

Titulación: Grado en Ingeniería Informática, Ingeniería en Sistemas de Información e InfoADE
Curso: 2024-2025. Convocatoria Ordinaria de Junio
Asignatura: Bases de Datos Avanzadas – Laboratorio

Practica 3: Seguridad, Usuarios y Transacciones.

ALUMNO 1:

Nombre y Apellidos: _____

DNI: _____

ALUMNO 2:

Nombre y Apellidos: _____

DNI: _____

Fecha: _____

Profesor Responsable: _____

Mediante la entrega de este fichero los alumnos aseguran que cumplen con la normativa de autoría de trabajos de la Universidad de Alcalá, y declaran éste como un trabajo original y propio.

En caso de ser detectada copia, se puntuará **TODA** la asignatura como Suspenseo – Cero.

Plazos

Tarea en laboratorio: Semana 21 de abril, Semana 28 de abril, Semana 5 de mayo y semana 12 de mayo.

Entrega de práctica: **Día 16 de mayo.** Aula Virtual

Documento a entregar: Este mismo fichero con las respuestas a las cuestiones planteadas, con el código SQL utilizado en cada uno de los apartados. **Así mismo se debe de entregar el fichero de configuración postgresql.conf usado en la práctica, así como los ficheros de log de postgres (obligatorios).** Se entregará en un ZIP comprimido: **DNI 'sdelosAlumnos_PECL3.zip**

AMBOS ALUMNOS DEBEN ENTREGAR EL FICHERO EN LA PLATAFORMA.

Introducción

El contenido de esta práctica versa sobre el manejo de las transacciones en sistemas de bases de datos, así como el control de la concurrencia y la recuperación de la base de datos frente a una caída del sistema. Las transacciones se definen como una unidad lógica de procesamiento compuesta por una serie de operaciones simples que se ejecutan como una sola operación. Entre las etiquetas BEGIN y COMMIT del lenguaje SQL se insertan las operaciones simples a realizar en una transacción. La sentencia ROLLBACK sirve para deshacer todos los cambios involucrados en una transacción y devolver a la base de datos al estado consistente en el que estaba antes de procesar la transacción. También se verá el registro diario o registro histórico del sistema de la base de datos (en PostgreSQL se denomina WAL: Write Ahead Login) donde se reflejan todas las operaciones sobre la base de datos y que sirve para recuperar ésta a un estado consistente si se produjera un error lógico o de hardware. La versión de postgres a utilizar deberá ser la versión 17 preferentemente.

Actividades y Cuestiones

En esta parte la base de datos **TELPARK** deberá de ser nueva y no contener datos. Además, consta de 5 actividades:

- Conceptos generales.
- Manejo de transacciones.
- Concurrencia.
- Registro histórico.
- Backup y Recuperación

Cuestión 0: Configurar el fichero de Error Reporting and Logging de PostgreSQL para que aparezcan recogidas las sentencias SQL DDL (Lenguaje de Definición de Datos) + DML (Lenguaje de Manipulación de Datos) generadas en dicho fichero. No se pide activar todas las sentencias. No activar la duración de la consulta. También se debe de configurar el log para que en el comienzo de la línea de registro de la información del log ("line prefix") aparezca el DNI de los alumnos que realizan la práctica (ambos), el nombre del host con su puerto, y la **fecha y hora** de la operación que se ha realizado.

Cuestión 1: ¿qué es el WAL en postgres? Si no está activo, activarlo. Determinar cuál es el directorio y el archivo donde se guarda. ¿Cuál es su tamaño? Al abrir el archivo con un editor de textos, ¿se puede deducir algo de lo que guarda el archivo?

Cuestión 2: Abrir una transacción que realice una operación de inserción de un cliente y un vehículo que pertenece a ese cliente en la base de datos **TELPARK**. Cerrar la transacción con éxito. Entonces abrir el archivo de diario ¿Se encuentra reflejada la operación en el archivo del sistema? ¿Por qué?

Cuestión 3: Aplicar el comando `pg_waldump.exe` al último fichero de WAL que se haya generado. Obtener las estadísticas de ese fichero y comentar qué se está viendo.

Cuestión 4: Determinar el identificador de la transacción que realizó la operación de inserción de la cuestión 2. Aplicar el comando anterior al último fichero de WAL que se ha generado y mostrar los registros que se han creado para esa transacción. ¿Qué se puede ver? Interpretar los resultados obtenidos.

Cuestión 5: Se va a crear un backup del clúster de postgres. Este backup será utilizado más adelante para recuperar el sistema frente a una caída del sistema. Realizar solamente el backup mediante el procedimiento descrito en el apartado 25.3 del manual (versión 17 es *"Continuous Archiving and point-in-time recovery (PITR)"*). Comentar el procedimiento realizado.

Cuestión 6: ¿Qué herramientas disponibles tiene PostgreSQL para controlar la actividad de la base de datos en cuanto a la concurrencia y transacciones? ¿Qué información es capaz de mostrar? ¿Dónde se guarda dicha información? ¿Cómo se puede mostrar?

Cuestión 7: Crear tres usuarios en la base de datos que puedan acceder a la base de datos **TELPARK** identificados como usuario1, usuario2 y usuario3 que tengan permisos de lectura/escritura a la base de datos **TELPARK**, pero que no puedan modificar su estructura. Describir el proceso seguido.

Cuestión 8: Abrir una transacción T1 en el usuario1 que realice las siguientes operaciones sobre la base de datos **TELPARK**. NO termine la transacción. Simplemente:

- Inserte un nuevo cliente de la Comunidad de Madrid con teléfono 918856931.
- Inserte un vehículo a su nombre de matrícula 1234MMG, de marca Audi, modelo A3 y color negro.

Realizar cualquier consulta SQL que muestre los datos anteriores insertados para ver que todo está correcto. Consultar la información sobre lo que se encuentra relacionado con la transacción en la actividad del sistema. ¿Qué conclusiones se pueden extraer?

Cuestión 9: Establecer una **nueva conexión** a la base de datos con el usuario2 (abrir otra sesión diferente a la abierta actualmente que pertenezca al usuario2) y realizar la misma consulta. ¿Se nota algún cambio? En caso afirmativo, ¿a qué puede ser debido el diferente funcionamiento en la base de datos para ambas consultas? ¿Qué información de actividad hay registrada en la base de datos en este momento relacionadas con las transacciones abiertas?

Cuestión 10: ¿Se encuentran los nuevos datos físicamente en las tablas de la base de datos? Entonces, ¿de dónde se obtienen los datos de la cuestión 8 y/o de la 9?

Cuestión 11: Finalizar con éxito la transacción T1 y realizar la consulta de la cuestión 8 y 9 sobre ambos usuarios conectados. ¿Qué es lo que se obtiene ahora? ¿Por qué?

Cuestión 12: Sin ninguna transacción en curso, abrir una transacción en un usuario cualquiera y realizar las siguientes operaciones:

- Insertar un nuevo cliente de la Comunidad de Cantabria.
- Insertar un nuevo vehículo con matrícula 6789HPT de ese cliente.
- Cambiar la matrícula anterior por 1234MMG.
- Insertar un nuevo cliente de la Comunidad de Madrid
- Insertar un nuevo vehículo del cliente anterior con matrícula 4567HPG
- Cerrar la transacción.

¿Cuál es el estado final de la base de datos? ¿Por qué?

Cuestión 13: Repetir la cuestión 8 con otra cliente y otro vehículo que sean distintos. Realizar la misma consulta de la cuestión 9, pero ahora terminar la transacción con un ROLLBACK y repetir la consulta con los mismos dos usuarios. ¿Cuál es el resultado? ¿Por qué?

Cuestión 14: Crear las siguientes tres tablas:

- Tabla1 con campo X de tipo real que sea PK.
- Tabla2 con campo Y de tipo real que sea PK.
- Tabla3 con campo Z de tipo real que sea PK.

Insertar una tupla en cada tabla con valor X=30, B=40 y C=50 respectivamente.

Cuestión 15: Se suministra la tabla siguiente donde se muestra la planificación que hay que seguir por tres transacciones concurrentes asociadas cada una a un usuario diferente que van a acceder en el orden marcado a las tablas anteriores. Se debe de abrir tres consolas SQL con cada uno de los usuarios (usuario1, usuario2 y usuario3) y seguir el orden de cada instrucción teniendo en cuenta que un READ es una sentencia SELECT y un WRITE es una sentencia UPDATE. Rellenar la tabla con los valores que ven cada uno de los usuarios en su ejecución de las instrucciones de cada transacción. Antes de la primera instrucción de cada transacción se debe ejecutar BEGIN;

| Secuencia | T1 Usuario1 | T2 Usuario2 | T3 Usuario3 | Valores Usuario1 | Valores Usuario2 | Valores Usuario3 | Sentencia SQL |
|-----------|----------------|----------------|----------------|---------------------|---------------------|---------------------|---------------|
| 1 | | READ(Y) | | | | | |
| 2 | | READ(Z) | | | | | |
| 3 | | Y=Y/Z | | | | | |
| 4 | | | READ(X) | | | | |
| 5 | | | X=X+20 | | | | |
| 6 | | | WRITE(X) | | | | |
| 7 | | WRITE(Y) | | | | | |
| 8 | | | READ(Y) | | | | |
| 9 | | | Y=20*Y+X | | | | |
| 10 | | | WRITE(Y) | | | | |
| 11 | READ(Z) | | | | | | |
| 12 | Z=Z-100 | | | | | | |
| 13 | READ(Y) | | | | | | |
| 14 | Y=Z/Y | | | | | | |
| 15 | WRITE(Z) | | | | | | |
| 16 | | Z=Y+100 | | | | | |
| 17 | | WRITE(Z) | | | | | |
| 18 | READ(X) | | | | | | |
| 19 | X=Y*Z | | | | | | |
| 20 | WRITE(X) | | | | | | |
| 21 | COMMIT | | | | | | |
| 22 | | COMMIT | | | | | |
| 23 | | | COMMIT | | | | |

Rellenar también la siguiente tabla con el estado del servidor y comentarios de lo que está ocurriendo en cada secuencia de ejecución de las instrucciones de cada transacción en el servidor postgres.

| Estado del servidor | Comentarios |
|---------------------|-------------|
| | |

Cuestión 16: ¿Cuál es el estado final de los datos X, Y y Z en la base de datos cuando han finalizado las tres transacciones? ¿Por qué?

Cuestión 17: Suponer que se produce una pérdida del cluster de datos y se procede a restaurar la instancia de la base de datos del punto 5. Realizar solamente la restauración (recovery) mediante el procedimiento descrito en el apartado 25.3.5 del manual (versión, 17) *"Continuous Archiving and point-in-time recovery (PITR)*". Indicar el proceso seguido. ¿Cuál es el estado final de la base de datos? ¿Por qué?

Cuestión 18: A la vista de los resultados obtenidos en las cuestiones anteriores, ¿Qué tipo de sistema de recuperación tiene implementado postgresQL? ¿Qué protocolo de gestión de la concurrencia tiene implementado? ¿Por qué? ¿Genera siempre planificaciones secuenciables? ¿Genera siempre planificaciones recuperables? ¿Tiene rollbacks en cascada? Justificar las respuestas.

Bibliografía

- Capítulo 9: System Information Functions and Operators, System Administration Functions.
- Capítulo 13: Concurrency Control.
- Capítulo 19: Server Configuration.
- Capítulo 25: Backup and Restore.
- Capítulo 27: Monitoring Database Activity.
- Capítulo 28: Reliability and the Write-Ahead log.