	LBN
Lesoto	LS	LSO
Liberia	LR	LBR
Libia	LY	LBY
Liechtenstein	LI	LIE
Lituania	LT	LTU
Luxemburgo	LU	LUX
Macao	MO	MAC
Macedonia, Antigua República Yugoslava de	MK	MKD
Madagascar	MG	MDG
Malaui	MW	MWI
Malasia	MY	MYS
Maldivas	MV	MDV
Malí	ML	MLI

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Malta	MT	MLT
Islas Marshall	MH	MHL
Martinica	MQ	MTQ
Mauritania	MR	MRT
Mauricio	MU	MUS
Mayotte	YT	MYT
México	MX	MEX
Micronesia, Estados Federados de	FM	FSM
Moldavia, República de	MD	MDA
Mónaco	MC	MCO
Mongolia	MN	MNG
Montenegro	ME	MNE
Montserrat	MS	MSR
Marruecos	MA	MAR

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Mozambique	MZ	MOZ
Myanmar	MM	MMR
Namibia	NA	NAM
Nauru	NR	NRU
Nepal	NP	NPL
Países Bajos	NL	NLD
Nueva Caledonia	NC	NCL
Nueva Zelanda	NZ	NZL
Nicaragua	NI	NIC
Níger	NE	NER
Nigeria	NG	NGA
Niue	NU	NIU
Isla Norfolk	NF	NFK
Islas Marianas del Norte	MP	MNP

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Noruega	NO	NOR
Omán	OM	OMN
Pakistán	PK	PAK
Palaos	PW	PLW
Territorio Palestino, Ocupado	PS	PSE
Panamá	PA	OFFSET
Papúa Nueva Guinea	PG	PNG
Paraguay	PY	PRY
Perú	PE	PER
Las Filipinas	PH	PHL
Pitcairn	PN	PCN
Polonia	PL	POL
Portugal	PT	PRT
Puerto Rico	PR	PRI

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Qatar	QA	QAT
Reunión	RE	REU
Rumania	RO	ROU
Federación Rusa	RU	RUS
Ruanda	RW	RWA
Saint Barthélemy	BL	BLM
Santa Elena, Ascensión y Tristán de Acuña	SH	SHE
San Cristóbal y Nieves	KN	KNA
Santa Lucía	LC	LCA
Saint Martin (parte francesa)	MF	MAF
San Pedro y Miquelón	PM	SPM
San Vicente y las Granadinas	VC	VCT
Samoa	WS	WSM
San Marino	SM	SMR

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Santo Tomé y Príncipe	ST	STP
Arabia Saudita	SA	SAU
Senegal	SN	SEN
Serbia	RS	SRB
Seychelles	SC	SYC
Sierra Leona	SL	SLE
República de Singapur	SG	SGP
Sint Maarten (parte holandesa)	SX	SXM
Eslovaquia	SK	SVK
Eslovenia	SI	SVN
Islas Salomón	SB	SLB
Somalia	SO	SOM
Sudáfrica	ZA	ZAF
Islas Georgias del Sur y Sandwich del Sur	GS	SGS

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Sudán del Sur	SS	SSD
España	ES	ESP
Sri Lanka	LK	LKA
Sudán	SD	SDN
Surinam	SR	SUR
Svalbard y Jan Mayen	SJ	SJM
Suazilandia	SZ	SWZ
Suecia	IF	SWE
Suiza	CH	CHE
República Árabe Siria	SY	SYR
Taiwán, Provincia de China	TW	TWN
Tayikistán	TJ	TJK
Tanzania	TZ	TZA
Tailandia	TH	THA

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Timor Oriental	TL	TLS
Togo	TG	TGO
Tokelau	TK	TKL
Tonga	TO	TON
Trinidad y Tobago	TT	TTO
Túnez	TN	TUN
Turquía	TR	TUR
Turkmenistán	TM	TKM
Islas Turcas y Caicos	TC	TCA
Tuvalu	TV	TUV
Uganda	UG	UGA
Ucrania	UA	UKR
Emiratos Árabes Unidos	AE	ARE
Reino Unido	GB	GBR

Nombre de país ISO	ISO 3116-1 Alpha-2	ISO 3116-1 Alpha-3
Estados Unidos	Estados Unidos	Estados Unidos
Islas Ultramarinas Menores de los Estados Unidos	UM	UMI
Uruguay	UY	URY
Uzbekistán	UZ	UZB
Vanuatu	VU	VUT
Venezuela, República Bolivariana de	VE	VEN
Vietnam	VN	VNM
Islas Vírgenes Británicas	VG	VGB
Islas Vírgenes de los Estados Unidos	VI	VIR
Wallis y Futuna	WF	WLF
Sahara Occidental	EH	ESH
Yemen	YE	YEM
Zambia	ZM	ZMB
Zimbabwe	ZW	ZWE

Notices

Avisos de USPS®

Pitney Bowes Inc. posee una licencia no exclusiva para publicar y vender bases de datos ZIP + 4® en medios magnéticos y ópticos. Las siguientes marcas comerciales son propiedad del Servicio Postal de los Estados Unidos: CASS, CASS Certified, DPV, eLOT, FASTforward, First-Class Mail, Intelligent Mail, LACS^{Link}, NCOA^{Link}, PAVE, PLANET Code, Postal Service, POSTNET, Post Office, RDI, Suite^{Link} , United States Postal Service, Standard Mail, United States Post Office, USPS, ZIP Code, y ZIP + 4. Esta lista no es exhaustiva de todas las marcas comerciales que pertenecen al servicio postal.

Pitney Bowes Inc. es titular de una licencia no exclusiva de USPS® para el procesamiento NCOA^{Link}®.

Los precios de los productos, las opciones y los servicios del software de Pitney Bowes no los establece, controla ni aprueba USPS® o el gobierno de Estados Unidos. Al utilizar los datos RDI™ para determinar los costos del envío de paquetes, la decisión comercial sobre qué empresa de entrega de paquetes se va a usar, no la toma USPS® ni el gobierno de Estados Unidos.

Proveedor de datos y avisos relacionados

Los productos de datos que se incluyen en este medio y que se usan en las aplicaciones del software de Pitney Bowes Software, están protegidas mediante distintas marcas comerciales, además de un o más de los siguientes derechos de autor:

© Derechos de autor, Servicio Postal de los Estados Unidos. Todos los derechos reservados.

© 2014 TomTom. Todos los derechos reservados. TomTom y el logotipo de TomTom son marcas comerciales registradas de TomTom N.V.

© 2016 HERE

Fuente: INEGI (Instituto Nacional de Estadística y Geografía)

Basado en los datos electrónicos de © National Land Survey Sweden.

© Derechos de autor Oficina del Censo de los Estados Unidos

© Derechos de autor Nova Marketing Group, Inc.

Algunas partes de este programa tienen © Derechos de autor 1993-2007 de Nova Marketing Group Inc. Todos los derechos reservados

© Copyright Second Decimal, LLC

© Derechos de autor Servicio de correo de Canadá

Este CD-ROM contiene datos de una compilación cuyos derechos de autor son propiedad del servicio de correo de Canadá.

© 2007 Claritas, Inc.

El conjunto de datos Geocode Address World contiene datos con licencia de GeoNames Project (**www.geonames.org**) suministrados en virtud de la licencia de atribución de Creative Commons (la “Licencia de atribución”) que se encuentra en **http://creativecommons.org/licenses/by/3.0/legalcode**. El uso de los datos de GeoNames (según se describe en el manual de usuario de Spectrum™ Technology Platform) se rige por los términos de la Licencia de atribución. Todo conflicto entre el acuerdo establecido con Pitney Bowes Software, Inc. y la Licencia de atribución se resolverá a favor de la Licencia de atribución exclusivamente en cuanto a lo relacionado con el uso de los datos de GeoNames.

3001 Summer Street
Stamford CT 06926-0700
USA

www.pitneybowes.com