

**SUBCONSULTAS**

# SUBCONSULTAS

- Podemos utilizar una consulta dentro de otra consulta
- El resultado de la consulta más “interior” se sustituye en la consulta más “exterior”

[illegible]

# SUBCONSULTAS

- Hay que fijarse bien en que devuelve la subconsulta
  - Si es un único valor, se puede comparar con =, <, >, >=, <=
  - Si devuelve varios valores, se compara con IN

```
SELECT matricula,marca,modelo
FROM automoviles
WHERE matricula IN (SELECT matricula
                    FROM contratos
                    WHERE fini>='2018-01-01');
```

# CUANTIFICADOR ALL

- El **cuantificador ALL** permite seleccionar las filas que cumplan con una determinada condición respecto de todos los valores devueltos por la subconsulta.

```
SELECT marca FROM automoviles
where marca <> ALL (SELECT DISTINCT marca
                    FROM contratos
                    INNER JOIN automoviles USING (matricula)
                    WHERE year(fini)=2018);
```

# CUANTIFICADOR ANY

- El **cuantificador ANY** permite seleccionar las filas que cumplan con una determinada condición para al menos uno de los valores devueltos por la subconsulta.

```
SELECT * FROM automoviles
WHERE precio < ANY (SELECT precio
                    FROM automoviles
                    WHERE marca='seat');
```

# UNION

- Se utiliza para combinar los resultados de dos consultas
- Tienen que devolver los mismos campos y llamarse igual
- Si no se llaman igual, entonces utilizamos alias
- UNION ALL, hace lo mismo pero no quita los repetidos.

```
SELECT matricula FROM contratos  
WHERE ffin IS NULL  
  
UNION  
  
SELECT matricula FROM automoviles WHERE marca="Renault";
```

**CORRECCION HOJA 13**

# EXISTS Y NOT EXISTS

- 13.- Devuelve los nombres de los fabricantes que tienen productos asociados. (Utilizando EXISTS o NOT EXISTS).

```
SELECT nombre
```

```
FROM fabricante
```

```
WHERE EXISTS (SELECT *
```

```
                FROM producto
```

```
                WHERE producto.id_fabricante=fabricante.id);
```

- Dentro de la subconsulta se utiliza un campo de la tabla de la consulta principal.



# EXISTS Y NOT EXISTS

14.- Devuelve los nombres de los fabricantes que no tienen productos asociados. (Utilizando EXISTS o NOT EXISTS).

```
SELECT nombre
```

```
FROM fabricante
```

```
WHERE NOT EXISTS (SELECT *
```

```
FROM producto
```

```
WHERE producto.id_fabricante=fabricante.id);
```

# CONSULTAS CORRELACIONADAS

- 15.- Lista el nombre de cada fabricante con el nombre y el precio de su producto más caro.

```
SELECT f.id, f.nombre, p.precio
FROM fabricante f
INNER JOIN producto p ON p.id_fabricante=f.id
WHERE p.precio=(SELECT max(p2.precio)
                FROM producto p2
                WHERE p2.id_fabricante=f.id);
```

- Dentro de la subconsulta, busca el máximo de precio de CADA fabricante de la consulta principal.

# CONSULTAS CORRELACIONADAS

- 16.- Devuelve un listado de todos los productos que tienen un precio mayor o igual a la media de todos los productos de su mismo fabricante.

```
SELECT p.*  
FROM producto p  
INNER JOIN fabricante f ON p.id_fabricante=f.id  
WHERE p.precio>=(SELECT avg(p2.precio)  
                  FROM producto p2  
                  WHERE p2.id_fabricante=f.id);
```

Misma jugada que antes, preguntas dentro de la consulta por un campo de una tabla exterior, para sacar la media de los productos de CADA fabricante.