BASES DE DATOS AVANZADAS

Enunciado de la Segunda Práctica – Técnicas de aceleración de consultas

1. Objetivos

- Expresar consultas sql de forma eficiente.
- Optimizar consultas sql mediante el uso de índices y el estudio de los planes de ejecución.
- Estudiar el impacto de los índices en las actualizaciones.
- Aplicar técnicas de desnormalización de bases de datos relacionales.

2. Enunciado

AdventureWorks es una base de datos publicada por Microsoft, que corresponde a la base de datos operacional de la empresa ficticia Adventure Works Cycles, dedicada a la fabricación y venta de bicicletas y accesorios en los mercados de Norteamérica, Europa y Asia, tanto online como en tiendas físicas a través de distribuidores. A partir de la versión AdventureWorks2019 se ha confeccionado la base de datos para realizar esta práctica, compuesta por unas pocas tablas en la que se detallan los clientes que realizan compras, los territorios de ventas, los productos a la venta, así como los pedidos de los clientes y el detalle de dichos pedidos. Los datos de ventas incluidos en la base de datos van desde febrero de 2019 hasta marzo de 2023.

Se van a considerar las siguientes tablas:

- **Producto**, que contiene los datos de los productos que vende la empresa
- **Categoría,** que contiene los datos de la categoría de los productos
- **Subcategoria,** que contiene los datos de la subcategoría de los productos
- **Cliente**, que contiene datos de los clientes que realizan los pedidos.
- **TerritorioVentas**, que contiene los datos de la clasificación de las zonas de ventas
- **Pedido**, que contiene la cabecera de cada pedido realizado por los clientes, con los datos más importantes de cada uno de ellos, incluyendo, entre otros, el código del pedido (*PedidoID*), el cliente que lo realiza (*ClienteID*), los impuestos del pedido (*Impuestos*) y los gastos de envío (*GastosEnvio*).
- **DetallePedido**, que contiene el detalle de los pedidos realizados por los clientes. Cada fila se corresponde con un producto del pedido. Incluye, entre otros, el código del pedido (PedidoID), el código del producto (*ProductoID*), el número de unidades (*Cantidad*) y el precio unitario (*PrecioUnitario*), así como el descuento en caso de tenerlo (*DescuentoUnitario*), que aparece expresado en tanto por 1 (un descuento del 10% aparece como 0.10).

3. Desarrollo de la práctica

La práctica se trabajará en las clases de laboratorio durante cuatro sesiones. En cada semana el alumno trabajará en los contenidos de la práctica tanto durante la clase presencial como fuera de ella. No se entregará ningún tipo de memoria ni fichero relativo a la práctica. Eso sí, se proporciona un guion de desarrollo de la práctica en el que el alumno pueda ir anotando los aspectos de resolución de la práctica.

Los conocimientos cubiertos en la práctica serán evaluados mediante una prueba escrita de forma individual.

Se realizarán los siguientes aparatos:

1. Crear la base de datos de la práctica mediante la ejecución del script "***dumpBDAprac2.sql***" disponible junto con este enunciado en el Moodle de la asignatura.

2. Optimización de consultas.
    a. Crear una consulta SQL que permita obtener el nombre de los productos que figuren en algún detalle de pedido con más de 5 unidades vendidas (atributo cantidad) sin que figuren valores duplicados en el resultado. Definir diferentes variantes de la consulta de forma que se aprecien diferencias significativas entre ellas respecto al coste de ejecución.
    b. Estudiar el plan de consulta, tomando nota de los costes del mismo.
    c. Crear las claves principales mediante el fichero script ***CrearClavesPrimarias.sql*** y nuevamente estudiar el plan de consulta, comparando costes con el punto anterior.
    d. Crear las claves foráneas mediante el fichero script ***CrearClavesForaneas.sql***, y nuevamente estudiar el plan de consulta, comparando costes con el punto anterior.
    e. Crear los índices que se estimen necesarios para mejorar la consulta.

3. Estudio de planes de consulta e índices.
    a. Eliminar los índices creados en el apartado anterior, manteniendo claves primarias y foráneas.
    b. Crear una consulta SQL que permita obtener el nombre de los productos de categoría Componente, subcategoría que contenga con la palabra Cuadro, y color Blue, que se hayan vendido en más pedidos a lo largo del tiempo que todos los productos de las subcategorías Dirección o Horquilla.
    c. Estudiar el plan de consulta, tomando nota de los costes de este.
    d. Crear los índices que se estimen necesarios para mejorar la consulta, sabiendo que para optimizar otros procesos del sistema se cuenta ya con un índice creado para el atributo nombre_categoria:

        create unique index categoria_nombrecategoria on categoriaproducto (nombre_categoria);

    e. Estudiar el plan de consulta con los nuevos índices y comparar resultados con los obtenidos en los puntos anteriores.
    f. Comprobar si la utilización de ***INDEX HINTS*** *(USE INDEX e IGNORE INDEX) sobre la sentencia e índices del apartado* ***d****) variarían el coste de ejecución calculado por el optimizador de consultas de MySql.*

4. Dada la siguiente consulta SQL:

```
select c.primer_nombre, c.apellidos
from pedidos p inner join clientes c on p.clienteID = c.clienteID
     inner join detallepedidos d ON p.PedidoID = d.PedidoID
where year(c.fecha_nacimiento) between 1980 and 1986 and genero = "F" and
   c.ingresos_anuales >= (
      select MAX(ingresos_anuales) AS `90th_percentile`
      from (
            select ingresos_anuales, PERCENT_RANK() OVER (ORDER BY ingresos_anuales)
                  AS percentile_rank
            from clientes
            where year(fecha_nacimiento) between 1970  and 1980 and genero = "F"
          ) AS ranked_sales
      where percentile_rank <= 0.9
    )
group by c.clienteID
having sum((precio_unitario - descuento_unitario) * cantidad) >= (
       select avg(total)
       from (
            select sum( (precio_unitario - descuento_unitario) * cantidad) as total
            from pedidos p inner join detallepedidos d ON p.PedidoID = d.PedidoID
            where year(fecha_venta) = 2021
            group by clienteID
     ) as totales2021
);
```

   a. Describir en lenguaje natural que realiza la consulta.
   b. Eliminar los índices creados en el apartado anterior, manteniendo claves primarias y foráneas.
   c. Crear los índices que permitan optimizar el coste de las consultas, analizando plan de consulta y coste para cada uno de los casos, justificando que solución es la mejor. Cambia la consulta como veas necesario, pero el resultado de esta no debe cambiar.

5. Estudio de índices en actualizaciones.
   a. Eliminar los índices creados en el apartado anterior manteniendo claves primarias y foráneas.
   b. Para aquellos pedidos realizados a lo largo del año 2023, incrementar el descuento unitario en un 1% para los productos vendidos con categoría Bicicleta y que ya tuvieran descuento, es decir, sea distinto de 0. Actualizar la tabla *detallepedidos* para que contemple dicho incremento tomando nota de su tiempo, plan y coste de ejecución. **Deshacer el cambio con 'rollback'.**
   c. Suponiendo la existencia de otros procesos actuando sobre la misma base de datos que requieren para optimizar cierto tipo de consultas de un índice en la tabla *detalleproductos* sobre los atributos *producotid*, *precio_unitario* y *descuento_unitario*, crear dicho índice mediante la siguiente sentencia sql:

      ```
      create index idx_detalleproductoprecio on detallepedidos (ProductoID, precio_unitario, descuento_unitario);
      ```

   d. Una vez creado el índice anterior, volver a ejecutar la sentencia del apartado 5b, comprobando tiempo, plan y coste de ejecución y compararlo razonadamente las diferencias entre ambas ejecuciones. **Volver a deshacer los cambios con 'rollback'**.

6. Desnormalización.
   a. Eliminar los índices creados en el apartado anterior, manteniendo claves primarias y foráneas.
   b. Crear una consulta que devuelva, para cada pedido el nombre del cliente, su país, la fecha del pedido, el total del pedido calculado como la suma de los detalles de cada pedido ***(precio_unitario – descuento_unitario) * cantidad.*** Tomar nota del coste y plan de ejecución.
   c. Aplicar la técnica o técnicas de desnormalización que se consideren más adecuadas para acelerar la consulta anterior, creando los scripts *sql* necesarios para modificar el esquema de la base de datos.
   d. Crear un script que actualice los datos implicados en la desnormalización.
   e. Crear los *triggers* necesarios para mantener actualizados los datos implicados en la desnormalización, dejándolos creados para los siguientes apartados.
   f. Realizar la consulta 6.b sobre la base de datos desnormalizada. Estudiar coste y plan comparándolo con el obtenido en el apartado 6b.

7. Establecer conclusiones generales de todo el presente trabajo.

4. Evaluación de la práctica.

No deberá realizarse ningún tipo de entrega para esta práctica, eso sí, los conocimientos adquiridos en su desarrollo serán objeto de examen individual. Para la realización de la práctica se utilizarán los mismos recursos software que en el desarrollo de la primera práctica.

El examen de la segunda práctica tendrá lugar el lunes día 24 de abril a las 9:00 en el bloque X.