

Base de Datos I

Trabajo Práctico Especial

2do Cuatrimestre 2019

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 07/11/2019, indicándose allí mismo la fecha de entrega.

Se incluye junto con el enunciado los archivos: **alquileres.csv** y **alquileres2.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 2 ejercicios.

Ejercicio 1

Para este ejercicio hemos decidido usar un archivo CSV (Comma Separated Values). El archivo **alquileres.csv** contiene información de los contratos de alquileres de distintos inmuebles administrados por una inmobiliaria, tales como:

- Fecha Desde Contrato (YYYYMM).
- Fecha Hasta Contrato (YYYYMM).
- ID del Departamento.
- ID del Inquilino.

El intervalo del contrato es cerrado al inicio y abierto al final, es decir, el año y mes de finalización no se encuentran incluidos. Es frecuente que el mismo inquilino decida volver a alquilar el mismo departamento para el siguiente período de tiempo disponible. En dichos casos, se genera una nueva entrada cuya fecha de inicio coincide con la fecha de finalización del contrato anterior. También es posible que se decida no seguir con el contrato. En estos casos el departamento es alquilado por otro inquilino. A veces el nuevo inquilino realiza el contrato en el siguiente período de tiempo inmediatamente disponible o a veces lo realiza un tiempo posterior dejando al departamento disponible.

a) Creación de la tabla CONTRATO.

Debe crearse una tabla llamada **CONTRATO** que será la receptora de datos del archivo **alquileres.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.

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

b) 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 **a)**. Los datos de los archivos csv provistos por la cátedra NO pueden ser modificados.

A continuación se presenta un ejemplo de los primeros registros de la tabla con datos cuyo origen es el archivo **alquileres.csv**:

CONTRATO			
FechaDesde	FechaHasta	Deptold	Personald
200504	200704	1	1
201107	201307	2	3
200904	201104	1	1
200709	200909	4	6
201104	201304	1	1
200507	200707	2	2
200704	200904	1	1
200707	200907	2	2
200907	201107	2	3
200511	200711	3	4
200801	201001	3	5
200509	200709	4	6

c) Creación de la tabla RESUMENCONTRATO y función CargarResumenContrato.

Debe crearse una tabla llamada **RESUMENCONTRATO** para guardar un reporte de los contratos por departamento. En dicho resumen, se van a ingresar los períodos de tiempo en el cual un departamento estaba siendo alquilado (independientemente de quién fue el inquilino).

La tabla **RESUMENCONTRATO** tiene la siguiente estructura:

RESUMENCONTRATO		
FechaDesde	FechaHasta	Deptold

Luego de ejecutar la función `CargaResumenContrato()`, la tabla `RESUMENCONTRATO` debe contener los siguientes datos, con los datos de ejemplo de los primeros registros:

RESUMENCONTRATO		
FechaDesde	FechaHasta	Deptold
200504	201304	1
200507	201307	2
200511	200711	3
200801	201001	3
200509	200909	4

Ejercicio 2

Para este ejercicio vamos a utilizar las siguientes tablas que conforman el módulo de autenticación de un sistema:

USUARIO	
Nombre	Password
jperez	pass1
mgomez	pass1
tbalbin	pass1
ucampos	pass1

ROL	
Nombre	Nivel
secretaria	0
gerente	1
revisor	2

ROLES	
Usuario	Rol
jperez	secretaria
mgomez	secretaria
tbalbin	secretaria
tbalbin	gerente
ucampos	revisor

, donde la tabla **ROLES** tiene dos claves foráneas: una hacia la tabla **USUARIO** y otra hacia la tabla **ROL**.

HISTORIALPASSWORD		
Usuario	Password	Fecha
mgomez	pass2	01/01/2019 00:00:00
tbalbin	pass15	01/01/2019 00:00:00
tbalbin	pass44	01/02/2019 00:00:00
ucampos	pass2	01/01/2019 00:00:00
ucampos	pass3	01/02/2019 00:00:00
ucampos	pass4	01/03/2019 00:00:00

Esta última tabla almacena las contraseñas no actuales de los usuarios.

a) Creación de las tablas e inserción de los datos.

Debe crearse las tablas descriptas anteriormente (**USUARIO**, **ROL**, **ROLES** y **HISTORIALPASSWORD**), teniendo en cuenta las diferentes características:

- El campo **Fecha** de la tabla **HISTORIALPASSWORD** es de tipo **TimeStamp**.
- El campo **Nivel** de la tabla **ROL** es de tipo **Integer** y es afectado por un **check** que verifica que el valor sea mayor o igual a cero.
- Todos los campos restantes de las tablas son del tipo **Text**.

Luego se deben insertar los registros mostrados anteriormente en las 4 tablas.

b) Creación de un trigger para validar la actualización de la contraseña.

Se desea agregar una validación con respecto a la actualización de la contraseña actual de un usuario, de tal manera que si alguna de las siguientes 2 condiciones no se verifican entonces la operación se rechaza (se debe lanzar una excepción):

- La nueva contraseña no puede coincidir con la clave que posee actualmente.
- La nueva contraseña no puede utilizar alguna contraseña de las almacenadas anteriormente en **HISTORIALPASSWORD** dependiendo de los roles asignados al usuario.
 - Aquellos usuarios con roles de Nivel = 0 no deben ser afectados por esta validación.
 - Por otro lado, los usuarios con roles de Nivel ≥ 1 deben implementarse medidas de mayor seguridad. Para esto se decidió que los usuarios no pueden reutilizar ninguna de las últimas N contraseñas anteriores almacenadas en **HISTORIALPASSWORD**, siendo N el valor máximo de todos los niveles correspondiente a los roles que tiene el usuario asignado.
- Adicionalmente si luego de las validaciones, corresponde la actualización de la nueva contraseña en la tabla **USUARIO**, se requiere que la contraseña anterior sea almacenada en la tabla **HISTORIALPASSWORD** con la fecha y hora del cambio.

En resumen, se debe crear un Trigger que realice las validaciones de la contraseña según lo explicado, y adicionalmente ingrese la información en la tabla **HISTORIALPASSWORD** en caso de corresponder.

Ejemplos:

- Si el usuario “jperez” desea cambiar su contraseña, debido a que no tiene registrado ninguna contraseña anterior en la tabla **HISTORIALPASSWORD**, solamente no va a poder usar la contraseña actual que tiene. Esto es independiente de los roles que tenga asignado.

```
UPDATE USUARIO SET PASSWORD = 'pass5' WHERE NOMBRE = 'jperez';
```

USUARIO	
Nombre	Password
jperez	pass5
mgomez	pass1
tbalbin	pass1
ucampos	pass1

HISTORIALPASSWORD		
Usuario	Password	Fecha
mgomez	pass2	01/01/2019 00:00:00
tbalbin	pass15	01/01/2019 00:00:00
tbalbin	pass44	01/02/2019 00:00:00
ucampos	pass2	01/01/2019 00:00:00
ucampos	pass3	01/02/2019 00:00:00
ucampos	pass4	01/03/2019 00:00:00
jperez	pass1	fecha y hora del cambio

- Si el usuario “mgomez” desea cambiar su contraseña, debido a que el rol que tiene asignado corresponde a un nivel 0 (cero), solamente no va a poder usar su contraseña actual.

```
UPDATE USUARIO SET PASSWORD = 'pass1' WHERE NOMBRE = 'mgomez';
```

Se recibe una excepción con el siguiente mensaje "No puede repetir el password anterior" y las tablas USUARIO e HISTORIALPASSWORD quedan iguales.

UPDATE USUARIO SET PASSWORD = 'pass2' WHERE NOMBRE = 'mgomez';

USUARIO	
Nombre	Password
jperez	pass5
mgomez	pass2
tbalbin	pass1
ucampos	pass1

HISTORIALPASSWORD		
Usuario	Password	Fecha
mgomez	pass2	01/01/2019 00:00:00
tbalbin	pass15	01/01/2019 00:00:00
tbalbin	pass44	01/02/2019 00:00:00
ucampos	pass2	01/01/2019 00:00:00
ucampos	pass3	01/02/2019 00:00:00
ucampos	pass4	01/03/2019 00:00:00
jperez	pass1	fecha y hora del cambio
mgomez	pass1	fecha y hora del cambio

- Si el usuario "tbalbin" desea cambiar la contraseña, además de la contraseña actual, no va a poder reutilizar la última contraseña registrada en el histórico (el máximo nivel asignado es 1). Entonces, no va a poder usar "pass44". Si podría usar "pass15".

UPDATE USUARIO SET PASSWORD = 'pass44' WHERE NOMBRE = 'tbalbin';

Se recibe una excepción con el siguiente mensaje "El password no debe coincidir con ninguno de los 1º passwords anteriores" y las tablas USUARIO e HISTORIALPASSWORD quedan iguales.

UPDATE USUARIO SET PASSWORD = 'pass30' WHERE NOMBRE = 'tbalbin';

USUARIO	
Nombre	Password
jperez	pass5
mgomez	pass2
tbalbin	pass30
ucampos	pass1

HISTORIALPASSWORD		
Usuario	Password	Fecha
mgomez	pass2	01/01/2019 00:00:00
tbalbin	pass15	01/01/2019 00:00:00
tbalbin	pass44	01/02/2019 00:00:00
ucampos	pass2	01/01/2019 00:00:00
ucampos	pass3	01/02/2019 00:00:00
ucampos	pass4	01/03/2019 00:00:00
jperez	pass1	fecha y hora del cambio
mgomez	pass1	fecha y hora del cambio
tbalbin	pass1	fecha y hora del cambio

- Si el usuario “ucampos” desea cambiar la contraseña, además de la contraseña actual, no va a poder reutilizar las últimas dos contraseñas registradas en el histórico (el máximo nivel asignado es 2). Entonces, no va a poder usar “pass3” ni “pass4”. Si podría usar “pass2”.

UPDATE USUARIO SET PASSWORD = 'pass3' WHERE NOMBRE = 'ucampos';

Se recibe una excepción con el siguiente mensaje "El password no debe coincidir con ninguno de los 2º passwords anteriores" y las tablas USUARIO e HISTORIALPASSWORD quedan iguales.

UPDATE USUARIO SET PASSWORD = 'pass2' WHERE NOMBRE = 'ucampos';

USUARIO	
Nombre	Password
jperez	pass5
mgomez	pass2
tbalbin	pass30
ucampos	pass2

HISTORIALPASSWORD		
Usuario	Password	Fecha
mgomez	pass2	01/01/2019 00:00:00
tbalbin	pass15	01/01/2019 00:00:00
tbalbin	pass44	01/02/2019 00:00:00
ucampos	pass2	01/01/2019 00:00:00
ucampos	pass3	01/02/2019 00:00:00
ucampos	pass4	01/03/2019 00:00:00
jperez	pass1	fecha y hora del cambio
mgomez	pass1	fecha y hora del cambio
tbalbin	pass1	fecha y hora del cambio
ucampos	pass1	fecha y hora del cambio

- Si se quiere hacer un update sobre un usuario que no existe en la tabla de USUARIO, las tablas USUARIO e HISTORIALPASSWORD quedan iguales porque no se realiza ninguna actualización.

UPDATE USUARIO SET PASSWORD = 'pass1' WHERE NOMBRE = 'salerno';

- Si se realiza un update masivo para todos los usuarios

UPDATE USUARIO SET PASSWORD = left(PASSWORD, length(PASSWORD)-1) || '3';

USUARIO	
Nombre	Password
jperez	pass3
mgomez	pass3
tbalbin	pass33
ucampos	pass33

HISTORIALPASSWORD		
Usuario	Password	Fecha
mgomez	pass2	01/01/2019 00:00:00
tbalbin	pass15	01/01/2019 00:00:00
tbalbin	pass44	01/02/2019 00:00:00
ucampos	pass2	01/01/2019 00:00:00
ucampos	pass3	01/02/2019 00:00:00
ucampos	pass4	01/03/2019 00:00:00
jperez	pass1	fecha y hora del cambio
mgomez	pass1	fecha y hora del cambio
tbalbin	pass1	fecha y hora del cambio
ucampos	pass1	fecha y hora del cambio
jperez	pass5	fecha y hora del cambio
mgomez	pass2	fecha y hora del cambio
tbalbin	pass30	fecha y hora del cambio
ucampos	pass2	fecha y hora del cambio

4. Entregables

Los alumnos deberán entregar los siguientes documentos:

- El script sql **funciones.sql** con el código necesario para crear las tablas, insertar los registros, las funciones y los triggers.
- 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. En el caso del Ejercicio 1, les brindamos un dataset adicional llamado **alquileres2.csv** que respeta el mismo formato para hacer pruebas.

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.