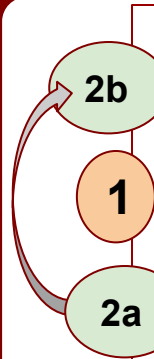


Guía rápida de queries SQL

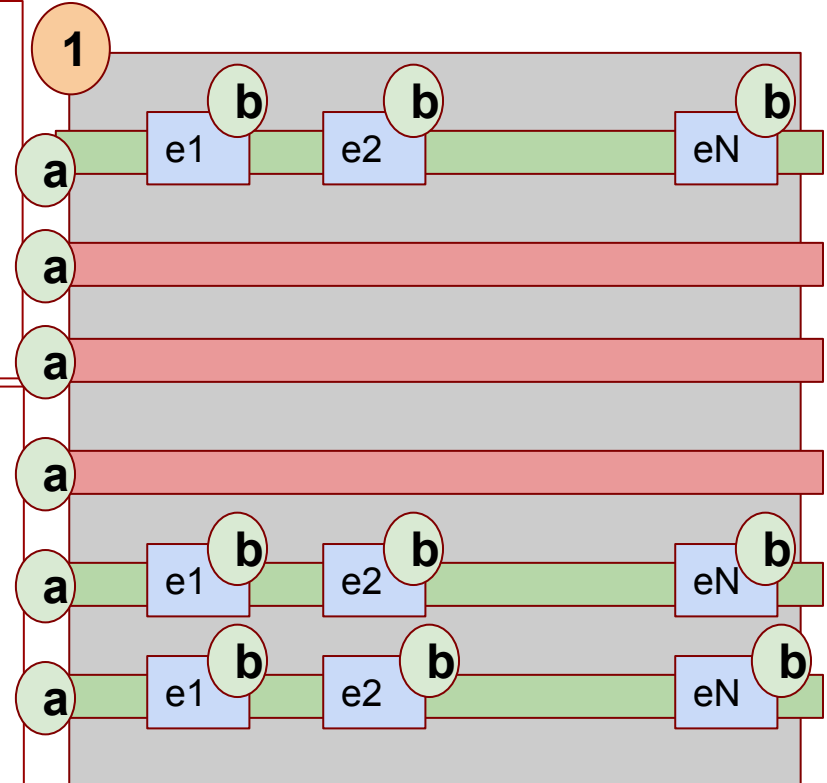
Realizado por A.Garay (dpto.de Informática)

Select básico

SELECT e_1, e_2, \dots, e_N
FROM t_1
WHERE condición



1. Se abre la tabla "T" como tabla de trabajo y se selecciona su primera fila.
2. ¿Se cumple la "condición" en la fila actual?
 - a. TRUE: Se computan y seleccionan las expresiones e_1, e_2, \dots, e_N
3. Se selecciona la siguiente fila (si es que falta alguna) y se vuelve al punto 2



FROM

Contenido	Ejemplo	Explicación
T	persona	Se trabaja sobre la tabla T (que debe existir en la Base de Datos)
T A	persona p	Se trabaja sobre la tabla T, y se le otorga el alias (o sobrenombre) A (útil después para la cláusula SELECT)
T1,T2	persona,coche	Se crea una tabla provisional llamada T1xT2 cuyas filas se crean a partir de la combinación de cada fila de T1 con cada fila de T2. Esta operación es conocida en el álgebra relacional como el producto cartesiano de las tablas T1 y T2 Una vez finalizada la consulta ¡¡se destruye la tabla T1xT2!!
T1 A1, T2, T3 A3	persona p, coche, aficion a	Se puede combinar el producto cartesiano con el establecimiento de alias, como se quiera

SELECT

Contenido	Ejemplo	Explicación
C1	apellido	Se selecciona el valor de la columna C1 <u>para la fila actualmente en proceso</u> C1 debe existir la tabla de trabajo del FROM, y su nombre debe ser único en esa tabla
T1.C1 A1.C1	persona.apellido p.apellido	Se selecciona la columna C1 de T1 o bien la columna C1 de la tabla cuyo alias es A1. Sólo tiene sentido si C1 existe en T1. Es la única forma de referirse a una columna cuyo nombre coincida con la de otra tabla y se haya hecho un producto cartesiano entre ambas
F_SIMPLE(C1)	saldo + 100 abs(saldo) abs(saldo+100)	Se aplica F_SIMPLE al valor concreto de la columna o expresión. Una función se puede aplicar a un nombre de columna, una constante o a otra expresión
F_AGR(C1)	count(*) count(nombre2) avg(saldo)	Se aplica la función agregada F sobre la columna C1 <u>después de haber procesado todas las filas de la tabla del FROM</u> . El asterisco entre paréntesis significa que se aplica la función a <u>todas</u> las filas. mientras que un nombre de columna implica que sólo a aquellas que no sean NULL en esa columna

WHERE

- Expresión booleana simple
 - constante: `true`, `false`
 - comparaciones: `2>3`, `3=4`
 - comparaciones que incluyan nombres de columnas: `saldo>100` o bien `nombre='pepe'`
 - **Los nombres de columnas se resolverán al valor concreto que tenga la fila actualmente en proceso para esa columna**
 - funciones simples, comparaciones mezcladas: `saldo+10 = abs(precio)`
- Expresiones booleanas simples conectadas con AND, OR, NOT:
`saldo>100 and not(nombre = 'pepe')`

Subconsultas (en WHERE)

- Sólo funcionan si la SELECT interna devuelve un único valor.

```
SELECT nombre  
FROM persona  
WHERE estatura >  
    (  
    SELECT estatura  
    FROM atleta  
    WHERE dni = 77  
    )
```

Subconsultas con cláusulas IN (en WHERE)

- Operación de pertenencia a un conjunto \in , sólo funciona si lo que se devuelve en el select interno tiene el mismo número de componentes (por cada elemento) que lo comparado (y de tipos compatibles)

```
SELECT nombre
FROM persona
WHERE pais IN
(
  SELECT nombre
  FROM pais
  WHERE
    poblacion>20000000
)
```

```
SELECT dni
FROM persona
WHERE (nombre,apellido) IN
(
  SELECT nombre,apellido
  FROM atleta
  WHERE
    estatura>180
)
```

Uso de ALL y ANY con subconsultas

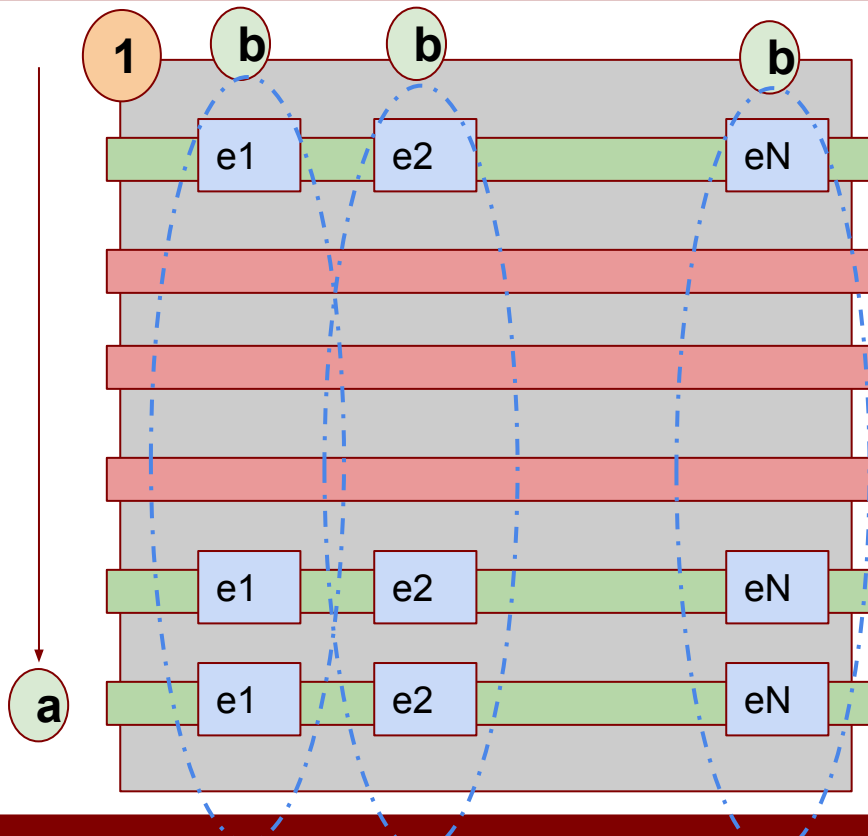
- Estas cláusulas permiten comparar un elemento (presente en una cláusula WHERE), con todo un conjunto (proveniente de una subconsulta)

```
SELECT nombre  
FROM persona  
WHERE sueldo > ALL  
    (  
    SELECT sueldo  
    FROM persona  
    WHERE  
        pais = 'España'  
    )
```

```
SELECT nombre  
FROM persona  
WHERE sueldo > ANY  
    (  
    SELECT sueldo  
    FROM persona  
    WHERE  
        pais = 'España'  
    )
```


Funciones de agregación

- Siempre devuelven una única fila
- Es importante entender que primero se procesan TODAS las filas de la tabla. (no se hace un SELECT en cada “pasada” por cada fila, sólo uno al final del todo)
- En la sentencia SELECT sólo pueden aparecer funciones de agregación o funciones que operen entre ellas.
 - Nunca pueden aparecer nombres de columnas sueltos o funciones que operen entre ellos



Funciones de agregación (ejemplos)

OK

```
SELECT avg(saldo)
FROM persona
```

```
SELECT avg(saldo), sum(saldo)
FROM persona
```

```
SELECT avg(saldo) / sum(saldo)
FROM persona
```

MAL (conceptualmente)

```
SELECT avg(saldo), saldo
FROM persona
```

```
SELECT avg(saldo) / saldo
FROM persona
```

Claúsula de agrupación GROUP BY

- Las funciones de agregación se aplican de forma indiscriminada sobre TODO el conjunto de filas seleccionadas por la cláusula WHERE
- Si queremos aplicarlas a grupos más pequeños, podemos hacerlo utilizando la cláusula GROUP BY.
- Téngase en cuenta que antes de aplicar GROUP BY, el motor de consultas aplicará el filtro de la cláusula WHERE
- En el SELECT sólo pueden aparecer funciones de agregación (con las mismas reglas explicadas antes) o bien el nombre de columnas que se hayan utilizado para agrupar en el "GROUP BY"

4 **SELECT e1,e2,...,eN**

1 **FROM t1**

2 **WHERE condición**

3 **GROUP BY c1,c2,...,cN**

GROUP BY (ejemplos)

OK

```
SELECT avg(saldo)
FROM persona
GROUP BY departamento
```

```
SELECT departamento, avg(saldo)
FROM persona
GROUP BY departamento
```

```
SELECT nom, ape, avg(saldo)
FROM persona
GROUP BY nom,ape
```

MAL (conceptualmente)

```
SELECT saldo
FROM persona
GROUP BY departamento
```

```
SELECT saldo, avg(saldo)
FROM persona
GROUP BY departamento
```

```
SELECT nom, saldo, avg(saldo)
FROM persona
GROUP BY nom,ape
```

Cláusula HAVING (el “where” de los group-by)

- Las funciones de agregación se aplican de forma indiscriminada sobre TODO el conjunto de filas seleccionadas por la cláusula WHERE
- Si queremos aplicarlas por grupos más pequeños, podemos hacerlo utilizando la cláusula GROUP BY.
- Téngase en cuenta que antes de aplicar GROUP BY, el motor de consultas aplicará la cláusula WHERE
- En el SELECT y en el HAVING sólo pueden aparecer funciones de agregación (con las mismas reglas explicadas anteriormente) o el nombre de las columnas que se hayan seleccionado en el “GROUP BY”

5 SELECT e1,e2,...,eN

1 FROM t1

2 WHERE condición

3 GROUP BY c1,c2,...,cN

4 HAVING condiciones

Estrategia general de consulta

- Observar en qué tabla/s están los datos finales que debe devolver la cláusula **SELECT**
 - Si están desperdigados por varias tablas tendremos que hacer un JOIN
 - Si están en una única tabla, utilizaremos siempre cláusulas IN (que son más eficientes), aunque los criterios WHERE estén desperdigados

Mejor “IN” que producto cartesiano

MEJOR

```
SELECT nombre, ape1
FROM persona p
WHERE dni IN
    (
        SELECT dni
        FROM coche
        WHERE
            dni_prop = p.dni
            AND mat = 7
    )
```

PEOR

```
SELECT p.nombre, p.ape1
FROM persona p, coche c
WHERE
    p.dni = c.dni_prop AND
    c.mat = 7
```

ÚNICA FORMA

```
SELECT p.nombre, c.marca
FROM persona p, coche c
WHERE
    p.dni = c.dni_prop AND
    c.mat = 7
```