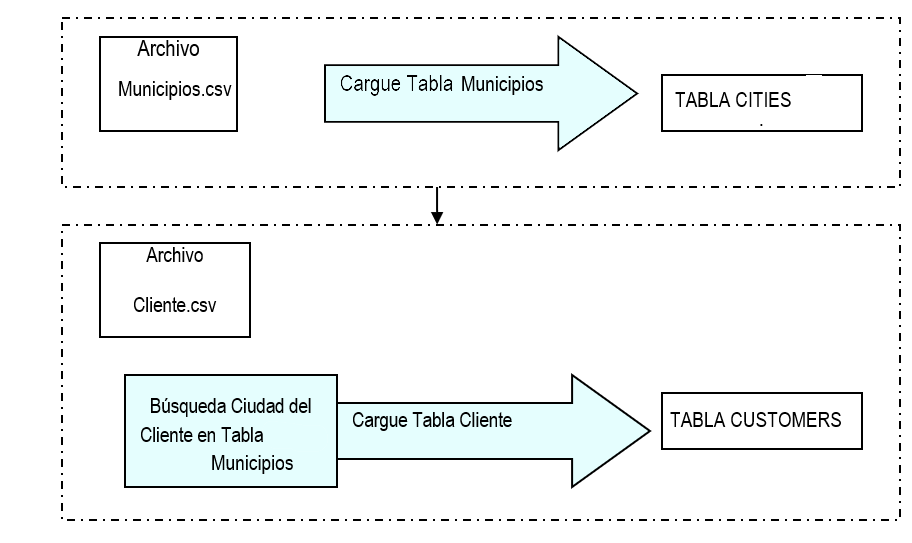
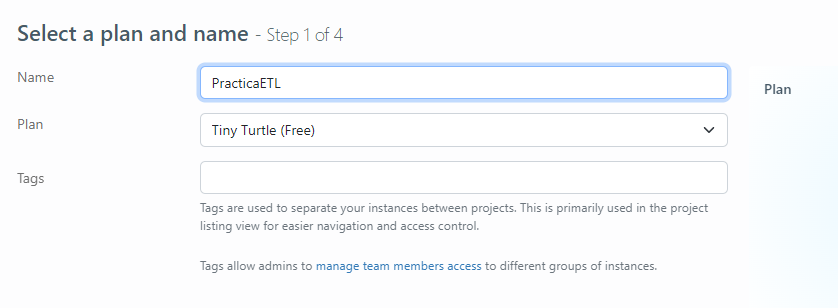
**EJERCICIO 1**

Descripción: Se tienen dos archivos, municipios.csv y cliente.csv, la práctica consiste en cargar el archivo municipio en la Tabla CITIES y cargar el Cliente.csv en la tabla CUSTOMERS, con el nombre del municipio, en vez de su código.

El archivo municipio fue tomado de Datos abiertos: Departamentos y municipios de Colombia <https://www.datos.gov.co/Mapas-Nacionales/Departamentos-y-municipios-de-Colombia/xdk5-pm3f>

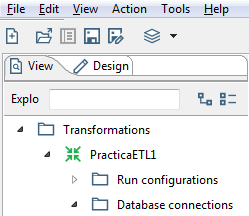


1. Cree en el Disco Duro una carpeta PracticaETL, donde va a guardar los archivos de las transformaciones y los archivos de la práctica.
2. Descargue los archivos de la práctica de esta carpeta
3. Para crear la nueva transformación, en Spoon, en el Menú ***File*** *>* ***New*** *>* ***Transformat*ion**
4. Vamos a guardarlo para que quede de una vez en una ubicación definida. ***File***> ***Save as***, y colóquele el nombre: PracticaETL1
5. Antes de continuar, debe crear en Postgresql, la base de datos donde se van a guardar los datos. Base de datos: PracticaETL

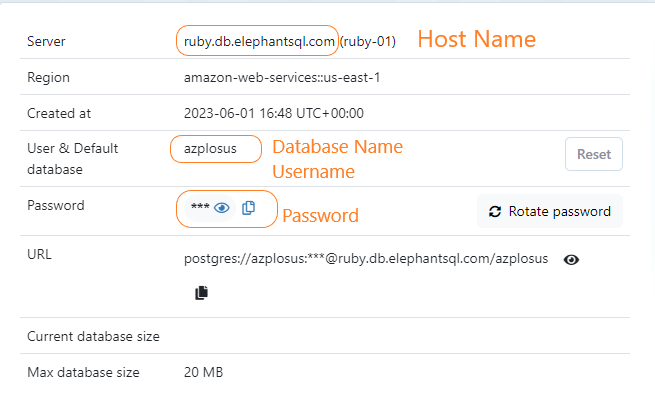


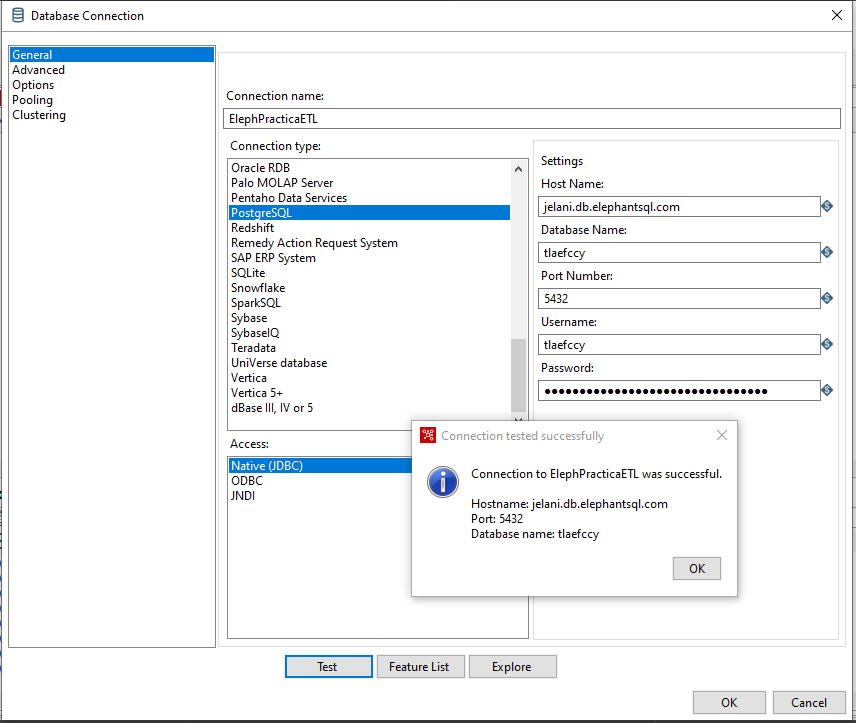
**Configurar la conexión a la base de datos.**

1. Seleccionar de la barra de herramientas de la Izquierda, View, en **Database connections** la opción **New**.

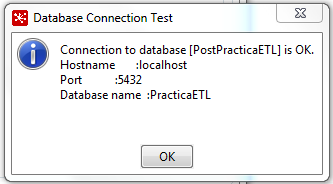


1. Configurar la conexión, en nuestro caso, Postgresql (Native JDBC). Colóquele un nombre y los datos solicitados de acuerdo con la información de la instancia en ElephantSQL. Tenga en cuenta que en Database Name y Username, se utiliza el mismo dato.





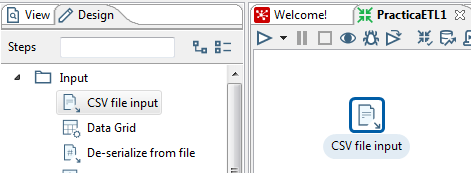
1. Se le da Test, para comprobar que la conexión está correcta.



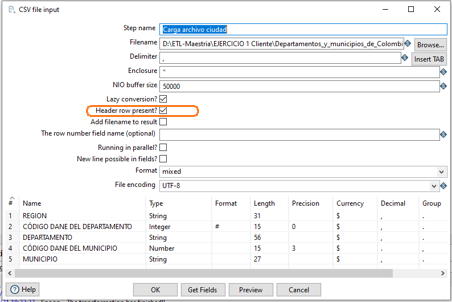
1. Y OK. Ya queda creada la conexión.

**Para agregar la entrada desde archivos planos**

1. De la barra de herramientas de la izquierda (**Design**), arrastre de Input, el elemento **CSV file input**



1. Al darle doble click, aparecen las propiedades, cambie el nombre a **Carga archivo ciudad**.
2. Haga clic en **Browse**, para encontrar la ubicación del archivo, busque la carpeta de datos de la práctica y abra el archivo **municipio.csv**. Presione Add.
3. Defina el delimitador del archivo, en este caso es Tabulador (entonces de click en **Insert TAB**).
4. Como la primera fila contiene los nombres de las columnas, seleccione la opción **Header row present**.
5. Al hacer Clic en **Get Fields**, muestra un resumen de las filas que se cargaron. Especificar 1000 líneas. File encoding: UTF-8 para garantizar manejo de tildes y ñs.



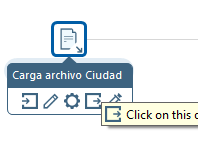
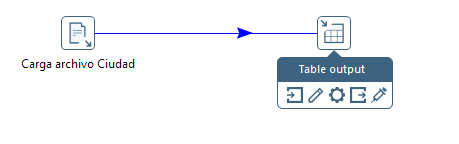
1. Y clic en OK.

En este momento ya esté lista la conexión al archivo **municipios.csv**, realice el mismo procedimiento anterior para realizar la conexión para el archivo **cliente.csv**. Revise el delimitador con que están los datos en el archivo.

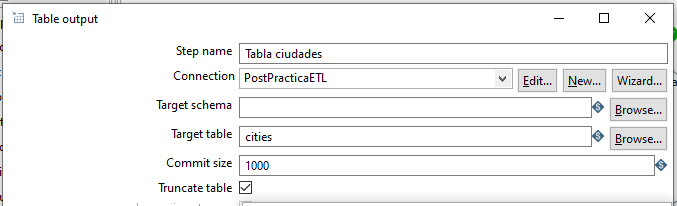
Ahora vamos a adicionar la conexión a la base de datos.

**Para agregar y configurar una conexión a la BD**

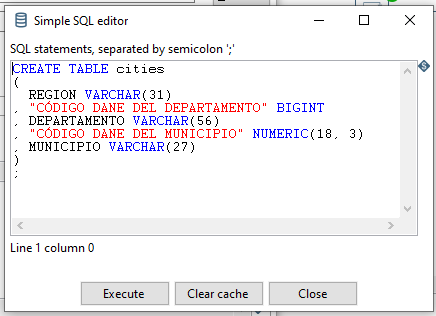
1. De la ventana de herramientas del lado izquierdo, escoja de **Output**, **Table Output** y suéltelo en el área de trabajo.
2. Haga clic sobre **Carga archivo ciudad** y aparecen unas opciones. Escoja la que se muestra en la siguiente imagen, aparece una flecha, suéltala en el elemento **Tabla Output**. Escogiendo la opción **Main output of step.**

1. Haga doble click sobre Tabla Output, cambie el nombre a **Tabla Municipios** y seleccione la conexión a la base de datos.



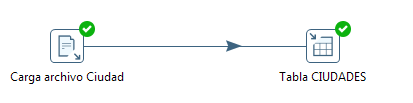
Antes de dar OK, presione Click en SQL, y aparece el código SQL para crear la Tabla. Execute y se creará la tabla.



En este punto, ya está listo para ejecutarse y probarse la carga de los datos en la Tabla **CITIES**.

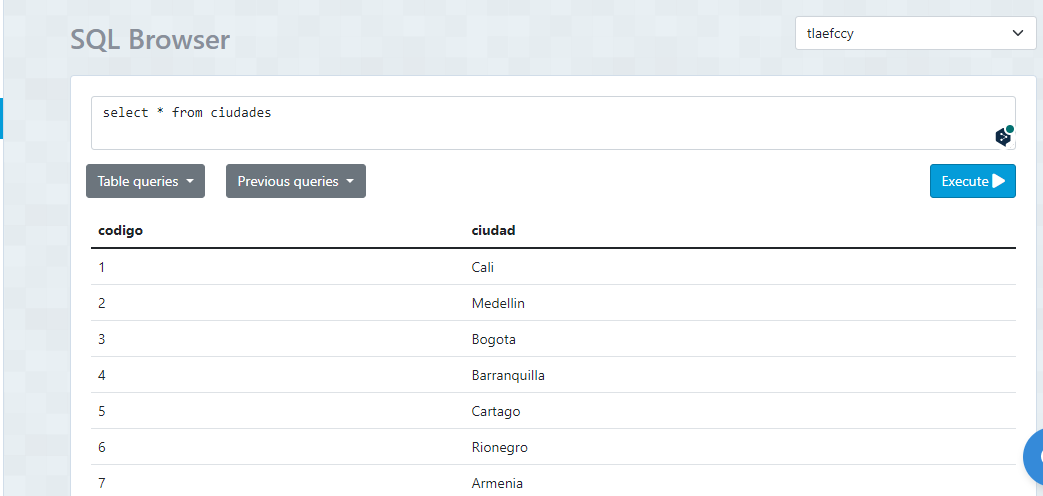
**Ejecutar trabajo o transformación:**

1. Haga clic en  para ejecutar, y **Run**.
2. Luego de correr, deben quedar las señales de que corrió correctamente. Y al verificar la tabla en la BD, deben estar todos los datos. Verifíquelo.



**Revisión en Postgresql:**

Luego de ejecutar la transformación, se puede revisar la base de datos creada previamente (PracticaETL).



Continuemos con la carga del archivo Clientes:

Transformaciones:

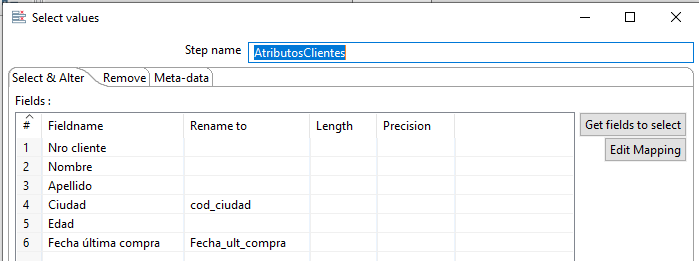
* Cambiar nombre de atributos
* Join de clientes con ciudades, para mostrar el nombre del municipio.

**Cambiar el nombre de los atributos:**

1. Seleccione el elemento Select Values del menú Transform .
2. Arrastrar de Carga archivo Clientes a Select Values, escogiendo la opción **Main output of step.**
3. Dar doble clic sobre Select Values, cambiar el nombre a **AtributosClientes** vamos a cambiar el nombre de algunos atributos:

Fecha última compra -> Fecha\_ult\_compra

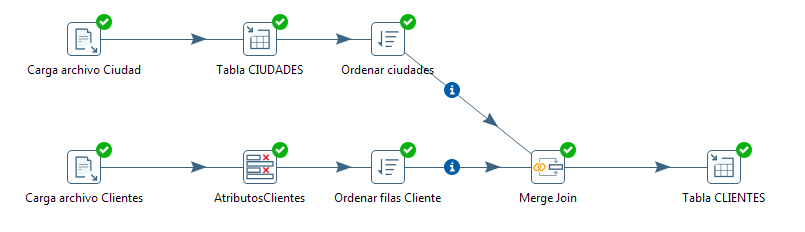
Ciudad -> cod\_ciudad



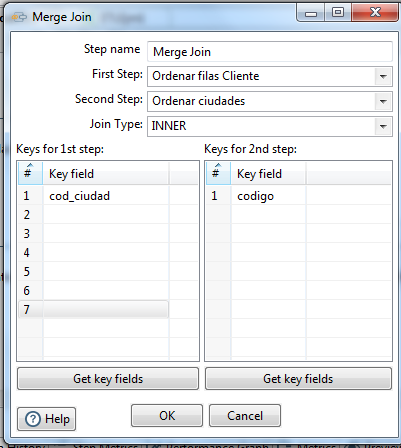
**Join entre dos flujos de datos:**

1. Seleccione el elemento **Merge Join** del menú **Joins**, arrástralo al área de trabajo. 
2. Antes de llevar los flujos de datos al join, se recomienda ordenar los flujos, por el atributo que se realizará el join. En este caso, el atributo cod\_ciudad de Clientes y el atributo codigo de Ciudades.

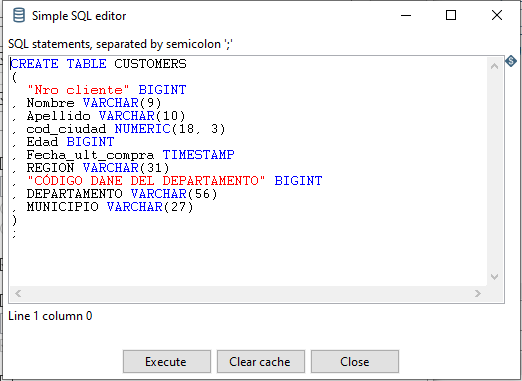
Arrastrar **Sort rows** del menú **Transform**. Arrastrar el flujo de datos de Tabla Ciudades. Y así mismo con el flujo de datos desde **AtributosClientes**.



1. Configure los atributos del Merge Join, por los que se realizará el join.



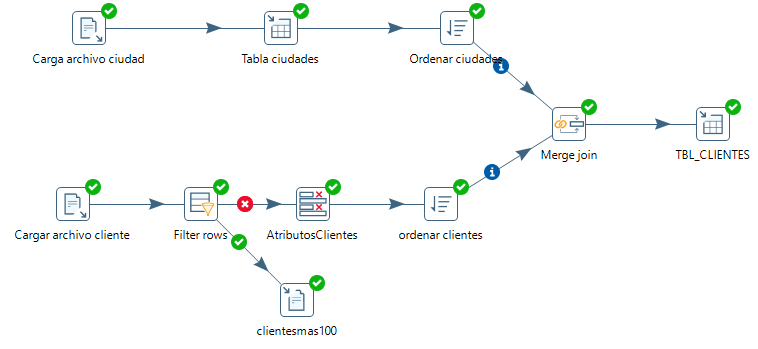
1. Realice la creación de la tabla CUSTOMERS, de forma similar a como se realizó con CITIES. Revise que el código de la ciudad no quede repetido.



1. Haga clic en  para ejecutar, y **Run**.
2. Verifique los datos que queden cargados en las tablas.

**CALIDAD DE DATOS:**

Vamos a adicionar una validación, si se fijan algunos clientes tienen edades mayores a 100 años, lo cual se considera un error, para ello se debe adicionar un paso que verifique la edad y si son mayores a 100, se guarde en un archivo “clientesEdadErrada”.



**REFERENCIAS IMPORTANTES:**

Pentaho DI

<https://wiki.pentaho.com/display/EAIes/.03+Conexiones+a+Bases+de+Datos>

Casters, M., Bouman, R., & Van Dongen, J. (2010). *Pentaho Kettle solutions: building open source ETL solutions with Pentaho Data Integration*. John Wiley & Sons.