

# Fundamentos de Bases de Datos.

## Práctica 9.

Profesor: M.I. Gerardo Avilés Rosas  
*gar@ciencias.unam.mx*  
Laboratorio: Luis Eduardo Castro Omaña  
*lalo\_castro@ciencias.unam.mx*

28 de octubre de 2019

Se dan a conocer especificaciones de entrega para la práctica 9.

### 1. DML

#### 1.1. 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.

#### 1.2. SINTAXIS BÁSICA :

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

#### 1.3. JOIN

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

Existen los siguientes tipos de uniones:

```
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`)

#### **1.4. INNER JOIN.**

Es el tipo de JOIN más común. El `INNER JOIN`s 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;
```

#### **1.5. 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;
```

#### **1.6. 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;
```

#### **1.7. FULL 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;
```

## 1.8. 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.

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 ] )

## 1.9. 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.

### 1.10. 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.

### 1.11. USO DE SUBCONSULTAS MÚLTIPLES FILAS.

En la sección anterior se comentó 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

### 1.12. 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);
```

### 1.13. 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;
```

### 1.14. 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);
```

## 2. Actividad

1. Modificar y agregar las tablas faltantes en la base de datos. Para ello deben ejecutar el script DML.sql, el cual contiene la corrección de la tabla Pedidos y las tablas faltantes para completar la base de datos.
2. Poblar la base de datos. El script DML.sql contiene las instrucciones para poblar la base de datos, 3308 valores deben ser ingresados a la base de datos.
3. Con la base de datos actualizada y debidamente poblada, resolver las siguientes consultas:
  - El nombre completo con título de cortesía y cantidad de años laborando del empleado con más antigüedad.
  - Id del pedido, fecha de entrega, fecha de envío de los pedidos que se hayan realizado por un empleado perteneciente al país 'UK' cuyo destino es México.
  - El nombre completo y la fecha de entrega de los clientes que hayan solicitado pedidos de productos con la categoría 'Seafood'.
  - Supón que la fecha actual es 1 de Enero de 1998. Encontrar los pedidos de los productos descontinuados que aún no han llegado (se considera que un producto no ha llegado si la fecha de entrega del pedido es mayor a la actual o null).
  - Todos los productos que hayan sido despachados por los empleados de la región 'Eastern'.

### 2.1. Punto Extra.

Para la entrega de esta práctica los únicos datos que deben contener su base de datos son los que se cargan con el script DML.sql. Para obtener un punto extra deberán crear un backup (.bak file) de la base de datos con los datos que se insertaron en la práctica pasada.

Se debe especificar los pasos necesarios para poder restaurar la base de datos en otra instancia (diferente a la suya) de SQL.

## 3. Entregables

Se debe crear un script consultas.sql el cual contendrá la solución a cada una de las consultas.

### **3.1. Punto Extra**

Agrega a tu reporte los pasos necesarios para poder restaurar su base de datos. Si al seguirse los pasos establecidos en su reporte no se restaura la bd no te tomará en cuenta el punto extra.

La entrega deberá ser el día lunes 4 de Noviembre de 2019.