

Titulación: Grado en Ingeniería Informática e Ingeniería en Sistemas de Información
Curso: 2022-2023. 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 6 de marzo, Semana 13 de marzo, Semana 20 de marzo y semana 27 de marzo.

Entrega de práctica: Semana 11 de abril (martes). 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 (15.1). 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 **Películas**. Básicamente la base de datos guarda personas y películas. Las personas pueden actuar en películas y además una de ellas puede dirigirla. Las películas tienen una serie de géneros y generan visualizaciones y críticas por parte de las personas dadas de alta. Se suministra el modelo relacional construido en **pgmodeler** donde dentro de cada tabla se comenta lo que se guarda, así como la descripción de cada campo.

Los datos referidos al año 2022 que hay que generar deben de ser los siguientes:

- Existen 20 géneros de contenidos diferentes, y el número de géneros que tiene asociada una película es un valor aleatorio que va entre 1 y 6.
- La tabla películas tiene 400.000 entradas. Las restricciones de algunos campos de la tabla son:
 - o El año debe ser un valor aleatorio entre 1950 y el 2023.
 - o La duración debe ser un valor aleatorio entre 90 y 240 minutos.
 - o El idioma debe ser un valor aleatorio entre 1 y 30 donde dos de ellos deben ser EN (english) y ES (español).
 - o La calificación será entre 0 y 10 estrellas.
- Cada película tiene un número de críticas que se eligen aleatoriamente entre 1 y 100. Todas las críticas son del año 2022, y el día y el mes se deben de generar aleatoriamente. La puntuación debe generarse aleatoriamente entre 0 y 10.
- Cada película tiene un director que se asigna aleatoriamente entre las personas que hay dadas de alta. El número de actores que tiene una película varía entre 10 y 30, debiendo elegirse aleatoriamente entre el número de personas que hay.
- Hay un total de 400.000 personas de 100 nacionalidades distintas, donde España debe ser una de ellas. Las nacionalidades se elegirán aleatoriamente entre todas las que hay. El año de nacimiento será un valor aleatorio desde 1950 hasta 2020, valores incluidos.
- Cada película tiene un número de visualizaciones comprendidas entre 1 y 100, todas realizadas en el año 2022, debiendo generarse el día y el mes 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 el contenido de las tablas de datos? Si es así, ¿Qué tipos de estadísticas se recolectan y donde se guardan?

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

- clientes(dni tipo numeric PRIMARY KEY, nombre tipo text, direccion tipo text, telefono tipo numeric)
- automoviles(bastidor tipo numeric PRIMARY KEY, proveedor tipo text, modelo tipo text, precio tipo numeric)
- compras(dni_cli tipo numeric que sea FOREIGN KEY del campo dni de la tabla clientes con restricciones de tipo RESTRICT en sus operaciones, bastidor_auto tipo numeric que sea FOREIGN KEY del campo bastidor de la tabla automóviles con restricciones de tipo RESTRICT en sus operaciones, fecha de tipo date, coste tipo numeric. La PRIMARY KEY debe ser bastidor_auto).

Se pide:

- Indicar el proceso seguido para generar esta base de datos.
- Cargar la información del fichero datos_clientes.csv, datos_autmoviles.csv y datos_compras.csv en dichas tablas de tal manera que sea lo más eficiente posible, asegurando la integridad y consistencia de la base de datos.

- 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 el número de automóviles del proveedor500 que no se han vendido. ¿Son correctos los resultados del comando EXPLAIN? ¿Por qué? Comparar con lo que se obtendría con el procedimiento visto en teoría obteniendo las estadísticas de postgres.

Cuestión 5: Aplicar el comando EXPLAIN a una consulta que obtenga el modelo de los automóviles con un precio mayor de 50000 que se vendieron en el mes de mayo. ¿Son correctos los resultados del comando EXPLAIN? ¿Por qué? Comparar con lo que se obtendría con el procedimiento visto en teoría obteniendo las estadísticas de postgres.

Cuestión 6: Aplicar el comando EXPLAIN a una consulta que obtenga el coste de las ventas de los automóviles vendidos entre el mes de febrero y agosto (incluidos), que sean del proveedor300 o del proveedor500, y además el teléfono de los clientes esté incluido entre los números 7000000 y 9000000 (incluidos). ¿Son correctos los resultados del comando EXPLAIN? ¿Por qué? Comparar con lo que se obtendría con lo visto en teoría obteniendo las estadísticas de postgres.

Cuestión 7: Generar los datos solicitados al comienzo de la práctica para la base de datos **Películas** 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 **Películas**. 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. ¿Existe diferencia entre los tiempos que ha obtenido y los que aparecen en el LOG de operaciones de postgresQL? ¿Por qué?

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 las películas que tienen más de 2 géneros diferentes cuyo idioma no sea inglés (EN) o español (ES) con una duración de más de 100 minutos, teniendo por lo menos 10 críticas con una puntuación de al menos 5 puntos, y habiendo tenido visualizaciones entre el mes de mayo y agosto por personas que no sean españolas, no siendo estas películas dirigidas por directores españoles”.

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 modificaciones realizaría para mejorar el rendimiento de la misma 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. 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 las películas almacenadas 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 de la consulta? ¿Por qué?

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 (15.1)

- Capítulo 14: Performance Tips.
- Capítulo 20: Server Configuration.
- Capítulo 15: Parallel Query.
- Capítulo 25: Routine Database Maintenance Tasks.
- Capítulo 52: Overview of PostgreSQL Internals.

- Capítulo 75: How the Planner Uses Statistics.
- <https://pgtune.leopard.in.ua/>
- <https://explain.dalibo.com/>