

Universidad Técnica Particular de Loja

Práctica 04

PLANTILLA DE RESPUESTAS

Alumno: Ramiro Vivanco

Aplicación de técnicas para mejorar tiempos de respuesta en consultas SQL

PARTE 1: PROCESAMIENTO DE CONSULTAS

Ejercicio 1

```
SELECT *  
FROM empleados e  
WHERE e.apellidos like 'A%'  
OR e.apellidos like 'B%'  
OR e.apellidos like 'C%';
```

Error semántico: AND no se puede filtrar un mismo varias veces, OR sí.

Ejercicio 2

```
SELECT *  
FROM empleados  
WHERE genero = 'M'
```

Ejercicio 3

Todos los datos serán verdaderos

PARTE 2: BUENAS PRÁCTICAS SQL

Pregunta 1

NOT NUM_PAG < 80 OR NOT EDITORIAL = 'LIMUSA')

NOT COD_POSTAL = '110401' AND SUELDO < 800

Pregunta 2

Buenas Prácticas SQL

1. A NIVEL DE BASE DE DATOS
 - El motor por defecto debe de ser relacional (INNO).
 - Nombre de la base de datos en MAYÚSCULA ej 'MIBASEDEDATOS'.
2. A NIVEL DE TABLAS
 - El nombre de las tablas debe ser en mayúsculas o CamelCase.
 - El nombre de las tablas debe ser en singular.
 - El nombre de las tablas debe ser descriptivos, no importa que tan largos sean siempre y cuando sean soportados por la base de datos.

- Si la tabla es un catálogo, usar la puedes nombrar como 'CATPRODUCTO', 'CAT_CLIENTE' o en CamelCase 'CatCliente'.
- Si la tabla tiene más de 2 palabras estas se deben poner juntas o con un guión bajo, nunca se debe de usar espacios, ej 'APELLIDO_PATERNO' o en CamelCase 'APaterno'.
- Evita el uso de tablas temporales, mejor utiliza alguna estructura de datos del lenguaje de programación que estés usando.

3. A NIVEL DE CAMPOS

- Todas las tablas deben de tener un Primary Key (PK) y debe de ser el primer campo de la tabla y debe de ser único e irrepetible.
- El nombre de los campos/columnas deben ser en singular.
- PK o llave primaria de todas las tablas debe de ser un entero (INT o BIGINT), es único, autoincremental, indexado y dependiendo del uso de la tabla, será ascendente o descendente.
- El nombre del PK empieza con 'ID' en mayúsculas guión bajo nombre de la tabla con mayúsculas, ej 'ID_NOMBRE_DE_LA_TABLA', 'ID_APELLIDO_PATERNO', 'ID_CONTACTO_TELEFONO_TABLA1_TABLA2'.
- Las llaves foráneas (foreign keys) (FK), deben ser un entero (INT o BIGINT), indexadas, y con la misma nomenclatura que un PK pero agregando 'FK_' al inicio, ej 'FK_ID_NOMBRE_DE_LA_TABLA', 'FK_ID_APELLIDO_PATERNO', 'ID_CONTACTO_TELEFONO_TABLA1_TABLA2'.
- Evitar usar 'NULL' es decir campos en nulo, todos los campos deben de tener una inicialización, esta puede ser vacía y en caso de que tenga otra, debe de estar comentada en la tabla.

4. A NIVEL DE TIPOS DE DATOS PARA SQL SERVER

- Para cadenas de caracteres se utilizará nvarchar.
- Para fechas se utilizará nvarchar(10) y con el formato aaaa-mm-dd
- Para horas se utilizará nvarchar(5) y con el formato HH:mm
- Estampa de tiempo simple nvarchar(8) y con el formato HH:mm:ss
- Bigint para estampa de tiempo completa (long en java).
- Para tipo de datos booleanos es bit (La sugerencia es no usarlo a menos que el dato a guardar indique que no tendrá cambios, ej sexo).

Pregunta 3

Hints

Los hints son pistas que se dan al optimizador SQL de Oracle para que elabore el plan de ejecución de una sentencia DML (sentencias de manipulación de datos como select, insert, update, delete, etc) según nosotros le aconsejemos.

Los hints se incorporan a una sentencia DML en forma de comentario y deben ir justo detrás del comando principal. Por ejemplo, si se tratara de una sentencia SELECT el formato sería el siguiente:

```
SELECT /*+ COMANDO-HINT */ ...
```

Toda consulta SELECT se ejecuta dentro del servidor en varios pasos. Para la misma consulta, pueden existir distintos caminos para conseguir el mismo resultado, por lo que el servidor es el responsable de decidir qué camino seguir para conseguir el mejor tiempo de respuesta.

Estos son los hints que se pueden utilizar para cambiar el modo de optimización del optimizador Oracle:

- **/*+ RULE */** Fuerza a que se utilice el optimizador basado en normas (rule-based optimizer).
- **/*+ CHOOSE */** Fuerza a que se utilice el optimizador basado en costes (cost-based optimizer).
- **/*+ ALL_ROWS */** -Fuerza a que se utilice el optimizador basado en costes y optimiza el plan de ejecución de la sentencia DML para que devuelva todas las filas en el menor tiempo posible.
- **/*+ FIRST_ROWS (n) */** - También fuerza a que se utilice el optimizador basado en costes y optimiza el plan de ejecución de la sentencia DML para que devuelva las "n" primeras filas en el menor tiempo posible