

Base de Datos I

Trabajo Práctico Especial

1^{er} Cuatrimestre 2024

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 06/06/2024, indicándose allí mismo la fecha de entrega.

Se incluye junto con el enunciado el archivo: **pedido_aula.csv**.

El TP deberá realizarse en grupos de 3 alumnos, a excepción de un grupo de 4 alumnos, y entregarse a través de la plataforma Campus ITBA hasta la fecha allí indicada.

3. Descripción del Trabajo

Se tiene una base de datos para registrar la asignación de aulas para los exámenes. Antes de realizar la asignación se requiere validar que no haya superposición de fechas y horarios para el aula en cuestión.

Supongamos que se tiene tabla **aula_examen** (**nroaula**, **fecha_hora**, **duracion**, **codmateria**, **confirmado**), donde *nroaula* almacena caracteres numéricos, *fecha* almacena fechas con horas, *duración* almacena el intervalo de tiempo [cerrado, abierto) de horas requerido para el examen en cuestión, *codmateria* almacena caracteres alfanuméricos y *confirmado* almacena valores verdadero o falso.

La columna *confirmado* se asigna como TRUE automáticamente si la banda horaria solicitada no está ocupada. Caso contrario se asigna como FALSE.

En este Trabajo Práctico se implementará este procedimiento para una tabla **aula_examen** (**nroaula**, **fecha_hora**, **duracion**, **codmateria**, **confirmado**) que deberán crear de acuerdo a los requerimientos. Adicionalmente, se proveerá un archivo CSV (Comma Separated Values) con información para realizar el trabajo. El archivo contiene información de los pedidos de asignación de aulas para distintos exámenes de distintas materias.

Las columnas del archivo son:

- **Aula:** Número del aula a asignar
- **Fecha y hora inicio:** Fecha del examen a asignar con la hora de inicio
- **Intervalo:** Intervalo de horas a asignar para la fecha en cuestión
- **Materia:** Código de materia del examen a asignar

La finalidad de este Trabajo Práctico Especial consiste en implementar lo antes descripto. Específicamente se debe hacer lo siguiente:

- a) Crear la tabla **aula_examen**
- b) Implementar un trigger para completar automáticamente la columna *confirmado*

- c) Importar los datos y cargar la tabla *aula_examen*
- d) Implementar la función para analizar los casos de asignación exitosa y no exitosa.

4. Explicación paso a paso

a) Creación de la tabla *aula_examen*

Debe crearse la tabla *aula_examen* con los tipos de datos adecuados para almacenar los datos procedentes del archivo CSV, adicionando el valor correspondiente de la columna *confirmado*. Para la columna *duracion* es obligatorio usar el tipo de dato "interval" de PostgreSQL. Definir la clave y las restricciones según corresponda.

b) Implementación de un trigger para completar automáticamente la columna *confirmado*

Debe implementarse mediante un trigger, el manejo de la columna *confirmado* de la tabla *aula_examen*, para confirmar si la asignación del aula fue exitosa o no. No está permitido usar el operador OVERLAPS de PostgreSQL.

Por ejemplo:

- Si el usuario realiza la operación *insert into aula_examen (nroaula, fecha_hora, duracion, codmateria) values (10, '12/10/2024 9:00:00,' interval '3 hour', '72.34');* entonces si asumimos que la tabla comienza vacía, se asigna el aula completando la columna *confirmado* con TRUE, quedando la tabla *aula_examen* de la siguiente manera:

AULA_EXAMEN				
nroaula	fecha_hora	duracion	codmateria	confirmado
10	12/10/2024 09:00:00	03:00:00	72.34	TRUE

- Si luego el usuario realiza la operación *insert into aula_examen (nroaula, fecha_hora, duracion, codmateria) values (10, '12/10/2024 8:00:00,' interval '2 hour', '72.37');* entonces debemos detectar que el intervalo solicitado se superpone con el intervalo asignado para la materia 72.34, por lo tanto, NO se asigna el aula, completando la columna *confirmado* con FALSE, quedando la tabla *aula_examen* de la de la siguiente manera:

AULA_EXAMEN				
nroaula	fecha_hora	duracion	codmateria	confirmado
10	12/10/2024 09:00:00	03:00:00	72.34	TRUE
10	12/10/2024 08:00:00	02:00:00	72.37	FALSE

- Si luego el usuario realiza la operación *insert into aula_examen (nroaula, fecha_hora, duracion, codmateria) values (10, '12/10/2024 8:00:00,' interval '5 hour', '72.34');* entonces debemos detectar que el intervalo solicitado se superpone con el intervalo asignado anteriormente para la misma materia 72.34, por lo tanto, NO se asigna el aula, completando la columna *confirmado* con FALSE, quedando la tabla *aula_examen* de la de la siguiente manera:

AULA_EXAMEN				
nroaula	fecha_hora	duracion	codmateria	confirmado
10	12/10/2024 09:00:00	03:00:00	72.34	TRUE
10	12/10/2024 08:00:00	02:00:00	72.37	FALSE
10	12/10/2024 08:00:00	05:00:00	72.34	FALSE

- Si luego el usuario realiza la operación *insert into aula_examen (nroaula, fecha_hora, duracion, codmateria) values (8, '12/10/2024 8:00:00', interval '5 hour', '72.34');* entonces como es una nueva aula para otra comisión de la misma materia se asigna el aula, completando la columna *confirmado* con TRUE, quedando la tabla *aula_examen* de la siguiente manera:

AULA_EXAMEN				
nroaula	fecha_hora	duracion	codmateria	confirmado
10	12/10/2024 09:00:00	03:00:00	72.34	TRUE
10	12/10/2024 08:00:00	02:00:00	72.37	FALSE
10	12/10/2024 08:00:00	05:00:00	72.34	FALSE
8	12/10/2024 08:00:00	05:00:00	72.34	TRUE

- Si luego el usuario realiza la operación *insert into aula_examen (nroaula, fecha_hora, duracion, codmateria) values (8, '11/11/2024 12:30:00', interval '1.5 hour', '72.34');* entonces como es una nueva fecha se asigna el aula, completando la columna *confirmado* con TRUE, quedando la tabla *aula_examen* de la siguiente manera:

AULA_EXAMEN				
nroaula	fecha_hora	duracion	codmateria	confirmado
10	12/10/2024 09:00:00	03:00:00	72.34	TRUE
10	12/10/2024 08:00:00	02:00:00	72.37	FALSE
10	12/10/2024 08:00:00	05:00:00	72.34	FALSE
8	12/10/2024 08:00:00	05:00:00	72.34	TRUE
8	11/11/2024 12:30:00	01:30:00	72.34	TRUE

- Si luego el usuario realiza por ejemplo la operación *insert into aula_examen (nroaula, fecha_hora, duracion, codmateria) values (10, '11/11/2024 13:00:00', interval '3 hour', '72.37');* como es una nueva aula se asigna el aula, completando la columna *confirmado* con TRUE, quedando la tabla *aula_examen* de la siguiente manera:

AULA_EXAMEN				
nroaula	fecha_hora	duracion	codmateria	confirmado
10	12/10/2024 09:00:00	03:00:00	72.34	TRUE
10	12/10/2024 08:00:00	02:00:00	72.37	FALSE
10	12/10/2024 08:00:00	05:00:00	72.34	FALSE
8	12/10/2024 08:00:00	05:00:00	72.34	TRUE
8	11/11/2024 12:30:00	01:30:00	72.34	TRUE
10	11/11/2024 13:00:00	03:00:00	72.37	TRUE

También es importante el manejo de errores para evitar el ingreso de datos incorrectos. Puede lanzar mensajes, excepciones, lo que crea conveniente, siempre que se entienda la razón de la misma. Por ejemplo, el caso que si el usuario realiza un pedido con un intervalo que abarca más de un día.

c) Importación de los datos

Utilizando el comando COPY de PostgreSQL, se deben importar TODOS los datos del archivo csv en la tabla creada en el punto a). El archivo csv provisto por la cátedra NO puede ser modificado.

d) Análisis de casos exitosos y no exitosos

El administrador de la base de datos realiza periódicamente un análisis de los casos exitosos y no exitosos para mejorar el sistema de asignación de aulas.

Se pide crear la función *analisis_asignaciones(dia_hora)* que recibe como parámetro una fecha y hora, que se tomará como fecha base a partir de la cual se consideraran los registros a tener en cuenta, la cual genere un reporte mostrando lo siguiente:

- Para aquellas asignaciones que fueron confirmadas cada materia, el promedio del tiempo que solicitó para exámenes por materia y fecha, ordenado de mayor promedio a menor en cada materia
- Para aquellas asignaciones que no fueron confirmadas, el intervalo que no pudieron asignarse con su fecha, agrupado por aula y ordenado por fecha y hora

El reporte tendrá las siguientes características:

- I. Título del reporte: "ANALISIS DE ASIGNACIONES"
- II. Encabezado de columnas:
"Variable Fecha Horas Nro Linea"
- III. Por cada Variable (código de materia o número de aula) tiene que aparecer un renglón en el reporte, con la información pedida para cada caso, adicionando el número de línea

En caso de que no existieran datos para fechas mayores a la fecha y hora ingresada, no se debe mostrar nada (ni siquiera el encabezado del reporte).

La función debe manejar posibles errores.

Por ejemplo:

- Si invocamos **analisis_asignaciones('2024-10-01 00:00:00')** se obtiene un informe que comienza de la siguiente manera (considerando los datos de ejemplo explicados en el **b)**):

```
INFO: -----
INFO: -----ANALISIS DE ASIGNACIONES-----
INFO: -----
INFO: Variable-----Fecha-----Horas-----Nro Linea--
INFO: -----
INFO: Materia: 72.34    2024-10-12    04:00:00          1
INFO: Materia: 72.34    2024-11-11    01:30:00          2
INFO: Materia: 72.37    2024-11-11    03:00:00          1
INFO: Aula: 10          2024-10-12    08:00:00 a 10:00:00  1
INFO: Aula: 10          2024-10-12    08:00:00 a 13:00:00  2
INFO: -----
```

- Si invocamos **analisis_asignaciones('2024-11-01 00:00:00')** se obtiene un informe que comienza de la siguiente manera (considerando los datos de ejemplo explicados en el **b)**):

```

INFO: -----
INFO: -----ANALISIS DE ASIGNACIONES-----
INFO: -----
INFO: Variable-----Fecha-----Horas-----Nro Linea--
INFO: -----
INFO: Materia: 72.34    2024-11-11    01:30:00          1
INFO: Materia: 72.37    2024-11-11    03:00:00          1
INFO: -----

```

- Si invocamos **analisis_asignaciones('2024-12-01 00:00:00')** no se obtiene nada

4. Entregables

Los alumnos deberán entregar los siguientes documentos:

- El script sql **funciones.sql** con el código necesario para crear las tablas, 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 del trigger, 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 el conjunto de datos entregado 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.