

ALL- comparación con todo lo que viene desde del paréntesis

IN (IMPORTANTE + UTILIZADO) - comprueba que un valor este dentro de un conjunto (dentro de la subconsulta) - pertenencia a un conjunto

EXIST - devuelve el registro que la consulta padre allá encontrado siempre y cuando lo que vaya después del existe

ANY - Para hacer una comparación

ALIAS:

un alias un nombre alternativo que se le da a una tabla o un campo, por comodidad(nombre o largo o que no se entiende bien) o porque a veces es inevitable(cuando tenemos que relacionar una tabla consigo misma me permitirá hacer referencia a la primera instancia o la segunda)

Cómo declarar un alias en el caso de los campos o tablas

campo (comodidad)

```
(SELECT name AS n, birth_date AS bd FROM drivers;)
```

tabla (necesidad)

```
(SELECT name, birth_date FROM drivers AS drv;)
```

Declarar un alias para un campo no permite usarlo dentro de la cláusula WHERE per si una tabla.

(si lo hacemos en una subconsulta después del SELECT \* FROM podremos usar los alias de los campos en la cláusula WHERE. Ej:

```
SELECT * FROM
```

```
(
```

```
    SELECT name AS n, birth_date AS bd
```

```
    FROM drives
```

```
)
```

```
WHERE bd > '200-01-01';
```

repasar diapositiva 33

BUSCAR TEXTO (patrones):

LIKE o NOT LIKE: se usa con dos comodines % o \_

?: se sustituye por una cadena que tenga 0 (nada de contenido) o más características.  
(LIKE 'A%')

!Las mayúsculas y las minúsculas influyen;  
convertir cadena a mayúscula: UPPER ('hola')  
convertir cadena a minúscula: LOWER(HOLA)

\_ : se sustituye por una cadena que tenga 1, ni más ni menos;  
(LIKE ' \_ \_ \_ \_')

OPERADORES DE AGREGACIÓN:  
su función es realizar cálculos

count: sirven para contar (da igual los registros) -> SELECT count(\*) FROM drivers; o  
SELECT count(id) FROM drivers WHERE name LIKE '%García%';

sum: sirve para sumar (coge un campo **numérico**)

avg: calcula la media aritmética (coge un campo **numérico**)

min: calcula el mínimo, te da el dato mínimo (cualquier campo)

max: calcula el máximo, te da el dato máximo (cualquier campo)

having: usado para verificar condiciones sobre las anteriores, para hacer comparaciones,  
va siempre antes del operador ( HAVING AVG(salary))

**group by:** lista de atributos a la que le quiero hacer la consulta, y lo mete todo en un grupo ( si hay 7 películas de un solo director, si los agrupo por el criterio del director y hago una consulta de películas solo me saldrá una vez el director, no 7 veces)

SELECT number\_plate, first\_name, last\_name (matrícula, nombre y apellido)  
FROM drivers INNER JOIN vehicles (de conductores con conexión a la tabla vehículos)  
ON drivers.id = vehicles.driver\_id (comprueba que el id de los conductores y de los  
vehículos sean iguales)

WHERE license\_date >= '2008-01-01' (Fecha después del 2008)  
GROUP BY last\_name, first\_name; (Agrupa los datos en el orden que le escribimos)

order by: Ordena los resultados que devuelve el programa (podremos poner el orden de  
forma ascendente ASC o descendente DESC)