

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 2: Carga Masiva de Datos, Procesamiento y Optimización de Consultas

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 calificará la asignatura como Suspenso – Cero.

Plazos

Tarea en Laboratorio: Semana 10 de marzo, Semana 17 de marzo, Semana 24 de marzo, semana 31 de marzo y 7 de abril.

Entrega de práctica: Día 20 de abril, domingo a las 23:59. Aula Virtual

Documento a entregar: Este mismo fichero con las respuestas a las cuestiones planteadas, el programa que genera los datos de carga de la base de datos, los ficheros de log de PostgreSQL de esta práctica. No se piden los ficheros de los datos en bruto de la base de datos. Se entregará en un ZIP comprimido llamado: **DNI'sdelosAlumnos_PECL2.zip**

AMBOS ALUMNOS DEBEN ENTREGAR EL FICHERO EN LA PLATAFORMA.

Introducción

El contenido de esta práctica versa sobre la monitorización de la base de datos, manipulación de datos, técnicas para una correcta gestión de los mismos, así como tareas de mantenimiento relacionadas con el acceso y gestión de los datos. También se trata el tema de procesamiento y optimización de consultas realizadas por PostgreSQL (17). Se analizará PostgreSQL en el proceso de carga masiva y optimización de consultas.

En general, la monitorización de la base de datos es de vital importancia para la correcta implantación de una base de datos, y se suele utilizar en distintos entornos:

- Depuración de aplicaciones: cuando se desarrollan aplicaciones empresariales no se suele acceder a la base de datos a bajo nivel, sino que se utilizan librerías de alto nivel y mapeadores ORM (Hibernate, Spring Data, MyBatis...) que se encargan de crear y ejecutar consultas para que el programador pueda realizar su trabajo más rápido. El problema en estos entornos está en que se pierde el control de qué están haciendo las librerías en la base de datos, cuántas consultas ejecutan, y con qué parámetros, por lo que la monitorización en estos entornos es vital para saber qué consultas se están realizando y poder optimizar la base de datos y los programas en función de los resultados obtenidos.
- Entornos de prueba y test de rendimiento: cuando una base de datos ha sido diseñada y se le cargan datos de prueba, una de las primeras tareas a realizar es probar que todos los datos que almacenan son consistentes y que las estructuras de datos dan un rendimiento adecuado a la carga esperada. Para ello se desarrollan programas que simulen la ejecución de aquellas consultas que se consideren de interés para evaluar el tiempo que le lleva a la base de datos devolver los resultados, de cara a buscar optimizaciones, tanto en la estructura de la base de datos como en las propias consultas a realizar.
- Monitorización pasiva/activa en producción: una vez la base de datos ha superado las pruebas y entra en producción, el principal trabajo del administrador de base de datos es mantener la monitorización pasiva de la base de datos. Mediante esta monitorización el administrador verifica que los parámetros de operación de la base de datos se mantienen dentro de lo esperado (pasivo), y en caso de que algún parámetro salga de estos parámetros ejecuta acciones correctoras (reactivo). Así mismo, el administrador puede evaluar nuevas maneras de acceso para mejorar aquellos procesos y tiempos de ejecución que, pese a estar dentro de los parámetros, muestren una desviación tal que puedan suponer un problema en el futuro (activo).

Para la realización de esta práctica será necesario generar una muestra de datos de cierta índole en cuanto a su volumen de datos. Para ello se generarán, dependiendo del modelo de datos suministrado, para una base de datos denominada **TELPARK**. Básicamente la base de datos guarda las reservas de los clientes que se realizan en el parking junto con su método de pago, los vehículos y las incidencias que se pueden producir. Se suministra el modelo relacional construido en **pgmodeler**.



Los datos referidos al año 2024 que hay que generar deben de ser los siguientes:

- Existen 3.000.000 clientes registrados en la APP de Telpark. Las provincias deben de ser asignada aleatoriamente de entre todas las provincias españolas.
- Existen 5.000.000 vehículos dados de alta. Se deben asignar los vehículos a los propietarios de manera aleatoria. Existen 500 marcas de automóviles diferentes donde cada una tiene entre 5 y 20 modelos, con colores entre 5 y 15 (el negro debe ser uno de ellos). La asignación de los valores de cada campo se debe de hacer de manera aleatoria.
- Existen 200.000 plazas de parking de la empresa Telpark. La planta debe ser un número aleatorio entre -10 y 0, y la sección un valor aleatorio comprendido entre las letras A-F.
- Se han producido 40.000.000 de reservas de plazas de parking por la APP. La fecha de inicio y final debe tener el día, mes, y hora generados de manera aleatorio, con una duración de estancia comprendida entre 1 día y 10 días.
- Cada reserva se ha pagado por un método de pago que puede ser efectivo, tarjeta de crédito, tarjeta de débito, PayPal, Bizzum o transferencia. La cantidad de la reserva debe ser el número de horas multiplicado por 3€. La fecha de pago debe ser anterior al comienzo de la estancia en el Parking.
- Se han generado 4.000.000 de incidencias generadas aleatoriamente de entre todas las reservas, donde el estado puede ser nueva, abierta, en proceso, resuelta y cerrada. Todas las asignaciones se hacen de manera aleatoria.

Actividades y Cuestiones

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. **Se ha de configurar también el servidor para que no use el procesamiento paralelo de consultas.**

Cuestión 1: ¿Tiene el servidor postgres un recolector de estadísticas sobre los datos de cada campo de las tablas? Si es así, ¿Qué tipos de estadísticas se recolectan y dónde se guardan?

Cuestión 2: Crear una nueva base de datos llamada **investigar** y que tenga las siguientes tablas con los siguientes campos y características:

- investigadores(codigo_investigador tipo numeric PRIMARY KEY, nombre tipo text, apellidos tipo text, salario tipo numeric)
- contratos(codigo_contrato tipo numeric PRIMARY KEY, nombre tipo text, entidad tipo text, coste tipo numeric)
- investigadores_contratos(codigo_investigador tipo numeric que sea FOREIGN KEY del campo codigo_investigador de la tabla investigadores con restricciones de tipo RESTRICT en sus operaciones, codigo_contrato tipo numeric que sea FOREIGN KEY del campo codigo_contrato de la tabla contratos con restricciones de tipo RESTRICT en sus operaciones, horas de tipo numeric. La PRIMARY KEY debe ser compuesta de codigo_investigador y codigo_contrato.

Se pide:

- Indicar el proceso seguido para generar esta base de datos.
- Cargar la información del fichero datos_empleados.csv, datos_proyectos.csv y datos_trabaja_proyectos.csv en dichas tablas de tal manera que sea lo más eficiente posible.
- Indicar los tiempos de carga.

Cuestión 3: Mostrar las estadísticas obtenidas en este momento para cada tabla. ¿Qué se almacena? ¿Son correctas? Si no son correctas, ¿cómo se pueden actualizar?

Cuestión 4: Aplicar el comando EXPLAIN a una consulta que obtenga la información de los investigadores con salario de menos de 24.000 euros. ¿Son correctos los resultados del comando EXPLAIN? ¿Por qué? Comparar con lo que se obtendría con lo visto en teoría obteniendo las estadísticas de las tablas con postgres.

Cuestión 5: Aplicar el comando EXPLAIN a una consulta que obtenga el número de contratos en los cuales el investigador trabaja 8 horas y tiene un salario de 20.000 euros. ¿Son correctos los resultados del comando EXPLAIN? ¿Por qué? Comparar con lo que se obtendría con lo visto en teoría obteniendo las estadísticas de las tablas con postgres.

Cuestión 6: Aplicar el comando EXPLAIN a una consulta que obtenga la información del nombre de los contratos y entidad que tienen un coste mayor de 10000 y menor que 15000, y tienen investigadores de salario mayor de 16000 euros y trabajan 5 horas en ellos. ¿Son correctos los resultados del comando EXPLAIN? ¿Por qué? Comparar con lo que se obtendría con lo visto en teoría obteniendo las estadísticas de las tablas con postgres.

Cuestión 7: Generar los datos solicitados al comienzo de la práctica para la base de datos **TELPARK** creando un programa para tal fin que deberá de estar en un único fichero y comentado. Pegar el código del fichero en el cuadro de texto que se adjunta a continuación.

Cuestión 8: Realizar la carga masiva de los datos generados en la cuestión 7 en la base de datos **TELPARK**. Indicar el proceso seguido y el orden de carga de las tablas, explicando el porqué de dicho orden; **y asegurando la consistencia e integridad de los datos cargados**. Comparar los tiempos en las tablas implicadas y explicar a qué es debida la diferencia.

Tabla	Tiempo (seg)

A partir de este momento en adelante, se deben de realizar las siguientes cuestiones con la base de datos que tiene la integridad referencial activada. Es obligatorio y queda prohibido cambiar la integridad referencial de la base de datos.

Cuestión 9: Realizar una consulta SQL que muestre el “porcentaje de clientes que cumplen que son de la comunidad de Andalucía realizando una reserva durante los meses de verano, pagando una cantidad de más de 150 €, estacionando sus vehículos por debajo de la planta -4, y que no hayan tenido incidencias cerradas con sus vehículos de color negro”.

Obtener el plan de ejecución con el resultado del comando EXPLAIN. Explicar la información obtenida en el plan de ejecución de postgresQL. Comparar el árbol obtenido por nosotros al traducir la consulta original al álgebra relacional y el que obtiene postgresQL. Comentar las posibles diferencias entre ambos árboles.

Consulta SQL creada, tiempo de ejecución y resultado obtenido:

Resultado comando EXPLAIN

Comentarios y explicaciones.

Cuestión 10: Usando PostgreSQL, y a raíz de los resultados de la cuestión anterior, ¿qué acciones realizaría para mejorar el rendimiento de la consulta sin cambiar el código SQL y por qué? Obtener la información pedida de la cuestión 9 y explicar los resultados. Obtener el plan de ejecución con el resultado del comando EXPLAIN. Comentar los resultados obtenidos y comparar con la cuestión anterior.

Consulta SQL creada, tiempo de ejecución y resultado obtenido:

Resultado comando EXPLAIN

Comentarios y explicaciones.

Cuestión 11: Usando PostgreSQL, borre el 30% de los clientes almacenados de manera aleatoria y todos sus datos relacionados de la manera más eficiente posible, ¿cuál ha sido el proceso seguido? ¿Y el tiempo empleado en el borrado? **Prohibido modificar la Integridad Referencial.**

Cuestión 12: Ejecute la consulta que pide los datos de la cuestión 9 de nuevo. Obtener el plan de ejecución con el resultado del comando EXPLAIN. Comparar con los resultados anteriores.

Consulta SQL creada, tiempo de ejecución y resultado obtenido:

Resultado comando EXPLAIN

Comentarios y explicaciones.

Cuestión 13: ¿Qué optimización/mejoras de la BD propondría para mejorar los resultados de dicho plan sin modificar el código SQL de la consulta? ¿Por qué? **No implementarlo ahora.**

Cuestión 14: Usando PostgreSQL, lleve a cabo las operaciones propuestas en la cuestión anterior y ejecute el plan de ejecución de la consulta que pide los datos de la cuestión 9. Obtener el plan de ejecución con el resultado del comando EXPLAIN. Compare los resultados del plan de ejecución con los de los apartados anteriores. Coméntelos.

Consulta SQL creada, tiempo de ejecución y resultado obtenido:

Resultado comando EXPLAIN

Comentarios y explicaciones.

Cuestión 15: A partir de lo visto y recopilado en toda la práctica. Describir y comentar cómo es el proceso de procesamiento y optimización que realiza PostgreSQL en las consultas del usuario.

Bibliografía

PostgreSQL (17)

- Capítulo 14: Performance Tips.
- Capítulo 19: Server Configuration.
- Capítulo 15: Parallel Query.
- Capítulo 24: Routine Database Maintenance Tasks.
- Capítulo 50: Overview of PostgreSQL Internals.
- Capítulo 68: How the Planner Uses Statistics.
- <https://pgtune.leopard.in.ua/>
- <https://explain.dalibo.com/>