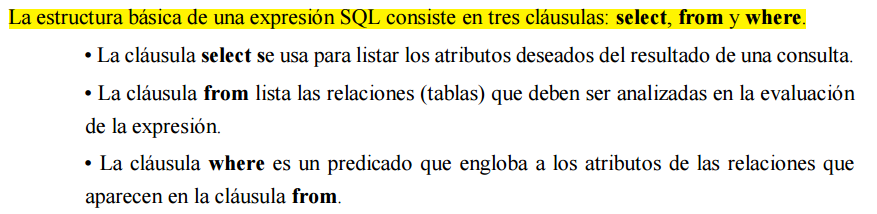
Structured Query Language

SQL



El lenguaje SQL y sus comandos se clasifican principalmente en cuatro categorías.

( usamos sobretodo las 3 resaltadas )

DDL: Lenguaje de definicion de datos.

DQL: Lenguaje de consulta de datos.

DML: Lenguaje de manipulacion de datos.

DCL: Lenguaje de control de datos.

--------------------------------------------- Teoria -----------------------------------------

( ⬇Clasificaciones mas importantes ⬇)

DDL : Data Definition Language (Lenguaje de Definición de Datos)

este lenguaje permite definir las estructuras que almacenarán los datos.

CREATE // ALTER // DROP // etc

DML : Lenguaje de Manipulación de Datos

permite realizar varias acciones con datos de una bd.

SELECT // INSERT // UPDATE // DELETE // etc

DQL : Lenguaje de Consulta de Datos

realizar consultas sobre los datos dentro de los objetos del esquema (tabla)

SELECT

( ⬇Menos importantes para la materia ⬇)

DLC : lenguaje de control de datos

Permite crear roles, permisos e integridad referencial, asi como el acceso a la bd

GRANT//REVOKE

TCL: lenguaje de control transicional

Permite hacer transaciones en la base de datos

COMMIT // ROLLBACK // SAVEPOINT

LA INTEGRIDAD REFERENCIAL es un sistema compuesto por reglas que permiten la garantía de las relaciones entre las filas de la tabla relacionadas para que sean

válidas y que estas relaciones no se eliminen ni se modifiquen por error.

--------------------------------------------- Base de Datos -----------------------------------------

CREATE DATABASE ( crear base de datos )

USE (selecionar la base de datos para trabajar- crear tablas,etc)

DROP DATABASE (eliminar base de datos)

=== === === === === === === TABLAS === === === === === === ===

1- CREATE TABLE nombre ()

sintaxis:

tabla (atributo tipo\_de\_dato propiedades,

atributo tipo\_de\_dato propiedades)

2- INSERT INTO tabla (atributo1,atributo2,etc) VALUES (‘valor1’,’valor2’,’1’,etc)

3- ALTER TABLE ( modificar una tabla ya existente)

1. UPDATE (update tabla) ( se usa para setear/cambiar una insercion)

Se pueden cambiar una o mas columnas de una tupla

Syn: UPDATE tabla set columna= columna\*2

1. DELETE =eliminar una tupla (se borra todo, no se puede borrar un atributo solamente)

Syn: DELETE FROM tabla WHERE condicion

Syn: DELETE FROM tabla (elimina todas las tuplas)

N: borra tuplas de una sola relacion

## **Resumen rápido**

* **UPDATE** → modifica **los datos dentro de la tabla**.
* **ALTER TABLE** → modifica **la forma/estructura de la tabla**.

✦-✦-✦-✦-✦-✦-✦-✦-✦-✦-✦ SELECTORES ✦-✦-✦-✦-✦-✦-✦-✦-✦-✦-✦-✦

SELECT (devuelve las tuplas con la columna selecionada)

SELECT \* FROM tabla (devuelve las tuplas con todos las columnas)

SELECT columna1,columna2,etc (selecionar varias columnas a mostrar, parecido a selector de agrupamiento en CSS)

SELECT columna FROM tabla1,tabla2 (crea un producto cartesiano uniendo las dos tablas)

COUNT (\*) (no devuelve tuplas, devuelve la cantidad de tuplas que cumplen las condiciones

N: no se puede usar con distinct

◈-◈-◈-◈-◈-◈-◈-◈-◈-◈-◈ CONDICIONALES ◈-◈-◈-◈-◈-◈-◈-◈-◈-◈-◈

3- WHERE = (select \* from tabla where(condicion)) (implementa una condición en un select )

4- (=// < // > // <= // >= ) = Operadores Lógicos (se usa con SELECT)

5- AND // OR // NOT = Condiciones Lógicas (en WHERE para mas de una condicion)

6- BETWEEN = entre un valor AND otro valor (tambien existe NOT BETWEEN)

7- distintc = (solo resultados distintos(sin copias))

8- IN = cuando queres especificar ciertos valores

ej envés de escribir, que sea sea las provincias mendoza, o sanluis o san juan con =

WHERE provincia = 'Mendoza'

OR provincia = 'San Juan'

OR provincia = 'San Luis';

se escribe IN cuando este atributo tenga cualquiera de estos valores

WHERE provincia IN ('Mendoza','San Juan','San Luis'); y no se nesesita conectores logicos

1. LIKE(%) = buscar por alguna letra envés de palabra Syn:LIKE‘%abc’

Escape(\) ‘caracter%’ : cualquier cadena que empieze ‘x%’

Para poder escribir ‘%caracter%’: cualquier cadena que contenga ‘%x%’

Un simbolo ‘\_ \_ \_’ : cualqueir palabra de 3 letras

/%porcentaje =(%porcentaje) ‘\_ \_ \_%’ : cualquier palabra de almenos 3 letras

(tambien se puede usar NOT LIKE)

9- JOIN tabla ON otra\_tabla = se usa para unir tablas (producto cartesiano)

10- DESCRIBE tabla = retorna informacion de los atributos, tipo de dato, si es llave,etc

1. NUILL =Buscar donde valor es nulo Syn: atributo= is null
2. as = renombrea tablas y columnas (Syn: tabla as tablanueva)

se puede usar tanto en SELECT como FROM

(sirve para escribir menos, no cambia la base de datos)

1. order by = ordena de forma acendente (asc) y descendente (des)

Por defecto es en orden acendente

Syn: order by atributo

Ordenar 2 parametros Syn: order by atributo(asc/des), otroatributo(asc/des)

N: se ordena cuando sea nesesario(es tardado y costoso)}

1. group by = ordena las tuplas en grupo, agrupandolas por el mismo

valor de un atributo especificado

Syn: gruoup by (columna)

1. having avg = teniendo un avg y un group by se puede usar esta clausula. Se promedio una columna, se dividen en grupos y a cada grupo se le asigna una selecion segun se nesesite

avg(columna) ---> having avg (columna) > 10

\_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ Claves para ATRIBUTOS \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_

PRIMARY KEY : define un atributo como clave primaria

FOREIGN KEY : se usa para cuando se quiere identificar que un atributo es clave foranea

NOT NULL : el valor no puede ser nulo

UNIQUE : el valor debe ser unico

------------------------------- CALCULOS/ Funciones de Agregacion -----------------------------------

SELECT SUM(columna-int) = Suma todos los valores de una Columna

SELECT AVG(columna-int) = Calcula el promedio de los valores de una Columna

SELECT MIN(columna) = Devuelve el valor minimo de una Columna

SELECT MAX(columna) = Devuelve el valor maximo de una Columna

COUNT

---------------------------------------- SubConsultas -----------------------------------------------