

Introducción

- **Lenguaje de manipulación de datos**
(*Data Manipulation Language*)
- Permite a los usuarios llevar a cabo las tareas de consulta o manipulación de la BD.
- El más popular es SQL.

Tablas de ejemplo

Autos

	Patente	Marca	Modelo	Color	Kilometraje
1	AAA111	Ford	Focus	Gris	0
2	BBB222	Ford	Fiesta	Verde	15000
3	CCC333	Vw	Gol Naftero	Negro	17000
4	DDD444	Vw	Gol Diesel	Azul	25000
5	EEE555	Fiat	Palio	Blanco	0
6	FFF665	Fiat	Spazio	Celeste	80000
7	GGG777	Renault	Clio	Rojo	12000
8	HHH888	Renault	4	Verde	100000
9	III999	Peugeot	504	Negro	80000
10	JJJ000	Renault	Megane	Blanco	8000

Cientes

	id_cliente	nombre	Domicilio	Telefono	Patente
1	1	Juan	Corrientes 1234	444-4444	AAA111
2	2	Pedro	Paraguay 5678	444-5555	BBB222
3	3	Marta	Pte Roca 129	444-7777	CCC333
4	4	Josefina	España 3456	444-8888	NULL
5	5	Rigoberto	Italia 789	444-9999	NULL

INSERT

Sintaxis básica para insertar datos en una tabla:

```
INSERT INTO nombre_tabla  
(campo1, campo2, ...) VALUES  
(valor1, 'cadena1', ...)
```

Ejemplo:

INSERT INTO autos

(patente,marca,modelo,color,kilometraje)

VALUES ('ABC123', 'RENAULT',
'MEGANE TR100', 'NEGRO DIAMANTE',
78000)

UPDATE

Para modificar los datos de una tabla se usa UPDATE. Sintaxis:

```
UPDATE nombre_tabla  
SET campo_1 = nuevo_valor  
WHERE condición
```

También es posible UPDATE múltiples campos al mismo tiempo. Sintaxis:

- **UPDATE nombre_tabla
SET campo_1 = valor1, campo_2 =
valor2 WHERE condición**

Ejemplo:

```
UPDATE autos SET Marca='Ford' WHERE  
Modelo='Focus'
```

Asigna la cadena “Ford” en el campo marca, a todos los registros que tengan “Focus” como valor en el campo Modelo.

DELETE

Para borrar registros de una tabla se usa **DELETE FROM**. Sintaxis:

```
DELETE FROM nombre_tabla  
WHERE condición
```

Ejemplo:

```
DELETE FROM Autos WHERE  
patente='ABC123'
```

Borra los registros de autos que tengan “ABC123” en el campo patente.

SELECT

Para seleccionar registros de una tabla se usa la cláusula **SELECT**. Sintaxis:

```
SELECT nombre_campo1,  
nombre_campo2 FROM nombre_tabla
```

El símbolo * (asterisco) permite seleccionar todos los campos de una tabla. Sintaxis:

```
SELECT * FROM nombre_tabla
```

Ejemplo:

SELECT marca **FROM** autos

Resultado:

	marca
1	Ford
2	Ford
3	VW
4	VW
5	Fiat
6	Fiat
7	Renault
8	Renault
9	Peugeot
10	Renault

Muestra sólo el campo “marca” de todos los registros de autos

Para seleccionar todos los registros distintos de una tabla se usa **DISTINCT**.

Sintaxis:

```
SELECT DISTINCT nombre_campo  
FROM nombre_tabla
```

Ejemplo:

SELECT DISTINCT marca **FROM** Autos

Resultado:

	marca
1	Fiat
2	Ford
3	Peugeot
4	Renault
5	Vw

Muestra sólo el campo “marca” de todos los registros de “autos”, omitiendo los valores repetidos.

Para seleccionar condicionalmente los datos de una tabla se usa la cláusula **WHERE:**

Sintaxis:

```
SELECT nombre_campo FROM  
nombre_tabla  
WHERE condición
```

Ejemplo:

```
SELECT telefono FROM Clientes WHERE  
nombre='Juan'
```

Resultado:

	telefono
1	444-4444

Muestra el teléfono de todos los clientes cuyo nombre sea Juan.

La condición del **WHERE** puede ser una condición compuesta. Estas están formadas por múltiples condiciones simples conectadas por **AND** u **OR**.

Sintaxis:

```
SELECT nombre_campo FROM  
nombre_tabla  
WHERE condición simple AND/OR  
condición simple
```

Ejemplo:

```
SELECT patente FROM Autos WHERE  
    modelo='Megane' AND kilometraje>50
```

Resultado:

	patente
1	JJJ000

Muestra la patente de los autos cuyo modelo sea MEGANE y su kilometraje sea mayor que 50.

La cláusula **WHERE** compara sus campos comúnmente con valores únicos, pero también es posible comparar con un "conjunto" de valores. Esto es realizable a través del operador **IN**:

Sintaxis:

```
SELECT nombre_campo  
FROM nombre_tabla  
WHERE nombre_campo IN (valor1,  
valor2, ...)
```

Ejemplo:

```
SELECT kilometraje FROM autos WHERE  
    marca in ('Ford', 'Renault', 'Fiat')
```

Resultado:

	kilometraje
1	0
2	15000
3	0
4	80000
5	12000
6	100000
7	8000

Muestra el kilometraje de los autos cuya marca sea Ford o Renault o Fiat.

BETWEEN permite la selección de un rango de valores.

Sintaxis:

```
SELECT nombre_campo  
FROM nombre_tabla  
WHERE nombre_campo BETWEEN  
valor1 AND valor2
```

Ejemplo:

```
SELECT * FROM autos WHERE  
kilometraje BETWEEN 10000 AND  
20000
```

Resultado:

	Patente	Marca	Modelo	Color	Kilometraje
1	BBB222	Ford	Fiesta	Verde	15000
2	CCC333	VW	Gol Naftero	Negro	17000
3	GGG777	Renault	Clio	Rojo	12000

Muestra todos los campos de los autos cuyo kilometraje esté entre 10.000 y 20.000 km.

LIKE permite hacer una búsqueda basada en un patrón en vez de especificar exactamente lo que se desea (como en **IN**) o determinar un rango (como en **BETWEEN**). Sintaxis:

```
SELECT nombre_campo  
FROM nombre_tabla  
WHERE nombre_campo  
LIKE patrón
```

Patrón generalmente consiste en comodines.

- **'A_Z'**: Todas las líneas que comience con 'A', otro carácter y termine con 'Z'. Por ejemplo, 'ABZ' y 'A2Z'
- **'F%'**: Todas las líneas que comienzan con 'F'. Por ejemplo, 'Fiat' y 'Ford'
- **'%t'**: Todas las líneas que terminan con 't'. Por ejemplo, 'Renault', 'Fiat' y 'Peugeot'
- **'%o%'**: Todas las líneas que contienen "o" en cualquier lado. Por ejemplo, 'Peugeot' y 'Ford'

Ejemplo:

```
SELECT * FROM autos WHERE modelo  
LIKE 'GOL%'
```

Resultado:

	Patente	Marca	Modelo	Color	Kilometraje
1	CCC333	VW	Gol Naftero	Negro	17000
2	DDD444	VW	Gol Diesel	Azul	25000

Muestra todos los campos de los autos cuyo modelo comience con GOL.

Para enumerar el resultado en un orden particular (ascendente y descendente) se usa **ORDER BY**. Sintaxis:

```
SELECT nombre_campo  
FROM nombre_tabla  
WHERE condición  
ORDER BY nombre_campo ASC/DESC
```

Es posible ordenar por más de una columna. En este caso, la cláusula **ORDER BY** anterior se convierte en:

```
ORDER BY nombre_campo1 ASC/DESC,  
nombre_campo2 ASC/DESC
```

Ejemplo:

```
SELECT * FROM autos WHERE  
  marca='Ford' ORDER BY modelo ASC,  
kilometraje DESC
```

Resultado:

	Patente	Marca	Modelo	Color	Kilometraje
1	BBB222	Ford	Fiesta	Verde	15000
2	AAA111	Ford	Focus	Gris	0

Muestra todos los campos de los autos de marca Ford ordenados alfabéticamente por modelo, y luego por kilometraje en forma descendente.

Es posible hacer cálculos matemáticos, para esto se utilizan las funciones:

- **AVG (promedio)**
- **COUNT (contar)**
- **MAX (máximo)**
- **MIN (mínimo)**
- **SUM (suma)**

La sintaxis para el uso de funciones es:

```
SELECT tipo de función(nombre_campo)  
FROM nombre_tabla
```

Ejemplo:

SELECT AVG(kilometraje) FROM autos

Resultado:

	(No column name)
1	33700.000000

Muestra el promedio de los kilometrajes de todos los autos.

Para agrupar por un campo específico se utiliza **GROUP BY**.

Sintaxis:

```
SELECT nombre_campo1,  
       SUM(nombre_campo2)  
FROM nombre_tabla  
GROUP BY nombre_campo1
```

Ejemplo:

```
SELECT marca, AVG(kilometraje) FROM  
autos GROUP BY marca
```

Resultado:

	marca	(No column name)
1	Fiat	40000.000000
2	Ford	7500.000000
3	Peugeot	80000.000000
4	Renault	40000.000000
5	Vw	21000.000000

Muestra todas las marcas de autos con su promedio de kilometraje agrupado por marca.