

Asignatura

Base de datos



**UNIVERSAE**  
Instituto Superior de FP

Asignatura

Base de datos

## UNIDAD 5

El lenguaje SQL. DML y DCL



**UNIVERSAE**  
Instituto Superior de FP



# Lenguaje de manipulación de datos (DML)

¿Como podemos utilizar los datos?

- Mediante DML (Data manipulation language)
- SQL contiene instrucciones para la manipulación de los datos

## Principales funciones

- Consulta de datos. **SELECT**
- Inserción de registros. **INSERT**
- Modificación de registros. **UPDATE**
- Eliminación. **DELETE**
- Crear vistas a partir de una consulta

## Guía de apoyo

<https://www.w3schools.com/sql/>





# Consultas. SELECT

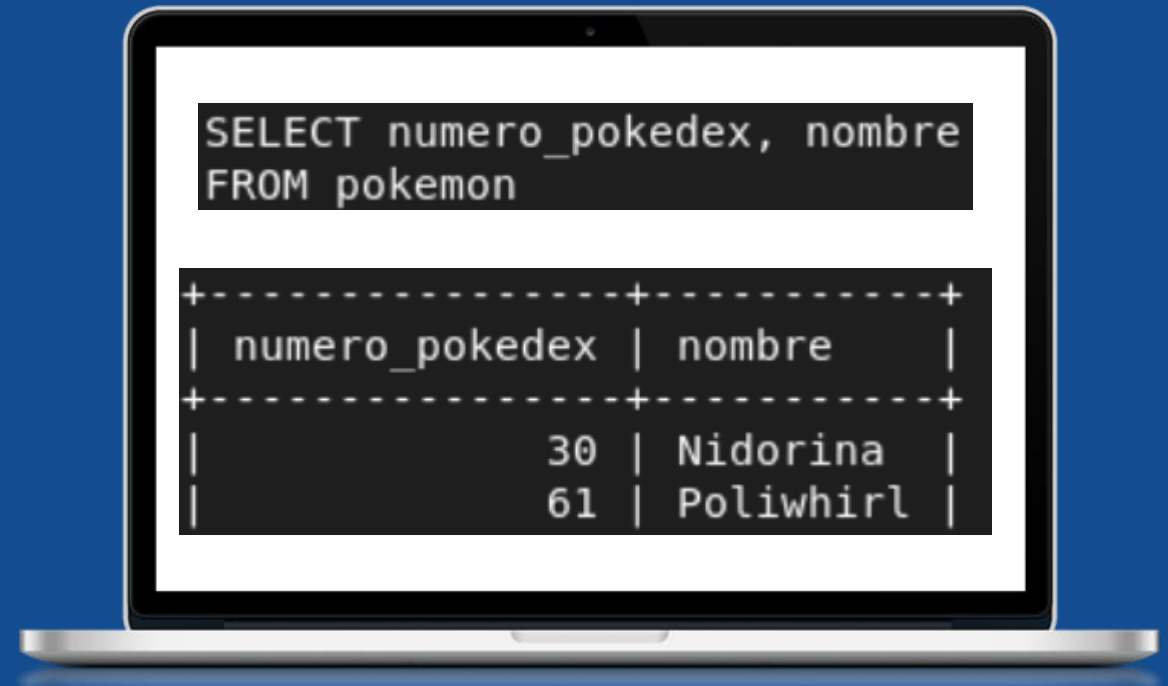
URL: [https://www.w3schools.com/sql/sql\\_select.asp](https://www.w3schools.com/sql/sql_select.asp)  
[https://www.w3schools.com/sql/sql\\_distinct.asp](https://www.w3schools.com/sql/sql_distinct.asp)  
[https://www.w3schools.com/sql/sql\\_orderby.asp](https://www.w3schools.com/sql/sql_orderby.asp)  
[https://www.w3schools.com/sql/sql\\_alias.asp](https://www.w3schools.com/sql/sql_alias.asp)

## Sintaxis

```
SELECT [DISTINCT] campos  
FROM tablas  
[ORDER BY campos];
```

## Conceptos

- Los campos se pueden especificar por su nombre o usar asterisco (\*) para especificar todos los campos
- La clausula **DISTINCT** especifica valores no repetidos.
- **ORDER BY** permite ordenar ascendentemente los valores de los campos. Para cambiar el orden, se añade **DESC**
- Se puede cambiar el nombre de los campos con los alias. **AS "nombre"**





# Consultas. WHERE

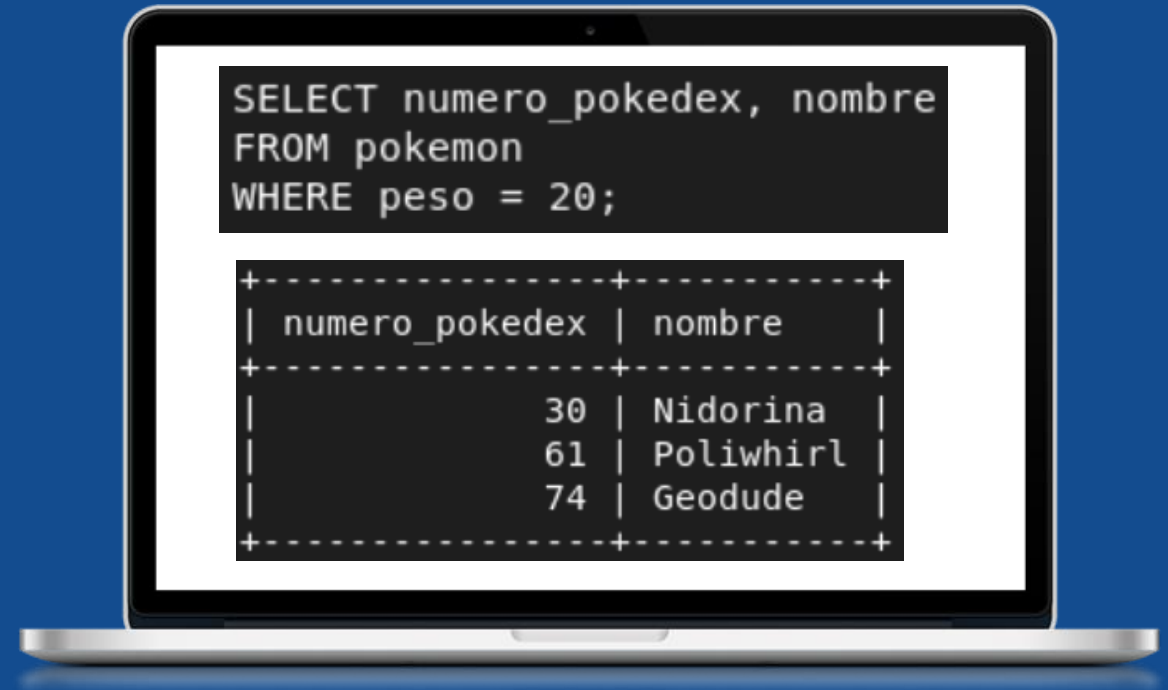
URL: [https://www.w3schools.com/sql/sql\\_where.asp](https://www.w3schools.com/sql/sql_where.asp)  
[https://www.w3schools.com/sql/sql\\_operators.asp](https://www.w3schools.com/sql/sql_operators.asp)  
[https://www.w3schools.com/sql/sql\\_between.asp](https://www.w3schools.com/sql/sql_between.asp)  
[https://www.w3schools.com/sql/sql\\_like.asp](https://www.w3schools.com/sql/sql_like.asp)  
[https://www.w3schools.com/sql/sql\\_and\\_or.asp](https://www.w3schools.com/sql/sql_and_or.asp)  
[https://www.w3schools.com/sql/sql\\_null\\_values.asp](https://www.w3schools.com/sql/sql_null_values.asp)

## Sintaxis

```
SELECT [DISTINCT] campos
FROM tablas
[WHERE condición]
[ORDER BY campos];
```

## Conceptos

- La condición puede estar formada por:
  - ☐ Operadores de comparación <, <=, >, >=, <>, =
  - ☐ Intervalos. BETWEEN
  - ☐ Operadores lógicos AND, OR, NOT
  - ☐ Comprobación de valores nulos. IS NULL o IS NOT NULL
  - ☐ Comparador de cadenas LIKE





# Ejemplo. WHERE

## BETWEEN

```
SELECT altura, nombre  
FROM pokemon  
WHERE altura BETWEEN 10 AND 30;
```

## NULL

```
SELECT *  
FROM tipo  
WHERE nombre IS NOT NULL;
```

## LIKE

```
SELECT *  
FROM pokemon  
WHERE nombre LIKE "B%";
```

## OPERADORES (AND, OR, NOT)

```
SELECT id_tipo  
FROM pokemon_tipo  
WHERE numero_pokedex = 1  
AND id_tipo = 1;
```



# Consultas. Funciones de agregación

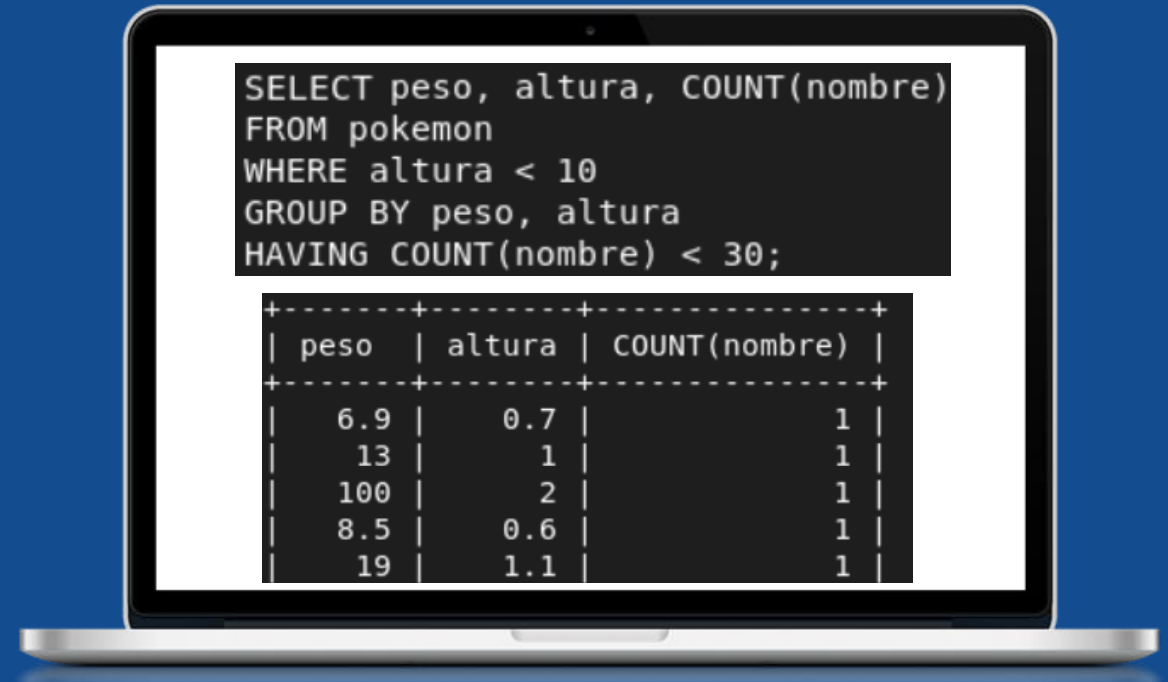
URL: [https://www.w3schools.com/sql/sql\\_groupby.asp](https://www.w3schools.com/sql/sql_groupby.asp)  
[https://www.w3schools.com/sql/sql\\_having.asp](https://www.w3schools.com/sql/sql_having.asp)  
[https://www.w3schools.com/sql/sql\\_min\\_max.asp](https://www.w3schools.com/sql/sql_min_max.asp)  
[https://www.w3schools.com/sql/sql\\_count\\_avg\\_sum.asp](https://www.w3schools.com/sql/sql_count_avg_sum.asp)

## Sintaxis

```
SELECT [DISTINCT] campos
FROM tablas
[WHERE condición]
[GROUP BY campos
    [HAVING condición_agrupación]]
[ORDER BY campos];
```

## Conceptos

- Las funciones de agregación son: COUNT(), MAX(), MIN(), SUM(), AVG()
- La agrupación se realiza mediante GROUP BY
- La condición de agrupación se realiza mediante HAVING





# Consultas. Relaciones

URL: [https://www.w3schools.com/sql/sql\\_join.asp](https://www.w3schools.com/sql/sql_join.asp)

## Sintaxis

- Opción 1

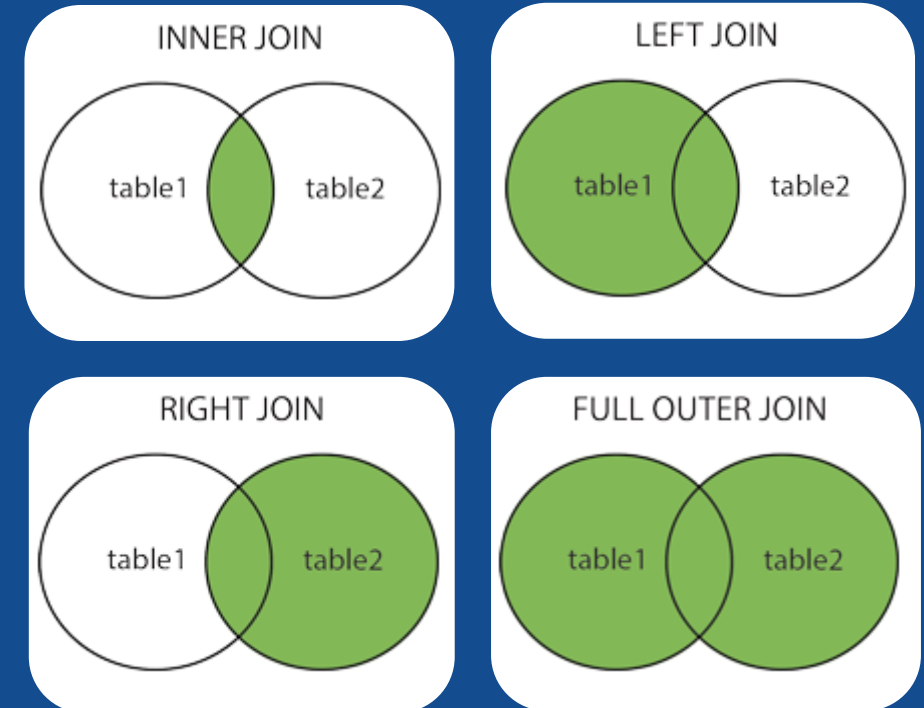
```
SELECT [DISTINCT] campos  
FROM tabla1, tabla2, ..., tablaN  
WHERE condición relación
```

- Opción 2

```
SELECT [DISTINCT] campos  
FROM tabla1  
[INNER/LEFT/RIGHT/FULL OUTER] JOIN tabla2 ON condición relación
```

## Conceptos

- Los campos que marcan la relación son las claves primarias y foráneas
- Si los campos que las unen las tablas se llaman igual, hay que usar alias en las tablas para diferenciarlas







# Ejemplo. Relaciones

## WHERE

```
SELECT pokemon.nombre, tipo.nombre
FROM pokemon, pokemon_tipo, tipo
WHERE pokemon.numero_pokedex = pokemon_tipo.numero_pokedex
AND pokemon_tipo.id_tipo = tipo.id_tipo
ORDER BY pokemon.nombre;
```

| nombre     | nombre   |
|------------|----------|
| Abra       | Psíquico |
| Aerodactyl | Volador  |
| Aerodactyl | Roca     |
| Alakazam   | Tierra   |
| Arbok      | Veneno   |

## INNER JOIN

```
SELECT pokemon.nombre, tipo.nombre
FROM pokemon
INNER JOIN pokemon_tipo
ON pokemon.numero_pokedex = pokemon_tipo.numero_pokedex
INNER JOIN tipo
ON pokemon_tipo.id_tipo = tipo.id_tipo;
```

| nombre    | nombre |
|-----------|--------|
| Squirtle  | Agua   |
| Wartortle | Agua   |
| Blastoise | Agua   |
| Psyduck   | Agua   |
| Golduck   | Agua   |
| Poliwag   | Agua   |
| Poliwhirl | Agua   |



# Consultas. Combinación

URL: [https://www.w3schools.com/sql/sql\\_union.asp](https://www.w3schools.com/sql/sql_union.asp)

## Sintaxis

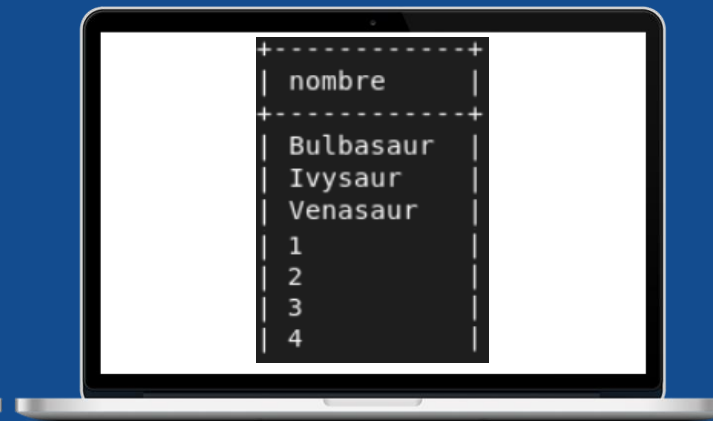
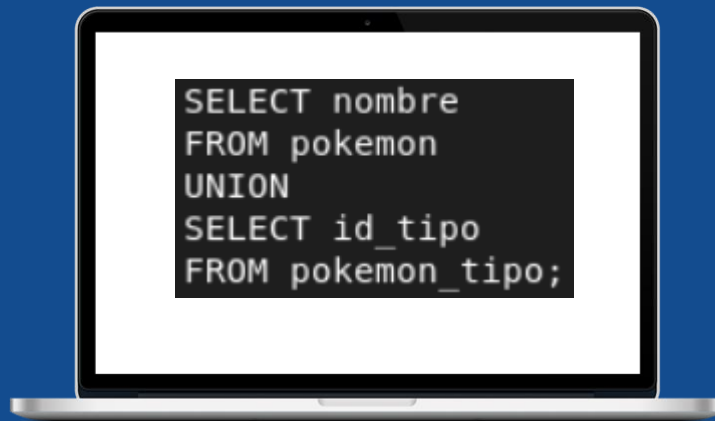
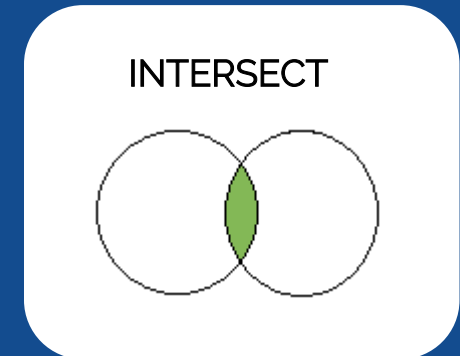
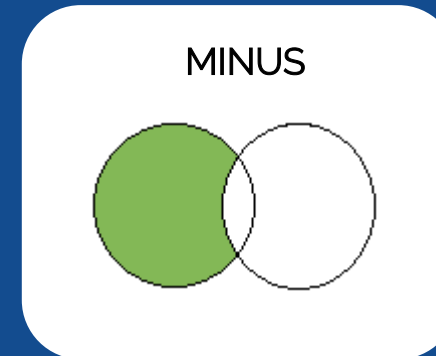
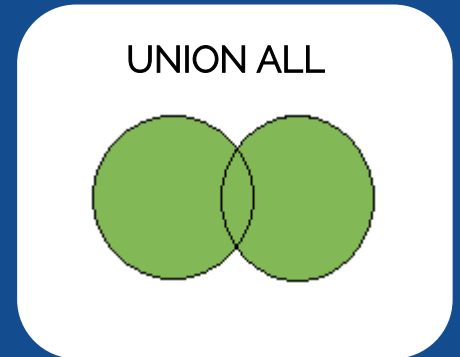
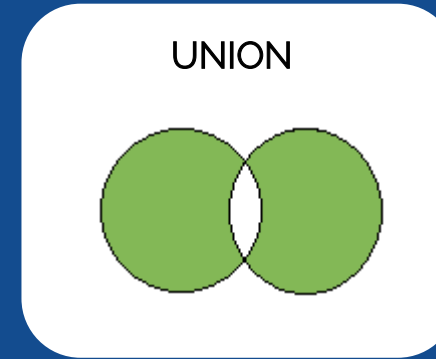
Consulta1

[UNION/UNION ALL/MINUS/INTERSECT]

Consulta2

## Conceptos

- Los números de campos deben ser iguales en las dos consultas





# Consultas. Subconsultas

URL: [https://www.w3schools.com/sql/sql\\_in.asp](https://www.w3schools.com/sql/sql_in.asp)

## Sintaxis

- Opción 1

SELECT [DISTINCT] campos

FROM tablas

WHERE condición = (consulta2)

- Opción 2

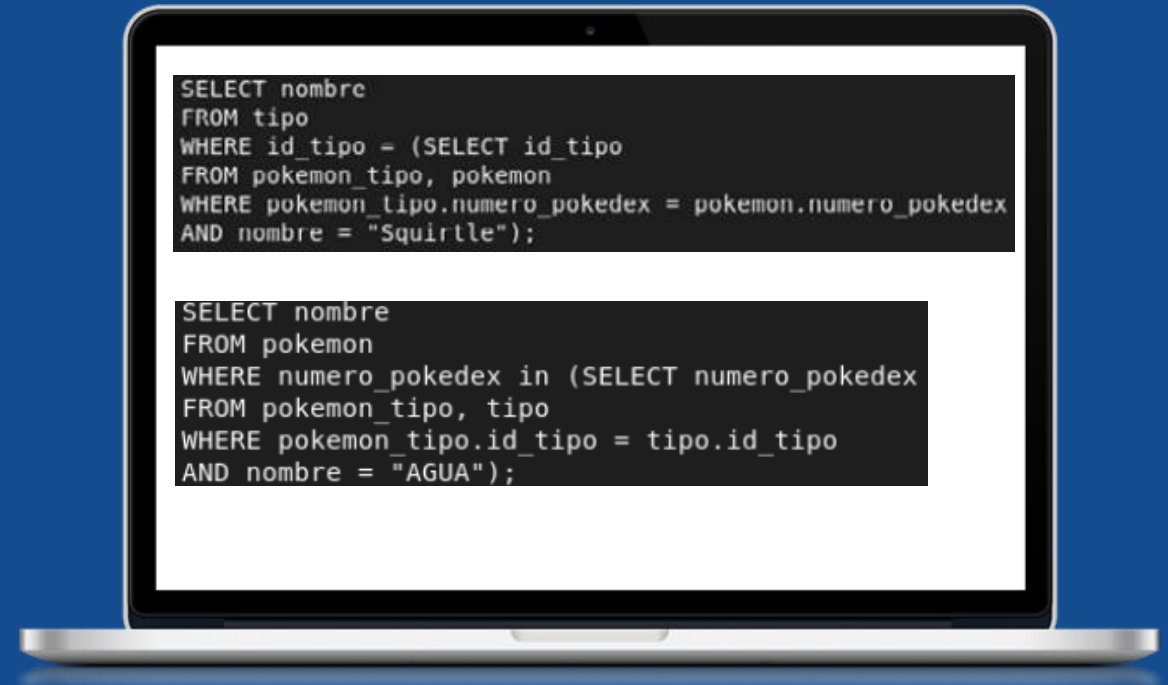
SELECT [DISTINCT] campos

FROM tablas

WHERE condición IN (consulta2)

## Conceptos

- La condición se puede realizar mediante:
  - ❑ Operadores de comparación. Para comparar un único valor
  - ❑ Clausula IN. Para un rango de valores.





# Funciones integradas

| Función     | Parámetros  | Valor devuelto   |
|-------------|---|--|
| ABS         | Número  | Valor absoluto del número  |
| MOD         | Número 1, Número 2  | Módulo o resto de la división del primero número entre el segundo.   |
| POSITION    | Cadena de caracteres 1,<br>Cadena de caracteres 2                 | Posición de la primera cadena dentro de la segunda.  |
| CHAR_LENGTH | Cadena de caracteres  | Tamaño de la cadena.   |
| SUBSTRING   | Cadena de caracteres,<br>Posición de inicio, Número de caracteres | Cadena de caracteres dentro de la cadena recibida como parámetro. El segundo de los parámetros define la posición de la cadena original a partir de la que se extraerá de la cadena que será devuelta. El tercer parámetro será el número de caracteres de la cadena resultante. |
| UPPER       | Cadena de caracteres  | Cadena de caracteres original en mayúsculas.   |
| LOWER       | Cadena de caracteres  | Cadena de caracteres original en minúsculas.   |

```
SELECT nombre, CHAR_LENGTH(nombre)
FROM pokemon;
```

```
+-----+-----+
| nombre | CHAR_LENGTH(nombre) |
+-----+-----+
| Bulbasaur | 9 |
| Ivysaur | 7 |
| Venasaur | 8 |
| Charmander | 10 |
+-----+-----+
```



# Insertar registros. INSERT

URL: [https://www.w3schools.com/sql/sql\\_insert.asp](https://www.w3schools.com/sql/sql_insert.asp)



## Sintaxis

- Opción 1

```
INSERT INTO tabla [(campo1[, campo2, ..., campoN])]  
VALUES (valor1[, valor2, ..., valorN]);
```

- Opción 2

```
INSERT INTO tabla [(campo1[, campo2, ..., campoN])]  
consulta;
```

```
INSERT INTO pokemon  
(numero_pokedex, nombre, peso, altura)  
VALUES  
("152", "Chikorita", "6", "1");
```

| numero_pokedex | nombre    | peso | altura |
|----------------|-----------|------|--------|
| 152            | Chikorita | 6    | 1      |



# Modificar registros. UPDATE

URL: [https://www.w3schools.com/sql/sql\\_update.asp](https://www.w3schools.com/sql/sql_update.asp)



## Sintaxis

```
UPDATE tabla  
SET  campo1 = valor1 [  
      campo2 = campo2  
      ...  
      campoN = valorN]  
[WHERE condición];
```

```
UPDATE pokemon  
SET numero_pokedex = 154  
WHERE numero_pokedex = 153;
```

| numero_pokedex | nombre  | peso | altura |
|----------------|---------|------|--------|
| 154            | Bayleef | 16   | 1      |



# Borrar registros. DELETE

URL: [https://www.w3schools.com/sql/sql\\_delete.asp](https://www.w3schools.com/sql/sql_delete.asp)



## Sintaxis

DELETE FROM tabla  
[WHERE condición];

```
DELETE FROM pokemon  
WHERE numero_pokedex = 153;
```

```
SELECT *  
FROM pokemon  
WHERE numero_pokedex > 152;
```

# Vistas

URL: [https://www.w3schools.com/sql/sql\\_view.asp](https://www.w3schools.com/sql/sql_view.asp)



## Las vistas

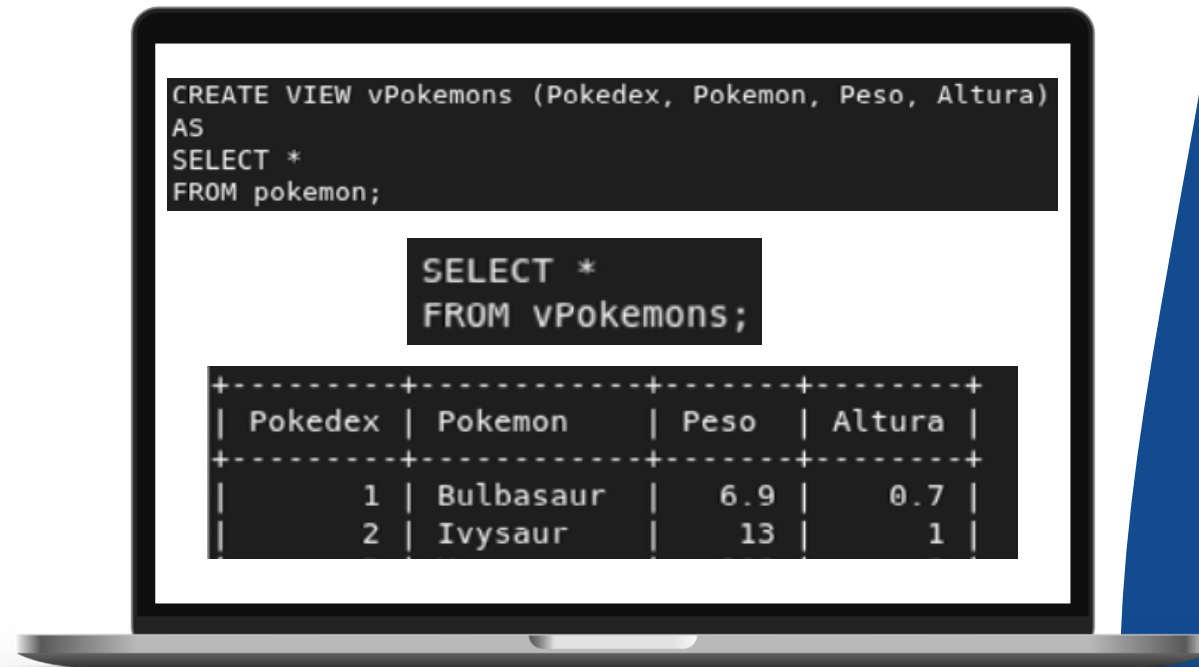
- Son tablas virtuales.
- Solo son de consulta.
- Se comporta como una consulta.
- Forman parte de la DDL



## Sintaxis

CREATE VIEW vista (campos)  
AS consulta;

DROP VIEW vista;







# Lenguaje de Control de datos (DCL)

## ¿Como administramos usuarios y sus acceso?

- Mediante DCL (Data control language)
- SQL contiene instrucciones para gestión de usuarios, roles, permisos e integridad referencial.

## Principales funciones

- Otorgar privilegios. **GRANT**

```
GRANT [CONNECT|RESOURCE|DBA|ALL|PRIVILEGES|SELECT|UPDATE|INSERT|DELETE]  
ON objeto  
TO usuarios  
[WITH GRANT OPTION];
```

- Retirar privilegios. **REVOKE**

```
REVOKE [CONNECT|RESOURCE|DBA|ALL PRIVILEGES|SELECT|UPDATE|INSERT|DELETE]  
FROM usuarios  
[ON objetos];
```

**URL:** <https://www.w3school-learn.com/2018/01/sql-grant-revoke-commands.html>





# Resumen

1. Lenguaje de manipulación de datos (DML)
2. Consultas. SELECT
3. Consultas. WHERE
4. Ejemplos. WHERE
5. Consultas. Funciones de agregación
6. Consultas. Relaciones
7. Ejemplo. Relaciones
8. Consultas. Combinación
9. Consultas. Subconsultas
10. Funciones integradas.
11. Insertar registros. INSERT
12. Actualizar registros. UPDATE
13. Borrar registros. DELETE
14. Vistas
15. Lenguaje de control de datos (DCL)

The background is a solid blue color. Overlaid on this are several faint, light-blue geometric patterns. These include a grid of small squares that form larger, irregular shapes, and numerous small, light-blue arrows pointing in various directions. The overall effect is a sense of movement and digital connectivity.

# UNIVERSAE

— CHANGE YOUR WAY —