**AYI ACADEMY: TALEND**

**Material práctico**



**Módulo 5: Procesamiento de datos**

Módulo 5: Procesamiento de datos

* **Temas:**
* Configuración del entorno
* Mapear datos usando tMap
* Unir datos usando tMap
* Capturar Join Rejects (rechazos de unión)
* Filtrar datos y capturar Filter Rejects (rechazos de filtro)
* Usar otros componentes para el procesamiento de datos
* Resumen

## Objetivos

Después de completar este módulo, podrá:

* Mapear datos de entrada hacia una salida usando el componente tMap
* Unir datos y recopilar rechazos de unión
* Filtrar datos y recopilar filtros rechazados
* Usar expresiones y variables en el editor de mapeo
* Ordenar y agregar datos usando otros componentes de procesamiento de datos

## Ejercicio

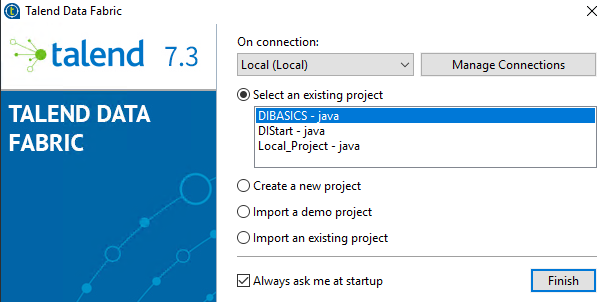
Este módulo cubre varios métodos de procesamiento de datos. La primera parte es una introducción a la transformación de datos utilizando el componente tMap (mapear datos, unir datos y recopilar rechazos de unión, filtrar datos y recopilar rechazos de filtro). La segunda parte se centra en otras tareas de procesamiento, como la agregación, la clasificación y el muestreo.

**Tema 1: Configuración del entorno**

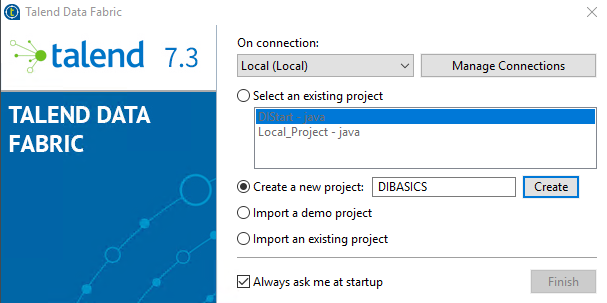
## Antes de empezar

Antes de desarrollar proyectos en Talend Studio, inicie Studio y abra un proyecto en su espacio de trabajo.

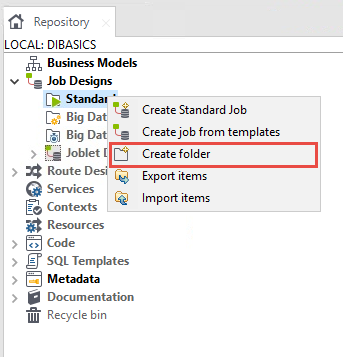
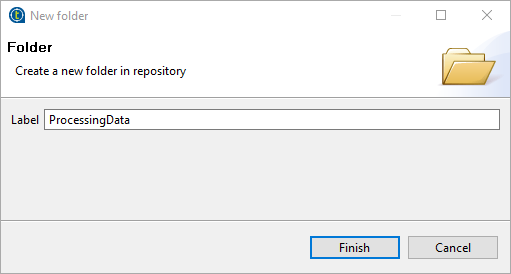
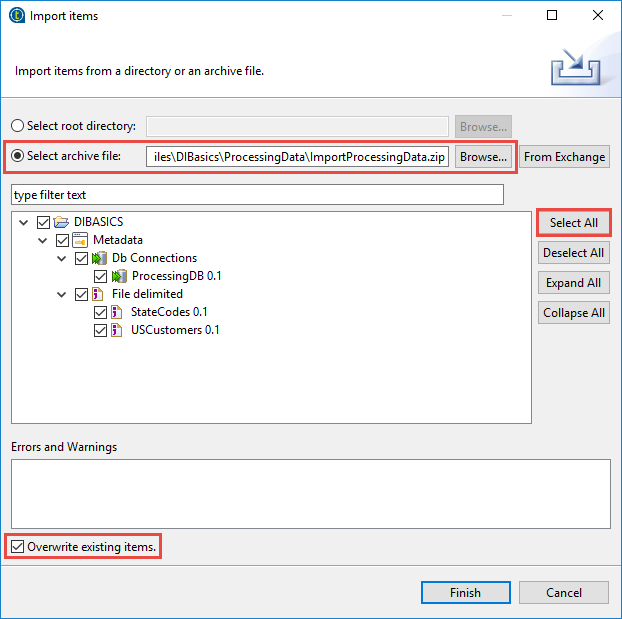
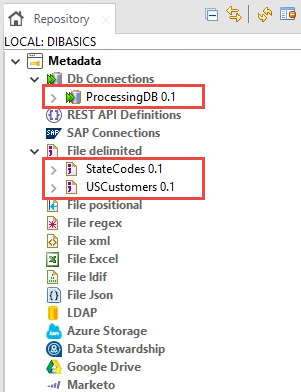
## Abrir un proyecto

**1.-** Haga doble click en el ícono **Studio** para ejecutar Talend Studio.  
  
Se muestra la pantalla de presentación de Talend Data Fabric, que le permite abrir un proyecto existente o crear uno nuevo.  
  
**2.-** Abra el proyecto DIBASICS.  
  
Confirme que el proyecto **DIBASICS** está disponible en la lista de proyectos existentes.  
  
 → Si aparece el proyecto **DIBASICS**, haga clic en él y luego haga clic en **Finalizar**.

**→** Si no encuentra el proyecto, siga estos pasos para crearlo:  
  
 **a.-** Para **On connection**, seleccione **Local (Local)**, lo que significa que cuando se crea el proyecto, se almacena en su máquina local.

**b.-** Haga click en **Create a new project** e ingrese *DIBASICS*.  
  
  
 **c.-** Haga click en **Create** y espere a que el proyecto aparezca en la lista. El proyecto que creaste estará seleccionado por defecto.  
  
 **d.-** Haga click en **Finish**.  
  
💡 Si se le solicita que se conecte a Talend Community, haga clic en **Skip this Step**.  
💡 Si se abre la página **Welcome**, haga click en **Start now!**

## Configurar el proyecto

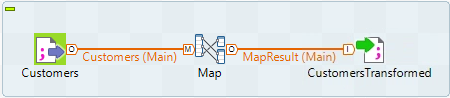
**1.-** En la barra **Quick Access**, asegúrese que la perspectiva **Integration** esté seleccionada.  
  
  
**2.-** Cree una carpeta para los Jobs.  
  
En **Repository** expanda **Job Designs**, haga click derecho en **Standard**, luego seleccione **Create folder**.  
  
En Label ingrese *ProcessingData* y presione **Finish**.  
  
La nueva carpeta aparecerá en Repository > Job Designs > Standard.  
  
💡 Almacenar trabajos en una carpeta en particular puede ayudarlo a mantenerse organizado. Para este ejercicio, la carpeta ProcessingData es la ubicación predeterminada donde se crean y almacenan sus Jobs.  
  
**3.-** Actualice un archivo.  
  
El archivo ImportProcessingData.zip se encuentra en C:\StudentFiles\DIBasics\ProcessingData.  
  
 **a.-** En la barra de herramientas **Quick Access**, haga click en el ícono **folder and up arrow**.  
  
  
 **b.-** En la ventana **Import items**, seleccione **Select archive file**, luego haga click en el botón **Browse**.  
  
 **c.-** Navegue hasta **C:\StudentFiles\DIBasics\ProcessingData** y abra el archivo **ImportProcessingData.zip**.  
  
 **d.-** Presione el botón **Select All**.  
  
 **e.-** Asegúrese de marcar **Overwrite existing items**, luego presione **Finish**  
Ha importado metadata para una conexión de base de datos y para archivos delimitados al repositorio.  


**Tema 2: Mapear datos usando tMap**

## Antes de comenzar

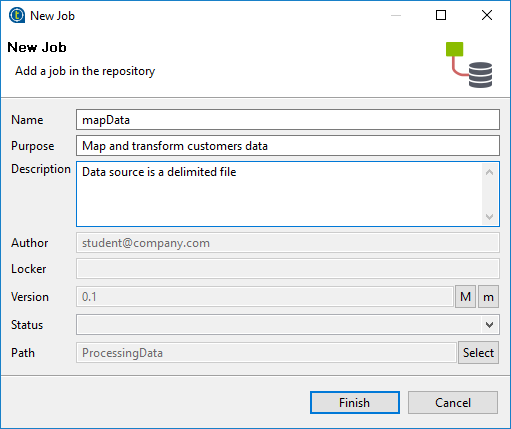
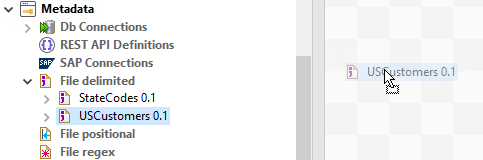
Este ejercicio le enseña cómo realizar una transformación básica de datos utilizando el componente tMap. Se enfoca en mapear entrada a salida usando tMap y crear expresiones usando funciones y operadores predefinidos. El objetivo es crear un Job que lea los datos del cliente de un archivo, concatene el nombre y el apellido en una sola columna, aplique una expresión para convertir la columna de estado en mayúsculas y escriba el resultado en un nuevo archivo.

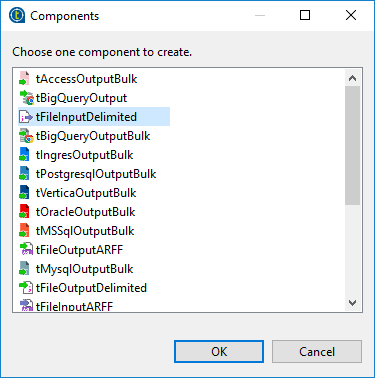
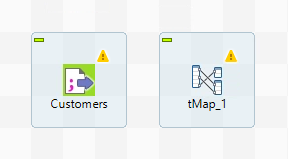
El Job deberá verse así:

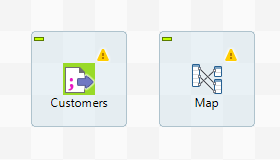
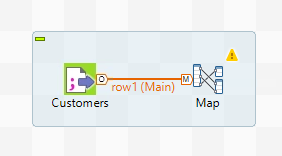


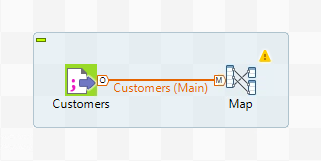
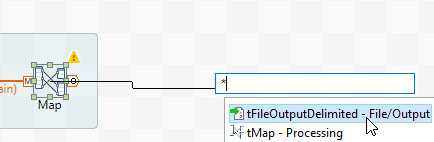
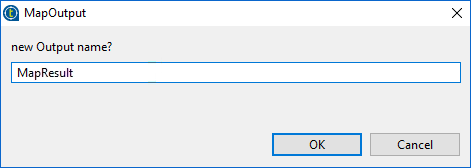
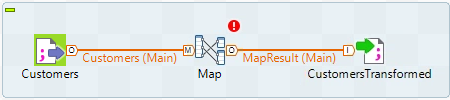
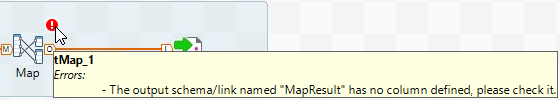
## Crear un Job que transforme los datos de clientes

En un nuevo Job, crear un archivo delimitado usando la metadata disponible en el repositorio. Agregue un componente **tMap** al Job para transformar datos y escribir los resultados en un archivo delimitado.

**1.-** Cree un Job en **Repository** > **Job Designs** > **Standard** > **ProcessingData** y nómbrelo *mapData*.  
  
  
**2.-** Coloque un componente **tFileInputDelimited** en el **Designer** arrastrando el archivo delimitado **USCustomers** desde la metadata del **Repository**.  
  
 **a.-** En el **Repository**, arrastre **USCustomers** al **Designer**.  


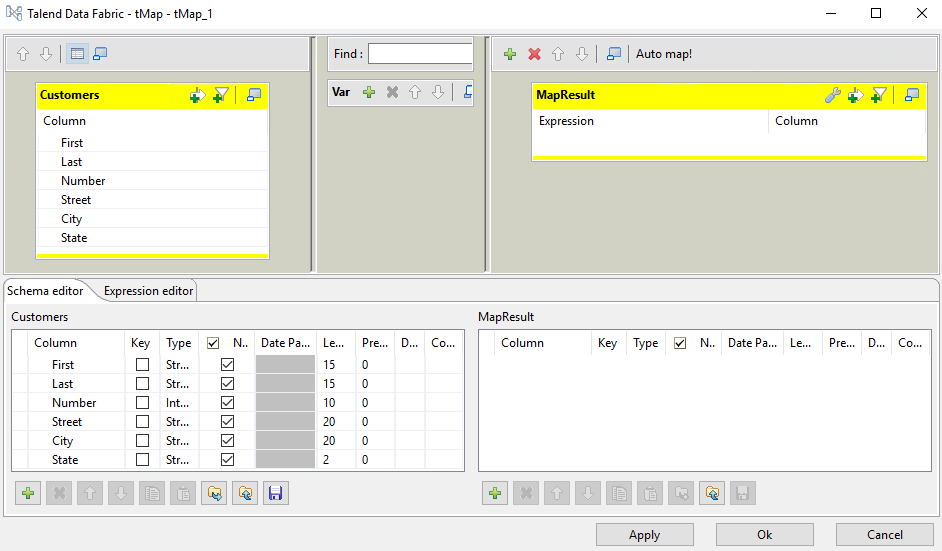
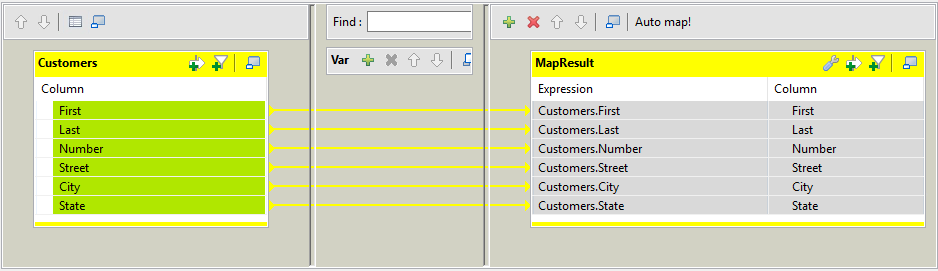
**b.-** Seleccione el componente **tFileInputDelimited**.  
  
  
 **c.-** Nombre el componente **tFileInputDelimited**: *Customers*.  
  
  
**3.-** Agregue un componente tMap.  
  
 **a.-** En **Designer** a la derecha de **Customers** agregue un componente **tMap**.  


**b.-** Nombre el componente **tMap**: *Map*.  
  
  
 **c.-** Haga click derecho sobre el componente **Customers**, y use **Main** row para conectarlo con el componente **Map**.  
  
Observe la advertencia junto al componente Map. La funcionalidad básica de tMap es asignar entrada a salida, por lo que debe estar vinculado a un componente de salida.

**4.-** Siguiendo las mejores prácticas, cambie el nombre de la conexión.  
  
Haga click en **row1** para editar la conexión, y nómbrela *Customers*.  
  
  
**5.-** Agregue un componente tFileOutputDelimited.  
  
 **a.-** Haga click derecho en **Map**, mantenga presionado el botón derecho del mouse y mueva el cursor fuera del componente.  
En la lista, haga doble clic en tFileOutputDelimited para agregarlo.  
  
 **b.-** Para nombrar la salida del componente Map, en **new Output name?**, ingrese MapResult (sin espacios) y haga clic en **OK**.  
  
  
La nueva conexión de salida es una Main row, al igual que la conexión de entrada.  
  
 **c.-** Cambie el nombre del componente tFileOutputDelimited.  
  
En **Designer**, haga clic para editar la label FileOutputDelimited\_1 e ingrese *CustomersTransformed*.  
  
  
**6.-** Investigue el error.  
  
Pase el puntero sobre el icono rojo o vea el error. No se define ninguna columna en el esquema de salida.  


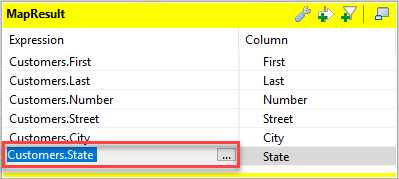
## Mapear rows

La transformación realizada en los datos de entrada se define en el editor de mapeo para el componente tMap.

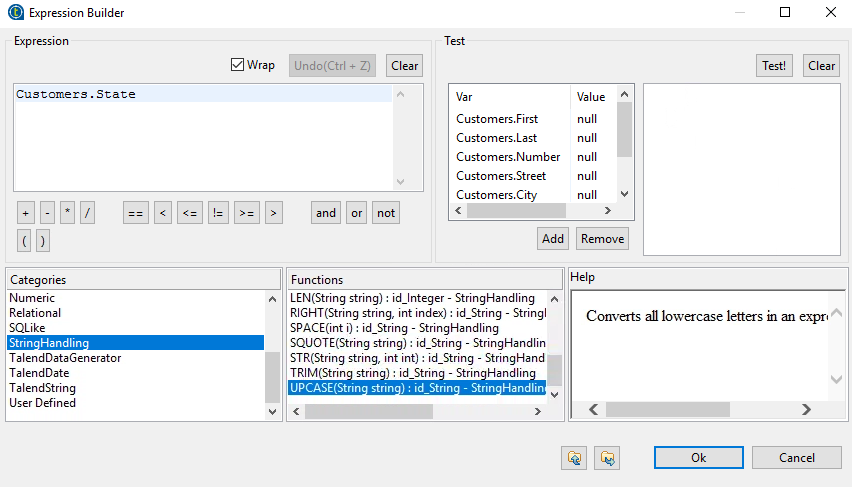
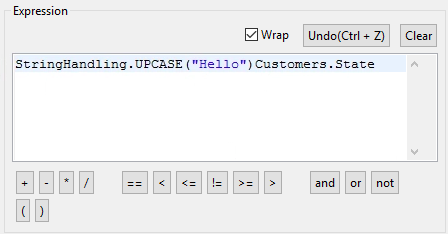
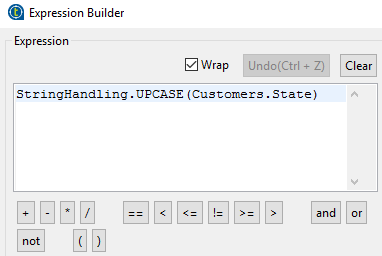
**1.-** Double-click **Map** to open the mapping editor.  
  
En la parte inferior del editor de mapeo, observe la tabla de la izquierda, **Customers**, y la de la derecha, **MapResult**. Estas tablas representan los esquemas de entrada y salida, con nombres que coinciden con las conexiones de entrada y salida. El esquema para Customers se copia del componente de entrada. El esquema para la nueva conexión, MapResult, no se ha definido  
  
Su objetivo con este Job es pasar todos los datos de Customers a MapResult mientras capitaliza los valores en la columna State.  
  
**2.-** Mapear todas las columnas de entrada.  
  
En Customers, haga clic en la fila superior, mantenga presionada la tecla shift y haga clic en la fila inferior para seleccionar todas las filas. Arrástrelas desde Customers a MapResult.  
Observe las flechas que indican la asignación de columnas de un esquema a otro. En la tabla MapResult, observe los nombres de las columnas y el contenido de los datos definidos en Expression fields. En este punto, las expresiones siguen el patrón table.name, lo que indica que los nombres de los datos y las columnas coinciden con los de Customers.

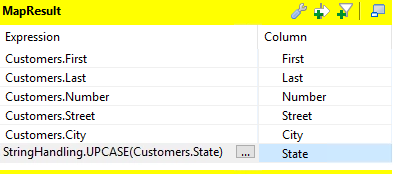
## Construyendo una expresión

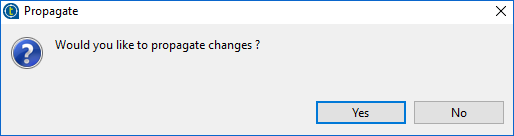
En este ejercicio, creará una expresión para convertir los nombres de State a mayúsculas.

**1.-** Seleccione la salida que desea modificar.  
  
Desea convertir los datos de la columna State a mayúsculas, por lo que en la tabla de salida de MapResult, haga clic en Customers.State y en el botón [...] que aparecerá.  
Se abre el Expression Builder. Esta ventana le permite construir expresiones en sintaxis Java. Puede ingresar expresiones manualmente o aprovechar la herramienta que simplifica la creación de tales expresiones.

**2.-** Cree una expresión para convertir datos a mayúsculas.  
  
En la categoría **StringHandling**, agregue la función **UPCASE** a su expresión.

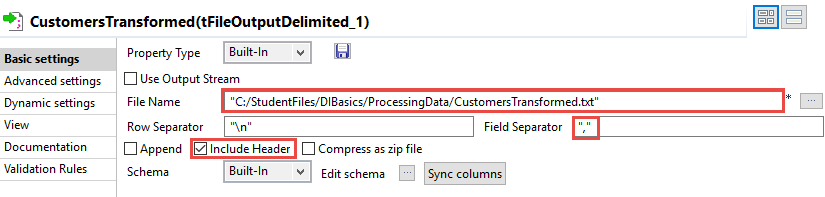
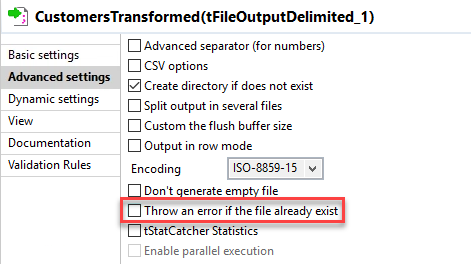
**a.-** Seleccione **Categories** > **StringHandling**, luego busque entre la lista **Functions** y seleccione **UPCASE**. En el área **Help**, verá una breve descripción de la función.  
  
  
  
 **b.-** Haga doble clic en **UPCASE** para insertar la función en la expresión.  
Observe el formato de la expresión insertada. De forma predeterminada, la función contiene la cadena literal "Hello". Debe reconfigurar esto para convertir el contenido de la columna State.  
  
 **c.-** Corrija la expresión moviendo Customers.State dentro de los paréntesis para que reemplace "Hello".  
  
  
💡 Esta expresión opera en una variable en lugar de una cadena literal, por lo que en este caso, asegúrese de eliminar las comillas que la encierran.  
  
 **d.-** Presione **Ok**.

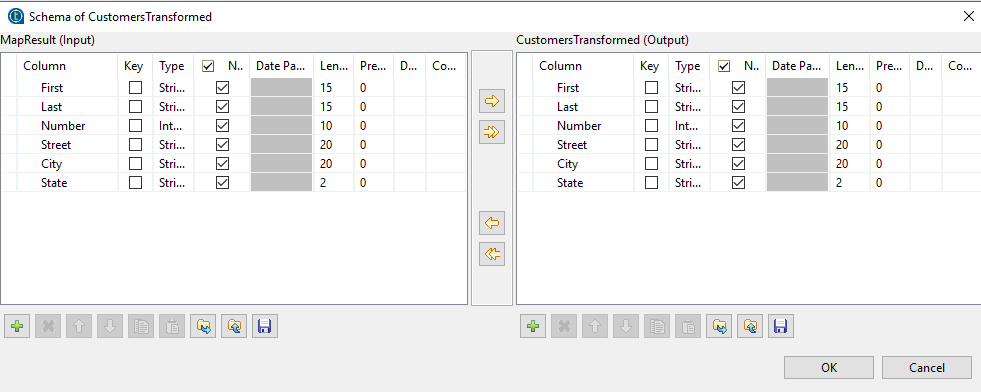
**3.-** Aplique los cambios.  
  
La nueva expresión aparece en la tabla MapResult para la columna de salida State.  
  
  
Ha terminado con la configuración del mapeo. Debido a que ha realizado cambios en el esquema de la conexión de salida, se le solicita que propague los cambios (refleje esos cambios en todos los componentes que utilizan el esquema).  
  
 **a.-** Para cerrar el editor de mapeo, presione **Ok**.

**b.-** Para propagar los cambios, presione **Yes**.  


## Configurar la salida

En este ejercicio, configurará el componente de salida, que escribe los resultados de la transformación en un archivo delimitado.

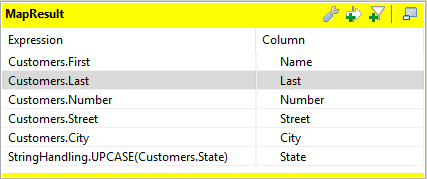
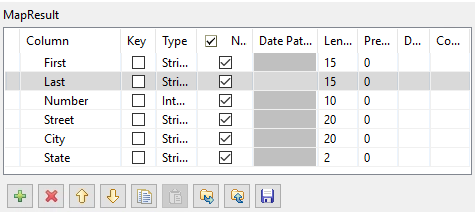
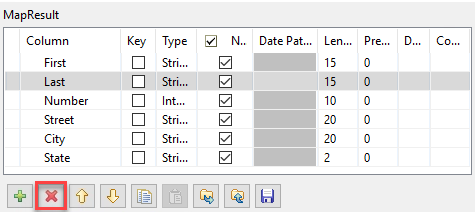
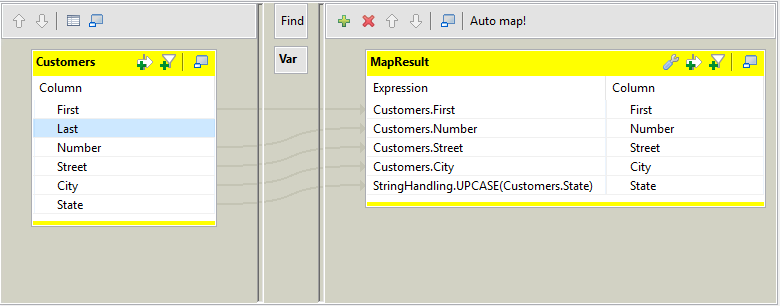
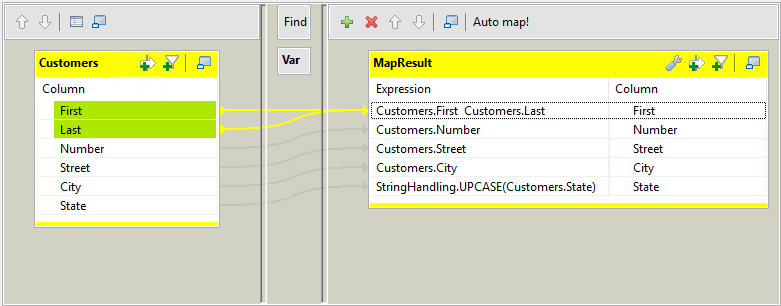
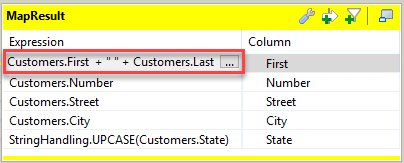
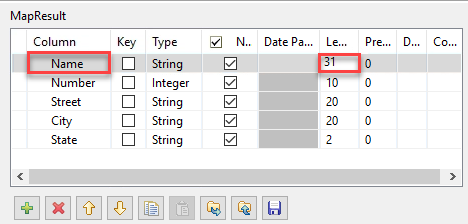
**1.-** Nombre el archivo de salida CustomersTransformed.txt y defina las propiedades básicas del archivo delimitado  
  
 **a.-** Haga doble clic en CustomersTransformed  
  
 **b.-** Cambie el nombre del archivo de salida a *C:/StudentFiles/DIBasics/ProcessingData/CustomersTransformed.txt*💡Puede usar el botón [...] para navegar al directorio antes de ingresar el nuevo nombre de archivo.  
  
 **c.-** Cambie el **Field Separator** a una coma y seleccione la casilla de verificación **Include Header** para incluir los nombres de las columnas como la primera fila del archivo de salida.  
  
  
  
 **d.-** En **Advanced settings**, desmarque **Throw an error if the file already exist**.  
Esta opción es útil para sobrescribir el archivo existente cuando se vuelve a ejecutar el Job

**2.-** Verifique el schema.  
  
En **Basic settings**, click **Edit schema** [**...**].  
  
Observe que el schema de CustomersTransformed coincida con el schema especificado en la tabla MapResult.  
  
No es necesario realizar cambios, así que haga clic en **OK**.

## Combinar columnas

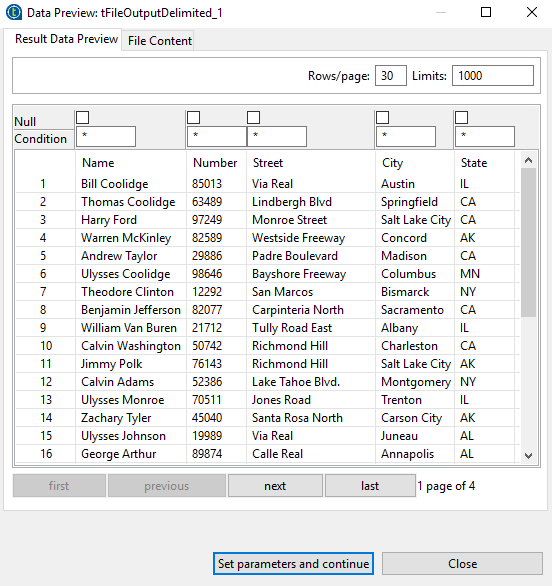
El componente tMap puede realizar una variedad de funciones de mapeo y transformación de datos. Imagine que su almacén de datos de destino requiere que el nombre y el apellido se combinen en una sola columna. Siga estos pasos para lograrlo.

**1.-** Actualice el schema.

En el editor de mapeo, elimine la columna Customers.Last.  
  
 **a.-** En **Designer**, Haga doble click en el componente **Map**.  
  
 **b.-** En la tabla MapResult, haga clic en Customers.Last.  
  
En la parte inferior de la ventana del **Schema editor**, observe que se selecciona la misma columna.  
  
Vas a combinar los datos de dos columnas en una sola columna, por lo que esta columna ya no es necesaria.  
  
 **c.-** Elimine la columa.  
  
Click **Remove selected items** [**X**] para borrar la columna **Last** del schema de salida.  
  
💡 En el schema editor de MapResult (en la parte inferior derecha del editor de mapeo), asegúrese de hacer clic en Remove selected items y no en Remove table en la parte superior derecha del editor de mapeo. Si hace clic en Eliminar tabla, elimina su mapeo y deberá reconstruirlo.  
  
La columna también desaparece de la tabla MapResult en la parte superior de la ventana. Observe que ya no hay un mapeo para la última columna en la tabla Customers.  
  
  
**2.-** Combine las columnas First y Last para formar una sola columna que contenga el nombre del cliente.  
  
Asigne los nombres y apellidos a una sola columna.  
  
Arrastre **Last** desde la tabla **Customers** hacia **First** en la tabla **MapResult**.  
  
Observe que la expresión ahora contiene referencias a dos columnas de la tabla **Customers**.  
  
**3.-** Corrija la expresión.  
  
Inserte un + entre las dos referencias. Esto es necesario para concatenar dos String juntos. Además, inserte un carácter de espacio para separar los dos nombres en la salida. Cuando haya terminado, la expresión debería quedar así:  
  
*Customers.First + " " + Customers.Last  
***4.-** Actualice el nombre y la longitud de la columna.  
  
En **Schema editor** en la parte inferior de la ventana, en la tabla MapResult, haga clic en First y cámbielo a Name. También actualice la longitud de ese campo de 15 a 31.  
  
  
  
**5.-** Click **Ok** para guardar los cambios y **Yes** para propagarlos.

## Ejecutar el Job

**1.-** Abra la pestaña **Run** y presione **Run**. El Job debe finalizar en **[exit code=0]**.

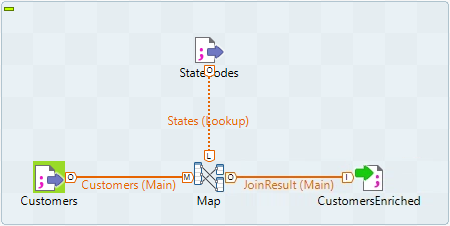
**2.-** Revisa la salida.  
  
Abra el visor de datos del componente CustomersTransformed.  
  
Observe que la columna Name incluye tanto el nombre como el apellido, y que los datos de la columna State están en mayúsculas.

**Tema 3: Unir datos usando tMap**

## Antes de comenzar

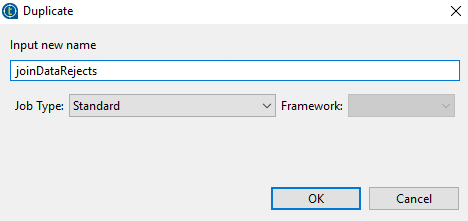
En este ejercicio, aprenderá cómo unir datos de múltiples entradas utilizando tMap. Un archivo que contiene códigos y nombres de estado se usa como una segunda entrada (lookup) para el componente de mapeo. Los datos de las dos entradas se unen para incluir el nombre del estado en la salida.

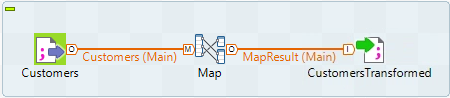
Al finalizar el ejercicio, el Job debería verse así:



## Duplicar el Job

Cree un nuevo Job duplicando el Job **mapData**.

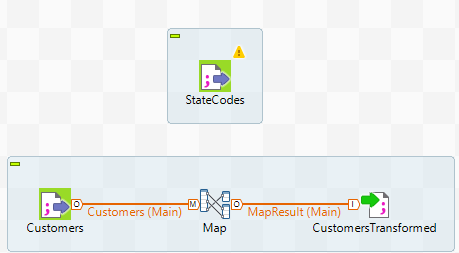
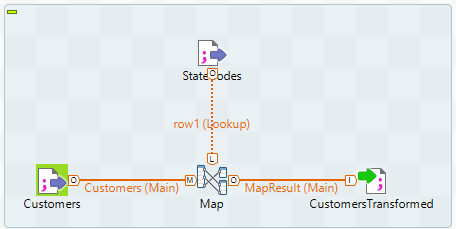
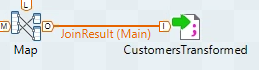
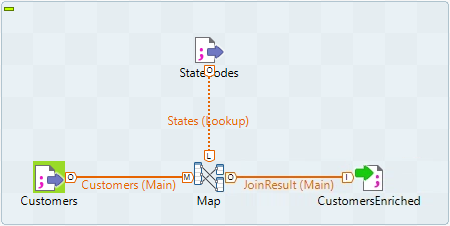
**1.-** Duplique mapData para crear un Job llamado *joinDataRejects*.  


**2.-** En **Repository**, abra el Job **joinDataRejects**.  


## Agregar un lookup

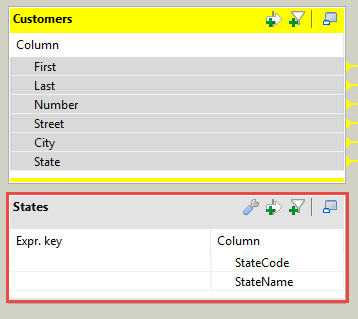
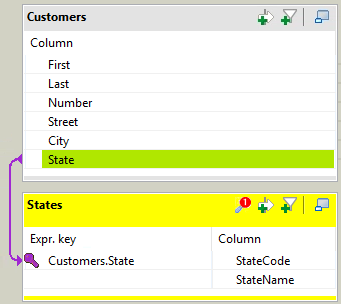
Los metadatos de StateCodes se pueden usar para leer datos del archivo States.txt que contiene la lista de códigos de estado de EE. UU..

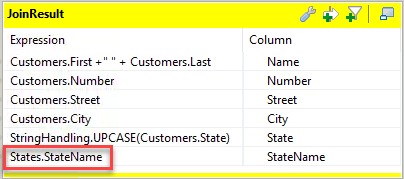
Usando estos metadatos, agregue un archivo delimitado al Job y conéctelo al componente Map. Debido a que tMap acepta solo una conexión principal, esta segunda conexión es una búsqueda.

**1.-** Crear un archivo delimitado usando metadata.  
  
Con los metadatos de StateCodes, agregue un archivo delimitado al trabajo y conéctelo al componente map.  
  
 **a.-** Coloque la metadata **StateCodes** en el **Designer**, sobre el componente **Map**.  
  
 **b.-** En el cuadro de diálogo Components, seleccione tFileInputDelimited y haga clic en OK.  
  
  
  
**2.-** Conecte el nuevo componente.  
  
Para crear una fila, haga clic con el botón derecho en StateCodes, seleccione **Row** > **Main**, y presione el componente **Map.**  
  
Observe que la fila es **lookup**, no **Main**. Puede proporcionar varias fuentes de entrada para un componente tMap, pero solo una puede ser la fila Principal; el resto son lookups.  
  
**3.-** Siguiendo las mejores prácticas, debe cambiar el nombre de las conexiones antes de abrir tMap.  
  
 **a.-** Click en **row1** y renómbrelo *States*.  
  
 **b.-** Click en **MapResult** y renómbrelo *JoinResult*.  
  
  
El Job debería verse así:  


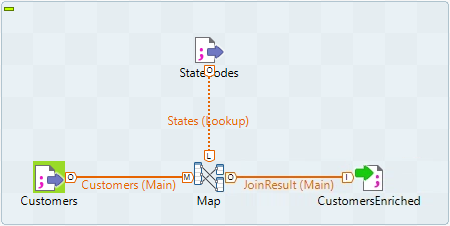
## Configurar el Join (unión)

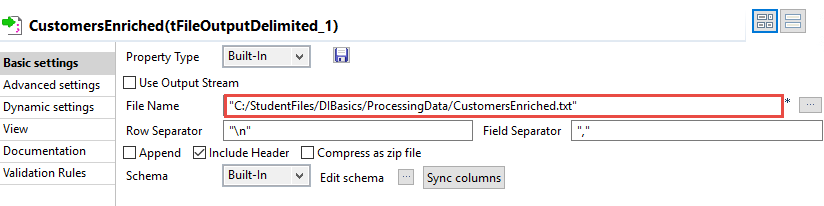
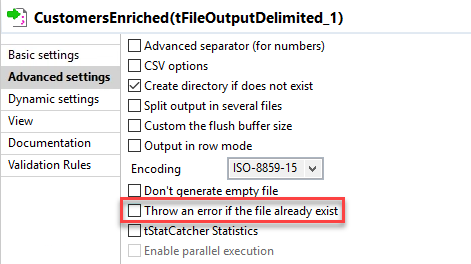
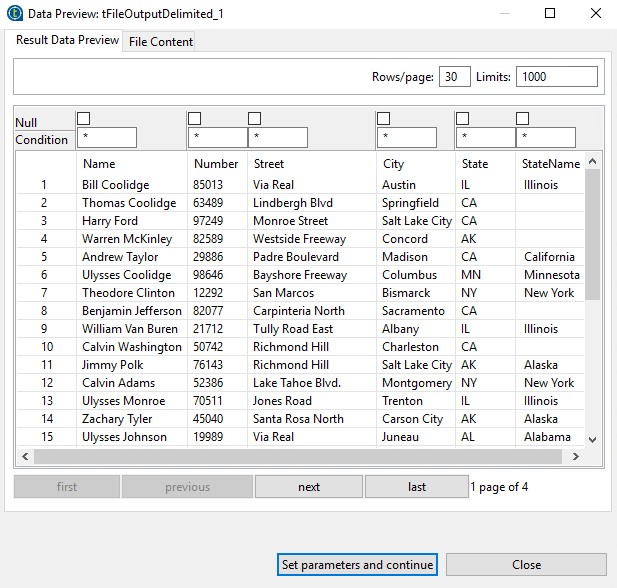
En el editor de mapeo, defina la unión entre los flujos de datos de Customers y States.

**1.-** Abra el editor de mapeo.  
  
  
En el editor de mapeo, observe el flujo de datos nuevo, States, a la izquierda, correspondiente a la fila de entrada que conecta StateCodes con el componente Map.  
  
**2.-** Cree una unión en state code.  
  
Arrastre **State** de la tabla **Customers** al campo **Expr.key** para **StateCode** en la tabla States.  
  
Esto crea una unión, por lo que el valor de la columna State en Customers se compara con el valor de la columna StateCode en States.  
  
💡 El tipo de combinación predeterminado es left outer join. En el siguiente ejercicio, utilizará un inner join.

**3.-** Mapear la tabla de salida.  
  
Arrastre StateName de la tabla de States a la fila inferior vacía de la tabla JoinResult.  
  
Esto ha enriquecido el flujo de datos de Customers con datos de la búsqueda.  
  
**4.-** Guarde los cambios  
  
Click **Ok**, luego **Yes** para propagar los cambios.

## Actualizar el Job

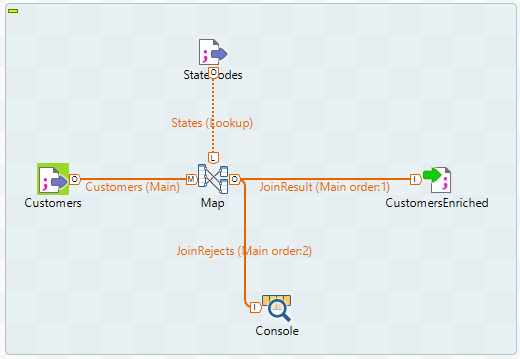
**1.-** Renombre el componente **CustomersTransformed** a *CustomersEnriched*.  
  
  
**2.-** Configure el componente CustomersEnriched.  
  
Asigne un nombre al archivo de salida y configure el componente para que no arroje un mensaje de error si el archivo ya existe.

**a.-** Cambie el nombre del archivo de salida a *C:/StudentFiles/ProcessingData/CustomersEnriched.txt*.  
  
  
 **b.-** En **Advanced settings**, desmarque **Throw an error if the file already exist**.  
  
  
**3.-** Ejecute el Job , luego revise el archivo de salida. Click derecho sobre **CustomersEnriched** y seleccione **Data viewer**.  
  
Observe que no todas las filas incluyen un valor en la columna StateName. Esto significa que el Job no está haciendo exactamente lo que pretendía, por lo que debe investigar.

**Tema 4: Capturar Join Rejects (rechazos de unión)**

## Antes de comenzar

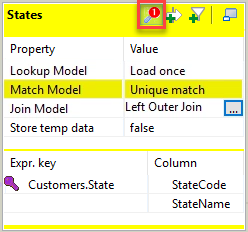
Su lookup no está funcionando como esperaba. En este ejercicio, configurará tMap para examinar las filas en las que falló el lookup. Al final, el Job se ve así:

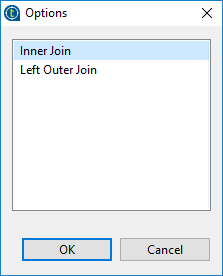


## Configurar el join model en tMap

Para investigar por qué algunas filas de búsqueda no coinciden, utilice inner join model.

**1.-** Mire la configuración de la tabla.  
  
Abra la configuración de **tMap** para ver el join model actual.  
  
 **a.-** Haga doble click en **Map**.  
  
 **b.-** En la tabla **States**, click en **tMap settings**.

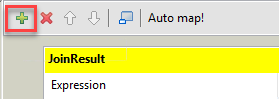
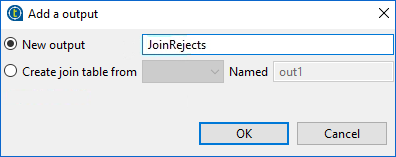
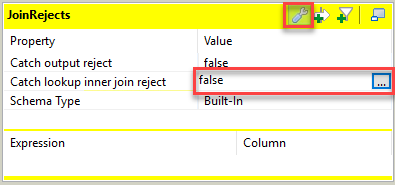
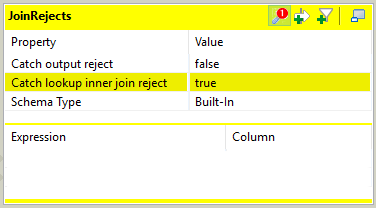
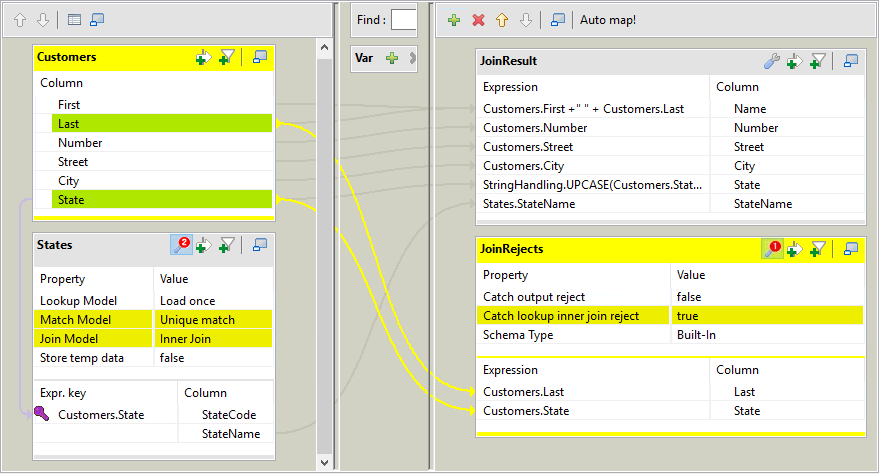
  
Aquí es donde puede configurar parámetros de tabla adicionales. Observe que Join Model **Join Model** usa **Left Outer Join**. Un left outer join incluye todas las filas de la tabla principal, incluso si no hay una fila en la tabla lookup con un valor coincidente en la columna de unión, lo que explica por qué algunas filas no incluyen el nombre del estado. Usar inner join puede ayudarlo a descubrir por qué algunas filas de búsqueda no coinciden.

**2.-** Configure el inner join.  
  
Cambie el join model a **Inner Join**.  
  
 **a.-** En la columna **Value**, click **Left Outer Join**.  
  
 **b.-** Click [**...**] para cambiar el valor.  
  
 **c.-** En la ventana **Options**, click **Inner Join**.  
  
  
 **d.-** Click **OK**.

Al cambiar join por inner join, las filas que no coinciden se excluyen de la salida y puede capturar los rechazos para solucionar problemas.

## Agregar una tabla rejects

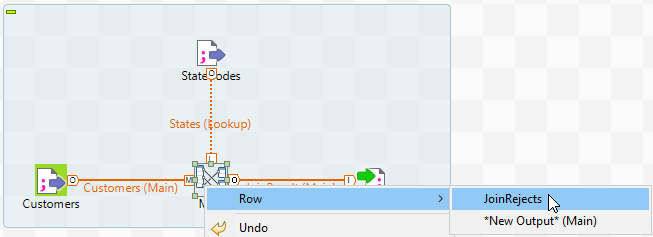
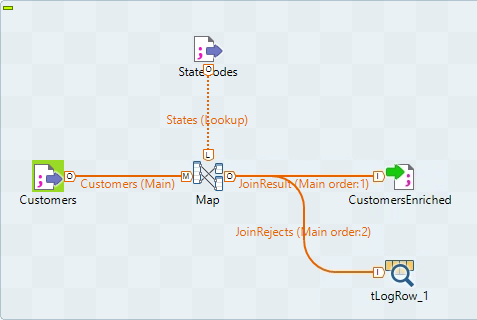
Para capturar los rechazos, agregue una nueva tabla de salida.

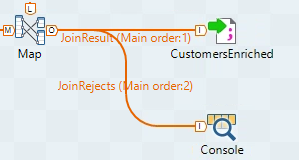
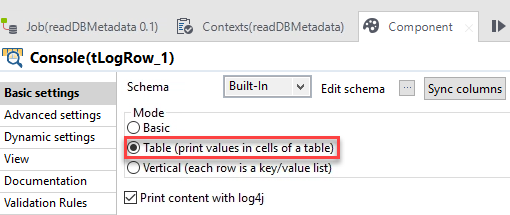
**1.-** Cree un nuevo flujo de datos de salida para capturar filas que fallan en la combinación interna para que pueda determinar el problema.  
  
 **a.-** En el editor de mapeo, haga click en el ícono **Add output table** [**+**].  
  
  
 **b.-** Ingrese *JoinRejects*, click **OK**.  
  
La nueva salida aparecerá debajo de las otras, con el nombre JoinResult.  
  
**2.-** Cambie la configuración de **JoinRejects** para capturar inner join rejects.  
  
 **a.-** En la tabla **JoinRejects**, click en el botón **tMap settings** [**wrench**].  
  
 **b.-** A la derecha de **Catch lookup inner join reject**, click **false**.  
  
 **c.-** Click [**...**] para cambiar el valor.  
  
  
 **d.-** En **Options**, click **true**.  
  
  
 **e.-** Click **OK**.  
  
**3.-** Defina las columnas para la nueva tabla de salida.  
  
Arrastre **Last** y **State** desde **Customers** hacia **JoinRejects**.  
  
Ahora la salida **JoinRejects** va a contener el nombre y el estado para los casos de inner join rejects.

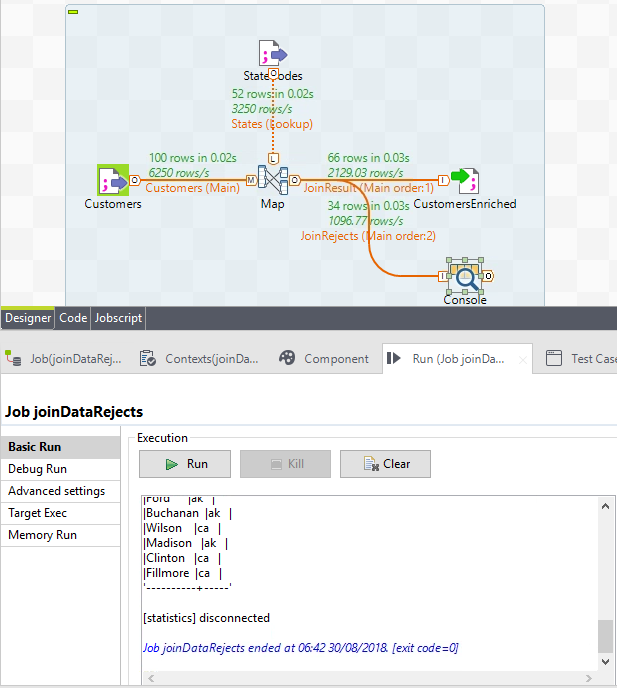
**4.-** Click **Ok**.

## Loguear rejects

Usando la conexión **JoinRejects**, envíe el log de los casos rechazados al componente **tLogRow**.  
  
**1.-** Agregue un componente de salida para loguear los rejects.  
  
 **a.-** Agregue un componente **tLogRow** justo debajo del componente **CustomersEnriched**.  
  
 **b.-** Conecte la nueva salida con **tLogRow**.

Observe que la fila JoinRejects corresponde a la tabla de salida que agregó al componente Map:  


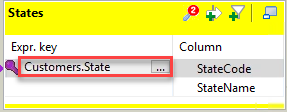
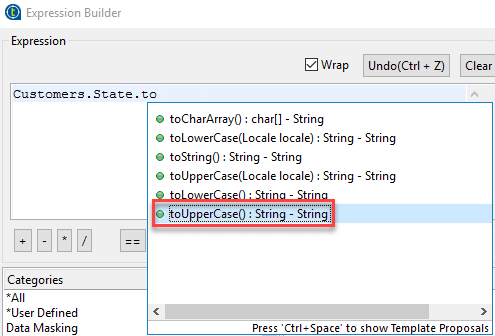
**2.-** Renombre el componente tLogRow a *Console*.  
  
  
**3.-** Configure el componente **Console** para imprimir los resultados en pantalla.  
  
 **a.-** En la pestaña **Component**, click **Basic settings**.  
  
 **b.-** Para **Mode**, seleccione **Table**.

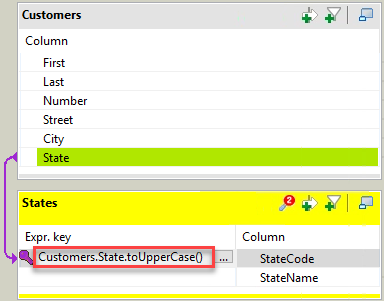
**4.-** Ejecute el Job y examine los resultados.  
  
En el output, verá:  
  
 → El resumen en Designer le dice que se procesaron 100 filas, de las cuales 34 fueron rechazadas.  
  
 → Los códigos de estado en las filas rechazadas están todos en minúsculas. No coinciden con los códigos en mayúsculas del archivo lookup.

Ahora que sabe que el lookup está fallando debido a los string en minúscula, puede corregir la configuración. Debe convertir los códigos de estado a mayúsculas antes de compararlos con las entradas en la tabla lookup.

## Configurar el componente Map

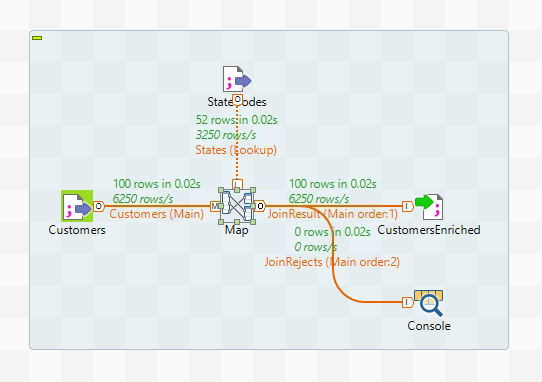
Use expresiones para convertir State a mayúsculas.

**1.-** En la tabla **States**, cree una expresión para convertir State a mayúsculas. Deberás usar una función diferente en este caso.  
  
 **a.-** Doble click en **Map**.  
  
 **b.-** En la tabla **States**, click **Customers.State** y [**...**] para abrir el **Expression Builder**.  
  
  
 **c.-** Después del último carácter en Customers.State, comience a escribir .to para obtener una lista de opciones.  
  
 **d.-** Doble click en **toUppercase()** para seleccionarlo.  


**e.-** Click **Ok**.  
  
El valor de la columna State se convertirá a mayúsculas antes de compararse con la tabla lookup.  
  
**2.-** Click **Ok**.

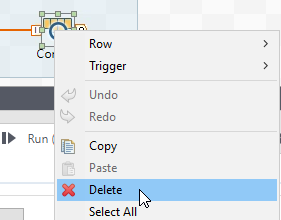
## Inspeccionar los resultados

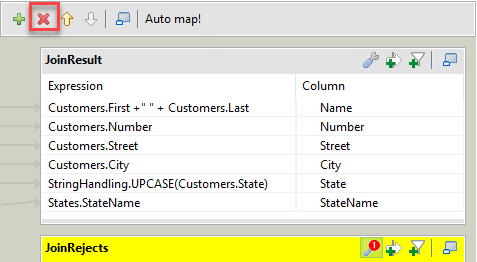
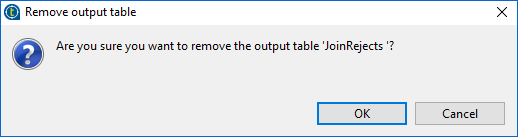
Ejecute el Job, luego vea los registros y datos rechazados en el archivo de salida.

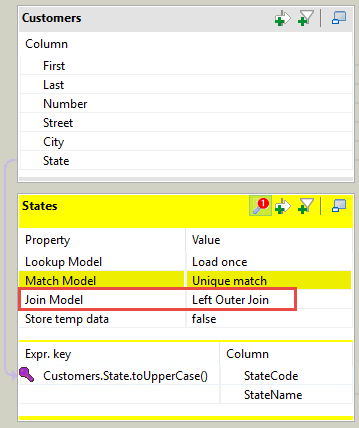
**1.-** Ejecute el Job. Observe que el componente tLogRow no procesó ninguna fila, por lo que no se encontraron rechazos. Todas las filas de entrada van al archivo de salida.  
  
  
**2.-** Use data viewer para comprobar que no haya pasado ningún campo sin State.

## Limpiar el Job

Ya no es necesario registrar los rechazos, por lo que puede limpiar el Job eliminando el componente Consola y la salida de JoinRejects.

**1.-** Remueva los componentes innecesarios.  
  
Click derecho en el componente **Console** y presione **Delete**.  


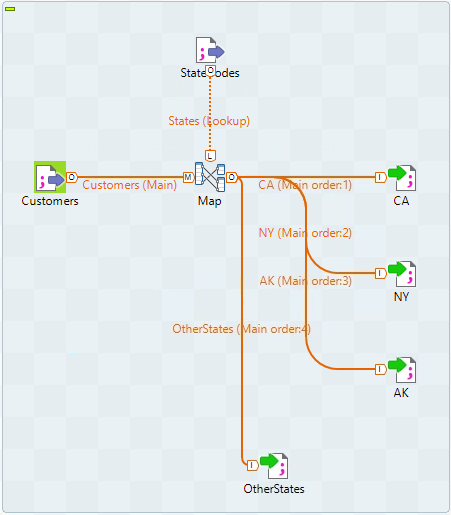
**2.-** Remueva la tabla innecesaria.  
  
En el editor de mapeo, borre la tabla **JoinRejects**.  
  
 **a.-** Doble click en **Map**.  
  
 **b.-** Click en la tabla **JoinRejects** y presione el ícono **Remove selected output table**.  
  
  
 **c.-** Haga click en **OK** cuando le pregunte si está seguro de borrar la tabla **JoinRejects**.  


**d.-** En la configuración de **States**, para **Join Model**, seleccione **Left Outer Join**.  
  
  
 **e.-** Click **Ok**.

**Tema 5: Filtrar datos y capturar Filter Rejects**

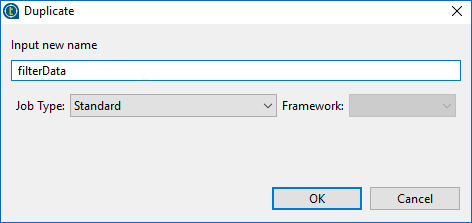
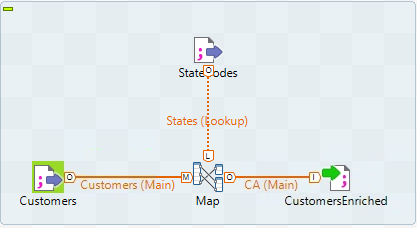
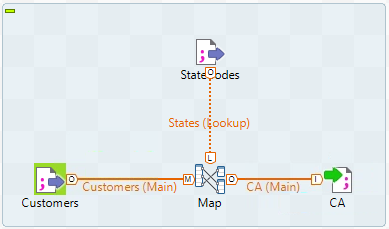
## Antes de comenzar

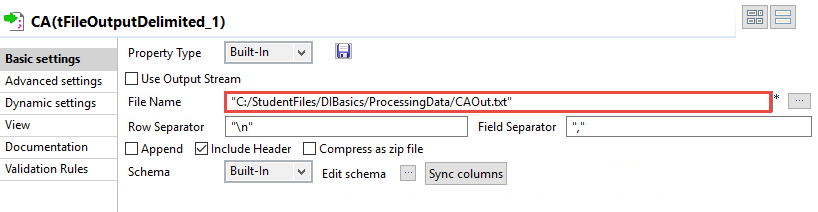
Supongamos que desea que su Job filtre la salida según el código de estado para que los archivos de salida resultantes estén restringidos a estados específicos. Al final, el job se ve así:



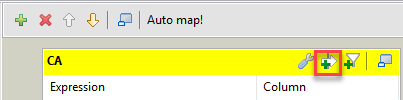
## Duplicar el Job

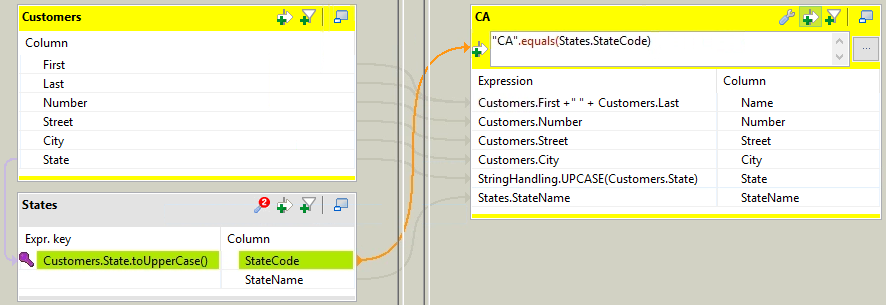
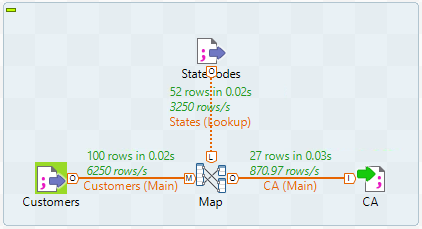
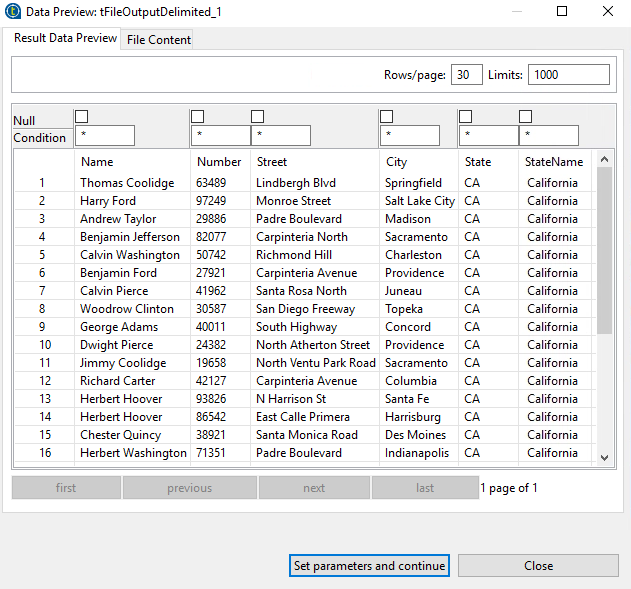
Cree un nuevo Job duplicando **joinDataRejects**.

**1.-** Duplique **joinDataRejects** y nómbrelo *filterData*.  
  
Haga doble click en el Job para abrirlo en el **Designer**.  
  
**2.-** Renombre **tFileOutputDelimited** y su connection row para reflejar el nuevo propósito del Job.  
  
 **a.-** Click en la row **JoinResult** para editar la conexión y renombrarla *CA*.  
  
  
 **b.-** Cambie el nombre de **tFileOutputDelimited** a *CA* (para los códigos de estado de California).  


**3.-** Configure el componente **CA**.  
  
En la pestaña **Component**, cambie el nombre del archivo de salida por *C:/StudentFiles/DIBasics/ProcessingData/CAOut.txt*.  


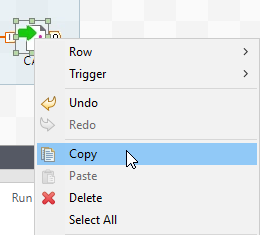
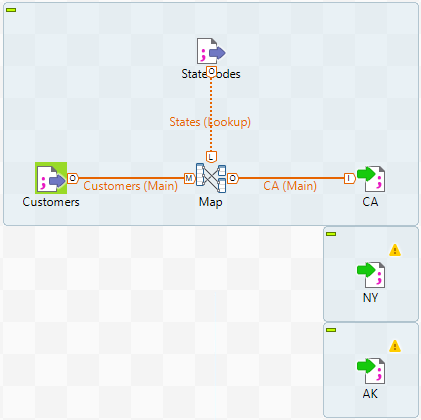
## Agregar un filtro

Create a filter to extract customers from California.  
  
**1.-** Agregue una expresión filtro.  
  
En la tabla **CA**, cree un filtro para seleccionar sólo los códigos de estado **"CA"  
  
 a.-** Abra el editor de mapeo.  
  
💡 Note que la salida se ha renombrado **CA**, al igual que en el **Designer**.  
  
 **b.-** En la esquina superior derecha de la salida **CA**, click **Enable/disable expression filter**.  


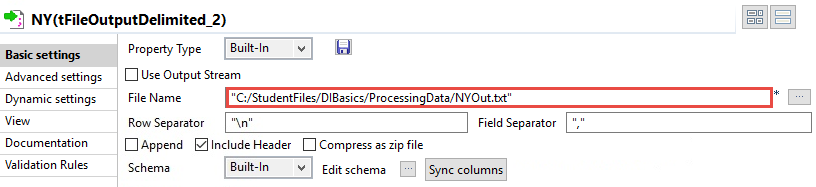
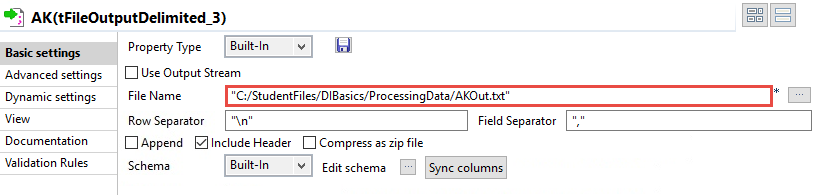
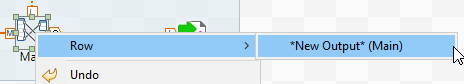
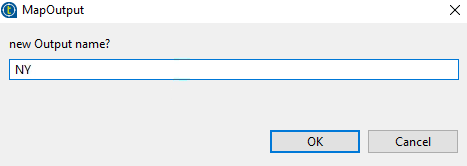
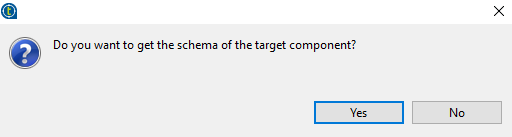
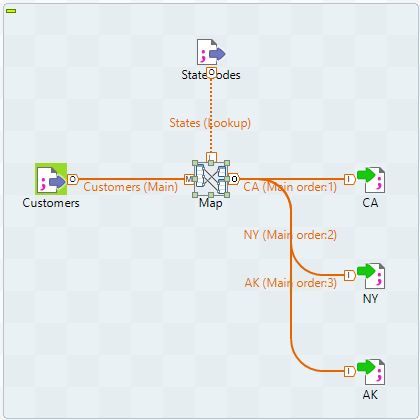
**c.-** Para la definición del filtro, ingrese la expresión *"CA".equals(States.StateCode)*.  
Puede soltar el campo State.StateCode en la expresión de filtro o escribir el nombre del campo.Observe que cuando la expresión del filtro está completa, una flecha naranja muestra la asignación de entrada a salida.  
  
💡 Esta expresión de filtro está escrita en Java. A Java no le gustan los valores nulos. El uso de esta sintaxis evita errores cuando StateCode es nulo.  
  
 **d.-** Click **Ok**.  
  
**2.-** Ejecute el Job y observe que el output contiene 27 rows.  
  
  
**3.-** Examine la salida.  
  
Abra data viewer para la salida del componente **CA**.  
  
Data viewer le mostrará solamente los clientes de California.  
  
Click **Close**.

## Añadir diferentes salidas para Alaska y New York

Repita los pasos anteriores para extraer clientes de Alaska y Nueva York en dos archivos de salida diferentes.

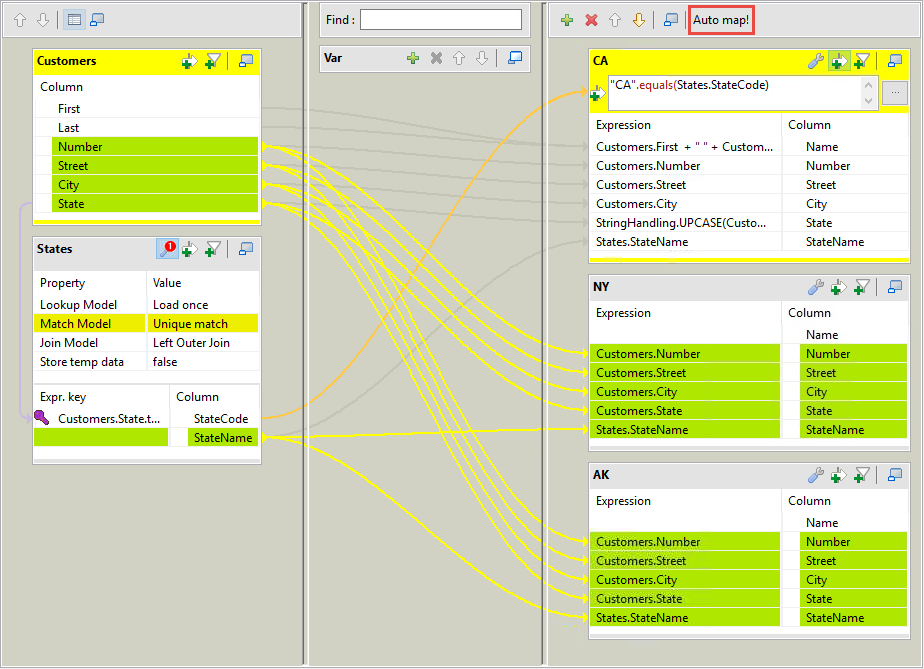
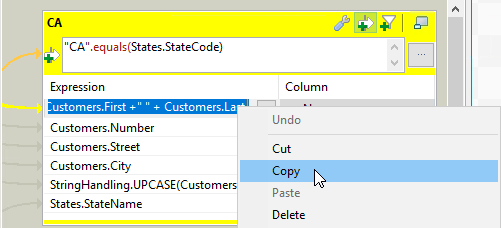
**1.-** Haga una copia del componente de salida existente.  
  
 **a.-** Click derecho en **CA** y seleccione **Copy**.  
  
  
 **b.-** En **Designer**, click derecho debajo del componente de salida, presione **Paste** para ubicar el nuevo componente **tFileOutputDelimited** debajo del existente, y renómbrelo *NY* (por New York).  
  
 **c.-** Repita este paso para obtener una nueva copia, y renómbrela *AK* (por Alaska).  


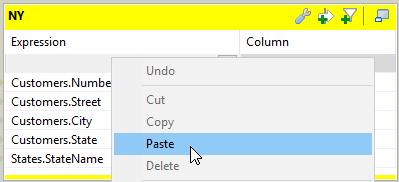
**2.-** Cambie la ruta de salida de los componentes.

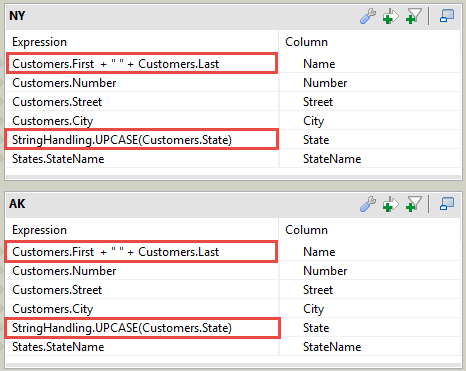
**a.-** Click **NY**, y para **File Name**, ingrese *"C:/StudentFiles/DIBasics/ProcessingData/NYOut.txt"*.  
  
  
 **b.-** Click **AK**, y para **File Name**, ingrese *"C:/StudentFiles/DIBasics/ProcessingData/AKOut.txt"*.  
  
  
**3.-** Cree nuevas salidas de tMap y conéctelas a **NY** y **AK**.  
  
 **a.-** Click derecho en el componente **Map** y seleccione **Row** > **\*New Output\* (Main)**. Vincule la nueva salida con **NY**.  
  
  
 **b.-** Ingrese como nombre *NY* y presione **OK**.  
  
 **c.-** Click **Yes** para obtener el schema del componente NY.  
  
  
 **d.-** Haga lo mismo para **AK**.  
  
El Job debería verse así:  


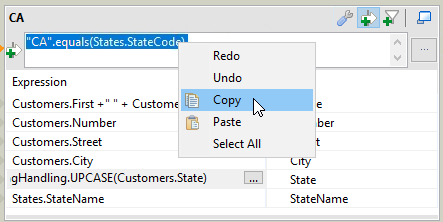
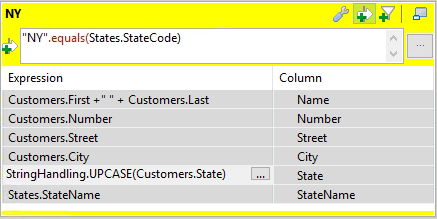
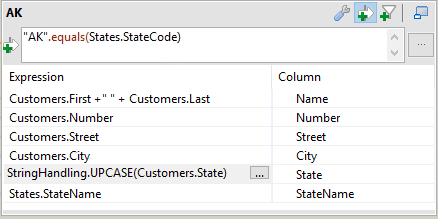
## Editar el componente Map

Create mapping for the new output tables.

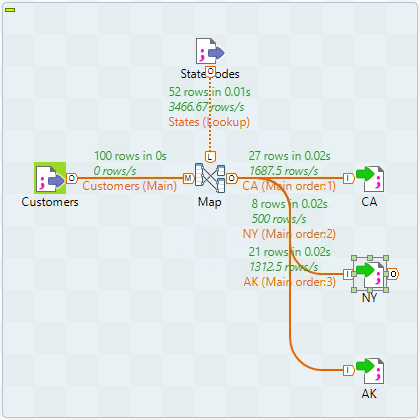
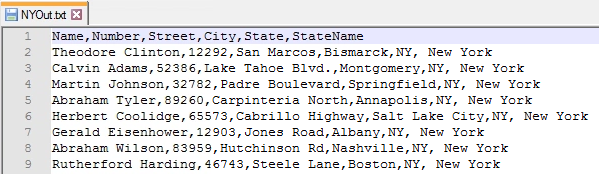
**1.-** Mapee la entrada con la nueva salida.  
  
Doble click en el componente **Map**.  
  
En la esquina superior derecha, haz clic en **Auto map!** El sistema crea una asignación entre la entrada y la salida en función de la definición del schema:  
  
Observe que no hay ninguna expresión asociada a la columna Name. Para la columna State, también falta la expresión en mayúsculas.  
  
**2.-** Copie las expresiones.  
  
La expresión de concatenación que crea para la salida de CA se puede reutilizar sin más modificaciones. Cópialo y pégalo en las otras salidas..  
  
 **a.-** En la salida **CA**, seleccione la expresión de la columna **Name**, haga click derecho y presione **Copy**.  


**b.-** En la salida **NY**, seleccione la expresión de la columna Name, haga click derecho y presione **Paste**.  
  
  
 **c.-** Repita este paso para la salida **AK**.

**3.-** Utilice la misma secuencia para copiar la expresión de la columna State de la salida de CA a NY y AK.   
  
  
**4.-** Copie la expresión de filtro que creó para CA y péguela en las otras salidas.  
  
 **a.-** Si el filtro no está abierto, haga clic en **Enable/disable the expression filter** en la esquina superior derecha de la salida de CA.

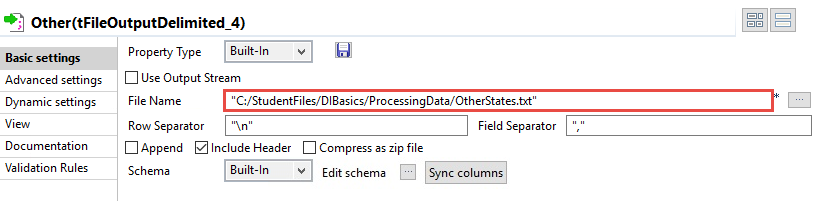
**b.-** Copie la expresión **"CA".equals(States.StateCode)**.  
  
  
 **c.-** Click en **Enable/disable the expression filter** de la salida **NY** y pegue la expresión, cambie **"CA"** po *"NY"*:  
  
  
 **d.-** Haga lo mismo para la salida **AK**.  
  
  
**5.-** Click **Ok**.

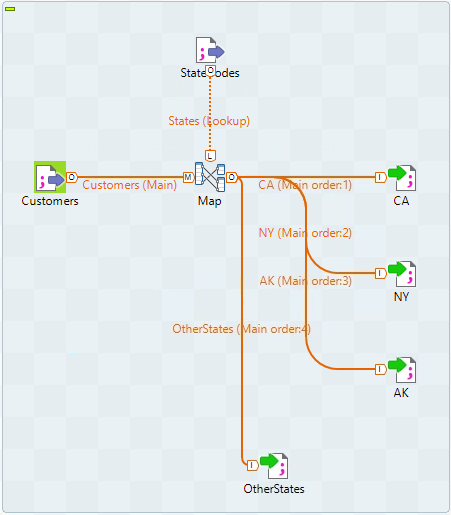
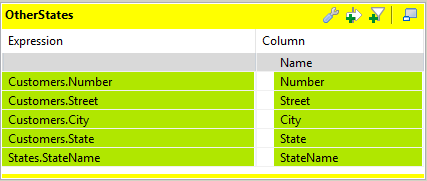
## Ejecutar el Job

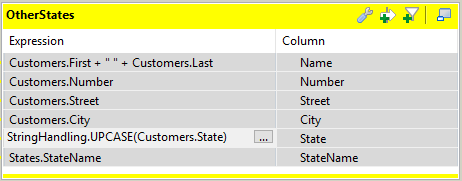
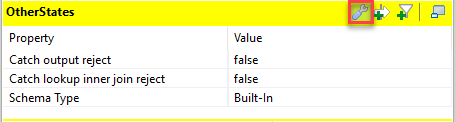
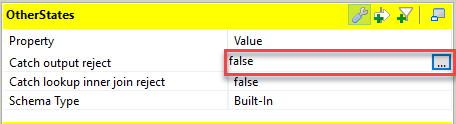
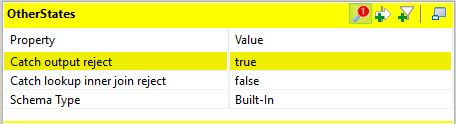
**1.-** Ejecute el trabajo y observe cómo los resultados se escriben en diferentes archivos de salida según los filtros.  
  
  
**2.-** Verifique los archivos de salida.  
  
Navegue hasta la carpeta **C:\StudentFiles\DIBasics\ProcessingData** y examine los tres archivos de salida. Este es un ejemplo tomado de NYOut.txt:  


## Recolectar filter rejects

Los registros que no coinciden con los filtros definidos en el mapeo se pueden recopilar de la misma manera que con los join rejects, esta vez utilizando la opción **Catch output reject** en la configuración de tMap. Para hacer esto, agregue otro componente de salida para recopilar registros correspondientes a estados que no sean Nueva York, California y Alaska.

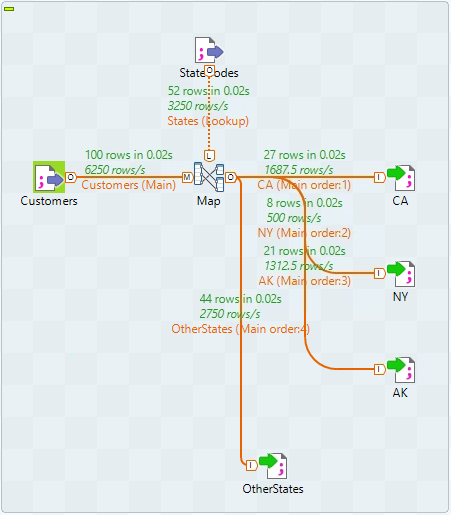
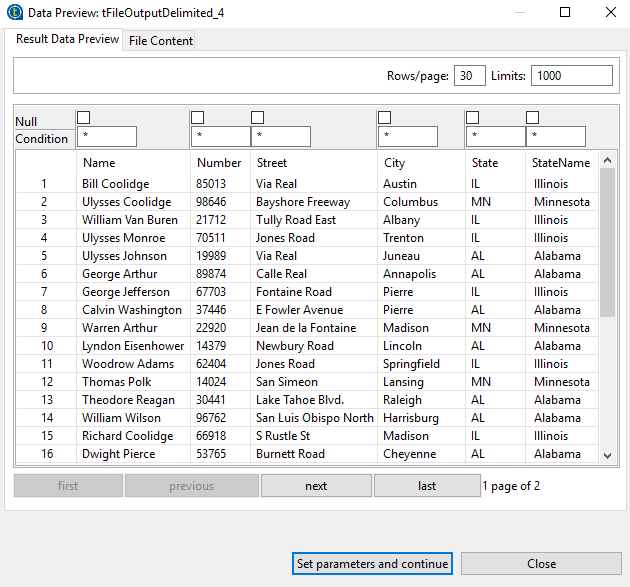
**1.-** Repita los pasos anteriores y cree un componente de salida llamado *OtherStates*, luego configure el file name *"C:/StudentFiles/DIBasics/ProcessingData/OtherStates.txt"*.  


**2.-** Conecte **Map** con **OtherStates** usando una **Main** row. Nombre la conexión *OtherStates*, click **Yes** para obtener el schema del componente en el mapeo.  
  
  
**3.-**.En el editor de mapeo, configure la salida de OtherStates como lo hizo para las anteriores.  
  
 **a.-** Doble click en **Map**.  
  
 **b.-** Click **Auto map!**  


**c.-** Copie las expresiones para **Name** y **State**.  
  
  
**4.-** En el editor de mapeo, configure **OtherStates** para capturar filter rejects.  
  
 **a.-** Click **tMap settings** (**wrench**) en la tabla **OtherStates**.  
  
  
 **b.-** En la columna **Value**, a la derecha de **Catch output reject**, click **false**, luego [**...**].  
  
  
 **c.-** En **Options**, seleccione **true** y presione **OK**.  
  
  
 **d.-** Click **Ok**.

## Ejecutar el Job

Ejecute el Job, luego vea los rechazos del filtro y el número de filas escritas en los archivos de salida.

**1.-** Ejecute el job y observe cómo los resultados se canalizan a diferentes archivos de salida en función de los filtros..  
  
  
**2.-** Visualizar los rechazos.  
  
En el componente **OtherStates**, abra data viewer y observe que los clientes rechazados no provengan de New York, California, o Alaska.  


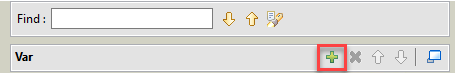
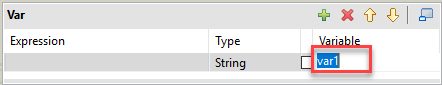
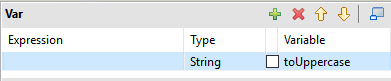
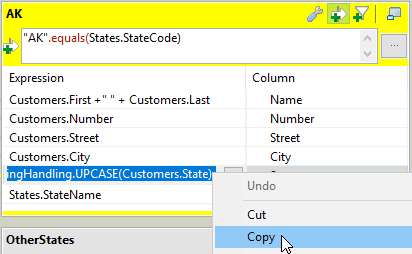
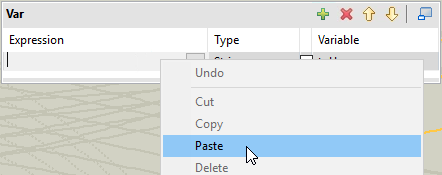
## Usar variables

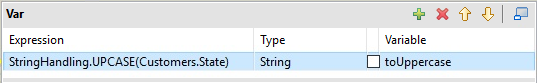
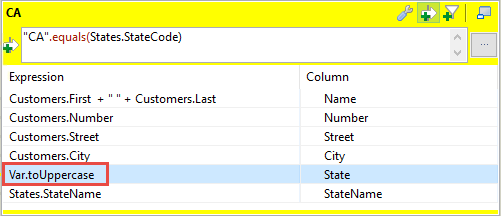
Una ventaja de usar variables es que puede crear expresiones más compactas, lo que facilita la reutilización.

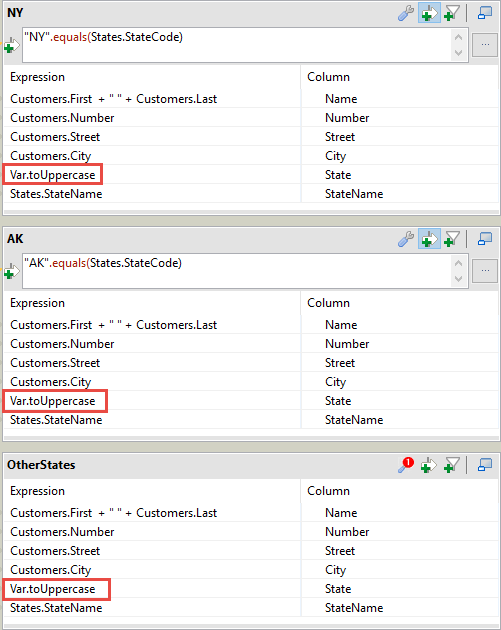
Para crear una variable, debe establecer un nombre, una expresión y type Java.

Modifique las tablas de salida para usar una variable en lugar de la expresión **StringHandling.UPCASE(Customers.State)**.

**1.-** En el editor de mapeo, cree una variable de tipo String con la expresión **StringHandling.UPCASE(Customers.State)**.  
  
 **a.-** Doble click en **Map**.

**b.-** En el panel central, a la derecha de Var, haga clic en [+].  
  
  
 **c.-** Debajo de **Variable**, click **var1** para editar el nombre.  
  
  
 **d.-** Ingrese *toUppercase*.  
  
  
 **e.-** Desde una de las tablas de salida, copie la expresión **StringHandling.UPCASE(Customers.State)**.  
  
  
 **f.-** Para definir el valor de la variable, pegue la expresión en **Expression**.  


La variable definida debería verse así:  
  
El typo por defecto es String, lo que es correcto porque esta expresión retorna un valor string.  
  
**2.-** Reemplace la expresión en las tablas de salida con la variable.  
  
 **a.-** En la salida de CA, reemplace la expresión con Var.toUppercase.  
  
  
💡 Para usar la finalización automática, ingrese var, presione CTRL+Espacio y haga doble clic en la variable que desee dentro de la lista de sugerencias.

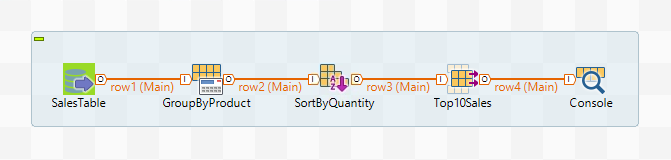
**a.-** Repita el paso previo para las salidas de **NY**, **AK**, y **OtherStates**.  
  
  
 **c.-** Click **Ok**.  
  
Tenga en cuenta que usar variables es fácil y es menos probable que introduzca errores.  
  
**3.-** Ejecute el Job.  
  
Abra la pestaña **Run** y presione **Run**. Confirme que el Job finaliza con **[exit code=0]**, y use data viewer para confirmar que los datos son iguales a los de la ejecución anterior.

**Tema 6: Usar otros componentes para el procesamiento de datos**

## Antes de comenzar

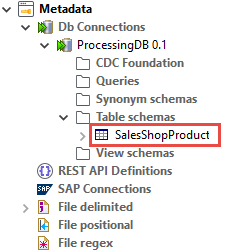
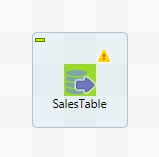
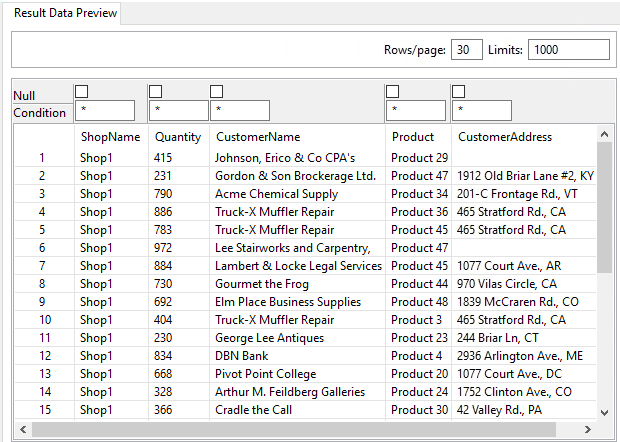
En esta sección se describen varias tareas de procesamiento de datos. Se centra en las operaciones de fila, como el muestreo, la agregación y la clasificación. El objetivo de este ejercicio es crear un Job que, basado en datos leídos de una tabla, calcule los diez productos más vendidos.

Al finalizar, el Job debería verse así:



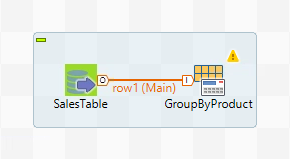
## Leer datos de una tabla MySQL

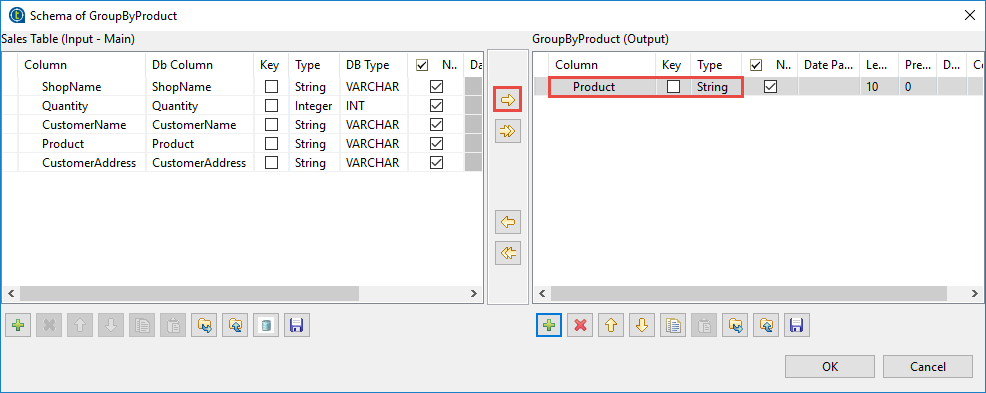
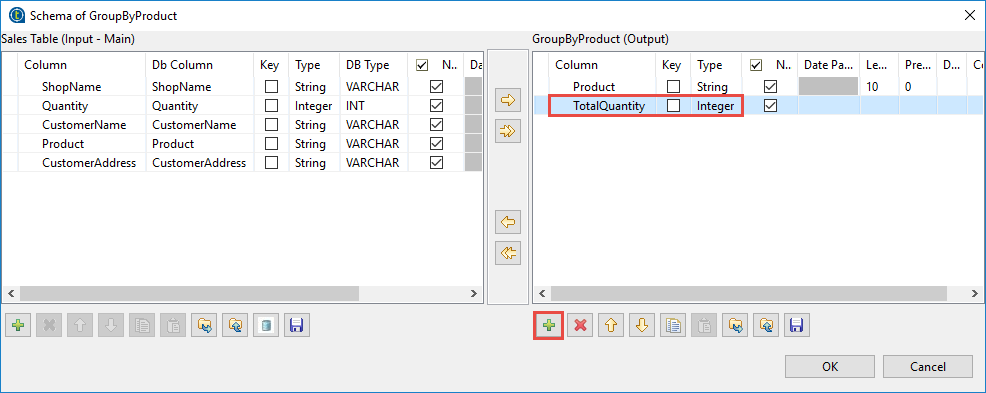
Los datos de ventas se almacenan en la tabla SalesProductShop. Para leer datos de esta tabla, configure un componente tDBInput utilizando los metadatos de SalesTableSchema disponibles en el Repositorio.

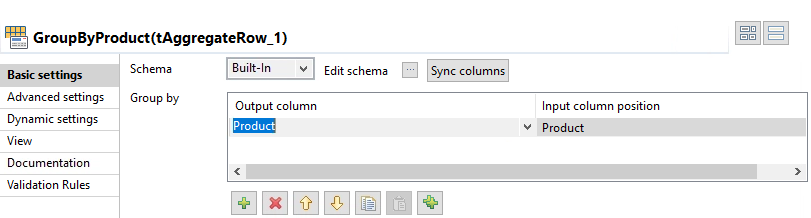
**1.-** Cree un Job.  
  
En **Repository** > **Job Designs** > **Standard** > **ProcessingData**, cree un standard Job y nómbrelo *processData*.  
  
**2.-** Agregue un componente para leer datos de una tabla MySQL.  
  
En el nuevo Job, agregue un componente **tDBInput** usando **SalesTableSchema** como schema.  
  
 **a.-** En **Repository** > **Metadata** > **Db Connections** > **ProcessingDB** > **Table schemas**, seleccione **SalesShopProduct** y arrástrelo al **Designer**.  
  
  
 **b.-** En la ventana **Components**, seleccione **tDBInput** y presione **OK**.  
  
 **c.-** Edite el nombre del componente e ingrese *SalesTable*.  
  
  
**3.-** Use data viewer para el componente **SalesTable** component para visualizar los datos extraídos de la tabla.  
  
La tabla SalesShopProduct almacena la cantidad de unidades asignadas por cliente, producto y tienda.

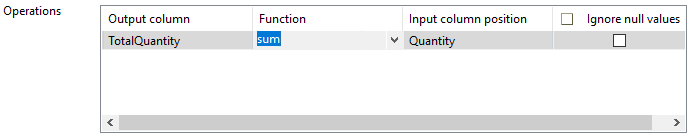
## Calcular la cantidad vendida por producto

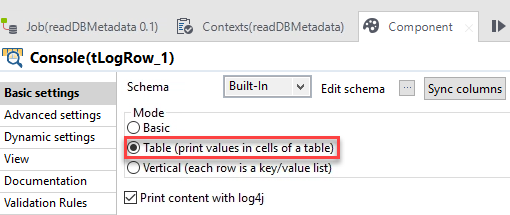
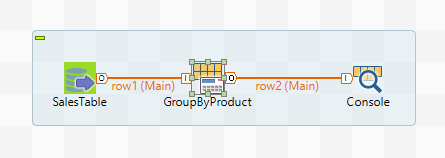
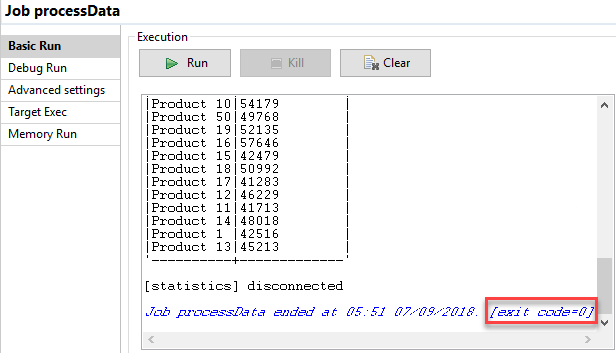
Para determinar los productos más vendidos, el primer paso es calcular la cantidad total de unidades vendidas por producto mediante el uso del componente tAggregateRow, para eliminar la información del cliente y la tienda del conjunto de datos.  
  
**1.-** En **Designer**, agregue un componente **tAggregateRow** a la derecha de **SalesTable**.  
  
Cambia el nombre por *GroupByProduct*.

**2.-** Conecta **SalesTable** y **GroupByProduct** usando una **Main** row.  


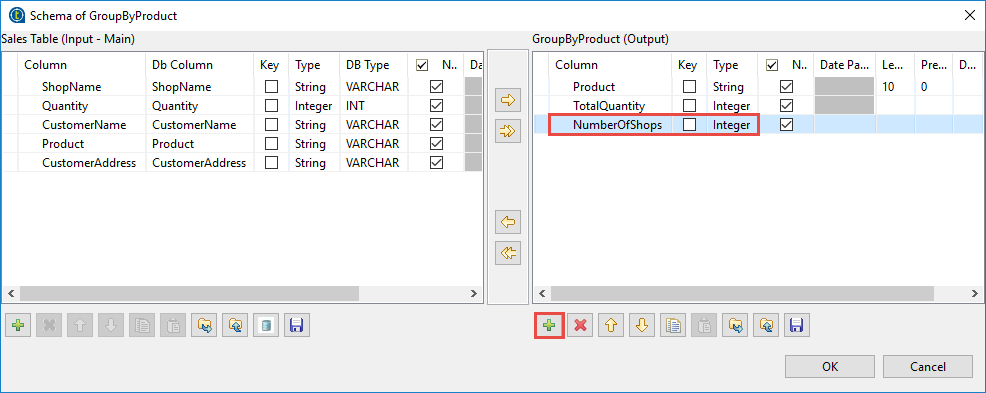
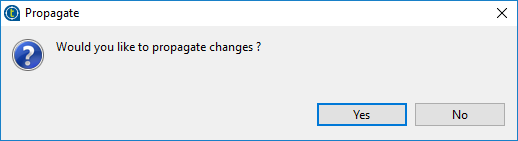
**3.-** Configure el schema del componente.  
  
El resultado esperado es una lista de productos con las respectivas cantidades totales vendidas. Defina un esquema de componentes con dos columnas:  
 → **Product**: copiado desde el schema de entrada  
 → **TotalQuantity**: nueva columna de tipo Integer  
  
 **a.-** Seleccione el componente GroupByProduct y abra la pestaña **Component**.  
  
 **b.-** A la derecha de **Edit schema**, click [**...**].  
  
 **c.-** En el schema de entrada, seleccione **Product** y haga click en el botón **arrow** para copiar la selección al schema de salida.  
  
  
 **d.-** En el schema de salida, agregue una columna, click [**+**]. Nómbrela *TotalQuantity*, y para **Type**, seleccione **Integer**.  
  
  
 **e.-** Click **OK**.

**4.-** Agrupe por producto.  
  
Declare la columna **Product** en la sección **Group by**.  
  
 **a.-** En **Component**, debajo de la sección **Group by**, click [**+**].  
La columna Product se agrega automáticamente.  
  
 **b.-** Observe que las columnas son listas desplegables en las que puede seleccionar todas las columnas de los esquemas de entrada y salida, respectivamente. Esto permite muchas opciones, pero en este caso, lo mantiene simple al conservar los valores predeterminados, como se muestra en la captura de pantalla:  


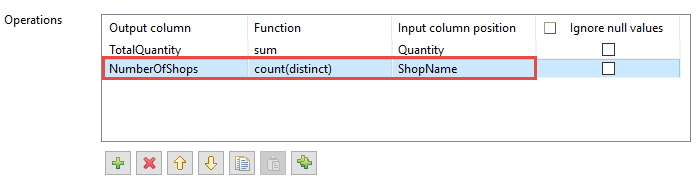
**5.-** Usar la agregación para sumar cantidades.  
  
En la tabla **Operations**, configure la salida **TotalQuantity** como la suma de la tabla **Quantity**.  
  
 **a.-** Debajo de la sección **Operations**, click [**+**].  
  
Asegúrese de que **TotalQuantity** está seleccionado como **Output column** y **Quantity** como **Input column position**.  
  
 **b.-** Para **Function**, seleccione **sum**.  
  
  
**6.-** Imprimir los datos de agregación en la consola.  
  
En **Designer**, conecte **GroupByProduct** con un componente **tLogRow**.  
  
 **a.-** Añada un componente **tLogRow** a la derecha de **GroupByProduct**.

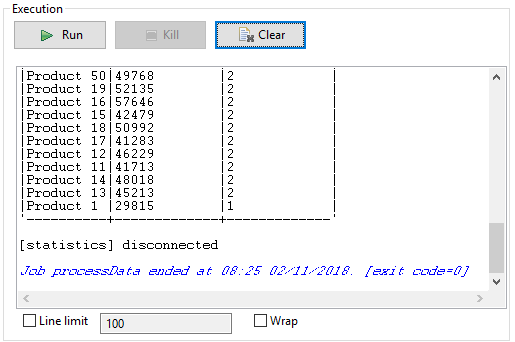
**b.-** Cambie el nombre del componente por *Console*.  
  
 **c.-** Configure el componente en Mode **Table**.  
  
  
 **d.-** Conecte **GroupByProduct** y **Console** usando **Main** row.  
  
  
 **e.-** En la pestaña **Run**, click **Run**.  
  
El Job debe finalizar con **[exit code=0]**.

## Calcular el número de tiendas por producto

Las columnas no numéricas también pueden utilizar funciones de agregación, los String se pueden contar o enumerar. Agregue una nueva operación en GroupByProduct para contar el número de tiendas en las que se ha vendido cada producto.  
  
**1.-** Actualice el schema de salida.  
  
Cree una nueva columna en el schema de salida y configure la operación.  
  
 **a.-** Haga doble click en **GroupByProduct**.  
  
 **b.-** A la derecha de **Edit schema**, click [**...**].  
  
 **c.-** En el output schema, añada una columna, click [**+**]. Nómbrela *NumberOfShops*, y para **Type**, seleccione **Integer**.  
  
  
 **d.-** Click **OK**.  
  
 **e.-** Click **Yes** para propagar los cambios.  


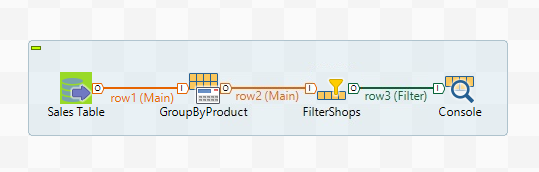
**2.-** Configure una nueva operación para contar el número de tiendas distintas por producto.

**a.-** Debajo de la sección **Operations**, click [**+**].  
  
Asegúrese de que **NumberOfShops** está seteado como **Output column** y **ShopName** como **Input column position**.  
  
 **b.-** Para **Function**, seleccione **count(distinct)**.  


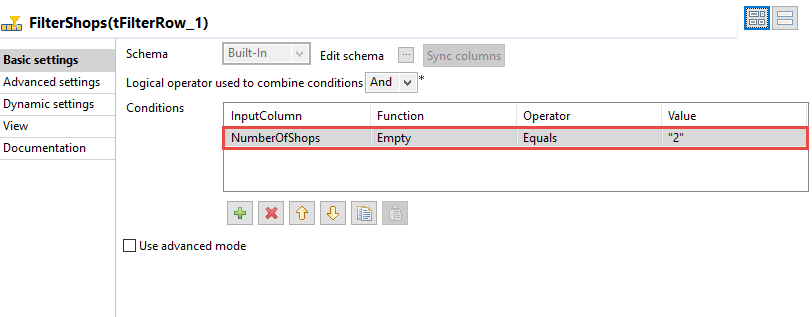
**3.-** Ejecute el Job.  
  
Click **Run**.  
  
La mayoría de los productos se han vendido en dos tiendas; solo unos pocos en una sola tienda.

## Filtrado de datos incompletos

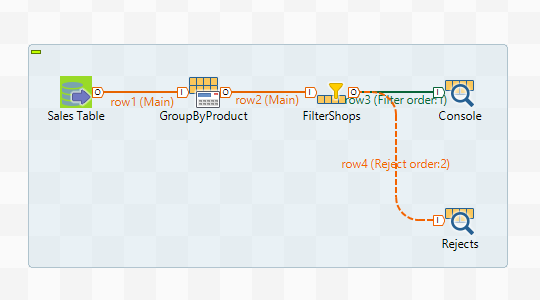
Anteriormente filtró datos usando tMap. El componente tFilterRows también se puede usar para configurar filtros más simples.

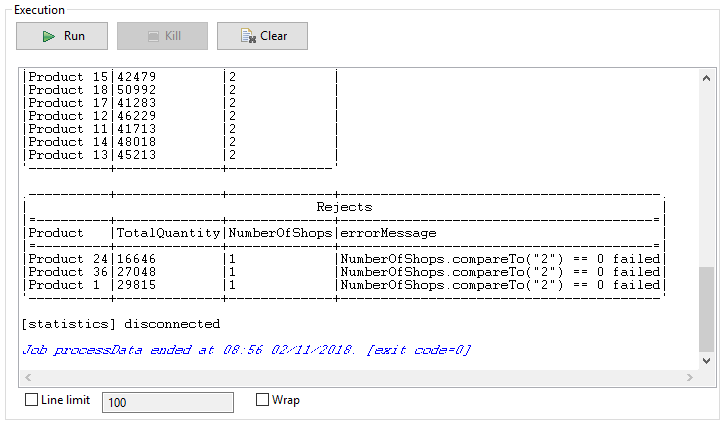
Estimando que faltan algunos datos de la segunda tienda, use el componente tFilterRow para filtrar los productos vendidos solo en una tienda. En la consola, muestre el conjunto de datos resultante y las filas rechazadas.  
  
**1.-** En **Designer**, añada un componente **tFilterRow** entre **GroupByProduct** y **Console** y nómbrelo *FilterShops*.  
  
Observe que tLogRow está conectado a tFilterRow por una fila de filtro verde.  
  
💡 Puede insertar un componente en la canalización de procesamiento de datos utilizando diferentes métodos:  
 → Seleccione una fila de conexión, escriba las primeras letras del componente y luego selecciónelo en la lista. El componente aparece conectado directamente.  
 → Coloque el componente en Designer y actualice las conexiones manualmente.

Si se le solicita que obtenga el esquema del componente de destino (el siguiente componente de la cadena), es preferible rechazar.  
  
En este ejercicio, cada vez que inserte un componente, utilice su método favorito.  
  
**2.-** Configure el filtro.  
  
Use tFilterRow para mantener las columnas donde **NumberOfShops** sea igual a "2".  
  
 **a.-** Abra la pestaña **Component**.  
  
 **b.-** Debajo de la tabla **Conditions**, click [**+**] y setee los siguientes parámetros:  
  
 → Para **InputColumn**, seleccione **NumberOfShops  
 →** Para **Funtion**, mantenga el valor por defecto  
 → Para **Operator**, mantenga el valor por defecto  
 → Para **Value**, ingrese *"2"*

El componente debe verse así:  


**3.-** Mostrar las columnas rechazadas.  
  
El componente tFilterRow tiene dos filas de salida, Filter y Reject.  
  
Utilice un componente tLogRow para mostrar los datos canalizados a la fila Reject del componente tFilterRow.  
  
 **a.-** Coloque un segundo **tLogRow** debajo del primero y nómbrelo *Rejects*.  
  
 **b.-** Configure el componente en **Table** mode.  
  
 **c.-** Conecte **tFilterRow** usando **Reject** row.

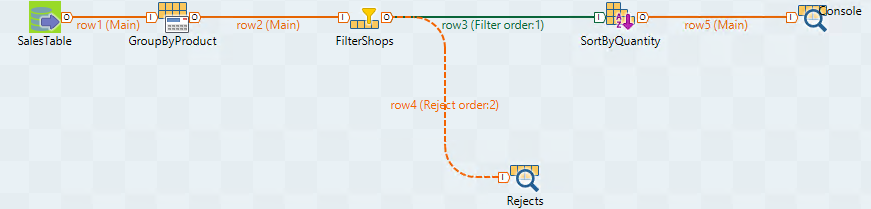


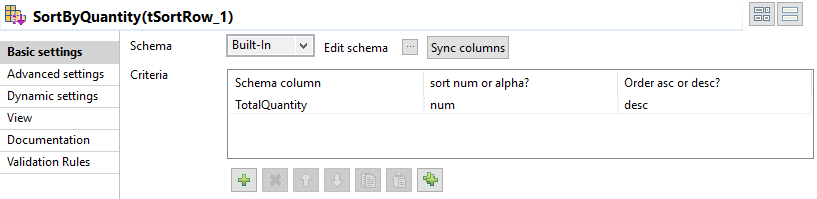
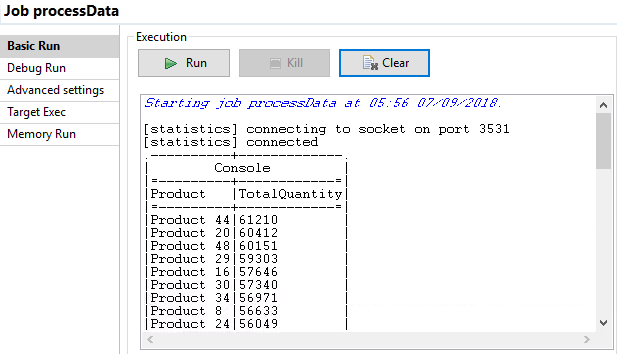
**4.-** Ejecute el Job.  
  
Click **Run**.  
  
Las dos salidas se muestran en la consola. Se agrega una columna de mensaje de error al flujo de rechazo.

## Clasificar por cantidad

El componente tSortRow ordena los valores de entrada numéricos o alfanuméricos según los criterios.

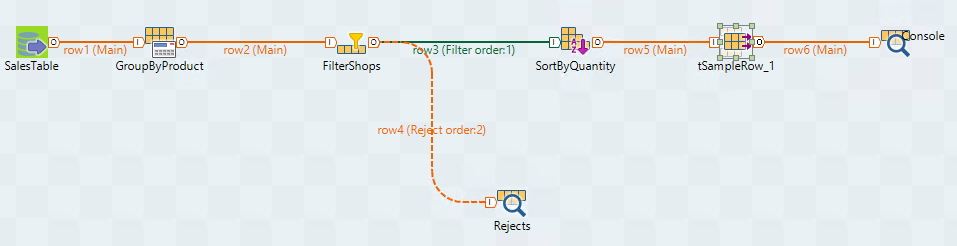
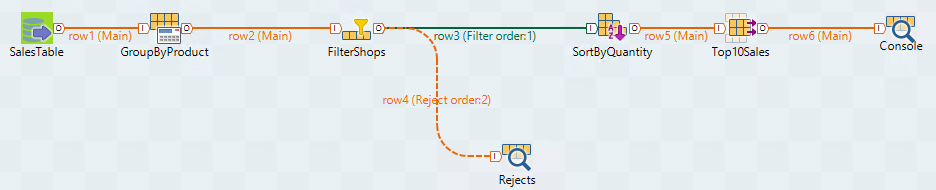
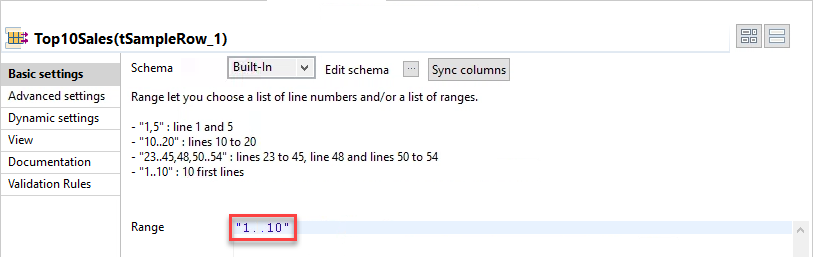
Use un componente tSortRow para ordenar productos por cantidad total vendida.

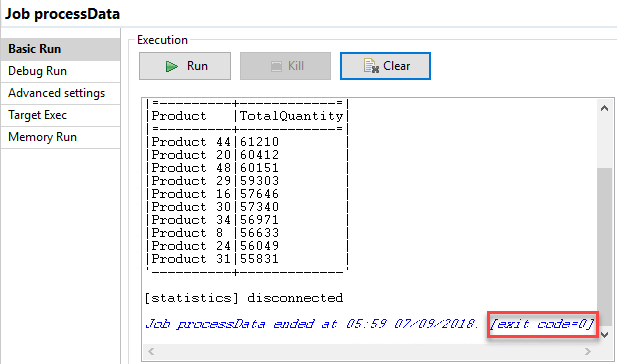
**1.-** En **Designer**, añada un componente **tSortRow** entre **FilterShops** y **Console** y nómbrelo *SortByQuantity*.  
  
  
**2.-** Configure el componente SortByQuantity.  
  
Para determinar los productos más vendidos, ordene los productos según la cantidad total en orden numérico descendente.

**a.-** Debajo de la tabla **Criteria**, click [**+**].  
  
 **b.-** Configure los nuevos criterios de filtro:  
  
 → Para **Schema column**, seleccione **TotalQuantity  
 →** Para **sort num or alpha?**, seleccione **num  
 →** Para **Order asc or desc?**, seleccione **desc**  
  
  
**3.-** Ejecute el Job.  
  
En la pestaña **Run**, click **Run**. Observe que las cantidades se enumeran en orden descendente.  


## Muestreo de filas para determinar los diez principales vendedores

Ahora que los productos se enumeran en orden descendente según la cantidad vendida, es fácil identificar los diez productos más vendidos. Utilice un componente tSampleRow para extraer las primeras diez filas de los resultados anteriores

**1.-** En **Designer**, añada un componente **tSampleRow** entre **SortByQuantity** y **Console**.  
  
  
**2.-** Renombre el componente a *Top10Sales*.  
  
  
**3.-** Configure el componente Top10Sales.  
  
Para obtener los diez productos principales, extraiga las primeras diez filas de la entrada enviada por SortByQuantity designando el rango como "1..10".  
  
 **a.-** Abra la pestaña **Component**.  
   
 **b.-** En **Range**, ingrese *"1..10"*.

**4.-** Ejecute el Job.  
  
En la pestaña **Run**, click **Run**.  
  
Observe que el Job ahora enumera los diez productos más vendidos.

## Resumen

En este módulo aprendiste cómo:

* Mapear datos de entrada a una salida usando el componente tMap
* Unir datos y recopilar join rejects
* Filtrar datos y recopilar filter rejects
* Usar expresiones y variables en el editor de mapeos
* Usar otros componentes de procesamiento de datos