INFORME
T.P.E. - Base de Datos I

- Grupo 3: Matías Heimann (57503), Johnathan Katan (56653) y Lóránt Mikolás (57347).

- Profesoras: Leticia Gomez, Valeria Soliani y Teresa Fontanella De Santis.

- Vencimiento de la entrega: 15/06/2018 (05:59).

Introducción

El objetivo principal de este trabajo práctico especial es extraer datos del sitio *Buenos Aires Data* de información abierta del Gobierno de la Ciudad de Buenos Aires sobre los recorridos de bicicletas públicas. Para esto se aplicaron conceptos de SQL Avanzado como PSM y Triggers ya que se debían cumplir ciertos requisitos sobre la migración y actualización de las tablas obtenidas. Además se buscó que las consultas realizadas fuesen genéricas.

Roles

En cuanto a la organización del trabajo si bien se vieron involucrados todos los integrantes en las diferentes tareas pudimos marcar los ejes principales encarados por cada uno:

1. Lóránt Mikolás: encargado de la investigación, encargado del informe y tester complementario.
2. Johnathan Katan: encargado de investigación secundaria, tester general y encargado del funcionamiento global del trabajo.
3. Matías Heimann: encargado del Trigger y encargado de funciones.

Funciones

Con respecto a las funciones utilizadas, presentamos a continuación la descripción de algunas de ellas.

Para empezar, se realizó una función superior llamada **migrate** que tiene varias subfunciones. El objetivo de migrate es pasar el contenido de la tabla aux (que a su vez recibe mediante un \copy todos los datos del .csv) a la tabla recorrido_final. La tabla recorrido_final no tiene el mismo formato ni las mismas restricciones que la tabla aux. Por lo tanto cada función sirve como un filtro o modificador de la tabla anterior.

Así pues presentamos aquí este grupo de funciones:

- removeRepeated(): esta función itera en los pares (usuario, fecha_hora_ret) en la tabla auxWithoutNULL. Por cada par se llama a la función selectSecond(param1, param2) y se les envía como parámetros el usuario y la fecha_hora_ret del par al que se hace referencia. Los únicos pares por los que se itera son los que están repetidos, los otros fueron previamente insertados en auxWithoutRepeated.
- selectSecond(param1, param2): con un cursor se selecciona la segunda tupla de mayor duración con usuario = param1 y fecha_hora_ret = param2. Luego esta tupla es insertada a la tabla auxWithoutRepeated.

- removeOverlaped(): esta función itera por los usuarios de la tabla auxWithFechaDev y llama a la función fixOverlaps(param).
- fixOverlaps(param): esta función transforma los intervalos solapados del usuario al que se le envía por parámetro a un ISUM.
- validateOverlap(): Esta función es ejecutada por un trigger al insertar una tupla en la tabla recorrido_final, y se encarga de detectar si se está insertando un intervalo de uso de bicicleta solapado para un mismo usuario. La implementación fue basada en el álgebra de intervalos de Allen y su definición de intervalo solapado.

Problemas encontrados y sus soluciones

Un problema importante en el trabajo práctico se detectó en la etapa final. Al realizar SELECT migracion() para el archivo test1.csv, se ejecutaba correctamente y se producían los resultados esperados en el tiempo esperado. Luego, cuando se hizo la misma prueba con el archivo recorridos_realizados_2016.csv, tardó tanto la ejecución de la función migración que se tuvo que interrumpir para investigar cuál era el problema.

Luego de investigar, nos dimos cuenta de que el error era que no se habían asignado a las tablas auxiliares sus respectivas claves primarias, de modo que las operaciones sobre estas tablas serían muy lentas. Para el archivo test1.csv, que tenía tan solo 23 filas, el tiempo era insignificante. Pero, para el archivo recorridos_realizados_2016.csv, que tenía más de 700000 filas, el tiempo cambia significativamente.

Por lo tanto, para solucionar este problema, se asignaron las correspondientes claves primarias a las tablas auxiliares creadas, y se logró reducir el tiempo de migración a un minuto y medio.

Otro problema encontrado fue el manejo de importación de input inválido cuando este se relacionaba a las fechas. Este problema se presentó cuando había una fecha o una hora que tenía espacios entre números o entre sus letras. El problema se resolvió borrando los espacios de las columnas fecha_hora_ret y tiempo_uso previo a su inserción a la tabla auxiliar auxWithoutNull. El método utilizado para sacar los espacios fue replace, y se reemplazó el ‘ ‘ por un ‘’.

Instrucciones para la ejecución

Para la migración de los datos del archivo recorridos-realizados-2016.csv a una tabla llamada recorrido_final, se deberá seguir las siguientes instrucciones:

1) En primer lugar, se debe ejecutar create_aux.sql, donde se creará una tabla que se usa para importar el archivo .csv, y otras tablas auxiliares para el filtrado de información no deseada durante el proceso de migración.

2) Para importar el archivo .csv a una tabla auxiliar creada en el paso anterior, se deberá ejecutar desde la terminal el siguiente comando: "\copy aux FROM $PATH DELIMITER ';' CSV HEADER;" donde en $PATH se deberá escribir la ubicación del archivo .csv. Los datos de dicho archivo se importarán a una tabla llamada aux.

3) Se deberá ejecutar funciones.sql, donde se crearán todas las funciones necesarias que se usarán
durante el proceso de migración.

4) Luego, se debe ejecutar migracion.sql, que creará la función migracion(), que se encarga de crear la tabla recorrido_final y de migrar los datos a dicha tabla.

5) Para migrar los datos a la tabla recorrido_final, se debe ejecutar la función migracion con el siguiente comando postgresql:
"SELECT migracion()". Esta función, aparte de migrar la información a la tabla recorrido_final, también se encarga de eliminar las tablas auxiliares generadas durante los pasos anteriores.

6) Por último, si se desea agregar un trigger a la tabla recorrido_final, que valide si se está insertando un intervalo solapado, se deberá ejecutar trigger.sql, que creará y asignará dicho trigger a la tabla.

Conclusión

En conclusión este trabajo nos sirvió para ampliar el conocimiento del grupo sobre SQL Avanzado (PSM y Triggers), entender la gran importancia de estas herramientas y sus potenciales usos. Así pues, se obtuvo el resultado buscado en la tabla recorrido_final y se pudieron aplicar restricciones que no hubieran sido posibles de implementar con las disponibles de forma estándar.

Bibliografía

Además de utilizar el material proveído por la cátedra se consultaron las siguientes referencias en la etapa de investigación:

- https://www.postgresql.org/docs/9.2/static/app-psql.html
- https://campus.itba.edu.ar/ (material de la cátedra)
- https://stackoverflow.com/ (si bien se tiene en claro que no es una fuente académica oficial, funcionó para investigar dudas similares a las que fueron surgiendo durante el desarrollo del trabajo práctico).