**AYI ACADEMY: TALEND**

**Material práctico**



**Módulo 3: Trabajando con base de datos**

Módulo 3: Trabajando con bases de datos

* **[Temas:](https://cdn5.dcbstatic.com/files/t/a/talendacademy_docebosaas_com/1651694400/bKlhlM5REIPIICbkJ0xX_Q/scorm/b26dc53e2d3e1a4b14723e3ed52096c2cb2fa507/Content/Training/WorkingWithMySQLdb/Overview.htm)**
* **Preparación del entorno.**
* **Creación de tablas en MySQL.**
* **Leer datos desde una tabla MySQL.**
* **Buenas prácticas.**
* **Resumen.**

## Objetivos

Después de completar este módulo, serán capaces de:

* Conectarse a una base de datos desde un Job Talend.
* Escribir en una tabla MySQL.
* Leer datos desde una tabla MySQL.
* Usar SQL Builder para customizar una consulta (query) SQL.
* Compartir una conexión de base de datos.
* Cerrar la conexión cuando termine el Job Talend.

## Ejercicio

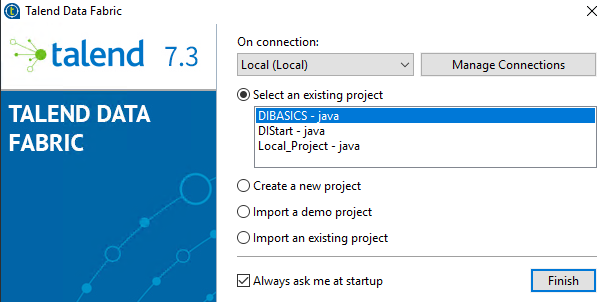
Estos ejercicios muestran cómo realizar operaciones básicas en tablas MySQL utilizando Talend Studio y, en particular, se centran en cómo crear una tabla, insertar datos y consultar datos.

**Tema 1: Preparación del entorno**

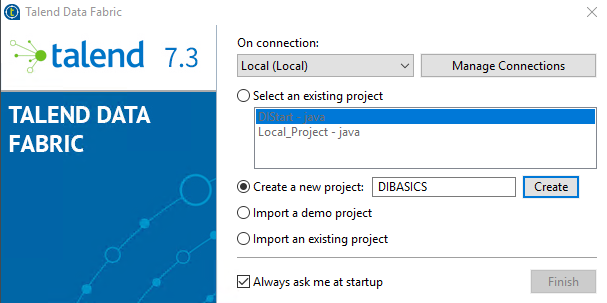
## Antes de comenzar

Antes de comenzar a trabajar con bases de datos, debemos iniciar Talend Studio y abrir un proyecto en nuestro workspace (espacio de trabajo).

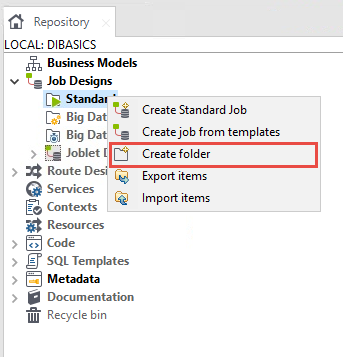
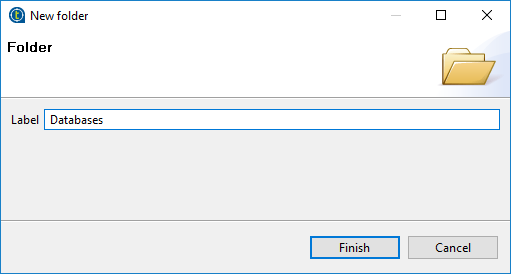
## Abrir un proyecto

**1.-** Haga doble clic en el icono de Studio en su escritorio para iniciar Talend Studio. Se abre la ventana **Talend Data Fabric**, que le permite abrir un proyecto existente o crear uno nuevo.  
  
**2.-** Abra el proyecto **DIBASICS.**   
  
**3.-** Confirme que el proyecto **DIBASICS** está disponible en la lista de proyectos existentes (existing project).  
  
**4.-** Si aparece el proyecto **DIBASICS**, haga clic en él y luego haga clic en **Finish**.  
  


**Si no poseen el proyecto, sigan estos pasos para crearlo:**

**1.-** En **On connection**, seleccione **Local (Local)**, lo que significa que cuando se crea el proyecto, se almacena en su máquina local.  
  
**2.-** Seleccione la opción **Create a new project** e ingrese como nombre ***DIBASICS***.  
  
  
  
**3.-** Haga click en **Create** y espere a que el proyecto aparezca en la lista. El proyecto que haya creado se seleccionará por defecto.  
  
**4.-** Haga click en el botón **Finish**.  
  
  
💡 Si se le solicita que se conecte a Talend Community, haga clic en **Skip this Step**.  
💡 Si se abre la **Welcome** **page** (página de bienvenida), haga click en **Start now.**

## Configurando el proyecto

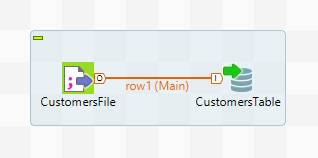
**1.-** En la barra de tareas ubique **Quick Access** y asegúrese de que la perspectiva **Integration** esté seleccionada.  
  
  
**2.-** Cree una nueva carpeta para sus Jobs. En el repositorio, expanda la opción **Job Designs**, haga clic derecho sobre **Standard**, y seleccione **Create folder**.  
  
  
**3.-** Ingrese *Databases* como **Label**, y haga clic en **Finish**.  
  
  
4.- La nueva carpeta se va a encontrar en **Repository** > **Job Designs** > **Standard**.

**Tema 2: Creación de tablas en MySQL**

## Antes de comenzar

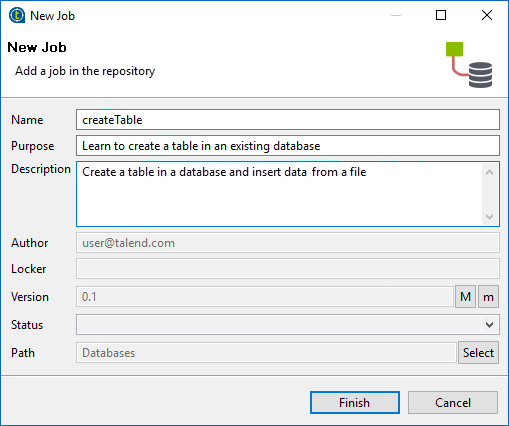
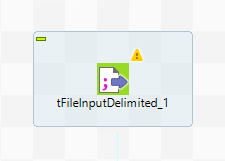
Este ejercicio le enseñará a conectarse a una base de datos, crear una tabla en MySQL, y llenarla con algunos datos. El objetivo es crear un Job que llene la nueva tabla con datos que obtiene del archivo CustomersFile.csv.

Al final, el Job debería verse así:



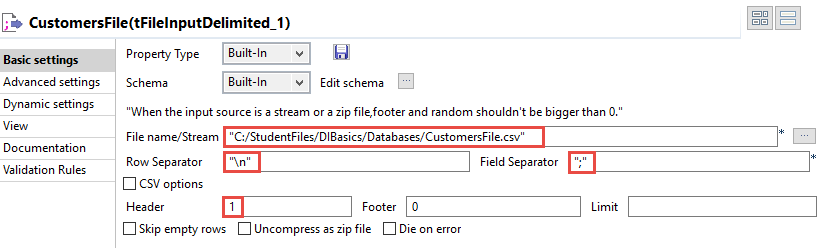
El entorno se ha configurado con una base de datos denominada training.

## Leer datos de un archivo estructurado

Para este ejercicio vamos a utilizar el archivo CustomersFile.csv ubicado en la carpeta C:\StudentFiles\DIBasics\Databases. Examine el contenido del archivo para entender su estructura. Cada línea contiene información básica de los clientes, separada por punto y coma. La primera línea es la cabecera(header), que contiene los nombres de cada columna.  
  
**1.-** Cree un nuevo standard Job en la carpeta **Databases** y nómbrelo *createTable*.  
  
  
  
**2.-** En el **Designer**, agregue un componente **tFileInputDelimited**.  
  
  
**3.-** Renombre el componente clickeando en su nombre y reemplácelo por *CustomersFile*.  

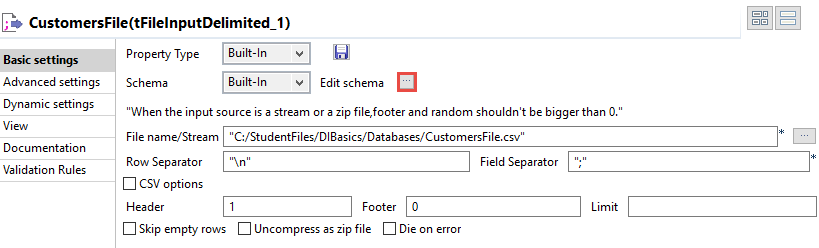

**4.-** Configure el componente especificando el nombre del archivo a leer y su estructura, definida por el separador de filas y el separador de campos.

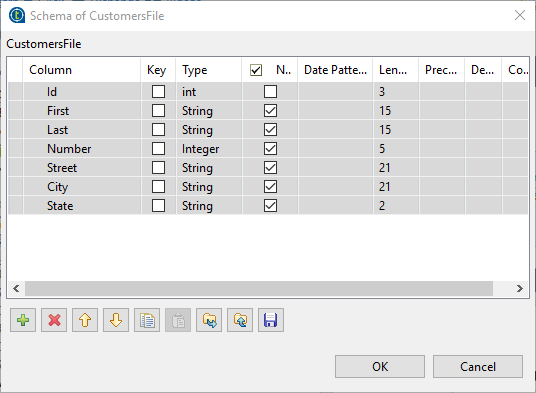
**a.-** Abrir la opción **Basic settings** en la pestaña **Component**.

**b.-** Para el **File name**, haga clic en [**...**] y seleccione la ruta del archivo **C:/StudentFiles/DIBasics/Databases/CustomersFile.csv**.  
  
**c.-** Deja **Row Separator** como viene por defecto (**"\n")**. Row Separator field identifica el final de una línea en el archivo.  
  
**d.-** En **Field Separator**, ingrese *";"* para especificar que el archivo contiene valores separados por punto y coma.  
  
**e.-** En la casilla de texto **Header**, ingrese *1*. Esto significa que la primera línea del archivo, que es la lista de los nombres de las columnas, debe ser ignorada.  


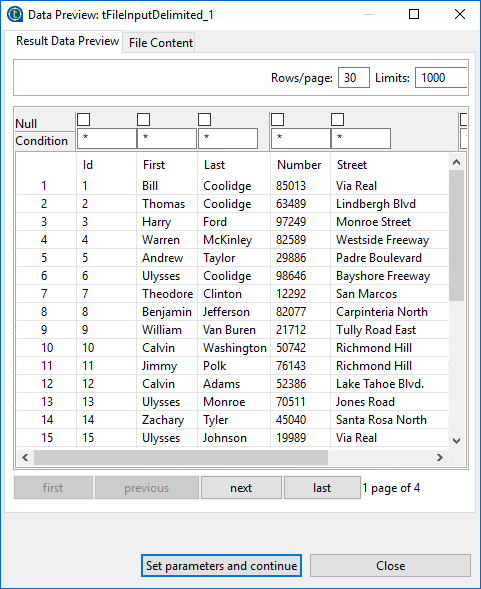
**5.-** Manualmente cree el schema añadiendo las columnas con sus respectivas propiedades.  
  
💡 En este ejercicio, vas a usar diferentes métodos para definir el schema de tu componente: puedes importarlo desde un archivo XML o copiarlos y pegarlos entre componentes.  
  
💡 Los schemas también puede ser creados por adelantado y almacenados en la metadata del repositorio. Con este método, los nombres de las columnas se importan automáticamente desde las filas de encabezado (header) del archivo de entrada y los esquemas se pueden reutilizar en varios componentes. Veremos esto más adelante.

**a.-** Haga clic sobre [**...**], ubicado a la derecha de **Edit schema**.



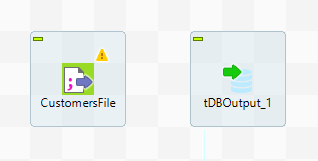
**b.-** Para cargar el schema desde un archivo XML, haga clic en el ícono **folder and arrow** en la barra de tareas y seleccione **"C:\StudentFiles\DIBasics\Databases\CustomersSchema.xml"**.  
  
  
  
 El Schema se actualizará de la siguiente manera.  
  
  
💡 Tenga en cuenta que la primera columna almacena el customer ID, por lo tanto, el atributo Nullable no está seleccionado.

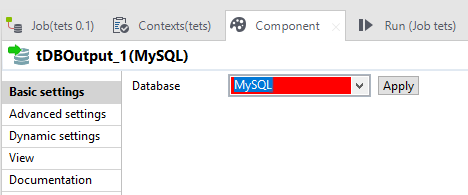
**c.-** Haga clic en **OK** para cerrar la ventana del schema.

**6.-** Use el **Data viewer** en el componente **CustomersFile** para previsualizar los datos extraídos del archivo.  
  


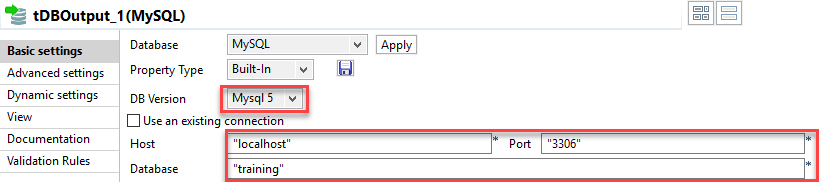
## Crear una tabla

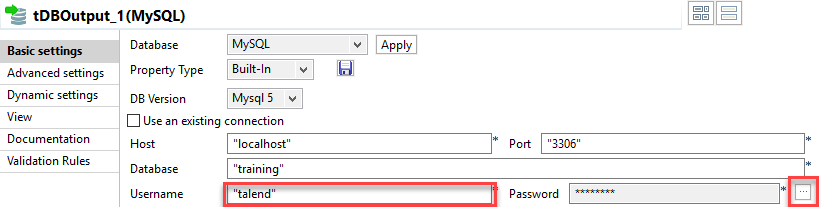
El componente tDBOutput es un componente genérico que realiza diferentes operaciones (incluidas write, create, update, and delete) en una tabla y en diferentes tipos de bases de datos. tDBOutput toma datos del componente anterior del Job y realiza una operación en la tabla de acuerdo con las propiedades definidas en la configuración.

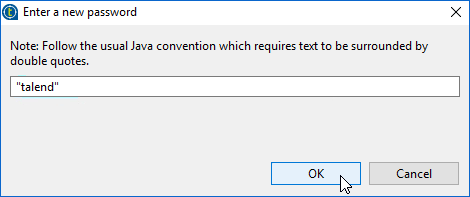
**1.-** En el **Designer**, agregue un componente **tDBOutput** a la derecha de **CustomersFile**.  


**2.-** Configure el componente para conectarse a la base de datos MySQL llamada **training**.  
  
 **a.-** Haga clic en el componente **tDBOutput** para configurar el tipo de base de datos. Para **Database**, seleccione **MySQL** en la lista desplegable y haga clic en **Apply**.  


**b.-** Para **DB Version**, seleccione **Mysql 5**.  
  
💡 Esta es la versión de la instancia de MySQL que se ejecuta en su entorno de práctica.

**c.-** Para **Host** ingrese *"localhost"*. Para **Port** ingrese *"3306"*, y para **Database** ingrese *"training"*.  


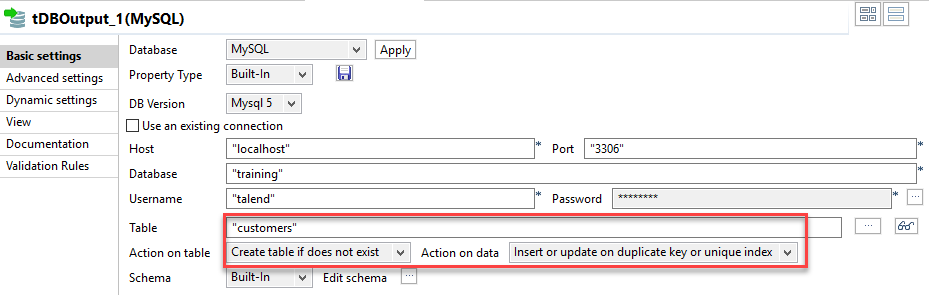
**d.-** Para **Username** ingrese *"talend"*.  


**e.-** Para ingresar el **Password**, haga click en [**...**]. En el cuadro de diálogo, ingrese *"talend"* y haga click en **OK**.  


💡 En la ventana **Enter a new password**, las comillas dobles son obligatorias porque Talend espera que el valor de este campo sea un string Java.

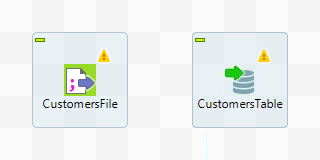
**3.-** Configure el componente para crear una nueva tabla e insertar datos.

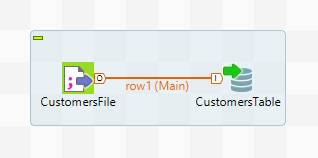
**a.-** Para **Table**, ingrese *"customers"*.

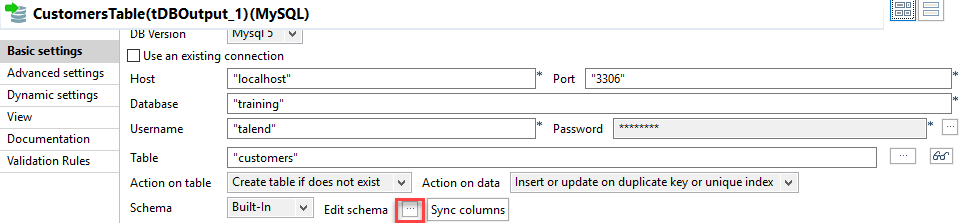
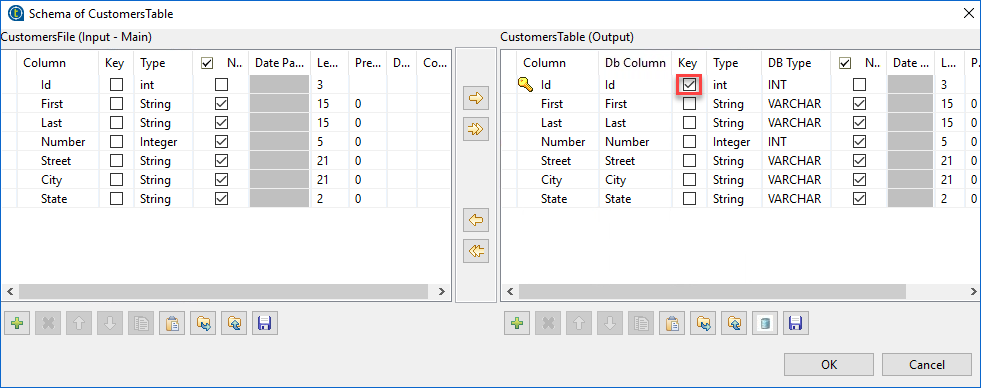
**b.-** En **Action on table** despliegue la lista, y seleccione **Create table if does not exist**. Para **Action on data**, seleccione **Insert or update on duplicate key or unique index**.  


## Llenar la tabla con datos obtenidos de un archivo de entrada

Siga los siguientes pasos para completar la tabla con datos del archivo de entrada.

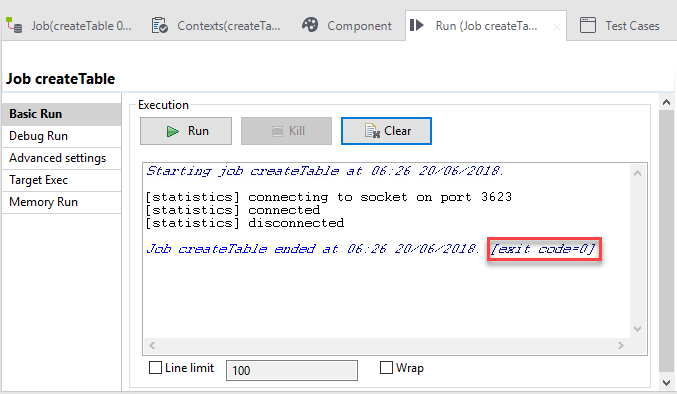
**1.-** Renombre el componente **tDBOutput** como *CustomersTable*.  


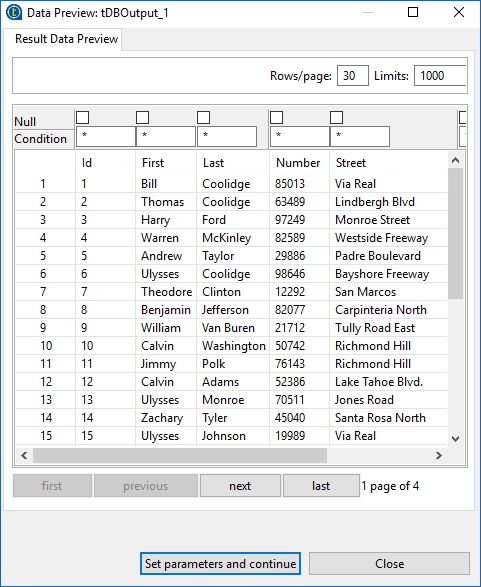
**2.-** Conecte **CustomersFile** y **CustomersTable** usando una **Main** row.  


**3.-** Modifique el schema de CustomersTable.  
  
El schema de este componente se utiliza para crear la estructura de la tabla. Una vez que los componentes están conectados, los schemas de CustomersTable y CustomersFile se sincronizan automáticamente. Sin embargo, el schema de CustomersTable necesita una clave primaria (PK Primary Key).  
  
Modifique el schema para definir la columna **Id** como la primary key.  
  
 **a.-** Haga click en **CustomersTable** para abrir la ventana **Component**.  
  
 **b.-** Haga click en [**...**] ubicado a la derecha de **Edit schema**.  
  
  
 **c.-** El schema de entrada (CustomersFile) y el de salida (CustomersTable) se muestran uno al lado del otro. Para declarar una primary key en el schema de la tabla, para la columna **Id**, seleccione el checkbox de **Key**.

💡 Tenga en cuenta que con cada column type, el schema de la tabla define el tipo de dato de la base de datos.

**4.-** Ejecute el Job y examine los resultados.  
  
 **a.-** En la pestaña **Run**, haga click en **Run** para ejecutar el Job.

**b.-** Confirme que el Job termina con **[exit code=0]**.  


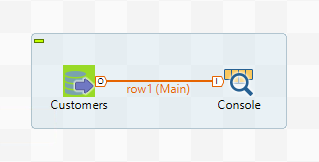
**5.-** Abra el visor de datos en el componente **CustomersTable** y confirme que los datos se cargaron en la tabla. Esta es una buena manera de confirmar que la configuración del componente es correcta.  


**Tema 3: Leer datos desde una tabla MySQL**

## Para comenzar

Este ejercicio abarca cómo usar los componentes de Studio para leer datos de tablas MySQL usando consultas personalizadas. El objetivo es crear un Job que consulte la tabla de clientes (creada en el ejercicio anterior) para imprimir la lista de clientes en consola.

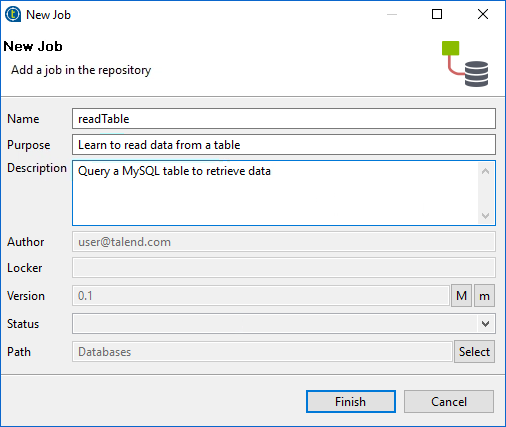
Al final, el Job debería verse así:



## Lectura de datos de una tabla MySQL

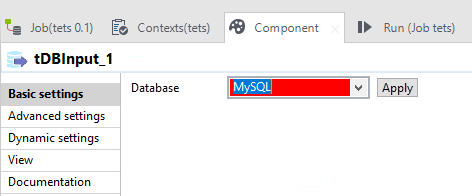
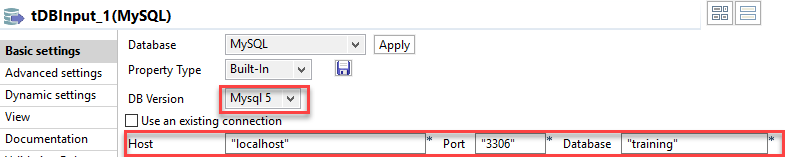
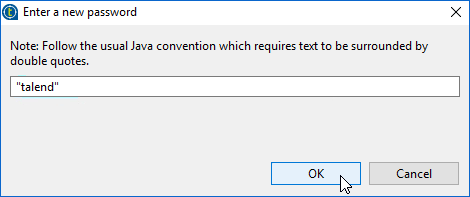
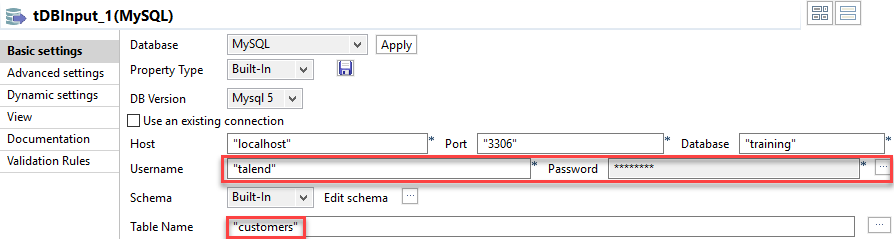
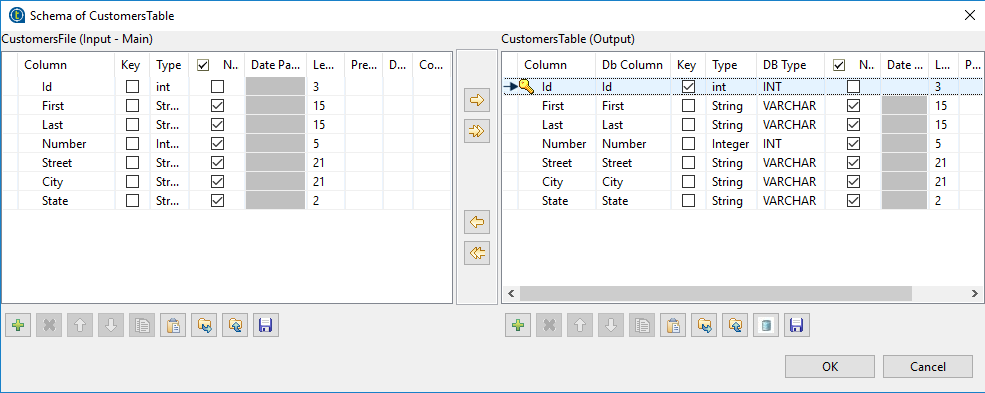
El componente tDBInput es un componente genérico que funciona con una variedad de bases de datos. tDBInput se puede configurar para ejecutar una consulta en una tabla según la definición del schema.

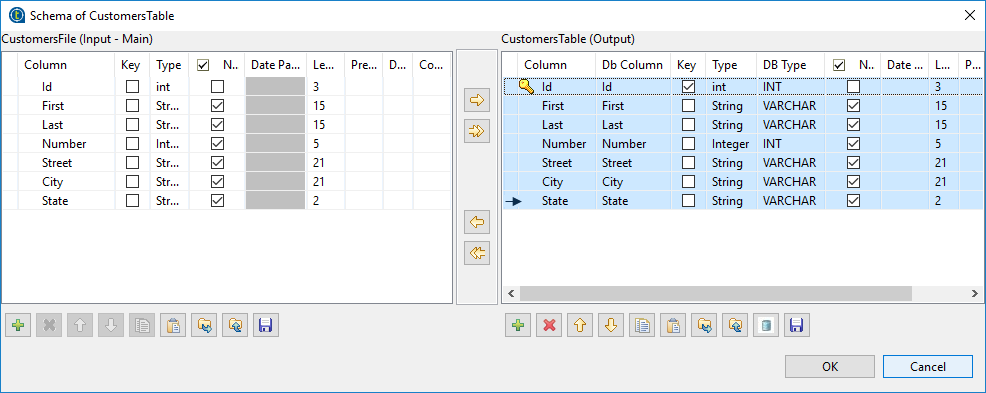
Para este ejercicio, se consulta la tabla de clientes para recuperar los clientes que viven en California. Para evitar errores, reutilice la definición del schema del ejercicio anterior.

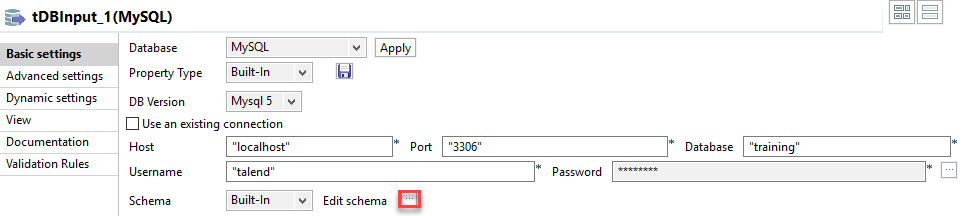
**1.-** Cree un nuevo standard Job en **Repository** > **Job Designs** > **Standard** > **Databases** y nómbrelo *readTable*.  


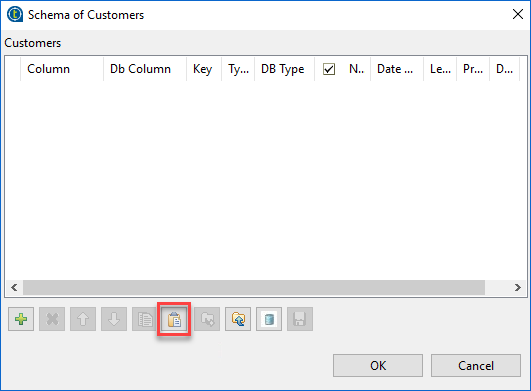
**2.-** En el **Designer**, agregue un componente **tDBInput** para leer los datos de una tabla MySQL.  

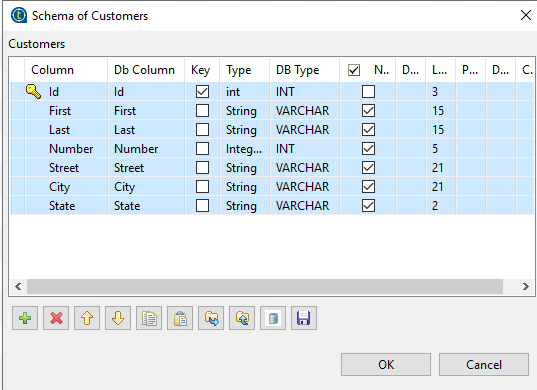

**3.-** Configure el componente para conectarse a la base de **training** y realice una consulta(query) a la tabla **customers**.

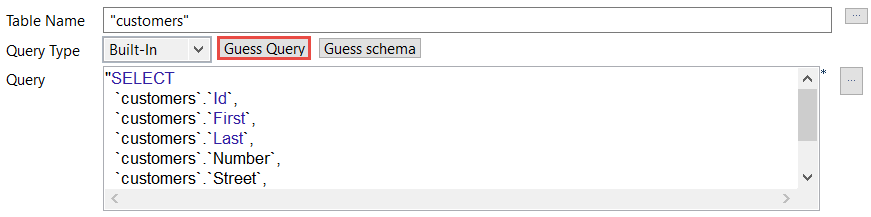
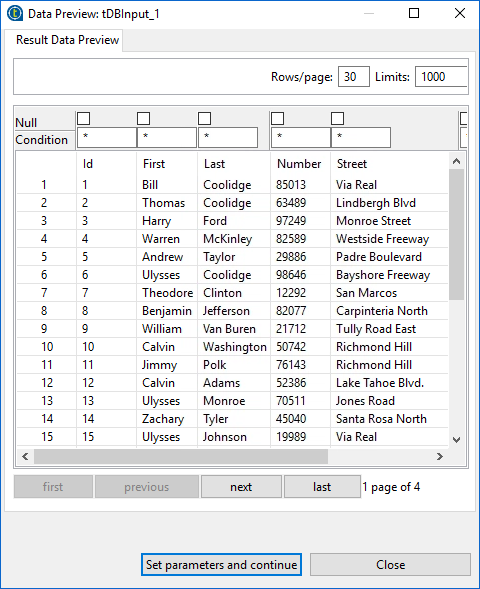
**a.-** Haga click en el componente para configurar el tipo de base de datos. Para **Database**, seleccione **MySQL** en la lista desplegable y presione **Apply**.  
  
  
 **b.-** Para **DB Version**, seleccione **Mysql 5**.  
  
 **c.-** Para **Host** ingrese *"localhost"*. Para **Port** ingrese *"3306"*, y para **Database** ingrese *"training"*.  
  
  
  
 **d.-** Para **Username** ingrese *"talend"*.  
  
 **e.-** Para **Password** haga click en [**...**] e ingrese *"talend"*. Las comillas dobles son obligatorias porque Talend espera que el valor de este campo sea un string Java. Presione **OK**.  
  
  
**f.-** Para **Table Name**, ingrese *"customers"*.  
  
  
  
**4.-** Renombre el componente **tDBInput** a *Customers*.  
  
  
**5.-** Copie la definición del schema del Job **createTable**.  
  
 **a.-** Abra el Job **createTable**.  
  
 **b.-** Haga click en el componente **CustomersTable**.  
  
 **c.-** En la pestaña Componente, a la derecha de **Edit schema**, presione el botón [**...**].  
  
 **d.-** En el lado derecho de la ventana, haga clic a la izquierda de la fila **Id** para seleccionarla, y observe la flecha.  


**e.-** Mientras presiona el botón izquierdo del mouse, deslice la flecha hacia abajo para seleccionar todas las filas.  
  
  
  
 **f.-** En la barra de herramientas, haga clic en **Copy selected items**. El schema se selecciona y se puede pegar en otro editor de schemas.  


**6.-** Vuelva al Job **readTable** para configurar el schema de la tabla **customers**.  
  
 **a.-** Haga click en el componente **Customers** para abrir la ventana component.  
  
 **b.-** A la derecha de **Edit schema**, haga click en el botón [**...**].  


**c.-** Para reutilizar el schema que ha copiado anteriormente, haga click en el ícono **Paste** ubicado al final de la ventana.  


**d.-** Confirme que el schema se ha actualizado correctamente, debe verse así:  


**7.-** Para actualizar la query haga click en el botón **Guess Query**.  
  
El schema del componente se utilizará para crear la query.  
  
**8.-** Utilice el data viewer del componente Customers para obtener una vista previa de los datos extraídos de la tabla.  


## Use SQL Builder para personalizar una query

SQL Builder le permite editar y validar queries. En este ejercicio deberás personalizar una query SELECT para filtrar sólo customers con el mismo State "CA":

*SELECT*

*`customers`.`Id`,*

*`customers`.`First`,*

*`customers`.`Last`,*

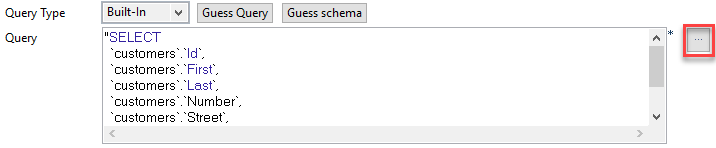
*`customers`.`Number`,*

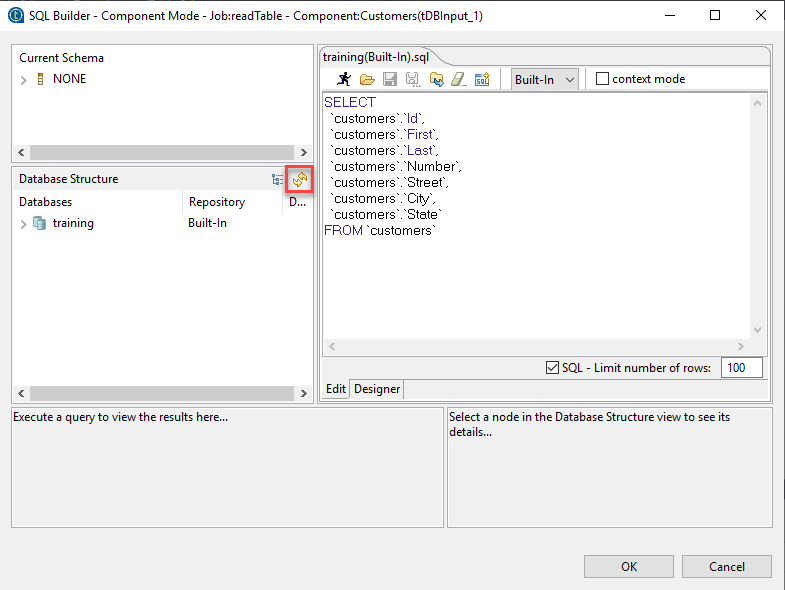
*`customers`.`Street`,*

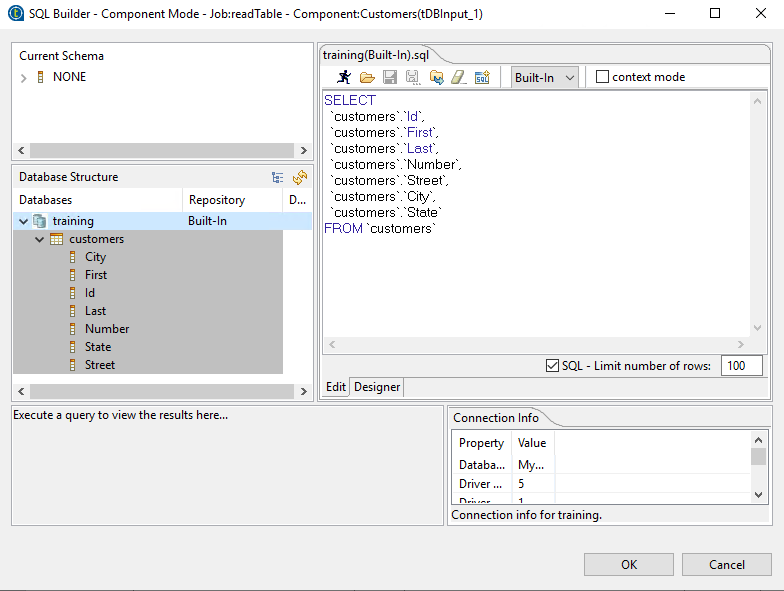
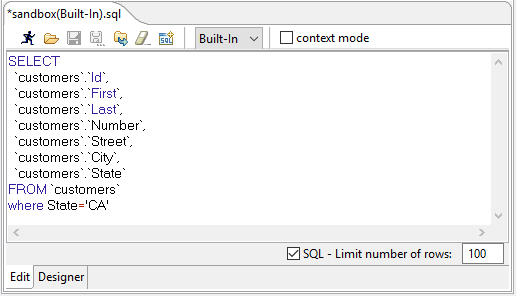
*`customers`.`City`,*

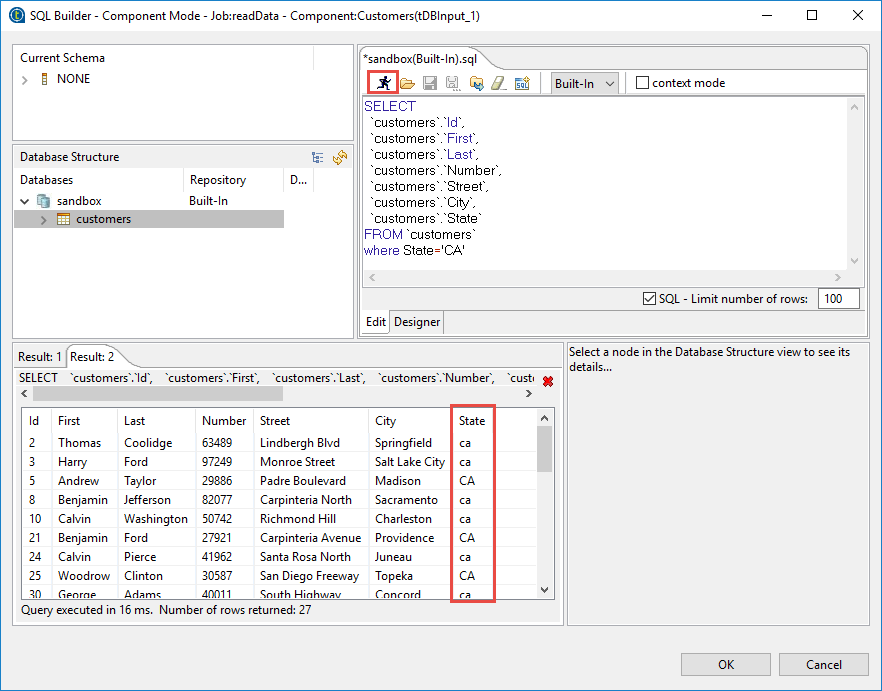
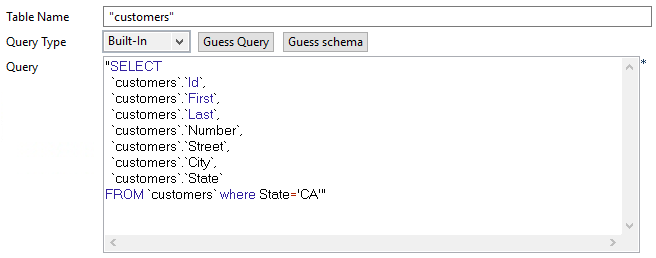
*`customers`.`State`*

*FROM `customers` where State= 'CA'*

**1.-** En el componente **Customers**, use **SQL Builder** para editar la query previamente definida.  
  
 **a.-** A la derecha de **Query**, seleccione [**...**] para abrir SQL Builder.  


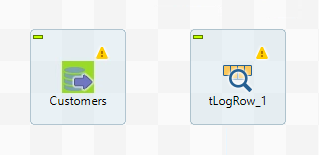
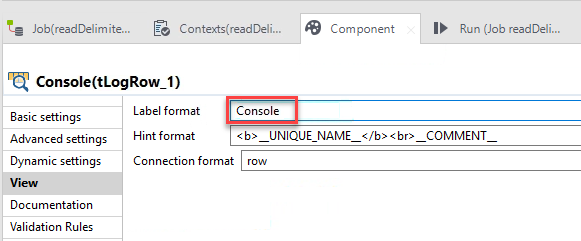
💡 Es posible que deba hacer clic aquí nuevamente si es la primera vez que inicia el editor.  
  
 **b.-** Presione el botón **Refresh**, para actualizar **Database Structure**.  


**c.-** Expanda **training** para ver la lista de tablas, y luego expanda la tabla **customers** para ver la lista de columnas.  
  
  
 **d.-** En el SQL editor, haga click al final de la query SELECT.  
  
 **e.-** Ingrese *where* y un espacio en blanco. Se mostrará una lista con las columnas de la tabla customers.  
  
 **f.-** Seleccione **State** y complete la query como se muestra a continuación:  


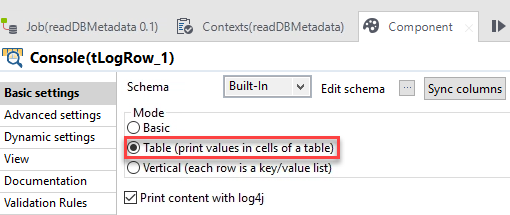
**g.-** Haga click en el ícono **Execute SQL** para ejecutar la query y chequear que el resultado sea correcto en la pestaña **Result**. Presione **OK**.  
  
  
  
 **h.-** En la pestaña **Component**, en el cuadro de texto **Query**, confirme que la sentencia SQL se haya actualizado.  


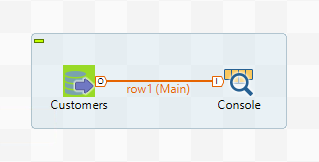
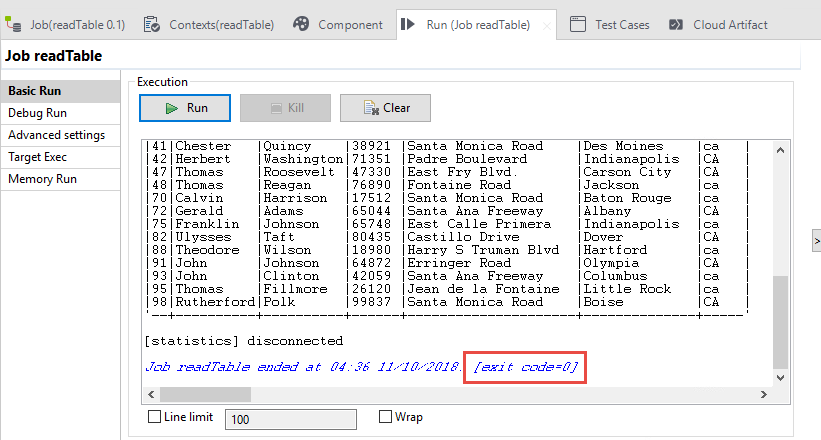
## Imprimir los datos de la tabla por consola

Para imprimir el contenido de la tabla por consola, conecte el componente Customers a un componente tLogRow.

**1.-** En el **Designer**, agregue un nuevo componente **tLogRow** a la derecha del componente **Customers**.  
  
 **2.-** Cambie el nombre del componente **tLogRow** a *Console*.  


**3.-** Configure el componente **tLogRow** para imprimir los resultados en una tabla.  
  
 **a.-** Abra la pestaña **Basic settings** en la ventana **Component**.  
  
 **b.-** Para imprimir los resultados en una tabla, en **Mode**, seleccione **Table**.

* 

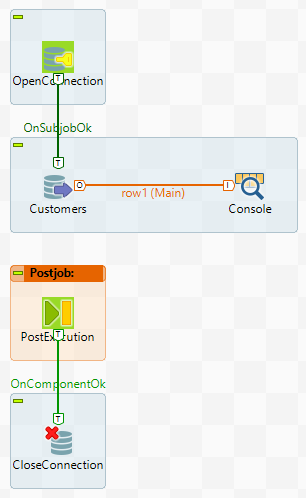
**4.-** Conecte los componentes **Customers** y **Console** usando una **Main** row.  
  
  
**5.-** Ejecuta el Job y analiza los resultados.  
  
 **a.-** En la ventana **Run**, presiona el botón **Run**. Se ejecuta el Job y se muestra en la consola la tabla que contiene la lista de clientes extraída del archivo.  


b.- Chequee que el log finalice con **[exit code=0]** y verifique que solo se enviaron clientes de California a la consola.

**Tema 4: Buenas prácticas**

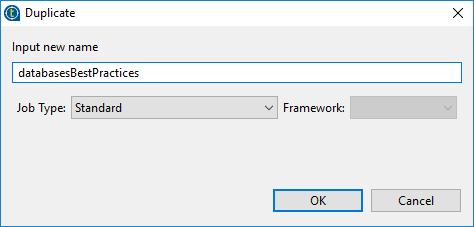
## Para comenzar

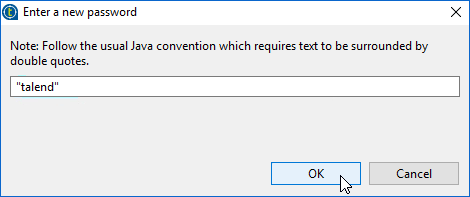
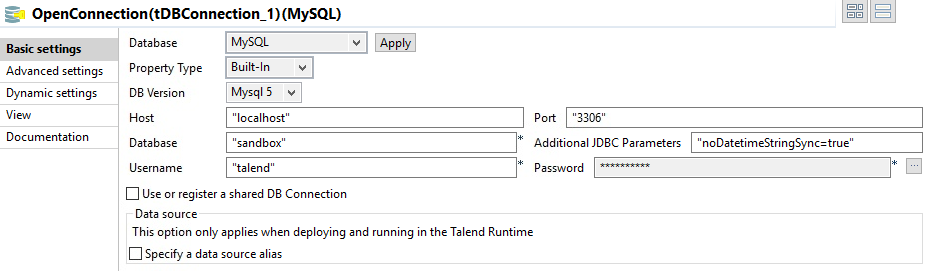
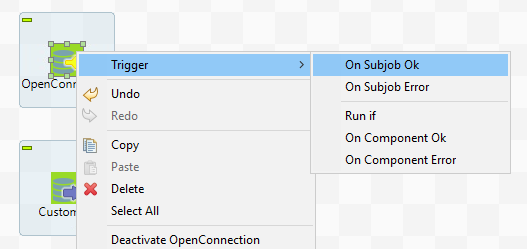
Este ejercicio le enseña cómo aplicar las mejores prácticas para guardar conexiones simultáneas a una base de datos. Duplicas tu Job más reciente y configuras una conexión a la base de datos de sandbox. Reutiliza esta conexión para configurar otro componente y cierra la conexión cuando finalice el Job.  
  
Al completar el ejercicio, su Job debería verse así:



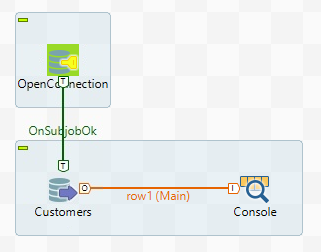
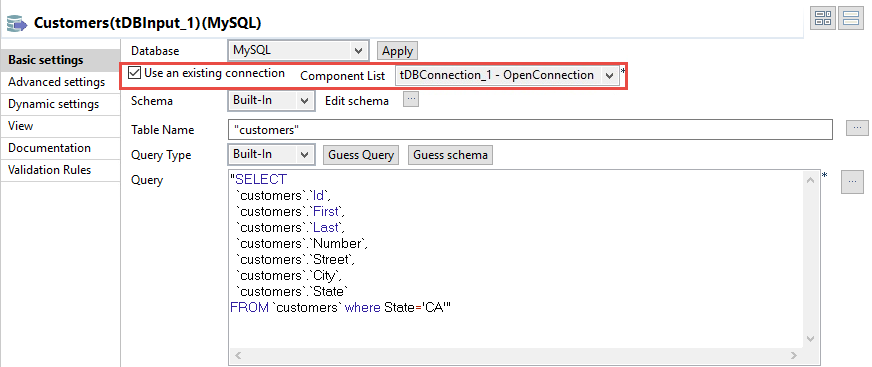
## Compartir una conexión de base de datos

En este ejercicio aprenderá a aplicar buenas prácticas en el Job readTable que acaba de crear. El primer paso es duplicar el Job y usar un componente tDBConnection para configurar una conexión a la base de datos de sandbox. Luego configura el componente tDBInput para usar esta conexión en lugar de los parámetros integrados (built-in).

**1.-** Duplique el Job **readTable** y nómbrelo *databasesBestPractices*.  
  
 **a.-** En el **Repository**, haga click derecho en el Joc **readTable** y seleccione **Duplicate**.  
  
 **b.-** En el cuadro de texto **Input new name**, ingrese *databasesBestPractices*.  
  
  
 **c.-** Presione **OK** para duplicar el Job. Luego abra **databasesBestPractices**.  
 **2.-** Configure la conexión.  
  
En el **Designer**, coloque un componente **tDBConnection** encima de **tDBInput** y conéctelo a la base de datos training.  
  
 **a.-** Agregue un **tDBConnection**.  
  
 **b.-** Haga doble click encima para abrir la ventana **Component** y utilice los siguientes parámetros para configurar el componente:  
  
 **→** Para **Database**, seleccione **MySQL** y presione **Apply**.  
 **→** Para **DB Version**, seleccione **Mysql 5**.  
 **→** Para **Host**, ingrese *"localhost"*. Para **Port** ingrese *"3306"*, y para **Database** ingrese *"training"*.  
 **→** Para **Username**, ingrese *"talend"*.  
 **→** Para **Password**, haga click en [**...**] e ingrese *"talend"*. Presione **OK**.

  
  
 **c.-** Para cambiar el nombre del componente, haga click en su nombre y reemplácelo con *OpenConnection*.   
  
El componente debería verse así:  
  
 **d.-** Conecta **tDBConnection** con **tDBInput** usando Trigger → **On Subjob OK**.  


💡 Siguiendo las buenas prácticas de diseño, los subJobs deben organizarse de arriba a abajo.

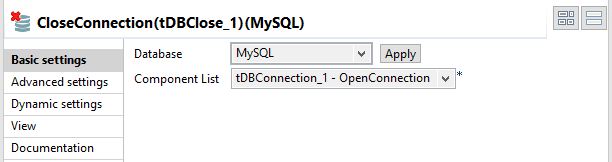
El Job debería verse así:  
  
  
**3.-** Configure **tDBInput** para utilizar una conexión existente.  
  
 **a.-** Haga doble click en **Customers** para abrir la ventana **Component**.  
  
 **b.-** Seleccione **Use an existing connection**.  
  
 La información de conexión se oculta y aparece la lista de componentes.  
  
 **c.-** En **Component List**, seleccione **tDB\_Connection\_1 - OpenConnection**.  
  
 El componente debería verse así:  


## 

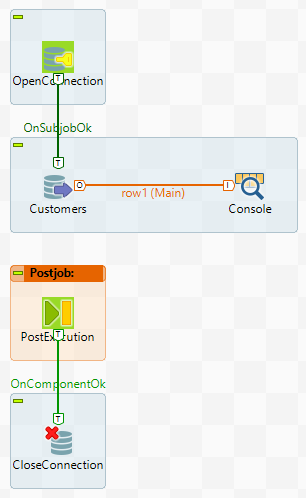
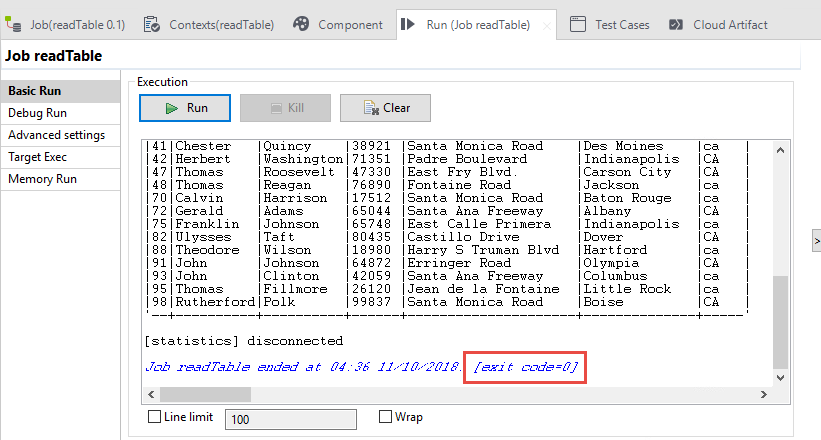
## 

## Cerrar una conexión de base de datos

Siguiendo las buenas prácticas cuando trabaje con bases de datos, debe cerrar la conexión utilizando un componente **tDBClose**. Para cerrar la conexión incluso si el Job falla, se debe llamar a **tDBClose** desde un componente **tPostJob**.

**1.-** Cierra la conexión.  
  
Coloque un componente **tPostJob** y un componente **tDBClose** debajo de los otros componentes y conéctelos.  
  
 **a.-** Coloque un componente **tPostJob** debajo del componente **tDBInput**. Nómbrelo *PostExecution*.  
  
 **b.-** Coloque un componente **tDBClose** debajo del componente **tPostJob**. Nómbrelo *CloseConnection*.  
  
 **c.-** Haga doble click en **tDBClose** para abrir la ventana **Component** y configure el componente de la siguiente manera:  
  
 **→** Para **Database**, seleccione **MySQL** y presione **Apply**.  
  
 **→** En **Component List**, seleccione **tDB\_Connection\_1 - OpenConnection**.  
  
 El componente debería verse de esta forma:  


**d.-** Para conectar **tPostJob** con **tDBClose**, use trigger → **On Component Ok**.  
  
  
💡 No necesita conectar tPostJob con el subJob anterior. Los componentes vinculados con tPostJob se ejecutan automáticamente cuando finaliza el Job, incluso si falla.

El Job debería verse así:  
  
  
**2.-** Ejecuta el Job y analiza los resultados.  
  
 El Job debería finalizar con **[exit code=0]**.

## Resumen

En este módulo aprendiste como:

* Conectarte a una base de datos desde un Job Talend
* Escribir en una tabla MySQL
* Leer datos de una tabla MySQL
* Usar SQL Builder para personalizar una query
* Compartir una conexión de base de datos
* Cerrar la conexión cuando termine el Job