

2 CONSULTES SQL BÀSIQUES

SENTÈNCIA SELECT

OPERADORS ARITMÈTICS

VALOR NULL

ÀLIES

CONCATENACIÓ:

- Operador de concatenació d'ORACLE: ||
- Funció de concatenació: CONCAT()

LITERALS

DISTINCT: Eliminar files duplicades

COMENTARIS:

- ORACLE: --, /* ... */
- MySQL: --, /* ... */, #

SENTÈNCIA SELECT

```
SELECT * | {[DISTINCT] column|expression [alias], ... }  
FROM table
```

- **SELECT**: llistat de columnes que es mostraran (projecció).
- **FROM**: taula de la que s'obtenen les dades.

Un SELECT sempre ha de tenir clàusula FROM !

SELECCIONAR TOTES LES COLUMNS DE LA TAULA

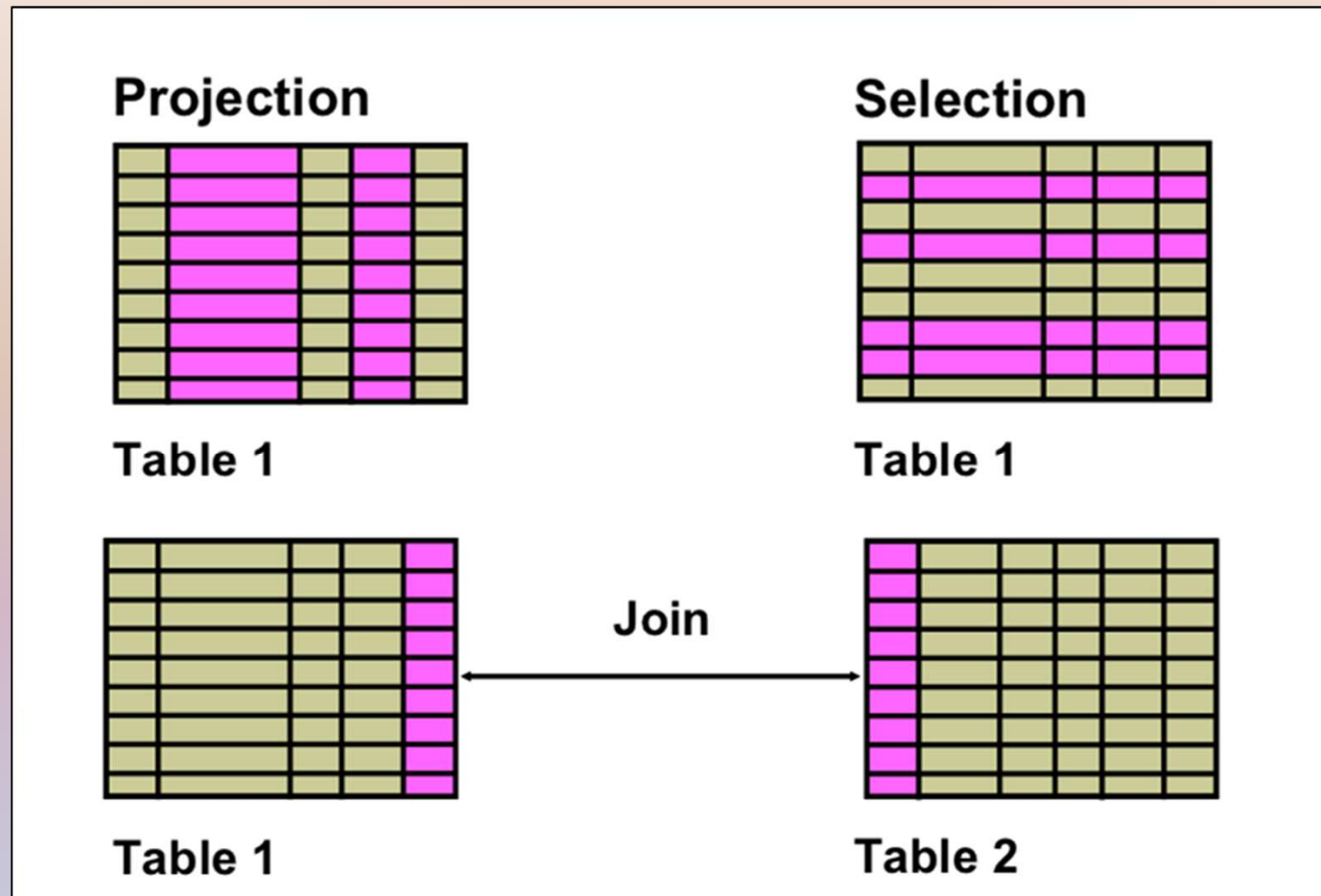
EXEMPLE

Mostra totes les dades de totes les columnes de la taula departaments:

```
SELECT *  
FROM DEPARTMENTS;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700
70	Public Relations	100	1700

SELECT: SELECCIÓ I PROJECCIÓ



SELECCIÓ DE COLUMNES (PROJECCIÓ)

EXAMPLE

Mostra totes les dades de les columnes **DEPARTMENT_ID** i **LOCATION_ID** de la taula **DEPARTMENTS**:

```
SELECT DEPARTMENT_ID, LOCATION_ID  
FROM DEPARTMENTS;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700
70	1700

REGLES SINTÀCTIQUES

- Les sentències SQL **NO** son **CASE-SENSITIVE**.
- Cada clàusula acostuma a ser situada en **línies diferents**. Millora la llegibilitat quan la sentència és llarga.
- Les sentències SQL poden acabar opcionalment en punt i coma (;): Però el ';' és obligatori quan s'executen múltiples sentències SQL.

OPERADORS ARIMTÈTICS

Es poden utilitzar operadors aritmètics per **calcular** els **valors** que es mostren en una **columna del SELECT**, en funció d'algun valor fix (*literal*) o de valors d'altres columnes de la taula.

Operador	Descripció
+	Suma
-	Resta
*	Multiplicació
/	Divisió

OPERADORS ARITMÈTICS

EXAMPLE

Mostrar el cognom, el salari mensual i el **salari anual** de tots els treballadors:

```
SELECT LAST_NAME, SALARY, 12*SALARY  
FROM EMPLOYEES;
```

LAST_NAME	SALARY	12*SALARY
King	24000	288000
Kochhar	17000	204000
De Haan	17000	204000
Whalen	4400	52800
Higgins	12000	144000
Gietz	8300	99600
Zlotkey	10500	126000
Abel	11000	132000
Taylor	8600	103200
Grant	7000	84000

PRECEDÈNCIA D'OPERADORS

```
SELECT LAST_NAME, SALARY, 12*SALARY+100  
FROM EMPLOYEES;
```

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
Whalen	4400	52900

```
SELECT LAST_NAME, SALARY, 12*(SALARY+100)  
FROM EMPLOYEES;
```

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200

VALOR NULL

NULL indica que el camp **no té valor**.

No confondre amb el 0 numèric o l'string buit ("") alfanumèric.

EXAMPLE - APEX

Mostrar el cognom de tots els empleats i si tenen comissió:

```
SELECT LAST_NAME, COMMISSION_PCT  
FROM EMPLOYEES;
```

LAST_NAME	COMMISSION_PCT
King	-
Kochhar	-
De Haan	-
Whalen	-
Higgins	-
Gietz	-
Zlotkey	.2
Abel	.3
Taylor	.2
Grant	.15

VALOR NULL

EXAMPLE – MariaDB - PhpMyAdmin

```
SELECT LAST_NAME, COMMISSION_PCT  
FROM EMPLOYEES;
```

LAST_NAME	COMMISSION_PCT
King	NULL
Kochhar	NULL
De Haan	NULL
Hunold	NULL
Ernst	NULL
Lorentz	NULL
Mourgos	NULL
Rajs	NULL
Davies	NULL
Matos	NULL
Vargas	NULL
Zlotkey	0.20
Abel	0.30
Taylor	0.20
Grant	0.15
Whalen	NULL
Hartstein	NULL
Fav	NULL

NULL EN EXPRESSIONS ARITMÀTIQUES

Operacions aritmètiques que continguin un valor **NULL** seran **avaluades a NULL**.

EXEMPLE

Mostrar cognom i salari anual, tenint en compte el percentatge de comissió

```
SELECT LAST_NAME, SALARY*(1+COMMISSION_PCT)
FROM EMPLOYEES;
```

LAST_NAME	SALARY*(1+COMMISSION_PCT)
King	-
Kochhar	-
De Haan	-
Whalen	-
Higgins	-
Gietz	-
Zlotkey	12600
Abel	14300

ÀLIES

Per defecte, el **nom d'una columna** del resultat és el nom de la columna de la taula o el nom de l'operació que genera les dades de la columna, per exemple *12*salary+100*.

L'àlies permet **canviar el nom** d'una columna del SELECT.

SINTAXI

```
SELECT columna | expressió [AS àlies]  
FROM taula;
```

ÀLIES

REGLES SINTÀCTIQUES

- L'àlies s'escriu **darrera del nom** de la columna.
- En alguns SGBDR, la paraula AS és opcional.
Millor posar-la sempre per compatibilitat de la consulta.
- S'ha de posar entre **cometes dobles** si conté espais o caràcters especials.
- És **CASE-SENSITIVE** (tal com s'escriu es mostra)

ÀLIES

EXAMPLE

Mostrar el cognom i la comissió de tots els empleats, amb els noms COGNOM i COMISSIÓ.

```
SELECT LAST_NAME AS "COGNOM", COMMISSION_PCT AS "COMISSIÓ"  
FROM EMPLOYEES;
```

COGNOM	COMISSIÓ
King	-
Kochhar	-
De Haan	-
Whalen	-
Higgins	-
Gietz	-
Zlotkey	.2
Abel	.3
Taylor	.2
Grant	.15

ÀLIES

EXEMPLE

Mostrar el nom i els salari anual com NOM i SALARI ANUAL

```
SELECT FIRST_NAME AS "NOM", 12*SALARY AS "SALARI ANUAL"  
FROM EMPLOYEES
```

NOM	SALARI ANUAL
Steven	288000
Neena	204000
Lex	204000
Jennifer	52800
Shelley	144000
William	99600
Eleni	126000
Ellen	132000
Jonathon	103200
Kimberely	84000

OPERADOR DE CONCATENACIÓ

(NO MySQL-Només ORACLE)

Es representa mitjançant dues barres verticals: ||

```
string1 || string2
```

Retorna l'string resultat de la unió dels dos strings.

EXAMPLE

Consulta que mostra nom i cognom de tots els empleats junts en una columna anomenada NOM_COMPLET.

```
SELECT FIRST_NAME || ' ' || LAST_NAME AS "NOM_COMPLET"  
FROM EMPLOYEES
```

NOM_COMPLET
Ellen Abel
Curtis Davies
Lex De Haan
Bruce Ernst

FUNCIÓ CONCATENACIÓ MariaDB

```
CONCAT(string1, string2, ...)
```

Retorna l'string resultat de la unió de tots els string que rebi com a paràmetre.

EXAMPLE

Consulta que mostra nom i cognom de tots els empleats junts en una columna anomenada NOM_COMPLET

```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS "NOM COMPLET"  
FROM employees
```

NOM COMPLET

Steven King

Neena Kochhar

Lex De Haan

Alexander Hunold

Bruce Ernst

Diana Lorentz

FUNCIÓ CONCATENACIÓ ORACLE

```
CONCAT (string1, string2)
```

Retorna l'string resultat de la unió dels **DOS** strings que rep com a paràmetre.

EXAMPLE

Consulta que mostra nom i cognom de tots els empleats junts en una columna anomenada NOM_COMPLET

```
SELECT CONCAT (FIRST_NAME, CONCAT (' ', LAST_NAME) )  
AS "NOM COMPLET"  
FROM employees
```

NOM COMPLET
Ellen Abel
Curtis Davies
Lex De Haan
Bruce Ernst
Pat Fay
William Gietz
Kimberely Grant
Michael Hartstein
Shelley Higgins
Alexander Hunold

Només comentar-la per què ha sortit la funció CONCAT de MariaDB.

Utilitzar operador de concatenació !!
Més senzill i fàcil de llegir !!

CADENES DE CARÀCTERS AMB LITERALS

L'exemple anterior utilitza un caràcter literal: ' '

```
SELECT FIRST_NAME || ' ' || LAST_NAME AS "NOM_COMPLET"  
FROM EMPLOYEES
```

Un literal és un **valor fixe**.

EXEMPLES DE LITERALS:

- Caràcter o string: 'dollars', '€'
- Numèric: 1000
- Data: '2021/11/25'

Els strings i les dates van **entre cometes simples**.

ELIMINAR FILES DUPLICADES

Quan es selecciona una columna d'una taula es mostren tots els valors.
Si un valor està repetit varis cops, es mostren totes les repeticions.

```
SELECT DEPARTMENT_ID  
FROM EMPLOYEES;
```

DEPARTMENT_ID
90
90
90
10
110
110
80
80
80
-

ELIMINAR FILES DUPLICADES

DISTINCT elimina les files duplicades del resultat.

```
SELECT DISTINCT DEPARTMENT_ID  
FROM EMPLOYEES;
```

DEPARTMENT_ID
50
110
90
10
-
20
60
80
8 rows returned in 0.01 seconds Download

ELIMINAR FILES DUPLICADES

EXEMPLE DE DISTINCT AMB MÉS D'UNA COLUMNA

Mostrar el número de detall i el número de producte de la taula detall de la BD empresa de MySQL.

```
SELECT DETALL_NUM, PROD_NUM  
FROM DETALL
```

Es mostren repetits el detall i el número de producte.

DETALL_NUM	PROD_NUM
2	100860
3	100860
1	100860
1	100860
1	100860
1	100860
1	100860
1	100860
2	100861
1	100861
1	100861
1	100861
2	100861
1	100861
1	100861
2	100861
2	100861

ELIMINAR FILES DUPLICADES

Resultat amb SELECT DISTINCT

```
SELECT DISTINCT DETALL_NUM, PROD_NUM  
FROM DETALL
```

Elimina les combinacions de detall i número de producte repetides.

DETALL_NUM	PROD_NUM
2	100860
3	100860
1	100860
2	100861
1	100861
1	100870
2	100870
3	100870
1	100871
2	100871
4	100871
3	100871
1	100890
3	100890
5	100890

DISTINCT elimina files repetides, tenint en compte **tots els camps** de la taula, no només el camp que segueix al DISTINCT.

COMENTARIS

ORACLE

```
--: comentaris d'una línia  
/* ... */: comentaris de més d'una línia
```

MySQL

```
SELECT 1+1; -- Comentari fins al final de línia  
SELECT 1+1; # Comentari fins al final de línia  
SELECT 1+1; Espai blanc comenta fins final de línia  
SELECT 1 /* Comentari de més d'una línia.  
           Es pot posar enmig de la sentència  
           */  
+1;
```