

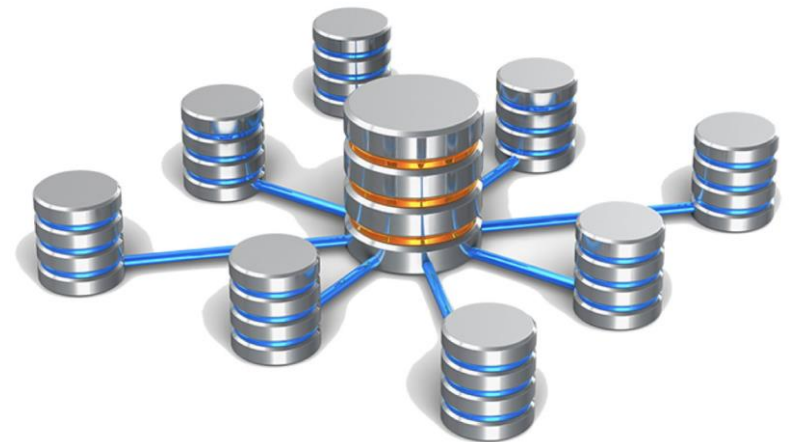
# BBDD

## Bases de Datos – Consultas Básicas de Selección



# Objetivos

- Consultar mediante SQL la información almacenada en una base de datos relacional.
- Realizar consultas simples sobre una tabla:
- Realizar consultas sobre el contenido de varias tablas.
- Realizar consultas con subconsultas.



# Lenguaje SQL

El lenguaje de programación **SQL** es el **lenguaje fundamental de los SGBD** relacionales y los elementos que lo componen son:

- a) **DML (*Data Manipulation Language*)**: es el lenguaje que consulta o manipula los datos ya existentes de nuestra BD.
- b) **DDL (*Data Definition Language*)**: permite la **definición, modificación y eliminación de las estructuras básicas** (BD, tablas, etc.) en un SGBD.
- c) **DCL (*Data Control Language*)**: administra a los usuarios de la BD, concediendo o denegando los permisos oportunos.
- d) **TCL (*Transaction Control Language*)**: lenguaje que controla el procesamiento de las transacciones de la BD.

# DML (Data Manipulation Language)


**DML** (Data Manipulation Language) o Lenguaje de Manipulación de Datos es la parte de SQL dedicada a la manipulación de los datos.

Las sentencias DML son las siguientes:

- **SELECT**: se utiliza para realizar consultas y extraer información de la base de datos.
- **INSERT**: se utiliza para insertar registros en las tablas de la base de datos.
- **UPDATE**: se utiliza para actualizar los registros de una tabla.
- **DELETE**: se utiliza para eliminar registros de una tabla.

En esta Unidad nos centraremos en el uso de la sentencia **SELECT**.

# SELECT (Consultas básicas de info)

- De forma introductoria, podemos decir que la sentencia **SELECT** nos permite recuperar las **columnas** que necesitemos de los registros de una o varias tablas.
- Además, también nos permite recuperar **valores calculados** (con operadores aritméticos: % \* + - / ) o **visualizar mensajes por pantalla** sin referenciar ninguna tabla.
- El **orden** de las cláusulas de la sentencia SELECT es importante.
- Para seleccionar todas las columnas, podemos usar \* (asterisco)
- Las **ambigüedades** (mismo nombre de columna entre distintas tablas) las podemos resolver anteponiendo el nombre de la tabla. 
- Podemos utilizar **alias** sobre columnas para mostrar otro nombre distinto de una columna o cálculo realizado.
- Podemos **evitar valores repetidos** de una determinada columna (DISTINCT).
- Podemos **ordenar los registros** recuperados en la consulta (ORDER BY).
- Podemos **limitar** el número de registros a recuperar.

*Orden de select* 

```
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr] ...
[into_option]
[FROM table_references
  [PARTITION partition_list]]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
[HAVING where_condition]
[WINDOW window_name AS (window_spec)
  [, window_name AS (window_spec)] ...]
[ORDER BY {col_name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[into_option]
[FOR {UPDATE | SHARE}
  [OF tbl_name [, tbl_name] ...]
  [NOWAIT | SKIP LOCKED]
  | LOCK IN SHARE MODE]
[into_option]

into_option: {
  INTO OUTFILE 'file_name'
    [CHARACTER SET charset_name]
    export_options
  | INTO DUMPFILE 'file_name'
  | INTO var_name [, var_name] ...
}
```

<https://dev.mysql.com/doc/refman/8.0/en/select.html>



# SELECT (Consultas básicas de info)

- La cláusula **FROM** de la sentencia SELECT determina las tablas sobre las que realizar la consulta.
- Las tablas también pueden tener **Alias** para acortar las referencias en nuestras sentencias SELECT.  
*table → renta*
- Tenemos dos **formas de referenciar una tabla** en una consulta: habiendo seleccionado la BD contenedora previamente (USE) o indicando el nombre de la BD seguido de un punto y del nombre de la tabla.
- La cláusula **WHERE** de la sentencia SELECT indica las condiciones que un registro debe cumplir para ser seleccionado:
- Operadores de comparación ( = > >= <= < <> != )
- Operadores de comparación de cadenas de texto (LIKE) junto con los comodines % y \_
- Evaluar valores nulos o no nulos (IS NULL / IS NOT NULL) o que estén dentro de un determinado conjunto ( IN (a,b,c) ) o que estén dentro de un determinado rango de valores o fechas ( BETWEEN(x AND y) )
- Operadores lógicos ( AND, &&, OR, ||, NOT, !)

*consulta nueva*

```
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr] ...
[into_option]
[FROM table_references
    [PARTITION partition_list]]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
[HAVING where_condition]
[WINDOW window_name AS (window_spec)
    [, window_name AS (window_spec)] ...]
[ORDER BY {col_name | expr | position}
    [ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[into_option]
[FOR {UPDATE | SHARE}
    [OF tbl_name [, tbl_name] ...]
    [NOWAIT | SKIP LOCKED]
    | LOCK IN SHARE MODE]
[into_option]

into_option: {
    INTO OUTFILE 'file_name'
        [CHARACTER SET charset_name]
        export_options
    | INTO DUMPFILE 'file_name'
    | INTO var_name [, var_name] ...
}
```

<https://dev.mysql.com/doc/refman/8.0/en/select.html>



# SELECT (Consultas básicas de info)

- El resultado de una consulta SELECT es siempre una **tabla de datos** con una o varias columnas y cero, una o varias filas.
- Podemos crear subconsultas que son consultas **SELECT** anidadas, una dentro de otra, en la cual, la consulta principal depende de las que están dentro de la consulta anidada (se ejecutan de adentro hacia afuera):

- **WHERE** col = (SELECT ...);
- **WHERE** col IN (SELECT ...);
- **WHERE** EXISTS (SELECT);
- **WHERE** col > ANY | ALL (SELECT);

- Usamos subconsultas cuando necesitamos incluir en la cláusula **WHERE** criterios de selección que solamente existen en otra tabla generalmente.
- El operador **"EXISTS"** se emplea en las subconsultas correlacionadas para restringir el resultado de una consulta exterior a los registros que cumplen la subconsulta (consulta interior). Estos operadores retornan **"true"** (si las subconsultas **retornan registros**) o **"false"** (si las subconsultas **no retornan registros**).

```
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr] ...
[into_option]
[FROM table_references
  [PARTITION partition_list]]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
[HAVING where_condition]
[WINDOW window_name AS (window_spec)
  [, window_name AS (window_spec)] ...]
[ORDER BY {col_name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[into_option]
[FOR {UPDATE | SHARE}
  [OF tbl_name [, tbl_name] ...]
  [NOWAIT | SKIP LOCKED]
  | LOCK IN SHARE MODE]
[into_option]

into_option: {
  INTO OUTFILE 'file_name'
    [CHARACTER SET charset_name]
    export_options
  | INTO DUMPFILE 'file_name'
  | INTO var_name [, var_name] ...
}
```

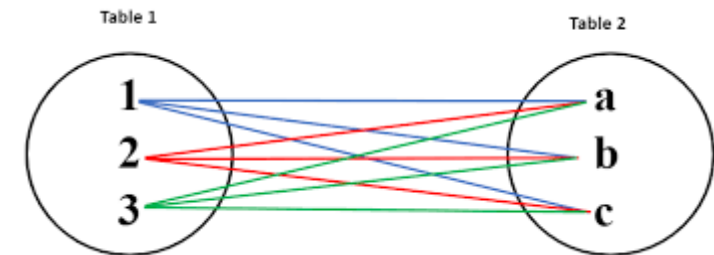
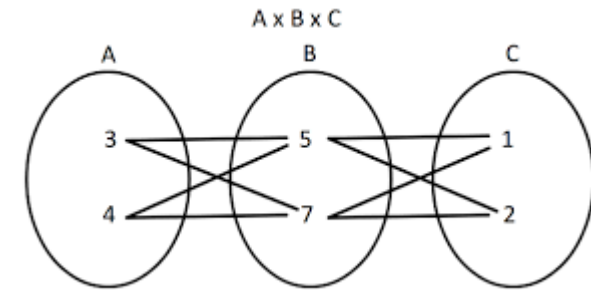
<https://dev.mysql.com/doc/refman/8.0/en/select.html>





# SELECT (Consultas multitabla)

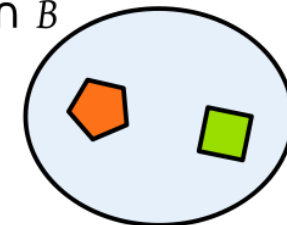
- Una consulta multitabla es aquella que recupera información de más de una tabla, generalmente uniéndolas por campos relacionados (*join*).
- En este tipo de consultas, en la cláusula FROM encontramos un listado de tablas separadas por comas.
- El resultado de una consulta multitabla en el que no se establezca la relación entre campos de las tablas involucradas, nos devolverá el **producto cartesiano** (todas las combinaciones de las filas de una tabla con las del resto de tablas).
- En cambio si establecemos una condición en la cláusula WHERE (**filtro del producto cartesiano**), de entre todas las combinaciones, podríamos recuperar únicamente aquellas filas en las que se comparta un mismo valor en un campo relacionado (**intersección**).
- Ese campo relacionado entre dos tablas no es otro que la clave primaria (PK) junto con sus claves foráneas (FK) migradas.
- Podremos ir enlazando tablas dentro de una misma consulta con el operador lógico AND.
- Si los campos relacionados tienen el mismo nombre, tendremos que identificar a qué tabla pertenecen cada uno para deshacer la ambigüedad.



$A = \{ \text{pentagon}, \text{diamond}, \text{square}, \text{rectangle} \}$

$B = \{ \text{star}, \text{square}, \text{triangle}, \text{pentagon} \}$

$A \cap B$



**JOIN = PRODUCTO CARTESIANO + FILTRO**

(Existen diferentes tipos de operaciones JOIN que veremos más adelante en la siguiente UD)



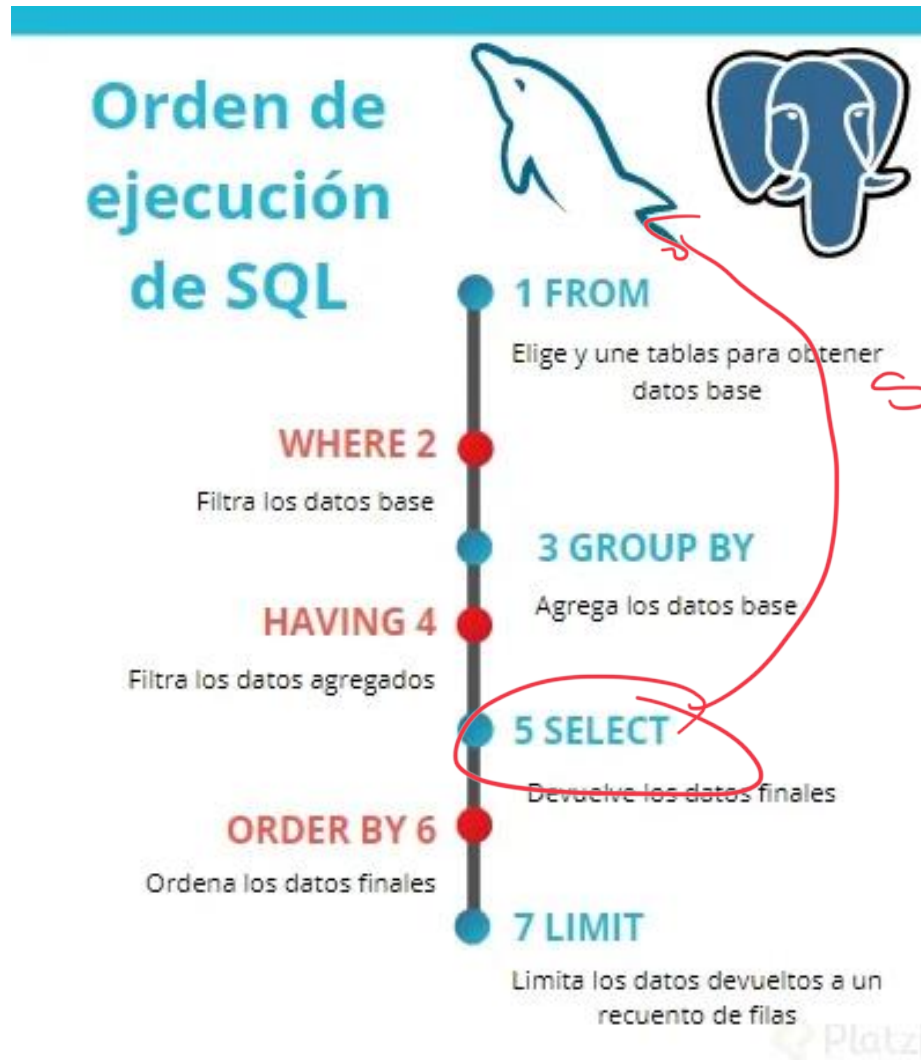


# SELECT (Consultas RESUMEN)

- Existen también consultas que resumen cierta información obteniendo datos calculados de varios registros en una tabla.
  - **SUM** (nomColumna): suma los valores
  - **AVG** (nomColumna): obtiene la media
  - **MIN** (nomColumna): obtiene el valor mínimo
  - **MAX** (nomColumna): obtiene el valor máximo
  - **COUNT** (nomColumna): cuenta el número de valores (no nulos)
  - **COUNT (\*)**: cuenta el número de valores de una fila (incluyendo nulos)
- Con las consultas resumen también podemos realizar agrupaciones de registros que comparten un mismo valor en una o varias columnas (**GROUP BY**).
- Cuando queremos filtrar los resultados calculados mediante agrupaciones, **NO** podemos hacerlo en la clausula **WHERE** puesto que esta actúa antes de la agrupación de los registros (**HAVING**).
- Las funciones de agregación listadas previamente solo pueden ser usadas en las clausulas **SELECT** y **HAVING**, nunca en la clausula **WHERE**.

# SELECT (Orden de las cláusulas)

Orden en el que **se ejecuta cada una de las cláusulas** que forman la sentencia SELECT.



*se suele poner ahí  
orden de las consultas*