# 3.2 Llenguatges: SQL

- Introducció
- Base de dades exemple
- 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 estàndard per al Model Relacional de bases de dades en els anys 1986-87 (ANSI/X3H2/RDL).
- Des d'aleshores se n'han fet vàries versions: SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016, SQL:2023.
- 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.



# Base de dades exemple

 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_c	dpt,	planta,	edific	i, ciu	utat_dpt)
	1	DIRECC	CIO	10	PAU CLAR	IS B	ARCELONA
	2	DIRECC	OI	8	RIOS ROS	AS M	MADRID
	3 1	MARQUE	TING	1	PAU CLAR	IS B	ARCELONA
projectes(num_proj	, non	n_proj, <sub>l</sub>	produc	te, pre	ssupost)		
1	IBD	TEL T	ELEVIS	IO 1	000000		
2	IBD'	VID	VIDEO	į į	500000		
empleats(num_emp	l, no	m_empl	, sou, e	ciutat_	empl, nun	n_dpt, nui	m_proj)
1	C	CARME	400000	) MATA	\RO	1	1
2	Εl	JGENIA	350000	) TOLE	EDO	2	2
3	J	OSEP	250000	SITG	ES	3	1

### Base de dades exemple

- Clau primària: Cada taula té una clau primària que permet identificar les files de la taula. Per ex. num\_dpt és la clau primària de la taula departaments. Això vol dir que cada departament té un num\_dpt que ha de ser únic entre tots els departaments, és a dir mai hi haurà dos departaments amb el mateix número de departament.
- Clau forana. Una clau forana permet relacionar les files de dues taules. Per ex. num\_dpt és una clau forana de la taula empleats que referencia la taula departaments. Això vol dir que entre les dades d'un empleat hi haurà també el departament al que pertany, i aixó ens permetrà saber el departament on treballa un empleat, i també els empleats que treballen a un departament.

departaments(num	dpt,	nom_c	dpt, p	lanta,	edifici,	ciutat_dpt)
	1	DIRECO	CIO	10	PAU CLARIS	BARCELONA
	2	DIRECC	CIO	8	RIOS ROSAS	MADRID
	3 N	MARQUE	TING	1	PAU CLARIS	BARCELONA
projectes( <u>num_proj</u>	, non	n_proj, ¡	product	te, pres	supost)	
1	IBD	TEL T	ELEVISI	0 10	00000	
2	IBD'	VID	VIDEO	5	00000	
empleats( <u>num_emp</u>	<u>l</u> , no	m_empl	l, sou, c	iutat_e	empl, (num_c	lpt (num_proj))
1	C	CARME	400000	MATAF	RO 1	1
2	Εl	JGENIA	350000	TOLE	00 2	2
3	J	OSEP	250000	SITGE	S 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 }.



#### Creació d'una taula: Restriccions de columna

#### restriccions\_col:

UNIQUE	La columna no pot tenir valors repetits
PRIMARY KEY	La columna és clau primària de la taula
REFERENCES <taula> [<col/>]</taula>	La columna és clau forana que referencia la taula indicada.
CHECK ( <condicions>)</condicions>	La columna ha de complir les condicions especificades. Només pot referir-se a la columna per la que es defineix.
NOT NULL	La columna no pot tenir valors nuls

Les restriccions de columna es refereixen a una única columna.

Per exemple, en la condició d'un CHECK de columna no es pot fer referència a més d'una columna.



#### Creació d'una taula: Restriccions de taula

#### restriccions\_taula:

UNIQUE ( <cols>)</cols>	El conjunt de les columnes especificades han de tenir valors únics entre les files de la taula
PRIMARY KEY ( <cols>)</cols>	El conjunt de les columnes especificades formen la clau primària
FOREIGN KEY ( <cols>) REFERENCES <taula> [<cols>]</cols></taula></cols>	El conjunt de columnes especificades formen una clau forana que referencia la taula indicada.
CHECK ( <condicions>)</condicions>	La taula ha de complir les condicions especificades. La condició pot referir-se a una o més columnes de la taula.

Les restriccions de taula poden referir-se a una o més columnes de la taula. Així, en cas de restriccions que tenen a veure amb més d'una columna cal usar una restricció de taula.

Per exemple, en cas de claus primàries compostes per més d'un columna o condició (CHECK) que tenen a veure amb més d'una columna.



### Creació d'una taula: Exemple

#### **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

departements(num\_dpt)

departaments(num\_dpt),

FOREIGN KEY (num\_proj) REFERENCES projectes(num\_proj));



#### Inserció de files en una taula

```
INSERT INTO <nom_taula> [(<columnes>)]
( VALUES {<valor<sub>1</sub>> | NULL}, ..., {<valor<sub>n</sub>> | 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).



### Inserció de files en una taula: Exemples

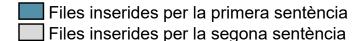
**INSERT INTO** empleats

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

INSERT INTO empleats (num\_empl, num\_dpt, num\_proj, nom\_empl)

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

empleats( <u>num_e</u>	mpl, nom_empl	, sou, c	iutat_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	A 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 compleixen les **condicions** especificades a la clàusula **WHERE**.

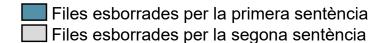


# Esborrat de files d'una taula: Exemples

**DELETE FROM** empleats **WHERE** num\_dpt=2;

**DELETE FROM** empleats **WHERE** sou <= 250000;

empleats( <u>ı</u>	num ei	mpl, nom_empl	, sou, c	iutat_empl,	num_dpt	num_pro	)))
	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 clàusula **SET** les columnes de les files de la **taula** que compleixen les **condicions** especificades a la clàusula **WHERE**.



### Modificació de files d'una taula: Exemples

**UPDATE** empleats **SET** sou = sou +10000

**WHERE** num\_dpt = 1;

**UPDATE** empleats

**SET** sou = sou + 50000, ciutat\_empl = 'VIC'

**WHERE** num\_empl = 11;

empleats( <u>r</u>	num_	<u>empl</u> , nom_empl	, sou, c	iutat_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	A 1	1
	11	NURIA	150000	VIC	3	2

Files modificades per la primera sentència
Files modificades per la segona sentència



#### 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 acompleixen les condicions especificades a la clàusula WHERE.
- En el cas de no posar la clàusula 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**.



# Consultes sobre una taula: Format bàsic - Exemple 1

**SELECT** \* **FROM** empleats;

empleats( <u>n</u>	um_en	npl, nom_empl	, sou, c	iutat_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 - Exemple 2

**SELECT** num\_empl, nom\_empl, sou **FROM** empleats;

empleats( <u>n</u>	empleats( <u>num_empl</u> , nom_empl, sou, ciutat_empl, <u>num_dpt</u> <u>num_proj</u> ))								
	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 - Exemple 3

**SELECT** num\_empl, nom\_empl, sou **FROM** empleats **WHERE** num\_dpt = 3;

empleats( <u>nu</u>	m_empl,	nom_empl	l, sou, ci	utat_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
  - aritmètics: \*, +, -, /
  - de comparació: =, <, >, <=, >=, <>
  - lògics: NOT, AND, OR
  - altres:
    - <columna> BETWEEN <límit<sub>1</sub>> AND <límit<sub>2</sub>>
    - <columna> IN (<valor<sub>1</sub>>,<valor<sub>2</sub>> [...,<valor<sub>N</sub>>])
    - <columna> LIKE <característica>
    - <columna> IS [NOT] NULL

Aquests operadors poden sortir a les condicions

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



### Operadors en les condicions: Exemple

empleats(n	um_	<u>empl</u> , nom_empl	l, sou, ci	utat_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	A 1	1	
	11	NURIA	150000	VIC	3	2	1

Dades obtingudes com a resultat de la consulta

#### 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 clàusula 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**.



410000

350000

150000

### Consultes sobre una taula: Ordenació - Exemple

num\_empl nom\_empl sou SELECT num\_empl, nom\_empl, sou CARME **FROM** empleats resultat **RICARDO 410000** WHERE num\_dpt IN (1,2) EUGENIA **ORDER BY** sou **DESC**, nom\_empl; 12 **NURIA** 

empleats( <u>r</u>	empleats( <u>num_empl</u> , nom_empl, sou, ciutat_empl, <u>num_dpt</u> ) <u>num_proj</u>									
	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				

Files que compleixen la condició del WHERE

Dades tal com s'obtenen (ordenades) com a resultat de la consulta



### 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.



### Consultes sobre una taula: Resultats sense repeticions

SELECT DISTINCT nom\_empl, sou
FROM empleats

WHERE num\_dpt IN (1,3);

nom\_empl sou

CARME 410000

JOSEP 250000

RICARDO 410000

NURIA 150000

empleats(	num_	<u>empl</u> , nom_empl	l, sou, ci	utat_empl, (ni	um_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	1

Files que compleixen la condició del WHERE
Dades obtingudes com a resultat de la consulta

### 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 acompleixen les condicions especificades a la clàusula WHERE.
  - COUNT:
    - COUNT(\*) número de files que compleixen la condició del where
    - COUNT(DISTINCT <columna>) número de valors diferents de la columna, sense comptar valors NULL, per a les files que compleixen la condició del where.
    - **COUNT(<columna>)** número de valors de la columna, sense comptar valors NULL, per a les files que compleixen la condició del where.
  - SUM (expressió), MIN(expressió), MAX(expressió), AVG(expressió):
    - expressió pot ser simplement una columna, o pot ser una càlcul a partir del valor de diferents columnes i constants.
    - **SUM**: dóna la suma dels valors resultants de calcular l'expressió per a les files que compleixen la condició del WHERE
    - MIN: dóna el valor mínim dels resultats de calcular l'expressió per a les files que compleixen la condició del WHERE
    - MAX: dóna el valor màxim dels resultats de calcular l'expressió per a les files que compleixen la condició del WHERE
    - AVG: dóna el valor promig dels resultats de calcular l'expressió per a les files que compleixen la condició del WHERE



### Consultes sobre una taula: Funcions d'agregació - Exemple

SELECT COUNT(\*) AS quantEmpl ,

COUNT(DISTINCT nom\_empl) AS quantNoms,
quantNoms,
SUM(sou\*0.1) AS partSou

FROM empleats

WHERE num\_dpt IN (1,3);

quantEmpl quantNoms partSou
5 4 137000

empleats( <u>n</u>	um_en	<u>npl,</u> nom_empl	, sou, ci	utat_empl, (r	um_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

Files que compleixen la condició del WHERE

Resultat de la consulta



### Consultes sobre una taula: Agrupació de files

**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 acompleixen les condicions especificades a la clàusula 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.
- Les **columnes\_a\_seleccionar** han de ser columnes que continguin el mateix valor per totes les files dins d'un grup.
- En el resultat es pot demanar també el valor de **funcions\_d'agregació** que es calculen per cadascun dels grups de files obtinguts.



### Consultes sobre una taula: Agrupació de files - Exemple

SELECT num\_dpt,

COUNT(\*) AS quantEmpl

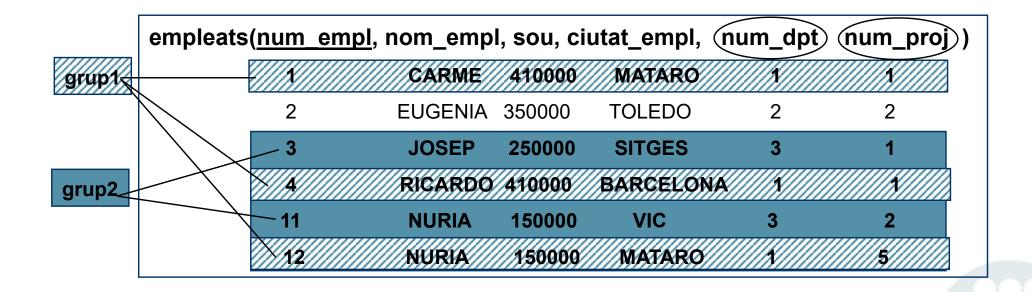
FROM empleats

WHERE num\_dpt IN (1,3)

GROUP BY num dpt;

num\_dpt quantEmpl

1 3
3 2



Files que compleixen la condició del WHERE, agrupades segons indica la clàusula GROUP BY
Dades obtingudes com a resultat de la consulta. Hi ha un resultat per cada grup

### Consultes sobre una taula: Condicions sobre grups

**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 clàusula HAVING, només apareix un resultat per cada grup que compleix les condicions\_per\_grups.
- Les **condicions\_per\_grups** seran comparacions entre constants, columnes amb un valor únic per cada grup i valors de funcions d'agregació (també amb un únic valor per cada grup). Per exemple: si s'agrupa els empleats per departament, el sou de cada empleat d'un departament pot ser diferent, però el MAX del sou dels empleats d'un departament té valor únic.
- Les **funcions d'agregació** té sentit aplicar-les a columnes que poden tenir valors diferents entre les files que composen els grups.



### Consultes sobre una taula: Condicions sobre grups - Exemple 1

SELECT num\_dpt, SUM(sou) AS sumaSous

FROM empleats

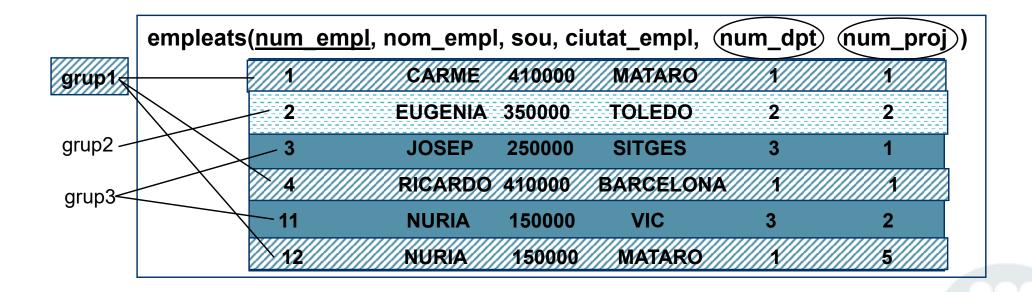
GROUP BY num\_dpt

resultat

num\_dpt sumaSous

1 970000

1 970000



En no haver-hi WHERE, totes les files agrupades segons indica la clàusula GROUP BY

Dades obtingudes com a resultat de la consulta, un resultat per cada grup que compleix la condició del HAVING.

### Consultes sobre una taula: Condicions sobre grups - Exemple 2

FROM empleats

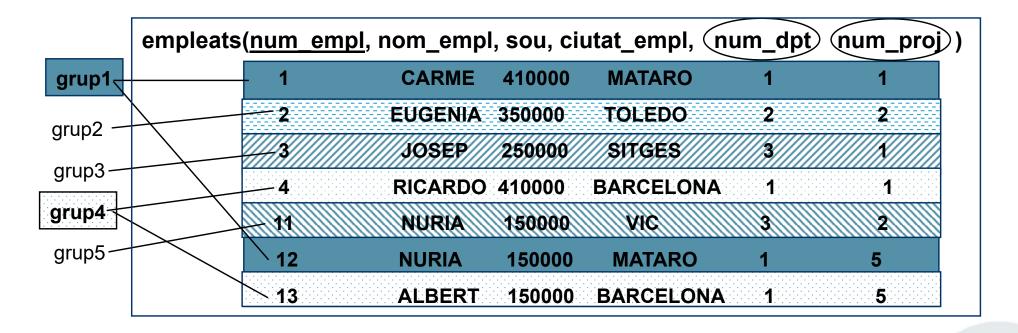
GROUP BY num\_dpt, ciutat\_empl

HAVING COUNT(\*) >= 2;

resultat

num\_dpt

1



En no haver-hi WHERE, són totes les files les que s'agrupen segons indica la clàusula GROUP BY

Dades obtingudes com a resultat de la consulta. Un resultat per cada grup que compleix la condició del HAVING, sense resultats repetits degut a la clàusula DISTINCT.

#### Consultes sobre més d'una taula: Format bàsic

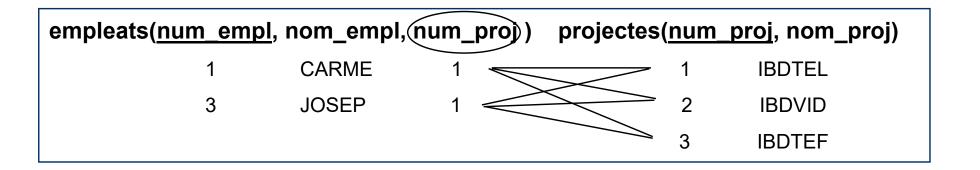
```
SELECT <columnes_a_seleccionar> | *
FROM <taules_a_consultar>
[ WHERE <condicions> ];
```

- 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 clàusula WHERE.
- En el cas de no posar la clàusula 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**.

### Consultes sobre més d'una taula: Format bàsic – Exemple 1

e.num\_empl e.nom\_empl e.num\_proj p.num\_proj p.nom\_proj

		1	CARME	1	1	IBDIEL	
CELECT *	resultat	1	CARME	1	2	IBDVID	
SELECT *	<b>─</b>	1	CARME	1	3	IBDTEF	
FROM empleats e, projec	ojectes p;	3	JOSEP	1	1	IBDTEL	
		3	JOSEP	1	2	IBDVID	
		3	JOSEP	1	3	IBDTEF	



Dades obtingudes com a resultat de la consulta. Cal notar que, de totes les files, les que segurament podem estar interessats, són aquelles marcades en negreta (són aquelles en què el número de projecte en què treballa l'empleat coincideix amb el número de projecte del projecte).

### Consultes sobre més d'una taula: Format bàsic – Exemple 2

SELECT e.num\_empl, p.num\_proj,
p.nom\_proj

FROM empleats e, projectes p

WHERE e.num\_proj = p.num\_proj;

se.num\_empl p.num\_proj p.nom\_proj

resultat

1 1 1 IBDTEL

3 1 IBDTEL

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		

Dades obtingudes com a resultat de la consulta. Cal notar que només apareixen aquelles combinacions en les que coincideix el número de projecte en què treballa l'empleat, amb el número de projecte del projecte.

### Consultes sobre més d'una taula Sintaxis alternativa – Clàusula Inner Join - Exemple 3

La condició de combinació de les taules es pot escriure:

- O bé en la clàusula WHERE
- O bé usant la clàusula JOIN en el FROM
  - INNER JOIN requereix la condició de combinació de manera explícita
  - NATURAL INNER JOIN fa la combinació per les columnes amb el mateix nom en les taules implicades.

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

FROM empleats e, projectes p

WHERE e.num_proj = p.num_proj;
```

```
SELECT e.num_empl, p.num_proj, p.nom_proj
FROM empleats e INNER JOIN projectes p ON e.num_proj = p.num_proj;
```

**SELECT** e.num\_empl, p.num\_proj, p.nom\_proj **FROM** empleats e **NATURAL INNER JOIN** projectes p;

Les tres sentències anteriors són totalment equivalents.



#### Consultes sobre més d'una taula: General

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

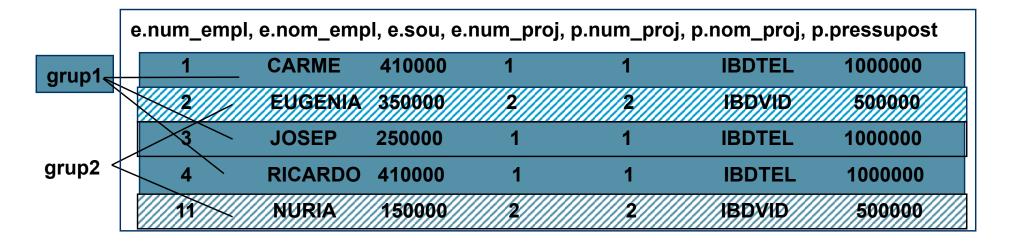


## Consultes sobre vàries taules: Exemple amb agrupació de files

SELECT p.num\_proj, p.nom\_proj
FROM projectes p, empleats e
WHERE p.num\_proj = e.num\_proj
GROUP BY p.num\_proj
HAVING p.pressupost < SUM(e.sou);</pre>

resultat p.num\_proj,p.nom\_proj

1 IBDTEL



Files resultants de la combinació dels empleats amb els projectes, agrupades segons la clàusula GROUP BY
Dades obtingudes com a resultat de la consulta. Un resultat per cada grup que compleix la condició del HAVING.
Només hi ha un projecte tal que el seu pressupost és inferior a la suma del sou dels empleats que hi treballen.

Cal notar que l'atribut p.nom\_proj pot estar entre els atributs del select perquè té un valor únic per cada grup. Igualment, l'atribut p.pressupost pot estar en la condició del having perquè té un valor únic per cada grup.

#### 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 clàusula ORDER\_BY han de ser un subconjunt de les columnes\_a\_seleccionar del primer SELECT.

## Consultes: Unió - Exemple

**SELECT** ciutat\_empl

FROM empleats

UNION

**SELECT** ciutat\_dpt

**FROM** departaments

ORDER BY ciutat\_empl DESC;

ciutat\_empl

SITGES

**MATARO** 

**MADRID** 

**BARCELONA** 

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

resultat

Dades obtingudes com a resultat de la consulta.



#### Consultes: Diferència

```
SELECT <columnes a seleccionar> | *
FROM < taules a consultar>
WHERE < columna taula > 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>);
```

- Hi ha dos formes alternatives de fer una diferència amb NOT IN o bé amb NOT EXISTS.
- Un NOT IN és cert si el valor de la columna columna\_taula no està en el resultat de la subconsulta.
- Un NOT EXISTS és cert si la subconsulta no dóna cap resultat.
- Hi ha altres maneres de fer una diferència (operador Except en SQL estàndard), encara que hi ha diferents noms de l'operador en diferents sistemes.

## **Consultes: Diferència – Exemples**

```
FROM projectes p

WHERE p.num_proj NOT IN (SELECT e.num_proj

FROM empleats e);

SELECT p.num_proj, p.nom_proj

FROM projectes p

WHERE NOT EXISTS (SELECT * FROM empleats e

WHERE p.num proj = e.num proj);
```

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

Dades obtingudes com a resultat de la consulta, en qualsevol de les dues alternatives.

En qualsevol cas, la consulta dóna aquells projectes que no tenen cap empleat assignat.

NOT IN: Un projecte és al resultat de la consulta si el seu número no està entre els números de projecte dels empleats. NOT EXISTS: Un projecte és al resultat de la consulta si no existeix cap empleat que tingui el seu número de projecte.

## Subconsultes en sentències Delete, Update i Select

Poden apareixer en aquelles sentències on hi ha la clàusula WHERE:

Esborrat de files d'una taulaDELETE 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 ..... );



## Subconsultes en sentències Delete - Exemple

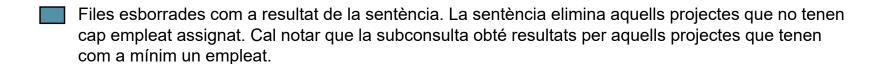
**DELETE FROM** projectes

WHERE NOT EXISTS (SELECT \*

FROM empleats e

**WHERE** e.num\_proj = projectes.num\_proj);

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





### Subconsultes en sentències Update - Exemple

**UPDATE** projectes

**SET** pressupost = pressupost + (pressupost \* 0,1)

WHERE 2 <= (SELECT COUNT(\*)

FROM empleats e

**WHERE** projectes.num\_proj = e.num\_proj);

empleats( <u>num_emp</u>	<u>l,</u> nom_empl,	num_proj))	projectes( <u>nu</u>	m_	<u>proj,</u> pressupost)
1	CARME	1		1	1100000
3	JOSEP	1		2	500000
				3	4500000
				4	2000000

Files modificades com a resultat de la sentència. La sentència puja el pressupost d'aquells projectes que tenen dos o més empleats assignats. Cal notar que la subconsulta el que fa és calcular el nombre d'empleats del projecte que s'està considerant si cal modificar o no.



### Subconsultes en sentències Select – Exemple 1

empleats( <u>n</u>	um_er	<u>npl,</u> nom_empl	, sou, ci	utat_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

Dades obtingudes com a resultat de la sentència. La sentència dóna els empleats que tenen un sou superior a la mitjana del sou de tots els empleats. Cal notar que la subconsulta retorna la mitjana del sou de tots els empleats de la taula empleats.

# **Subconsultes en sentències Select – Exemple 2**

FROM projectes p

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

FROM empleats e

WHERE e.num\_proj=p.num\_proj);

The sultat puants Proj Sobre Press

Quants Proj Sobre Press

quants Proj Sobre Press

projectes(num_proj,nom_proj,p.pressupost)		empleats(nu	ım_empl	, nom_emր	ol, sou, nu	ım_proj)	
1	IBDTEL	1000000		1	CARME	410000	1
2	IBDVID	500000		2//////	EUGENIA	350000	2////
				3	JOSEP	250000	1
				4	RICARDO	O 410000	1
					NURIA	150000	<u>//2</u> ////

- Dades obtingudes com a resultat de la sentència. La sentència dóna la quantitat de projectes que tenen un pressupost inferior a la suma del sou dels empleats que hi estan assignats. Cal notar que la subconsulta retorna la suma del sou dels empleats assignats al projecte que s'està considerant.
- Files seleccionades en la subconsulta quan es considera el projecte número 1. La suma dels sous és 1070000.
- Files seleccionades en la subconsulta quan es considera el projecte número 2. La suma dels sous és 500000

## Subconsultes en sentències Insert i clàusules Having

En aquelles sentències on hi ha la clàusula HAVING:

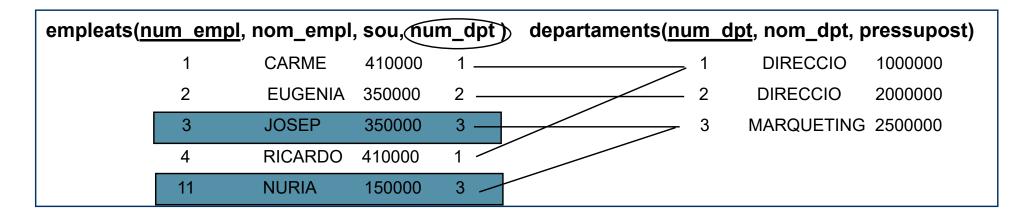
```
SELECT <columnes_a_seleccionar>
FROM <taules_a_consultar>
WHERE <condicions>
GROUP BY <columnes_agrupació>
HAVING ..... (SELECT ..... );
```

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 ..... );
```



# Subconsultes en clàusules Having – Exemple



- Dades obtingudes com a resultat de la sentència. La sentència dóna els departaments tals que la suma del sou dels seus empleats és superior a la suma del sou dels empleats del departament número 3. Cal notar que la subconsulta retorna la suma del sou dels empleats del departament número 3.
- Files seleccionades en la subconsulta sigui quin sigui el departament que es considera. La suma dels sous és 500000.

## Subconsultes en sentències Insert – Exemple

INSERT INTO clients
 (SELECT num\_empl,nom\_empl,200000
 FROM empleats
 WHERE num\_dpt IN (2,3));

