

Base de Datos I

Trabajo Práctico Especial

1er Cuatrimestre 2020

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 y restricciones no disponibles de forma estándar (que no pueden resolverse con Primary Keys, Foreign Keys, etc.).

2. **Modalidad**

El Trabajo Práctico estará disponible en el Campus a partir del 11/06/2020, indicándose allí mismo la fecha de entrega.

Se incluye junto con el enunciado los archivos: **fed_emergency_disaster.csv**.

El TP deberá realizarse en grupos de 4 alumnos y entregarse a través de la plataforma Campus ITBA hasta la fecha allí indicada.

3. **Descripción del Trabajo**

Este trabajo especial está estructurado en un ejercicio.

Ejercicio

En el sitio de Kaggle se pueden encontrar distintos datasets de acceso público. En esta ocasión vamos a utilizar una reducción del dataset de Federal Emergencies and Disasters que incluye un registro de cada emergencia federal o desastre declarado por el Presidente de los Estados Unidos desde 1953.

La información es provista en un archivo CSV (Comma Separated Values). El archivo **fed_emergency_disaster.csv** contiene información de las emergencias o desastres federales declarados por Estados Unidos, tales como:

- Declaration Number.
- Declaration Type.
- Declaration Date (DD/MM/YYYY).
- State
- Disaster_Type.

Antes de insertar el archivo en una tabla de eventos, se quiere interceptar la inserción de la fecha y cambiarla por un FK a una dimensión temporal normalizada que está predefinida con la siguiente estructura:

```
create table YEAR
(
    year integer not null check (year < 2500),
    isleap boolean default false,
    primary key(year)
);

create table QUARTER
(
    id serial not null,
    quarternumber integer not null check(quarternumber between 1 and 4),
    yearfk integer not null,
    primary key(id),
    unique(quarternumber, yearfk),
    foreign key(yearfk) references year
);

create table MONTH
(
    id serial not null,
    monthid integer not null check (monthid between 1 and 12),
    monthdesc varchar(20) check (monthdesc in ('enero', 'febrero', 'marzo', 'abril', 'mayo',
    'junio', 'julio', 'agosto', 'septiembre', 'octubre', 'noviembre', 'diciembre')),
    quarterfk integer not null,
    primary key(id),
    unique(monthid, quarterfk),
    foreign key (quarterfk) references quarter
);

create table DATEDETAIL
(
    id serial not null,
    day integer not null check (day between 1 and 31),
    dayofweek varchar(20) check (dayofweek in ('lunes', 'martes', 'miercoles', 'jueves',
    'viernes', 'sabado', 'domingo')),
    weekend boolean default false,
    monthfk integer not null,
    primary key(id),
    unique(day, monthfk),
    foreign key(monthfk) references month
);
```

a) Creación de las tablas de la dimensión temporal.

Deben crearse las tablas mencionadas anteriormente, **YEAR**, **QUARTER**, **MONTH**, **DATEDETAIL**.

b) Creación de la tabla EVENT.

Debe crearse una tabla llamada **EVENT** que será la receptora de los datos del archivo **fed_emergency_disaster.csv**. Los campos y restricciones de la tabla deben crearse en base al análisis de los datos. Recordar que los archivos csv son archivos de texto que pueden abrirse fácilmente con cualquier editor. Se recomienda que los nombres de los campos coincidan con los de las columnas del archivo csv.

Para el caso particular del campo *Declaration Date*, se deberá cambiar su contenido para que el mismo haga referencia a la key de la tabla **DATEDETAIL** antes de insertarlo en la tabla **EVENT**.

En base a los datos, se deben crear las claves y constraints apropiados.

c) Importación de los datos

Utilizando el comando COPY de PostgreSQL, se deben importar TODOS los datos de los archivos **csv** en la tabla creada en el punto **b)**. Los datos de los archivos csv provistos por la cátedra NO pueden ser modificados.

Creación de un trigger para:

1) Determinar la FK de la dimensión temporal

Para insertar los datos en la tabla **EVENT** es necesario interceptar la inserción de la fecha y agregar la FK a la dimensión temporal de la tabla **DATEDETAIL**.

2) Cargar los valores de la estructura de la dimensión temporal

Además de insertar el evento, se deben poblar las distintas tablas que conforman la dimensión temporal, siempre y cuando las mismas no existan en dichas tablas.

Por ejemplo, si partimos de las tablas de la dimensión temporal vacías y si al principio en el archivo CSV viene la fecha '09/02/2020', se debe insertar una tupla en cada una de las tablas de tiempo, quedando las tablas de la siguiente manera:

YEAR	
<u>year</u>	isleap
2020	true

QUARTER		
<u>id</u>	quarternumber	yearfk
8	1	2020

MONTH			
<u>id</u>	monthid	monthdesc	quarterfk
12	2	febrero	8

DATEDETAIL				
<u>id</u>	day	dayofweek	weekend	monthfk
8	9	domingo	true	12

Sin embargo, si luego viene '10/01/2020' se reaprovecha el año 2020 y el primer quarter de ese año, quedando ambas tablas igual que antes:

YEAR	
<u>year</u>	isleap
2020	true

QUARTER		
<u>id</u>	quarternumber	yearfk
8	1	2020

Entonces solamente se deben insertar una tupla en **MONTH** y otro en **DATEDETAIL**, quedando las tablas de la siguiente manera:

MONTH			
<u>id</u>	monthid	monthdesc	quarterfk
12	2	febrero	8
13	1	enero	8

DATEDETAIL				
<u>id</u>	day	dayofweek	weekend	monthfk
8	9	domingo	true	12
9	10	viernes	false	13

Si luego viene nuevamente para otro evento para el '09/02/2020', no tiene que insertar ninguna tupla.

Las tuplas en las tablas tienen que tener todos los datos bien completos (isleap, monthdesc, dayofweek, weekend, etc).

4. Entregables

Los alumnos deberán entregar los siguientes documentos:

- El script sql **funciones.sql** con el código necesario para crear/borrar las tablas, las funciones y los triggers. Tener en cuenta que solamente se ejecutará sólo un único COPY en la línea de comandos de PostgreSQL para para poblar automáticamente tanto la dimensión temporal como la tabla **EVENT**, según lo explicado más arriba.
- 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 de los triggers, 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.
 - También se debe detallar aquí el proceso de importación de los datos realizado.
 - El informe debe tener como máximo 3 páginas.

5. Evaluación

La evaluación del trabajo se llevará a cabo utilizando los parámetros establecidos en la rúbrica asociada a la actividad en el 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 pero podrán también hacer pruebas con otros conjuntos de datos de similares características para evaluar el funcionamiento en distintos escenarios.

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.