

Laboratorio #4 ETL

Evaluación de herramientas ETL

Autores:

Miguel Ángel Acosta (201914976)

Andrés Felipe Rincón (201914118)

Ángela Liliana Jiménez (201912941)

Tabla de Contenidos

Bono #1 Preparación de datos	2
Bono #2 Datos Completos	4
Diagrama de alto nivel del proceso de ETL	5
Documentación del proceso y las transformaciones realizadas - Modificaciones a los esquemas en Spoon	5
Resultados Spoon	16
Documentación del proceso y las transformaciones realizadas - Modificaciones a los esquemas en BigQuery	18
Resultados BigQuery	18
Documentación del proceso y las transformaciones realizadas - Modificaciones a los esquemas en la 3ra herramienta	18
Resultados 3ra herramienta	18
Respuestas a las preguntas del laboratorio	18
Comparación de herramientas	18

Bono #1 Preparación de datos

El código python necesario para el preprocesamiento de datos está en Preparacion_Datos.ipynb, para crear los archivos con los datos pre procesados, es necesario que los archivos se encuentren en la carpeta /data/Muestra o /data/Completos, esto con relación al notebook, es decir, la idea es que al momento de ejecutar el notebook los archivos estén con la siguiente distribución:



Los archivos serán creados en su respectiva carpeta, es decir, si queremos crear los datos preprocesados de los csv de Muestra, estos serán creados en la carpeta Muestra, con el mismo nombre que el csv original y el sufijo *_preprocessed*.

En In[2] se puede especificar si el preprocesamiento se realizará para Muestra o para Completos

```
In [2]: #Type (Muestra/ for "muestra" and Completos/ for "completos")
        tipo = "Completos/"
        #tipo = "Muestra/"
```

Con respecto al preprocesamiento, encontramos lo siguiente:

- Inicialmente recibimos la siguiente cantidad de entradas de los csv originales:

CSV	Número de entradas
City	116295
Customer	403
Employee	213
Stockitem	672
Date_Table	1461
Fact_Order	231412

- Varias dimensiones tienen entradas que son completamente nulas, por lo que se decide remover estas entradas. Es importante resaltar que si eliminamos una entrada de una dimensión tenemos que garantizar la consistencia en la tabla de hechos, a continuación damos un ejemplo con la dimensión city:

```
df_order = df_order[df_order.City_Key.isin(df_city.City_Key)]
```

En este punto hemos removido entradas del dataframe `df_city`, (dimensión city) ahora nos quedaremos con las entradas de `df_order` (tabla de hechos) cuya llave foránea correspondiente a ciudad se encuentre en la dimensión ciudad, de tal forma eliminamos las entradas de la tabla de hechos que se refirieran a una entrada en la dimensión city que hayamos eliminado.

- En la dimensión Stockitem tenemos muchos valores nulos para los valores de color, brand y size_val (90%, 47%, 30% de valores nulos respectivamente) retirar las entradas que tengan valores nulos en este caso puede ser perjudicial por que perderíamos una cantidad de datos considerable, no solo en esta dimensión sino en la tabla de hechos, por lo que se decide llenar estos valores nulos por "unknown" y se procede a eliminar las entradas con registros nulos (ya sin contar estos valores anteriores como nulos)
- Se elimina el valor `WWI_Stock_Item_ID` de todas las entradas de la dimensión Stockitem porque este valor no existe en el modelo

- Finalmente después de retirar las inconsistencias tenemos la siguiente cantidad de entradas:

Dimensión	Número de entradas
City	116294
Customer	402
Employee	212
Stockitem	671
Date_Table	1461
Fact_Order	130901

Bono #2 Datos Completos

Para el proceso de ETL en las distintas herramientas usamos los datos Completos (luego de preprocesamiento), se puede evidenciar esto en los resultados de las consultas para cada herramienta, en donde se ve como la cantidad de datos corresponde a lo reportado anteriormente.

Particularmente para pentaho spoon Usamos la forma de carga de datos normal. Existía la forma de cargar los datos con la herramienta bulkload, que básicamente consiste en cambiar el proceso de transformación a uno como el siguiente (ponemos como ejemplo la dimensión city):



Y trabajamos en *CSV City* y en *PostgreSQL bulk loader*

The screenshot shows the 'CSV file input' dialog box in Pentaho Spoon. The 'Step name' is 'CSV City'. The 'Filename' is set to '\$[Internal.Entry.Current.Directory]/data/Completo/dimension_city_preprocessed.csv'. The 'Delimiter' is ',' and the 'Enclosure' is '"'. The 'NIO buffer size' is 50000. The 'Lazy conversion?' checkbox is checked. The 'Header row present?' checkbox is checked. The 'Add filename to result' checkbox is unchecked. The 'The row number field name (optional)' field is empty. The 'Running in parallel?' checkbox is unchecked. The 'New line possible in fields?' checkbox is unchecked. The 'Format' is set to 'mixed'. The 'File encoding' is set to 'UTF-8'. Below these settings is a table with 10 columns: #, Name, Type, Format, Length, Precision, Currency, Decimal, Group, and Trim type. The table contains 9 rows of data for various fields like City_Key, City, State_Province, Country, Continent, Sales_Territory, Region, Subregion, and Latest_Recorded_Population. At the bottom of the dialog are buttons for '? Help', 'OK', 'Get Fields', 'Preview', and 'Cancel'.

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	City_Key	Integer	#	15	0	\$,	.	none
2	City	String		17		\$,	.	none
3	State_Province	String		26		\$,	.	none
4	Country	String		13		\$,	.	none
5	Continent	String		13		\$,	.	none
6	Sales_Territory	String		14		\$,	.	none
7	Region	String		8		\$,	.	none
8	Subregion	String		16		\$,	.	none
9	Latest_Recorded_Population	Integer	#	15	0	\$,	.	none

PostgreSQL bulk loader

Step name: PostgreSQL bulk loader

Connection: lab4_wwwi [Edit...] [New...] [Wizard...]

Target schema: public

Target table: city [Browse...]

Load action: Insert

DB Name Override:

Enclosure: "

Delimiter: ;

Stop on error: ☐

Fields to load:

#	Table field	Stream field	Date Mask
1	City_Key	City_Key	
2	City	City	
3	State_Province	State_Province	
4	Country	Country	
5	Continent	Continent	
6	Sales_Territory	Sales_Territory	
7	Region	Region	
8	Subregion	Subregion	
9	Latest_Recorded_Population	Latest_Recorded_Population	

[Get fields] [Edit mapping]

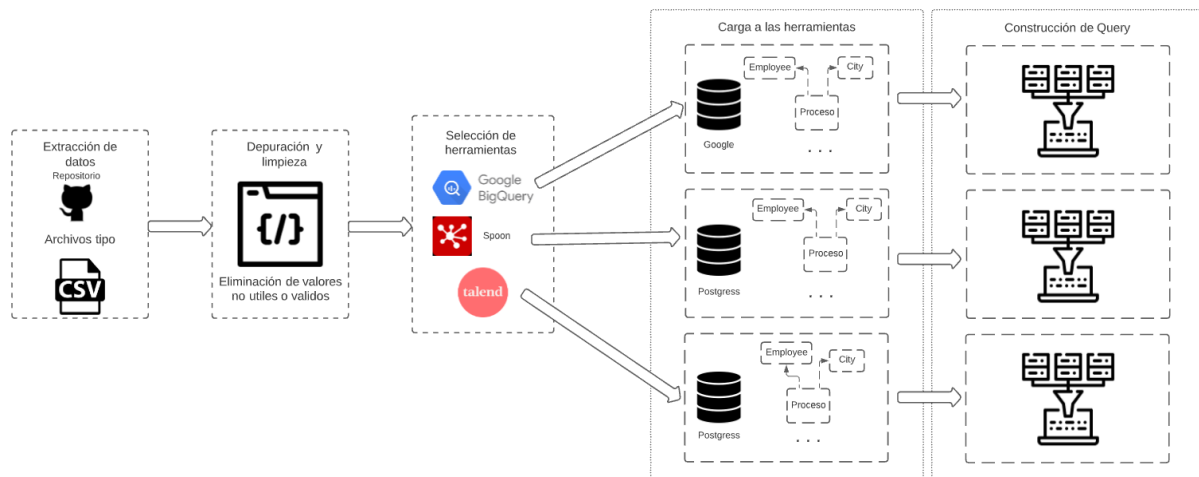
[?] Help [OK] [Cancel] [SQL]

Realizar el proceso de ETL con este esquema demostró cargar los datos mucho más rápido que el esquema normal, pero tiene un problema que causó que descartamos este método y es que bulkload es un método de carga de datos que solo realiza operaciones insert/truncate, es decir o inserta los datos o borra todo para insertar datos, esto es problemático porque:

- Para la primera carga no hay problemas, pero para una segunda ejecución del job, las llaves chocaran con las existentes.
- Eso se puede arreglar eliminando los datos, sin embargo por la existencia de relaciones entre tablas es necesario revisar las reglas de estas llaves.
- Luego de lo mencionado, que son principalmente, problemas operativos la razón por la que principalmente lo descartamos es una razón de negocio y es que estamos hablando de eliminar todos los registros de la base de datos, lo que no es algo aceptable en este contexto.

En resumen, estamos usando los datos Completos, los usamos con la herramienta de carga normal, que aunque carga un poco más lento que bulk load funciona, y no presenta los problemas antes mencionados.

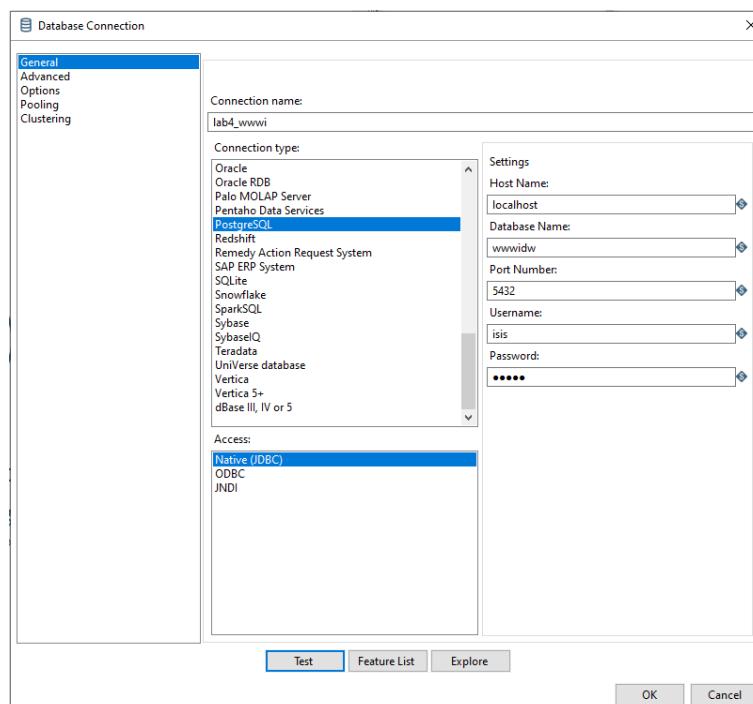
Diagrama de alto nivel del proceso de ETL



Documentación del proceso y las transformaciones realizadas - Modificaciones a los esquemas en Spoon

Creamos una base de datos en postgresql y nos conectamos a ella en pentaho

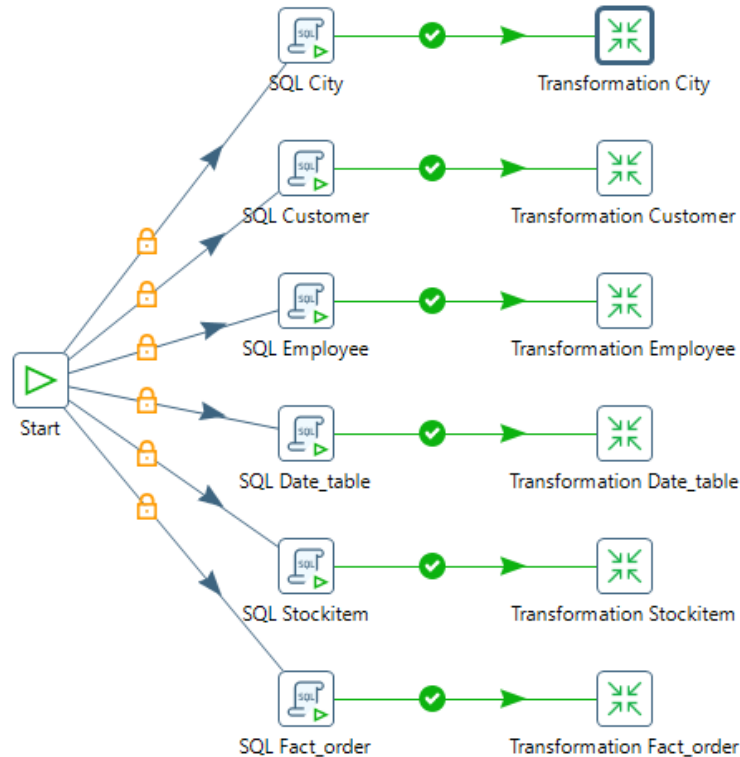
```
postgres=# CREATE USER isis;
CREATE ROLE
postgres=# CREATE DATABASE wwwidw;
CREATE DATABASE
postgres=# ALTER USER isis WITH PASSWORD '12345';
ALTER ROLE
postgres=# \q
```



Luego creamos un job *job_dimensiones* y creamos un esquema Start → SQL → Transformation, como se ve a continuación para *date_table*:

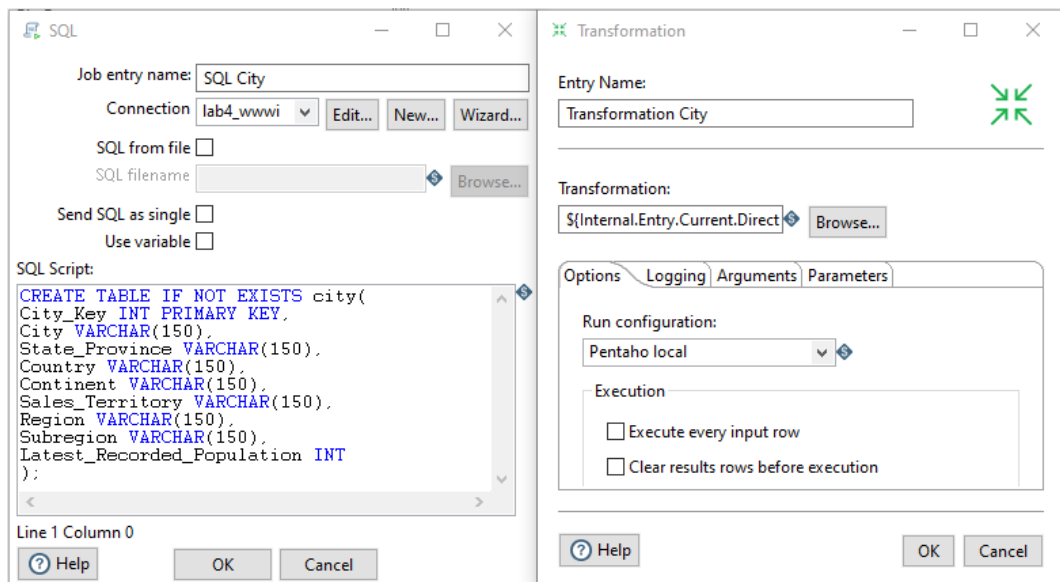


Para al final tener el siguiente esquema:



Trabajo para City:

Trabajamos en el sql *SQL City* y en la transformación *Transformation City*



Creamos la transformación *Transformation City*



Trabajamos en *CSV City* y en *Insert / update*

SQL

Job entry name: SQL Customer

Connection

lab4_wwwi

Edit...

New...

Wizard...

SQL from file

SQL filename

Browse...

Send SQL as single

Use variable

SQL Script:

```
CREATE TABLE IF NOT EXISTS customer(
Customer_Key INT PRIMARY KEY,
Customer VARCHAR(150),
Bill_To_Customer VARCHAR(150),
Category VARCHAR(150),
Buying_Group VARCHAR(150),
Primary_Contact VARCHAR(150),
Postal_Code INT
);
```

Line 1 Column 0

Help

OK

Cancel

Transformation

Entry Name:

Transformation Customer

Transformation:

Internal.Entry.Current.Directory

Browse...

Options

Logging

Arguments

Parameters

Run configuration:

Pentaho local

Execution

Execute every input row

Clear results rows before execution

Help

OK

Cancel

Creamos la transformación *Transformation Customer*

```

graph LR
    A[CSV customer] --> B[Insert / update]
  
```

Trabajamos en *CSV Customer* y en *Insert / update*

CSV file input

Step name

CSV customer

Filename

Internal.Entry.Current.Directory/data/Completo/dimension_customer_preprocessed.c

Browse...

Delimiter

,

Insert TAB

Enclosure

"

NIO buffer size

50000

Lazy conversion?

Header row present?

Add filename to result

The row number field name (optional)

Running in parallel?

New line possible in fields?

Format

mixed

File encoding

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Customer_Key	Integer	#	15	0	\$,	.	none
2	Customer	String		38		\$,	.	none
3	Bill_To_Customer	String		27		\$,	.	none
4	Category	String		12		\$,	.	none
5	Buying_Group	String		13		\$,	.	none
6	Primary_Contact	String		19		\$,	.	none
7	Postal_Code	Integer	#	15	0	\$,	.	none

Help

OK

Get Fields

Preview

Cancel

Insert / update

Step name: Insert / update

Connection: lab4_wwwi [Edit... New... Wizard...]

Target schema: public [Browse...]

Target table: customer [Browse...]

Commit size: 100

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	Customer_Key	=	Customer_Key	

[Get fields]

Update fields:

#	Table field	Stream field	Update
1	Customer_Key	Customer_Key	Y
2	Customer	Customer	Y
3	Bill_To_Customer	Bill_To_Customer	Y
4	Category	Category	Y
5	Buying_Group	Buying_Group	Y
6	Primary_Contact	Primary_Contact	Y
7	Postal_Code	Postal_Code	Y

[Get update fields] [Edit mapping]

[?] Help [OK] [Cancel] [SQL]

Trabajo para Employee:

Trabajamos en el sql *SQL Employee* y en la transformación *Transformation Employee*

SQL

Job entry name: SQL Employee

Connection: lab4_wwwi [Edit... New... Wizard...]

SQL from file ☐

SQL filename: [Browse...]

Send SQL as single ☐

Use variable ☐

SQL Script:

```
CREATE TABLE IF NOT EXISTS employee(
Employee_Key INT PRIMARY KEY,
Employee VARCHAR(150),
Preferred_Name VARCHAR(150),
Is_Salesperson BOOLEAN
);
```

Line 1 Column 0

[?] Help [OK] [Cancel]

Transformation

Entry Name: Transformation Employee

Transformation: \${Internal.Entry.Current.Direct} [Browse...]

Options [Logging] [Arguments] [Parameters]

Run configuration: Pentaho local

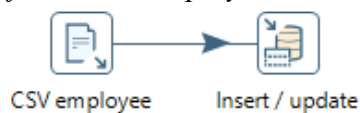
Execution

☐ Execute every input row

☐ Clear results rows before execution

[?] Help [OK] [Cancel]

Creamos la transformación *Transformation Employee*



Trabajamos en *CSV Employee* y en *Insert / update*

CSV file input

Step name: CSV employee

Filename: \${Internal.Entry.Current.Directory}/data/Completo/dimension_employee_preprocessed. Browse...

Delimiter: Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Employee_Key	Integer	#	15	0	\$,	.	none
2	Employee	String		18		\$,	.	none
3	Preferred_Name	String		8		\$,	.	none
4	Is_Salesperson	Boolean				\$,	.	none

Help OK Get Fields Preview Cancel

Insert / update

Step name: Insert / update

Connection: lab4_wwwi Edit... New... Wizard...

Target schema: public Browse...

Target table: employee Browse...

Commit size: 100

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	Employee_Key	=	Employee_Key	

Get fields

Update fields:

#	Table field	Stream field	Update
1	Employee_Key	Employee_Key	Y
2	Employee	Employee	Y
3	Preferred_Name	Preferred_Name	Y
4	Is_Salesperson	Is_Salesperson	Y

Get update fields Edit mapping

Help OK Cancel SQL

Trabajo para Date_Table:

Trabajamos en el sql *SQL Date_table* y en la transformación *Transformation Date_table*

SQL

Job entry name: SQL Date_table

Connection

lab4_wwwi

Edit...

New...

Wizard...

SQL from file

SQL filename

Browse...

Send SQL as single

Use variable

SQL Script:

```
CREATE TABLE IF NOT EXISTS date_table(
Date_key timestamp PRIMARY KEY,
Day_Number INT,
Day_val INT,
Month_val VARCHAR(50),
Short_Month VARCHAR(50),
Calendar_Month_Number INT,
Calendar_Year INT,
Fiscal_Month_Number INT,
Fiscal_Year INT
);
```

Line 1 Column 0

Help

OK

Cancel

Transformation

Entry Name:

Transformation Date_table

Transformation:

\$(Internal.Entry.Current.Directi

Browse...

Options

Logging

Arguments

Parameters

Run configuration:

Pentaho local

Execution

Execute every input row

Clear results rows before execution

Help

OK

Cancel

Creamos la transformación *Transformation Date_Table*

```

graph LR
    A[Document Icon] --> B[Database Icon]
    A --- C[CSV Date_table]
    B --- D[Insert / update]
  
```

Trabajamos en *CSV Date_Table* y en *Insert / update*

CSV file input

Step name

CSV Date_table

Filename

\$(Internal.Entry.Current.Directory)/data/Completo/dimension_date_preprocessed.csv

Browse...

Delimiter

,

Insert TAB

Enclosure

"

NIO buffer size

50000

Lazy conversion?

Header row present?

Add filename to result

The row number field name (optional)

Running in parallel?

New line possible in fields?

Format

mixed

File encoding

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Date_key	Date	yyyy-MM-dd			\$,	.	none
2	Day_Number	Integer	#	15	0	\$,	.	none
3	Day_val	Integer	#	15	0	\$,	.	none
4	Month_val	String		8		\$,	.	none
5	Short_Month	String		3		\$,	.	none
6	Calendar_Month_Number	Integer	#	15	0	\$,	.	none
7	Calendar_Year	Integer	#	15	0	\$,	.	none
8	Fiscal_Month_Number	Integer	#	15	0	\$,	.	none
9	Fiscal_Year	Integer	#	15	0	\$,	.	none

Help

OK

Get Fields

Preview

Cancel

Insert / update

Step name: Insert / update

Connection: lab4_wwwi

Target schema: public

Target table: date_table

Commit size: 100

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	Date_key	=	Date_key	

Update fields:

#	Table field	Stream field	Update
1	Date_key	Date_key	Y
2	Day_Number	Day_Number	Y
3	Day_val	Day_val	Y
4	Month_val	Month_val	Y
5	Short_Month	Short_Month	Y
6	Calendar_Month_Number	Calendar_Month_Number	Y
7	Calendar_Year	Calendar_Year	Y
8	Fiscal_Month_Number	Fiscal_Month_Number	Y
9	Fiscal_Year	Fiscal_Year	Y

Buttons: ? Help, OK, Cancel, SQL

Trabajo para Stockitem:

Trabajamos en el sql *SQL Stockitem* y en la transformación *Transformation Stockitem*

SQL

Job entry name: SQL Stockitem

Connection: lab4_wwwi

SQL from file: ☐

SQL filename:

Send SQL as single: ☐

Use variable: ☐

SQL Script:

```
CREATE TABLE IF NOT EXISTS stockitem(
Stock_Item_Key INT PRIMARY KEY,
Stock_Item VARCHAR(200),
Color VARCHAR(50),
Selling_Package VARCHAR(50),
Buying_Package VARCHAR(50),
Brand VARCHAR(50),
Size_val VARCHAR(50),
Lead_Time_Days INT,
Quantity_Per_Outer INT,
Is_Chiller_Stock BOOLEAN,
Tax_Rate DECIMAL,
Unit_Price DECIMAL,
Recommended_Retail_Price DECIMAL,
Typical_Weight_Per_Unit DECIMAL
);
```

Line 1 Column 0

Buttons: ? Help, OK, Cancel

Transformation

Entry Name: Transformation Stockitem

Transformation: \${Internal.Entry.Current.Direct}

Options: Logging Arguments Parameters

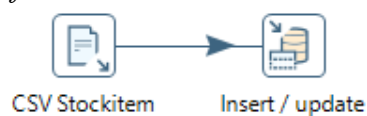
Run configuration: Pentaho local

Execution:

- ☐ Execute every input row
- ☐ Clear results rows before execution
- ☐ Clear results files before execution
- ☒ Wait for remote transformation to complete

Buttons: ? Help, OK, Cancel

Creamos la transformación *Transformation Stockitem*



Trabajamos en *CSV Stockitem* y en *Insert / update*

CSV file input

Step name: CSV Stockitem

Filename: F:\Semestre 6\Semestre 6 Inteligencia de Negocios\ISIS-3301-L4\data\Completos\dimen: Browse...

Delimiter: , Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Stock_Item_Key	Integer	#	15	0	\$,	.	none
2	Stock_Item	String		58		\$,	.	none
3	Color	String		11		\$,	.	none
4	Selling_Package	String		6		\$,	.	none
5	Buying_Package	String		6		\$,	.	none
6	Brand	String		7		\$,	.	none
7	Size_val	String		13		\$,	.	none
8	Lead_Time_Days	Integer	#	15	0	\$,	.	none
9	Quantity_Per_Outer	Integer	#	15	0	\$,	.	none
10	Is_Chiller_Stock	Boolean				\$,	.	none
11	Tax_Rate	Number	#,.	15	0	\$,	.	none
12	Unit_Price	Number	#,.	15	0	\$,	.	none
13	Recommended_Retail_Price	Number	#,.	15	0	\$,	.	none
14	Typical_Weight_Per_Unit	Number	#,.	15	0	\$,	.	none

Help OK Get Fields Preview Cancel

Insert / update

Step name: Insert / update

Connection: lab4_wwwi Edit... New... Wizard...

Target schema: public Browse...

Target table: stockitem Browse...

Commit size: 100

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	Stock_Item_Key	=	Stock_Item_Key	

Get fields

Update fields:

#	Table field	Stream field	Update
1	Stock_Item_Key	Stock_Item_Key	Y
2	Stock_Item	Stock_Item	Y
3	Color	Color	Y
4	Selling_Package	Selling_Package	Y
5	Buying_Package	Buying_Package	Y
6	Brand	Brand	Y
7	Size_val	Size_val	Y
8	Lead_Time_Days	Lead_Time_Days	Y
9	Quantity_Per_Outer	Quantity_Per_Outer	Y
10	Is_Chiller_Stock	Is_Chiller_Stock	Y
11	Tax_Rate	Tax_Rate	Y
12	Unit_Price	Unit_Price	Y
13	Recommended_Retail_Price	Recommended_Retail_Price	Y
14	Typical_Weight_Per_Unit	Typical_Weight_Per_Unit	Y

Get update fields Edit mapping

Help OK Cancel SQL

Trabajo para Fact_order:

Trabajamos en el sql *SQL Fact_order* y en la transformación *Transformation Fact_order*

SQL

Job entry name: SQL Fact_order

Connection: lab4_wwwi

Edit...

New...

Wizard...

SQL from file

SQL filename:

Browse...

Send SQL as single statement?

Use variable substitution?

SQL Script:

```
CREATE TABLE IF NOT EXISTS fact_order(
Order_Key INT PRIMARY KEY,
City_Key INT REFERENCES city (city_key),
Customer_Key INT REFERENCES customer (customer_key),
Stock_Item_Key INT REFERENCES stockitem (stock_item_key),
Order_Date_Key DATE REFERENCES date_table (date_key),
Picked_Date_Key DATE REFERENCES date_table (date_key),
Salesperson_Key INT REFERENCES employee (employee_key),
Picker_Key INT REFERENCES employee (employee_key),
Package VARCHAR(50),
Quantity INT,
Unit_Price DECIMAL,
Tax_Rate DECIMAL,
Total_Excluding_Tax DECIMAL,
Tax_Amount DECIMAL,
Total_Including_Tax DECIMAL
);
```

Line 1 Column 0

Help

OK

Cancel

Transformation

Entry Name: Transformation Fact_order

Transformation:

\${Internal.Entry.Current.Directory}/Transl

Browse...

Options

Logging

Arguments

Parameters

Run configuration:

Pentaho local

Execution

Execute every input row

Clear results rows before execution

Clear results files before execution

Wait for remote transformation to complete

Follow local abort to remote transformation

Help

OK

Cancel

Creamos la transformación *Transformation Fact_order*

CSV fact_order → Insert / update

Trabajamos en *CSV Fact_order* y en *Insert / update*

CSV file input

Step name: CSV fact_order

Filename: \${Internal.Entry.Current.Directory}/data/Completo/fact_order_preprocessed.csv

Browse...

Delimiter: ,

Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion?

Header row present?

Add filename to result

The row number field name (optional):

Running in parallel?

New line possible in fields?

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Order_Key	Integer	#	15	0	\$,	.	none
2	City_Key	Integer	#	15	0	\$,	.	none
3	Customer_Key	Integer	#	15	0	\$,	.	none
4	Stock_Item_Key	Integer	#	15	0	\$,	.	none
5	Order_Date_Key	Date	yyyy-MM-dd			\$,	.	none
6	Picked_Date_Key	Date	yyyy-MM-dd			\$,	.	none
7	Salesperson_Key	Integer	#	15	0	\$,	.	none
8	Picker_Key	Integer	#	15	0	\$,	.	none
9	Package	String		6		\$,	.	none
10	Quantity	Integer	#	15	0	\$,	.	none
11	Unit_Price	Number	#,.	15	0	\$,	.	none
12	Tax_Rate	Number	#,.	15	0	\$,	.	none
13	Total_Excluding_Tax	Number	#,.	15	0	\$,	.	none
14	Tax_Amount	Number	#,.	15	0	\$,	.	none
15	Total_Including_Tax	Number	#,.	15	0	\$,	.	none

Help

OK

Get Fields

Preview

Cancel

Insert / update

Step name: Insert / update

Connection: lab4_wwwi

Target schema: public

Target table: fact_order

Commit size: 100

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	Order_Key	=	Order_Key	

Update fields:

#	Table field	Stream field	Update
1	Order_Key	Order_Key	Y
2	City_Key	City_Key	Y
3	Customer_Key	Customer_Key	Y
4	Stock_Item_Key	Stock_Item_Key	Y
5	Order_Date_Key	Order_Date_Key	Y
6	Picked_Date_Key	Picked_Date_Key	Y
7	Salesperson_Key	Salesperson_Key	Y
8	Picker_Key	Picker_Key	Y
9	Package	Package	Y
10	Quantity	Quantity	Y
11	Unit_Price	Unit_Price	Y
12	Tax_Rate	Tax_Rate	Y
13	Total_Excluding_Tax	Total_Excluding_Tax	Y
14	Tax_Amount	Tax_Amount	Y
15	Total_Including_Tax	Total_Including_Tax	Y

Buttons: Help, OK, Cancel, SQL

Resultados Spoon

Consultas para City:

wwwdw/postgres@PostgreSQL 13

Query Editor Query History Scratch Pad

```
1 select * from city;
```

Data Output Explain Messages Notifications

	city_key [PK] integer	city character varying (150)	state_province character varying (150)	country character varying (150)	continent character varying (150)	sales_territory character varying (150)	region character varying (150)	subregion character varying (150)	latest_recorded_population integer
1	1	Carrollton	New York	United States	North America	Mideast	Americas	Northern America	0
2	2	Carrollton	Virginia	United States	North America	Southeast	Americas	Northern America	4574
3	3	Carrollton	Illinois	United States	North America	Great Lakes	Americas	Northern America	2484
4	4	Carrollton	Missouri	United States	North America	Plains	Americas	Northern America	3784
5	5	Carrollton	Ohio	United States	North America	Great Lakes	Americas	Northern America	3241
6	6	Carrollton	Kentucky	United States	North America	Southeast	Americas	Northern America	3938
7	7	Carrollton	Georgia	United States	North America	Southeast	Americas	Northern America	24388
8	8	Carrollton	Alabama	United States	North America	Southeast	Americas	Northern America	1019
9	9	Carrollton	Mississippi	United States	North America	Southeast	Americas	Northern America	190
10	10	Carrollton	Texas	United States	North America	Southwest	Americas	Northern America	0
11	11	Carrollton	Maryland	United States	North America	Mideast	Americas	Northern America	0

Successfully run. Total query runtime: 264 msec. 116294 rows affected.

Consultas para Customer:

wwwidw/postgres@PostgreSQL 13

Query Editor

Query History

Scratch Pad

1

select * from customer;

Data Output

Explain

Messages

Notifications

	customer_key [PK] integer	customer character varying (150)	bill_to_customer character varying (150)	category character varying (150)	buying_group character varying (150)	primary_contact character varying (150)	postal_code integer
1	1	Tailsin Toys (Head Office)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Waldemar Fisar	90410
2	2	Tailsin Toys (Sylvanite- MT)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Lorena Cindric	90216
3	3	Tailsin Toys (Peeples Valley- AZ)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Bhaargav Rambhatla	90205
4	4	Tailsin Toys (Medicine Lodge- KS)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Daniel Roman	90152
5	5	Tailsin Toys (Gasport- NY)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Johanna Huitling	90261
6	6	Tailsin Toys (Jessie- ND)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Biswajeet Thakur	90298
7	7	Tailsin Toys (Frankewing- TN)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Kalidas Nadar	90761
8	8	Tailsin Toys (Bow Mar- CO)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Kanti Kotadia	90484
9	9	Tailsin Toys (Netcong- NJ)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Sointu Aalto	90129
10	10	Tailsin Toys (Wimbledon- ND)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Sidd	
11	11	Tailsin Toys (Newell- PA)	Tailsin Toys (Head Office)	Novelty Shop	Tailsin Toys	Elise Joun	90106

✓ Successfully run. Total query runtime: 257 msec. 402 rows affected.

Consultas para Employee:

wwwidw/postgres@PostgreSQL 13

Query Editor

Query History

Scratch Pad

1

select * from employee;

Data Output

Explain

Messages

Notifications

	employee_key [PK] integer	employee character varying (150)	preferred_name character varying (150)	is_salesperson boolean
1	1	Lily Code	Lily	true
2	2	Isabella Rupp	Isabella	false
3	3	Ethan Onslow	Ethan	false
4	4	Amy Trefl	Amy	true
5	5	Jai Shand	Jai	false
6	6	Anthony Grosse	Anthony	true
7	7	Taj Shand	Taj	true
8	8	Hudson Hollinworth	Hudson	true
9	9	Jack Potter	Jack	true
10	10	Piper Koch	Piper	false
11	11	Murphy Onslow	Murphy	true

✓ Successfully run. Total query runtime: 154 msec. 212 rows affected.

Consultas para Stockitem:

wwwidw/postgres@PostgreSQL 13

Query Editor

Query History

Scratch Pad

1

select * from stockitem;

Data Output

Explain

Messages

Notifications

	stock_item_key [PK] integer	stock_item character varying (200)	color character varying (50)	selling_package character varying (50)	buying_package character varying (50)	brand character varying (50)	size_val character varying (50)	lead_time integer
1	162	RC toy sedan car with remote control (Black) 1/50 scale	Black	Each	Each	Northwind	1/50 scale	
2	30	Clear packaging tape 48mmx100m	Unknown	Each	Each	Unknown	48mmx100m	
3	163	IT joke mug - hardware: part of the computer that can be kicked (Black)	Black	Each	Each	Unknown	Unknown	
4	164	IT joke mug - hardware: part of the computer that can be kicked (White)	White	Each	Each	Unknown	Unknown	
5	255	Developer joke mug - old C developers never die (Black)	Black	Each	Each	Unknown	Unknown	
6	256	Developer joke mug - old C developers never die (White)	White	Each	Each	Unknown	Unknown	
7	257	Developer joke mug - this code was generated by a tool (Black)	Black	Each	Each	Unknown	Unknown	
8	258	Developer joke mug - this code was generated by a tool (White)	White	Each	Each	Unknown	Unknown	
9	259	Developer joke mug - a foo walks into a bar (Black)	Black	Each	Each	Unknown	Unknown	
10	260	Developer joke mug - a foo walks into a bar (White)	White	Each				

✓

Successfully run. Total query runtime: 149 msec. 671 rows affected.

Consultas para Date_table:

wwwidw/postgres@PostgreSQL 13

Query Editor

Query History

Scratch Pad

1

select * from date_table;

Data Output

Explain

Messages

Notifications

	date_key [PK] timestamp without time zone	day_number integer	day_val integer	month_val character varying (50)	short_month character varying (50)	calendar_month_number integer	calendar_year integer	fiscal_month_number integer	fiscal_year integer	
1	2013-01-01 00:00:00		1	1 January	Jan		1	2013	3	2013
2	2013-01-02 00:00:00		2	2 January	Jan		1	2013	3	2013
3	2013-01-03 00:00:00		3	3 January	Jan		1	2013	3	2013
4	2013-01-04 00:00:00		4	4 January	Jan		1	2013	3	2013
5	2013-01-05 00:00:00		5	5 January	Jan		1	2013	3	2013
6	2013-01-06 00:00:00		6	6 January	Jan		1	2013	3	2013
7	2013-01-07 00:00:00		7	7 January	Jan		1	2013	3	2013
8	2013-01-08 00:00:00		8	8 January	Jan		1	2013	3	2013
9	2013-01-09 00:00:00		9	9 January	Jan		1	2013	3	2013
10	2013-01-10 00:00:00		10	10 January	Jan					
11	2013-01-11 00:00:00		11	11 January	Jan					

✓ Successfully run. Total query runtime: 97 msec. 1461 rows affected.

Consultas para Fact_order:

wwwidw/postgres@PostgreSQL 13

Query Editor Query History

Scratch Pad

```
1 select * from fact_order;
```

	order_key [PK] integer	city_key integer	customer_key integer	stock_item_key integer	order_date_key date	picked_date_key date	salesperson_key integer	picker_key integer	package character varying (50)	quantity integer	unit_price numeric	tax_rate numeric	total_u numer
1	6	70644	57	90	2013-01-01	2013-01-01	19	11	Each	5	32	15	
2	38	69146	224	58	2013-01-01	2013-01-02	15	20	Each	100	22	15	
3	66	43830	268	149	2013-01-01	2013-01-02	15	20	Each	10	230	15	
4	101	68504	105	1	2013-01-01	2013-01-02	4	20	Each	80	50	15	
5	105	58731	376	64	2013-01-01	2013-01-02	6	20	Each	100	15	15	
6	109	48409	387	50	2013-01-01	2013-01-02	11	20	Each	90	102	15	
7	119	37955	215	156	2013-01-01	2013-01-02	15	20	Each	6	30	15	
8	139	70600	368	82	2013-01-01	2013-01-02	9	20	Pair	120	5	15	
9	146	57097	232	58	2013-01-01	2013-01-02	19	20	Each	40	22	15	
10	147	57097	232	79	2013-01-01	2013-01-02	19						

✓ Successfully run. Total query runtime: 319 msec. 130901 rows affected.

Documentación del proceso y las transformaciones realizadas - Modificaciones a los esquemas en BigQuery

Creación de tablas:

*CONSUL... 3 X PUBLIC... X *CONSUL... 4 X BILAB4 X REDACTAR CONSULTA NUEVA

bilab4

CREAR TABLA

USO COMPARTIDO COPIAR BORRAR

Información del conjunto de datos

ID de conjunto de datos bilab4angelamiguelandres: bilab4

Creado 6 nov. 2021, 14:21:43 UTC-5

Ver acciones to 60 días

predeterminado de la tabla

Última modificación 6 nov. 2021, 14:21:43 UTC-5

Ubicación de los datos US

Descripción

EDITAR DETALLES

Se seleccionaron los archivos procesados de los datos

Google Cloud Platform BILAB4Angelamiguelandres

Explorador

Visualizando los proyectos fijados.

public city

public costumer

public date

public employee

public fact_order

MÁS RESULTADOS

Abrir

Disco local (D:) Documentos bilab4 BIS-3801-14 data Completos

Nombre	Fecha de modificación	Tipo	Tamaño
dimension_city	6/11/2021 2:10 p. m.	Archivo de valores...	13,502 KB
dimension_city_preprocessed	6/11/2021 2:10 p. m.	Archivo de valores...	10,658 KB
dimension_customer	6/11/2021 2:10 p. m.	Archivo de valores...	47 KB
dimension_customer_preprocessed	6/11/2021 2:10 p. m.	Archivo de valores...	43 KB
dimension_date	6/11/2021 2:10 p. m.	Archivo de valores...	71 KB
dimension_date_preprocessed	6/11/2021 2:10 p. m.	Archivo de valores...	62 KB
dimension_employee	6/11/2021 2:10 p. m.	Archivo de valores...	8 KB
dimension_employee_preprocessed	6/11/2021 2:10 p. m.	Archivo de valores...	7 KB
dimension_stock_item	6/11/2021 2:10 p. m.	Archivo de valores...	86 KB
dimension_stock_item_preprocessed	6/11/2021 2:10 p. m.	Archivo de valores...	74 KB
fact_order	6/11/2021 2:10 p. m.	Archivo de valores...	23,919 KB
fact_order_preprocessed	6/11/2021 2:10 p. m.	Archivo de valores...	18,905 KB

Nombre: fact_order_preprocessed

Abrir

Crear tabla

Origen

Crear tabla desde Subir

Seleccionar archivo * EXPLORAR

Formato del archivo Avro

Destino

Proyecto * bilab4angelamiguelandres EXPLORAR

ID de conjunto de datos * bilab4

Nombre de la tabla *

Tipo de tabla Tabla nativa

Esquema

El archivo fuente define el esquema.

Configuración de particiones y clústeres

Partición Sin particionar

CREAR TABLA CANCELAR

cada tabla fue creada:

Google Cloud Platform BILab4AngelMiguelAndres bigquery

ZONA DE PRUEBAS Configura la facturación para disfrutar todas las funciones de BigQuery. Más información DESCARTAR ACTUALIZAR

INFORMACIÓN Y CARACTERÍSTICAS ACCESO DIRECTO INHABILITAR LAS PESTAÑAS DEL EDITOR

Explorador

Comienza a escribir para buscar.

Visualizando los proyectos fijados.

bilab4angelmiguelandres

bilab4

public city

public costumer

public date

public employee

public fact_order

public stockitem

MÁS RESULTADOS

MÁS RESULTADOS

bilab4

Información del conjunto de datos

ID de conjunto de datos bilab4angelmiguelandres.bilab4

Creado 6 nov. 2021, 14:21:43 UTC-5

Vencimiento predeterminado de la tabla 60 días

Última modificación 6 nov. 2021, 14:21:43 UTC-5

Ubicación de los datos US

Descripción

EDITAR DETALLES

Esquema de la tabla stockitem:

Google Cloud Platform BILab4AngelMiguelAndres bigquery

ZONA DE PRUEBAS Configura la facturación para disfrutar todas las funciones de BigQuery. Más información DESCARTAR ACTUALIZAR

INFORMACIÓN Y CARACTERÍSTICAS ACCESO DIRECTO INHABILITAR LAS PESTAÑAS DEL EDITOR

Explorador

Comienza a escribir para buscar.

Visualizando los proyectos fijados.

bilab4angelmiguelandres

bilab4

public city

public costumer

public date

public employee

public fact_order

public stockitem

MÁS RESULTADOS

MÁS RESULTADOS

public stockitem

CONSULTA COMPARTIR COPIAR INSTANTÁNEA BORRAR EXPORTAR

ESQUEMA DETALLES VISTA PREVIA

Esquema de la tabla

Filtro Ingresar el nombre o el valor de la propiedad

Nombre del campo	Tipo	Modo	Etiquetas de política	Descripción
Stock_Item_Key	INTEGER	NULLABLE		
Stock_Item	STRING	NULLABLE		
Color	STRING	NULLABLE		
Selling_Package	STRING	NULLABLE		
Buying_Package	STRING	NULLABLE		
Brand	STRING	NULLABLE		
Size_val	STRING	NULLABLE		
Lead_Time_Days	INTEGER	NULLABLE		
Quantity_Per_Outer	INTEGER	NULLABLE		
Is_Chiller_Stock	BOOLEAN	NULLABLE		
Tax_Rate	FLOAT	NULLABLE		
Unit_Price	FLOAT	NULLABLE		
Recommended_Retail_Price	FLOAT	NULLABLE		
Typical_Weight_Per_Unit	FLOAT	NULLABLE		

EDITAR ESQUEMA VER POLÍTICAS DE ACCESO DE FILA

Resultados BigQuery

Consultas para City:

*CONSUL... 4

+

REDACTAR CONSULTA NUEVA

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

✓ Esta consulta procesará 11.8 M

1

SELECT * FROM `bilab4angelamiguelandres.bilab4.public city` LIMIT 1000

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 11.8 MB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	City_Key	City	State_Province	Country	Continent	Sales_Territory	Region	Subregion	Latest_R
1	6245	Ely	Iowa	United States	North America	Plains	Americas	Northern America	
2	63815	Ely	Iowa	United States	North America	Plains	Americas	Northern America	
3	90434	Ely	Iowa	United States	North America	Plains	Americas	Northern America	
4	12696	Ira	Iowa	United States	North America	Plains	Americas	Northern America	
5	63988	Ira	Iowa	United States	North America	Plains	Americas	Northern America	
6	90607	Ira	Iowa	United States	North America	Plains	Americas	Northern America	
7	14966	Oto	Iowa	United States	North America	Plains	Americas	Northern America	
8	62721	Oto	Iowa	United States	North America	Plains	Americas	Northern America	
9	82313	Oto	Iowa	United States	North America	Plains	Americas	Northern America	
10	28766	Ute	Iowa	United States	North America	Plains	Americas	Northern America	

Filas por página: 1001 - 100 de 1000Primera página<<>>>| Última página

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

✓ Esta consulta procesará 11.8 MIB cuando se ejecute.

1

SELECT * FROM `bilab4angelamiguelandres.bilab4.public city` where city_key = 6245 LIMIT 1000

Ubicación de procesamiento: US

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 11.8 MB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	City_Key	City	State_Province	Country	Continent	Sales_Territory	Region	Subregion	Latest_Recorded_Population
1	6245	Ely	Iowa	United States	North America	Plains	Americas	Northern America	1776

Consultas para Customer:

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 46.9 KiB cuando se ejecute.

1

SELECT * FROM `bilab4angelamigueIandres.bilab4.public costumer` LIMIT 1000

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 46.9 KB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	Customer_Key	Customer	Bill_To_Customer	Category	Buying_Group	Primary_Contact	Postal_Code
1	1	Tailspin Toys (Head Office)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Waldemar Fisar	90410
2	2	Tailspin Toys (Sylvanite- MT)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Lorena Cindric	90216
3	3	Tailspin Toys (Peeples Valley- AZ)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Bhaargav Rambhatla	90205
4	4	Tailspin Toys (Medicine Lodge- KS)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Daniel Roman	90152
5	5	Tailspin Toys (Gasport- NY)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Johanna Huiting	90261
6	6	Tailspin Toys (Jessie- ND)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Biswajeet Thakur	90298
7	7	Tailspin Toys (Frankewing- TN)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Kalidas Nadar	90761
8	8	Tailspin Toys (Bow Mar- CO)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Kanti Kotadia	90484
9	9	Tailspin Toys (Netcong- NJ)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Sointu Aalto	90129
10	10	Tailspin Toys (Wimbledon- ND)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Siddhartha Parkar	90061
11	11	Tailspin Toys (Devault- PA)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Elnaz Javan	90185
12	12	Tailspin Toys (Biscay- MN)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Heloisa Fernandes	90054
13	13	Tailspin Toys (Stonefort- IL)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Razeena Hosseini	90685
14	14	Tailspin Toys (Long Meadow- MD)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Tereza Valentova	90633

Filas por página: 1001 - 100 de 402

Primera página<<>>Última página

*CONSUL... 3

REDACTAR CONSULTA NUEVA

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 46.9 KiB cuando se ejecute.

1

SELECT * FROM `bilab4angelamigueIandres.bilab4.public costumer` where Customer_Key= 1 LIMIT 1000

Ubicación de procesamiento: US

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 46.9 KB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	Customer_Key	Customer	Bill_To_Customer	Category	Buying_Group	Primary_Contact	Postal_Code
1	1	Tailspin Toys (Head Office)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Waldemar Fisar	90410

Consultas para Date:

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 98.7 KIB cuando se ejecute.

1

SELECT * FROM `bilab4angelamiguelandres.bilab4.public date`

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.0 s, se almacenó en caché)

Información del trabajoResultadosJSON

Fila	Date_key	Day_Number	Day_val	Month_val	Short_Month	Calendar_Month_Number	Calendar_Year	Fiscal_Month_Number	Fiscal_Year
1	2013-05-01	1	1	May	May	5	2013	7	2013
2	2013-05-02	2	2	May	May	5	2013	7	2013
3	2013-05-03	3	3	May	May	5	2013	7	2013
4	2013-05-04	4	4	May	May	5	2013	7	2013
5	2013-05-05	5	5	May	May	5	2013	7	2013
6	2013-05-06	6	6	May	May	5	2013	7	2013
7	2013-05-07	7	7	May	May	5	2013	7	2013
8	2013-05-08	8	8	May	May	5	2013	7	2013
9	2013-05-09	9	9	May	May	5	2013	7	2013
10	2013-05-10	10	10	May	May	5	2013	7	2013
11	2013-05-11	11	11	May	May	5	2013	7	2013
12	2013-05-12	12	12	May	May	5	2013	7	2013
13	2013-05-13	13	13	May	May	5	2013	7	2013
14	2013-05-14	14	14	May	May	5	2013	7	2013

Filas por página: 100 1 - 100 de 1461Primera página<<>>Última página

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 98.7

1

SELECT* FROM `bilab4angelamiguelandres.bilab4.public date` WHERE Date_key = '2013-05-01' LIMIT 100

Ubicación de procesamiento: US

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 98.7 KB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	Date_key	Day_Number	Day_val	Month_val	Short_Month	Calendar_Month_Number	Calendar_Year	Fiscal_Month_Nun
1	2013-05-01	1	1	May	May	5	2013	

Consultas para Employee:

EJECUTAR

MÁS

GUARDAR

PROGRAMACIÓN

Esta

1

SELECT * FROM `bilab4angelamiguelandres.bilab4.public employee` LIMIT 1000

Resulta...nsulta

GUARDAR LOS RESULTADOS

Se completó la consulta (pasaron 0.4 s, se procesaron 6.2 KB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	Employee_Key	Employee	Preferred_Name	Is_Salesperson
1	5	Jai Shand	Jai	false
2	24	Jai Shand	Jai	false
3	33	Jai Shand	Jai	false
4	44	Jai Shand	Jai	false
5	53	Jai Shand	Jai	false
6	60	Jai Shand	Jai	false
7	64	Jai Shand	Jai	false
8	69	Jai Shand	Jai	false
9	104	Jai Shand	Jai	false
10	111	Jai Shand	Jai	false
11	114	Jai Shand	Jai	false
12	131	Jai Shand	Jai	false
13	137	Jai Shand	Jai	false
14	141	Jai Shand	Jai	false

Filas por página:1001 - 100 de 212Primera página<<>>Última página

EJECUTAR

MÁS

GUARDAR

PROGRAMACIÓN

Esta consulta procesará 14.7 MIB cuando se ejecute

1

SELECT * FROM `bilab4angelamiguelandres.bilab4.public employee` where employee_key = 5 LIMIT 1000

Ubicación de procesamiento: US

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 6.2 KB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	Employee_Key	Employee	Preferred_Name	Is_Salesperson
1	5	Jai Shand	Jai	false

Consulta para fact_order:

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 14.7 MIB cuando se ejecute

1

SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` LIMIT 1000

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 14.7 MB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	Order_Key	City_Key	Customer_Key	Stock_Item_Key	Order_Date_Key	Picked_Date_Key	Salesperson_Key	Picker_Key	Package	Quantity	Unit_Price	Tax_Rate	Total_Excluding_Tax	Tax_Amount	Total_Including_Tax
1	202379	87114	397	218	2016-01-05	2016-01-05	19	123	Each	2	2.5	15.0	5.0	0.75	5.75
2	210495	96334	291	218	2016-02-18	2016-02-18	143	147	Each	2	2.5	15.0	5.0	0.75	5.75
3	212061	104397	279	219	2016-02-26	2016-02-26	152	163	Each	2	2.5	15.0	5.0	0.75	5.75
4	212129	78038	299	219	2016-02-26	2016-02-26	19	163	Each	2	2.5	15.0	5.0	0.75	5.75
5	213548	61717	329	218	2016-03-03	2016-03-03	143	157	Each	2	2.5	15.0	5.0	0.75	5.75
6	208650	83476	388	218	2016-02-06	2016-02-06	143	143	Each	4	2.5	15.0	10.0	1.5	11.5
7	216927	85883	262	218	2016-03-23	2016-03-23	170	147	Each	4	2.5	15.0	10.0	1.5	11.5
8	206547	83476	388	219	2016-01-26	2016-01-26	158	163	Each	8	2.5	15.0	20.0	3.0	23.0
9	210228	90400	377	219	2016-02-17	2016-02-17	151	151	Each	8	2.5	15.0	20.0	3.0	23.0
10	214298	51780	348	219	2016-03-08	2016-03-08	143	158	Each	8	2.5	15.0	20.0	3.0	23.0
11	216532	113196	295	219	2016-03-22	2016-03-22	165	168	Each	8	2.5	15.0	20.0	3.0	23.0
12	216785	90614	346	219	2016-03-22	2016-03-22	19	168	Each	8	2.5	15.0	20.0	3.0	23.0
13	217085	72610	338	219	2016-03-23	2016-03-23	161	147	Each	8	2.5	15.0	20.0	3.0	23.0
14	201930	68728	245	219	2016-01-04	2016-01-04	163	147	Each	6	2.5	15.0	15.0	2.25	17.25

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 14.7 MIB cuando se ejecute

1

SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` WHERE Order_key = 202379 LIMIT 1000

Ubicación de procesamiento: US

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.4 s, se procesaron 14.7 MB)

Información del trabajoResultadosJSONDetalles de la ejecución

Fila	Order_Key	City_Key	Customer_Key	Stock_Item_Key	Order_Date_Key	Picked_Date_Key	Salesperson_Key	Picker_Key	Package	Quantity	Unit_Price	Tax_Rate	Total_Excluding_Tax	Tax_Amount	Total_Including_Tax
1	202379	87114	397	218	2016-01-05	2016-01-05	19	123	Each	2	2.5	15.0	5.0	0.75	5.75

Consulta para stockitem:

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 90.3 KIB cuando se ejecute.

1

SELECT * FROM `bilab4angelaniguelandres.bilab4.public.stockitem` LIMIT 1000

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.4 s, se procesaron 90.3 KB)

Información del trabajo

Resultados

JSON

Detalles de la ejecución

Fila	Stock_Item_Key	Stock_Item	Color	Selling_Package	Buying_Package	Brand	Size_val	Lead_Time_Days	Quantity_Per_Outer	Is_Chiller_Stock	Tax_Rate	Unit_Price	Recommended_Retail_Price
1	30	Clear packaging tape 48mmx100m	Unknown	Each	Each	Unknown	48mmx100m	14	20	false	14.0	3.5	5.2
2	322	Clear packaging tape 48mmx100m	Unknown	Each	Each	Unknown	48mmx100m	14	20	false	14.0	3.5	5.2
3	642	Clear packaging tape 48mmx100m	Unknown	Each	Each	Unknown	48mmx100m	14	20	false	14.0	3.5	5.2
4	36	Shipping carton (Brown) 305x305x305mm	Unknown	Each	Each	Unknown	305x305x305mm	14	25	false	14.0	3.5	5.2
5	328	Shipping carton (Brown) 305x305x305mm	Unknown	Each	Each	Unknown	305x305x305mm	14	25	false	14.0	3.5	5.2
6	636	Shipping carton (Brown) 305x305x305mm	Unknown	Each	Each	Unknown	305x305x305mm	14	25	false	14.0	3.5	5.2
7	67	Small sized bubblewrap roll 10m	Unknown	Each	Each	Unknown	10m	14	10	false	14.0	4.5	6.7
8	359	Small sized bubblewrap roll 10m	Unknown	Each	Each	Unknown	10m	14	10	false	14.0	4.5	6.7
9	605	Small sized bubblewrap roll 10m	Unknown	Each	Each	Unknown	10m	14	10	false	14.0	4.5	6.7
10	80	Furry animal socks (Pink) L	Unknown	Pair	Packet	Unknown	L	12	12	false	12.0	5.0	7.4
11	443	Furry animal socks (Pink) L	Unknown	Pair	Packet	Unknown	L	12	12	false	12.0	5.0	7.4
12	81	Furry animal socks (Pink) M	Unknown	Pair	Packet	Unknown	M	12	12	false	12.0	5.0	7.4
13	442	Furry animal socks (Pink) M	Unknown	Pair	Packet	Unknown	M	12	12	false	12.0	5.0	7.4
14	82	Furry animal socks (Pink) S	Unknown	Pair	Packet	Unknown	S	12	12	false	12.0	5.0	7.4

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

Esta consulta procesará 90.3 KIB cuando se ejecute.

1

SELECT * FROM `bilab4angelaniguelandres.bilab4.public.stockitem` where stock_item_key = 30 | LIMIT 1000

Ubicación de procesamiento: US

Resultados de la consulta

GUARDAR LOS RESULTADOS

EXPLORAR DATOS

Se completó la consulta (pasaron 0.3 s, se procesaron 90.3 KB)

Información del trabajo

Resultados

JSON

Detalles de la ejecución

Fila	Stock_Item_Key	Stock_Item	Color	Selling_Package	Buying_Package	Brand	Size_val	Lead_Time_Days	Quantity_Per_Outer	Is_Chiller_Stock	Tax_Rate	Unit_Price	Recommended_Retail_Price	Typical_Weight_Per_Ur
1	30	Clear packaging tape 48mmx100m	Unknown	Each	Each	Unknown	48mmx100m	14	20	false	14.0	3.5	5.23	0

Validación de restricciones de llave foránea en fact_orders:

Para este proceso se definieron 2 pasos:

1. La columna no tiene nulls

Esta verificación se hará para todas las relaciones. Para el primer paso se hace una query que nos da los nulos de cada Fk, la idea es que este valor sea vacío

city_Key(public city FK)

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS


1 SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` where city_key is null

Ubicación de procesamiento: US

Resultados de la consulta

Se completó la consulta (pasaron 0.3 s, se procesaron 14.7 MB)

Información del trabajo Resultados JSON Detalles de la ejecución

 No se obtuvieron resultados para la consulta.

Customer_Key(public costumer FK)

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS


1 SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` WHERE Customer_Key IS NULL

Ubicación de procesamiento: US

Resultados de la consulta

Se completó la consulta (pasaron 0.2 s, se procesaron 14.7 MB)

Información del trabajo Resultados JSON Detalles de la ejecución

 No se obtuvieron resultados para la consulta.

Stock_Item_Key(public stockitem FK)

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS


1 SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` WHERE Stock_Item_Key IS NULL

Ubicación de procesamiento: US

Resultados de la consulta

Se completó la consulta (pasaron 0.2 s, se procesaron 14.7 MB)

Información del trabajo Resultados JSON Detalles de la ejecución

 No se obtuvieron resultados para la consulta.

Order_Date_Key(public date FK)

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

1 SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` WHERE Order_Date_Key IS NULL

Ubicación de procesamiento: US

Resultados de la consulta

Se completó la consulta (pasaron 0.3 s, se procesaron 14.7 MB)

Información del trabajo Resultados JSON Detalles de la ejecución

No se obtuvieron resultados para la consulta.

Picked_Date_Key(public date FK)

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

1 SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` WHERE Picked_Date_Key IS NULL

Ubicación de procesamiento: US

Resultados de la consulta

Se completó la consulta (pasaron 0.5 s, se procesaron 14.7 MB)

Información del trabajo Resultados JSON Detalles de la ejecución

No se obtuvieron resultados para la consulta.

Salesperson_Key(public employee FK)

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

1 SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` WHERE Salesperson_Key IS NULL

Ubicación de procesamiento: US

Resultados de la consulta

Se completó la consulta (pasaron 0.3 s, se procesaron 14.7 MB)

Información del trabajo Resultados JSON Detalles de la ejecución

No se obtuvieron resultados para la consulta.

Picker_Key(public employee FK)

EJECUTAR

GUARDAR

PROGRAMACIÓN

MÁS

1 SELECT * FROM `bilab4angelamiguelandres.bilab4.public fact_order` WHERE Picker_Key IS NULL

Ubicación de procesamiento: US

Resultados de la consulta

Se completó la consulta (pasaron 0.3 s, se procesaron 14.7 MB)

Información del trabajo Resultados JSON Detalles de la ejecución

No se obtuvieron resultados para la consulta.

2. Existe la FK en PK de la dimensión

Para esta parte nos basaremos en el bono, en el cual para limpiar los datos se eliminaron los nulos:

Removemos las filas con elementos nulos.

```
len_o = len(df_city)
df_city.dropna(inplace = True)
```

Y se usaron solo los datos que contarán con una llave foránea que exista en las diferentes dimensiones:

City_Key(public city FK)

```
len_o = len(df_order)
if tipo == "Completos/":
    df_order = df_order[df_order.City_Key.isin(df_city.City_Key)]
else:
    df_order = df_order[df_order.city_key.isin(df_city.City_Key)]
```

Customer_Key(public costumer FK)

```
if tipo == "Completos/":
    df_order = df_order[df_order.Customer_Key.isin(df_customer.Customer_Key)]
else:
    df_order = df_order[df_order.customer_key.isin(df_customer.Customer_Key)]
```

Stock_Item_Key(public stockitem FK)

```
if tipo == "Completos/":
    df_order = df_order[df_order.Stock_Item_Key.isin(df_stock_item.Stock_Item_Key)]
else:
    df_order = df_order[df_order.stock_item_key.isin(df_stock_item.Stock_Item_Key)]
```

Order_Date_Key(public date FK) y Picked_Date_Key(public date FK)

```
if tipo == "Completos/":
    df_order = df_order[df_order.Order_Date_Key.isin(df_date.Date_key)]
    df_order = df_order[df_order.Picked_Date_Key.isin(df_date.Date_key)]
else:
    df_order = df_order[df_order.order_date_key.isin(df_date.Date_key)]
    df_order = df_order[df_order.picked_date_key.isin(df_date.Date_key)]
```

Salesperson_Key(public employee FK) y Picker_Key(public employee FK)

```
if tipo == "Completos/":
    df_order = df_order[df_order.Salesperson_Key.isin(df_employee.Employee_Key)]
    df_order = df_order[df_order.Picker_Key.isin(df_employee.Employee_Key)]
else:
    df_order = df_order[df_order.salesperson_key.isin(df_employee.Employee_Key)]
    df_order = df_order[df_order.picker_key.isin(df_employee.Employee_Key)]
```

De esta forma estamos seguros de que esta condición se aplica en todas las llaves foráneas.

Documentación del proceso y las transformaciones realizadas - Modificaciones a los esquemas en la 3ra herramienta

Talend Open Studio for Data Integration - tMap - tMap_1

Find:

Var:

Auto map!

row1

Column
_row_ID_
_City_Key_
City
_State_Province_
Country
Continent
_Sales_Territory_
Region
Subregion
_Latest_Recorded_Population_

out1

Expresión	Column
row1._row_ID_	_row_ID_
row1._City_Key_	_City_Key_
row1._City_	_City_
row1._State_Province_	_State_Province_
row1._Country_	_Country_
row1._Continent_	_Continent_
row1._Sales_Territory_	_Sales_Territory_
row1._Region_	_Region_
row1._Subregion_	_Subregion_
row1._Latest_Recorded_Population_	_Latest_Recorded_Population_

Editor de esquemas - Editor de expresiones

row1

Columna	CL	Tipo	N.	Date Pattern (Ctrl+Spac...	Longitud	Precision	Default	Comentario
_row_ID_	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		7	0		
_City_Key_	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2	0		
City	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		19	0		
_State_Province_	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		28	0		
Country	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15	0		
Continent	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15	0		
_Sales_Territory_	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		16	0		
Region	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
Subregion	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		18	0		

out1

Columna	CL	Tipo	N.	Date Pattern (Ctrl+Spac...	Longitud	Precision	Default	Comentario
_row_ID_	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		7	0		
_City_Key_	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2	0		
City	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		19	0		
_State_Province_	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		28	0		
Country	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15	0		
Continent	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15	0		
_Sales_Territory_	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		16	0		
Region	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
Subregion	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		18	0		

Apply OK Cancel

Talend Open Studio for Data Integration (7.3.1.20200219.1130) | lab4 (Conexión: Local)

Archivo Editar View Ventana Ayuda

Repositorio LOCAL: lab4

- Business Models
- Job Designs
 - demo
 - csvtodatabase 0.1
- Contexts
- Código
- SQL Templates
- Metadata
 - DB Connections
 - File definitions
 - city 0.1
 - File positional
 - File regex
 - File xml
 - File facet
 - File idf
 - File join
 - LDK
 - Azure Storage
 - Google Drive
 - Marketo
 - Salesforce
 - Snowflake
 - Talend MDM
 - Rules Management
 - Servicio Web
 - crn

Trabajo: job csvtodatabase 0.1

Job csvtodatabase 0.1

city (116295 rows in 1.76s) -> tMap_1 (116295 rows in 1.76s) -> tLogFlow_1 (66227.20 rows/s)

Trabajo: job csvtodatabase 0.1

Basic Run

Execution

Run Kill Clear

Advanced settings

Target Exec

Memory Run

Job csvtodatabase ended at 16:43 06/11/2021. [exit code = 0]

Line limit 100 Wrap

Default

Nombre Valor

Buscar componente...

Favoritos

Utilizados recientemente

Big Data

Business Intelligence

Business

Cloud

Custom Code

Data Quality

Databases

DevNET

ELT

ESB

Fichero

Internet

Log & Errors

Log4j

AssetCatcher

ICronometerStart

ICronometerStop

tdie

FlowMeter

FlowMeterCatcher

LogCatcher

LogFlow

StatCatcher

tiWarn

Misc

Orchestration

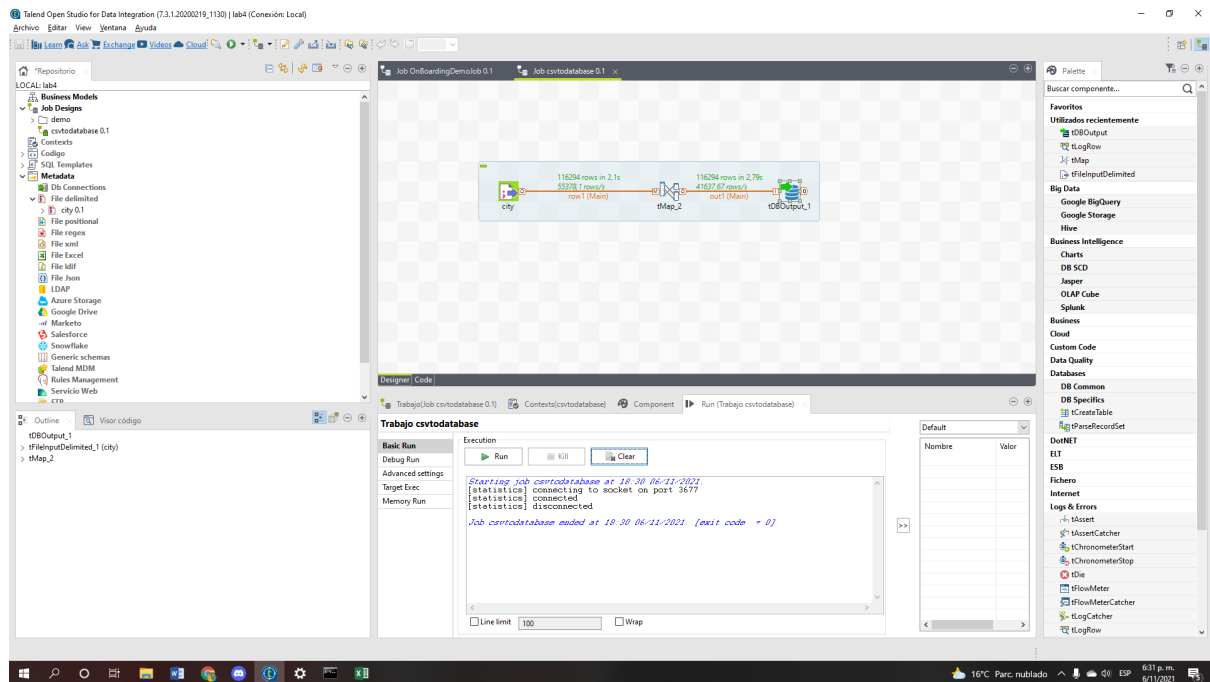
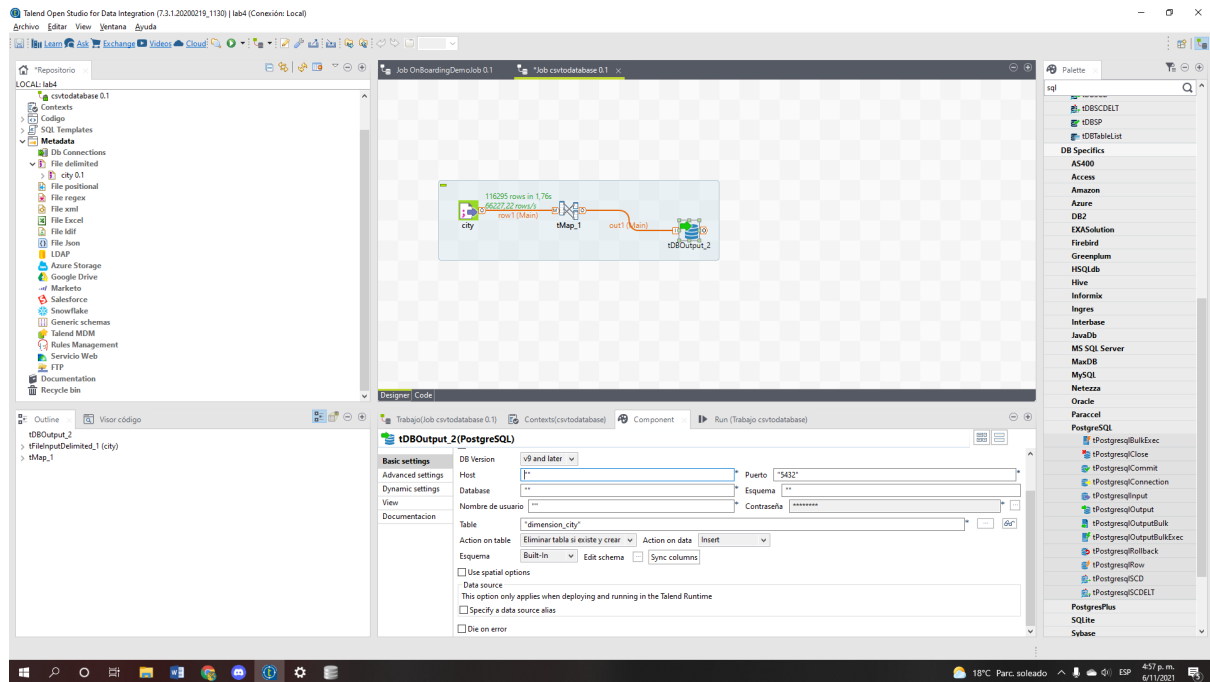
Processing

Sistema

Unstructured

XML

negocio



Resultados 3ra herramienta

Dimensión City:

wwwidw/postgres@PostgreSQL 13

Query EditorQuery History

Scratch Pad

1SELECT * FROM city;

Data OutputExplainMessagesNotifications

	City_Key integer	City character varying (100)	State_Province character varying (100)	Country character varying (100)	Continent character varying (100)	Sales_Territory character varying (100)	Region character varying (100)	Subregion character varying (100)	Latest_Recorded_Population integer
1		1 Carrollton	New York	United States	North America	Midwest	Americas	Northern America	
2		2 Carrollton	Virginia	United States	North America	Southeast	Americas	Northern America	
3		3 Carrollton	Illinois	United States	North America	Great Lakes	Americas	Northern America	
4		4 Carrollton	Missouri	United States	North America	Plains	Americas	Northern America	
5		5 Carrollton	Ohio	United States	North America	Great Lakes	Americas	Northern America	
6		6 Carrollton	Kentucky	United States	North America	Southeast	Americas	Northern America	
7		7 Carrollton	Georgia	United States	North America	Southeast	Americas	Northern America	2
8		8 Carrollton	Alabama	United States	North America	Southeast	Americas	Northern America	
9		9 Carrollton	Mississippi	United States	North America	Southeast	Americas	Northern America	
10		10 Carrollton	Texas	United States	North America	Southwest	Americas	Northern America	11
11		11 Carrollton Manor	Maryland	United States	North America	Midwest	Americas	Northern America	
12		12 Carrolltown	Pennsylvania	United States	North America	Midwest	Americas	Northern America	
13		13 Carrothers	Ohio	United States	North America	Great Lakes	Americas	Northern America	

Dimensión Customer

wwwidw/postgres@PostgreSQL 13

Query EditorQuery History

Scratch Pad

1SELECT * FROM customer;

Data OutputExplainMessagesNotifications

	Customer_Key integer	Customer character varying (100)	Bill_To_Customer character varying (100)	Category character varying (100)	Buying_Group character varying (100)	Primary_Contact character varying (100)	Postal_Code integer
1		1 Tailspin Toys (Head Office)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Waldemar Fisar	90410
2		2 Tailspin Toys (Sylvanite- MT)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Lorena Cindric	90216
3		3 Tailspin Toys (Peeples Valley- ...	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Bhaargav Rambhatia	90205
4		4 Tailspin Toys (Medicine Lodg...	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Daniel Roman	90152
5		5 Tailspin Toys (Gasport- NY)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Johanna Huiting	90261
6		6 Tailspin Toys (Jessie- ND)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Biwajeet Thakur	90298
7		7 Tailspin Toys (Frankewing- TN)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Kalidas Nadar	90761
8		8 Tailspin Toys (Bow Mar- CO)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Kanti Kotadia	90484
9		9 Tailspin Toys (Netcong- NJ)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Sointu Aalto	90129
10		10 Tailspin Toys (Wimbledon- ND)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Siddhartha Parkar	90061
11		11 Tailspin Toys (Devault- PA)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Einaz Javan	90185
12		12 Tailspin Toys (Biscay- MN)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Heloisa Fernandes	90054
13		13 Tailspin Toys (Stonefort- IL)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Razeena Hosseini	
14		14 Tailspin Toys (Bismarck- ND)	Tailspin Toys (Head Office)	Novelty Shop	Tailspin Toys	Teresa Velazquez	90699

Successfully run. Total query runtime: 71 msec. 402 r

Dimensión Date

wwwidw/postgres@PostgreSQL 13

Query EditorQuery HistoryScratch Pad

1

SELECT * FROM date;

Data Output

ExplainMessagesNotifications

	Date_key timestamp without time zone	Day_Number integer	Day_val integer	Month_val character varying (10)	Short_Month character varying (10)	Calendar_Month_Number integer	Calendar_Year integer	Fiscal_Month_Number integer	Fiscal_Year integer	
1	0006-07-06 00:00:00		1	1 January	Jan		1 2013		3 2013	
2	0007-07-06 00:00:00		2	2 January	Jan		1 2013		3 2013	
3	0008-07-05 00:00:00		3	3 January	Jan		1 2013		3 2013	
4	0009-07-05 00:00:00		4	4 January	Jan		1 2013		3 2013	
5	0010-07-06 00:00:00		5	5 January	Jan		1 2013		3 2013	
6	0011-07-06 00:00:00		6	6 January	Jan		1 2013		3 2013	
7	0012-07-05 00:00:00		7	7 January	Jan		1 2013		3 2013	
8	0013-07-05 00:00:00		8	8 January	Jan		1 2013		3 2013	
9	0014-07-06 00:00:00		9	9 January	Jan		1 2013		3 2013	
10	0015-07-06 00:00:00		10	10 January	Jan		1 2013		3 2013	
11	0016-07-05 00:00:00		11	11 January	Jan		1 2013		3 2013	
12	0017-07-05 00:00:00		12	12 January	Jan		1 2013		3 2013	
13	0018-07-06 00:00:00		13	13 January	Jan		1 2013		3 2013	
14	0019-07-06 00:00:00		14	14 January	Jan		1 2013		3 2013	

Dimensión Employee

Query Editor

Query History

1 SELECT * FROM employee;

Data Output

Explain

Messages

Notifications

	Employee_Key integer		Employee character varying (100)		Preferred_Name character varying (100)		Is_Salesperson character varying (100)	
1		1	Lily Code		Lily		True	
2		2	Isabella Rupp		Isabella		False	
3		3	Ethan Onslow		Ethan		False	
4		4	Amy Trefl		Amy		True	
5		5	Jai Shand		Jai		False	
6		6	Anthony Grosse		Anthony		True	
7		7	Taj Shand		Taj		True	
8		8	Hudson Hollinworth		Hudson		True	
9		9	Jack Potter		Jack		True	
10		10	Piper Koch		Piper		False	
11		11	Hudson Onslow		Hudson		True	
12		12	Sophia Hinton		Sophia		True	
13		13	Henry Forlonge		Henry		False	
14		14	Stella Bessheim		Stella		False	

Dimensión Stock

Query Editor		Query History	Sc
1	SELECT * FROM stock;		

Data Output		Explain	Messages	Notifications			
	Stock_Item_Key integer	Stock_Item character varying (100)	Color character varying (100)	Selling_Package character varying (100)	Buying_Package character varying (100)	Brand character varying (100)	Size_val character varying (100)
1		1 Void fill 400 L bag (White) 400L	Unknown	Each	Each	Unknown	400L
2		2 Void fill 300 L bag (White) 300L	Unknown	Each	Each	Unknown	300L
3		3 Void fill 200 L bag (White) 200L	Unknown	Each	Each	Unknown	200L
4		4 Void fill 100 L bag (White) 100L	Unknown	Each	Each	Unknown	100L
5		5 Air cushion machine (Blue)	Unknown	Each	Each	Unknown	Unknown
6		6 Air cushion film 200mmx200...	Unknown	Each	Each	Unknown	325m
7		7 Air cushion film 200mmx100...	Unknown	Each	Each	Unknown	325m
8		8 Large replacement blades 18...	Unknown	Each	Each	Unknown	18mm
9		9 Small 9mm replacement blad...	Unknown	Each	Each	Unknown	9mm
10		10 Packing knife with metal inser...	Unknown	Each	Each	Unknown	18mm
11		11 Packing knife with metal inser...	Unknown	Each	Each	Unknown	9mm
12		12 Permanent marker red 5mm n...	Unknown	Each	Each	Unknown	5mm
13		13 Permanent marker blue 5mm ...	Unknown	Each	Each	Unknown	5mm

✔ Successfu

✓ Successfully

Dimensión Order

Query Editor		Query History	Scratch Pad
1	SELECT * FROM fact;		

Data Output		Explain	Messages	Notifications					
	Order_Key integer	City_Key integer	Customer_Key integer	Stock_Item_Key integer	Order_Date_Key timestamp without time zone	Picked_Date_Key timestamp without time zone	Salesperson_Key integer	Picker_Key integer	Package character varying (100)
1	5	70644	57	14	0006-07-06 00:00:00	0006-07-06 00:00:00		19	11 Each
2	6	70644	57	90	0006-07-06 00:00:00	0006-07-06 00:00:00		19	11 Each
3	7	70644	57	170	0006-07-06 00:00:00	0006-07-06 00:00:00		19	11 Each
4	38	69146	224	58	0006-07-06 00:00:00	0007-07-06 00:00:00		15	20 Each
5	39	69146	224	180	0006-07-06 00:00:00	0007-07-06 00:00:00		15	20 Each
6	40	69146	224	105	0006-07-06 00:00:00	0007-07-06 00:00:00		15	20 Each
7	45	44863	264	110	0006-07-06 00:00:00	0007-07-06 00:00:00		12	20 Each
8	46	44863	264	126	0006-07-06 00:00:00	0007-07-06 00:00:00		12	20 Each
9	66	43830	268	149	0006-07-06 00:00:00	0007-07-06 00:00:00		15	20 Each
10	67	43830	268	38	0006-07-06 00:00:00	0007-07-06 00:00:00		15	20 Each
11	100	68504	105	148	0006-07-06 00:00:00	0007-07-06 00:00:00		4	20 Each
12	101	68504	105	1	0006-07-06 00:00:00	0007-07-06 00:00:00		4	20 Each
13	102	68504	105	53	0006-07-06 00:00:00	0007-07-06 00:00:00		4	20 Each

✔ Successfully run. Total query runtime: 00:00:00.000

✓ Successfully run. Total query runti

Respuestas a las preguntas del laboratorio

¿Por qué se utiliza el comando “IF NOT EXISTS” en la sentencia, en el contexto del proceso de ETL?

R: Porque estamos desarrollando una herramienta que pueda importar datos para realizar un proceso de ETL para exportarlos a nuestro datamart, e ir acumulando y actualizando información, para que esto ocurra, la herramienta debe poder ejecutarse una y otra vez, y esto implica que cada vez que se ejecute, la herramienta debe trabajar con lo que haya en el datamart en ese momento y es muy posible que las tablas ya están creadas, a menos de que sea la primera ejecución o se haya decidido añadir una dimensión nueva.

¿Por qué para la columna de día se utiliza el nombre day_val y no “day”?

Day es una palabra reservada para una función de sql que retorna el día de una fecha seleccionada.

¿De dónde se obtiene la información sobre las columnas que hay que crear en la tabla?

Al menos en pentaho spoon se utiliza directamente un script para creación de tablas, en BigQuery se saca del csv y en Talend se saca del csv también.

¿Cuál es la diferencia entre un ‘job’ y una ‘transformación’?

Un job tiene que ver con el flujo de control, es decir cómo establecer una serie de tareas y establecer un orden para su ejecución. Las transformaciones son una serie de tareas que implican la transformación de datos y pueden ejecutarse en paralelo (mientras que en los job's se ejecutan en orden).

¿Por qué se hace uso del nodo ‘Insert/Update’ y no del ‘Table Output’?

Porque Insert/Update como dice el nombre ingresa nuevos datos y Table Output solo inserta datos, y que solo se inserten datos es un problema (como se explicó anteriormente en la parte del bono#1 con bulkload).

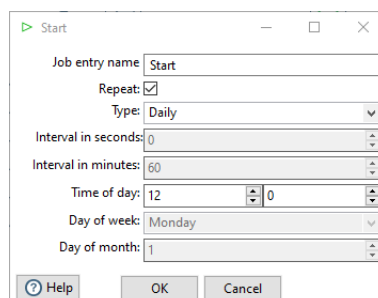
¿Cuál es la finalidad de los campos ‘The key(s) to lookup the values:’ y ‘Update fields’ del nodo ‘Insert/Update’?

El campo ‘The key(s) to lookup the values:’ nos sirve para que cuando estemos exportando la información a la base de datos, se revise para cada entrada que vamos a exportar y si existe una entrada en la base de datos cuyos valores correspondientes a los parámetros que le pasamos al campo antes mencionado existen, de ser así sabemos que este campo no se debe insertar, sino actualizar.

El campo ‘Update fields’ sirve para saber, para los registros que vayamos a actualizar, qué campos han de ser actualizados.

¿Será posible programar la fecha y hora en la cual la organización puede correr de forma automática el proceso de ETL?

La pregunta está en el contexto de la herramienta spoon, y en esta herramienta si es posible, se tendría que configurar el nodo start, como se ve a continuación:



The screenshot shows a 'Start' job configuration window. The 'Job entry name' is 'Start'. The 'Repeat' checkbox is checked. The 'Type' is 'Daily'. The 'Interval in seconds' is 0. The 'Interval in minutes' is 60. The 'Time of day' is 12. The 'Day of week' is Monday. The 'Day of month' is 1. There are 'Help', 'OK', and 'Cancel' buttons at the bottom.

En este ejemplo estaríamos configurando el job para ejecutarse todos los días.

Comparación de herramientas

En principio en todas las herramientas se logró realizar el mismo proceso, sin embargo encontramos las siguientes diferencias entre los distintas herramientas

	Google BigQuery	Spoon Pentaho	Talend
Conexión a la DB	Se suben a un proyecto en la nube (drive) y se administra el tiempo de duración	Permite conectar a postgresql, pero a la vez a otras plataformas locales o pasandoles su dirección en red	Permite exportar los datos a diferentes tipos de bases de datos como sqlite, postgresDB, mysql, etc.
Creación de tablas	Se agregan las tablas, bien sea subidas o creadas desde ceros, al proyecto.	Mediante Scripts	El interfaz ofrece distintas herramientas para crear las tablas, aunque también se pueden crear utilizando el lenguaje java.
Job's	No maneja el concepto de job.	Crea un job inicial de donde parte el flujo de ejecución.	Utiliza el concepto de job para organizar un set de instrucciones que van a darle forma a la información.
Transformaciones	No maneja el concepto de transformaciones.	Es necesario crear múltiples transformaciones para desarrollar el proyecto.	No maneja el concepto de transformaciones.

A partir de esto podemos concluir que talend y pentaho son las que más opciones nos dan como herramientas de ETL, sin embargo Google BigQuery nos ofrece la ventaja de que podemos trabajar en la nube.