Laboratorio #5 – ETL con manejo de historia

Andrés M. Ochoa Toro (201913554)

Manuel Felipe Porras Tascón(201913554)

Fernando Andrés Ávalos López(201911468)

ISIS 3710 – Inteligencia de Negocios

Universidad de los Andes, Bogotá, Colombia

{am.ochoat, fa.avalos, mf.porras[}@uniandes.edu.co](mailto:%7d@uniandes.edu.co)

Fecha de presentación: Noviembre 20 de 2021

Tabla de contenido

[1 Introducción 1](#_Toc88330957)

[2 Perfilamiento y procesamiento de datos 2](#_Toc88330958)

[3 Ventajas y desventajas de estrategias del manejo 2](#_Toc88330959)

[4 Perfilamiento y procesamiento de datos 2](#_Toc88330960)

[5 Herramientas proceso ETL 2](#_Toc88330961)

[5.1 Spoon 2](#_Toc88330962)

[5.2 PostgreSQL y Python 4](#_Toc88330963)

[6 Recomendaciones 6](#_Toc88330964)

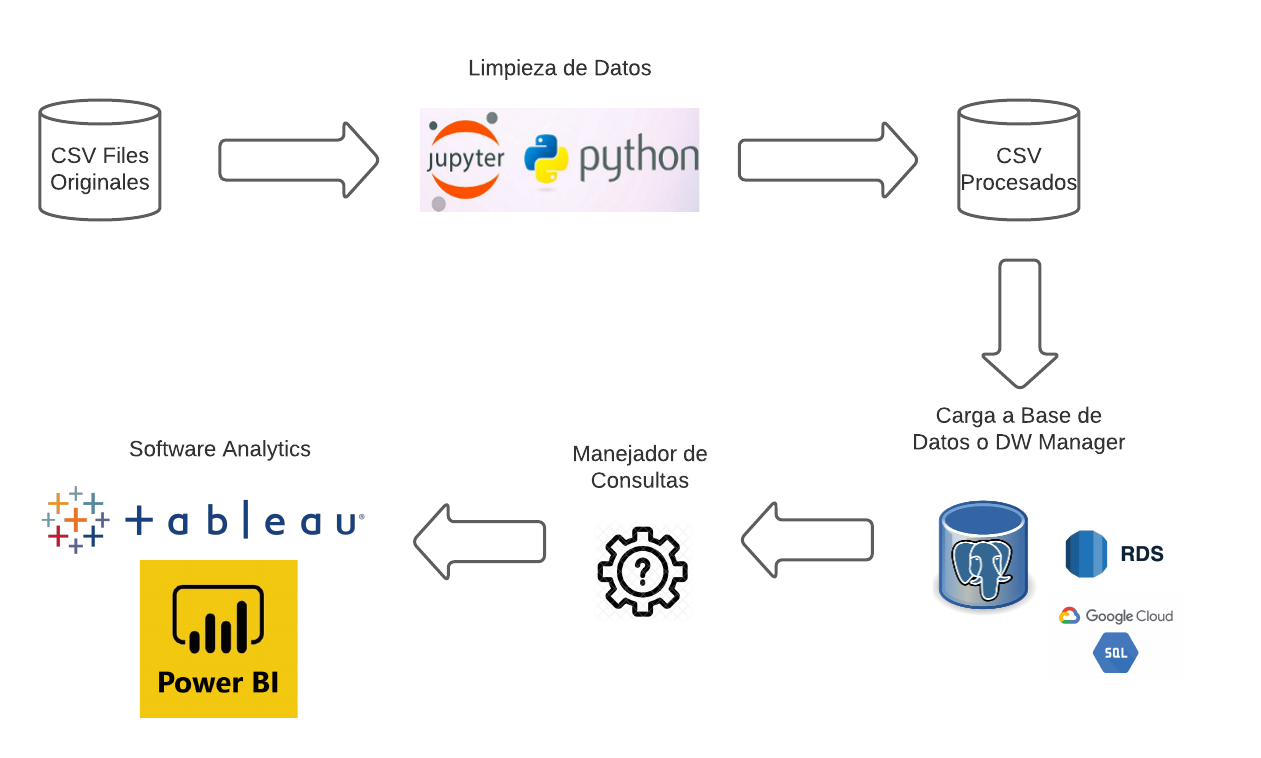
# Introducción

WWI (World Wide Importers) es una empresa encargada de realizar importaciones y venderlas a diferentes clientes en diferentes ciudades de Estados Unidos. En esta ocasión, WWI desea optimizar sus ganancias, pues consideran que algunos de sus productos no están generando las ganancias que deberían. Para esta primera fase, se desea realizar la creación de la base de datos, la carga de datos y la comprobación del correcto funcionamiento del proceso realizado a través de algunas consultas iniciales.

En esta etapa de la consultoría, la empresa desea implementar el proceso ETL que le permita extraer los datos de órdenes desde unos archivos CSV y almacenarlos en un modelo dimensional tal que les permita realizar análisis para mejorar entre otros elementos, su eficiencia operativa. A continuación, se presenta el modelo multidimensional que se desea obtener:

# Perfilamiento y procesamiento de datos

El proceso ETL realizado en este laboratorio, se describe en el siguiente diagrama ETL.



# Ventajas y desventajas de estrategias del manejo

Las ventajas y desventajas de las maneras propuestas para el manejo de historias se describen a continuación.

|  |  |  |  |
| --- | --- | --- | --- |
| Estrategia | Desventajas | Ventajas | |
| Tipo 1 | Esta estrategia olvida toda acción que se haya realizado, lo cual no permite realizar consultas o análisis con datos anteriores a los que se tienen actualmente. Esto presenta una dificultad para los análisis que necesite la empresa sobre sus acciones. | Los recursos respecto a la capacidad de almacenamiento no aumentan a medida de las actualizaciones realizadas. Esto permite que los recursos de almacenamiento necesarios por la empresa no representen un gran gasto. Así mismo, representa un esquema bastante simple y fácil de operar. |
| Tipo 2 | Esta estrategia representa una gran complejidad respecto a la capacidad de almacenamiento. Si se tiene un conjunto significativo de datos que sean actualizados muy frecuentemente, el número de filas crecerá muy rápidamente por la necesidad de conservar los cambios. | En un conjunto de datos, que tenga un poco número de filas y sean actualizadas con poca frecuencia, se permite la posibilidad de conservar un gran historial de datos. Esto genera que al momento de realizar un análisis profundo del negocio y los datos que lo componen. |
| Tipo 3 | Para datos que cambian impredeciblemente representa un esquema poco eficiente, ya que sería necesario la creación de una gran cantidad de columnas con valores nulos. | Para un conjunto de datos en el cual sea predecible las columnas que van a cambiar, representa una forma eficiente de a nivel de almacenamiento, esto porque permite establecer una columna que capture los cambios. Así mismo representa una forma para análisis puntuales en los cambios que se pueden necesitar. |

# Perfilamiento y procesamiento de datos

Para este caso, se usaron los mismos datos que se obtuvieron como resultado del laboratorio pasado. En el caso de los nuevos datos que se proporcionan, se hace un nuevo proceso de limpieza de datos para eliminar los valores nulos, los cuales pueden representar un problema en su manejo en la base de datos.

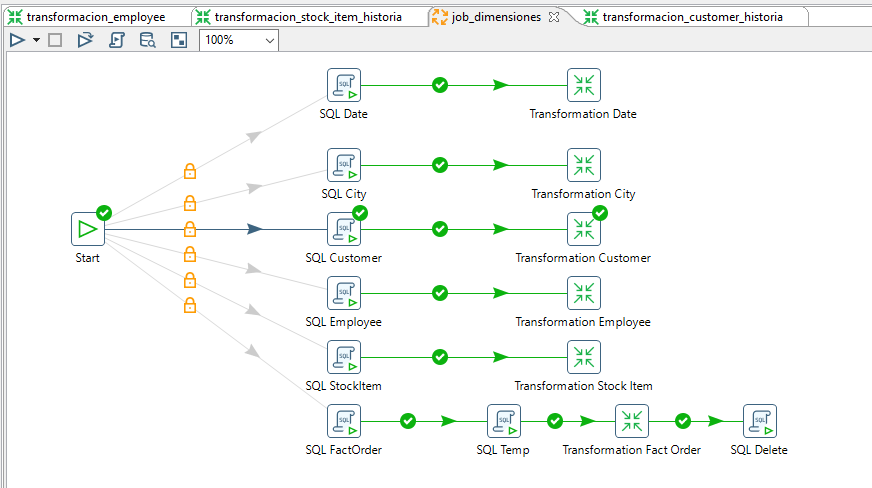
# Herramientas proceso ETL

Para el proceso de ETL, equivalente al laboratorio 5, se continúa usando la base de datos que se creó en el laboratorio 4, y se añade el uso de las siguientes herramientas.

## Spoon

Haciendo uso de la herramienta Spoon, se realizó el siguiente procedimiento ETL.

En primer lugar, se implemento el tipo 2 de Kimball tanto para la dimensión pedida en la guía del laboratorio que fue *stock\_item* como para la dimensión *customer* que es pedida como entregable. Además, se modificó la tabla de hechos como era pedida y como era necesario. Se adjuntan las pruebas correspodientes de las implementaciones:



**Preguntas:**

* ¿Cuál es el objetivo de la columna *tk\_stock\_item*?

El objetivo de esta columna es tener una referencia con los registros de los objetos a vender ya que el objetivo de tener otra tabla llamada *StockItem\_historia* es llevar un historial y registro de los productos como lo sería una implementación de un diagrama multidimensional de tipo 2 propuesto por Kimball.

* ¿Qué significa cada una de estas opciones?

Básicamente, la descripción de cada una de las opciones es la siguiente:

* Technical key: Es la lalve subrogada que tiene auto incremento y es única.
* Version Field: Guarda la versión de la entrada de la dimensión. Ejemplo: 1 ,2, 3 …
* Start of date range: Indica la fecha de inicio de la versión.
* End of date range: Indica la fecha de finalización de la versión.
* Lookup/ Update Fields: Acá existen varias opciones con respecto a los campos que se van a actualizar o se va accionar uno de los siguientes comandos sobre ese campo:
* Insert: Esta opción implementa un tipo II lentamente cambiando la dimensión política. Si la diferencia es detectada para una o más mapeos que tienen la opción de Insert, luego una fila es añadida a la tabla de dimensión.
* Update: Esta opción simplemente actualiza las filas encontradas. Además, este puede usarse para implementar un Tipo I cambiando lentamente la dimensión.
* Punch through: La opcion The punch through implementa una actualización , pero en vez de solo actualizar las respectivas filas , este actualizará todas las versiones de la fila en un tipo 2.
* Date of last insert or update (without stream field as source): Esta opción deja el paso mantener un registro de fecha que guarda la fecha de inserción o actualización automáticamente con la fecha del sistema.
* Date of last insert (without stream field as source): Esta opción deja el paso mantener un registro de fecha que guarda la fecha de inserción automáticamente con la fecha del sistema.
* Date of last update (without stream field as source): Esta opción deja el paso mantener un registro de fecha que guarda la fecha de actualización automáticamente con la fecha del sistema.
* Last version (without stream field as source): Esta opción deja el paso de mantener un flag para indicar si la fila es la última version , todo automáticamente.
* ¿Cómo se puede saber que el proceso ETL manejó los cambios entre los dos archivos?` Pista: Puede revisar los valores de las columnas creadas para el manejo de historia.

Se puede entender que se manejaron los cambios entre los dos archivos, ya que se debieron haber creado otros registros con ligeros cambios en la información y referenciados como \_version 2, así indicando que hubo un cambio y se generó otro registro para obtener el registro actual manteniendo el registro pasado.

* ¿Por qué se debe hacer el cambio para que la columna de la llave subrogada sea de tipo *Unique*?

Se hace para que sea único esa llave subrogada, ya que esa llave no tiene ningún valor para el negocio, pero nos permite tener un identificador en los registrados y así estos estén diferenciados.

* ¿Qué pasa si uno de los datos reportados en la tabla de hechos no existe en alguna de las dimensiones?

No tendría sentido, ya que la tabla de hechos en su mayoría tiene llaves foráneas por lo que generaría un error.

* ¿Qué sugiere para evitar esa situación?

Para evitar esta situación, se sugiere que se limpien los datos dado que se hago un mapeo entre los datos de la tabla de hechos y ver si cada una de las llaves foráneas existen y de lo contrario no tomar en cuenta esos datos.

## PostgreSQL y Python

Como segundo conjunto de herramientas, se combina PostgreSQL para representar el *Data Warehouse* y Python para realizar el debido proceso.

Para este caso, primero se procede a crear una conexión con la base de datos existente. Posteriormente se procede a cargar los datos actuales de la base de datos. Este primer proceso de carga se ve acompañado con la creación de nuevas columnas para el manejo de historia tipo 2. Estas columnas van a representar la versión, la fecha de inicio y fin de la versión, y la nueva llave para el *stock\_item*. Estos nuevos datos se llenan con valores por defecto, sea la versión uno o la máxima y mínima fecha soportada

Para el manejo de los nuevos datos, se procede a cargar el archivo en un nuevo *dataframe,* al cual se le realiza un proceso de limpieza para eliminar los valores nulos que existan o en caso extremo eliminar toda una columna, como se hizo con la columna *Brand.* Después de este paso, se procede a procesar los dos conjuntos de datos de tal manera que se comprueba la existencia de la llave *stock\_item\_key* en ambos conjuntos de datos. Si esto se da, se procede a comparar cada una de las columnas para ver si hay alguna actualización, y proceder a realizar a realizar el respectivo manejo de historias.

El respectivo resultado de este proceso se vería en la base de datos con las versiones que representan la actualización de los productos en un rango de fecha. Lo cual se observa en la siguiente imagen.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Resultados manejo de historias stock item 1

Pantalla de computadora

Descripción generada automáticamente con confianza baja

Resultados manejo de historias stock item 2

Como siguiente paso, se procede a procesar la tabla de hechos. En este caso se realiza un proceso similar a lo mencionado para *stock\_item.* Se parte de la carga de los conjuntos de datos a un *dataframe* para posteriormente realizar la creación del nuevo datafreme que va a ser guardado en la base de datos.

El proceso realizado en este caso es comparar la llave *stock\_item\_key,* existente en el conjunto de datos *fact\_order*, para buscarla en *dataframe* correspondiente a *stock\_item.* Si esta llave se encuentra se procede a comparar la fecha en que se realizó la orden en el rango que abarca la versión del producto en *stock\_item.* El resultado de este es la actualización de *stock\_item\_key* con el valor de *tk\_stock\_item* de la fila encontrada.

# Recomendaciones