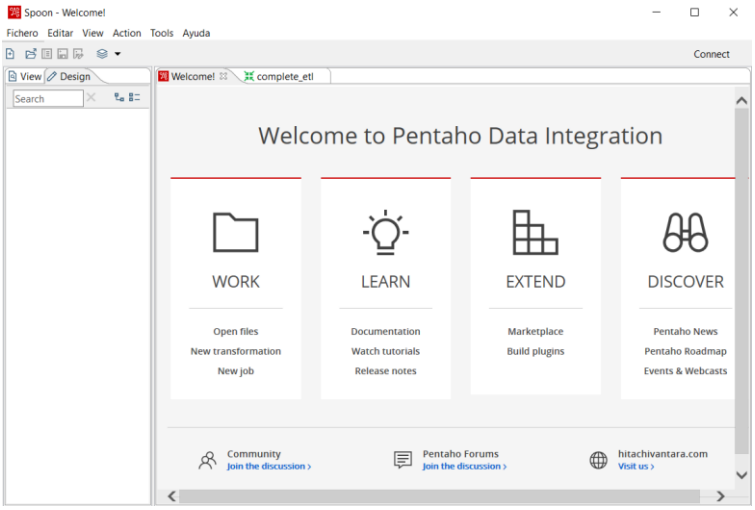
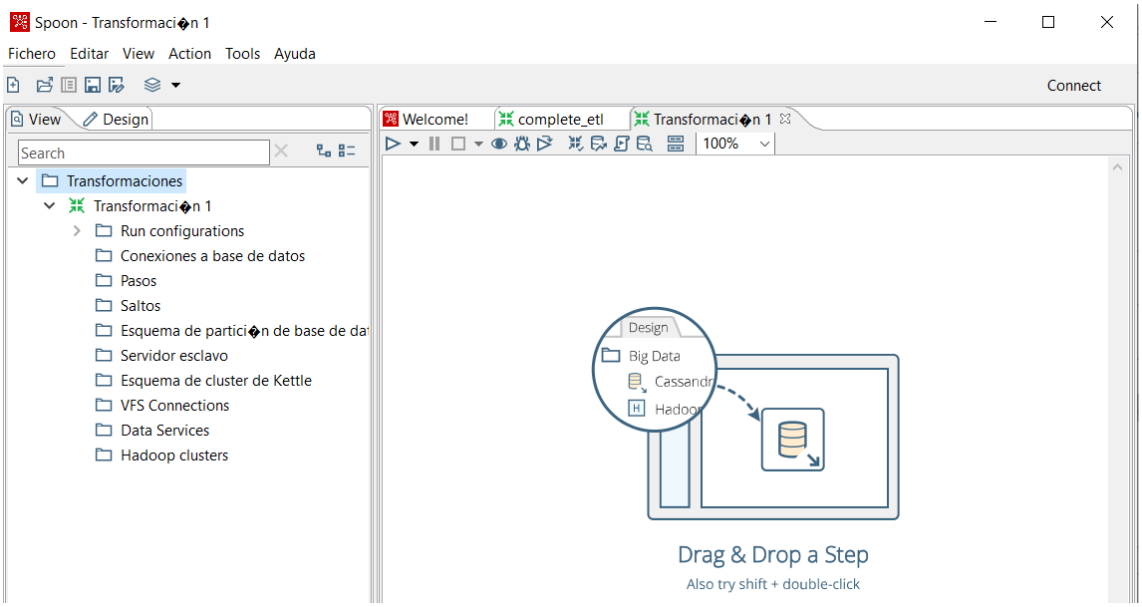
Task3. Data transformation

In this task I am going to show how I have done the ETL process for the Airlines database using Pentaho data integration.

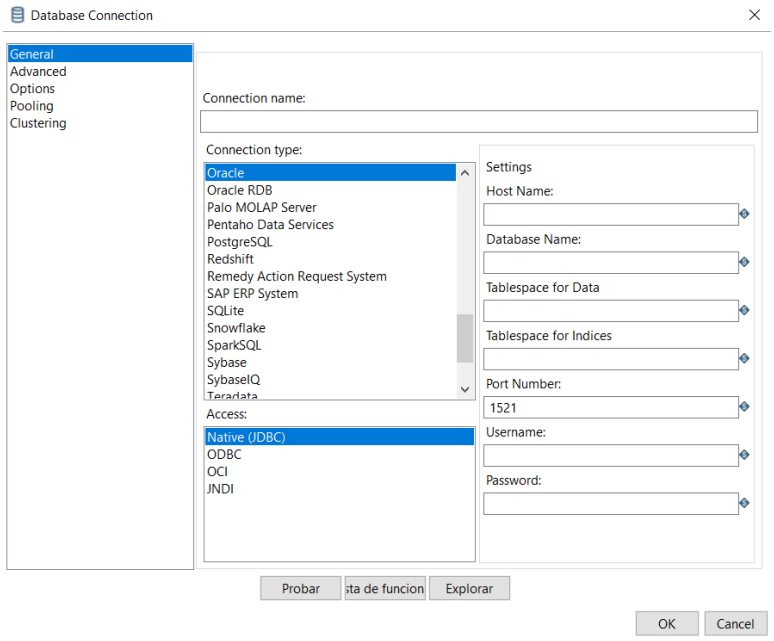
First, we have to open the data integration program.



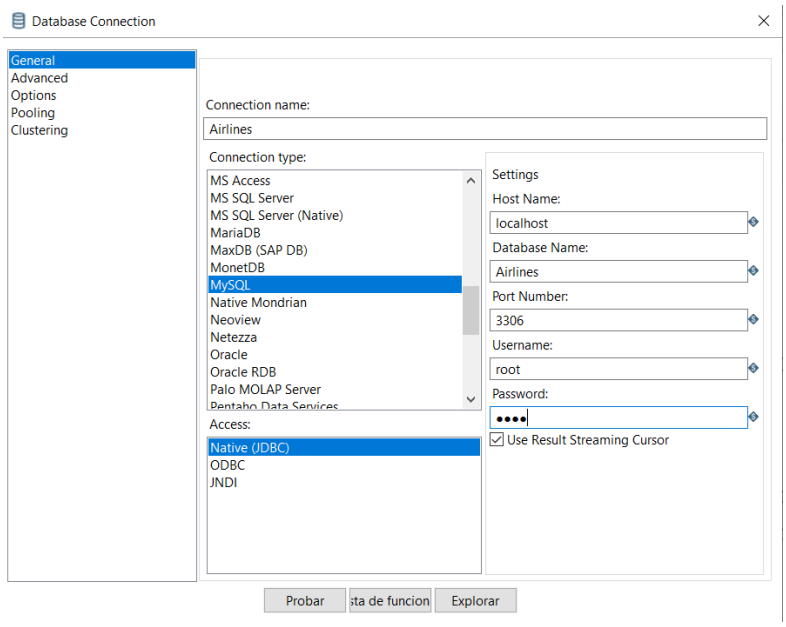
Now I am going to create a new transformation to start the ETL process.



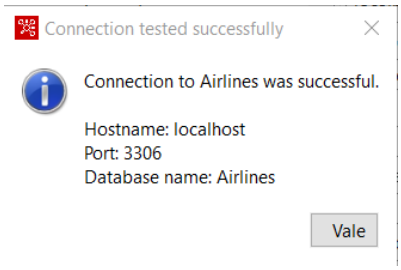
The next step is to connect it to the Airlines database. I am going to do that by right clicking on ‘Conexiones a bases de datos’ -> New.



After clicking it appears this window. In it we have to put the database login info as we did in the schema-workbench.



When the info is on the connection we click on ‘Probar’ to check if the connection can be made. To see that everything id ok we have to see this window.



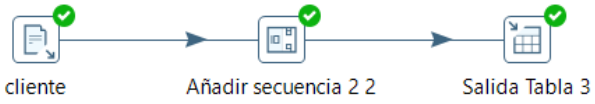
Now that the connection can be made we have to click in ‘Ok’ and we can start the ETL process.

-ETL

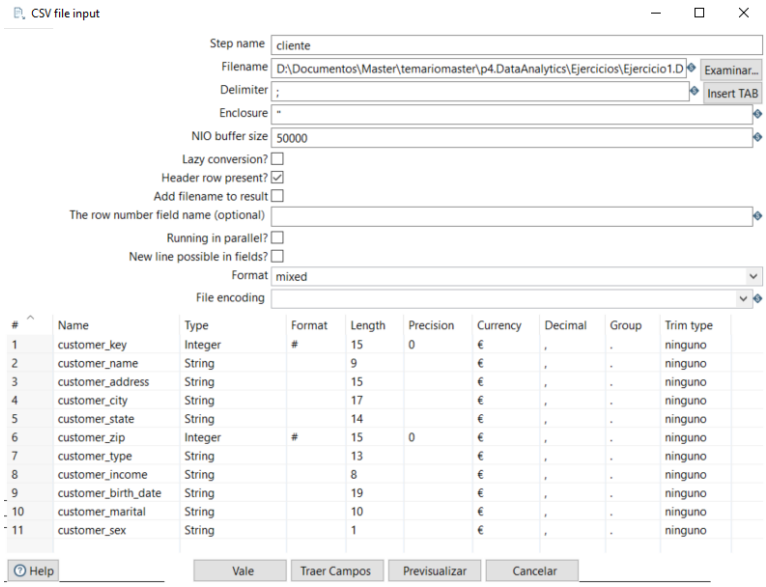
The first thing we are going to do is load the dimension tables.

We are going to start by uploading the client table.

-Customer

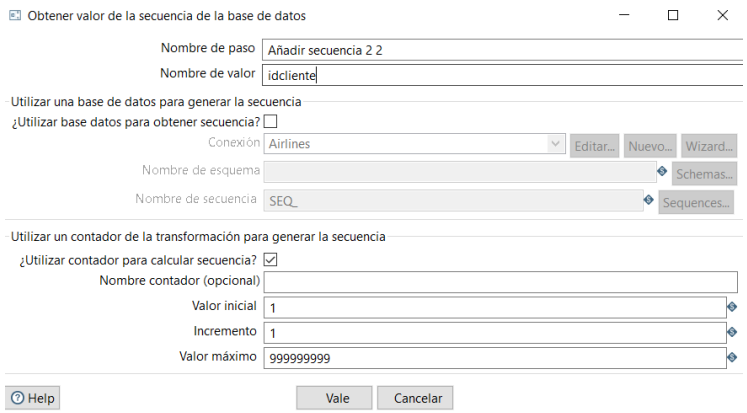


In the first step we have the entry of the CSV table.

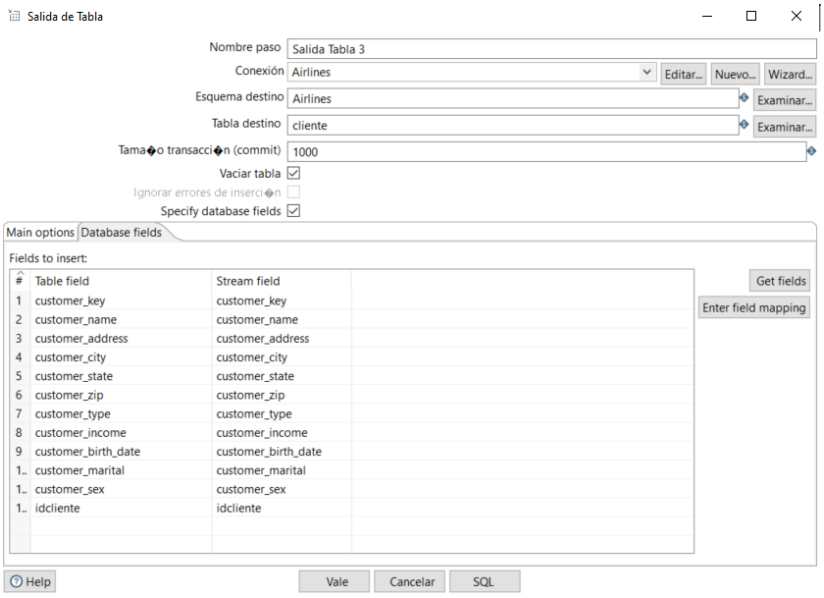


In this step we have to select the address of the CSV file we have for the client, we also have to put the semicolon separator because it is a CSV file what we are importing. After that we click on “Traer campos” and then in “Vale”.

In the next step we added a add sequence block because we want to create an autogenerated value in order to create ‘idcliente’ so we can identify the clients in the fact table.



Once we have the 'idcliente' attribute ready' the las step is to upload the table to the cliente table in MySql server.



To do that we have to select the destiny table in the field "Tabla destino" and then set the fields to insert. We can see that the "idcliente" field has been inserted correctly.

If we Run the program, we can do a SQL query in the Airlines database to check that the data has been inserted.

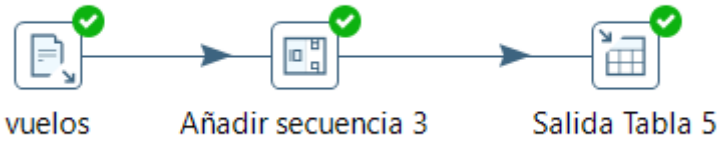
```
1  select * from Airlines.canal;
2  select * from Airlines.aeropuerto;
3  select * from Airlines.cliente;
4  select * from Airlines.tarifa;
5  select * from Airlines.vuelo;
6  select * from Airlines.trayecto order by flown_key,cliente;
7  select * from Football.player;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

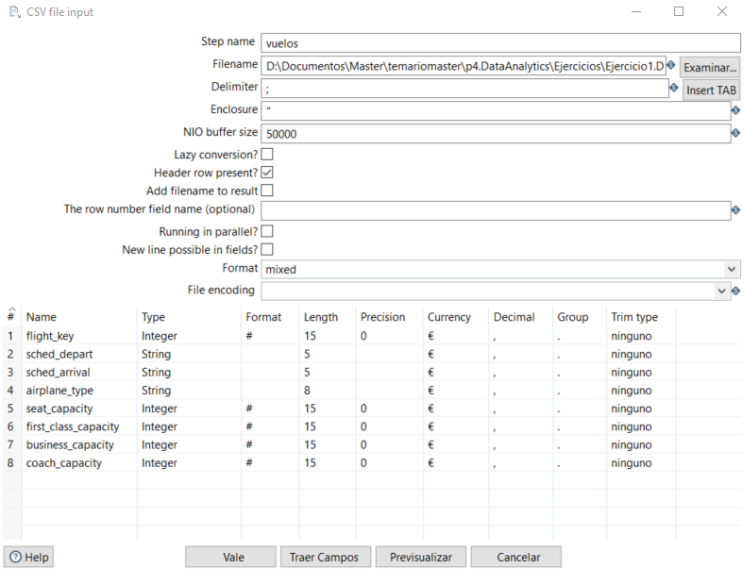
idcliente	customer_key	customer_name	customer_address	customer_city	customer_state	customer_zip	customer_type	customer_inco
1	1	Anderson	1607 Shady Lane	Birmingham	Alabama	40928	Employed	$102.000
2	2	Antoni	3859 Shady Lane	Tuscaloosa	Alabama	35294	Employed	$35.000
3	3	Appley	1923 Shady Lane	Anchorage	Alaska	58358	Employed	$47.000
4	4	Ashby	9369 Shady Lane	Juneau	Alaska	90421	Employed	$94.000
5	5	Barr	7593 Shady Lane	Flagstaff	Arizona	67536	Employed	$93.000
6	6	Barrett	5332 Shady Lane	Phoenix	Arizona	88392	Employed	$117.000
7	7	Bennett	4116 Shady Lane	Little Rock	Arkansas	23848	Self Employed	$35.000
8	8	Boone	3100 Shady Lane	Midville	Arkansas	88536	Employed	$93.000
9	9	Clarke	7808 Shady Lane	San Diego	California	39238	Employed	$61.000
10	10	Clewett	3997 Shady Lane	Red Bluff	California	37374	Employed	$87.000
11	11	Cluster	8640 Shady Lane	Denver	Colorado	61892	Military	$44.000
12	12	Coghlin	7143 Shady Lane	Steamboat S...	Colorado	92682	Self Employed	$44.000
13	13	Davis	8765 Shady Lane	Hartford	Connecticut	94452	Employed	$91.000

-Flights

Now I am going to add the flight table, we are going to do the same as in the customers one.

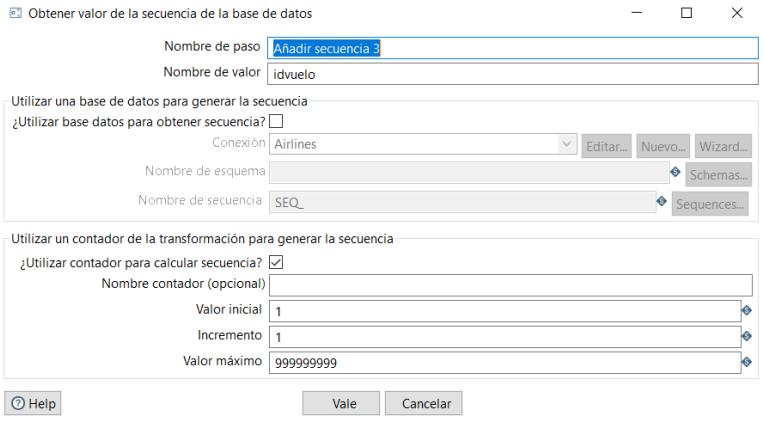


In the first step we have the block to import the CSV file.

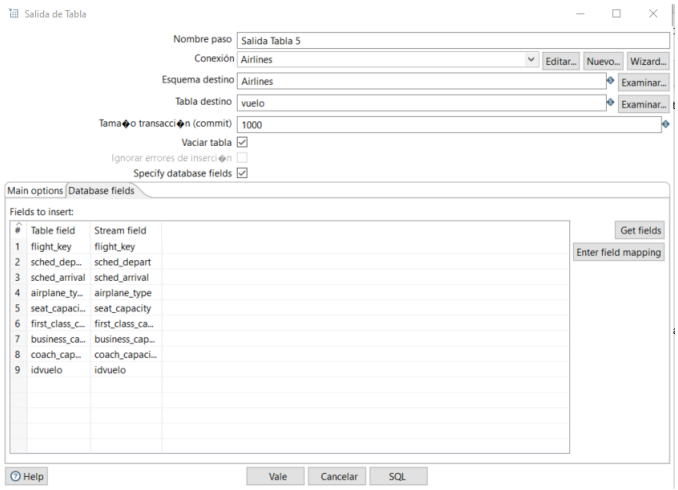


#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	flight_key	Integer	#	15	0	€	,	.	ninguno
2	sched_depart	String		5		€	,	.	ninguno
3	sched_arrival	String		5		€	,	.	ninguno
4	airplane_type	String		8		€	,	.	ninguno
5	seat_capacity	Integer	#	15	0	€	,	.	ninguno
6	first_class_capacity	Integer	#	15	0	€	,	.	ninguno
7	business_capacity	Integer	#	15	0	€	,	.	ninguno
8	coach_capacity	Integer	#	15	0	€	,	.	ninguno

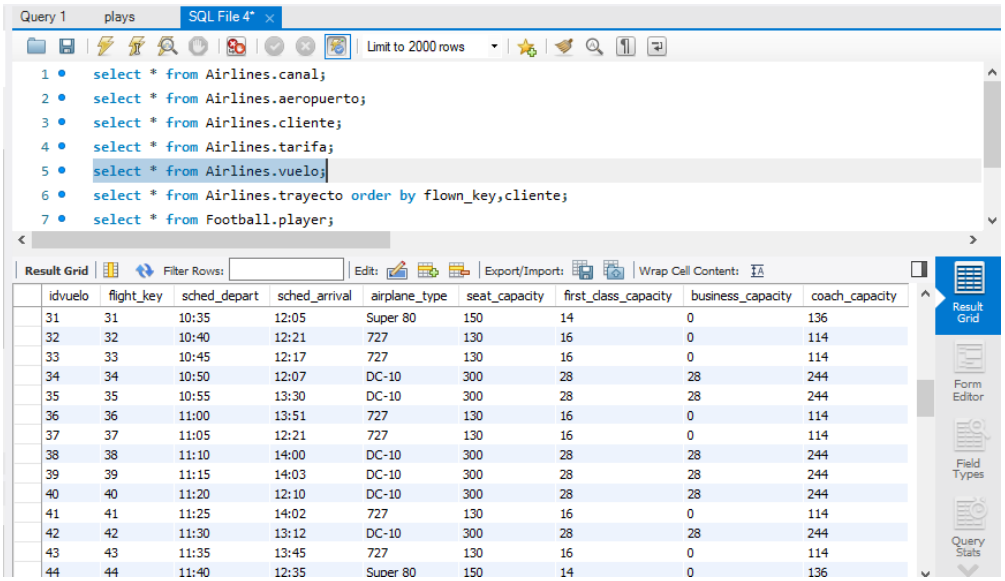
After that I am going to generate the 'idvuelo' attribute the same way I did in the client table.



And in the las step I am going to choose the destiny table in the Airlines database with the 'idvuelo' generated.

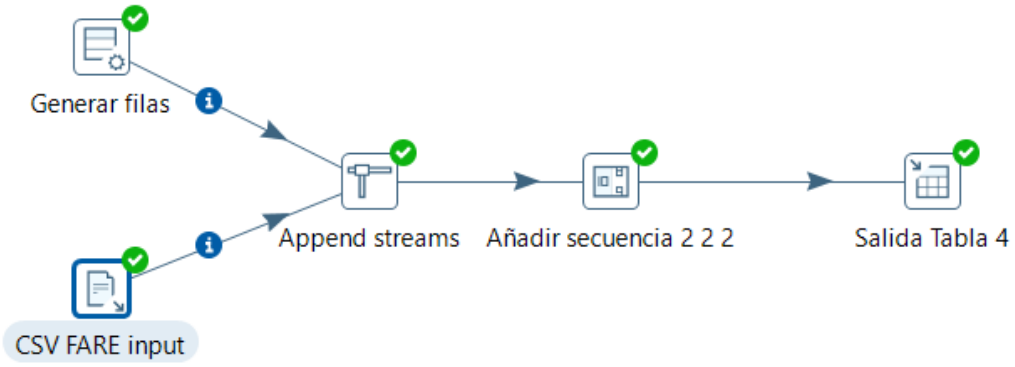


To check that it has worked we are going to do the SQL query.

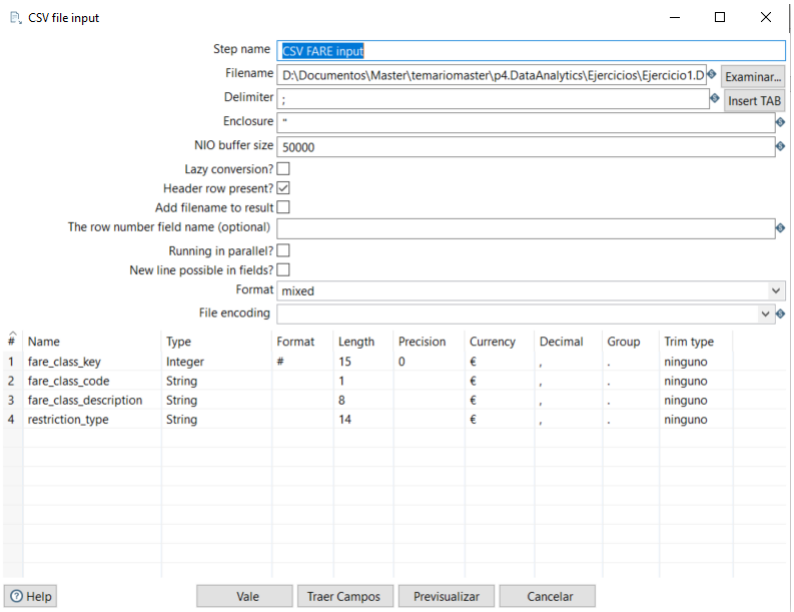


-Fare

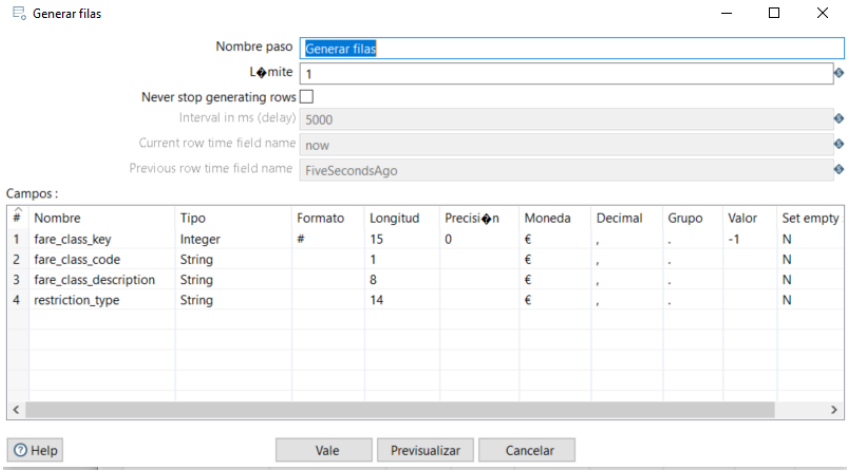
Now we are going to import the fare table. With this table we have a problem because the fare CSV we have four kinds of fare_type and in the frequentflayer table we can see that this number goes up to 20 kind of fares. This could mean that the data we have doesn't have all the fare types or that this kind of fares have been eliminated. It could also mean that the people that took the data didn't do it properly so for this kind of data whose fare_key goes up of 4 we are going to create a row in the fare table with a 5 idfare and a -1 in fare_key to store there the incorrect data, this way we don't lose it.



The CSV fare input is the same block as in the cases explained before.

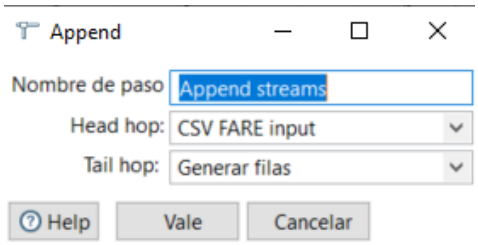


Now I have the generate row block.

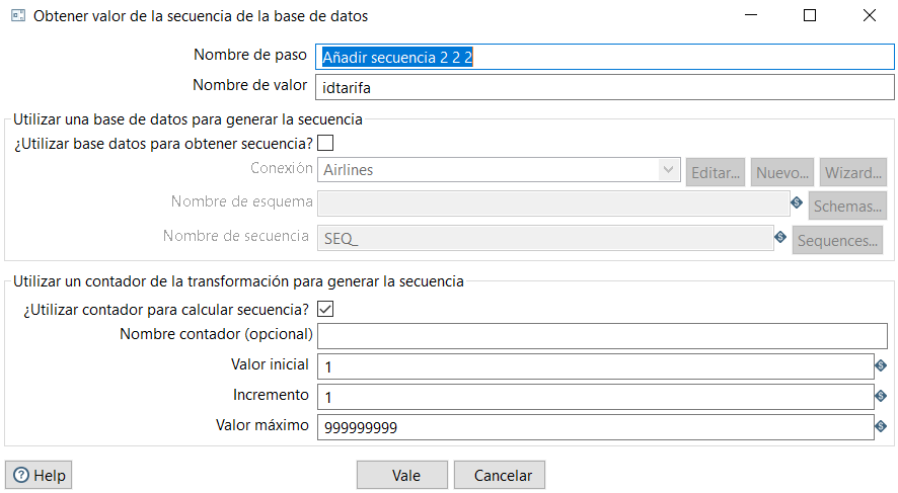


This row creates a row where the value of 'fare_class_key' is -1.

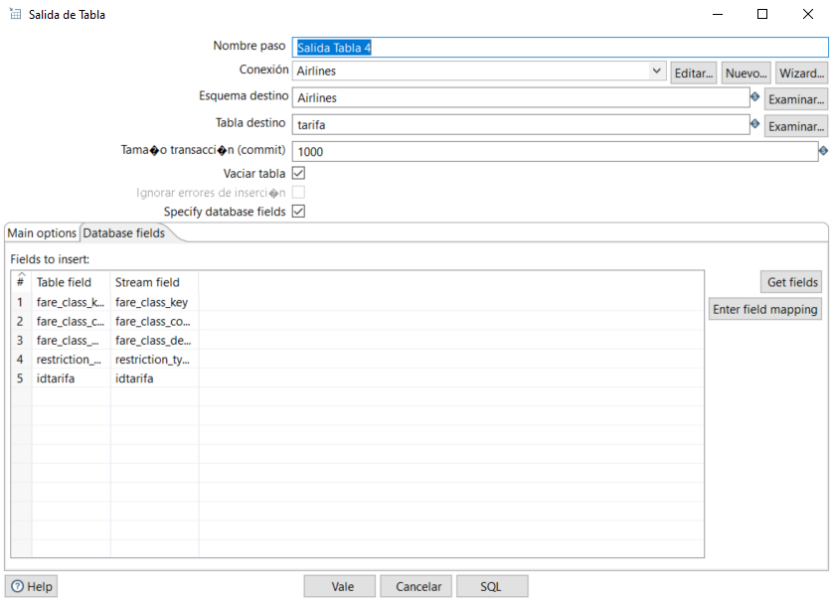
After that we find the append streams block, it unifies the CSV and the generated row from the block before.



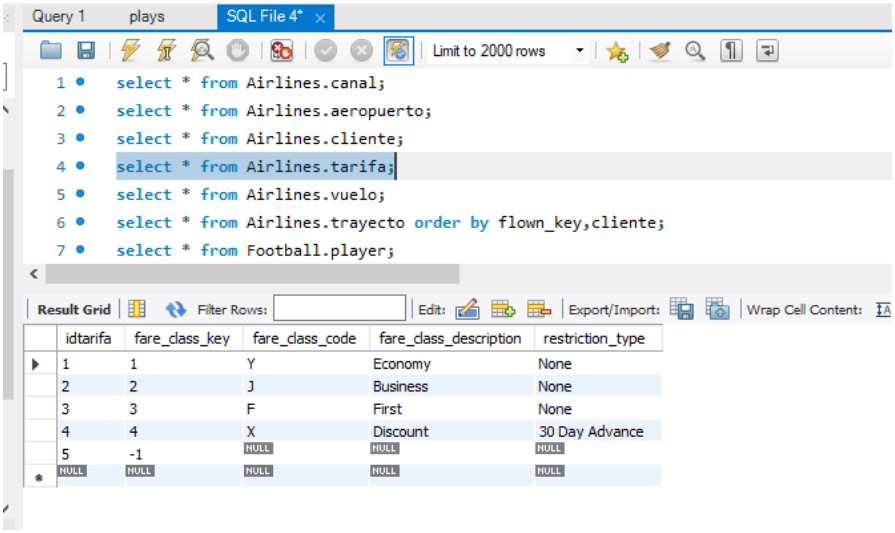
After that we have the generate sequence once again.



And in the last place the output for the Airlines database.



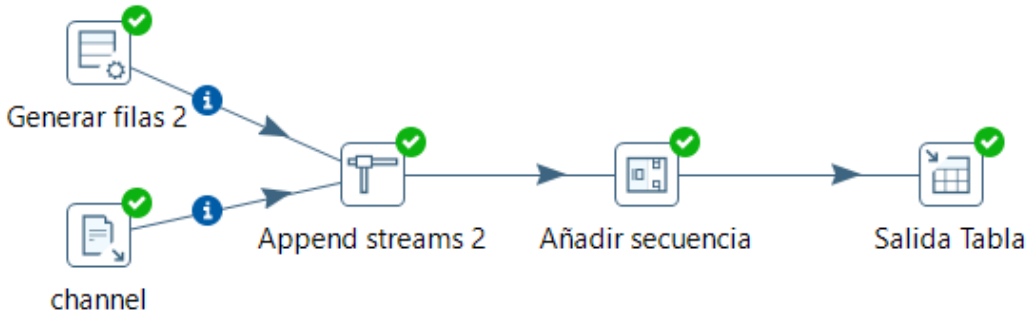
To check again we are going to do the SWL query.



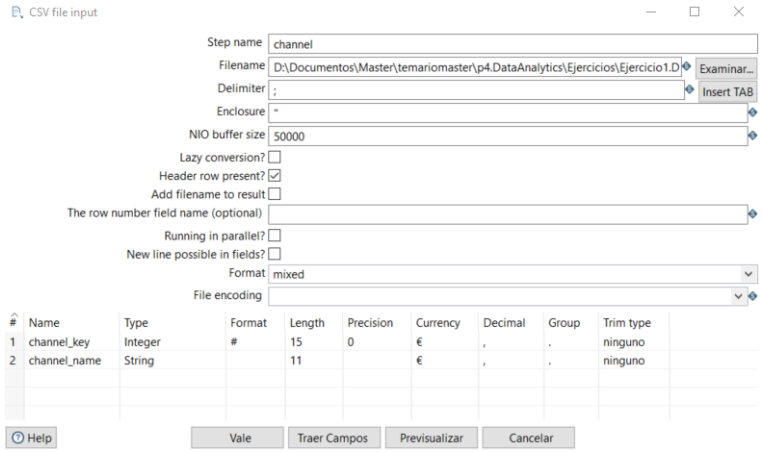
We can see how the row has been created.

-Channel

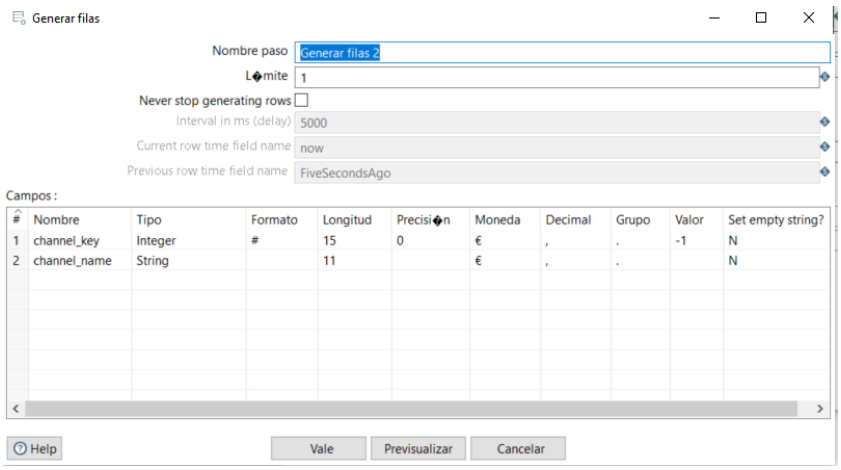
Now we are going to import the channel CSV, in this case it happens the same thing as before so to fix it we are algo going to generate a row.



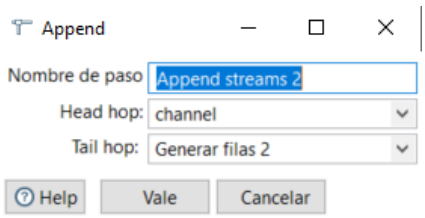
First we import the CSV.



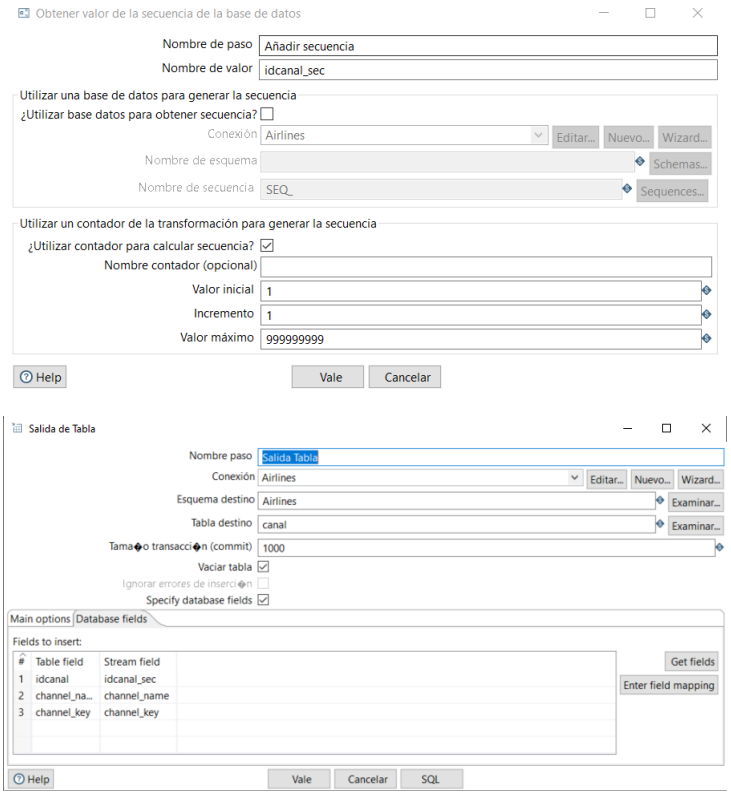
After that we generate the row with value -1 for the 'channel_key' attribute.



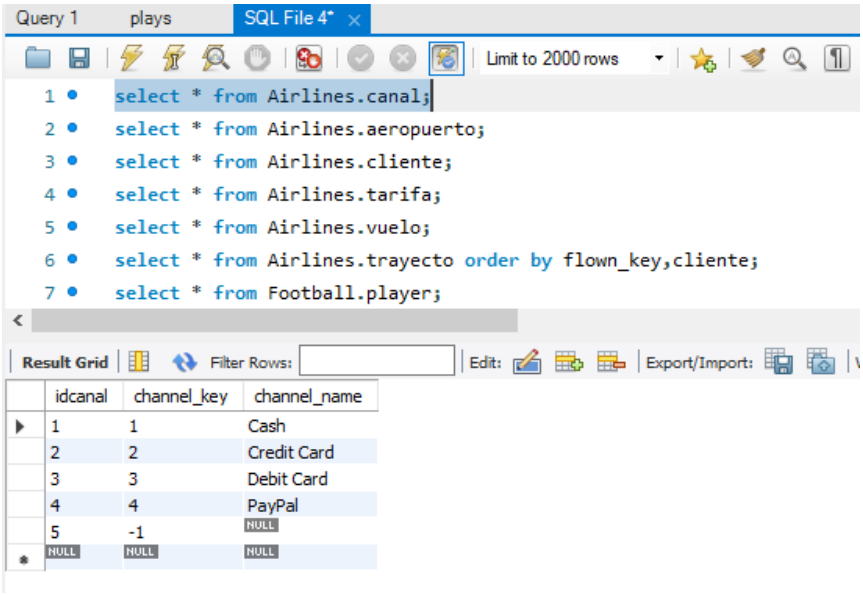
We append the data.



Now we add the sequence and generate the output.

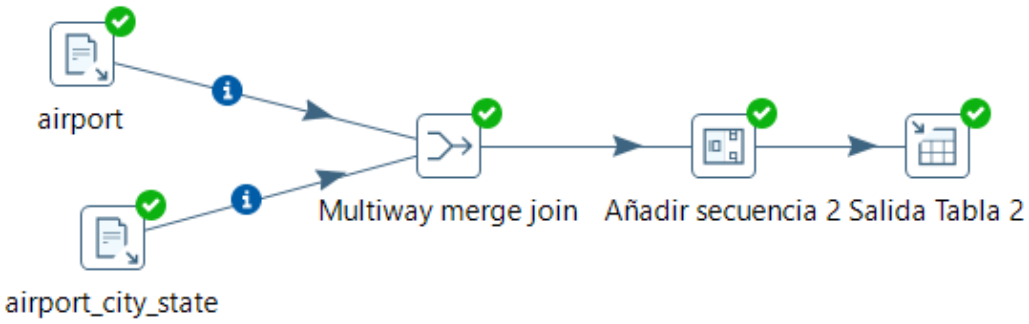


Last we are going to check MySQL again.

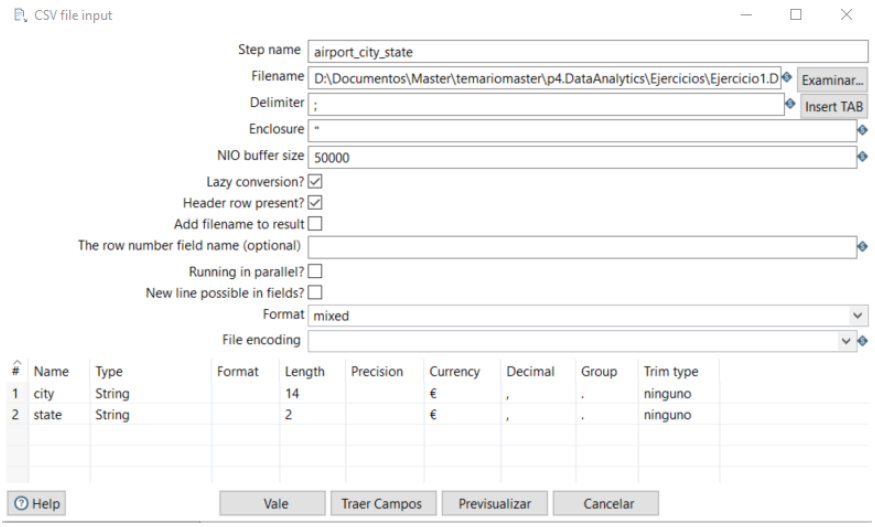


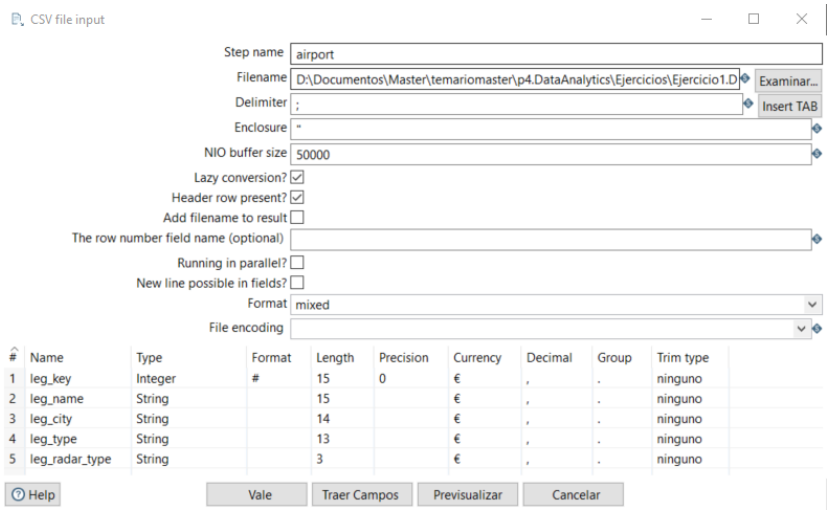
-Airport

In this table we have to unify the airport table and the airport city_state_table. To do that we have to import both CSV files and do a multiway merge join.

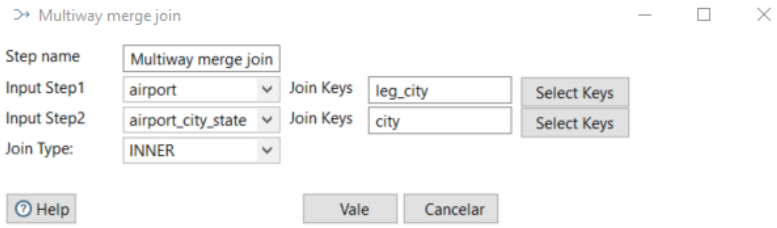


First we import de CSVs

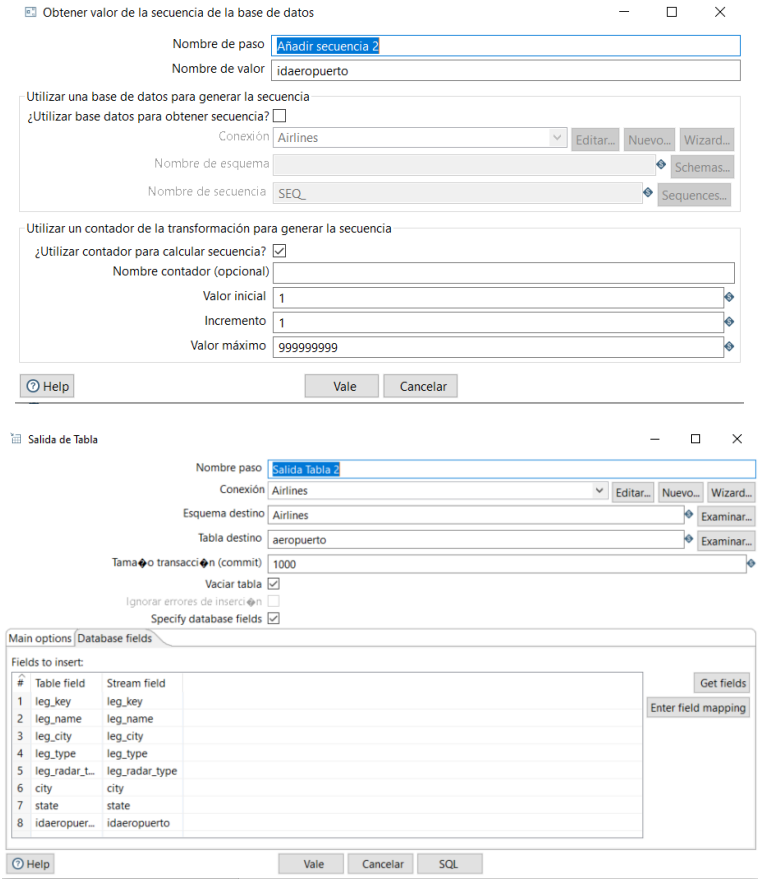




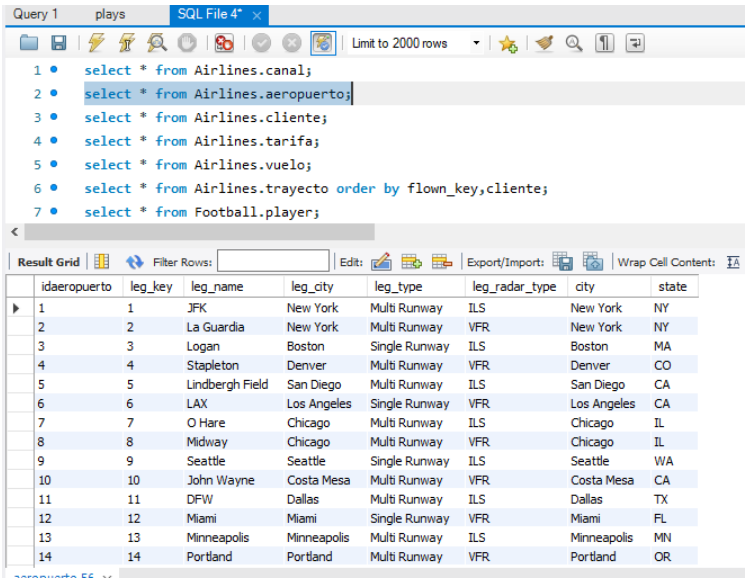
After that we have to choose where we want to unify the tables, in this case what we want is to unify them by city.



After that we add the sequence and select the output.



Now we are going to check the data in Airlines database.



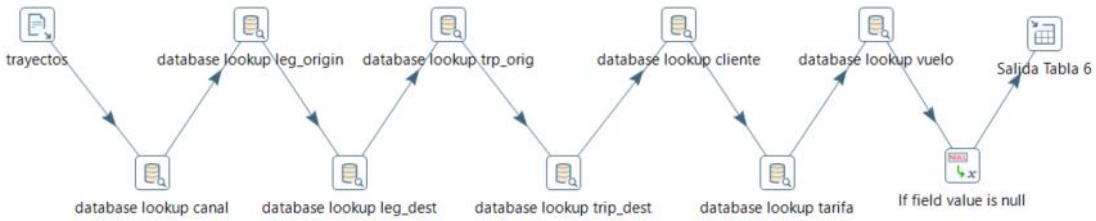
idaeropuerto	leg_key	leg_name	leg_city	leg_type	leg_radar_type	city	state
1	1	JFK	New York	Multi Runway	ILS	New York	NY
2	2	La Guardia	New York	Multi Runway	VFR	New York	NY
3	3	Logan	Boston	Single Runway	ILS	Boston	MA
4	4	Stapleton	Denver	Multi Runway	VFR	Denver	CO
5	5	Lindbergh Field	San Diego	Multi Runway	ILS	San Diego	CA
6	6	LAX	Los Angeles	Single Runway	VFR	Los Angeles	CA
7	7	O Hare	Chicago	Multi Runway	ILS	Chicago	IL
8	8	Midway	Chicago	Multi Runway	VFR	Chicago	IL
9	9	Seattle	Seattle	Single Runway	ILS	Seattle	WA
10	10	John Wayne	Costa Mesa	Multi Runway	VFR	Costa Mesa	CA
11	11	DFW	Dallas	Multi Runway	ILS	Dallas	TX
12	12	Miami	Miami	Single Runway	VFR	Miami	FL
13	13	Minneapolis	Minneapolis	Multi Runway	ILS	Minneapolis	MN
14	14	Portland	Portland	Multi Runway	VFR	Portland	OR

We can see that the merge join has been done perfectly.

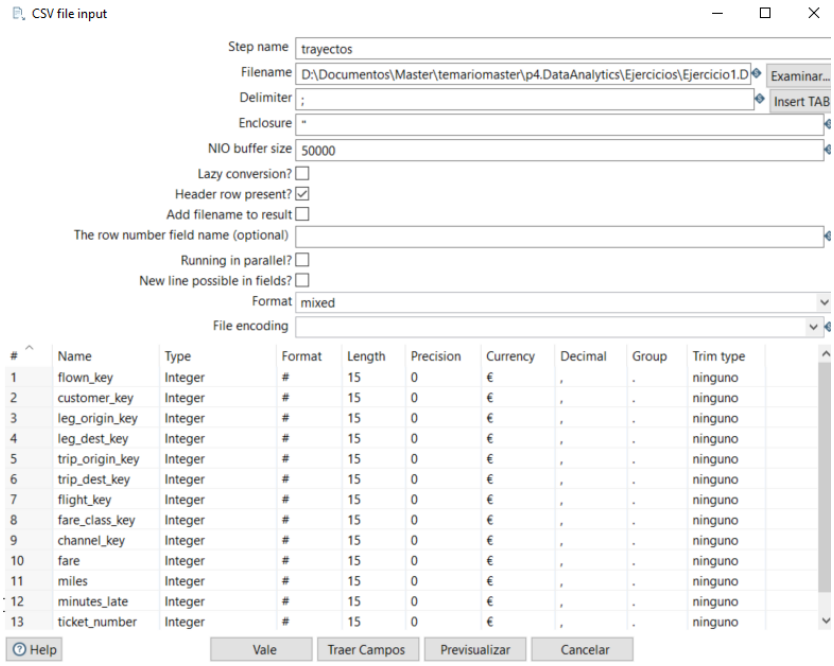
Now it is time to import the fact table.

-Frequent Flyer.

To import the fact table, I have preceded following the upcoming steps.

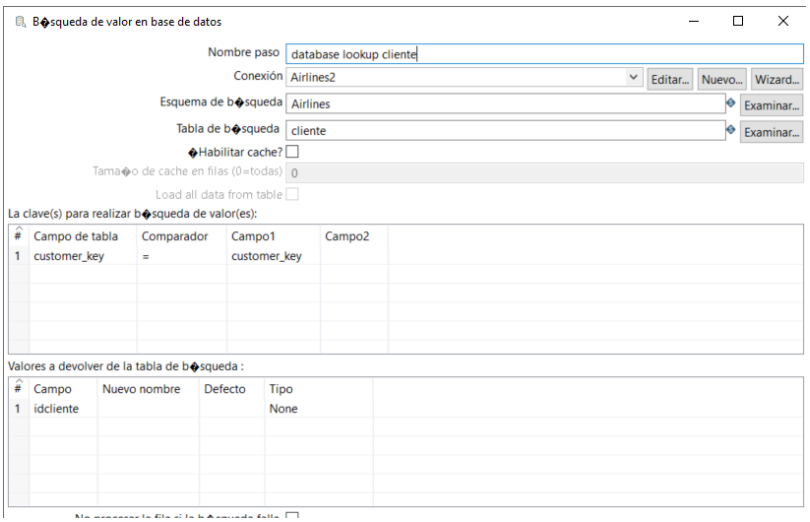


In the first place I have imported the CSV file that has the frequent flyer information.



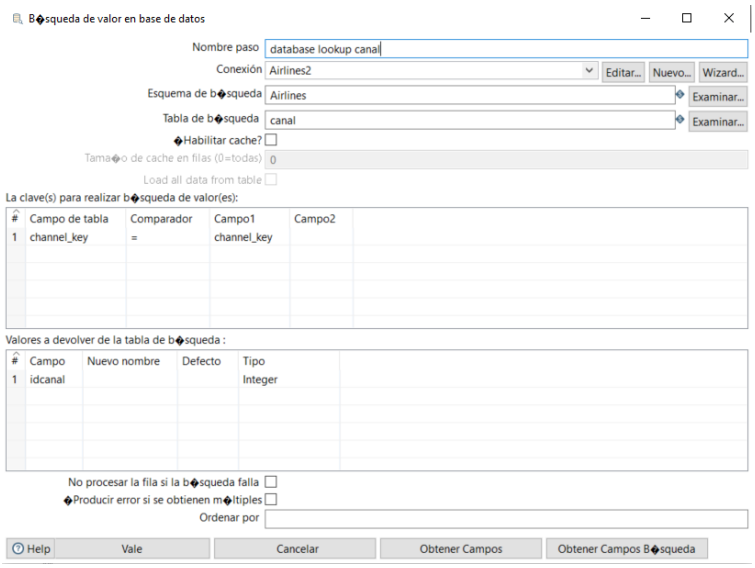
Now that I have the table to work with. I have done serval lookups tables to check that the key information in the fact table related to the id of the foreign key is equal to the primary key of the dimension table. If that happens it assigns the 'idvalue' from the dimension table to that value.

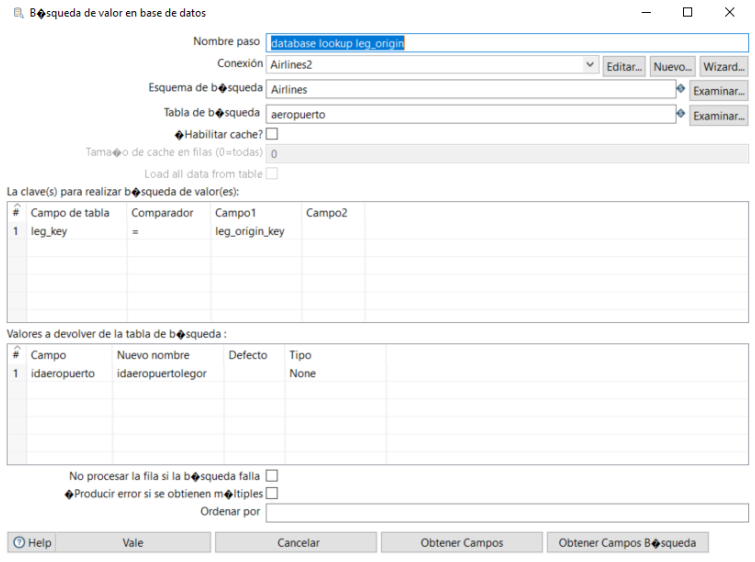
Lest see the client example to have a clear understanding.

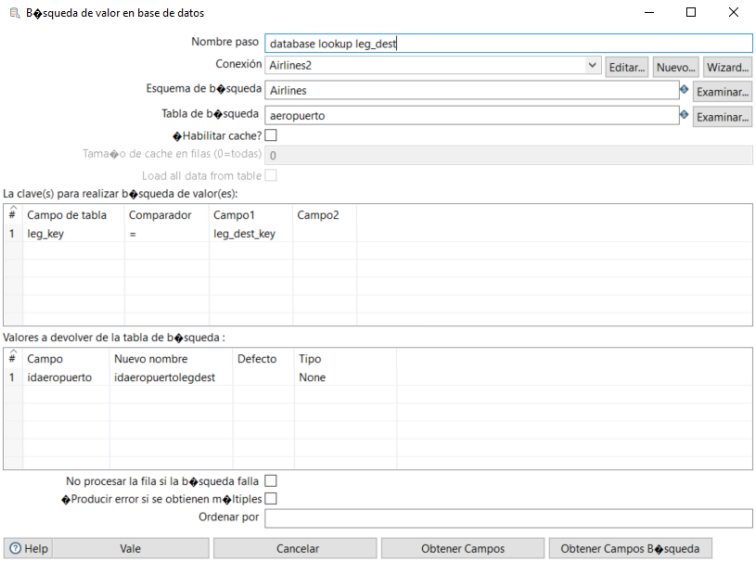


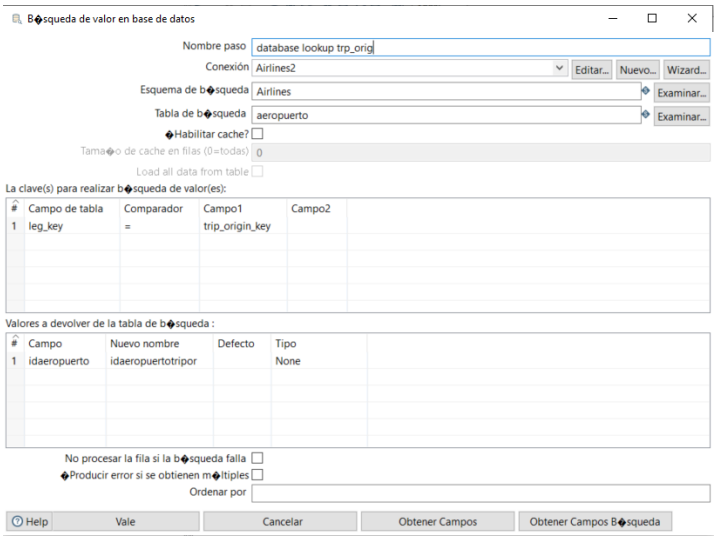
If we pay attention to the image I am checking if the 'customer_key' from frequentflyer is the same as the 'customer_key' from client table. If that happens, I assign the value 'idcliente' to the output of the fact table.

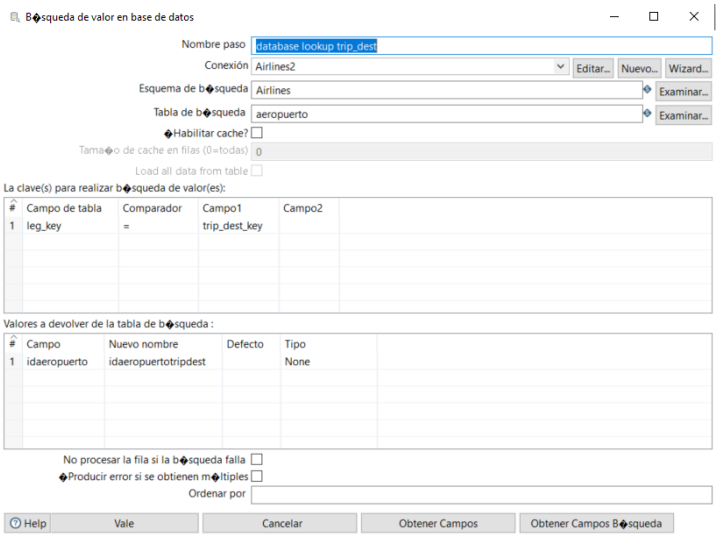
I have done that for every foreign key of the fact table so ill just show the images doing that.

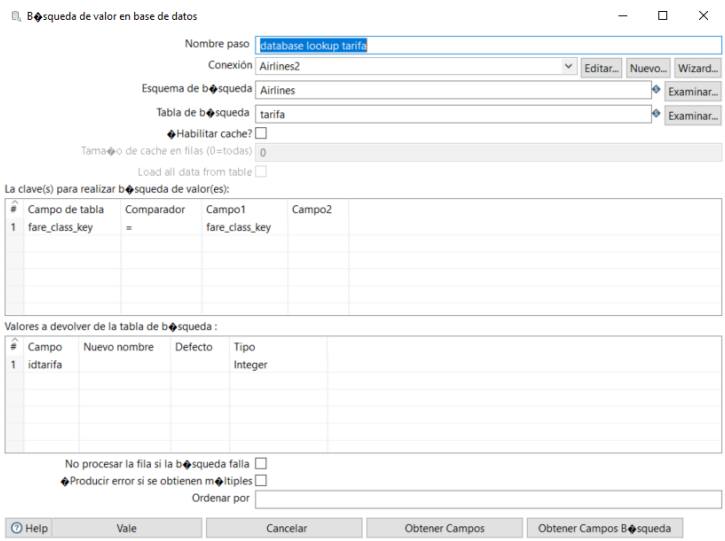

B&#x0000;squeda de valor en base de datos
Nombre paso
database lookup canal
Conexión
Airlines2
Editar...
Nuevo...
Wizard...
Esquema de b&#x0000;squeda
Airlines
Examinar...
Tabla de b&#x0000;squeda
canal
Examinar...
&#x0000;Habilitar cache?
Tama&#x0000;o de cache en filas (0=todas)
0
Load all data from table
La clave(s) para realizar b&#x0000;squeda de valor(es):
Campo de tabla Comparador Campo1 Campo2
1 channel_key = channel_key
Valores a devolver de la tabla de b&#x0000;squeda :
Campo Nuevo nombre Defecto Tipo
1 idcanal Integer
No procesar la fila si la b&#x0000;squeda falla
&#x0000;Producir error si se obtienen m&#x0000;ltiples
Ordenar por
Help
Vale
Cancelar
Obtener Campos
Obtener Campos B&#x0000;squeda

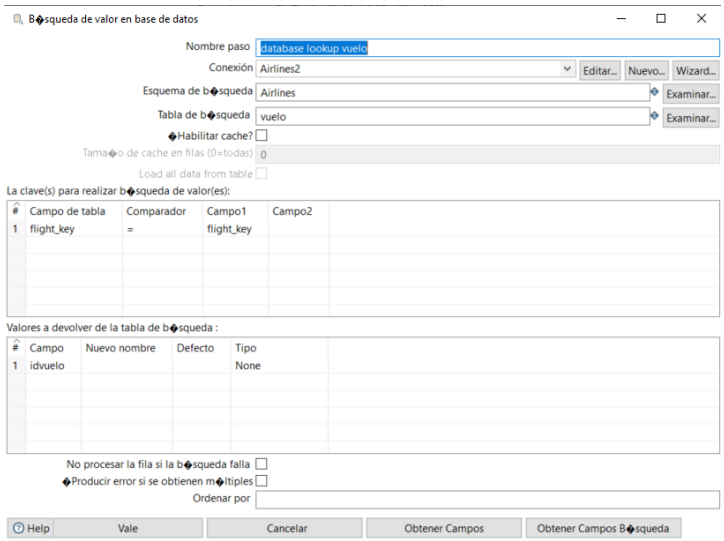

B&#x0000;squeda de valor en base de datos
Nombre paso
database lookup leg_origin
Conexión
Airlines2
Editar...
Nuevo...
Wizard...
Esquema de b&#x0000;squeda
Airlines
Examinar...
Tabla de b&#x0000;squeda
aeropuerto
Examinar...
&#x0000;Habilitar cache?
Tama&#x0000;o de cache en filas (0=todas)
0
Load all data from table
La clave(s) para realizar b&#x0000;squeda de valor(es):
Campo de tabla Comparador Campo1 Campo2
1 leg_key = leg_origin_key
Valores a devolver de la tabla de b&#x0000;squeda :
Campo Nuevo nombre Defecto Tipo
1 idaeropuerto idaeropuertolegor None
No procesar la fila si la b&#x0000;squeda falla
&#x0000;Producir error si se obtienen m&#x0000;ltiples
Ordenar por
Help
Vale
Cancelar
Obtener Campos
Obtener Campos B&#x0000;squeda


B&#x0000;squeda de valor en base de datos
Nombre paso
database lookup leg_dest
Conexión
Airlines2
Editar...
Nuevo...
Wizard...
Esquema de b&#x0000;squeda
Airlines
Examinar...
Tabla de b&#x0000;squeda
aeropuerto
Examinar...
&#x0000;Habilitar cache?
Tama&#x0000;o de cache en filas (0=todas)
0
Load all data from table
La clave(s) para realizar b&#x0000;squeda de valor(es):
Campo de tabla Comparador Campo1 Campo2
1 leg_key = leg_dest_key
Valores a devolver de la tabla de b&#x0000;squeda :
Campo Nuevo nombre Defecto Tipo
1 idaeropuerto idaeropuertolegdest None
No procesar la fila si la b&#x0000;squeda falla
&#x0000;Producir error si se obtienen m&#x0000;ltiples
Ordenar por
Help
Vale
Cancelar
Obtener Campos
Obtener Campos B&#x0000;squeda


B&#65533;squeda de valor en base de datos
Nombre paso
database lookup trp_orig
Conexión
Airlines2
Editar...
Nuevo...
Wizard...
Esquema de b&#65533;squeda
Airlines
Examinar...
Tabla de b&#65533;squeda
aeropuerto
Examinar...
&#65533;Habilitar cache?
Tama&#65533;o de cache en filas (0=todas)
0
Load all data from table
La clave(s) para realizar b&#65533;squeda de valor(es):
Campo de tabla Comparador Campo1 Campo2
1 leg_key = trip_origin_key
Valores a devolver de la tabla de b&#65533;squeda :
Campo Nuevo nombre Defecto Tipo
1 idaeropuerto idaeropuertotripor None
No procesar la fila si la b&#65533;squeda falla
&#65533;Producir error si se obtienen m&#65533;ltiples
Ordenar por
Help
Vale
Cancelar
Obtener Campos
Obtener Campos B&#65533;squeda


B&#65533;squeda de valor en base de datos
Nombre paso
database lookup trip_dest
Conexión
Airlines2
Editar...
Nuevo...
Wizard...
Esquema de b&#65533;squeda
Airlines
Examinar...
Tabla de b&#65533;squeda
aeropuerto
Examinar...
&#65533;Habilitar cache?
Tama&#65533;o de cache en filas (0=todas)
0
Load all data from table
La clave(s) para realizar b&#65533;squeda de valor(es):
Campo de tabla Comparador Campo1 Campo2
1 leg_key = trip_dest_key
Valores a devolver de la tabla de b&#65533;squeda :
Campo Nuevo nombre Defecto Tipo
1 idaeropuerto idaeropuertotripdest None
No procesar la fila si la b&#65533;squeda falla
&#65533;Producir error si se obtienen m&#65533;ltiples
Ordenar por
Help
Vale
Cancelar
Obtener Campos
Obtener Campos B&#65533;squeda

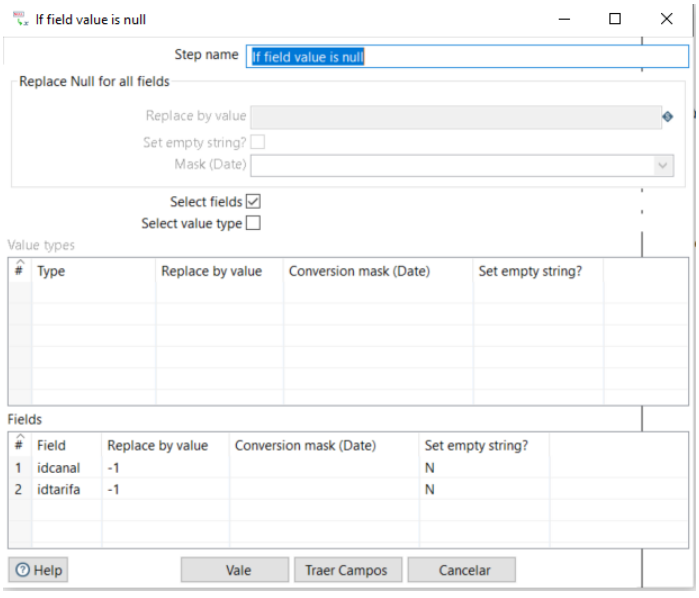

B&#65533;squeda de valor en base de datos
Nombre paso
database lookup tarifa
Conexión
Airlines2
Editar...
Nuevo...
Wizard...
Esquema de b&#65533;squeda
Airlines
Examinar...
Tabla de b&#65533;squeda
tarifa
Examinar...
&#65533;Habilitar cache?
Tama&#65533;o de cache en filas (0=todas)
0
Load all data from table
La clave(s) para realizar b&#65533;squeda de valor(es):
Campo de tabla Comparador Campo1 Campo2
1 fare_class_key = fare_class_key
Valores a devolver de la tabla de b&#65533;squeda :
Campo Nuevo nombre Defecto Tipo
1 idtarifa Integer
No procesar la fila si la b&#65533;squeda falla
&#65533;Producir error si se obtienen m&#65533;ltiples
Ordenar por
Help
Vale
Cancelar
Obtener Campos
Obtener Campos B&#65533;squeda



Now that we have done all the lookups, if you remember, in the channel and fare tables we created an additional row to store the data we had in the fact table that did not match the data we have in the dimension table.

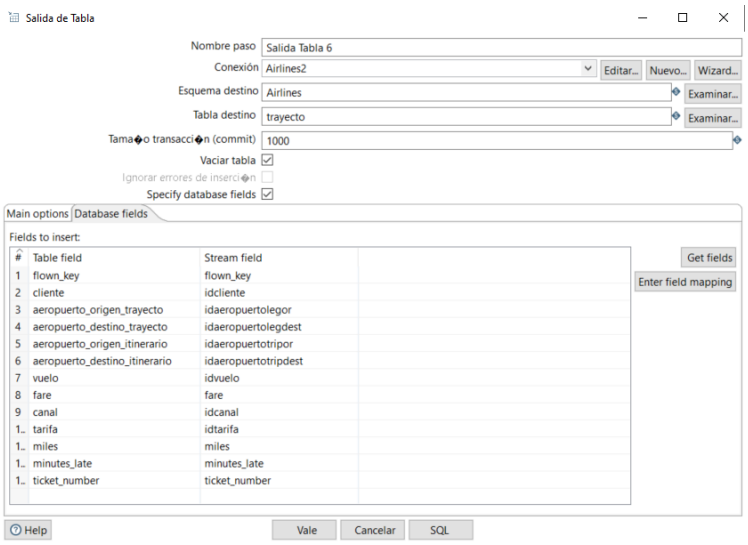
For example, when we do a lookup to that kind o value, if we are comparing and we get a fare_key of 4 in the dimension table and we get a fare_key of 19 in the fact table, to that value the pentaho data integration its going to assign a null in the key field.

That is why we use the block ‘if null value’.

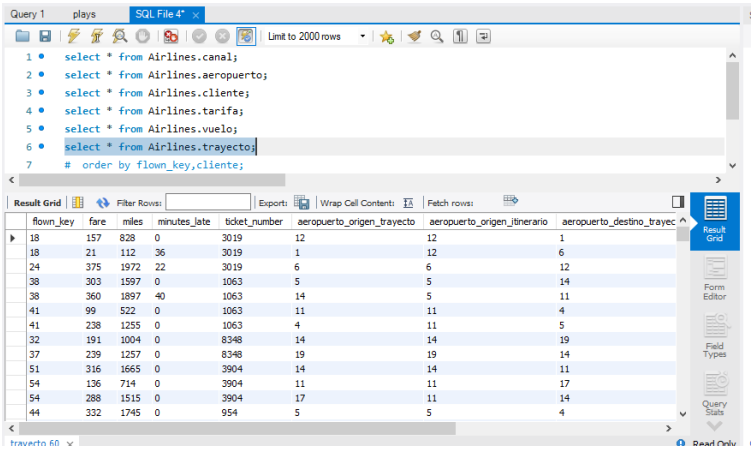


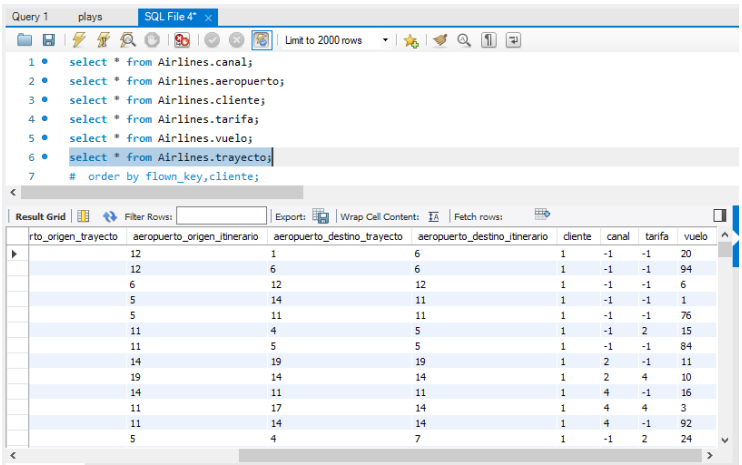
With this we say, if the field ‘idcanal’ or ‘idtarifa’ has a null value assign a -1 to that field, that way we have that kind of data localized and we don’t have null values.

To sum up the data integration we have to upload the processed data to the fact table in the Airlines dataset. To do that we are going to use the block we have been using before.



To check that the data has been uploaded correctly we are going to do a SQL query.





In here we can check the -1 value for the channel and fare bad values.