

Base de Datos I

Trabajo Práctico Especial

1er Cuatrimestre 2017

1. Objetivo

El objetivo de este Trabajo Práctico Especial es aplicar, según corresponda, los conceptos de SQL Avanzado (PSM, Triggers) vistos a lo largo del curso.

2. Contexto

Se le llama dominio a un nombre único para identificar URLs o páginas en internet (por ejemplo: google.com).

Existe un protocolo llamado WHOIS que permite obtener información del propietario de un dominio. Existe servidores llamados WHOIS Servers, que almacenan dicha información, es decir, la asociación entre dominio y propietario. Así la información puede consultarse (Lookup).

Por ejemplo, ingresando a <http://whois.domaintools.com/> se pueden verificar los datos del dominio itba.edu.ar.

3. Datos

La tabla WHOIS_SAMPLE muestra la estructura utilizada para almacenar la información de necesaria sobre dominios y propietarios.

```
CREATE TABLE whois_sample
(
  domainName          TEXT PRIMARY KEY,
  registrarName       TEXT,
  contactEmail        TEXT,
  whoisServer         TEXT,
  nameServer          TEXT,
  createDate          DATE,
  updatedAt           DATE,
  expiresDate         DATE,
  status              TEXT,
  registrant_email    TEXT,
  registrant_name     TEXT,
  registrant_organization TEXT,
  registrant_street1 TEXT,
  registrant_street2 TEXT,
  registrant_city     TEXT,
  registrant_state    TEXT,
  registrant_postalCode TEXT,
  registrant_country  TEXT,
  registrant_fax      TEXT,
  registrant_telephone TEXT
);
```

Cada dominio registrado en la tabla contiene los datos de la persona o entidad que lo ha registrado junto con sus datos geográficos tales como: país, estado, ciudad, código postal, calle, fax y teléfono.

Además, contiene las fechas de creación, actualización y expiración del dominio

4. Ejercicios

4.1) Crear la tabla WHOIS_SAMPLE y poblarla importando los datos extraídos del archivo **whois_sample.csv**, utilizando el comando COPY FROM de Postgres SQL.

<https://www.postgresql.org/docs/9.2/static/sql-copy.html>

Tener en cuenta que si la importación se ejecuta en una base Postgres local, se deben otorgar permisos de lectura y ejecución al usuario de la base de datos para que pueda acceder al csv

En caso de ejecutarlo en el server Pampero, es necesario hacerlo por consola usando \COPY.

El comando para ejecutar un script sql por consola es el siguiente

```
psql -h bdl.it.itba.edu.ar -U nombreusuario -f script.sql PROOF
```

Se deberá entregar el script **importación.sql** con los comandos usados para importar los datos.

4.2)

Definiciones

- Se denomina **valor central** de una lista de **N** elementos, al valor que se obtiene de ordenar alfabéticamente de manera ascendente (teniendo en cuenta mayúsculas y minúsculas), los mencionados **N** elementos y tomar el que se encuentra en la posición $[n/2]+1$.
- Se denomina **dominio expirado** al dominio cuya fecha de expiración es menor a la fecha del día.
- Se denomina **dominio activo** al dominio cuya fecha de expiración es mayor o igual a la fecha del día.

Dado un código postal, se quiere conocer la cantidad de dominios expirados y la cantidad de dominios activos. Para ello, se ha creado la siguiente tabla:

```
create table muestra(  
country      TEXT,  
postalcode   TEXT,  
city         TEXT,  
status       TEXT CHECK (status IN ('ACTIVO','EXPIRADO')),  
quantity     INTEGER,  
PRIMARY KEY(postalcode, country, status));
```

En esta tabla, se quieren almacenar las cantidades de servidores activos y expirados pertenecientes a un cierto código postal y país teniendo en cuenta que un mismo código postal puede existir en varios países.

Se pide escribir la función **inserta_estado** que recibe como parámetro un país y un código postal e inserta en la tabla MUESTRA dos tuplas representativas con la siguiente información:

Tupla 1:

- *Country*: corresponde al valor del país pasado como parámetro.
- *PostalCode*: corresponde al valor de código postal pasado como parámetro.
- *City*: dado que para un mismo país y código postal pueden existir múltiples ciudades, sólo se almacenará el valor central (definición **a**) de las ciudades correspondientes a los dominios activos del país y código postal pasados como parámetro.
- *Status*: ACTIVO
- *Quantity*: cantidad de dominios activos según la definición c).

Tupla 2:

- *Country*: corresponde al valor del país pasado como parámetro.
- *PostalCode*: corresponde al valor de código postal pasado como parámetro.
- *City*: dado que para un mismo país y código postal pueden existir múltiples ciudades, sólo se almacenará el valor central (definición **a**) de las ciudades correspondientes a los dominios expirados del país y código postal pasados como parámetro.
- *Status*: EXPIRADO
- *Quantity*: cantidad de dominios expirados según la definición **b**).

En caso de que se produzca un error por Primary Key duplicada, la función **inserta_estado** debe informar por consola que ya existe un *status* almacenado para el Código Postal y País, terminando normalmente pero evitando el INSERT

Asumimos que las tablas WHOIS_SAMPLE y MUESTRA existen.

No se sabe de antemano la cantidad de valores que puede tener la tabla WHOIS_SAMPLE, los cuales pueden ser muchos. Los datos son sólo una instancia posible.

No se debe generar una estructura auxiliar (array o vector) de datos para manejarlos.

En caso de que los parámetros no existan en la tabla o en cualquier otro caso de error, no se deben insertar tuplas en la tabla MUESTRA y la función debe terminar normalmente.

Ejemplo 1:

Si la Fecha del sistema es: 2013-05-30, y las tablas de dominio, inscripto y administrador están vacías, al ejecutar `select inserta_estado(2000, 2)`, la tabla MUESTRA debería quedar con los datos que tenía antes de la invocación de la función.

Ejemplo 2:

Si la Fecha del sistema es: 2017-06-07, y las tablas están cargadas con la data de ejemplo, al ejecutar `select inserta_estado('AUSTRALIA', '2142')`, debería mostrarse:

MUESTRA				
Country	PostalCode	City	Status	Quantity
AUSTRALIA	2142		EXPIRADO	0
AUSTRALIA	2142	South Granville	ACTIVO	3

Explicación:

Para AUSTRALIA, con código postal 2010 existen las siguientes ciudades correspondiente a dominios activos:

"Granville"

"GRANVILLE"

"South Granville"

En este caso, como hay 3 valores, el valor-central elegido es GRANVILLE.

Dado que no hay dominios expirados, el atributo *city* queda vacío en ese caso.

4.3)

Definición

Se denominan **n-vecinos** de un *código postal inicial* a los N códigos postales siguientes a dicho código postal, luego de ordenarlos alfabéticamente en orden ascendente.

Se pide crear la función **nvecinos** que recibe como parámetro el nombre de un país, un código postal, que llamaremos código postal inicial, y un número e inserta en la tabla MUESTRA, dicho código postal más sus N códigos postales vecinos, invocando a la función *inserta_estado* creada en el ítem 4.2.

En caso de que de que el código postal solicitado no existiera, se tomará como **código postal inicial** al código postal siguiente en el mismo orden alfabético y se insertarán en la tabla MUESTRAS los n-vecinos de este nuevo código inicial.

Si existieran menos vecinos que la cantidad indicada por N, sólo se guardarán en MUESTRAS los vecinos existentes.

Si el valor de N es 0, sólo se insertará el código postal inicial.

Si el valor de N es menor que 0, la tabla MUESTRA queda intacta, es decir, no se inserta nada ni se borra nada.

Dado que la tabla MUESTRA puede no estar vacía, esta función debe ocuparse de vaciarla totalmente antes de comenzar las inserciones.

Ejemplo 1

Si invocamos a la función `SELECT nvecinos('UNITED STATES','20003',4)`, al ejecutar `SELECT * FROM muestra`; se obtiene:

Country	postalcode	City	status	quantity
UNITED STATES	20003		ACTIVO	0
UNITED STATES	20003	Washington	EXPIRADO	1
UNITED STATES	20006		ACTIVO	0
UNITED STATES	20006	Washington	EXPIRADO	1
UNITED STATES	20007	Washington	ACTIVO	2
UNITED STATES	20007	Washington	EXPIRADO	4
UNITED STATES	20009	Washington	ACTIVO	1
UNITED STATES	20009		EXPIRADO	0

UNITED STATES	20062		ACTIVO	0
UNITED STATES	20062	Washington	EXPIRADO	1

En este caso, la tabla muestra contiene 2 tuplas para el código postal 2003 y 2 tuplas para cada uno de sus 4 códigos postales posteriores.

Ejemplo 2

Si invocamos a la función `SELECT nvecinos ('ARGENTINA','9405',8)`, al ejecutar `SELECT * FROM muestra;` se obtiene:

Country	postalcode	City	status	quantity
ARGENTINA	9405	El Calafate	ACTIVO	1
ARGENTINA	9405		EXPIRADO	0
ARGENTINA	C1425BHN		ACTIVO	0
ARGENTINA	C1425BHN	Sarandí	EXPIRADO	1
ARGENTINA	C1427BRJ		ACTIVO	0
ARGENTINA	C1427BRJ	Autonoma de Buenos Aires	EXPIRADO	1

En este caso, el código postal 9405 sólo tenía 2 vecinos, por eso no se insertaron los ocho vecinos solicitados en la invocación de la función.

Ejemplo 3

Si invocamos a la función `SELECT nvecinos ('ARGENTINA','55555',2)`, al ejecutar `SELECT * FROM muestra;` se obtiene:

Country	postalcode	City	status	quantity
ARGENTINA	6530		ACTIVO	0
ARGENTINA	6530	Carlos Casares	EXPIRADO	1
ARGENTINA	7600		ACTIVO	0
ARGENTINA	7600	Mar Del Plata	EXPIRADO	1
ARGENTINA	8000		ACTIVO	0
ARGENTINA	8000	Bahía Blanca	EXPIRADO	1

En este caso, el código postal 55555 no existe en la tabla `WHOIS_SAMPLE`. El siguiente código postal en orden alfabético para Argentina, es el 6530. Por eso, se inserta en `MUESTRA`, dicho código postal y sus 2 vecinos.

4. Modalidad

El TP deberá realizarse en grupos de 2 alumnos y se entregará a través de la plataforma Campus ITBA hasta la fecha allí indicada.

5. Entregables

Los alumnos deberán entregar los siguientes documentos

- El script sql **importación.sql** con los comandos utilizados para insertar los datos en las tablas
- El script sql **funciones.sql** con los códigos de todas las funciones creadas.