CONSULTAS EN SQL

Para consultar datos de una tabla, se puede utilizar la instrucción **SELECT**.

```
SELECT columna1, columna2, ... FROM nombre_tabla;
```

Para seleccionar una lista completa de todos los registros de una tabla se usa **select * from** seguido del nombre de la tabla que queremos consultar (SELECT * FROM clientes).

La función **SELECT * FROM** es una instrucción básica en SQL que se utiliza para recuperar todos los datos de todas las columnas de una tabla en una base de datos. En otras palabras, esta instrucción se utiliza para seleccionar todas las filas y todas las columnas de una tabla.

```
MariaDB [ecoDosRuedas]> select * from clientes;
+---------+---------+-----------+-------------------------+------------+------------+------------------+
| alias   | nombre  | apellidos | email                   | celular    | contrasena | fecha_nacimiento |
+---------+---------+-----------+-------------------------+------------+------------+------------------+
| diva    | Ana     | Diaz      | divaDiaz@gmail.com      | 3156478294 |   10240058 | 2000-11-02       |
| dreamer | Luis    | Rojas     | dreamerRojas@gmail.com  | 3102897461 |   33025684 | 1993-12-03       |
| green   | Jorge   | Rodriguez | greenRodriguez@gmail.com| 3205467784 |   33887490 | 2003-10-30       |
| lucky   | Pedro   | Perez     | luckyperez@gmail.com    | 3102458794 |   14256133 | 1990-05-23       |
| malopez | Maria   | Lopez     | malopez@gmail.com       | 3221456897 |   44789465 | 1987-02-15       |
| neon    | Nelson  | Ruiz      | neonRuiz@gmail.com      | 3002145789 |   88426513 | 2002-06-24       |
| ninja   | Andres  | Cruz      | ninjaCruz@gmail.com     | 3128745699 |   22784511 | 1998-01-31       |
| rose    | Claudia | Mendez    | roseMendez@gmail.com    | 3162221034 |   11225468 | 1997-09-12       |
+---------+---------+-----------+-------------------------+------------+------------+------------------+
8 rows in set (0,001 sec)
```

El asterisco (*) se utiliza para seleccionar todas las columnas de la tabla. Si se quisiera seleccionar solo ciertas columnas, puedes reemplazar el asterisco con los nombres de las columnas separados por comas.
Para seleccionar los datos de un campo especifico usamos (SELECT nombre del campo FROM nombre de la tabla)

```
MariaDB [ecoDosRuedas]> select nombre from clientes;
+---------+
| nombre  |
+---------+
| Ana     |
| Luis    |
| Jorge   |
| Pedro   |
| Maria   |
| Nelson  |
| Andres  |
| Claudia |
+---------+
8 rows in set (0,001 sec)
```

Filtrar Datos con WHERE

Si deseas obtener registros que cumplan ciertas condiciones, puedes usar la cláusula WHERE.

```
SELECT columna1, columna2, ... FROM nombre_tabla WHERE condicion;
```

```
MariaDB [ecoDosRuedas]> select nombre, apellidos from clientes where nombre = 'Jorge';
+--------+-----------+
| nombre | apellidos |
+--------+-----------+
| Jorge  | Rodriguez |
+--------+-----------+
1 row in set (0,001 sec)
```

Para realizar la consulta con mayor exactitud se puede hacer uso de los siguientes operadores:

- Igual =
- AND
- OR
- >
- <
- < =
- >=
- <>
- ¡=
- BETWEEN _ AND _ (Entre un valor y otro)
- LIKE (Datos que empiecen o terminen por una letra determinada)
- IN

```
MariaDB [ecoDosRuedas]> select * from clientes where alias = 'neon';
+-------+--------+-----------+--------------------+------------+------------+------------------+
| alias | nombre | apellidos | email              | celular    | contrasena | fecha_nacimiento |
+-------+--------+-----------+--------------------+------------+------------+------------------+
| neon  | Nelson | Ruiz      | neonRuiz@gmail.com | 3002145789 |   88426513 | 2002-06-24       |
+-------+--------+-----------+--------------------+------------+------------+------------------+
1 row in set (0,001 sec)
```

```
MariaDB [ecoDosRuedas]> select * from clientes where nombre = 'Ana' and apellidos = 'Diaz';
+-------+--------+-----------+--------------------+------------+------------+------------------+
| alias | nombre | apellidos | email              | celular    | contrasena | fecha_nacimiento |
+-------+--------+-----------+--------------------+------------+------------+------------------+
| diva  | Ana    | Diaz      | divaDiaz@gmail.com | 3156478294 |   10240058 | 2000-11-02       |
+-------+--------+-----------+--------------------+------------+------------+------------------+
1 row in set (0,001 sec)
```

```
MariaDB [ecoDosRuedas]> select * from clientes where fecha_nacimiento >= '1990-05-23';
+---------+--------+-----------+-------------------------+------------+------------+------------------+
| alias   | nombre | apellidos | email                   | celular    | contrasena | fecha_nacimiento |
+---------+--------+-----------+-------------------------+------------+------------+------------------+
| diva    | Ana    | Diaz      | divaDiaz@gmail.com      | 3156478294 |   10240058 | 2000-11-02       |
| dreamer | Luis   | Rojas     | dreamerRojas@gmail.com  | 3102897461 |   33025684 | 1993-12-03       |
| green   | Jorge  | Rodriguez | greenRodriguez@gmail.com| 3205467784 |   33887490 | 2003-10-30       |
| lucky   | Pedro  | Perez     | luckyperez@gmail.com    | 3102458794 |   14256133 | 1990-05-23       |
| neon    | Nelson | Ruiz      | neonRuiz@gmail.com      | 3002145789 |   88426513 | 2002-06-24       |
| ninja   | Andres | Cruz      | ninjaCruz@gmail.com     | 3128745699 |   22784511 | 1998-01-31       |
| rose    | Claudia| Mendez    | roseMendez@gmail.com    | 3162221034 |   11225468 | 1997-09-12       |
+---------+--------+-----------+-------------------------+------------+------------+------------------+
7 rows in set (0,001 sec)
```

```
MariaDB [ecoDosRuedas]> select * from clientes where fecha_nacimiento between '1990-05-23' and '2000-09-01';
+---------+---------+-----------+-----------------------+------------+------------+------------------+
| alias   | nombre  | apellidos | email                 | celular    | contrasena | fecha_nacimiento |
+---------+---------+-----------+-----------------------+------------+------------+------------------+
| dreamer | Luis    | Rojas     | dreamerRojas@gmail.com | 3102897461 |   33025684 | 1993-12-03       |
| lucky   | Pedro   | Perez     | luckyperez@gmail.com   | 3102458794 |   14256133 | 1990-05-23       |
| ninja   | Andres  | Cruz      | ninjaCruz@gmail.com    | 3128745699 |   22784511 | 1998-01-31       |
| rose    | Claudia | Mendez    | roseMendez@gmail.com   | 3162221034 |   11225468 | 1997-09-12       |
+---------+---------+-----------+-----------------------+------------+------------+------------------+
4 rows in set (0,001 sec)
```

Realizar consultas con un ordenamiento especifico

- Order by
- ASC
- DESC

```
MariaDB [ecoDosRuedas]> select * from clientes order by fecha_nacimiento asc;
+---------+---------+-----------+-------------------------+------------+------------+------------------+
| alias   | nombre  | apellidos | email                   | celular    | contrasena | fecha_nacimiento |
+---------+---------+-----------+-------------------------+------------+------------+------------------+
| malopez | Maria   | Lopez     | malopez@gmail.com       | 3221456897 |   44789465 | 1987-02-15       |
| lucky   | Pedro   | Perez     | luckyperez@gmail.com    | 3102458794 |   14256133 | 1990-05-23       |
| dreamer | Luis    | Rojas     | dreamerRojas@gmail.com  | 3102897461 |   33025684 | 1993-12-03       |
| rose    | Claudia | Mendez    | roseMendez@gmail.com    | 3162221034 |   11225468 | 1997-09-12       |
| ninja   | Andres  | Cruz      | ninjaCruz@gmail.com     | 3128745699 |   22784511 | 1998-01-31       |
| diva    | Ana     | Diaz      | divaDiaz@gmail.com      | 3156478294 |   10240058 | 2000-11-02       |
| neon    | Nelson  | Ruiz      | neonRuiz@gmail.com      | 3002145789 |   88426513 | 2002-06-24       |
| green   | Jorge   | Rodriguez | greenRodriguez@gmail.com | 3205467784 |   33887490 | 2003-10-30       |
+---------+---------+-----------+-------------------------+------------+------------+------------------+
8 rows in set (0,001 sec)
```

De Agregación

- Count (Contar registros o columnas)
- Sum (Sumar los resultados de los registros)
- Max (Mostrar el valor máximo)
- Min (Mostrar el valor mínimo)
- Avg (El valor promedio)

count

Si se deseo saber cuantas personas de la tabla clientes se llaman Sofía

```
MariaDB [ecoDosRuedas]> select count(*) as nombre from clientes where nombre = 'Andres';
+--------+
| nombre |
+--------+
|      1 |
+--------+
1 row in set (0,001 sec)
```

Con **AS** se asigna un **alias** a la columna resultante.

Consultar la suma de los precios de las bicicletas

```
MariaDB [ecoDosRuedas]> select sum(precio) as suma_precios from bicicletas;
+--------------+
| suma_precios |
+--------------+
|      7550000 |
+--------------+
1 row in set (0,001 sec)
```

Consultar el precio máximo de las bicicletas

```
MariaDB [ecoDosRuedas]> select max(precio) as precioMax from bicicletas;
+-----------+
| precioMax |
+-----------+
|   2000000 |
+-----------+
1 row in set (0,001 sec)
```

```
MariaDB [ecoDosRuedas]> select min(precio) as precioMin from bicicletas;
+-----------+
| precioMin |
+-----------+
|    950000 |
+-----------+
1 row in set (0,001 sec)
```

Agrupar Resultados

Si se desea realizar operaciones agregadas, como sumas o conteos, se puede utilizar la cláusula GROUP BY.

```
SELECT columna1, SUM(columna2) as suma_columna2 FROM nombre_tabla G
ROUP BY columna1;
```

```
MariaDB [ecoDosRuedas]> select alias, count(*) as numero_clientes from clientes group by nombre;
+---------+-----------------+
| alias   | numero_clientes |
+---------+-----------------+
| diva    |               1 |
| ninja   |               1 |
| rose    |               1 |
| green   |               1 |
| dreamer |               1 |
| malopez |               1 |
| neon    |               1 |
| lucky   |               1 |
+---------+-----------------+
8 rows in set (0,001 sec)
```

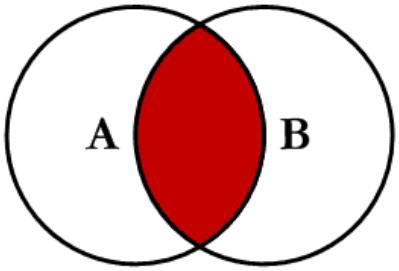
UNIONES (JOIN)

Los **Joins** o combinaciones de SQL permiten que nuestros sistemas de administración de base de datos relacionales estén bien "relacionados".

La instrucción JOIN en SQL se utiliza para combinar datos de dos o más tablas relacionales en una sola consulta. La unión de tablas se realiza mediante columnas comunes (generalmente claves primarias y foráneas) que se usan para establecer relaciones entre las tablas.

La cláusula JOIN indica qué registros deben ser seleccionados de cada tabla, según los valores de las columnas comunes.

La cláusula JOIN permite asociar dos o más tablas en base a una columna que tengan en común, capara cada fila de la primera tabla con cada fila de la segunda tabla, el resultado será la intersección entre ambas tablas.



Vamos a crear la base de datos de una institución, la cual tendrá la tabla **cursos** y la tabla **alumnos** para fines prácticos

```
MariaDB [(none)]> CREATE DATABASE eduTecnet;
Query OK, 1 row affected (0,000 sec)
```

Se crea la tabla cursos

```
MariaDB [eduTecnet]> CREATE TABLE cursos(
    -> idCurso INT AUTO_INCREMENT PRIMARY KEY,
    -> nombre_curso VARCHAR(30),
    -> creditos INT
    -> );
Query OK, 0 rows affected (0,023 sec)
```

Se insertan algunos cursos a la tabla

```
MariaDB [eduTecnet]> INSERT INTO cursos (nombre_curso, creditos) VALUES ('Ofimát
ica', 3), ('Arquitectura de PC', 3), ('Sistemas Operativos', 4);
Query OK, 3 rows affected (0,020 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Listamos los cursos creados

```
MariaDB [eduTecnet]> SELECT * FROM cursos;
+---------+---------------------+----------+
| idCurso | nombre_curso        | creditos |
+---------+---------------------+----------+
|       1 | Ofimática           |        3 |
|       2 | Arquitectura de PC  |        3 |
|       3 | Sistemas Operativos |        4 |
+---------+---------------------+----------+
3 rows in set (0,001 sec)
```

Se crea la tabla alumnos

```
MariaDB [eduTecnet]> CREATE TABLE alumnos(
    -> idAlumno INT AUTO_INCREMENT PRIMARY KEY,
    -> nombre_alumno VARCHAR(30),
    -> apellidos VARCHAR(30),
    -> idCurso INT,
    -> FOREIGN KEY (idCurso) REFERENCES cursos (idCurso)
    -> );
Query OK, 0 rows affected (0,027 sec)
```

Se crean algunos registros en la tabla alumnos

```
MariaDB [eduTecnet]> INSERT INTO alumnos (nombre_alumno, apellidos, idCurso) VALUES
 ('Martha', 'López', '2'), ('Luis Carlos', 'Cordoba', '3'), ('Ana María', 'Gómez',
'2'), ('Juan Andrés', 'Sarmiento', NULL), ('Oscar', 'Sanchez', '1');
Query OK, 5 rows affected (0,019 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Se listan los alumnos creados

```
MariaDB [eduTecnet]> SELECT * FROM alumnos;
+----------+--------------+-----------+---------+
| idAlumno | nombre_alumno | apellidos | idCurso |
+----------+--------------+-----------+---------+
|        1 | Martha        | López     |       2 |
|        2 | Luis Carlos   | Cordoba   |       3 |
|        3 | Ana María     | Gómez     |       2 |
|        4 | Juan Andrés   | Sarmiento |    NULL |
|        5 | Oscar         | Sanchez   |       1 |
+----------+--------------+-----------+---------+
5 rows in set (0,001 sec)
```

Si quisiéramos listar los alumnos y el curso al cual están matriculados, lo haríamos de la siguiente forma.

```
MariaDB [eduTecnet]> SELECT *
    -> FROM alumnos a
    -> INNER JOIN cursos c
    -> ON a.idCurso = c.idCurso;
```

```
+----------+---------------+-----------+---------+---------+---------------------+----------+
| idAlumno | nombre_alumno | apellidos | idCurso | idCurso | nombre_curso        | creditos |
+----------+---------------+-----------+---------+---------+---------------------+----------+
|        1 | Martha        | López     |       2 |       2 | Arquitectura de PC  |        3 |
|        2 | Luis Carlos   | Cordoba   |       3 |       3 | Sistemas Operativos |        4 |
|        3 | Ana María     | Gómez     |       2 |       2 | Arquitectura de PC  |        3 |
|        5 | Oscar         | Sanchez   |       1 |       1 | Ofimática           |        3 |
+----------+---------------+-----------+---------+---------+---------------------+----------+
4 rows in set (0,019 sec)
```

Se puede notar que el alumno **Juan Andrés Sarmiento** no aparece en el resultado, esto se debe a que no tiene un **id de curso** especificado. Esto se debe a que la cláusula **JOIN** muestra como resultado la **intersección** de ambas tablas.

También podemos notar que el resultado nos muestra 6 columnas en total tres de la tabla **alumnos** y tres de la tabla **cursos,** esto se debe a que en la sentencia usamos el ***** con lo cual se indica que tome todos los valores.

Podemos ser más específicos y pedirle que nos muestre solo los campos que le solicitemos.

```
MariaDB [eduTecnet]> SELECT nombre_alumno, nombre_curso FROM alumnos a INNER JOIN cursos c ON a.idCurso = c.idCurso;
```

```
+---------------+---------------------+
| nombre_alumno | nombre_curso        |
+---------------+---------------------+
| Martha        | Arquitectura de PC  |
| Luis Carlos   | Sistemas Operativos |
| Ana María     | Arquitectura de PC  |
| Oscar         | Ofimática           |
+---------------+---------------------+
4 rows in set (0,001 sec)
```

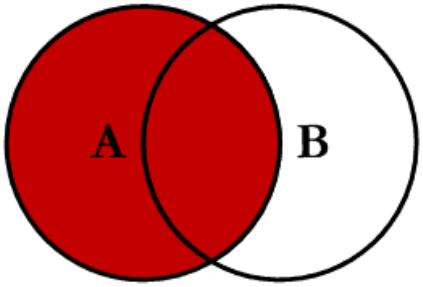
Usando LEFT JOIN

La cláusula **LEFT JOIN** permite buscar elementos en dos tablas dando prioridad a los elementos de la tabla del lado izquierdo y hace la búsqueda o la comparación con las filas de la tabla de la derecha.

Es similar al INNER JOIN pero en este caso se muestran todos los valores de la tabla de la izquierda aunque no tenga ninguna correspondencia en la tabla de la derecha.

Si el valor de las dos filas cumple la condición del JOIN, la cláusula de LEFT JOIN crea una nueva fila que contiene todas las columnas de las dos tablas incluyendo aquellas que no cumplen la condición de la unión.

Si el valor no coincide, el LEFT JOIN crea igualmente una fila que contiene todos los datos de la tabla de la izquierda y la correspondencia con la tabla de la derecha se completa con NULL.




```
MariaDB [eduTecnet]> SELECT * FROM alumnos a LEFT JOIN cursos c ON a.idCurso = c.idCurso;
```

En la consulta realizada la tabla **alumnos** sería la tabla de la izquierda por lo cual se mostrarán todas sus filas.

La tabla **cursos** sería la del lado derecho, si se encontró alguna coincidencia con las filas de la tabla alumnos se muestran esos valores, de lo contrario se mostrará como NULL.

idAlumno	nombre_alumno	apellidos	idCurso	idCurso	nombre_curso	creditos
1	Martha	López	2	2	Arquitectura de PC	3
2	Luis Carlos	Cordoba	3	3	Sistemas Operativos	4
3	Ana María	Gómez	2	2	Arquitectura de PC	3
4	Juan Andrés	Sarmiento	NULL	NULL	NULL	NULL
5	Oscar	Sanchez	1	1	Ofimática	3

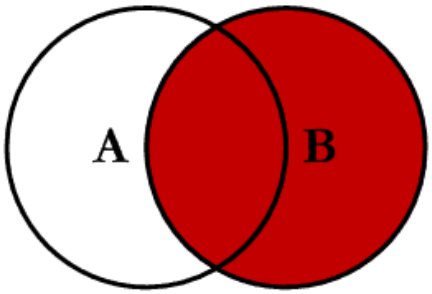
También podemos especificar las columnas que queremos visualizar para que no nos muestre todas.

```
MariaDB [eduTecnet]> SELECT nombre_alumno, apellidos, nombre_curso FROM alumnos a LEFT JOIN cursos c ON a.idCurso = c.idCurso;
```

nombre_alumno	apellidos	nombre_curso
Martha	López	Arquitectura de PC
Luis Carlos	Cordoba	Sistemas Operativos
Ana María	Gómez	Arquitectura de PC
Juan Andrés	Sarmiento	NULL
Oscar	Sanchez	Ofimática

Usando RIGHT JOIN

Es muy similar a la cláusula LEFT JOIN pero en este caso la prioridad la tiene la tabla de la derecha.




```
MariaDB [eduTecnet]> SELECT * FROM alumnos a RIGHT JOIN cursos c ON a.idCurso = c.idCurso;
```

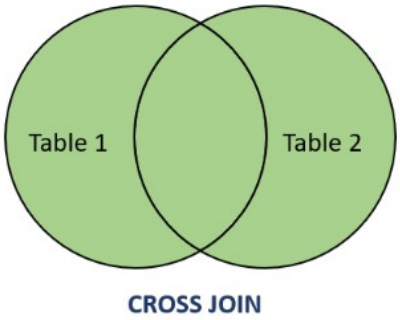
En esta consulta la tabla de la izquierda es **alumnos** y la de la derecha es **cursos.** La tabla asociada al **FROM** será siempre la tabla **LEFT.**

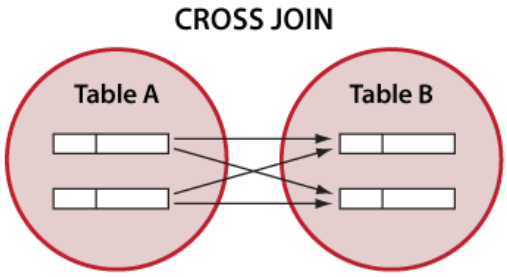
El resultado mostrará todos los cursos al menos una vez. Y si no hay ningún alumno asociado a un curso determinado, se mostrará NULL. Pero el curso aparecerá de igual forma.

idAlumno	nombre_alumno	apellidos	idCurso	idCurso	nombre_curso	creditos
1	Martha	López	2	2	Arquitectura de PC	3
2	Luis Carlos	Cordoba	3	3	Sistemas Operativos	4
3	Ana María	Gómez	2	2	Arquitectura de PC	3
5	Oscar	Sanchez	1	1	Ofimática	3

CROSS JOIN

Es un tipo especial de JOIN que devuelve el producto cartesiano entre las tablas que forman parte de la unión.





Este join toma cada fila de una tabla y la une a cada fila de otra tabla.

```
MariaDB [eduTecnet]> SELECT * FROM alumnos CROSS JOIN cursos;
```

idAlumno	nombre_alumno	apellidos	idCurso	idCurso	nombre_curso	creditos
1	Martha	López	2	1	Ofimática	3
1	Martha	López	2	2	Arquitectura de PC	3
1	Martha	López	2	3	Sistemas Operativos	4
2	Luis Carlos	Cordoba	3	1	Ofimática	3
2	Luis Carlos	Cordoba	3	2	Arquitectura de PC	3
2	Luis Carlos	Cordoba	3	3	Sistemas Operativos	4
3	Ana María	Gómez	2	1	Ofimática	3
3	Ana María	Gómez	2	2	Arquitectura de PC	3
3	Ana María	Gómez	2	3	Sistemas Operativos	4
4	Juan Andrés	Sarmiento	NULL	1	Ofimática	3
4	Juan Andrés	Sarmiento	NULL	2	Arquitectura de PC	3
4	Juan Andrés	Sarmiento	NULL	3	Sistemas Operativos	4
5	Oscar	Sanchez	1	1	Ofimática	3
5	Oscar	Sanchez	1	2	Arquitectura de PC	3
5	Oscar	Sanchez	1	3	Sistemas Operativos	4