Base de Datos I
Trabajo Práctico Especial
1er Cuatrimestre 2018

1. Objetivo

El objetivo de este Trabajo Práctico Especial es aplicar los conceptos de SQL Avanzado (PSM, Triggers) vistos a lo largo del curso, para implementar funcionalidades no disponibles de forma estándar (que no pueden resolverse con Primary Keys, Foreign Key, etc)

2. Descripción del Trabajo

El sitio Buenos Aires Data, https://data.buenosaires.gob.ar/, ofrece datasets de información abierta del Gobierno de la Ciudad de Buenos Aires. Entre otros datasets, se encuentra el que contiene información sobre los recorridos de bicicletas públicas para todos los años a partir del 2010. Estos datos se encuentran en formato .csv (comma separated values) y para este trabajo utilizaremos los datos correspondientes al año 2016. Hay un archivo .csv para cada año.

Es posible que el formato varíe ligeramente año a año, pero en líneas generales, los archivos csv contienen las siguientes columnas:

PERIODO	ID_USUARIO	FECHA_HORA_RETIRO	ORIGEN_ESTACION	NOMBRE_ORIGEN	DESTINO_ESTACION	NOMBRE_DESTINO	TIEMPO_USO	FECHA_CREACION

La información detallada de cada columna se encuentra junto con los archivos de datos en:

https://data.buenosaires.gob.ar/layout/H1giQXf7kx/preview

Se busca en este trabajo práctico especial utilizar los datos del archivo csv para poblar la tabla RECORRIDO_FINAL que tiene la siguiente estructura:

```
CREATE TABLE recorrido_final
(periodo          TEXT,
usuario           INTEGER,
fecha_hora_ret    TIMESTAMP NOT NULL,
est_origen        INTEGER NOT NULL,
est_destino       INTEGER NOT NULL,
fecha_hora_dev    TIMESTAMP NOT NULL CHECK(fecha_hora_dev >=
fecha_hora_ret),
PRIMARY KEY(usuario,fecha_hora_ret));
```

La siguiente es la correspondencia entre los campos en el csv y los de la tabla RECORRIDO_FINAL

CSV	**RECORRIDO_FINAL**
periodo	periodo
id_usuario	usuario
fecha_hora_retiro	fecha_hora_ret
origen_estacion	est_origen
destino_estacion	est_destino
fecha_hora_retiro + tiempo_uso	fecha_hora_dev

Analizando los datos del csv se han detectado dos problemas que pueden surgir en la migración:

1) Podrían contener valor NULL los campos: *usuario, fecha_hora_retiro, est_origen, destino_estacion, tiempo_uso.* Este último además podría no representar un tiempo o su valor ser < 0.
2) Los campos *id_usuario+fecha_hora_retiro* pueden estar repetidos
3) Para un mismo usuario, pueden existir intervalos solapados de uso de las bicicletas.

Por ejemplo, en los datos del recorrido de 2016 aparece el usuario 74710 con los siguientes datos:

201608	74710	13/08/2016 13:09	28	PLAZA BOEDO	528	PLAZA BOEDO	0H 18MIN 9SE	11/06/2013
201608	74710	13/08/2016 13:22	28	PLAZA BOEDO	528	PLAZA BOEDO	0H 52MIN 5SE	11/06/2013

Aquí se ve que la *fecha_hora_retiro* de la primera fila + el *tiempo_uso* que tuvo prestada la bicicleta, 13/08/2016 13:09 + 18min9seg=13/08/2016 13:27:09 es mayor que la *fecha_hora_retiro* en la fila siguiente: 13/08/2016 13:22.

Se quiere realizar la migración de los datos del csv a la tabla RECORRIDO_FINAL resolviendo los problemas 1), 2) y 3) de la siguiente manera:

1) Las filas en el csv con este tipo de problema, deben descartarse, es decir, no se migran a la tabla RECORRIDO_FINAL.
2) Dado que en RECORRIDO_FINAL *usuario+fecha_hora_*ret son clave, en caso de que detecten en el csv varios registros que coincidan en el *id* del usuario y la *fecha_hora_retiro*, se deben ordenar los datos por el atributo *tiempo_uso* y migrar a la tabla RECORRIDO_FINAL la segunda tupla de acuerdo con dicho orden. El resto de las filas del mismo usuario y fecha_hora_retiro se descartan, es decir, no pasan a la tabla RECORRIDO_FINAL.
3) Si para un usuario se detectan intervalos solapados encadenados, se debe migrar a la tabla RECORRIDO_FINAL, una tupla que contenga el ISUM de dichos intervalos. La estación origen debe coincidir con el intervalo de menor *fecha_hora_retiro* y la estación destino debe coincidir con la estación del intervalo con mayor *fecha_hora_retiro*.

Además, una vez que se han migrado los datos exitosamente, se pretende garantizar que estos errores no vuelvan a producirse al ingresar nuevos datos a RECORRIDO_FINAL.

3. Procedimiento

a) Se deben crear la función Postgresql **migración()** que realice la migración desde el csv a la tabla RECORRIDO_FINAL resolviendo los problemas 1, 2 y 3 de la manera que se explica en el ítem anterior. Dicha función no debe retornar nada.
Los alumnos pueden crear las tablas temporarias y las funciones auxiliares que consideren necesarias para lograr el objetivo solicitado.

b) Y para garantizar que, para un usuario dado, no se vuelvan a introducir intervalos solapados en la tabla RECORRIDO_FINAL, se debe crear el trigger **detecta_solapado** que al intentar insertar una tupla en la tabla RECORRIDO_FINAL cuyo intervalo de uso de una bicicleta para un usuario fuera solapado con otro intervalo para el mismo usuario, rechace la inserción y emita por consola un aviso indicando el motivo del rechazo.

Tener en cuenta que:

- En la tabla RECORRIDO_FINAL no hay un atributo que represente el tiempo de uso sino que en base a la hora de retiro y el tiempo de uso del csv se debe calcular el valor del atributo *fecha_hora_dev* que es de tipo TIMESTAMP.
- No se tiene que tener en cuenta para la migración el atributo del csv *fecha_creación:* este atributo se descarta, no se migra.
- No es posible cambiar los tipos de datos de RECORRIDO_FINAL, los datos del csv deben convertirse y adaptarse mediante funciones Postgresql para que la migración sea exitosa.
- Es importante el orden en que se resuelven los problemas: primero se debe resolver el 1 y luego el 2 y por último el 3.
- Al finalizar se deben eliminar todas las tablas temporarias.

Se debe garantizar que antes de ejecutar **migracion()** la tabla RECORRIDO_FINAL está vacía.

Ejemplo:

a) Se tienen los siguientes datos en el archivo test1.csv (una muestra de 23 filas del archivo **recorridos_realizados_2016.csv)**

1	PERIODO	ID_USUARIO	FECHA_HORA_RETIRO	ORIGEN_EST	NOMBRE_OF	DESTINO_ES	NOMBRE_DESTINO	TIEMPO_USC	FECHA_CREACION
2	201601	8	07/01/2016 19:53	9	PARQUE LAS	56	PLAZA PALERMO VIEJO	0H 8MIN 34S	01/12/2010
3	201601	8	13/01/2016 16:28	7	OBELISCO	56	PLAZA PALERMO VIEJO	0H 6MIN 49S	01/12/2010
4	201603	90	10/03/2016 08:46	23	SUIPACHA	12	PLAZA VICENTE LOPEZ	0H 4MIN 2SE	01/12/2010
5	201605	90	19/05/2016 09:51	16	LEGISLATURA	46	CHILE	0H 25MIN 56	01/12/2010
6	201611	328	01/11/2016 17:32	27	MONTEVIDE	5	PLAZA ITALIA	0H 20MIN 46	01/12/2010
7	201611	328	14/11/2016 20:59	3	ADUANA	27	MONTEVIDEO	0H 23MIN 12	01/12/2010
8	201611	328	14/11/2016 20:59	5	PLAZA ITALIA	5	ADUANA	0H 38MIN 0S	01/12/2010
9	201611	328	14/11/2016 20:59	5	PLAZA ITALIA	5	PLAZA ITALIA	0H 40MIN 0S	01/12/2010
10	201608	74710	13/08/2016 13:03	28	PLAZA BOED	528	PLAZA BOEDO	0H 19MIN 4S	11/06/2013
11	201608	74710	13/08/2016 13:09	28	PLAZA BOED	528	PLAZA BOEDO	0H 18MIN 9S	11/06/2013
12	201608	74710	13/08/2016 13:22	28	PLAZA BOED	528	PLAZA BOEDO	0H 52MIN 5S	11/06/2013
13	201608	74710	13/08/2016 13:28	28	PLAZA BOED	528	PLAZA BOEDO	0H 19MIN 46	11/06/2013
14	201608	74710	13/08/2016 13:35	28	PLAZA BOED	528	PLAZA BOEDO	0H 36MIN 12	11/06/2013
15	201608	74710	13/08/2016 13:42	28	PLAZA BOED	46	CHILE	0H 12MIN 52	11/06/2013
16	201608	74710	13/08/2016 13:57	28	PLAZA BOED	528	PLAZA BOEDO	0H 11MIN 35	11/06/2013
17	201608	74710	13/08/2016 14:06	28	PLAZA BOED	528	PLAZA BOEDO	0H 5MIN 5SE	11/06/2013
18	201608	74710	22/08/2016 08:25	28	PLAZA BOED	528	PLAZA BOEDO	0H 18MIN 50	11/06/2013
19	201608	74710	25/08/2016 08:11	28	PLAZA BOED	528	PLAZA BOEDO	0H 55MIN 31	11/06/2013
20	201609	74710	28/09/2016 18:50	28	PLAZA BOED	528	PLAZA BOEDO	0H 6MIN 2SE	11/06/2013
21	201609	74710	28/09/2016 19:37	28	PLAZA BOED	528	PLAZA BOEDO	0H 44MIN 45	11/06/2013
22	201609	74710	28/09/2016 19:56	28	PLAZA BOED	528	PLAZA BOEDO	0H 9MIN 18S	11/06/2013
23	201609	74710	29/09/2016 11:08	28	PLAZA BOED	528	PLAZA BOEDO	0H 36MIN 4S	11/06/2013

Luego de ejecutar:

```
SELECT migracion();
SELECT * FROM recorrido_final;
```

Se obtiene:

	periodo text	usuario integer	fecha_hora_ret timestamp without time zone	est_origen integer	est_destino integer	fecha_hora_dev timestamp without time zone
1	201601	8	2016-01-07 19:53:00	9	56	2016-01-07 20:01:34
2	201601	8	2016-01-13 16:28:00	7	56	2016-01-13 16:34:49
3	201603	90	2016-03-10 08:46:00	23	12	2016-03-10 08:50:02
4	201605	90	2016-05-19 09:51:00	16	46	2016-05-19 10:16:56
5	201611	328	2016-11-01 17:32:00	27	5	2016-11-01 17:52:46
6	201611	328	2016-11-14 20:59:00	5	5	2016-11-14 21:37:00
7	201608	74710	2016-08-13 13:03:00	28	46	2016-08-13 13:54:52
8	201608	74710	2016-08-13 13:57:00	28	528	2016-08-13 14:11:05
9	201608	74710	2016-08-22 08:25:00	28	528	2016-08-22 08:43:50
10	201608	74710	2016-08-25 08:11:00	28	528	2016-08-25 09:06:31
11	201609	74710	2016-09-28 18:50:00	28	528	2016-09-28 18:56:02
12	201609	74710	2016-09-28 19:37:00	28	528	2016-09-28 20:05:18
13	201609	74710	2016-09-29 11:08:00	28	528	2016-09-29 11:44:04

Explicación del resultado:

Las filas 2 a 6 de **test1.csv** coinciden individualmente con las filas 1 a 5 de RECORRIDO_FINAL y se calculó el atributo *fecha_hora_dev* en base a su correspondiente *fecha_hora_retiro+tiempo_uso.*
Las filas 7, 8 y 9 de **test1.csv** presentan un problema tipo 2) – clave duplicada, por lo cual dado que la segunda fila ordenada por *fecha_hora_retiro+tiempo_uso* es la 8, dicha fila es la única que pasa a la tabla y en el RECORRIDO_FINAL es la tupla 6.
Las filas 10 a 15 de **test1.csv,** correspondientes al usuario 74710, presentan un problema tipo 3) ya que a cada una de ellas corresponden los siguientes intervalos:

```
fila 10 [2016-08-13 13:03:00,2016-08-13 13:22:04]
fila 11 [2016-08-13 13:09:00,2016-08-13 13:27:09]
fila 12 [2016-08-13 13:22:00,2016-08-13 14:14:05]
fila 13 [2016-08-13 13:28:00,2016-08-13 13:47:46]
fila 14 [2016-08-13 13:35:00,2016-08-13 14:11:12]
fila 15 [2016-08-13 13:42:00,2016-08-13 13:54:52]
```

Por eso, en la tupla 7 de la tabla aparece el ISUM de dichos intervalos con la estación origen correspondiente a la fila 10 de **test1.csv** y la estación destino correspondiente a la fila 15 de **test1.csv.**

La fila 16 no presenta solapamiento con lo cual se transforma en la tupla 8 de la tabla RECORRIDO_FINAL.

Así continúa todo el análisis.

b) Asumiendo que se han migrado exitosamente los datos y que la tabla RECORRIDO_FINAL contiene los datos mostrados en el ejemplo anterior, los siguientes son los resultados obtenidos de intentar insertar las siguientes tuplas:

- `INSERT INTO recorrido_final VALUES('201601',8,'2016-01-18 16:28:00',23,23, '2016-01-13 20:28:00');`

Esta tupla se inserta sin problemas.

	periodo text	usuario integer	fecha_hora_ret timestamp without time zone	est_origen integer	est_destino integer	fecha_hora_dev timestamp without time zone
1	201601	8	2016-01-07 19:53:00	9	56	2016-01-07 20:01:34
2	201601	8	2016-01-13 16:28:00	7	56	2016-01-13 16:34:49
3	201601	8	2016-01-18 16:28:00	23	23	2016-01-13 20:28:00
4	201603	90	2016-03-10 08:46:00	23	12	2016-03-10 08:50:02
5	201605	90	2016-05-19 09:51:00	16	46	2016-05-19 10:16:56
6	201611	328	2016-11-01 17:32:00	27	5	2016-11-01 17:52:46
7	201611	328	2016-11-14 20:59:00	5	5	2016-11-14 21:37:00
8	201608	74710	2016-08-13 13:03:00	28	46	2016-08-13 13:54:52
9	201608	74710	2016-08-13 13:57:00	28	528	2016-08-13 14:11:05
10	201608	74710	2016-08-22 08:25:00	28	528	2016-08-22 08:43:50
11	201608	74710	2016-08-25 08:11:00	28	528	2016-08-25 09:06:31
12	201609	74710	2016-09-28 18:50:00	28	528	2016-09-28 18:56:02
13	201609	74710	2016-09-28 19:37:00	28	528	2016-09-28 20:05:18
14	201609	74710	2016-09-29 11:08:00	28	528	2016-09-29 11:44:04

- `INSERT INTO recorrido_final VALUES('201601',74710,'2016-09-29 11:30:00',23,23, '2016-09-29 11:32:00');`

Se produce una excepción con el cartel 'INSERCION IMPOSIBLE POR SOLAPAMIENTO' y no se producen cambios en la tabla.

4. Modalidad

El Trabajo Práctico estará disponible en Campus a partir del 07/06/2018, indicándose allí mismo, la fecha de entrega.

Se incluye junto con el enunciado:

a) El archivo **recorridos-realizados-2016.csv** tal como se encuentra en https://data.buenosaires.gob.ar/dataset/bicicletas-publicas.
b) Archivo para realizar pruebas **test1.csv** (ejemplo del enunciado).

El TP deberá realizarse en grupos de 3 alumnos y entregarse a través de la plataforma Campus ITBA hasta la fecha allí indicada.

5. Entregables

Los alumnos deberán entregar los siguientes documentos:

- El script sql **funciones.sql** con el código necesario para crear las tablas que utilicen, los comandos para la importación, todas las funciones y el trigger.

- Un informe que debe contener:
  - El rol de cada uno de los participantes del grupo. Si bien en el TP deben estar involucrados todos los integrantes, se debe asignar un rol de supervisión de cada una de las tareas. Mínimamente los roles son: encargado del informe, encargado de las funciones, encargado del trigger, encargado del funcionamiento global del proyecto y encargado de investigación. Pueden asignarse más roles en caso de requerirse.
  - Todo lo investigado para realizar el TP.

- Las dificultades encontradas y cómo se resolvieron.
- El informe debe tener como máximo 3 páginas.

6. Evaluación

La evaluación del trabajo se llevará a cabo teniendo en cuenta los parámetros establecidos en la rúbrica asociada a la actividad en Campus.

Se tendrá en cuenta que las consultas, más allá del funcionamiento (lo cual es fundamental), sean genéricas.

Los docentes ejecutarán el proceso usando los conjuntos de datos entregados.

El informe deberá estar completo y sin faltas de ortografía.

En caso de que el trabajo no cumpliera los requisitos básicos para ser aprobado, los alumnos serán citados en la fecha de recuperatorio para defenderlo y corregir los errores detectados.

GLOSARIO

1) Intervalo Solapado

Tomando como base las relaciones definidas para **Allen's Interval algebra**, https://en.wikipedia.org/wiki/Allen%27s_interval_algebra, definimos como **Intervalo Solapado** a aquello que coinciden con los siguientes casos:

X m Y
X o Y
X s Y
X d Y
X f Y
X = Y

2) ISUM (Intervalos Solapados Unificados Maximalmente): dados N intervalos $[I_i,F_i]$ con $1 \leq i \leq N$, ordenados por I_i, todos solapados de a pares, se denomina ISUM al intervalo $[I_1,F_N]$ que se obtiene de tomar como inicio el **I** del primer intervalo y como fin el **F** del último intervalo. Es decir, se deben **ordenar** todos los intervalos por su hora de inicio y tomar la hora de inicio del primer intervalo y como hora de finalización la hora de finalización del último intervalo según dicho orden.

Por ejemplo, si $[I_1,F_1]$ y $[I_2,F_2]$ son intervalos solapados y $[I_2,F_2]$ y $[I_3,F_3]$ son intervalos solapados, entonces el ISUM es $[I_1,F_3]$