

## **Trabajo Práctico Especial**



### **Grupo 12**

- Micaela Perillo (62625) - miperillo@itba.edu.ar
- Alba Bjorkman (65989) - abjorkman@itba.edu.ar
- Nicolas Dario Casella (62311) - ncasella@itba.edu.ar

### **Profesores**

- Leticia Irene Gomez
- Cecilia Rodriguez Babino

**Materia:** Bases de Datos I (72.37)

Segundo Cuatrimestre, 2023

## Introducción

En el presente informe, detallaremos acerca de la implementación de nuestro trabajo práctico desarrollado usando los conceptos avanzados de SQL para implementar funcionalidades y restricciones no disponibles de forma estándar.

## Roles

La división de tareas propuesta fue la siguiente

- Nicolás Casella: encargado del trigger
- Micaela Perillo: encargada del informe
- Alba Bjorkman: encargada de las funciones

Los otros roles, encargado del funcionamiento global del proyecto y encargado de investigación, fueron llevados a cabo por todo el grupo. Se decidió esto debido a que consideramos que era importante que todos podamos entender que estábamos haciendo y verificar que el trabajo funcionará correctamente.

## Investigación previa

Antes de comenzar con el trabajo, decidimos ponerlos al día con la teoría y los trabajos prácticos de la materia. Fue suficiente para que no tengamos que buscar más información acerca de funciones y triggers. Fuera de eso, investigamos el uso de “Infinity” en las bases de datos y también buscamos información sobre cómo importar un archivo csv a la base de datos, con el comando *COPY*.

## Desarrollo

### Importación de Datos

Para importar los datos, primero se debe abrir una conexión en Pampero con el siguiente comando

```
ssh nombreUsuario@pampero.itba.edu.ar
```

Luego, se debe ejecutar

```
psql -h bd1.it.itba.edu.ar -U nombreusuario PROOF
```

Ya desde la línea de comandos de psql se corre el siguiente comando, que importará todos los datos del archivo csv a la tabla empleado.

```
\COPY empleado FROM empleados.csv csv header delimiter ;
```

### Desarrollo del Trabajo

Primero creamos las tablas *empleado* y *empleado\_tt*, siguiendo las especificaciones del enunciado. Luego, definimos la función *empregister*, que según

el TG\_OP, que puede ser “INSERT”, “UPDATE” o “DELETE”, inserta o actualiza los valores en la tabla *empleado\_tt* según corresponda.

Por otro lado, en la función *historial\_empleados*, se recibe un *timestamp* y se imprime un reporte mostrando para cada estado (Vigente, Vigente Anterior y No Vigente), los valores de los atributos de empleado, junto al número de cambios realizados para cada empleado. El estado se obtuvo con una función auxiliar *checkState()* que decide según *tt\_izq* y *tt\_der*.

## Problemas

Tuvimos problemas inicialmente con *Infinity*, ya que no lo detectaba por un error que luego nos dimos cuenta que era de sintaxis. Para arreglarlo, se nos había ocurrido utilizar el máximo valor de *Timestamp* en vez de *Infinity*, pero al final no fue necesario.

También, tuvimos problemas en la importación de datos desde el .csv. Al intentar hacerlo, se obtenía un error “*Permission denied*”. Esto se solucionó porque lo estábamos haciendo desde DBVisualizer y al hacer un *COPY* no nos daba los permisos necesarios, pero al hacer un \COPY sí.