

# 3 SELECCIÓ FILES I ORDENACIÓ

## ORDENACIÓ DE FILES: ORDER BY

- Ordenar per un camp
- Ordenar per àlies
- Ordenar per número de columna
- Per més d'un camp

## SELECCIÓ DE FILES: WHERE

- Operadors de comparació
- Operador **BETWEEN**: rang de valors
- Operador de pertinença: **IN**
- Comparar caràcters: comodins (*wilcards*) (%) i \_ i operador **LIKE**
- Caràcter **ESCAPE**
- Operadors **IS NULL**, **IS NOT NULL**
- Operadors lògics: **AND**, **OR** i **NOT**

## LIMITAR EL NOMBRE DE FILES DEL RESULTAT

### ORACLE:

**FETCH FIRST** – **FETCH NEXT**

### MariaDB

**LIMIT** 6 **OFFSET** 0 – **LIMIT** 0,6

# ORDER BY

Segons la teoria del model relacional, el resultat d'una consulta és un **conjunt** de files. Com a conjunt, el resultat **no té ordre**.

A la pràctica, els SGBD sí que mostren el resultat “ordenat”:

- En **Oracle**, les files de les taules segueixen l'ordre en que s'insereixen.
- En **MariaDB**, el resultat de la consulta es mostra en ordre ascendent per la clau.

Es pot canviar l'ordre del resultat de la consulta amb la clàusula **ORDER BY**.

Si hi ha clàusula **ORDER BY**, ha de ser la **darrera sentència** del **SELECT**.

**NOTA:** Fem ORDER BY primer perquè com l'ordenació dels resultats és diferent en ORACLE i MariaDB, si ordenem és més fàcil comparar resultats.

# ORDER BY

Si l'ordre és **ascendent**:

- Si el camp és **numèric**: del valor més baix al més alt.
- Si el camp és **caràcter**: en ordre alfabètic.
- Si el camp és una **data**: de més antiga a més nova.
- L'ordre dels valors **NULL** depèn del SGBDR:
  - En **ORACLE** es mostren al final en ordre ascendent i al principi en ordre descendent.
  - En **MariaDB** es mostren al principi en ordre ascendent i al final de ordre descendent.

# ORDER BY

**ORDER BY {camp [ASC|DESC] [ NULLS FIRST | NULLS LAST]}**

- **Camp** indica el camp pel que s'ordena.
- **ASC|DESC**: Indica el tipus d'ordenació. Ascendent per defecte.
- **NULLS FIRST | NULLS LAST**: Si els valors NULL es mostren al principi o al final. No tots els SGBD ! **MySQL no !**

	DB2	MySQL	Oracle	PostgreSQL	SQLite	SQL Server
Lectura de índex hacia atrás	✓	✓	✓	✓	✓	✓
Order by ASC/DESC	✓	✓	✓	✓	✓	✓
Index ASC/DESC	✓	✗	✓	✓	✓	✓
Order by NULLS FIRST/LAST	✓	✗	✓	✓	✗	✗
Default NULLS order	Great	Small	Great	Great	Small	Small
Index NULLS FIRST/LAST	✗	✗	✗	✓	✗	✗

Es pot ordenar per un camp que no aparegui al SELECT !

# ORDER BY

## EXEMPLE

Mostra l'id, el cognom i la data de contractació de tots els empleats, ordenats per data de contractació, de més nou a més antic.

```
SELECT EMPLOYEE_ID, LAST_NAME, HIRE_DATE
FROM EMPLOYEES
ORDER BY HIRE_DATE DESC
```

EMPLOYEE_ID	LAST_NAME	HIRE_DATE
149	Zlotkey	01/29/2000
124	Mourgos	11/16/1999
178	Grant	05/24/1999
107	Lorentz	02/07/1999
144	Vargas	07/09/1998
176	Taylor	03/24/1998
143	Matos	03/15/1998
202	Fay	08/17/1997
142	Davies	01/29/1997
174	Abel	05/11/1996
201	Hartstein	02/17/1996

EMPLOYEE_ID	LAST_NAME	HIRE_DATE
149	Zlotkey	2000-01-29
124	Mourgos	1999-11-16
178	Grant	1999-05-24
107	Lorentz	1999-02-07
144	Vargas	1998-07-09
176	Taylor	1998-03-24
143	Matos	1998-03-15
202	Fay	1997-08-17
142	Davies	1997-01-29
174	Abel	1996-05-11
201	Hartstein	1996-02-17

# ORDER BY

## EXEMPLE

Mostra l'id, el cognom i el bonus dels empleats

```
SELECT EMPLOYEE_ID, LAST_NAME, BONUS  
FROM EMPLOYEES  
ORDER BY BONUS
```

En ordre ascendent els NULL es mostren  
la final

EMPLOYEE_ID	LAST_NAME	BONUS
176	Taylor	1250
149	Zlotkey	1500
174	Abel	1700
200	Whalen	-
205	Higgins	-
206	Gietz	-
178	Grant	-
124	Mourgos	-
141	Rajs	-
142	Davies	-

144	Vargas	-
103	Hunold	-
104	Ernst	-
176	Taylor	1250
149	Zlotkey	1500
174	Abel	1700

**ORDER BY BONUS NULLS FIRST**

Mostra NULLS al principi  
(Compte que n'hi ha més de 10)

# ORDER BY – ORDENAR PER ÀLIES

## EXEMPLE

Ordenació utilitzant l'àlies de la columna funciona bé en ORACLE, però no en MariaDB:

```
SELECT LAST_NAME, 12*SALARY AS "SALARI ANUAL"  
FROM employees  
ORDER BY "SALARI ANUAL";
```

LAST_NAME	SALARI ANUAL
Vargas	30000
Matos	31200
Davies	37200
Rajs	42000
Lorentz	50400
Whalen	52800
Mourgos	69600
Fay	72000
Ernst	72000

LAST_NAME	SALARI ANUAL
King	288000.00
Kochhar	204000.00
De Haan	204000.00
Hunold	108000.00
Ernst	72000.00
Lorentz	50400.00
Mourgos	69600.00
Rajs	42000.00
Davies	37200.00
Matos	31200.00
Vargas	30000.00
Zlotkey	126000.00



# ORDER BY – NÚMERO DE COLUMNA

## EXEMPLE

Ordenació utilitzant el **número de columna**:

```
SELECT LAST_NAME, JOB_ID, DEPARTMENT_ID, HIRE_DATE  
FROM EMPLOYEES  
ORDER BY 3;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Whalen	AD_ASST	10	09/17/1987
Fay	MK_REP	20	08/17/1997
Hartstein	MK_MAN	20	02/17/1996
Rajs	ST_CLERK	50	10/17/1995
Vargas	ST_CLERK	50	07/09/1998
Mourgos	ST_MAN	50	11/16/1999

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Grant	SA_REP	NULL	1999-05-24
Whalen	AD_ASST	10	1987-09-17
Fay	MK_REP	20	1997-08-17
Hartstein	MK_MAN	20	1996-02-17
Matos	ST_CLERK	50	1998-03-15
Davies	ST_CLERK	50	1997-01-29
Rajs	ST_CLERK	50	1995-10-17

Dintre del mateix DEPARTMENT\_ID, l'ordre dels registres és diferent: ORACLE per ordre d'inserció, MariaDB per la clau → SEGONA ordenació per LAST\_NAME

Si afegim o eliminem camps de la consulta es pot modificar el número de columna per la que ordenem !!!



# ORDER BY – MÉS D'UN CAMP

## EXEMPLE

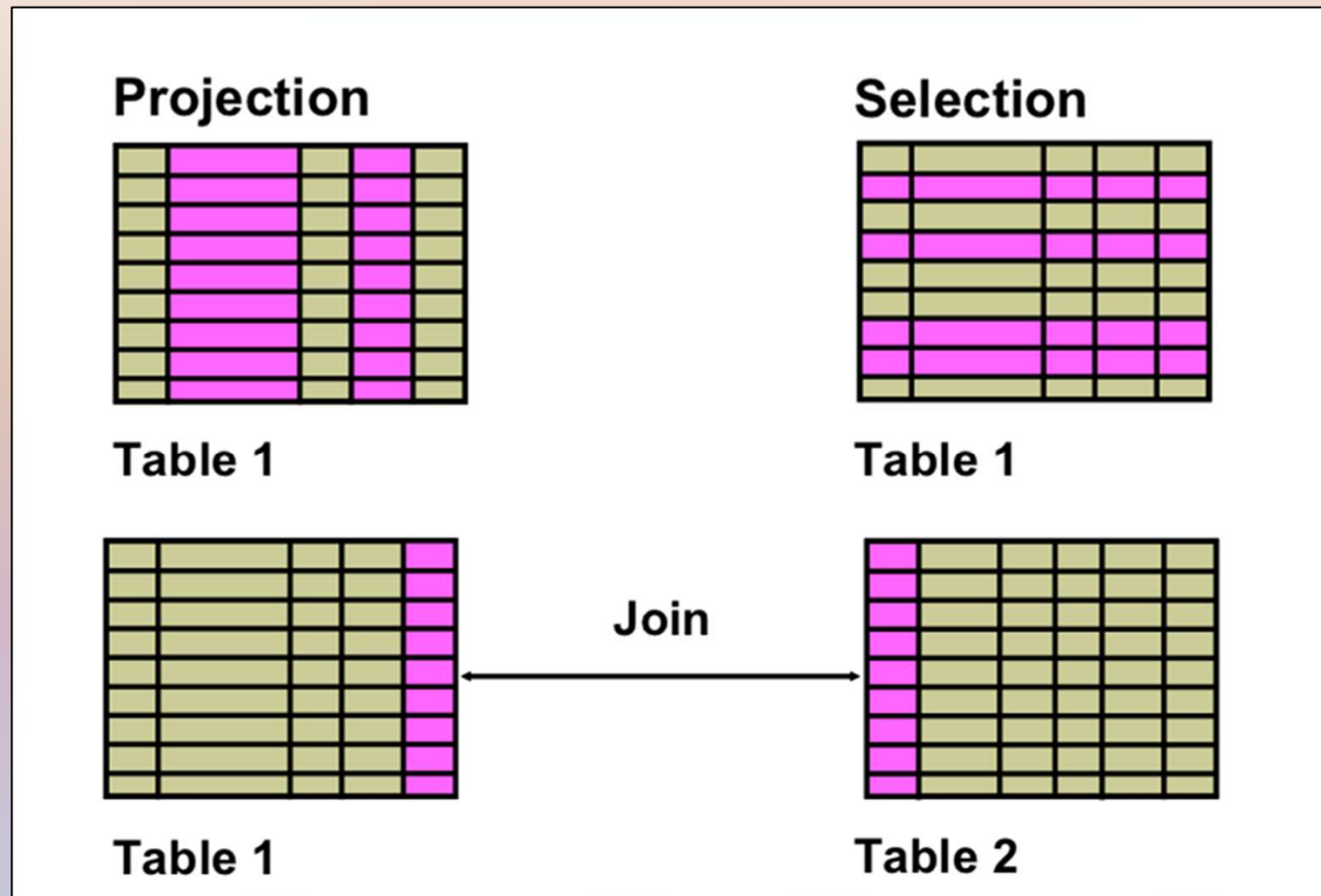
Ordenem per DEPARTMENT\_ID, i dintre del mateix DEPARTMENT\_ID, per cognom:

```
SELECT LAST_NAME, JOB_ID, DEPARTMENT_ID, HIRE_DATE
FROM EMPLOYEES
ORDER BY DEPARTMENT_ID, LAST_NAME;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Whalen	AD_ASST	10	09/17/1987
Fay	MK_REP	20	08/17/1997
Hartstein	MK_MAN	20	02/17/1996
Davies	ST_CLERK	50	01/29/1997
Matos	ST_CLERK	50	03/15/1998
Mourgos	ST_MAN	50	11/16/1999
Rajs	ST_CLERK	50	10/17/1995
Vargas	ST_CLERK	50	07/09/1998

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Grant	SA_REP	NULL	1999-05-24
Whalen	AD_ASST	10	1987-09-17
Fay	MK_REP	20	1997-08-17
Hartstein	MK_MAN	20	1996-02-17
Davies	ST_CLERK	50	1997-01-29
Matos	ST_CLERK	50	1998-03-15
Mourgos	ST_MAN	50	1999-11-16
Rajs	ST_CLERK	50	1995-10-17
Vargas	ST_CLERK	50	1998-07-09

# SELECT: SELECCIÓ I PROJECCIÓ



# SELECCIÓ DE FILES: FILTRAR

```
SELECT columnes  
FROM taula  
[WHERE condició];
```

La clàusula **WHERE** indica una **condició** (simple o composta) que han de complir les files seleccionades.

## EXAMPLE

Mostra tots els treballadors del departament 110

```
SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID=110
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
205	Higgins	110
206	Gietz	110

# SELECCIÓ DE FILES

**No** es poden utilitzar **àlies** al WHERE. Només admet noms de columnes.

```
SELECT EMPLOYEE_ID, LAST_NAME,  
       DEPARTMENT_ID AS DEPARTMENT  
FROM EMPLOYEES  
WHERE DEPARTMENT=110
```

Considera l'àlies DEPARTMENT com el nom d'una columna i mostra un missatge d'error per què no hi ha cap columna DEPARTMENT a la taula EMPLOYEES.

MySQL diu: ?

#1054 - Unknown column 'DEPARTMENT' in 'where clause'



ORA-00904: "DEPARTMENT": invalid identifier

# OPERADORS DE COMPARACIÓ

OPERADOR	SIGNIFICAT
=	Igual
>	Més gran que
>=	Més gran o igual a
<	Més petit que
<=	Més petit o igual que
<> (!= o ^=) (MySQL !=)	Diferent a
[NOT] BETWEEN... AND ...	Entre dos valors (inclusiu)
[NOT] IN (conjunt)	Coincideix amb una llista de valors
[NOT] LIKE	Coincideix amb un patró d'un caràcter
IS NULL	És valor null

Qualsevol comparació amb **NULL** és **falsa**, inclús NULL=NULL.

# OPERADORS COMPARACIÓ

## EXAMPLE

Mostra el cognom i el salari de tots els empleats que tenen un salari inferior a 3000:

```
SELECT LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE SALARY <= 3000;
```

LAST_NAME	SALARY
Matos	2600
Vargas	2500

# RANG DE VALORS: OPERADOR BETWEEN

**BETWEEN** selecciona un rang de valors entre un límit superior i un límit superior, ambdós inclosos.

## EXAMPLE

Mostra cognom i salari de tots els empleats que tenen un salari entre 2500 i 3500 €

```
SELECT LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE SALARY BETWEEN 2500 AND 3500;
```

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500



# RANG DE VALORS: OPERADOR BETWEEN

Les dues condicions serien equivalents:

```
WHERE SALARY BETWEEN 2500 AND 3500;
```

```
WHERE SALARY >= 2500  
AND  
SALARY <= 3500;
```

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

# OPERADOR DE PERTINENÇA: IN

L'operador **IN** comprova si un valor pertany a un conjunt de valors.

## EXAMPLE

Mostra l'id, el cognom i el manager\_id de tots els empleats que tenen com a manager\_id el 100, 101 o 201

```
SELECT EMPLOYEE_ID, LAST_NAME, MANAGER_ID  
FROM EMPLOYEES  
WHERE MANAGER_ID IN (100, 101, 201)  
ORDER BY EMPLOYEE_ID;
```

EMPLOYEE_ID	LAST_NAME	MANAGER_ID
101	Kochhar	100
102	De Haan	100
124	Mourgos	100
149	Zlotkey	100
200	Whalen	101
201	Hartstein	100
202	Fay	201
205	Higgins	101

8 rows returned in 0.01 seconds [Download](#)

# OPERADOR DE PERTINEÇA: IN

Les dues consultes són equivalents:

```
SELECT EMPLOYEE_ID, LAST_NAME, MANAGER_ID  
FROM EMPLOYEES  
WHERE MANAGER_ID IN (100, 101, 201);
```

```
SELECT EMPLOYEE_ID, LAST_NAME, MANAGER_ID  
FROM EMPLOYEES  
WHERE MANAGER_ID = 100  
      OR  
      MANAGER_ID = 101  
      OR  
      MANAGER_ID = 201;
```

# OPERADOR DE PERTINEÇA: IN

## EXAMPLE (IN amb caràcters)

Mostra el *location\_id*, la *ciutat* i la *província* de totes les « locations » que tinguin com a *country\_id* CA (Canadà) o UK (United Kingdom).

```
SELECT LOCATION_ID, CITY, STATE_PROVINCE  
FROM LOCATIONS  
WHERE COUNTRY_ID IN ('CA', 'UK');
```

LOCATION_ID	CITY	STATE_PROVINCE
1800	Toronto	Ontario
2500	Oxford	Oxford

L'operador IN seria equivalent a la condició:

```
COUNTRY_ID='CA' OR COUNTRY_ID='UK'
```

Com és = i no LIKE no es poden utilitzar COMODINS en un IN !!

# COMPARACIÓ DE CARÀCTERS

- Els **caràcters** van entre **cometes simples**.
- Els caràcters són **case-sensitive** en **ORACLE**, però no en MySQL.  
Hi ha SGBD que són CASE-SENSITIVE i d'altres que no. Per que les consultes siguin el més "*compatibles*" possible es recomana utilitzar funcions de conversió de caràcters a majúscules o minúscules al comparar.
- Per comparar caràcters es pot utilitzar l'operador **=** o l'operador **LIKE**.

# COMPARACIÓ DE CARÀCTERS

## EXEMPLE ORACLE

Mostra nom i cognom de l'empleat que té per nom **Taylor**:

### CONSULTES EQUIVALENTS

```
SELECT FIRST_NAME, LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME = 'Taylor'
```

```
SELECT FIRST_NAME, LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME LIKE 'Taylor'
```

## ORACLE

FIRST_NAME	LAST_NAME
Jonathon	Taylor

## MariaDB

FIRST_NAME	LAST_NAME
Jonathon	Taylor

# COMPARACIÓ DE CARÀCTERS

## EXEMPLE

Modifiquem la consulta posant totes les lletres en minúscula:

```
SELECT FIRST_NAME, LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME = 'taylor'
```

## ORACLE

no data found

## MariaDB

FIRST_NAME	LAST_NAME
Jonathon	Taylor



# CARÀCTERS COMODÍ (*WILCARDS*)

Caràcters comodins:

%: representa 0 més caràcters

\_: representa un caràcter

Per comparar amb cadenes amb comodins: l'operador **LIKE**.

- **LIKE 'Torres'**: cadena 'Torres'.
- **LIKE 'Torr%'**: qualsevol cadena iniciada per 'Torr'.
- **LIKE '%S%'**: qualsevol cadena que contingui 'S'.
- **LIKE '\_o%'**: qualsevol cadena que tingui com a segon caràcter una 'o'.

# CARÀCTER ESCAPE - MariaDB

Si necessitem buscar una cadena que tingui **caràcters especials** (ie. comodins o ') com es pot fer que algun dels caràcters **no s'interpreti** i d'altres sí ??

Utilitzem el caràcter **ESCAPE**: '\'

## EXAMPLE

Mostra el *nom* i el *job\_id* de tots els empleats per als que el job\_id comença amb 'AD\_':

```
SELECT last_name, job_id  
FROM employees  
WHERE job_id LIKE 'AD\_%';
```

last_name	job_id
Whalen	AD_ASST
King	AD PRES
Kochhar	AD_VP
De Haan	AD_VP

# CARÀCTER ESCAPE - ORACLE

Utilitzem la paraula clau **ESCAPE**.

## EXAMPLE

Mostra el *nom* i el *job\_id* de tots els empleats per als que el job\_id comença amb 'AD\_':

```
SELECT last_name, job_id  
FROM employees  
WHERE job_id LIKE 'AD\_%' ESCAPE '\';
```

LAST_NAME	JOB_ID
Whalen	AD_ASST
King	AD PRES
Kochhar	AD_VP
De Haan	AD_VP

4 rows returned in 0.00 seconds [Download](#)

Es pot utilitzar qualsevol caràcter com a caràcter ESCAPE !

# CONDICIONS NULL

Operadors específics per comprovar si un valor és NULL (**IS NULL**) o no (**IS NOT NULL**).

## EXAMPLE

Mostra el nom complet i la comissió de tots els empleats que tinguin comissió:

```
SELECT FIRST_NAME, LAST_NAME, COMMISSION_PCT  
FROM EMPLOYEES  
WHERE COMMISSION_PCT IS NOT NULL;
```

FIRST_NAME	LAST_NAME	COMMISSION_PCT
Eleni	Zlotkey	.2
Ellen	Abel	.3
Jonathon	Taylor	.2
Kimberely	Grant	.15

4 rows returned in 0.00 seconds [Download](#)

# OPERADORS LÒGICS

OPERADOR	SIGNIFICAT
<b>AND</b>	Retorna CERT si totes dues condicions són certes
<b>OR</b>	Retorna CERT si almenys una condició es certa
<b>NOT</b>	Retorna CERT si la condició es falsa.

# OPERADOR AND

Per què una condició amb **AND** sigui **certa**, **totes les condicions** que la formen han de ser **certes**.

## EXAMPLE

Mostra l'id d'empleat, cognom, job\_id i salari de tots empleats que cobrin 10000 o més i que treballin de managers (job\_id acaba amb MAN)

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
FROM EMPLOYEES
WHERE SALARY >= 10000 AND JOB_ID LIKE '%MAN';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

# OPERADOR OR

Per què una condició amb **OR** sigui **certa**, **almenys una** de les condicions que la formen ha de ser **certa**.

## EXAMPLE

Mostra l'id d'empleat, cognom, job\_id i salari de tots empleats que cobrin 10000 o més o que treballin de managers (job\_id acaba amb MAN)

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, SALARY
FROM EMPLOYEES
WHERE SALARY >= 10000 OR JOB_ID LIKE '%MAN'
ORDER BY EMPLOYEE_ID;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
124	Mourgos	ST_MAN	5800
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
205	Higgins	AC_MGR	12000

8 rows returned in 0.01 seconds [Download](#)



# OPERADOR NOT

Nega una condició.

## EXAMPLE

Mostra el cognom i el job\_id dels treballadors que **no** siguin IT\_PROG, ST\_CLERCK ni SA\_REP.

```
SELECT LAST_NAME, JOB_ID
FROM EMPLOYEES
WHERE JOB_ID NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP')
ORDER BY LAST_NAME
```

LAST_NAME	JOB_ID
De Haan	AD_VP
Fay	MK_REP
Gietz	AC_ACCOUNT
Hartstein	MK_MAN
Higgins	AC_MGR
King	AD_PRES
Kochhar	AD_VP
Mourgos	ST_MAN
Whalen	AD_ASST
Zlotkey	SA_MAN

10 rows returned in 0.00 seconds [Download](#)

# REGLES DE PRECEDÈNCIA

Determinen l'**ordre** en el que s'avaluen les expressions.

PRECEDÈNCIA	OPERADOR
1	Aritmètics
2	Concatenació
3	Condicions de comparació
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Diferent
7	NOT
8	AND
9	OR

Es poden utilitzar **parèntesis** per sobreescriure les regles de precedència, o simplement per millorar la llegibilitat.

# LIMITAR NOMBRE FILES RESULTAT

## ORACLE

**EXAMPLE:** Mostrar només les 6 primeres files del resultat.

```
SELECT employee_id, first_name, last_name
FROM EMPLOYEES
ORDER BY FIRST_NAME
FETCH FIRST 6 ROWS ONLY;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
104	Bruce	Ernst
107	Diana	Lorentz
6 rows returned in 0.01 seconds		<a href="#">Download</a>

# LIMITAR NOMBRE FILES RESULTAT

## ORACLE

**EXAMPLE:** Mostrar de la fila 4 a la 10 del resultat.

```
SELECT employee_id, first_name, last_name  
FROM EMPLOYEES  
ORDER BY FIRST_NAME  
OFFSET 3 ROWS  
FETCH NEXT 7 ROWS ONLY;
```

Salta les 3 primeres, per començar a la 4

Si mostra de la 4 a la 10 són 7 files

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
103	Alexander	Hunold
104	Bruce	Ernst
107	Diana	Lorentz
124	Kevin	Mourgos
141	Trenna	Rajs
142	Curtis	Davies
143	Randall	Matos

# LIMITAR NOMBRE FILES RESULTAT

## MariaDB

```
LIMIT 10 OFFSET 15  -- LIMIT 15, 10
```

Retorna 10 registres a partir del 15: del 15 al 24.

**EXAMPLE:** Mostra els 6 primers registres del resultat

```
SELECT CLIENT_COD, NOM  
FROM client  
ORDER BY FIRST_NAME  
LIMIT 0, 6
```

ORDER BY no és la darrera sentència del SELECT, sinó LIMIT