

SET sql_mode = "ONLY_FULL_GROUP_BY"; poner al principio y ejecutar con rayo

SELECT indico lo que quiero que traiga la tabla poniendo el nombre de la columna
* son todas las columnas
si son varias separo con coma (,)
acá también puedo poner funciones de agregación: count(), max(), min(), avg(), sum() y dentro de los () el campo a contar, sumar, etc.
con AS y un "XX" entre comillas cambiar el título que muestra la columna por XX

FROM el origen de datos, de qué columna lo traigo (siempre es una sola)

WHERE condición a cumplir
pueden ser varias con AND y OR
formato fecha: 1969/07/21
ej. IN => WHERE id IN (15, 19, 21, 35)
si es texto lo que debe buscar va entre "" (where país = "Brazil")
para LIKE uso "_s_" (tres letras la del medio s), "%s%" (contiene s), "s%" (empieza con s)

Operadores

=	→ Igual a	IS NULL	→ Es nulo
>	→ Mayor que	BETWEEN	→ Entre dos valores
>=	→ Mayor o igual que	IN	→ Lista de valores
<	→ Menor que	LIKE	→ Se ajusta a...
<=	→ Menor o igual que		
<>	→ Diferente a		
!=	→ Diferente a		

GROUP BY columna de agrupación. Agrupo los datos de la tabla según un criterio (categoría, género, etc). *Devuelve una sola fila por grupo.*

HAVING condición que debe cumplir el grupo seleccionado (null, más de 1, menos de 5, etc)

ORDER BY qué columna ordena la tabla resultado. Y defino si desc ó asc
Por default es ascendente (de menor a mayor). Con fechas de más antiguo a más reciente.

LIMIT define el número de filas a mostrar. (Limit = 3 muestra sólo los tres primeros)
OFFSET no muestra un número de filas. (OFFSET 5 = no muestra, saltea, las primeras 5).

`INSERT INTO` tabla (id, apellido, nombre, pasaporte, fechaNac) `VALUES` (`DEFAULT`, "Perez",
"Pepe",...)

`DELETE FROM` nombreTabla `WHERE` servicio_basico_id =5

Siempre usar WHERE o borra toda la tabla

Si el campo está también en una tabla relacionada hay que hacer un delete de cada tabla.

```
DELETE FROM habitacion_x_servicio_basico WHERE servicio_basico_id = 5;  
DELETE FROM servicio_basico WHERE id = 5;
```

`DISTINCT` dentro de `SELECT` elimina la duplicación de filas que sean exactamente iguales

VARIAS TABLAS (*mejor usar JOIN*)

Primero me armo el diagrama relacional de la tabla haciendo Database/Reverse Engineer ó (Ctrl-R). siempre Next y me queda en EER el diagrama. Luego tocando en la línea que une las tablas me muestra las relaciones de claves foráneas.

Puedo traer info de más de una tabla:

- Seleccionándolas en `FROM`
- Y generando la igualdad de la clave foráneas con `WHERE`
- Será un `WHERE` si es uno a muchos
- Si es muchos a muchos tengo que traer la info de las 3 tablas (las 2 + la pivot) y 2 `WHERE`

```
SELECT movies.id, title, genre_id, genres.id, name  
FROM movies, genres  
WHERE genre_id = genres.id
```

```
SELECT title, first_name, last_name  
FROM movies, actor_movie, actors  
WHERE movie_id = movies.id  
AND actor_id = actors.id
```

Renombrar tablas y llamar columnas

- movies.id => llama en la tabla movies a la columna id
- en `SELECT` ó `FROM` llamo a la tabla y le doy un alias con un espacio y el nuevo nombre (ej:
SELECT facturas f => luego cuando diga f traerá la tabla facturas)

Luego la uso con punto ej.: f.total será la columna **total** de la tabla f, es decir, tabla **facturas**

JOIN

SELECT nombres columnas **FROM** tabla (una sola) **JOIN** la otra tabla **ON** armo igualdad de claves

Para la igualdad de claves veo el Diagrama e igualo las claves de cada tabla que necesito:

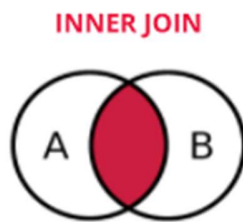
Ejemplo : c.categorialD = p.CategorialD

Igualo categorialD de tabla categoria (c) con CategorialD de tabla producto(p)

INNER JOIN trae SOLO los registros donde se cruzan tablas (registros que tiene datos en ambas tablas simultáneamente.) No trae Marta ni factura 12.

```
SELECT factura.id AS nro_factura, apellido, nombre, fecha
FROM cliente
INNER JOIN factura
ON cliente.id = factura.cliente_id;
```

cliente		
id	nombre	apellido
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García



factura		
id	cliente_id	fecha
11	2	12/09/2019
12	null	20/09/2019
13	1	24/09/2019

nro_factura	apellido	nombre	fecha
11	Sanchez	Clara	12/09/2019
13	Perez	Juan	24/09/2019

RIGHT JOIN trae TODOS los registros con datos en la tabla de la derecha (la que está después del JOIN), aunque no haya registros vinculados de la otra tabla. De la tabla izquierda solo los vinculados (no trae Marta)

```
SELECT factura.id AS nro_factura, apellido, nombre, fecha
FROM cliente
RIGHT JOIN factura
ON cliente.id = factura.cliente_id;
```

cliente		
id	nombre	apellido
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

RIGHT JOIN



factura		
id	cliente_id	fecha
11	2	12/09/2019
12	null	20/09/2019
13	1	24/09/2019

nro_factura	apellido	nombre	fecha
11	Sanchez	Clara	12/09/2019
12	null	null	20/09/2019
13	Perez	Juan	24/09/2019

LEFT JOIN trae TODOS los registros con datos en la tabla de la izquierda (la que está antes del JOIN), aunque no haya registros vinculados de la otra tabla. De la tabla derecha sólo los vinculados (no viene factura 12)

```
SELECT factura.id AS nro_factura, apellido, nombre, fecha
FROM cliente
LEFT JOIN factura
ON cliente.id = factura.cliente_id;
```

cliente		
id	nombre	apellido
1	Juan	Perez
2	Clara	Sanchez
3	Marta	García

LEFT JOIN



factura		
id	cliente_id	fecha
11	2	12/09/2019
12	null	20/09/2019
13	1	24/09/2019

nro_factura	apellido	nombre	fecha
11	Sanchez	Clara	12/09/2019
13	Perez	Juan	24/09/2019
null	García	Marta	null

Si tengo más de una tabla uso un **JOIN** por cada par de tablas y genero un **ON** con un par de igualdades por cada JOIN

FUNCIONES DE ALTERACION (ver mas en archivo, se pueden agrupar)

Permiten realizar operaciones con los datos que recibo. Van en **SELECT, WHERE , ORDER BY**

- CONCAT** une datos de dos columnas en una (SELECT CONCAT(nombre," ",apellido)
- COALESCE** si no tiene un dato completa con lo que pido (SELECT title, COALESCE(name, "no tiene genero")
- NOW** me da la fecha y hora actual.
- DATEDIFF** diferencia de fechas DATEDIFF (NOW(), release_date)
- EXTRACT** toma una parte del campo EXTRACT(day FROM reléase:Date)
- DATE FORMAT** cambio la forma de verlo DATE FORMAT (reléase_date. "%d/%m/%Y")
- REPLACE** cambia algo del campo REPLACE (title, "Harry","Juancito") cambia Harry por Juan en title
- CASE WHEN** agrego o cambio algo en base a condiciones, indico con **THEN** lo que voy a dar como resultado. **ELSE** para lo que no cumple las condiciones dadas. Cierro con **END AS** y le doy título a esa columna (de lo contrario sale todo el case)

```
SELECT id, titulo, rating,
CASE
  WHEN rating < 4 THEN 'Mala'
  WHEN rating BETWEEN 4 AND 6 THEN 'Regular'
  WHEN rating BETWEEN 7 AND 9 THEN 'Buena'
  ELSE 'Excelente'
END AS calificacion
FROM pelicula;
```

VISTAS

Son consultas predefinidas para poder repetirlas fácilmente.

Mejor hacer el script del select primero y si funciona la convierto en vista.

- CREATE VIEW** nombre_de_la_vista AS SELECT Para crearla.
- ALTER VIEW** nombre_de_la_vista AS SELECT Para modificarla
- DROP VIEW** nombre_de_la_vista Para eliminarla
- SELECT * FROM** vista_coche; Para invocarla (llamarla)
- SELECT * FROM** vista_coche **WHERE** ...; Para invocarla (llamarla) y agregarle condiciones, etc.