

Fundamentos de Bases de Datos.

Práctica 10.

Profesor: M.I. Gerardo Avilés Rosas

gar@ciencias.unam.mx

Laboratorio: Luis Eduardo Castro Omaña

lalo_castro@ciencias.unam.mx

Laboratorio: Carlos Augusto Escalona Navarro

caen@ciencias.unam.mx

23 de abril de 2018

Se dan a conocer especificaciones de entrega para la práctica 10.

1. DML

1.1. Lenguaje de manipulación de datos.

El Lenguaje de Manipulación de Datos (Data Manipulation Language, DML) es un idioma proporcionado por los Sistemas Manejadores de Bases de Datos (SMBD) que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional.

Elementos del lenguaje de manipulación de datos:

- Insert .- Es una sentencia de inserción (INSERT) de SQL la cual se encarga de agregar uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Ejemplo:

```
INSERT INTO cursada ('alumno', 'materia') VALUES ('pepe', 'spd2');
```

- Delete .- Es una sentencia de eliminación o borrado (DELETE) de SQL la cual es la encargada de borrar uno o más registros existentes en una tabla.

Ejemplo:

```
DELETE FROM cursada WHERE materia = 'spd2'
```

- Update .- Es una sentencia de actualización o modificación (UPDATE) de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

Ejemplo:

```
UPDATE cursada SET materia = 'spd3' WHERE alumno = 'pepe';
```

- Select .- SELECT es la instrucción de SQL que devuelve un conjunto de resultados de registros de una o más tablas.

El SDBD se encarga de traducir la consulta en un "plan de consulta" que puede variar entre ejecuciones, versiones de bases de datos y software de base de datos. Esta funcionalidad se denomina "optimizador de consultas", ya que es responsable de encontrar el mejor plan de ejecución posible para la consulta, dentro de las restricciones aplicables.

La instrucción SELECT tiene muchas cláusulas opcionales:

- WHERE especifica qué filas recuperar.
- ORDER BY especifica un orden en el que devolver las filas.
- AS proporciona un alias que se puede usar para cambiar el nombre temporalmente de tablas o columnas.

2. SELECT

Use una instrucción o subconsulta SELECT para recuperar datos de una o más tablas, tablas de objetos, vistas, vistas de objetos o vistas materializadas. Si una parte o la totalidad del resultado de una declaración SELECT es equivalente a una vista materializada existente, Oracle Database puede usar la vista materializada en lugar de una o más tablas especificadas en la instrucción SELECT.

Esta sustitución se llama reescritura de consultas. Se lleva a cabo solo si la optimización de costes está habilitada y el parámetro

QUERY_REWRITE_ENABLED está establecido en TRUE.

Para determinar si ha ocurrido una escritura de consulta, use la instrucción EXPLAIN PLAN.

2.1. SINTAXIS BÁSICA :

```
SELECT */ columna1, columna2,...  
FROM nombre-tabla  
[WHERE condición]  
[GROUP BY columna1, columna2.... ]  
[HAVING condición-selección-grupos ]  
[ORDER BY columna1 [DESC], columna2 [DESC]... ]
```

3. JOIN

Oracle JOINS se utiliza para recuperar datos de varias tablas. Un JOIN se realiza cada vez que dos o más tablas se unen en una declaración SQL.

Hay 4 tipos diferentes de uniones de Oracle:

```
INNER JOIN (o a veces llamado simple join)  
LEFT OUTER JOIN (o a veces llamado LEFT JOIN)  
RIGHT OUTER JOIN (o algunas veces llamado RIGHT JOIN)  
FULL OUTER JOIN (o a veces llamado FULL JOIN)
```

3.1. INNER JOIN.

Es el tipo de JOIN más común. El INNER JOIN devuelve todas las filas de varias tablas donde se cumple la condición de unión.

```
SELECT columns  
FROM table1  
INNER JOIN table2  
ON table1.column = table2.column;
```

3.2. LEFT OUTER JOIN.

Este tipo de combinación devuelve todas las filas de la tabla IZQUIERDA especificada en la condición ON y solo aquellas filas de la otra tabla donde los campos combinados son iguales ie. que se cumple la condición de unión.

```
SELECT columns  
FROM table1  
LEFT [OUTER] JOIN table2  
ON table1.column = table2.column;
```

3.3. RIGHT OUTER JOIN.

Este tipo de unión devuelve todas las filas de la tabla DERECHA especificada en la condición ON y solo aquellas filas de la otra tabla donde los campos combinados son iguales ie. que se cumple la condición de unión.

```
SELECT columns
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;
```

3.4. RIGHT OUTER JOIN.

Este tipo de unión devuelve todas las filas de la tabla IZQUIERDA y la tabla DERECHA con nulos en el lugar donde no se cumple la condición de unión.

```
SELECT columns
FROM table1
FULL [OUTER] JOIN table2
ON table1.column = table2.column;
```

4. FUNCIONES DE AGREGACION.

Las funciones de agregado realizan un cálculo sobre un conjunto de valores y devuelven un solo valor. Si exceptuamos la función COUNT, todas las funciones de agregado ignoran los valores NULL. Las funciones de agregado se suelen utilizar con la cláusula GROUP BY de la instrucción SELECT.

Todas las funciones de agregado son deterministas. Esto significa que las funciones de agregado devuelven el mismo valor cada vez que se las llama con un conjunto específico de valores de entrada.

Todas estas funciones se aplican a una sola columna, que especificaremos entre paréntesis, excepto la función COUNT, que se puede aplicar a una columna o indicar un *. La diferencia entre poner el nombre de una columna o un *, es que en el primer caso no cuenta los valores nulos para dicha columna, y en el segundo si.

Las funciones de grupo son :

COUNT(* | [DISTINCT | ALL])

Cantidad de filas con expr no nulo. Con * se cuentan todas las filas incluyendo duplicadas y valores nulos.

SUM(columna)

AVG([DISTINCT | ALL] n)

Valor promedio de n.

MIN([DISTINCT | ALL])

MAX([DISTINCT | ALL])

5. SUBCONSULTAS

El uso de subconsultas es una técnica que permite utilizar el resultado de una tabla SELECT en otra consulta SELECT. Permite solucionar consultas complejas mediante el uso de resultados previos conseguidos a través de otra consulta.

El SELECT que se coloca en el interior de otro SELECT se conoce con el término de SUBSELECT. Ese SUBSELECT se puede colocar dentro de las cláusulas WHERE, HAVING, FROM o JOIN.

5.1. USO DE SUBCONSULTAS SIMPLES.

Las subconsultas simples son aquellas que devuelven una única fila. Si además devuelven una única columna, se las llama subconsultas escalares, ya que devuelven un único valor.

5.2. USO DE SUBCONSULTAS MÚLTIPLES FILAS.

En la sección anterior se comentaba que las subconsultas sólo pueden devolver una fila. Pero a veces se necesitan consultas del tipo: mostrar el atributo A y B de la tabla X cuyo atributo A supera al de cualquier elemento del subconjunto de X.

La subconsulta necesaria para ese resultado mostraría todos los A del subconjunto de X. Pero no podremos utilizar un operador de comparación directamente ya que esa subconsulta devuelve más de una fila. La solución a esto es utilizar instrucciones especiales entre el operador y la consulta, que permiten el uso de subconsultas de varias filas.

ANY o SOME .- Compara con cualquier registro de la subconsulta. La instrucción es válida si hay un registro en la subconsulta que permite que la comparación sea cierta. Se suele utilizar la palabra ANY (SOME es un sinónimo)

ALL .- Compara con todos los registros de la consulta. La instrucción resulta cierta si es cierta toda comparación con los registros de la subconsulta

IN .- No usa comparador, ya que sirve para comprobar si un valor se encuentra en el resultado de la subconsulta

NOT IN .- Comprueba si un valor no se encuentra en una subconsulta

5.3. CONSULTAS CON EXISTS.

Este operador devuelve verdadero si la consulta que le sigue devuelve algún valor. Si no, devuelve falso. Se utiliza normalmente mediante consultas correlacionadas.

Ejemplo:

```
SELECT tipo,modelo, precio_venta FROM piezas p
WHERE EXISTS (SELECT tipo,modelo FROM existencias
WHERE tipo=p.tipo AND modelo=p.modelo);
```

5.4. SUBCONSULTAS EN LA INSTRUCCIÓN UPDATE.

La instrucción UPDATE permite modificar filas. Es muy habitual el uso de la cláusula WHERE para indicar las filas que se modificarán. Esta cláusula se puede utilizar con las mismas posibilidades que en el caso del SELECT, por lo que es posible utilizar subconsultas.

Ejemplo:

```
UPDATE empleados SET sueldo=sueldo*1.10
WHERE id_seccion =(SELECT id_seccion FROM secciones
WHERE nom_seccion='Producción');
```

También podemos utilizar subconsultas en la cláusula SET de la instrucción UPDATE.

Ejemplo:

```
UPDATE empleados SET puesto_trabajo=(SELECT puesto_trabajo
FROM empleados WHERE id_empleado=12) WHERE seccion=23;
```

5.5. SUBCONSULTAS EN LA INSTRUCCIÓN DELETE.

Al igual que en el caso de las instrucciones INSERT o SELECT, DELETE dispone de cláusula WHERE y en dichas cláusulas podemos utilizar subconsultas.

Ejemplo:

```
DELETE empleados WHERE id_empleado IN
(SELECT id_empleado FROM errores_graves);
```

6. Actividad

Deberán crear un nuevo table space como en la práctica pasada.

En nuevo table space debera contener el esquema de la base de datos que se proporciona en la carpeta de la practica, para solo es necesario correr el script practica10.sql el cual contiene la creacion de las tablas del siguiente esquema relacional.

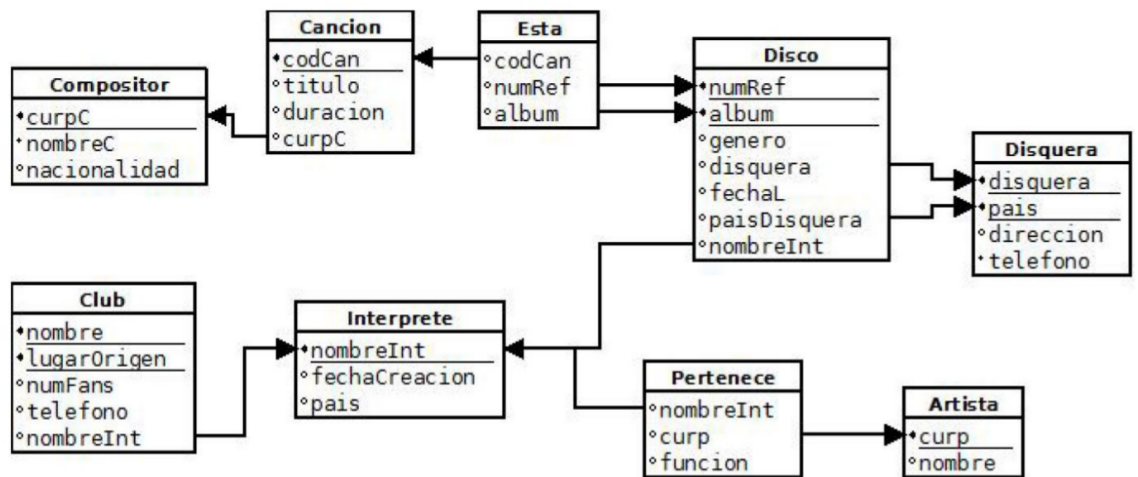


Figura 1: Diagrama de Disqueras.

7. Entregables

Se debe crear un script consultas.sql el cual contendrá la solución de las siguientes consultas.

1. Selecciona el nombre de los intérpretes que no sean de México. (Un intérprete puede ser un grupo o un solista.)
2. Obtener el título de las canciones con más de 5 minutos de duración.
3. Obtener el nombre y el lugar de origen de cada club de fans de intérpretes de México así como el nombre del intérprete al que admiran.
4. Obtener los nombres de las canciones que dan nombre al disco en el que aparecen.

5. El dúo dinámico por fin se jubila para sustituirlos se pretende hacer una selección sobre todos los pares de artistas españoles distintos tales que el primero sea voz y el segundo guitarra. Obtener dicha selección.
6. Obtener el título de la canción de mayor duración indicando la duración.
7. Obtener los nombre de los artistas de grupos con clubes de fans de más de 500 personas y que el grupo sea de Inglaterra.
8. Obtener el décimo (Debe haber sólo 9 por encima de el) club con mayor número de fans indicando este número 3).
9. Obtener el nombre de los artistas que tengan la función de bajo en un único grupo y que además este tenga más de dos miembros.
10. Indica el nombre del compositor que más canciones ha creado y el título de estas.
11. Obtener el año en el que hubo mayor lanzamientos de discos.

Se debe crear un script DML.sql el cual contendrá 10 insert, 10 updates y 10 delete sobre las tablas creadas con el escript practica10.sql.

La entrega deberá ser el día lunes 1 de Mayo de 2018.