

TP Bases de Datos

1er cuatrimestre 2019

Grupo 2

Brian Ezequiel Ail	bail@itba.edu.ar	49254
Florencia Petrikovich	fpetrikovich@itba.edu.ar	58637
Pablo Oseroff	poseroff@itba.edu.ar	58415

Fecha: 13/06/2019

Introducción

En este TP, se aplicó los conceptos de SQL Avanzado visto a lo largo del curso para implementar funcionalidades y restricciones no disponibles de forma estándar, en otras palabras, que no se pueden resolver con Primary Keys, Foreign Keys, entre otros.

A cada integrante del grupo se le asignaron roles que llevaron a cabo, sin embargo si alguno necesitaba asistencia se intercambiaban momentáneamente los mismo. Los roles de cada integrante son como sigue:

- Brian Ail
 - a. Encargado de la creación del repositorio git y los archivos correspondientes.
 - b. Encargado de la creación de las tablas y la importación de los datos de SalesByRegion.csv a la tabla intermedia. (a) y (b)
- Pablo Oseroff
 - a. Encargado de la creación del trigger para la correcta inserción de las tuplas en la tabla definitiva a partir de las tuplas de la tabla intermedia (c).
 - b. Encargado de la impresión a pantalla correcta de la tabla creada en el punto e, Reporte de Ventas Histórico.
 - c. Encargado de las correcciones y sugerencias hechas por las profesoras
- Florencia Petrikovich
 - a. Encargada de las funciones para calcular el margen móvil (d) y crear la tabla para el reporte de ventas histórico (e).
 - b. Encargada del informe

Investigación

Para este TP, casi toda la teoría que se tuvo que utilizar fue dada en clase. Uno de los temas que se investigó fue la sintaxis de las funciones para cuando se quiere devolver una tabla. A la vez, para simplificar el código de la tabla que se crea en el punto e, se investigó cómo devolver una tabla donde en su cláusula FROM se llamaba a otra función que devuelve otra tabla. Esto se decidió debido a la complejidad la consulta. La misma terminó teniendo muchas consultas anidadas y con esta solución no solo se simplificó la lectura del código, sino que también se modularizar la creación de las tablas distintas.

Se investigó también, cómo crear tipos de datos propios en forma de estructura que se usó mucho para la última función ReporteVenta. Su creación permite tener un resultado con nombres mnemotécnicos y al usar el SELECT INTO permite realizar una única consulta en vez de una individual para cada variable.

Por último, aparte de usar la guía proporcionada en el TP, se investigó más a fondo cómo funcionaba la impresión a pantalla y si existían funcionalidades extra que faciliten la misma. Se buscó una manera de unificar la impresión por pantalla y hacer los cálculos no atados a la impresión. De esa manera se pueden obtener los datos sin requerir realizar la impresión.

Importación de datos

Para importar se utilizó el comando COPY de PostgreSQL. Para importar los datos se debe correr el comando especificado en Aclaraciones.

Dificultades

Una de las dificultades que se tuvo que encarar en este proyecto fue el uso adecuado de los cursores. En una primera instancia, se había creado un cursor en la función MargenMovil la cual contenía las tuplas que se deseaban considerar. Usando el cursor, se iteraba por las mismas mientras se incrementaba una variable contador y se sumaba el Revenue-Cost de esa tupla a una variable sum. Antes de finalizar, se retornaba el promedio (sum/counter). Esto resulta ineficiente dado que lo mismo se podía realizar con facilidad mediante una consulta que utilizan las funciones de agregación SUM y COUNT.

Otra dificultad fue la impresión correcta en el punto e. El problema provenía mayormente que se quería crear una tabla ya ordenada para que la impresión sea lo más simple posible. Sin embargo, resultaba imposible ordenar el campo categoría una vez que se agregaba a la categoría "Sales Channel" o "Customer Type" al comienzo. Dicho esto, se tuvo que separar la categoría en dos columnas, Category y Category_desc, donde Category contenía solamente "Sales Channel" o "Customer Type" y Category_desc ordenaba las categorías de manera descendiente. De esta manera, las tuplas quedan ordenadas y el contenido de estas dos columnas se junta durante la impresión.

Por ultimo, elegir la clave fue otra dificultad. Al comenzar, no se definió ninguna clave, sin embargo, esto permitiría el ingreso de tuplas erróneas que no habíamos considerado. Por esta razón, se hizo una clave con todos los campos excepto Cost, Revenue y Territory. Territory a pesar de ser el mismo en todas las tuplas del csv, puede ser que se agregue una tupla con un territorio diferente. Este valor no debería sumarse al del Territorio ya insertado. Por ejemplo si tengo: 01/01/2011 Repairs North America Internet Retail 12350.0 10849.0 y 01/01/2011 Repairs Argentina Internet Retail 12350.0 10849.0, no podría sumar sus revenues y cost en una tupla porque no existiría un valor particular para Territory. Podrían concatenarse los Strings como "North America & Argentina" pero sería una violación a 1NF. Por esta razón, Territory se agrego a la clave definida.

Aclaraciones

- Si a ReporteVentas se le envía un valor negativo, no habrá excepción sino que solo no mostrará nada
- Decidimos realizar una consulta por año para obtener el total ya que consideramos que habrá muchos años que tienen que reportarse y por lo tanto obtener los valores de todos los años sería una consulta muy pesada que no merece ser hecha.
- Correr el ./runImport.sh desde Pampero
- Para conectarse a pampero se puede usar ./connectToDb.sh