



## 2- Sentències SELECT bàsiques

NF1 - Llenguatge SQL: DML

UF2 – Llenguatge SQL: DML i DDL

## Que veurem?

1. Consulta a dades
  - Sentència SELECT
  - Clàusules de SELECT
  - Selecció de columnes
  - Selecció de files
2. Operadors Aritmètics
3. Operadors de comparació i lògics
4. Operadors de comparació de cadenes de caràcters
5. NULL i NOT NULL
6. Comprovacions amb conjunt de valors
7. Comprovacions d'operadors AND i OR



## 1. Consulta a dades

- ☐ Per recuperar informació o per realitzar consultes a la base de dades farem servir una única sentència SELECT.
- ☐ L'usuari fa servir aquesta sentència per indicar el que vol obtenir i no on ni com.
- ☐ De la consulta es pot obtenir: qualsevol unitat de dades, totes les dades, qualsevol subconjunt de dades, qualsevol conjunt de subconjunts de dades.

## Sentència SELECT

El format de la sentència SELECT és el següent:

```
SELECT [ALL/DISTINCT ]  
[expre_colum1, expre_colum2, ..., expre_column | * ]  
FROM [nom_taula1, nom_taula2, ..., nom_taula n]  
[WHERE condició]  
[ORDER BY expre_colum [DESC|ASC] [,expre_colum  
[DESC|ASC]]...];
```

Les sentències SQL acostumen a acabar en punt i coma (;)



## 1. Consulta a dades (II)

### Clàusules SELECT

- ❑ L'única clàusula de la sentència SELECT que és obligatòria és FROM, la resta són opcionals.

❖ **FROM:** FROM [nom\_taula1, nom\_taula2, ..., nom\_taula n]

Especifica la taula o llista de taules de les que es recuperaran les dades.

- ❑ Especifica la taula o llista de taules de les quals es recuperaran les dades.

➤ **Exemple**, consultem el *country\_name*, *region\_id* a la taula **COUNTRIES**:

```
SELECT country_name, region_id  
FROM countries;
```

<u>country_name</u>	<u>region_id</u>
Belgium	1
Switzerland	1
Germany	1
Denmark	1
France	1
Italy	1
Netherlands	1
United Kingdom	1
Argentina	2
Brazil	2



## Clàusules SELECT (II)

- ❑ És possible associar un nou nom a les taules mitjançant àlies.
- **Exemple**, si la taula COUNTRIES li donem el nom de C (àlies), les columnes de la taula aniran acompanyades de C. Exemple, consultem el *country\_name*, *region\_id* a la taula **COUNTRIES**:

```
SELECT c.country_name, c.region_id  
FROM countries c;
```

A la taula countries li hem donat un àlies. Els àlies acostumen a ser les inicials del nom de la taula.

<u>country_name</u>	<u>region_id</u>
Belgium	1
Switzerland	1
Germany	1
Denmark	1
France	1
Italy	1
Netherlands	1
United Kingdom	1
Argentina	2
Brazil	2



## Clàusules SELECT (III)

### ❖ **WHERE** [WHERE CONDICIÓN]

Obté les files que compleixen la condició expressada. La complexitat de la condició és pràcticament il·limitada.

El format de la condició és: ***expressió operador expressió***

- Les expressions poden ser una constant, una expressió aritmètica, un valor nul, o el nom d'una columna.
- Els operadors de comparació són els següents:

= , > , < , >= , <= , != , <>

IN , NOT IN , BETWEEN , NOT BETWEEN

LIKE

- Es poden construir condicions múltiples fent servir els operadors lògics booleans estàndards: AND, OR i NOT.
- Està permès emprar parèntesi per forçar l'ordre d'avaluació.



## Clàusules SELECT (IV)

### ❖ **WHERE** [WHERE CONDICIÓN]

- **Ex1** – Mostrar de la taula país aquells països on el codi de regió sigui 3.

```
SELECT country_name  
FROM countries  
WHERE region_id=3;
```

country_name	region_id
Australia	3
China	3
HongKong	3
India	3
Japan	3
Singapore	3



## Clàusules SELECT (V)

### WHERE [WHERE CONDICIÓN]

- **Ex2**– Mostrar empleats, les dades a mostrar són (id\_empleat, nom i cognom), d'aquells empleats que tenen un salari superior o igual a 10000 i que l'ID del seu manager és el 100.

```
SELECT e.employee_id, e.last_name, e.first_name, e.salary, e.manager_id
FROM   employees e
WHERE  e.salary>10000 and e.manager_id=100;
```

employee_id	last_name	first_name	salary	manager_id
101	Kochhar	Neena	17000.00	100
102	De Haan	Lex	17000.00	100
114	Raphaely	Den	11000.00	100
145	Russell	John	14000.00	100
146	Partners	Karen	13500.00	100
147	Errazuriz	Alberto	12000.00	100
148	Cambrault	Gerald	11000.00	100
149	Zlotkey	Eleni	10500.00	100
201	Hartstein	Michael	13000.00	100





## Clàusules SELECT (VI)

### ❖ **WHERE** [WHERE CONDICIÓN]

- **Ex3**– Mostrar els empleats (id\_empleat, nom, cognom i comissió) d'aquells empleats que tinguin comissió aplicada.

```
SELECT e.employee_id, e.last_name, e.first_name, e.salary, e.commission_pct  
FROM   employees e  
WHERE  commission_pct is not null;
```

employee_id	last_name	first_name	salary	commission_pct
145	Russell	John	14000.00	0.40
146	Partners	Karen	13500.00	0.30
147	Errazuriz	Alberto	12000.00	0.30
148	Cambrault	Gerald	11000.00	0.30
149	Zlotkey	Eleni	10500.00	0.20
150	Tucker	Peter	10000.00	0.30
151	Bernstein	David	9500.00	0.25
152	Hall	Peter	9000.00	0.25



## Clàusules SELECT (VII)

❖ **ORDER BY** [**ORDER BY** expre\_columna [DESC|ASC] [,expre columna [DESC|ASC]]...]

➤ Aquesta clàusula especifica el criteri de classificació del resultat de la consulta. ASC especifica una ordenació ascendent, i DESC descendent.

➤ Pot contenir expressions amb valors de columnes, per exemple:

**Ex1** – Mostrar els empleats (id\_empleat, nom i cognom a una columna que s'anomeni nom\_empleat) ordenats pel seu departament.

```
SELECT e.employee_id, concat(e.last_name, ', ', e.first_name) as nom_empleat,  
       e.department_id  
FROM   employees e  
ORDER BY e.department_id;
```

employee_id	nom_empleat	depart_id
178	Grant, Kimberly	
200	Whalen, Jennifer	10
201	Hartstein, Michael	20
202	Fay, Pat	20
114	Raphaely, Den	30
115	Khoo, Alexander	30
116	Baida, Shelli	30
117	Tobias, Sigal	30
119	Colmenares, Karen	30



## Clàusules SELECT (VIII) (order by)

- Així mateix, és possible niar els criteris. El situat més a l'esquerra serà el principal. Per exemple:
- **Ex2** – Mostrar els empleats (id\_empleat, nom i cognom a una columna que s'anomeni total\_empleats) ordenats pel seu departament (ascendent) i pel seu id de treball (descendent).

```
SELECT e.employee_id, concat(e.last_name, ', ', e.first_name) as nom_empleat,  
       e.department_id, e.job_id  
FROM   employees e  
ORDER BY e.department_id, e.job_id desc;
```

employee_id	nom_empleat	department_id	job_id
178	Grant, Kimberly		SA_REP
200	Whalen, Jennifer	10	AD_ASST
202	Fay, Pat	20	MK_REP
201	Hartstein, Michael	20	MK_MAN
114	Raphaely, Den	30	PU_MAN
115	Khoo, Alexander	30	PU_CLERK
116	Baida, Shelli	30	PU_CLERK



## Clàusules SELECT (IX) (order by)

- També es pot indicar mitjançant un nombre, que indica la posició de la columna a la dreta de SELECT, el criteri de classificació.
- **Ex3** – La següent sentència ordena els resultats per la segona columna.

```
SELECT e.employee_id, concat(e.last_name, ', ', e.first_name) as nom_empleat  
FROM   employees e  
ORDER BY 2 desc;
```

employee_id	nom_empleat
149	Zlotkey, Eleni
200	Whalen, Jennifer
120	Weiss, Matthew
196	Walsh, Alana
123	Vollman, Shanta
162	Vishney, Clara
144	Vargas, Peter
112	Urman, Jose Manuel
155	Tuvault, Oliver



## Clàusules SELECT (X) (Order by)

- ✓ **ALL:** Amb la clausula ALL recuperem totes les files, encara que algunes estiguin repetides. És l'opció per defecte i no cal explicitar-la.
- ✓ **DISTINCT:** Només recupera les files que són diferents.
  - **Ex4** – Si mostrem tots els id\_departaments de la taula empleats, apareixen totes les files de la taula EMPLOYEES on la columna DEPARTMENT\_ID no sigui nul·la; apareixen departament (department\_id) repetits.

```
SELECT e.department_id  
FROM employees e;
```

department_id
10
20
20
30
30
30
30
30

Amb DISTINCT s'eliminen les repeticions

```
SELECT DISTINCT e.department_id  
FROM employees e;
```

department_id
10
20
30



## SELECCIÓ DE COLUMNES

- ❑ El format de SELECT que ens permet seleccionar columnes d'una taula és el següent:

```
SELECT [ALL/DISTINCT ]  
[expre_colum1, expre_colum2, ..., expre_column | * ]  
FROM [nom_taula1, nom_taula2, ..., nom_taula n];
```

- ❑ Anem a veure exemples de consultes sobre les taules EMPLOYEES i DEPARTMENT.
- ❑ Per saber quina és l'estructura de la nostra taula (taula i les seves columnes) farem servir la comanda **DESC** nom taula:

Field	Type	Null	Key	Default	Extra
employee_id	int unsigned	NO	PRI		
first_name	varchar(20)	YES			
last_name	varchar(25)	NO			
email	varchar(25)	NO			
phone_number	varchar(20)	YES			
hire_date	date	NO			
job_id	varchar(10)	NO	MUL		
salary	decimal(8,2)	NO			
commission_pct	decimal(2,2)	YES			
manager_id	int unsigned	YES	MUL		
department_id	int unsigned	YES	MUL		



## SELECCIÓ DE COLUMNES

NOT NULL	Significa que la columna no pot tenir valors nuls.
VARCHAR2	Tipus de dades cadena de longitud variable. Pot contenir qualsevol caràcter. La longitud màxima de la cadena es defineix entre parèntesi.
DATE	Tipus de dades data.
NUMBER	Tipus de dades numèric. El nombre de dígit s'expressa entre parèntesi

- ❑ Per a la selecció de totes les columnes de la taula EMPLOYEES, podem recuperar les files de dues formes:

1. Posem els noms de totes les columnes, un darrere l'altre, separats per comes:

```
SELECT e.employee_id, e.first_name, e.last_name, e.email, e.phone_number  
FROM employees e;
```

2. Posem \*, que representa a totes les columnes de la taula:

```
SELECT e.*  
FROM employees e;
```



## CREAR I UTILITZAR ALIES DE COLUMNES

- ❑ Quan es consulta la base de dades, els noms de les columnes s'usen com a capçaleres del resultat.
- ❑ Si el nom resulta massa llarg, curt o críptic, existeix la possibilitat de canviar-ho creant un ÀLIES.
- ❑ L'ÀLIES es posa entre cometes, a la dreta del nom de la columna.

```
SELECT e.employee_id AS 'NUM_EMP', e.first_name AS 'NOM', e.last_name AS 'COGNOM'  
FROM   employees e;
```

NUM_EMP	NOM	COGNOM
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
104	Bruce	Ernst
105	David	Austin
106	Valli	Pataballa





## 2. OPERADORS ARITMÈTICS

- Els operadors aritmètics serveixen per formar expressions amb constants, valors de columnes i funcions de valors de columnes.

Operador aritmètic	Operació
+	Suma
-	Resta
*	Multiplicació
/	Divisió

- **Ex1** → Calcular la comissió final de l'empleat amb id\_empleat 171. El resultat ha de quedar reflectit a una columna que anomenarem SAL/COM

```
SELECT e.salary/e.commission_pct AS 'SAL/COM'  
FROM employees e  
WHERE e.employee_id=171;
```

**SAL/COM**

49333.333333



### 3. OPERADORS DE COMPARACIÓ I LÒGICS

□ La següent taula mostra els operadors de comparació:

Operador comparatiu	Operació
=	Igual a
>	Major que
>=	Major o igual que
<	Menor que
<=	Menor o igual que
!= ó <>	Diferent de



### 3. OPERADORS DE COMPARACIÓ I LÒGICS

- ❑ La següent taula mostra els operadors lògics:

Operador lògic	Operació
AND	Retorna el valor TRUE quan les dues condicions són veritables.
OR	Retorna el valor TRUE quan una de les dues condicions és veritable.
NOT	Retorna el valor TRUE si la condició és falsa.

- **Ex1** → Mostrar els empleats els quals el seu salari sigui igual o major que 13000 i que no pertany al departament 20.

```
SELECT e.employee_id, e.salary, e.department_id
FROM   employees e
WHERE  e.salary > 13000
      AND e.department_id !=20;
```

employee_id	salary	department_id
100	24000.00	90
101	17000.00	90
102	17000.00	90
145	14000.00	80
146	13500.00	80



## 4. OPERADORS DE COMPARACIÓ DE CADENES DE CARÀCTERS

- ❑ Per comparar cadenes de caràcters, fins ara hem fet servir l'operador de comparació igual (=).

➤ **Ex** → obtenir els cognoms dels programadors que l'id del seu cap és el 103.

```
SELECT e.employee_id, e.first_name, e.last_name, e.manager_id  
FROM employees e  
WHERE e.job_id > 'IT_PROG' AND e.manager_id = 114;
```

- ❑ Aquest operador no ens serveix per realitzar determinades consultes, com ara:
  - Obtenir les dades dels empleats que el seu cognom comenci per «P» o «obtenir les dades dels empleats que incloguin la paraula «Smith»»
  - Per poder realitzar aquest tipus de consultes en SQL hem de fer servir l'operador LIKE que permet utilitzar els següents caràcters especials dins les cadenes de comparació:
    - ✓ % Comodí → representa qualsevol cadena de 0 o més caràcters.
    - ✓ ' \_ ' Marcador de posició → representa un caràcter qualsevol.



## 4. OPERADORS DE COMPARACIÓ DE CADENES DE CARÀCTERS

- Dins la clàusula WHERE aquest operador s'utilitzarà de la següent manera:  
WHERE column LIKE 'caràcters especials'
- En una sentència WHERE es pot fer servir varies clàusules LIKE niades per operadors AND/OR.

- Ex1 → Mostrar tots els empleats que el seu cognom sigui «Smith».

```
SELECT e.employee_id, e.last_name  
FROM employees e  
WHERE e.last_name LIKE 'Smith'
```

employee_id	last_name
159	Smith
171	Smith

- Ex2 → Mostrar tots els empleats que el seu nom comenci per S.

```
SELECT e.employee_id, e.first_name  
FROM employees e  
WHERE e.first_name LIKE 'S%'
```

employee_id	first_name
100	Steven
116	Shelli
117	Sigal
123	Shanta
128	Steven
138	Stephen



## 4. OPERADORS DE COMPARACIÓ DE CADENES DE CARÀCTERS

- Ex3 → Mostrar tots els empleats que el seu nom contingui una «h» com a segon caràcter

```
SELECT e.employee_id, e.first_name  
FROM employees e  
WHERE e.first_name LIKE '_h%'
```

employee_id	first_name
116	Shelli
123	Shanta
153	Christopher
179	Charles
205	Shelley

- Ex4 → Mostrar tots els cognoms dels empleats que comenci per «S» i que tinguin una «a» en el seu interior.

```
SELECT e.employee_id, e.first_name  
FROM employees e  
WHERE e.first_name LIKE 'S%a%'
```

employee_id	first_name
117	Sigal
123	Shanta
161	Sarath
166	Sundar
173	Sundita
192	Sarah
194	Samuel
203	Susan



## 5. NULL i NOT NULL

- ❑ Es diu que una columna d'una fila és NULL si està completament buida.
- ❑ Per comprovar si el valor d'una columna és nul fem servir l'expressió: *columna IS NULL*.
- ❑ Si volem saber si el valor d'una columna no és nul fem servir l'expressió: *columna IS NOT NULL*.
- ❑ Quan comparem amb valors NULL i NOT NULL no podem fer servir operadors d'igualtat, major o menor.

➤ Ex1 → **Mostrar tots els empleats el quals no tenen comissió.**

```
SELECT e.employee_id, e.first_name, e.last_name, e.commission_pct  
FROM employees e  
WHERE e.commission_pct is null
```

employee_id	first_name	last_name	commission_pct
100	Steven	King	<i>null</i>
101	Neena	Kochhar	<i>null</i>
102	Lex	De Haan	<i>null</i>
103	Alexander	Hunold	<i>null</i>
104	Bruce	Ernst	<i>null</i>



## 6. COMPROVACIONS AMB CONJUNT DE VALORS

- ❑ Podem comparar una columna o una expressió amb una llista de valors fent servir els operadors IN i BETWEEN.

### Operador IN

- ❑ L'operador IN ens permet comprovar si una expressió pertany o no (NOT) a un conjunt de valors, fent possible la realització de comparacions múltiples.
- ❑ El seu format és: *<expressió> [NOT] IN (llista de valors separats per comes)*
  - Ex1 → Consulta els cognoms de la taula EMPLOYEES els quals el seu departament és el 10 o el 30.

```
SELECT e.employee_id, e.first_name, e.last_name, e.department_id  
FROM employees e  
WHERE e.department_id in (10,30)
```

employee_id	first_name	last_name	department_id
200	Jennifer	Whalen	10
114	Den	Raphaely	30
115	Alexander	Khoo	30
116	Shelli	Baida	30
117	Sigal	Tobias	30
118	Guy	Himuro	30
119	Karen	Colmenares	30





## 6. COMPROVACIONS AMB CONJUNT DE VALORS

### Operador IN

- Ex2 → Consulta els cognoms de la taula EMPLOYEES els quals el seu departament No són el 10 o el 30.

```
SELECT e.employee_id, e.first_name, e.last_name, e.department_id  
FROM employees e  
WHERE e.department_id not in (10,30)
```

employee_id	first_name	last_name	department_id
100	Steven	King	90
101	Neena	Kochhar	90
104	Bruce	Ernst	60

- Ex3 → Consulta els cognoms de la taula EMPLOYEES els quals el seus oficis no siguin IT\_PROG, FI\_ACCOUNT, FI\_MGR.

```
SELECT e.employee_id, e.first_name, e.last_name, e.job_id  
FROM employees e  
WHERE e.job_id in ('IT_PROG', 'FI_ACCOUNT', 'FI_MGR')
```

employee_id	first_name	last_name	job_id
109	Daniel	Faviet	FI_ACCOUNT
108	Nancy	Greenberg	FI_MGR
103	Alexander	Hunold	IT_PROG



## 6. COMPROVACIONS AMB CONJUNT DE VALORS

### Operador BETWEEN

- ❑ L'operador BETWEEN comprova si un valor està compromès o no (NOT) dins un rang de valors, des d'un valor final.
- ❑ El seu format és: *<expressió> [NOT] BETWEEN valor\_inicial AND valor\_final*
  - Ex1 → A partir de la taula EMPLOYEES, obté el cognom i el salari dels empleats que tenen un salari comprés entre 3500 i 5000.

```
SELECT e.employee_id, e.first_name, e.last_name, e.salary  
FROM employees e  
WHERE e.salary between 3500 and 5000
```

employee_id	first_name	last_name	salary
105	David	Austin	4800.00
106	Valli	Pataballa	4800.00
107	Diana	Lorentz	4200.00
137	Renske	Ladwig	3600.00
141	Trenna	Rajs	3500.00



## 6. COMPROVACIONS AMB CONJUNT DE VALORS

### Operador BETWEEN

- Ex2 → A partir de la taula EMPLOYEES, obté el cognom i el salari dels empleats que tenen un salari que no estigui comprés entre 3500 i 5000.

```
SELECT e.employee_id, e.first_name,  
       e.last_name, e.salary  
FROM   employees e  
WHERE  e.salary NOT between 3500 and 5000
```

employee_id	first_name	last_name	salary
100	Steven	King	24000.00
101	Neena	Kochhar	17000.00
102	Lex	De Haan	17000.00
103	Alexander	Hunold	9000.00
104	Bruce	Ernst	6000.00
115	Alexander	Khoo	3100.00
116	Shelli	Baida	2900.00
117	Sigal	Tobias	2800.00
118	Guy	Himuro	2600.00
119	Karen	Colmenares	2500.00

Aquesta consulta retorna els empleats que tenen un salari que es menor que 3500 i major de 5000.



## 7. COMBINACIÓ D'OPERADORS AND i OR

- ❑ Els operadors AND i OR es poden combinar de forma il·limitada, però hi ha que tenir cura al fer-los servir i utilitzar parèntesis per agrupar aquelles expressions que es destinen avaluar de manera conjunta.
  - ❑ Si no fem servir els parèntesis, es possible que els resultats no siguin els desitjats.
  - ❑ L'ordre de prioritats dels operadors lògics és el següent: primer NOT, després AND i, per últim, OR. Si volem alterar l'ordre farem servir parèntesi.
- Ex1 → A partir de la taula EMPLOYEES, obté el cognom, salari i dept\_nº dels empleats sigui major de 2000 en els departaments 10 i 20.

```
SELECT e.employee_id, e.first_name, e.last_name, e.salary, e.department_id  
FROM employees e  
WHERE e.salary > 2000 AND (e.department_id=10 OR e.department_id=20)
```

employee_id	first_name	last_name	salary	department_id
200	Jennifer	Whalen	4400.00	10
201	Michael	Hartstein	13000.00	20
202	Pat	Fay	6000.00	20



# Preguntes!!!!