

2019

Base de Datos I

Informe TPE

Integrantes: Britos, Nicolás Ignacio (59529)

Mail: nbritos@itba.edu.ar

Oliva, Juan Martín (58664)

Roca, Agustín (59160)

Fecha de entrega: 14 de noviembre 2019

ITBA



Trabajo Práctico Especial

1. Autores:

*Britos, Nicolás Ignacio:

Legajo: 59529.

Mail: nbritos@itba.edu.ar

Rol: Encargado de los triggers y del informe.

*Oliva, Juan Martín:

Legajo: 58664.

Rol: Encargado de las funciones.

*Roca, Agustín:

Legajo: 59160.

Rol: Encargado de investigación y del funcionamiento general del proyecto.

1. Uso

1.1. Ejercicio 1

Ejecutando el archivo “funciones.sql” se crearán todas las tablas y funciones utilizadas en el primer ejercicio. Sin embargo, como los datos se encuentran en archivos CSV es necesario ejecutar la sentencia:

```
SELECT copy_contrato(PATH_TO_CSV);
```

reemplazando “PATH_TO_CSV” con la dirección al archivo de CSV a cargar (que posee el mismo formato que el incluido con el enunciado). Esta función se encargará de realizar el correcto parseo de los valores en este archivo y cargarlos en la tabla CONTRATO.

Luego, para cargar la tabla “RESUMENCONTRATO” es necesario ejecutar la sentencia:

```
SELECT CargaResumenContrato();
```

1.2. Ejercicio 2

Ejecutando el archivo “funciones.sql” se crearán todas las tablas y funciones utilizadas en el segundo ejercicio. No es necesario ejecutar ninguna otra sentencia adicional. Luego, los ítems del enunciado pueden probarse con las mismas sentencias provistas en el enunciado.

2. Decisiones de implementación y dificultades encontradas

2.1. Ejercicio 1

Sobre la implementación de este ejercicio (en términos de funciones y triggers) la misma fue bastante directa debido a la poca cantidad de código SQL necesario para realizarlo y al alto acoplamiento del mismo.

A medida que fuimos desarrollando el primer ejercicio fueron surgiendo varios problemas, el primero de ellos relacionado con la carga de datos CSV utilizando la base de datos provista por la cátedra (bajo Pampero): resulta que no es posible realizar esto si el usuario no es superadmin. Debido a esto, trabajamos con conexiones locales en nuestras propias computadoras a PostgreSQL.

Además del problema de carga de archivos, nos encontramos con que PostgreSQL no reconoce el formato de fecha presente en ellos al cargar el archivo de manera directa, por lo que envolvimos la sentencia “COPY” en la función “copy_contrato” (la cual recibe el filepath del CSV) que crea una tabla temporaria, similar a la tabla “CONTRATO” pero con los tipos “TEXT” en los campos de “FechaDesde” y “FechaHasta”. Una vez cargados los datos en esta tabla temporaria, es posible volcar los datos a la tabla “CONTRATO” formateando, con funciones provistas por PostgreSQL, a un formato de fecha reconocido.

2.2. Ejercicio 2

Durante el desarrollo de este ejercicio no nos encontramos con ninguna dificultad grave que no se solucionó leyendo el manual de PostgreSQL ya que todo lo implementado fue visto en clase y en las prácticas.

Este ejercicio, al ser más complejo en termino de datos y funciones, optamos por modularizarlo y crear varias funciones que pueden ser luego utilizadas en otros entornos, las cuales son: “get_n_old_passwords”, “repeats_password” y “validacion_contrasena_funcion”. Esta última se vale de las primeras dos para determinar si la contraseña es valida (cumple con lo especificado en el enunciado) y, si es así, agrega una entrada al historial de contraseñas con la información pertinente. Las primeras dos funciones no son más que queries SQL que determinan el máximo nivel de rol de un usuario y si el mismo repite alguna contraseña del historial de contraseñas. La función “validacion_contrasena_funcion” es utilizada por el trigger “validacion_contrasena_trigger”, el cual es ejecutado antes de cada actualización sobre la tabla “USUARIO” como se pide en el enunciado.

3. Búsqueda de información

Con respecto a la búsqueda de información para completar el trabajo practico, nos valimos de las clases teóricas subidas al campus que tratan de SQL3 y del manual de PostgreSQL disponible online.