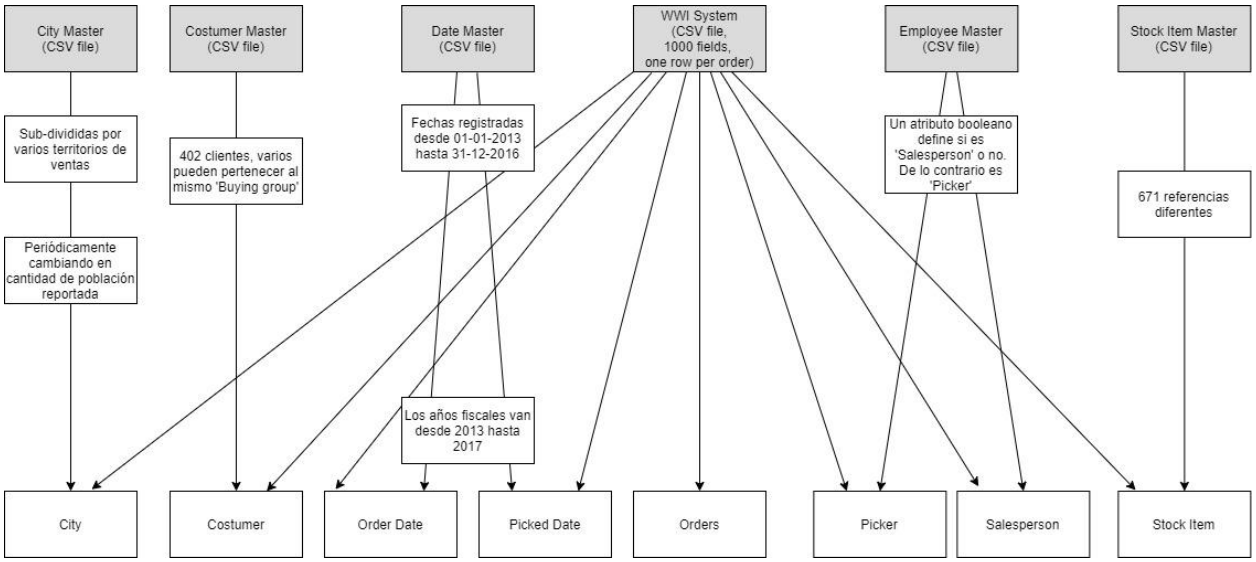
Daniel Perilla, d.perilla11@uniandes.edu.co - 201327313
Julio Morales, ja.morales11@uniandes.edu.co - 201327050
Julian David Mendoza Ruiz, jd.mendozar@uniandes.edu.co - 201730830

Laboratorio 5

Diagrama alto nivel proceso ETL



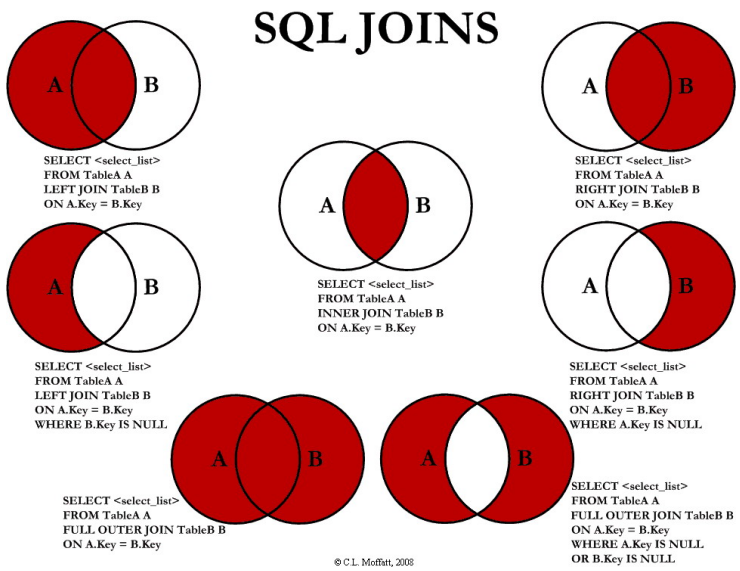
Proceso ETL con Google Big Query

Construcción del modelo definitivo a partir de usar el comando “INNER JOIN” entre las tablas:

Fila	order_key	city_key	City_Key_1	customer_key	Customer_Key_1	order_date_key	Date_key	picked_date_key	Date_key_1	salesperson_key	Employee_Key	picker_key	Employee_Key_1	stock_item_key	Stock_Item_Key_1
1	20	1	1	25	25	2013-11-15	2013-11-15	2015-06-25	2015-06-25	167	167	173	173	182	182
2	600	1	1	35	35	2016-07-20	2016-07-20	2013-06-27	2013-06-27	31	31	62	62	274	274
3	697	1	1	105	105	2016-06-23	2016-06-23	2014-09-16	2014-09-16	196	196	68	68	354	354
4	735	1	1	210	210	2013-12-23	2013-12-23	2015-07-25	2015-07-25	19	19	166	166	471	471
5	745	1	1	97	97	2014-03-11	2014-03-11	2014-01-01	2014-01-01	125	125	19	19	657	657
6	761	1	1	109	109	2015-06-13	2015-06-13	2013-07-15	2013-07-15	38	38	74	74	521	521
7	777	1	1	107	107	2016-06-17	2016-06-17	2016-10-02	2016-10-02	49	49	150	150	81	81
8	824	1	1	144	144	2015-07-03	2015-07-03	2013-11-21	2013-11-21	50	50	50	50	447	447

```
SELECT fact.order_key, fact.city_key, city.City_Key,fact.customer_key, customer.Customer_Key, fact.order_date_key, dateOrder.Date_key,
fact.picked_date_key, datePicked.Date_key, fact.salesperson_key, employee.Employee_Key, fact.picker_key, picker.Employee_Key,
fact.stock_item_key, stockItem.Stock_Item_Key
FROM `lab5-312422.Lab5.fact_order` AS fact
INNER JOIN `lab5-312422.Lab5.city` AS city ON city.City_Key = fact.city_key
INNER JOIN `lab5-312422.Lab5.customer` AS customer ON customer.Customer_Key = fact.customer_key
INNER JOIN `lab5-312422.Lab5.date` AS datePicked ON datePicked.Date_key = fact.picked_date_key
INNER JOIN `lab5-312422.Lab5.date` AS dateOrder ON dateOrder.Date_key = fact.order_date_key
INNER JOIN `lab5-312422.Lab5.employee` AS employee ON employee.Employee_Key = fact.salesperson_key
INNER JOIN `lab5-312422.Lab5.employee` AS picker ON picker.Employee_Key = fact.picker_key
INNER JOIN `lab5-312422.Lab5.stock_item` AS stockItem ON stockItem.Stock_Item_Key = fact.stock_item_key;
```

Aquí se puede observar cómo combinamos todas las tablas de dimensión con la tabla de hechos a partir de INNER JOIN, y en este caso estamos mostrando tanto la columna llave dentro de la tabla de hechos como en la tabla de dimensión correspondiente.
Al utilizar INNER JOIN nos estamos asegurando de que en caso de que haya algún ID dentro de la tabla de hechos que no corresponda dentro a un ID de la tabla de dimensión correspondiente, esta fila no será incluida en el resultado de la consulta.



Aquí podemos observar que al usar INNER JOIN se obtiene la intersección entre ambos conjuntos y solamente sus elementos que estén en ambos, por lo que no se incluye en el resultado ningún elemento que no esté en la intersección (a diferencia de LEFT JOIN, RIGHT JOIN o FULL OUTER JOIN).

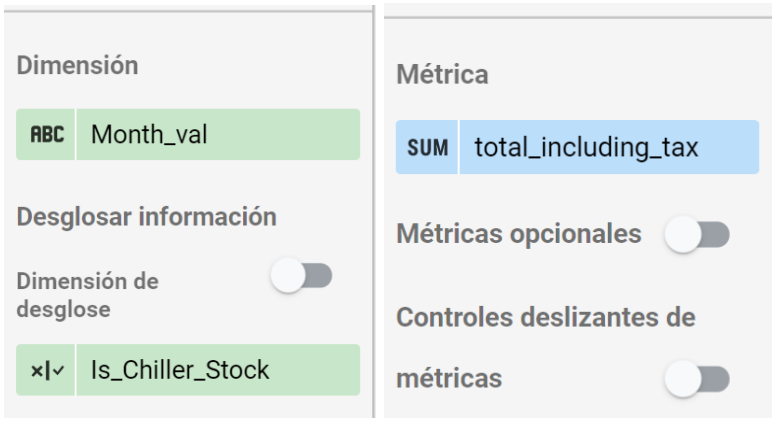
Modelos dimensionales poblados.

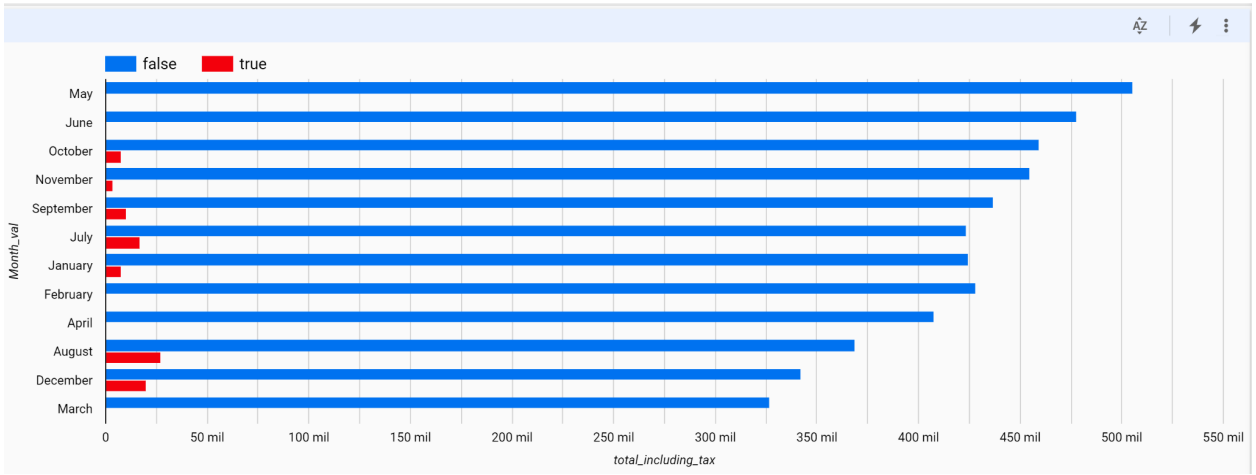
```sql
SELECT fact.order_key, fact.package, fact.quantity, fact.unit_price, fact.tax_rate, fact.total_including_tax,
city.City, city.State_Province, city.Sales_Territory,dateOrder.Calendar_Year, dateOrder.Month_val,
stockItem.Stock_Item, stockItem.Is_Chiller_Stock, customer.Customer, employee.employee ,
FROM `lab5-312422.Lab5.fact_order` AS fact
INNER JOIN `lab5-312422.Lab5.city` AS city ON city.City_Key = fact.city_key
INNER JOIN `lab5-312422.Lab5.customer` AS customer ON customer.Customer_Key = fact.customer_key
INNER JOIN `lab5-312422.Lab5.date` AS datePicked ON datePicked.Date_key = fact.picked_date_key
INNER JOIN `lab5-312422.Lab5.date` AS dateOrder ON dateOrder.Date_key = fact.order_date_key
INNER JOIN `lab5-312422.Lab5.employee` AS employee ON employee.Employee_Key = fact.salesperson_key
INNER JOIN `lab5-312422.Lab5.employee` AS picker ON picker.Employee_Key = fact.picker_key
INNER JOIN `lab5-312422.Lab5.stock_item` AS stockItem ON stockItem.Stock_Item_Key = fact.stock_item_key;
```

Fila	order_key	package	quantity	unit_price	tax_rate	total_including_tax	City	State_Province	Sales_Territory	Calendar_Year	Month_val	Stock_Item
1	20	M	905	888.19	30	9966.05	Carrollton	New York	Mideast	2013	November	Developer joke mug - inheritance is the OO wa
2	600	L	691	1700.8	39	7679.05	Carrollton	New York	Mideast	2016	July	Developer joke mug - that's a hardware probler
3	697	XL	157	3705.56	46	1586.72	Carrollton	New York	Mideast	2016	June	10 mm Double sided bubble wrap 50m
4	735	S	301	4296.82	8	1651.7	Carrollton	New York	Mideast	2013	December	RC toy sedan car with remote control (Yellow)
5	745	L	38	187.86	16	4483.82	Carrollton	New York	Mideast	2014	March	Tape dispenser (Blue)
6	761	M	702	3916.59	56	7535.39	Carrollton	New York	Mideast	2015	June	Ogre battery-powered slippers (Green) S
7	777	M	969	2308.21	66	1675.78	Carrollton	New York	Mideast	2016	June	Furry animal socks (Pink) M

En esta consulta se hizo un join de todas las dimensiones con la tabla de hechos mostrando sólo ciertas columnas que podían ser de utilidad para generar reportes OLAP, más adelante se mostraron unos ejemplos.

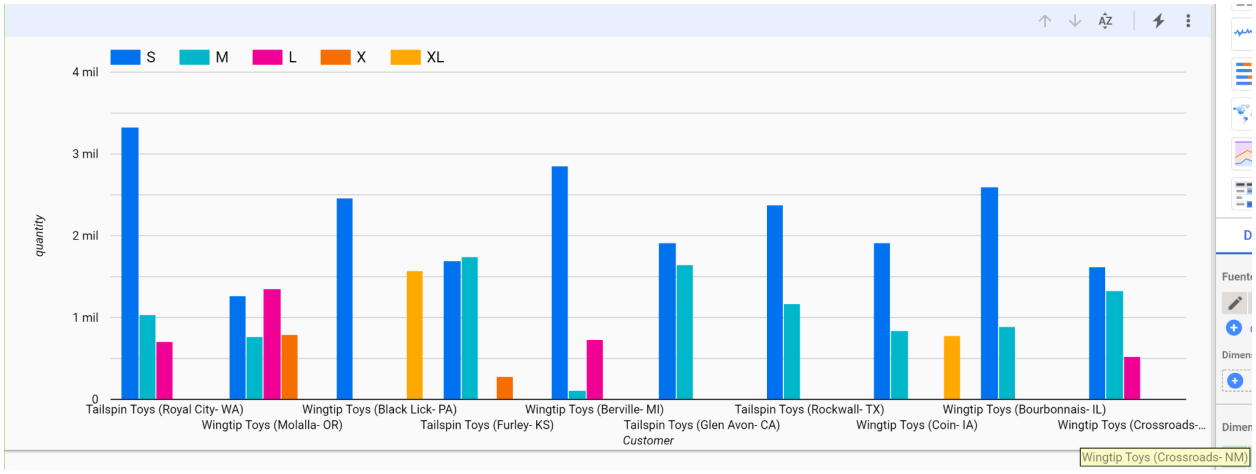
Ventas totales por mes según si es refrigerado o no



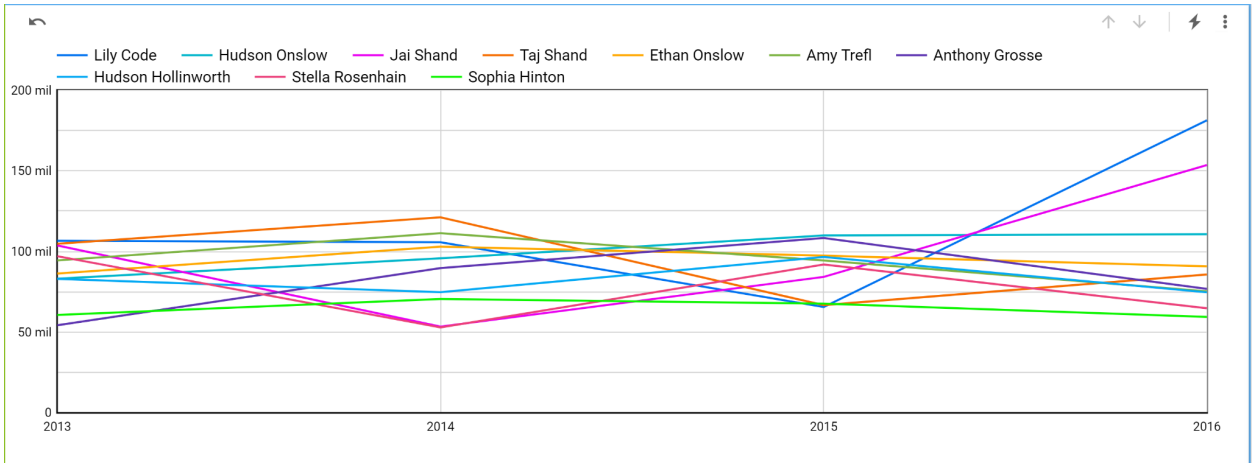


Usando Data Studio para hacer el mismo proceso de Pentaho y Excel para generar reportes OLAP, podemos obtener una gráfica donde visualizamos cuando dinero generan los productos que son refrigerados vs los que no por mes. Si bien la comparación no es adecuada debido a que WWI apenas está empezando con esa parte del negocio, sí es importante ver cuáles son los meses que generan menos ingresos para ambos tipos de productos.

Los productos no refrigerados tienen su mayor ingreso en mayo y progresivamente se reduce hasta llegar a marzo. Por otro lado, los productos no refrigerados no solo no comparten este comportamiento, sino que no hay un comportamiento claro de sus ingresos generados más allá de verse que los meses de mayo, junio, febrero, abril y marzo.



La siguiente gráfica muestra una comparación de todos los clientes de WWI y se compara la cantidad de unidades compradas de un tamaño de paquete en específico. Se puede observar que en general todos los clientes compran paquetes de tamaño pequeño pero hay muy pocos clientes de paquetes Large, Extra y Extra Large, por lo que podría ser de interés entender el porqué y cómo esto podría llevar a encontrar otro tipo de clientes.

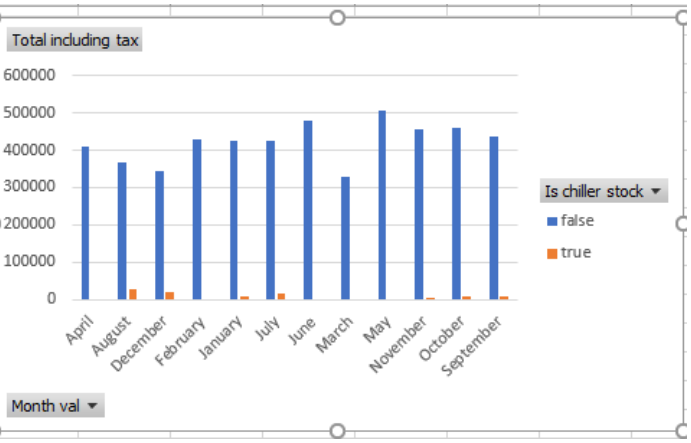


Esta última gráfica muestra las ventas totales de cada uno de los empleados de WWI durante cada año calendario. A partir de esta gráfica es posible no solo ver la evolución del rendimiento de cada empleado sino ver cuáles empleados han tenido bajas en rendimiento y cuales han mejorado en rendimiento (como es el caso de Jai Shand y Lily Code).

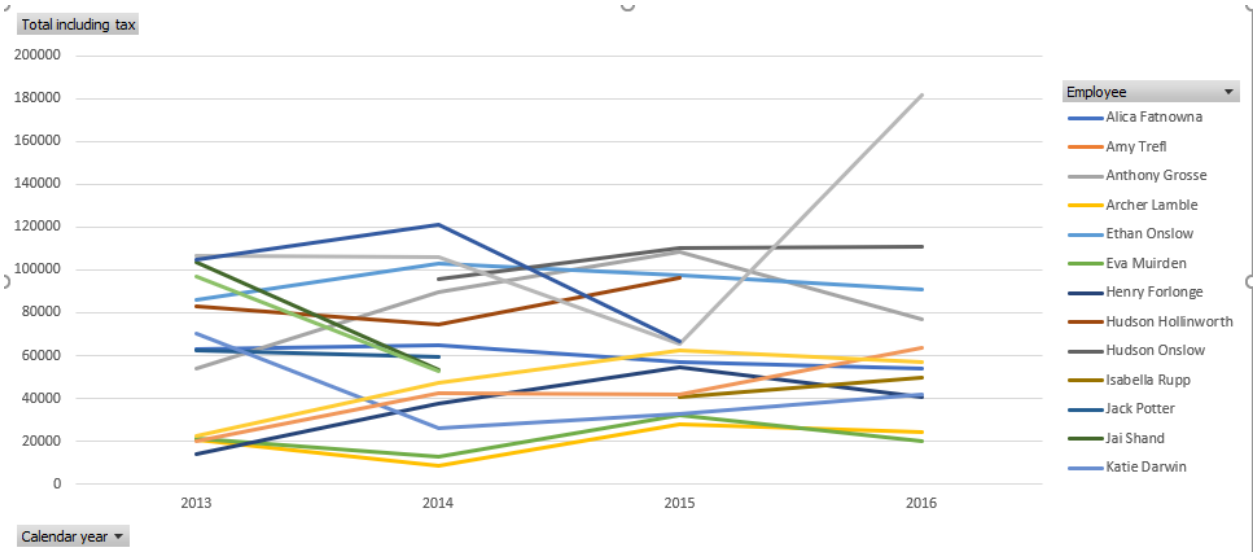
Análisis OLAP con Excel

Ventas totales por mes según si es refrigerado o no

Total including tax	Column Labels		
Row Labels	false	true	Grand Total
April	407692.78		407692.78
August	368595.04	27260.96	395856
December	342325.63	19967.53	362293.16
February	428176.89		428176.89
January	424513.32	7921.39	432434.71
July	423691.27	16866.98	440558.25
June	477814.31		477814.31
March	326870.87		326870.87
May	505516.57		505516.57
November	454615.93	3754.41	458370.34
October	459157.28	7720.19	466877.47
September	436649.59	9998.9	446648.49
Grand Total	**5055619.48**	**93490.36**	**5149109.84**



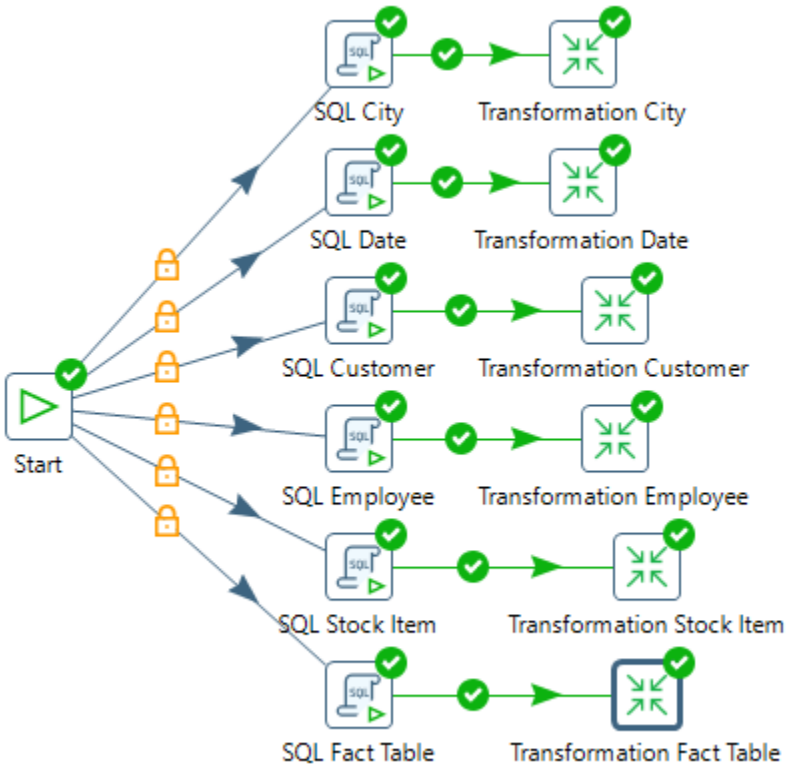
Para este diagrama hay que destacar que obtuvimos los mismos resultados que los de el análisis realizados en la parte de google. Esto nos muestra que ambas herramientas dan un análisis parecido con respecto a lo que es este tipo de análisis.



En este diagrama a comparación del realizado con la parte de Google, encontramos que se incluyen tres empleados más, la razón por la que se toman en este caso puede ser porque a pesar que estos empleados no están presentes durante toda la duración que se tiene en cuenta, si estuvieron presentes durante una porción, por eso el sistema de análisis los tuvo en cuenta.

Transformaciones y jobs definidos en pentaho PDI.

Proceso ETL con Pentaho spoon



Comparación entre las 2 herramientas.

Con respecto a la fase de extracción de los datos desde los sistemas de origen, es más directo hacerlo desde la herramienta BigQuery debido a que se hace directamente desde Google Drive con un reconocimiento automático de las columnas y el formato, mientras que en Pentaho Spoon se tiene que crear todo el flujo y la realizar la creación previa de una base de datos para poder extraer los mismos.

Con ambas herramientas nos dimos cuenta de que se puede implementar el cubo OLAP, por esa razón concluimos que es cuestión de preferencia del usuario escoger entre cual herramienta es más de su conveniencia.

Preguntas

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

El ETL requiere de un cierto nivel de automatización por lo cual se requiere de tomar medidas preventivas en aquellas instrucciones que podrían generar problemas sin la intervención humana. Específicamente, como se trata de un proceso transaccional donde se crean bases de datos relacionales es necesario mantener las propiedades ACID (Atomicity, Consistency, Isolation, Durability). En este caso al usar la instrucción “Create Table” sin el comando genera error cuando la relación ya existe, debido a que el permitir el sobreescribir una tabla o actualizarla podría hacer que los datos pierdan coherencia y por lo tanto no había consistencia dentro de la base de datos.

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

Se utiliza el nombre de “day_val” en vez de “day” debido a que “day” hace parte de las palabras reservadas en SQL, es decir es una palabra que el sistema no permite que se utilice como variable pues ya es utilizada para una determinada función en el sistema.

Reserved Keywords (Transact-SQL) - SQL Server | Microsoft Docs

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

La información sobre los requerimientos de las columnas se obtienen de los datos. Dependiendo de los datos que tengamos en la tabla del CSV, se puede entonces identificar en primera instancia el nombre de la columna y en segunda el tipo de dato que se va manejar.

¿Cuál es la diferencia entre un “Job” y una “Transformación” en Spoon?

Un “Job” es lo que sería un flujo del proceso, es decir los steps en macroscopicos que se realizan en el proceso deseado. La “Transformación” por otro lado se refiere al proceso interno que va en el nodo Transformation en la parte de Job, en esta parte, se especifica las tareas a realizar sobre las tablas. En nuestro caso estas fueron las de input y insertion de los datos de cada tabla.

En el contexto del proceso ETL, ¿Por qué se hace uso del nodo Insert/Update y no del Table Output?
En el contexto del proceso ETL se hace uso del nodo Insert/Update y no el de Table Output porque a pesar de que Table Output es mucho más rápido que Insert/Update, este no puede realizar actualizaciones de los datos, lo que es un requerimiento vital en los procesos ETL.

Table Output vs Insert/Update (pentaho.com)

¿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.” es el campo que se encarga de definir cuales son la(s) llave(s) que va a tener una tabla, teniendo en cuenta estos valores el sistema ejecutará el programa y determinará cuando una tabla se puede crear o no.

El campo de “Update fields” demarca los campos que se desean modificar según la nueva inserción de datos. Esta puede usarse para determinar datos que no desean modificarse.