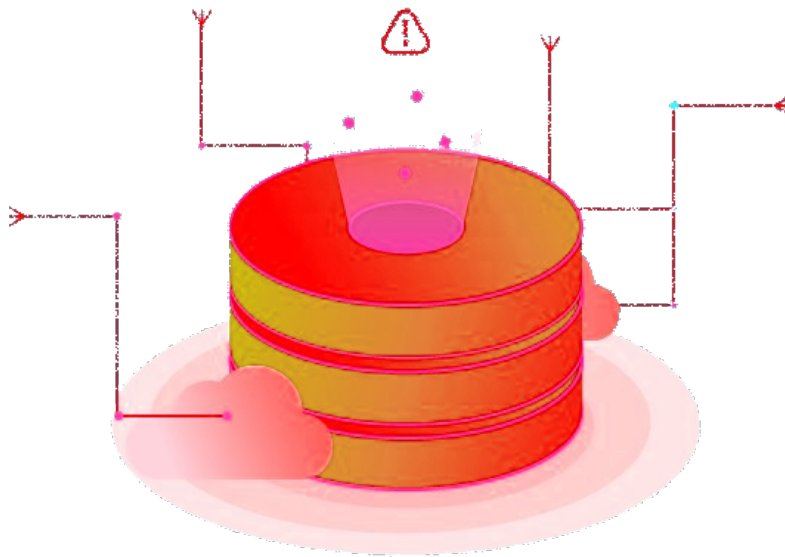


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS, 2024-I
FUNDAMENTOS DE BASES DE DATOS



PRÁCTICA 09:
DML. Consultas Avanzadas.

PROFESOR:
Gerardo Avilés Rosas

AYUDANTES DE TEORÍA:
Gerardo Uriel Soto Miranda
Valeria Fernanda Manjarrez Angeles

AYUDANTES DE LABORATORIO:
Ricardo Badillo Macías
Jerónimo Almeida Rodríguez

DML

Operadores

Elementos básicos para el procesamiento de datos. Son **usados en expresiones, toman uno o dos argumentos y regresan un valor**. PostgreSQL soporta operadores para todos los tipos de datos.

Lista de Operadores

```
+ - * / < > = ~ ! @ # % ^ & | ' ' ?
```

Operadores como multiplicación, división, tienen mayor precedencia que otros, los operadores lógicos o de comparación tienen la menor precedencia. Operadores con la misma precedencia son ejecutados de izquierda a derecha. Mas información en: <https://www.postgresql.org/docs/current/sql-syntax-lexical.html>

Caracteres Especiales

El lenguaje SQL incluye los siguientes caracteres especiales:

- **Parentesis ()** Controlan la precedencia de las operaciones en un grupo de expresiones. También identifican tuplas o atributos en un tipo compuesto, o parámetros de una función.
- **Corchete []** Elementos de un arreglo.
- **Dos puntos :** Para acceder a elemento de arreglo.
- **Doble dos puntos ::** Para castear tipos.
- **Coma ,** Separar elementos de una lista.
- **Punto .** Para separar nombre de *schemas*, tablas y columnas una de otra.
- **Punto y coma ;** Terminar una instrucción.
- **Asterisco *** Referirse a todos los campos de una tabla.

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 la Bases de Datos**. El lenguaje de manipulación de datos más popular hoy en 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.

```
INSERT INTO nombreTabla(columna1, columna2, ...)  
VALUES (valor1, valor2, ...);
```

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

```
DELETE FROM nombreTabla  
WHERE condición;  
  
DELETE FROM nombreTabla;
```

- **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.

```
UPDATE nombreTabla
SET columna1 = valor1, columna2 = valor2, ...
WHERE condición;
```

SELECT

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

El *SMBD* se encarga de traducir la consulta en un "plan de consulta" que puede variar entre ejecuciones, versiones de base 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 una condición al recuperar un conjunto de datos de una tabla o conjunto de tablas.

```
SELECT columna1, columna2, ...
FROM nombreTabla
WHERE condición;
```

- **ORDER BY:** Establece el orden de las filas de resultado en función de las columnas que se indiquen.

```
SELECT columna1, columna2, ...
FROM nombreTabla
ORDER BY columna1, columna2 ASC, columna3 DESC, ...;
```

- **AS:** Proporciona un alias que se puede usar para cambiar el nombre temporalmente de tablas o columnas.

```
SELECT columna1 AS algunNombre1, columna2 AS algunNombre2, ...
FROM nombreTabla;

SELECT columna1, columna2, ...
FROM nombreTabla AS algunNombre;
```

Funciones de agregación

Las funciones de agregado realizan un cálculo sobre un conjunto de valores y devuelven un solo valor. Si ejecutamos 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 . (con se cuenta valores nulos).

Las funciones de agregación son:

- COUNT(*| DISTINCT | ALL|) Cantidad de filas con expr no nulo. Con * se cuenta todas las filas incluyendo duplicadas y valores nulos.
- SUM(columna)
- AVG(|DISTINCT | ALL | n) Valor promedio de n.
- MIN (|DISTINCT | ALL|)
- MAX (|DISTINCT | ALL|)

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.

Uso de subconsultas simples

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

Uso de subconsultas múltiples filas

Las subconsultas sólo pueden devolver una fila. Pero también es necesario 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.

Las operaciones son:

- ANY o SOME: Compara con cualquier registro de la subconsulta. La instrucción es válida si hay registro en la subconsulta que permite que la comparación sea cierta.
- ALL: Compara con todos los registros de la consulta. La instrucción resulta cierta si es cierta toda comparación con los registros de la consulta.
- 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.

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.

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.

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.

Actividades.

- Se debe de realizar un script DML llamado Query.sql que contenga la solución a las siguientes consultas:
 - Los veterinarios y cuidadores que comparten el apellido.
 - Los animales herbivoros que pertenezcan al bioma bosque templado.
 - Alimentos de tipo semilla, que son distribuidos en el aviario.
 - Los cuidadores y proveedores que vivan en el mismo estado.
 - Biomas que posean al menos 10 animales.

Cada consulta debera al menos regresar 5 registros.

IMPORTANTE: No esta permitido hacer uso de JOIN, en caso de que sea necesario cruce de información deberán hacer uso de subconsultas.

- En el documento Práctica09.pdf, deberan agregar la solucion de cada consulta, ademas de una pequeña explicación de como lo solucionaron.



Figura 1: Actividades.

Entregables.

Deberán subir un archivo con formato *zip* a *Google Classroom*, de acuerdo a lo indicado en los lineamientos de entrega. Debe de estar organizado de la siguiente manera, (suponiendo que el nombre del equipo que está entregando es *Dream Team*).

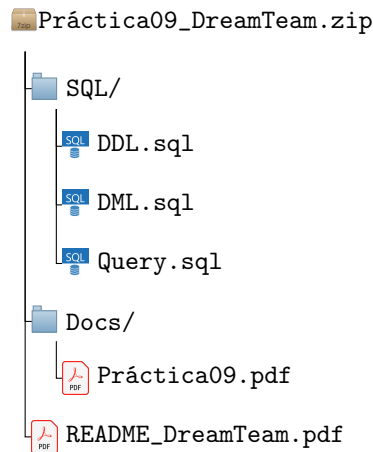


Figura 2: Entregables.

Nota.

Para cualquier duda o comentario que pudiera surgirles al hacer este trabajo, recuerden que cuentan con la asignación de este entregable en el grupo de *Classroom*, en donde seguramente encontrarás las respuestas que necesites.



Figura 3: Nota.