

SQL (Structured Query Language)

- Introducció
 - El Model Relacional
 - Sentències SQL
 - Creació d'una taula
 - Inserció / modificació / esborrat de files d'una taula
 - Consultes
-

Introducció

- Llenguatge estructurat de definició, actualització i consulta de bases de dades
 - Proposat per un departament d'investigació d'IBM
 - Adoptat com a estandard per al Model Relacional de bases de dades en els anys 1986-87 (ANSI/X3H2/RDL).
 - Des d'aleshores se n'han fet varies versions: SQL/89, SQL/92, SQL:1999 i SQL-2003.
 - Malgrat l'estandarització, els diferents constructors de SGBDR (Sistemes de Gestió de Bases de Dades Relacionals) ofereixen variants de SQL.
 - Nosaltres veurem el nucli comú de SQL per a tots els SGBDR, tenint en compte que per alguns la sintaxi de les sentències pot variar.
 - Pot ser utilitzat de manera interactiva o de manera hostatjada entre les sentències d'un programa.
-

Model Relacional

- Una base de dades relacional està composta de **Taules (Relacions)** amb un conjunt de **Columnes (Atributs)** i un conjunt **Files (Tuples)**.

departaments(num_dpt, nom_dpt, planta, edifici, ciutat_dpt)					
	1	DIRECCIO	10	PAU CLARIS	BARCELONA
	2	DIRECCIO	8	RIOS ROSAS	MADRID
	3	MARQUETING	1	PAU CLARIS	BARCELONA
projectes(num_proj, nom_proj, producte, pressupost)					
	1	IBDTEL	TELEVISIO	1000000	
	2	IBDVID	VIDEO	500000	
empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)					
	1	CARME	400000	MATARO	1 1
	2	EUGENIA	350000	TOLEDO	2 2
	3	JOSEP	250000	SITGES	3 1

Model Relacional

- **Clau primària:** per ex. **num_dpt** és la clau primària de la taula departaments.
- **Clau forana.** per ex. **num_dpt** és clau forana de la taula empleats.

departaments(<u>num_dpt</u> , nom_dpt, planta, edifici, ciutat_dpt)					
1	DIRECCIO	10	PAU CLARIS	BARCELONA	
2	DIRECCIO	8	RIOS ROSAS	MADRID	
3	MARQUETING	1	PAU CLARIS	BARCELONA	
projectes(<u>num_proj</u> , nom_proj, producte, pressupost)					
1	IBDTEL	TELEVISIO	1000000		
2	IBDVID	VIDEO	500000		
empleats(<u>num_empl</u> , nom_empl, sou, ciutat_empl, <u>num_dpt</u> , <u>num_proj</u>)					
1	CARME	400000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1

Creació d'una taula

```
CREATE TABLE <nom_taula>  
    (<nom_columna> <tipus_dades> [<restriccions_col>] [<val_per_defecte>]  
    [, <nom_columna> <tipus_dades> [<restriccions_col>] [<val_per_defecte>]...]  
    [<restriccions_taula>]);
```

- **tipus_dades:** INTEGER, FLOAT(precisió), REAL, CHAR(n), NUMERIC(precisió,escala), DECIMAL(precisió,escala), SMALLINT, DOUBLE PRECISION, VARCHAR(n), DATE,....
- **val_per_defecte:** Valor per defecte d'una columna per a una fila que s'insereix a la taula.

DEFAULT { <literal> | NULL }.

Restriccions de taula i de columna

- **restriccions_taula:**
 - UNIQUE (<cols>)
 - PRIMARY KEY (<cols>)
 - FOREIGN KEY (<cols>) REFERENCES <taula> [<cols>]
 - CHECK (<condicions>)
- **restriccions_col:**
 - UNIQUE
 - PRIMARY KEY
 - REFERENCES <taula> [<col>]
 - CHECK (<condicions>)
 - NOT NULL

Són **necessàriament restriccions de taula** aquelles que tenen a veure amb més d'una columna (Ex: clau primària formada per dues o més columnes).

Exemple de creació d'una taula

```
CREATE TABLE empleats
( num_empl      INTEGER,
  nom_empl      CHAR(30) NOT NULL,
  sou           INTEGER DEFAULT 100000
                    CHECK (sou>80000),
  ciutat_empl   CHAR(30),
  num_dpt       INTEGER,
  num_proj      INTEGER,
PRIMARY KEY (num_empl),
FOREIGN KEY (num_dpt) REFERENCES departaments(num_dpt),
FOREIGN KEY (num_proj) REFERENCES projectes(num_proj));
```

Insertió de files en una taula

```
INSERT INTO <nom_taula> [(<columnes>)]  
( VALUES {<valor1> | NULL}, ..., {<valorn> | NULL} ) | <consulta> ;
```

- En cas de no posar les **columnes** a continuació del **nom_taula**, els valors han de correspondre exactament als valors de les columnes en el **CREATE TABLE** i en el mateix ordre.
- En cas de posar les **columnes**, els valors han de correspondre als valors de les columnes explicitades i en el mateix ordre.
- Els valors de les columnes de la fila o files a inserir es poden obtenir també com a resultat d'una **consulta** (veure subconsultes).

Exemples d'inserció de files en una taula

INSERT INTO empleats

VALUES (4, 'RICARDO', 400000, 'BARCELONA',1,1);

INSERT INTO empleats (num_empl, num_dpt, num_proj, nom_empl)

VALUES (11, 3, 2, 'NURIA');

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)					
1	CARME	400000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	400000	BARCELONA	1	1
11	NURIA	100000	NULL	3	2

Esborrat de files d'una taula

```
DELETE FROM <taula>  
WHERE <condicions>;
```

- S'eliminen de la **taula** les files que acompleixen les **condicions** especificades a la clausula **WHERE**.
-

Exemples d'esborrat de files d'una taula

DELETE FROM empleats

WHERE num_dpt=2;

DELETE FROM empleats

WHERE sou <= 250000;

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)

1	CARME	400000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	400000	BARCELONA	1	1
11	NURIA	100000	NULL	3	2

Modificació de files d'una taula

```
UPDATE <taula>  
SET <col> = {expressió/ NULL} [, <col> = {expressió/ NULL}...]  
WHERE <condicions> ;
```

- Es modifiquen de la manera indicada a la clausula **SET** les columnes de les files de la **taula** que aconsegueixen les **condicions** especificades a la clausula **WHERE**.

Exemples de modificació de files d'una taula

```
UPDATE empleats  
SET sou = sou + 10000  
WHERE num_dpt = 1;
```

```
UPDATE empleats  
SET sou = sou + 50000, ciutat_empl = 'VIC'  
WHERE num_empl = 11;
```

empleats(num_empl, nom_empl, sou, ciutat_empl, **num_dpt**, **num_proj**)

1	CARME	410000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	410000	BARCELONA	1	1
11	NURIA	150000	VIC	3	2

Consultes sobre una taula: Format bàsic

```
SELECT <columnes_a_seleccionar> | *  
FROM <taula_a_consultar>  
[ WHERE <condicions> ] ;
```

- El resultat de la consulta és el valor de les **columnes_a_seleccionar** de la **taula_a_consultar** únicament per a la fila o files que aconsegueixen les **condicions** especificades a la clausula **WHERE**.
- En el cas de no posar la clausula **WHERE**, el resultat és el valor de les **columnes_a_seleccionar** per totes les files de la **taula_a_consultar**.
- Si posem un * en lloc de **columnes_a_seleccionar** indica que estem interessats en totes les columnes de la **taula_a_consultar**.

Exemple de consultes bàsiques sobre una taula - 1

```
SELECT *  
FROM empleats;
```

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)

1	CARME	410000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	410000	BARCELONA	1	1
11	NURIA	150000	VIC	3	2

Exemple de consultes bàsiques sobre una taula - 2

```
SELECT num_empl, nom_empl, sou  
FROM empleats;
```

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)					
1	CARME	410000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	410000	BARCELONA	1	1
11	NURIA	150000	VIC	3	2

Exemple de consultes bàsiques sobre una taula - 3

```
SELECT num_empl, nom_empl, sou  
FROM empleats  
WHERE num_dpt = 3;
```

empleats(<u>num_empl</u> , nom_empl, sou, ciutat_empl, num_dpt, num_proj)					
1	CARME	410000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	410000	BARCELONA	1	1
11	NURIA	150000	VIC	3	2

Operadors en les condicions

- **operadors**
 - **de comparació:** =, <, >, <=, >=, <>
 - **lògics:** NOT, AND, OR
 - **altres:**
 - » <columna> **BETWEEN** <límit₁> **AND** <límit₂>
 - » <columna> **IN** (<valor₁>, <valor₂> [...,<valor_N>])
 - » <columna> **LIKE** <característica>
 - » <columna> **IS [NOT] NULL**

Aquests operadors poden sortir a les **condicions**

- En la clausula **WHERE** de les sentències d'esborrat (**DELETE**), modificació (**UPDATE**) i consulta (**SELECT**)
 - En la clausula **CHECK** de les sentències de creació d'una taula (**CREATE TABLE**).
-

Exemple de consultes amb tot tipus d'operadors

```
SELECT num_empl, nom_empl
FROM empleats
WHERE NOT(num_dpt = 2) AND
      ( ciutat_empl IN ('MATARO', 'SITGES', 'BARCELONA') OR
        ciutat_empl LIKE 'V%') AND
      num_proj IS NOT NULL AND
      sou BETWEEN 400000 AND 500000;
```

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)						
1	CARME	410000	MATARO	1	1	
2	EUGENIA	350000	TOLEDO	2	2	
3	JOSEP	250000	SITGES	3	1	
4	RICARDO	410000	BARCELONA	1	1	
11	NURIA	150000	VIC	3	2	

Consultes sobre una taula: Resultats sense repeticions

```
SELECT [ DISTINCT | ALL] <columnes_a_seleccionar>  
FROM <taula_a_consultar>  
[ WHERE <condicions> ] ;
```

- Si volem que el resultat d'una consulta se'ns doni sense repeticions, cal utilitzar la paraula clau **DISTINCT**.
- Si no es posa res se'ns donarà el resultat amb repeticions (en cas de que n'hi hagin). Cosa que també es pot demanar explícitament amb la paraula clau **ALL**.

Exemple de consultes amb resultats sense repeticions

SELECT DISTINCT nom_empl, sou
FROM empleats
WHERE num_dpt **IN** (1,3);

resultat →

nom_empl	sou
CARME	410000
JOSEP	250000
RICARDO	410000
NURIA	150000

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)					
1	CARME	410000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	410000	BARCELONA	1	1
11	NURIA	150000	VIC	3	2
12	NURIA	150000	MATARO	1	5

Consultes sobre una taula: Funcions d'agregació

```
SELECT <funcions_d'agregació>  
FROM <taula_a_consultar>  
[ WHERE <condicions> ] ;
```

- Són funcions que s'apliquen sobre el conjunt de files de la **taula_a_consultar** que aconpleixen les **condicions** especificades a la clausula **WHERE**.
 - **COUNT**: dóna:
 - » el número de files seleccionades (COUNT(*))
 - » el número de valors diferents d'una certa columna (COUNT(DISTINCT <columna>))
 - **SUM**: dóna la suma dels valors d'una columna per les files seleccionades
 - **MIN**: dóna el valor mínim d'una columna per les files seleccionades
 - **MAX**: dóna el valor màxim d'una columna per les files seleccionades
 - **AVG**: dóna el valor promig d'una columna per les files seleccionades

Exemple de consultes amb funcions agregades

SELECT COUNT(*) **AS** quantEmpl ,

COUNT(DISTINCT nom_empl) **AS** quantNoms,

AVG(sou) **AS** avgSou

FROM empleats

WHERE num_dpt **IN** (1,3);

resultat →

quantEmpl quantNoms avgSou

5

4

274000

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)

1	CARME	410000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	410000	BARCELONA	1	1
11	NURIA	150000	VIC	3	2
12	NURIA	150000	MATARO	1	5

Consultes sobre una taula: Ordenació

```
SELECT <columnes_a_seleccionar> | *  
FROM <taula_a_consultar>  
[ WHERE <condicions> ]  
ORDER BY <columna> [DESC | ASC],.... ;
```

- En el resultat s'obté les dades ordenades segons les columnes que s'explicitin a la clausula **ORDER BY**.
- Si per a una columna no es posa **DESC** s'enten que la classificació, segons els seus valors, es vol que sigui ascendent. Cosa que també es pot demanar explícitament amb la paraula clau **ASC**.

Exemple de consultes amb ordenació de resultats

SELECT num_empl, nom_empl, sou
FROM empleats
WHERE num_dpt **IN** (1,2)
ORDER BY sou **DESC**, nom_empl;

resultat →

num_empl	nom_empl	sou
1	CARME	410000
4	RICARDO	410000
2	EUGENIA	350000
12	NURIA	150000

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)					
1	CARME	410000	MATARO	1	1
2	EUGENIA	350000	TOLEDO	2	2
3	JOSEP	250000	SITGES	3	1
4	RICARDO	410000	BARCELONA	1	1
11	NURIA	150000	VIC	3	2
12	NURIA	150000	MATARO	1	5

Consultes sobre una taula: Amb agrupació de files -1

```
SELECT <columnes_a_seleccionar> [,<funcions_d'agregació>]  
FROM <taula_a_consultar>  
[ WHERE <condicions> ]  
GROUP BY <columnes_segons_les_que_agrupar>;
```

- S'organitza en grups les files de la **taula_a_consultar** que aconpleixen les **condicions** especificades a la clausula **WHERE**, segons el seu valor per les **columnes_segons_les_que_agrupar**.
 - El resultat de la consulta és el valor de les **columnes_a_seleccionar** per cadascun dels grups de files obtinguts.
 - En el resultat es pot demanar també el valor de **funcions_d'agregació** que es calculen per cadascun dels grups de files obtinguts.
 - Les **columnes_a_seleccionar** han de ser un subconjunt de les **columnes_segons_les_que_agrupar**.
-

Exemple de consultes amb agrupació de files - 1

```
SELECT num_dpt,  
        COUNT(*) AS quantEmpl  
FROM empleats  
WHERE num_dpt IN (1,3)  
GROUP BY num_dpt;
```

resultat →

num_dpt	quantEmpl
1	3
3	2

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)						
grup1	1	CARME	410000	MATARO	1	1
	2	EUGENIA	350000	TOLEDO	2	2
grup2	3	JOSEP	250000	SITGES	3	1
	4	RICARDO	410000	BARCELONA	1	1
	11	NURIA	150000	VIC	3	2
	12	NURIA	150000	MATARO	1	5

Consultes sobre una taula: Amb agrupació de files -2

```
SELECT <columnes_a_seleccionar> [, <funcions_d'agregació>]  
FROM <taula_a_consultar>  
[ WHERE <condicions> ]  
GROUP BY <columnes_segons_les_que_agrupar>  
HAVING <condicions_per_grups>;
```

- En el cas de posar la clausula **HAVING**, el resultat només apareix per als grups que aconsegueixen les **condicions_per_grups**.
- Les **condicions_per_grups** seran comparacions entre constants, valors de les columnes per les que s'ha definit grups i valors de funcions d'agregació.
- Les funcions d'agregació només té sentit aplicar-les a columnes que no estan en el conjunt de **columnes_segons_les_que_agrupar**.

Exemple de consultes amb agrupació de files - 2

SELECT num_dpt, SUM(sou) **AS** sumaSous

FROM empleats

GROUP BY num_dpt

HAVING COUNT(*) >= 3;

resultat →

num_dpt	sumaSous
1	970000

empleats(<u>num_empl</u> , nom_empl, sou, ciutat_empl, num_dpt, num_proj)						
grup1	1	CARME	410000	MATARO	1	1
	2	EUGENIA	350000	TOLEDO	2	2
	3	JOSEP	250000	SITGES	3	1
	4	RICARDO	410000	BARCELONA	1	1
	11	NURIA	150000	VIC	3	2
	12	NURIA	150000	MATARO	1	5

Exemple de consultes amb agrupació de files - 3

SELECT DISTINCT num_dpt

FROM empleats

GROUP BY num_dpt, ciutat_empl

HAVING COUNT(*) >= 2;

resultat → num_dpt
1

empleats(num_empl, nom_empl, sou, ciutat_empl, num_dpt, num_proj)						
grup1	1	CARME	410000	MATARO	1	1
grup2	2	EUGENIA	350000	TOLEDO	2	2
grup3	3	JOSEP	250000	SITGES	3	1
grup4	4	RICARDO	410000	BARCELONA	1	1
	11	NURIA	150000	VIC	3	2
grup5	12	NURIA	150000	MATARO	1	5
	13	ALBERT	150000	BARCELONA	1	5

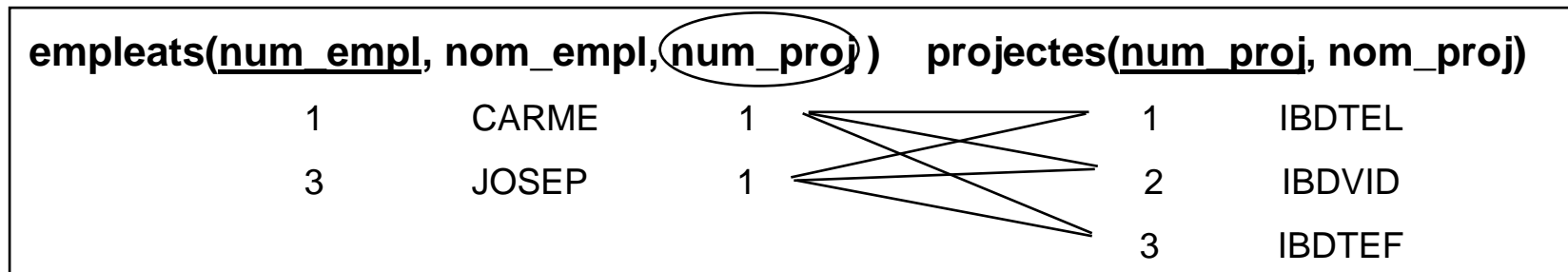
Consultes sobre més d'una taula: Format bàsic

```
SELECT <columnes_a_seleccionar> | *  
FROM <taules_a_consultar>  
[ WHERE <condicions> ] ;
```

- El resultat de la consulta és el valor de les **columnes_a_seleccionar** de les **taules_a_consultar** únicament per a la fila o files que compleixen les **condicions** especificades a la clausula **WHERE**.
- En el cas de no posar la clausula **WHERE**, el resultat és el valor de les **columnes_a_seleccionar** per a les files obtingudes del producte cartesià de les files a les **taules_a_consultar**.
- Si posem un * en lloc de les **columnes_a_seleccionar** indica que estem interessats en totes les columnes de les **taules_a_consultar**.

Exemple de consultes sobre més d'una taula - 1

		e.num_empl	e.nom_empl	e.num_proj	p.num_proj	p.nom_proj
SELECT *	FROM empleats e, projectes p;	1	CARME	1	1	IBDTEL
		1	CARME	1	2	IBDVID
		1	CARME	1	3	IBDTEF
		3	JOSEP	1	1	IBDTEL
		3	JOSEP	1	2	IBDVID
		3	JOSEP	1	3	IBDTEF



Exemple de consultes sobre més d'una taula - 2

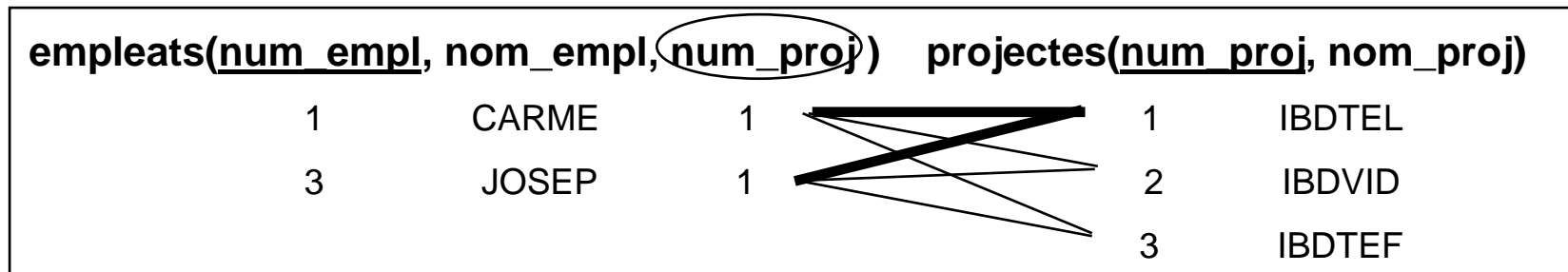
SELECT e.num_empl, p.num_proj, p.nom_proj

FROM empleats e, projectes p

WHERE e.num_proj = p.num_proj;

resultat →

e.num_empl	p.num_proj	p.nom_proj
1	1	IBDTEL
3	1	IBDTEL



Consultes sobre més d'una taula: General

- En les consultes sobre més d'una taula es pot aplicar totes les variants que hem mostrat per a les consultes sobre una única taula. És a dir:
 - resultats sense repeticions
 - funcions d'agregació
 - ordenació de les dades
 - agrupació de files
-

Consultes: Unió

```
SELECT <columnes_a_seleccionar> | *  
FROM <taules_a_consultar>  
[ WHERE <condicions> ]  
UNION  
SELECT <columnes_a_seleccionar> | *  
FROM <taules_a_consultar>  
[ WHERE <condicions> ]  
[ORDER BY <columna> [DESC|ASC],...] ;
```

- El resultat és la unió del resultat obtingut de les dues sentències **SELECT**.
 - Les **columnes_a_seleccionar** en les dues sentències **SELECT** han de ser semànticament compatibles
 - Sempre s'obté resultats sense repeticions (en molts SGBDR ja surten ordenats).
 - Les columnes que apareixen a clausula **ORDER_BY** han de ser un subconjunt de les **columnes_a_seleccionar** del primer **SELECT**.
-

Exemple de consultes amb unió

```
SELECT ciutat_empl
FROM empleats
UNION
SELECT ciutat_dpt
FROM departaments
ORDER BY ciutat_empl DESC;
```

resultat →

ciutat_empl
SITGES
MATARO
MADRID
BARCELONA

empleats(<u>num_empl</u> , nom_empl, ciutat_empl)			departaments(<u>num_dpt</u> , ciutat_dpt)	
1	CARME	MATARO	1	BARCELONA
3	JOSEP	SITGES	2	MADRID
			3	BARCELONA

Consultes: Diferència

```
SELECT <columnes_a_seleccionar> | *  
FROM <taules_a_consultar>  
[ WHERE <columna> NOT IN ( SELECT <columna_a_seleccionar>  
                                FROM <taules_a_consultar>  
                                [ WHERE <condicions> ]) );
```

```
SELECT <columnes_a_seleccionar> | *  
FROM <taules_a_consultar>  
[ WHERE NOT EXISTS ( SELECT *  
                        FROM <taules_a_consultar>  
                        [ WHERE <condicions> ]) );
```

Exemple de consultes amb diferència

```

SELECT p.num_proj, p.nom_proj
FROM projectes p
WHERE p.num_proj NOT IN (SELECT e.num_proj
                           FROM empleats e);

```

resultat →

```

SELECT p.num_proj, p.nom_proj
FROM projectes p
WHERE NOT EXISTS (SELECT * FROM empleats e
                   WHERE p.num_proj = e.num_proj);

```

p.num_proj	p.nom_proj
2	IBDVID
3	IBDTEF
4	IBDCOM

empleats(<u>num_empl</u> , nom_empl, <u>num_proj</u>)			projectes(<u>num_proj</u> , nom_proj)	
1	CARME	1	1	IBDTEL
3	JOSEP	1	2	IBDVID
			3	IBDTEF
			4	IBDCOM

Subconsultes - 1

- Poden apareixer en aquelles sentències on hi ha la clausula **WHERE**:
 - Esborrat de files d'una taula
DELETE FROM <taula>
WHERE (**SELECT**);
 - Modificació de files d'una taula
UPDATE <taula>
SET
WHERE (**SELECT**);
 - Consultes una taula o més taules
SELECT <columnes_a_seleccionar>
FROM <taules_a_consultar>
WHERE (**SELECT**);
-

Exemple de subconsulta en una sentència d'esborrat

```
DELETE FROM projectes  
WHERE NOT EXISTS (SELECT *  
                    FROM empleats e  
                    WHERE e.num_proj = projectes.num_proj);
```

empleats(num_empl, nom_empl, num_proj) **projectes**(num_proj, nom_proj)

1 CARME

1

3 JOSEP

1

1 IBDTEL

2 IBDVID

3 IBDTEF

4 IBDCOM

Exemple de subconsulta en una sentència de modificació

```
UPDATE projectes
SET pressupost = pressupost + (pressupost * 0,1)
WHERE 2 <= (SELECT COUNT(*)
             FROM empleats e
             WHERE projectes.num_proj = e.num_proj);
```

empleats(num_empl, nom_empl, num_proj)			projectes(num_proj, pressupost)	
1	CARME	1	1	1100000
3	JOSEP	1	2	500000
			3	4500000
			4	2000000

Exemple de subconsulta en una sentència de consulta - 1

```
SELECT e.num_empl, e.nom_empl
FROM empleats e
WHERE e.sou > ( SELECT AVG(e1.sou)
                FROM empleats e1);
```

empleats(<u>num_empl</u> , nom_empl, sou, ciutat_empl, num_dpt, num_proj)						
1	CARME	410000	MATARO	1	1	
2	EUGENIA	350000	TOLEDO	2	2	
3	JOSEP	250000	SITGES	3	1	
4	RICARDO	410000	BARCELONA	1	1	
11	NURIA	150000	VIC	3	2	

Exemple de subconsulta en una sentència de consulta - 2

SELECT p.num_proj, p.nom_proj

FROM projectes p

WHERE p.pressupost < (SELECT SUM(e.sou)

FROM empleats e

WHERE e.num_proj=p.num_proj);

resultat →

p.num_proj, p.nom_proj

1

IBDTEL

projectes(num_proj, nom_proj, p.pressupost)

1	IBDTEL	1000000
2	IBDVID	500000

empleats(num_empl, nom_empl, sou, num_proj)

1	CARME	410000	1
2	EUGENIA	350000	2
3	JOSEP	250000	1
4	RICARDO	410000	1
11	NURIA	150000	2

Exemple de subconsulta en una sentència de consulta - 2 (continua)

SELECT p.num_proj, p.nom_proj

FROM projectes p, empleats e

WHERE p.num_proj = e.num_proj

GROUP BY p.num_proj, p.nom_proj, p.pressupost

HAVING p.pressupost < SUM(e.sou);

resultat →

p.num_proj, p.nom_proj

1

IBDTEL

e.num_empl, e.nom_empl, e.sou, e.num_proj, p.num_proj, p.nom_proj, p.pressupost							
grup1	1	CARME	410000	1	1	IBDTEL	1000000
	2	EUGENIA	350000	2	2	IBDVID	500000
grup2	3	JOSEP	250000	1	1	IBDTEL	1000000
	4	RICARDO	410000	1	1	IBDTEL	1000000
	11	NURIA	150000	2	2	IBDVID	500000

Subconsultes -2

- En aquelles sentències on hi ha la clausula **HAVING**:

- Consultes amb agrupació de files

```
SELECT <columnes_a_seleccionar>  
FROM <taules_a_consultar>  
WHERE <condicions>  
GROUP BY <columnes_agrupació>  
HAVING ..... (SELECT ..... );
```

- Finalment, en sentències d'inserció de files a una taula (en aquest cas el resultat de la subconsulta és un conjunt de files que ha de ser compatible amb la definició de la taula en el **CREATE TABLE**)

```
INSERT INTO <taula>  
(SELECT ..... );
```

Exemple de subconsulta en una sentència de consulta amb agrupació de files

SELECT d.num_dpt, d.nom_dpt, SUM(e.sou) **AS** sumaSous

FROM departament d, empleats e

WHERE d.num_dpt = e.num_dpt

GROUP BY d.num_dpt, d.nom_dpt

HAVING SUM(e.sou) > (**SELECT** SUM(e1.sou)

FROM empleats e1

WHERE e1.num_dpt = 3));

resultat →

d.num_dpt	d.nom_dpt	sumaSous
1	DIRECCIO	820000

empleats(num_empl, nom_empl, sou, num_dpt)				departaments(num_dpt, nom_dpt)	
1	CARME	410000	1	1	DIRECCIO
2	EUGENIA	350000	2	2	DIRECCIO
3	JOSEP	350000	3	3	MARQUETING
4	RICARDO	410000	1		
11	NURIA	150000	3		

Exemple de subconsulta en una sentència d'inserció

```
INSERT INTO clients  
  (SELECT num_empl,nom_empl,200000  
   FROM empleats  
   WHERE num_dpt IN (2,3));
```

clients(num_client, nom, credit)	empleats(num_empl, nom_empl, num_dpt)
---	---

2	EUGENIA	200000
3	JOSEP	200000

1	CARME	1
2	EUGENIA	2
3	JOSEP	3
4	RICARDO	1

