

Recuperado de:

<https://www.btelligent.com/en/blog/staging-area-potential-in-comparison-to-source-system/>

# IMPLEMENTACIÓN DE MODELO DIMENSIONAL EN PENTAHO DATA INTEGRATION (PDI)

Por:

Mg. Esther Balbuena

Copyright 2019 Mg. Esther Balbuena Alarcón. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.3 o cualquier otra versión posterior publicada por la Free Software Foundation. Una copia de la licencia se puede encontrar en el siguiente enlace:

<http://www.gnu.org/licenses/fdl-1.3.html>

Fecha	Versión	Autor	Detalle del cambio
14/04/2019	0.1	Mg. Esther Balbuena	Versión Inicial.

Para ver la traducción no oficial de la Licencia de Documentacion Libre de GNU, seguir el siguiente enlace: [http://stuff.danexnow.org/gfdl\\_es.html](http://stuff.danexnow.org/gfdl_es.html)

# IMPLEMENTACIÓN DE MODELO DIMENSIONAL

**Objetivo:** Implementar el modelo dimensional de una datawarehouse usando las bases de datos dimensionales y Pentaho Data Integration para los procesos ETL.

## 1) Requisitos previos

- a) Tener instalado Pentaho Data Integration versión Community.

Descarga desde:

<https://community.hitachivantara.com/docs/DOC-1009931-downloads>

### Pentaho Community Edition 8.2

#### Business Analytics Platform

Pentaho's simplified, and interactive approach empowers business users to access, discover and blend any data types regardless of their size.

All OS

#### Data Integration

Pentaho's Data Integration, also known as Kettle, delivers powerful extraction, transformation, and loading (ETL) capabilities.

All OS

#### Report Designer

The Report Designer is a graphical tool that generates reports from data streamed through the Data Integration engine.

Windows / Linux

Una vez descargado el archivo, procedemos a descomprimirlo en algún lugar apropiado (sugerido: C:\)

- b) Tener instalado MySQL Server y Workbench

Descargar desde el siguiente enlace:

<https://dev.mysql.com/downloads/mysql/>

Instalar MySQL Server y MySQL Workbench para la versión de Sistema Operativo que cuente en su máquina.

- c) Configuración:

- *Descargar de:* <https://dev.mysql.com/downloads/connector/j/>
- *Descomprimir el archivo zip* (in my case it was **mysql-connector-java-5.1.31.zip**)
- *Copiar el archivo .jar* (**mysql-connector-java-5.1.45.-bin.jar**) y pegarlo en el folder **Lib** del PDI:
- **PC:** C:\pdi-ce-8.2.0.0-342\data-integration\lib , o la dirección donde se descomprimió PDI
- **Mac:** /Applications/data-integration/lib

- d) Reconocer la estructura de datos de la fuente de datos fichero\_book.csv

```
"user_code"/"user"/user_id/"producto"/nota/texto/resaltar/subrayar/tachar/manoalzada/rectangulo/elipse/"nive  
"000007b7-a199-47a4-8288-441880f97a11"/"Usuario Usuario 55000"/55000/"ESA172025"/1/0/0/0/0/0/1/"Secundaria  
e historia. 4 ESO. Savia"/"Alumno"/"Castilla-La Mancha"  
"0000c53a-899d-4051-a579-e74cfd5e4d1f"/"Usuario Usuario 55001"/55001/"ESA146820"/0/0/0/0/0/0/0/"Secundaria
```

- 2) **Creación del staging area** para guardar los datos desde la fuente de datos.
- Se debe tener el servicio de MySQL arriba.
  - Abrir MySQL Workbench y crear la Base de Datos stg\_apellido (en mi caso stg\_ebalbuena)
  - Crear la tabla f\_salida que almacenará los datos que vienen de la fuente de datos fichero\_book.csv



```
CREATE DATABASE IF NOT EXISTS `stg_ebalbuena` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `stg_ebalbuena`;

-- Table structure for table `f_salida`

DROP TABLE IF EXISTS `f_salida`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `f_salida` (
  `user_code` varchar(100) NOT NULL,
  `user` varchar(100) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
  `producto` varchar(100) DEFAULT NULL,
  `nota` int(11) DEFAULT NULL,
  `texto` int(11) DEFAULT NULL,
  `resaltar` int(11) DEFAULT NULL,
  `subrayar` int(11) DEFAULT NULL,
  `tachar` int(11) DEFAULT NULL,
  `manoalzada` int(11) DEFAULT NULL,
  `rectangulo` int(11) DEFAULT NULL,
  `elipse` int(11) DEFAULT NULL,
  `nivel` varchar(100) DEFAULT NULL,
  `curso` varchar(100) DEFAULT NULL,
  `asignatura` varchar(100) DEFAULT NULL,
  `titulo` varchar(250) DEFAULT NULL,
  `rol` varchar(100) DEFAULT NULL,
  `zona` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='tabla f_salida';
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

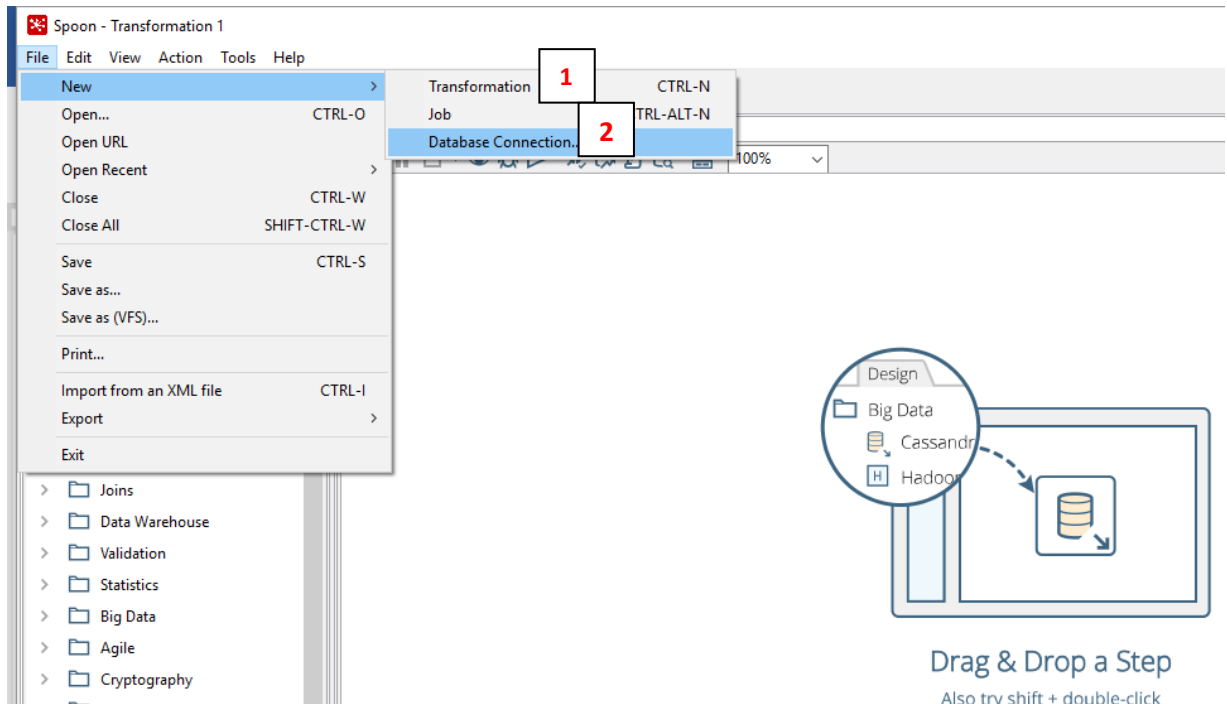
### 3) Creación de una conexión de Pentaho Data Integration y MySQL.

- a) Iniciar Pentaho Data Integration ejecutando:

 Spoon.bat	14/11/2018 17:21	Archivo por lotes ...	5 KB
 spoon.command	14/11/2018 17:21	Archivo COMMA...	2 KB
 spoon.ico	14/11/2018 17:21	Icono	362 KB

- b) Crear una conexión a base de datos:

1. Crear nueva transformación
2. Crear nueva conexión a base de datos

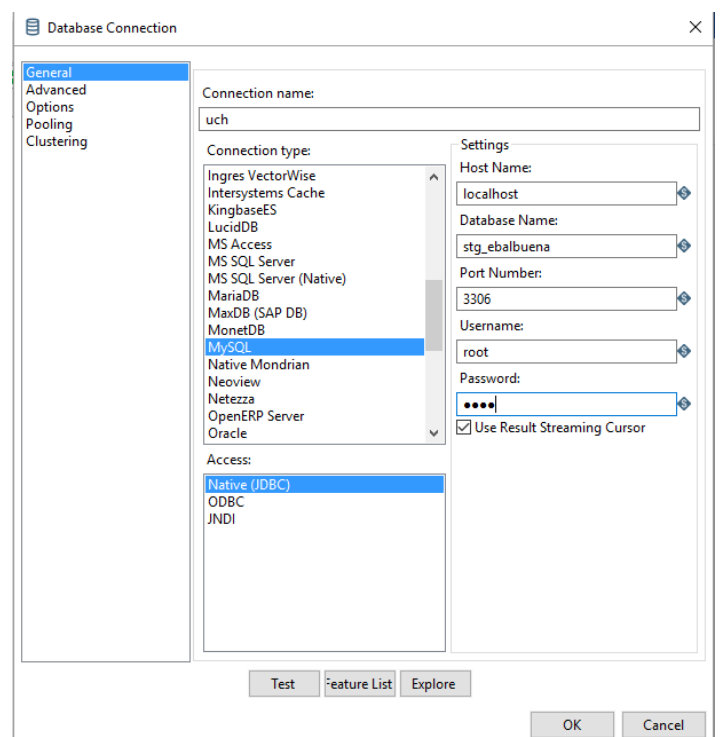
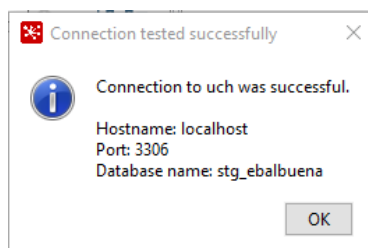


Configurar la conexión:

Parámetros:

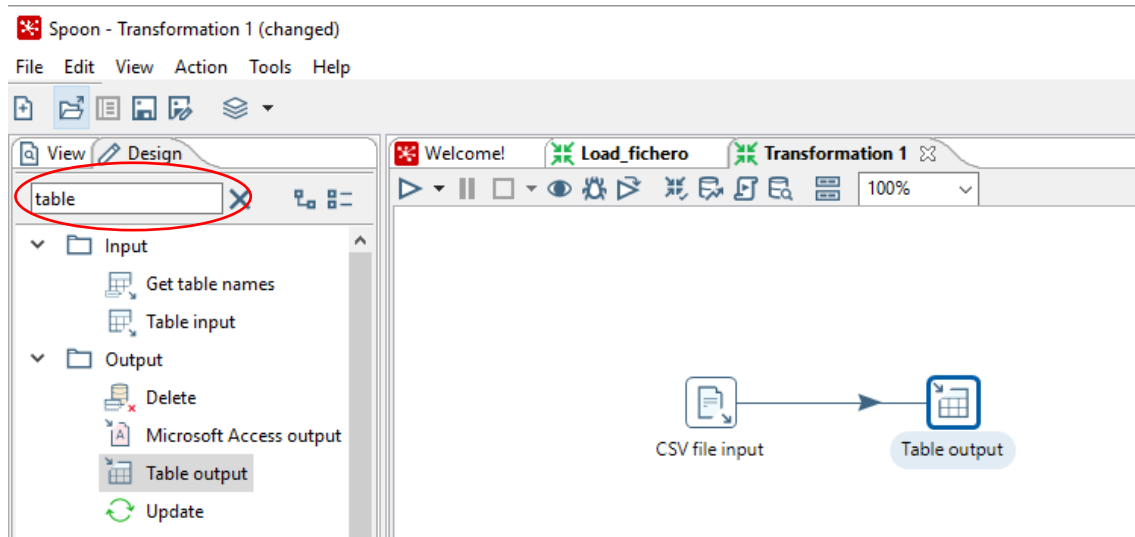
- ✓ *Connection name: uch*
- ✓ *Connection type: MySQL*
- ✓ *Access: Native (JDBC)*
- ✓ *Host name: localhost*
- ✓ *Database name: stg\_ebalbuena*
- ✓ *Port number: 3306*
- ✓ *User name: root*
- ✓ *Password: root*

c) Presionar botón test:

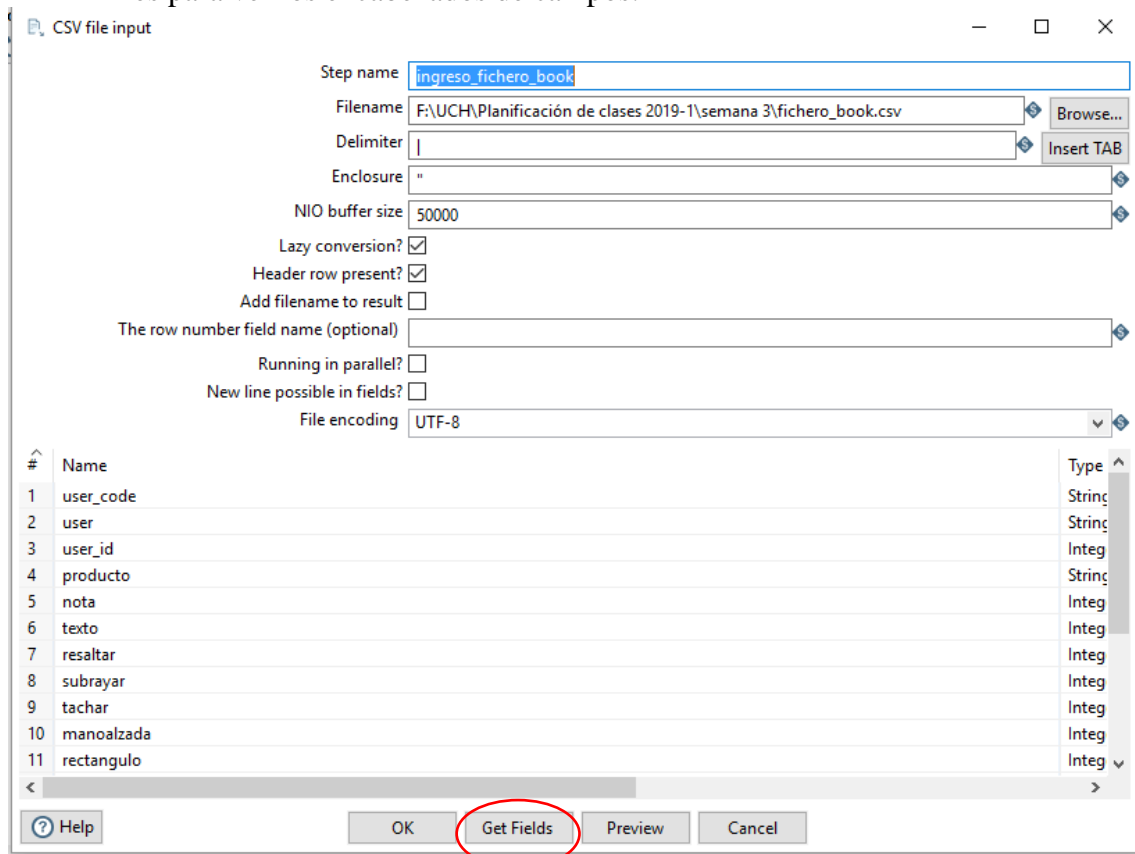


#### 4) Cargar los datos desde la fuente de datos al staging área.

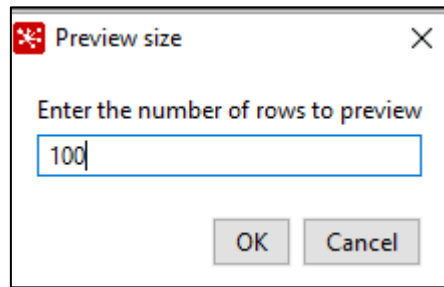
Usaremos dos pasos: CSV file input y Table output. Puede usar el buscador para hallarlos con facilidad



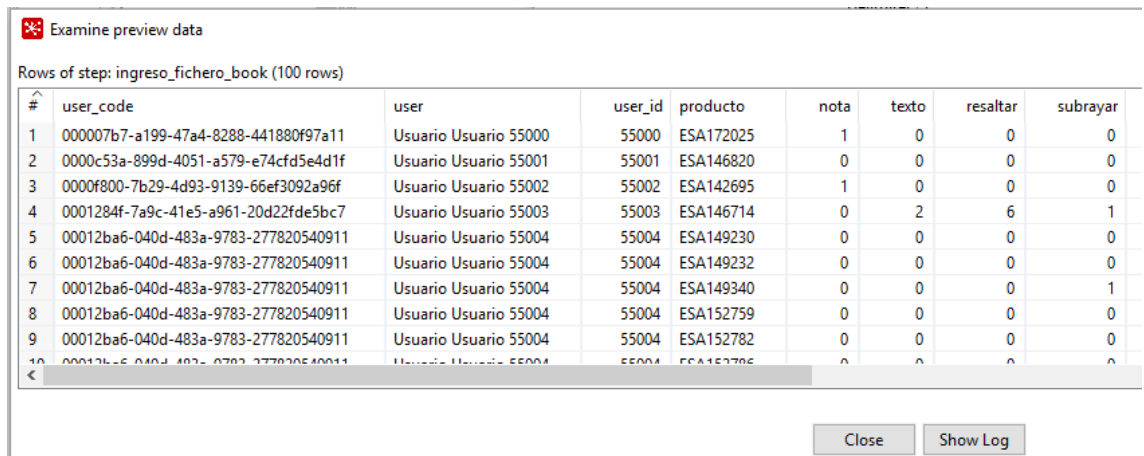
- a) Configuramos el paso CSV file input. Le damos doble click al paso CSV file input y aparecerá la siguiente ventana donde completamos la dirección de la fuente de datos, es decir, la dirección de fichero\_book, completamos el delimitador de campos que en este caso es | y con qué carácter están cerrados los campos. Escogemos UTF-8 como codificador de archivo. Le damos click a Get fields para ver los encabezados de campos:



También podemos tener una vista previa de los datos presionando el botón Preview en la pantalla anterior:



A dialog box titled "Preview size" with a close button (X) in the top right corner. It contains a text input field with the value "100" and two buttons at the bottom: "OK" and "Cancel".

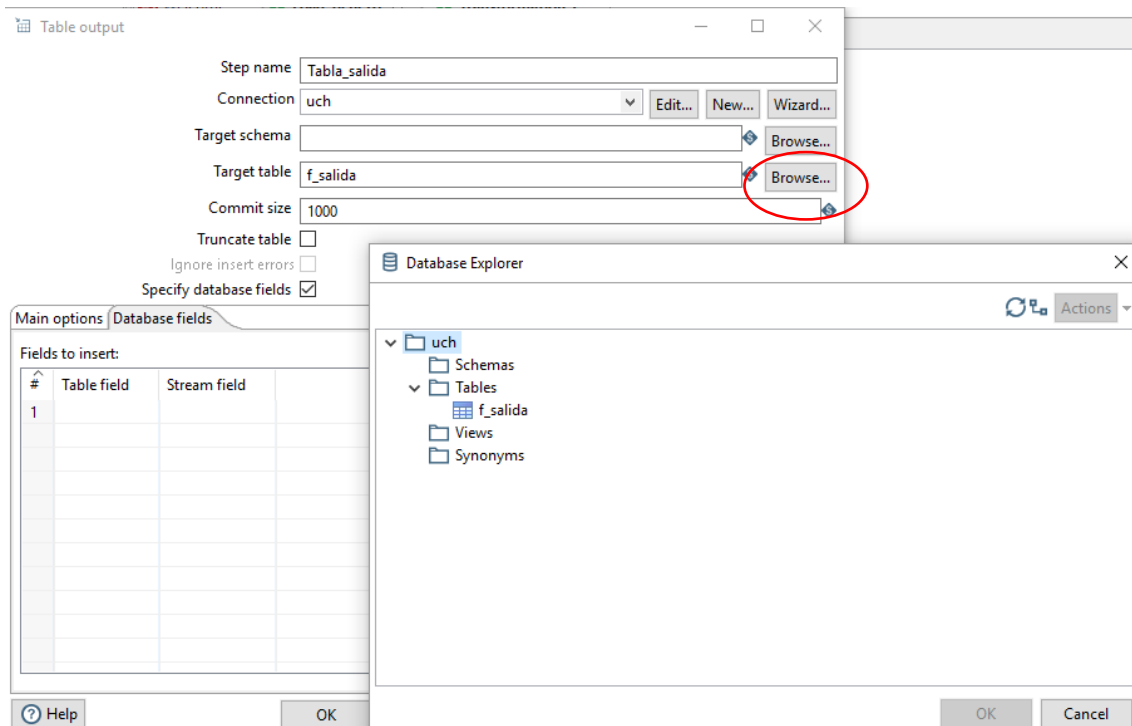


A window titled "Examine preview data" showing a table of data. The title bar includes a close button (X). Below the title, it says "Rows of step: ingreso\_fichero\_book (100 rows)". The table has 9 columns: #, user\_code, user, user\_id, producto, nota, texto, resaltar, and subrayar. The data is displayed in a scrollable area, showing rows 1 through 10. At the bottom right, there are two buttons: "Close" and "Show Log".

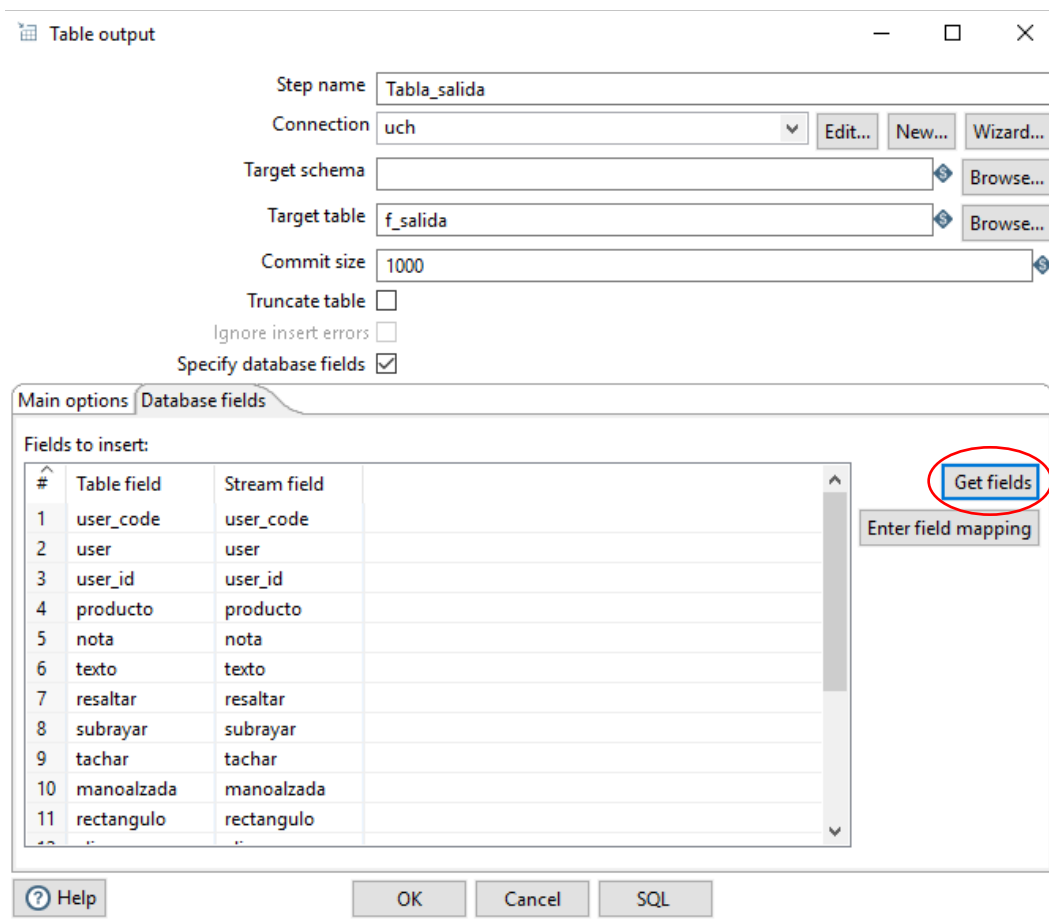
#	user_code	user	user_id	producto	nota	texto	resaltar	subrayar
1	000007b7-a199-47a4-8288-441880f97a11	Usuario Usuario 55000	55000	ESA172025	1	0	0	0
2	0000c53a-899d-4051-a579-e74cfd5e4d1f	Usuario Usuario 55001	55001	ESA146820	0	0	0	0
3	0000f800-7b29-4d93-9139-66ef3092a96f	Usuario Usuario 55002	55002	ESA142695	1	0	0	0
4	0001284f-7a9c-41e5-a961-20d22fde5bc7	Usuario Usuario 55003	55003	ESA146714	0	2	6	1
5	00012ba6-040d-483a-9783-277820540911	Usuario Usuario 55004	55004	ESA149230	0	0	0	0
6	00012ba6-040d-483a-9783-277820540911	Usuario Usuario 55004	55004	ESA149232	0	0	0	0
7	00012ba6-040d-483a-9783-277820540911	Usuario Usuario 55004	55004	ESA149340	0	0	0	1
8	00012ba6-040d-483a-9783-277820540911	Usuario Usuario 55004	55004	ESA152759	0	0	0	0
9	00012ba6-040d-483a-9783-277820540911	Usuario Usuario 55004	55004	ESA152782	0	0	0	0
10	00012ba6-040d-483a-9783-277820540911	Usuario Usuario 55004	55004	ESA152786	0	0	0	0

Cerramos esta pantalla y le damos Ok a la pantalla emergente principal.

- b) Configuramos el paso Table output. Le damos doble click al Table output y aparecerá la siguiente ventana donde elegiremos la conexión uch que se configuró en un paso anterior. El target table lo buscamos en la base datos MySQL que es donde está el staging área dando click al botón Browse.

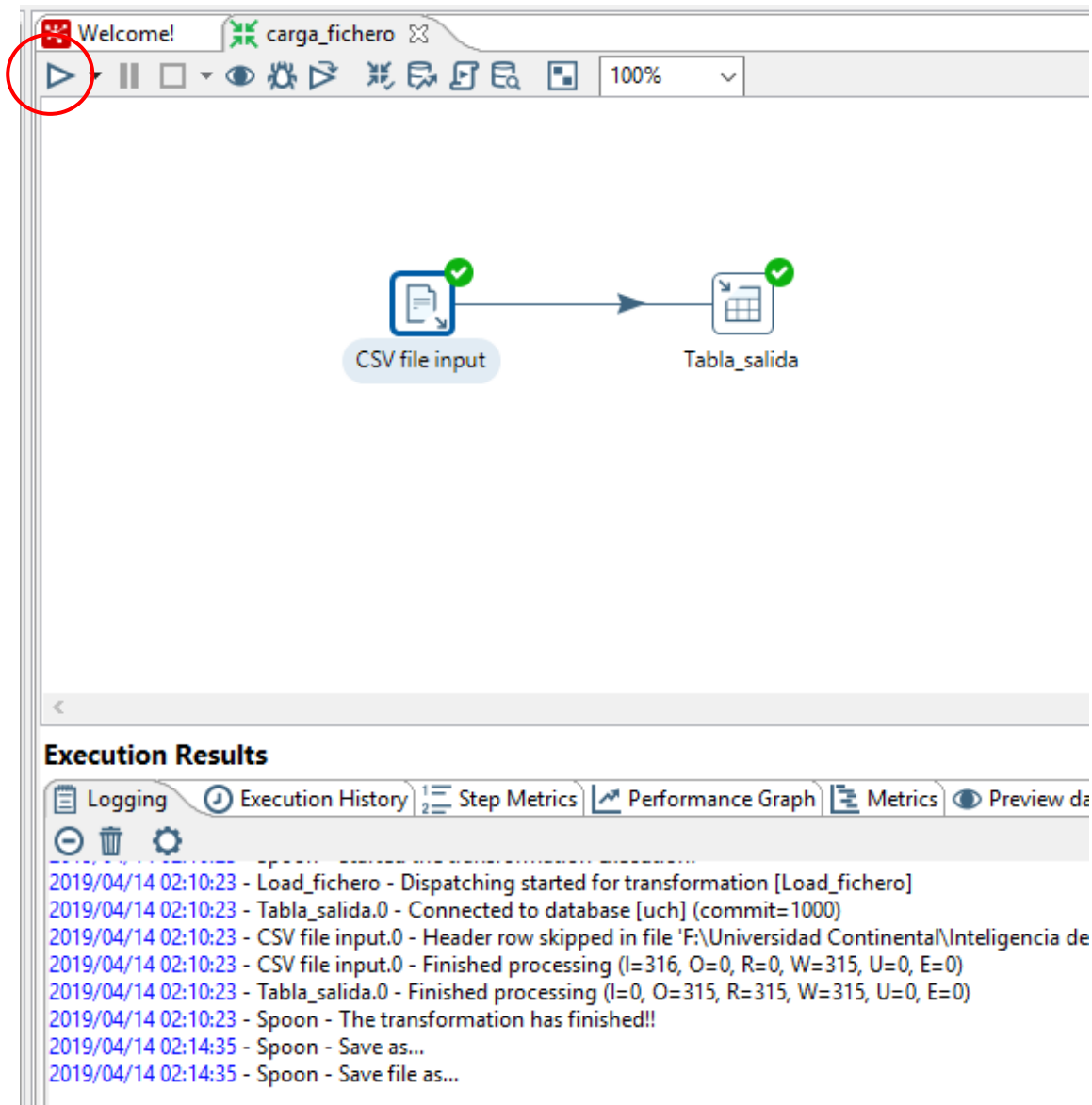


Presionamos el botón get filed para obtener todos los campos de la entrada y que pasarán a la tabla del staging en MySql.





- c) Ejecutamos la transformación haciendo click en RUN, nos pedirá que lo grabemos ( nombre.ktr), grabe en su carpeta de trabajo. Con lo cual se habrá almacenado los datos de la fuente (fichero\_book.csv) al staging área ( tabla f\_salida en base de datos stg\_ebalbuena)



The screenshot shows the SAP Data Services interface. At the top, there's a toolbar with a red circle highlighting the 'Run' button (a green play icon). Below the toolbar, a diagram shows a 'CSV file input' node connected to a 'Tabla\_salida' node, both with green checkmarks indicating successful execution. The bottom section, titled 'Execution Results', shows a log of events:

```

2019/04/14 02:10:23 - Load_fichero - Dispatching started for transformation [Load_fichero]
2019/04/14 02:10:23 - Tabla_salida.0 - Connected to database [uch] (commit=1000)
2019/04/14 02:10:23 - CSV file input.0 - Header row skipped in file 'F:\Universidad Continental\Inteligencia de
2019/04/14 02:10:23 - CSV file input.0 - Finished processing (I=316, O=0, R=0, W=315, U=0, E=0)
2019/04/14 02:10:23 - Tabla_salida.0 - Finished processing (I=0, O=315, R=315, W=315, U=0, E=0)
2019/04/14 02:10:23 - Spoon - The transformation has finished!!
2019/04/14 02:14:35 - Spoon - Save as...
2019/04/14 02:14:35 - Spoon - Save file as...
  
```

Verificamos en Mysql Workbench que los datos han sido cargados:

1 • `SELECT * FROM stg_ebalbuena.f_salida;`

user_code	user	user_id	producto	nota	texto	resaltar	subrayar	tachar	mar
000007b7-a199-47a4-8288-441880f97a11	Usuario Usuario	55000	ESA172025	1	0	0	0	0	0
0000c53a-899d-4051-a579-e74cfd5e4d1f	Usuario Usuario	55001	ESA146820	0	0	0	0	0	0
0000f800-7b29-4d93-9139-66ef3092a96f	Usuario Usuario	55002	ESA142695	1	0	0	0	0	1
0001284f-7a9c-41e5-a961-20d22fde5bc7	Usuario Usuario	55003	ESA146714	0	2	6	1	0	78
00012ba6-040d-483a-9783-277820540911	Usuario Usuario	55004	ESA149230	0	0	0	0	0	2
00012ba6-040d-483a-9783-277820540911	Usuario Usuario	55004	ESA149232	0	0	0	0	0	1
00012ba6-040d-483a-9783-277820540911	Usuario Usuario	55004	ESA149340	0	0	0	1	0	0
00012ba6-040d-483a-9783-277820540911	Usuario Usuario	55004	ESA152759	0	0	0	0	0	6
00012ba6-040d-483a-9783-277820540911	Usuario Usuario	55004	ESA152782	0	0	0	0	0	11
00012ba6-040d-483a-9783-277820540911	Usuario Usuario	55004	ESA152786	0	0	0	0	0	37
00012e2b-bf1c-40f8-a0a8-78b28a66cbe3	Usuario Usuario	55005	ESA146896	0	0	0	0	0	1
00018c72-0e37-475a-a631-70bf55c68647	Usuario Usuario	55006	ESA146850	0	0	0	0	0	0
0001d441-e09e-44a3-9a9c-187b1c4838f1	Usuario Usuario	55007	ESP143544	3	5	2	10	0	1
00021b1f-1e40-404f-831c-9d57c6cf4369	Usuario Usuario	55008	ESA152823	0	0	0	0	0	1

**5) Implementación del modelamiento dimensional correspondiente al datawarehouse.**

Tabla de hecho	fact_fichero
Tablas de dimensiones	dim_asignatura dim_rol dim_nivel dim_usuario dim_producto

Creamos una base de datos usando MySql WorkBench, que corresponde al datawarehouse que almacenará los datos para el análisis tomando el esquema estrella.

```
CREATE DATABASE IF NOT EXISTS `stg_ebalbuena` /*!40100 DEFAULT CHARACTER SET utf8 */;  
USE `stg_ebalbuena`;
```

Script de creación de tabla dimensión (Usar este script como modelo para crear las demás tablas de dimensiones.

```
-- Table structure for table `dim_asignatura`  
  
DROP TABLE IF EXISTS `dim_asignatura`;  
  
CREATE TABLE `dim_asignatura` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `desc` varchar(50) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=604 DEFAULT CHARSET=utf8;
```

**Script de creación de la tabla de hecho: fact\_fichero:**

```
CREATE TABLE `fact_fichero` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `id_asignatura` int(11) NOT NULL,  
  `id_nivel` int(11) NOT NULL,  
  `id_rol` int(11) NOT NULL,  
  `id_usuario` int(11) NOT NULL,  
  `id_producto` int(11) NOT NULL,  
  `c_notas` int(11) NOT NULL DEFAULT '0',  
  `c_texto_libre` int(11) NOT NULL DEFAULT '0',  
  `c_resaltar_texto` int(11) NOT NULL DEFAULT '0',  
  `c_subrayar_texto` int(11) NOT NULL DEFAULT '0',  
  `c_tachar_texto` int(11) NOT NULL DEFAULT '0',
```

```

`c_mano_alzada` int(11) NOT NULL DEFAULT '0',
`c_rectangulo` int(11) NOT NULL DEFAULT '0',
`c_elipse` int(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`id`),
UNIQUE KEY `id_unique` (`id_usuario`,`id_producto`),
KEY `id_asignatura` (`id_asignatura`),
KEY `id_nivel` (`id_nivel`),
KEY `id_rol` (`id_rol`),
KEY `id_producto` (`id_producto`),
KEY `IDX` (`id_usuario`,`id_producto`)
) ENGINE=InnoDB AUTO_INCREMENT=836001 DEFAULT CHARSET=utf8;

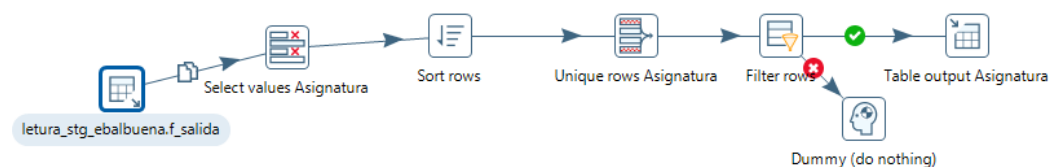
```

Verificamos que se haya creado la estructura deseada:



## 6) Carga de datos desde el staging área (stg\_ebalbuena.f\_salida) a nuestro Datawarehouse (olap\_ebalbuena)

- a) Carga de datos a las tablas dimensiones. Abrimos una nueva transformación y realizaremos los procesos ETL para cada dimensión usando los siguientes pasos:



Configuramos cada paso, lo cual debe ser realizado para cada dimensión.

1a) PASO: lectura\_stg\_ebalbuena.f\_salida

Table input

Step name:

Connection:

SQL

```
SELECT * FROM stg_ebalbuena.f_salida
```

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size

Configuramos la conexión uch que apunta al staging área.

Database Connection

General

Connection name:

Connection type:

- KingbaseES
- LucidDB
- MS Access
- MS SQL Server
- MS SQL Server (Native)
- MariaDB
- MaxDB (SAP DB)
- MonetDB
- MySQL
- Native Mondrian
- Neoview
- Netezza
- OpenERP Server
- Oracle
- Oracle RDR

Access:

- Native (JDBC)
- ODBC
- JNDI

Settings

Host Name:

Database Name:

Port Number:

Username:

Password:

☒ Use Result Streaming Cursor

2a) PASO: Select values Asignatura.

Select values

Step name:

Select & Alter Remove Meta-data

Fields:

#	Fieldname	Rename to	Length	Precision
1	asignatura			

Get fields to select  
Edit Mapping

Include unspecified fields, ordered ☐

Help OK Cancel

### 3a) PASO: Sort rows

Sort rows

Step name:

Sort directory:  Browse...

TMP-file prefix:

Sort size (rows in memory):

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	asignatura	Y	N	N	0	N

Help OK Cancel Get Fields

### 4a) PASO: Unique rows Asignatura

Unique rows

Step name:

Settings

Add counter to output? ☐ Counter field:

Redirect duplicate row ☐ Error description:

Fields to compare on (no entries means: compare complete row)

#	Fieldname	Ignore case
1	asignatura	N

Help OK Cancel Get

### 5a) Paso: Filter rows

The screenshot shows the 'Filter rows' configuration window. The 'Step name' is 'Filter rows'. The 'Send 'true' data to step:' is 'Table output Asignatura'. The 'Send 'false' data to step:' is 'Dummy (do nothing)'. The condition is defined as 'asignatura IS NOT NULL'. There are two empty input fields for the condition. The window has 'Help', 'OK', and 'Cancel' buttons.

Step name:

Send 'true' data to step:

Send 'false' data to step:

The condition:

### 6a) PASO: Table output Asignatura con la conexión dwh a olap\_ebalbuena

The screenshot shows the 'Table output' configuration window. The 'Step name' is 'Table output Asignatura'. The 'Connection' is 'dwh'. The 'Target schema' is 'olap\_ebalbuena'. The 'Target table' is 'dim\_asignatura'. The 'Commit size' is '1000'. The 'Truncate table' checkbox is unchecked. The 'Ignore insert errors' checkbox is unchecked. The 'Specify database fields' checkbox is checked. The 'Main options' tab is selected. The 'Fields to insert:' table has one row with 'desc' in the 'Table field' column and 'asignatura' in the 'Stream field' column. The window has 'Help', 'OK', 'Cancel', and 'SQL' buttons.

Step name:

Connection:

Target schema:

Target table:

Commit size:

Truncate table: ☐

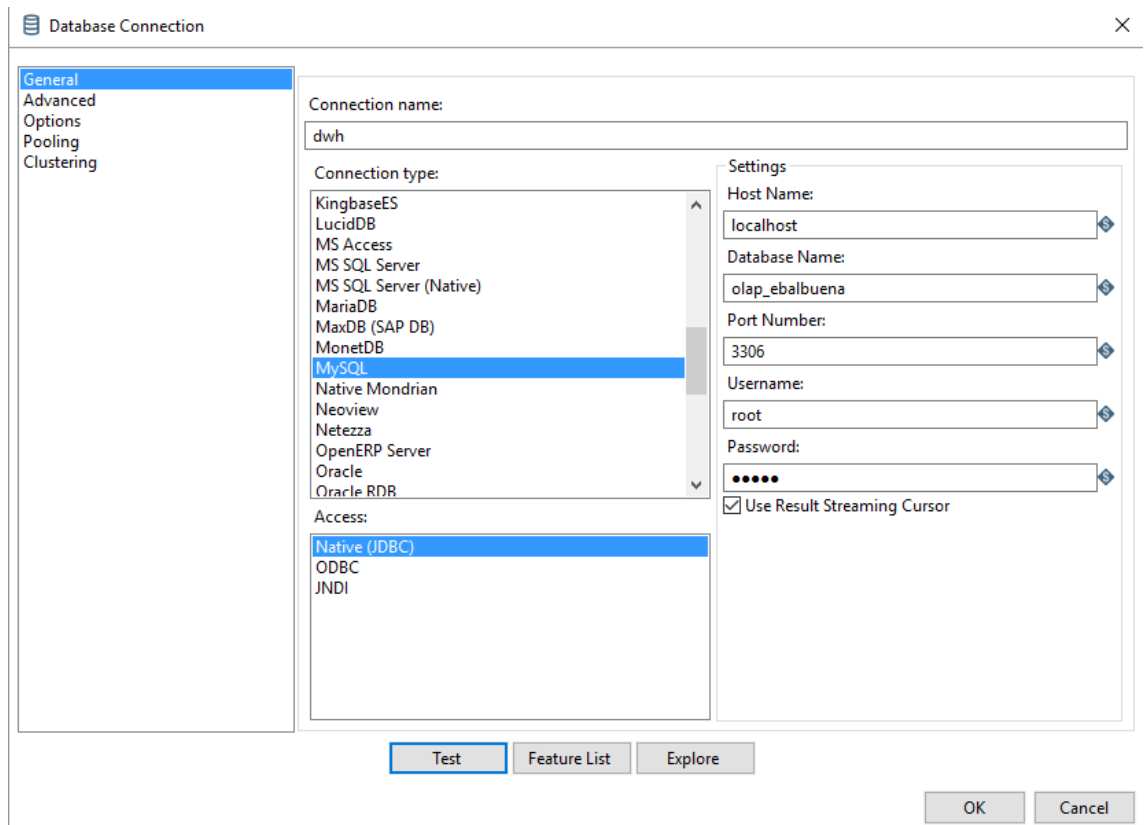
Ignore insert errors: ☐

Specify database fields: ☒

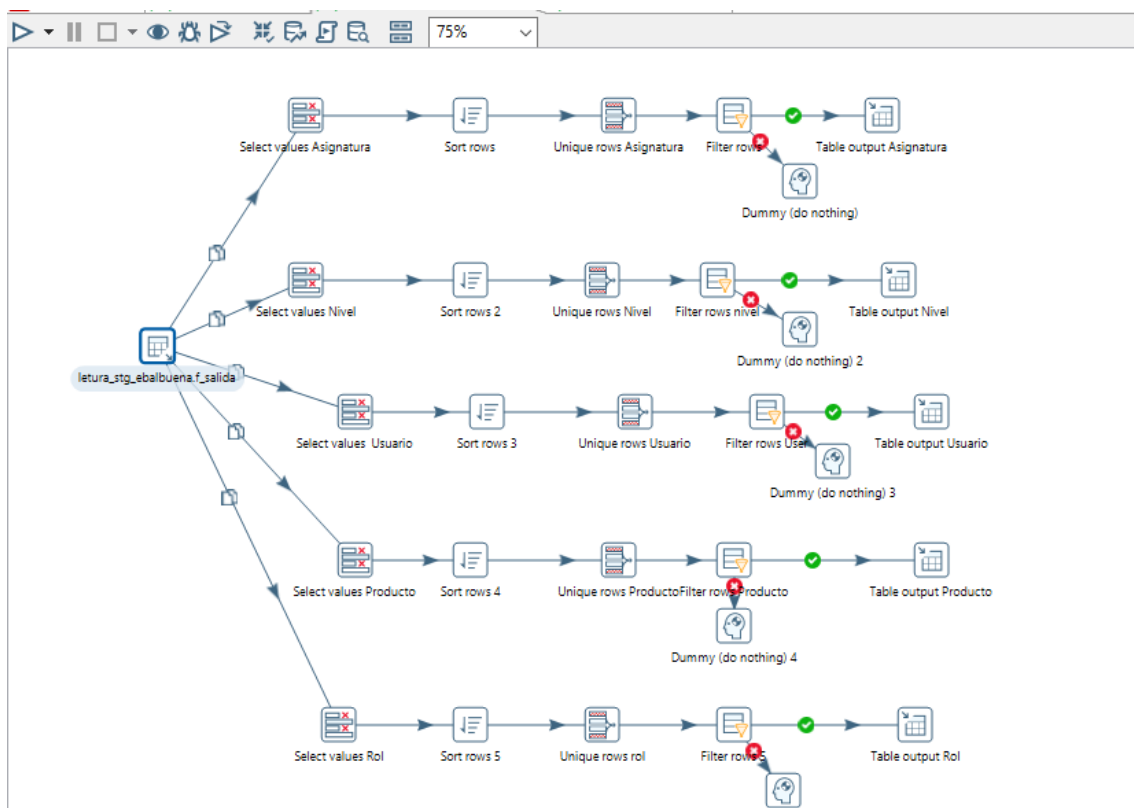
Main options | Database fields

Fields to insert:

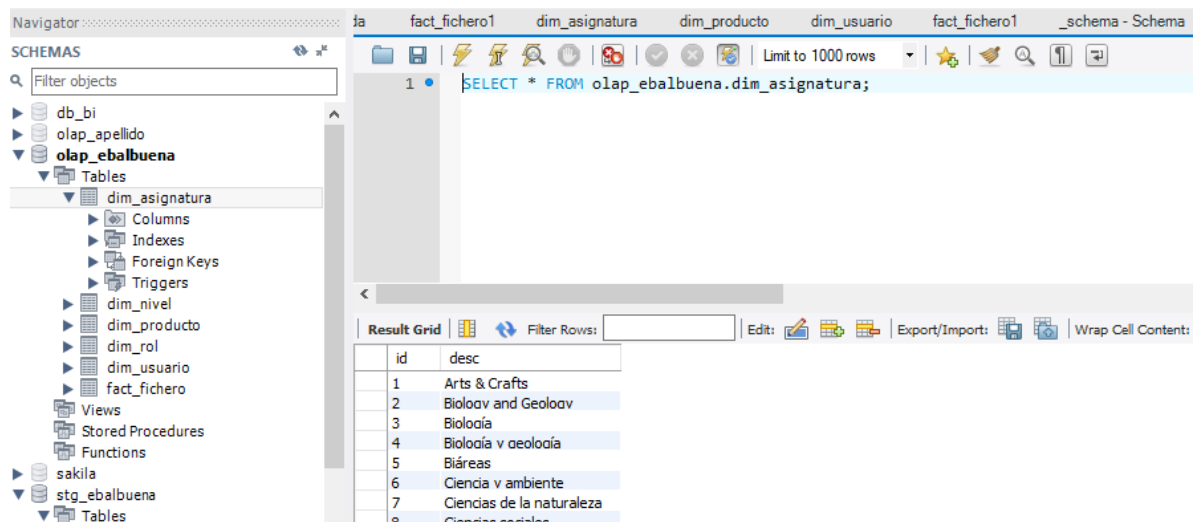
#	Table field	Stream field
1	desc	asignatura



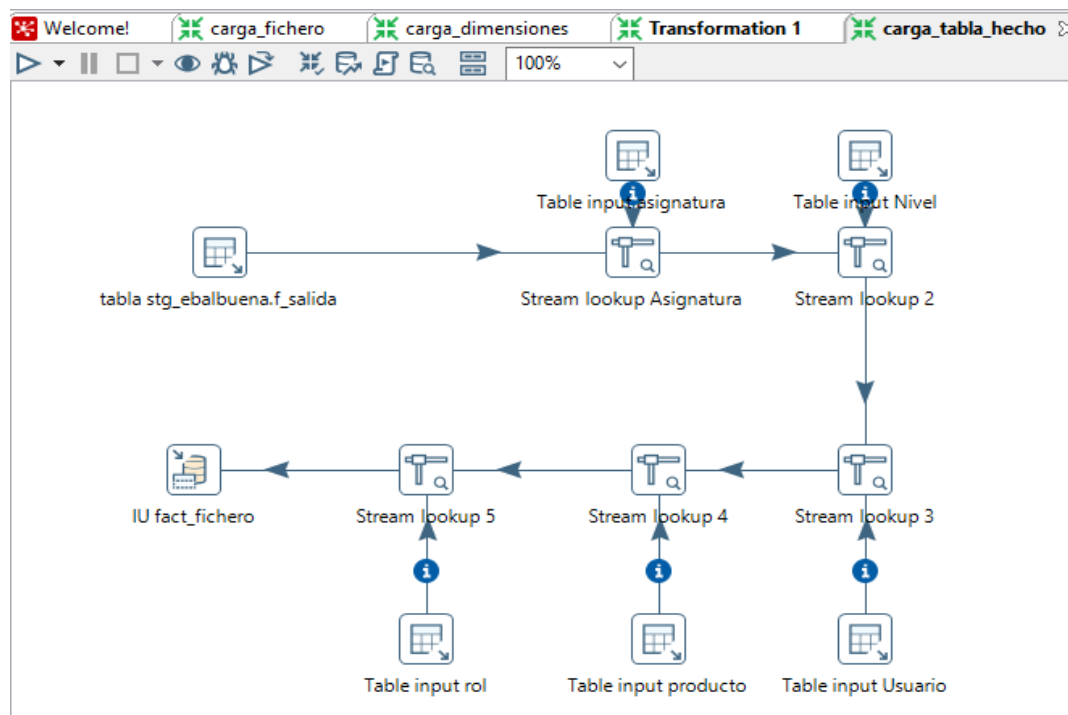
Usar los mismos pasos de 2a a 6a para cada dimensión. El modelo final debe tener todas las dimensiones que se definieron en el modelo:



Corremos la actual transformación y luego verificamos que los datos se hayan cargado en nuestra base de datos.




- b) Carga de datos a la tabla de hechos: fact\_fichero. Abrimos una nueva transformación y realizaremos los procesos ETL para cargar la tabla de hechos usando los siguientes pasos:





Configuración de cada paso:

1)  tabla stg\_ebalbuena.f\_salida

**PASO** →

**Table input**

Step name:

Connection:

SQL

```
SELECT * FROM stg_ebalbuena.f_salida
```

Line 1 Column 0

Enable lazy conversion ☐

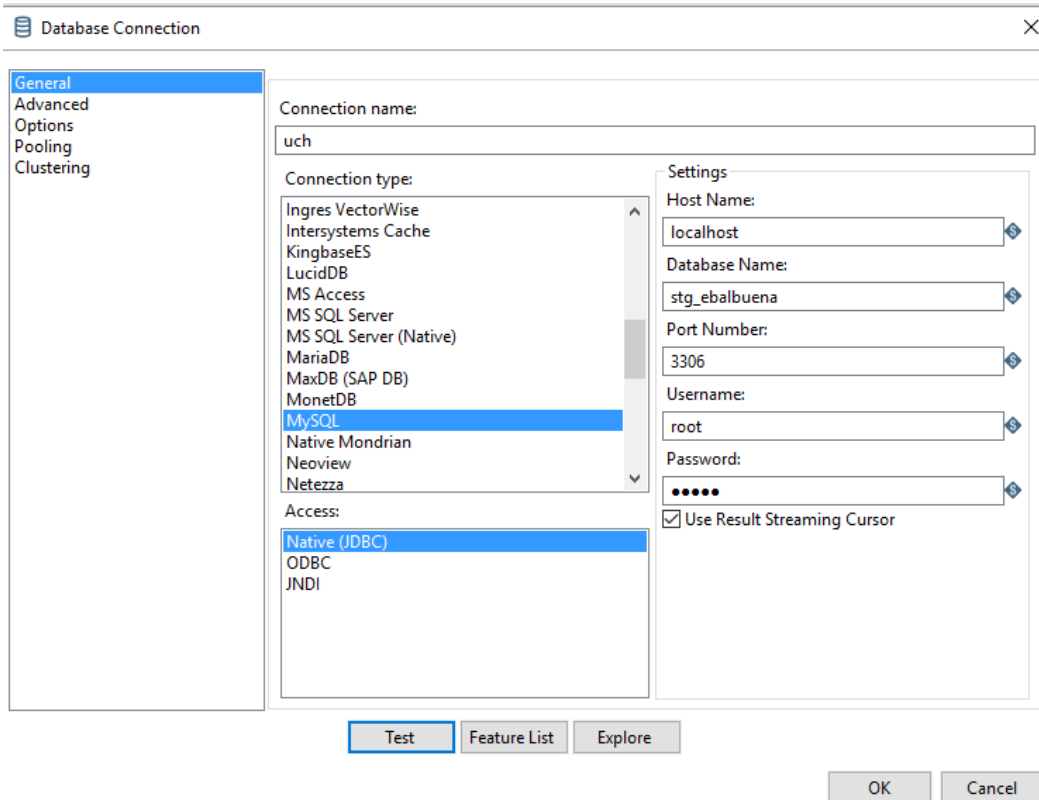
Replace variables in script? ☐

Insert data from step

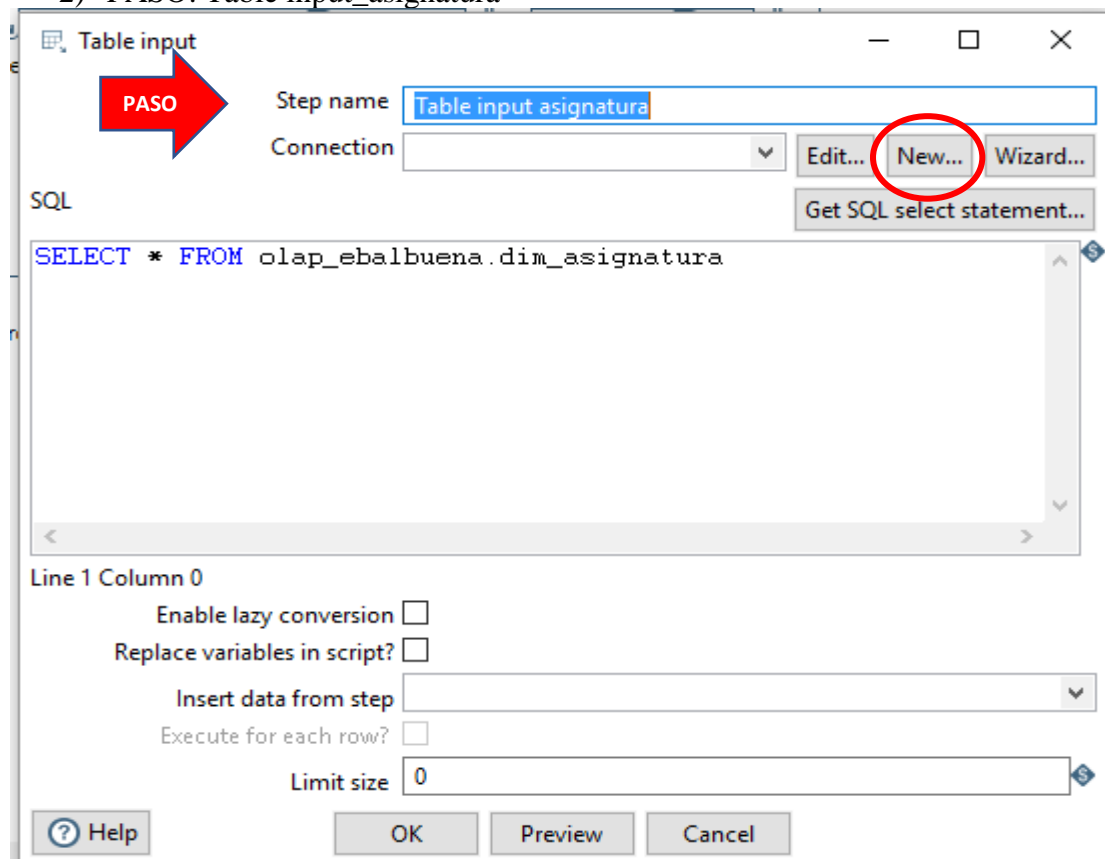
Execute for each row? ☐

Limit size

Usamos la conexión a stg\_ebalbuena



2) PASO: Table input\_asignatura



Database Connection

General  
Advanced  
Options  
Pooling  
Clustering

Connection name:  
dwh

Connection type:  
Ingres VectorWise  
InterSystems Cache  
KingbaseES  
LucidDB  
MS Access  
MS SQL Server  
MS SQL Server (Native)  
MariaDB  
MaxDB (SAP DB)  
MonetDB  
MySQL  
Native Mondrian  
Neoview  
Netezza  
OpenERP Server  
Oracle


Access:  
Native (JDBC)  
ODBC  
JNDI

Settings  
Host Name:  
localhost  
Database Name:  
olap\_ebalbuena  
Port Number:  
3306  
Username:  
root  
Password:  
•••••  
☒ Use Result Streaming Cursor

Test Feature List Explore

OK Cancel

Examinamos la data con una vista previa:

 Examine preview data

Rows of step: Table input asignatura (100 rows)

#	id	desc
1	1	Arts & Crafts
2	2	Biology and Geology
3	3	Biología
4	4	Biología y geología
5	5	Biáreas
6	6	Ciencia y ambiente
7	7	Ciencias de la naturaleza
8	8	Ciencias sociales
9	9	Comunicación
10	10	Conocimiento del medio

### 3) PASO: Stream lookup Asignatura

Stream lookup

**PASO**

Step name: Stream lookup Asignatura

Lookup step: Table input asignatura

The key(s) to look up the value(s):

#	Field	LookupField
1	asignatura	desc

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	id	id_asignatura	-1	Integer

Preserve memory (costs CPU) ☒

Key and value are exactly one integer field ☐

Use sorted list (i.s.o. hashtable) ☐

Help OK Cancel Get Fields Get lookup fields

**asignatura**, es el campo que viene de la tabla f\_salida (staging área)

**desc**, es el campo de la tabla de hecho fact\_fichero donde se guardará ese dato.

4) Los pasos 2 y 3 deben configurarse de manera similar para cada dimensión.

### 5) PASO:



En este paso se realiza la carga de datos a la tabla de hechos en el Datawarehouse que está dado la tabla olap\_ebalbuena.fact\_fichero.

**Insert / update**

**PASO** → Step name:

Connection:

Target schema:

Target table:

Commit size:

Don't perform any updates: ☒

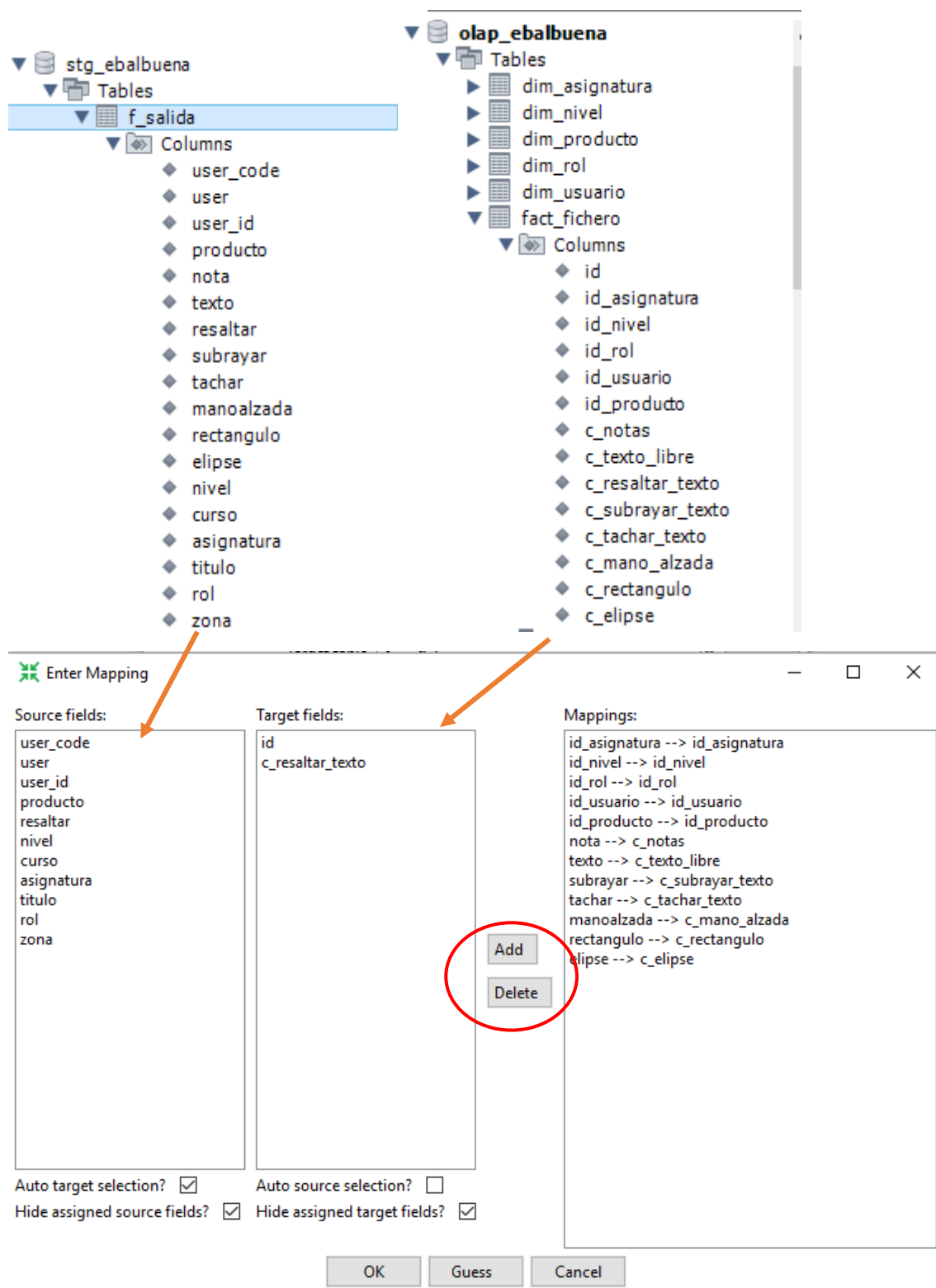
The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	id_usuario	=	id_usuario	
2	id_producto	=	id_producto	

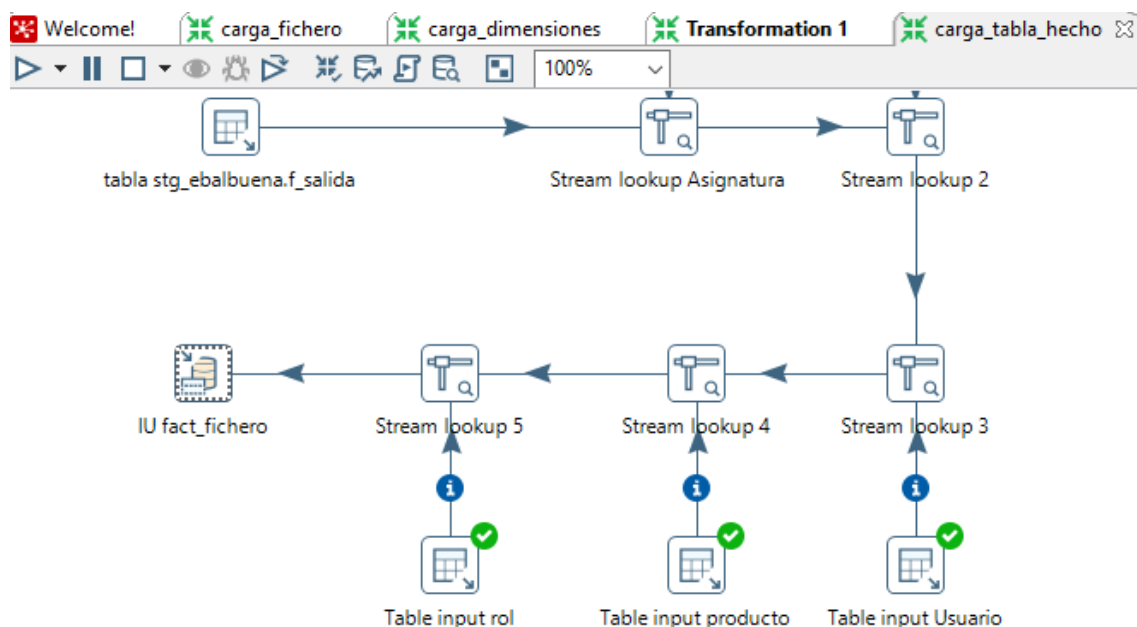
Update fields:

#	Table field	Stream field	Update
1	id_asignatura	id_asignatura	Y
2	id_nivel	id_nivel	Y
3	id_rol	id_rol	Y
4	id_usuario	id_usuario	N
5	id_producto	id_producto	N
6	c_notas	nota	Y
7	c_texto_libre	texto	Y
8	c_subrayar_texto	subrayar	Y
9	c_tachar_texto	tachar	Y
10	c_mano_alzada	manoalzada	Y
11	c_rectangulo	rectangulo	Y
12	c_elipse	elipse	Y

Usamos la opción Edit mapping para relacionar los campos que vienen del **staging : f\_salida** con los campos de **la tabla de hechos olap\_ebalbuena.fact\_fichero**.



Corremos la transformación y veremos en la zona de Logging como avanza la ejecución hasta llegar al último paso.



## Execution Results

Logging	Execution History	Step Metrics	Performance Graph	Metrics	Preview data
<p>2019/04/14 03:25:54 - IU fact_fichero.0 - linenr 100000</p> <p>2019/04/14 03:26:06 - Stream lookup 3.0 - linenr 1850000</p> <p>2019/04/14 03:26:13 - Stream lookup 2.0 - linenr 150000</p> <p>2019/04/14 03:26:18 - Stream lookup 4.0 - linenr 150000</p> <p>2019/04/14 03:27:04 - Stream lookup 5.0 - linenr 150000</p> <p>2019/04/14 03:27:04 - tabla stg_ebalbuena.f_salida.0 - linenr 200000</p> <p>2019/04/14 03:27:19 - Stream lookup Asignatura.0 - linenr 200000</p> <p>2019/04/14 03:27:20 - IU fact_fichero.0 - linenr 150000</p>					

Verificamos que los datos se hayan cargado en la Base de Datos

m\_usuariofact\_fichero1\_schema - Schemaf\_salidadim\_asignaturadim\_rolfact\_productof\_salidadim\_productofact\_fichero

1

SELECT \* FROM olap\_ebalbuena.fact\_fichero;

<

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

IA

Fetch rows:

	id	id_asignatura	id_nivel	id_rol	id_usuario	id_producto	c_notas	c_texto_libre	c_resaltar_texto	c_subrayar_texto	c_tachar_
	446449	235	39	3	19406	4920	0	0	0	0	0
	446450	238	39	3	19411	4927	0	0	0	0	0
	446451	235	38	3	19416	4909	0	0	0	0	0
	446452	231	39	3	19418	5306	0	0	0	0	0
	446453	248	35	3	19420	5137	0	0	0	0	0
	446454	232	38	3	19421	4889	0	0	0	0	0
	446455	211	38	3	19421	4934	0	0	0	0	0
	446456	223	39	3	19424	5231	0	0	0	0	0
	446457	248	39	3	19425	5138	0	0	0	0	0
	446458	235	39	3	19425	5604	0	0	0	0	0
	446459	232	38	3	19426	4974	0	0	0	0	0
	446460	248	38	3	19426	5148	0	0	0	0	0
	446461	212	38	3	19426	5272	0	0	0	0	0
	446462	232	39	3	19427	5160	0	0	0	0	0

**¡Muy bien!** Ya tenemos nuestra datawarehouse con los datos cargados. Ya estamos listos para explotar los datos.