



CFGS ASIX/DAW
Mòdul 2: Bases de dades
Prof: Jordi Quesada

INS Joan d'Àustria



4. Consultes

4.7. Subconsultes

4.7.1 Sintaxis

Subconsultes



De vegades ens trobem que per obtenir certa informació ens cal tenir un càlcul realitzat prèviament.

Per exemple:

- Mostra els empleats que cobren més que Clara.



Caldría saber quin és el sou de Clara per poder respondre

Subconsultes



Però: com que puc mirar quin és el sou de Clara, jo no tinc cap problema....



```
select nom_empl from Empleats where sou > 1500
```

No obstant:

El sou de Clara pot canviar en qualsevol moment, per tant, aquesta consulta no és fiable



Subconsultes



Una altra opció que podríem pensar consistiria en dividir el problema en 2 consultes separades:

```
select sou from Empleats where nom_Empl = 'Clara';
```

SOU
1500

```
select nom_empl from Empleats where sou > 1500
```

NOM_EMPL
Maria
Pere
Jordi

Subconsultes



No obstant:

Fer moltes consultes a una base de dades és menys eficient que fer una sola, especialment si es refereixen a una taula



Subconsultes



SOLUCIÓ:



- ✓ Fer servir subconsultes
- Una subconsulta no és més que una consulta anidada, és a dir, una consulta dins d'una altra
- Hi ha dos tipus:
 - Subconsultes simples
 - Subconsultes de múltiples files



4. Consultes

4.7. Subconsultes

4.7.2. Subconsultes simples

Subconsultes simples



Una subconsulta simple és la que ens retorna
un únic resultat

Repetim: un únic resultat



Per tant: Ens por tornar 2 resultats? NO!!!

Per tant: Ens por tornar 100 resultats? NO!!!

Per tant: Ens por tornar 0 resultats? NO!!!

Subconsultes simples



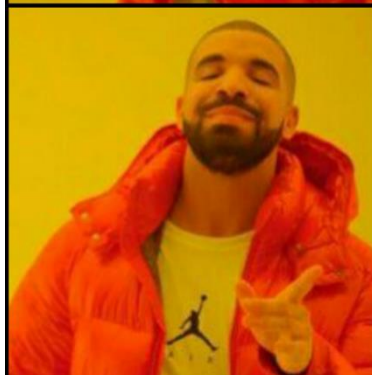
I què és un resultat? Un resultat vol dir un valor.



```
NOM_EMPL
-----
Maria
Pere
Jordi
Clara
SQL>
```

```
SQL> select nom_emp
no rows selected
```

CODI_EMPL	NOM_EMPL	COGNOM_EMPL
5	Clara	Blanca



```
SOU
-----
1500
```

```
NOM_EMPL
-----
Maria
```

Subconsultes simples



Si la subconsulta NO ens retorna 1 únic resultat tindrem errors:

- Si retornem més d'una columna:

```
mysql> select *  
-> from estudiant  
-> where casa_id =  
-> ( select nom, casa_id from estudiant where id=5);  
ERROR 1241 (21000): Operand should contain 1 column(s)  
mysql>
```



- Si retornem més d'una fila:

```
mysql>  
mysql> select *  
-> from estudiant  
-> where casa_id =  
-> ( select casa_id from estudiant where nom like '%r%');  
ERROR 1242 (21000): Subquery returns more than 1 row  
mysql>
```

Subconsultes simples



Si una subconsulta és la que ens retorna un valor, el funcionament serà utilitzar-la amb parèntesis per tal que s'executi abans

```
select nom_empl  
from Empleats  
where sou > (  
  
);
```

```
select sou from Empleats where nom_Empl = 'Clara'
```

Observa els paréntesis i
que la subconsulta no acaba en ;

Subconsultes simples



Ja ho podem provar:

```
select nom_empl
from Empleats
where sou > (
select sou from Empleats where nom_Empl='Clara'
);
```

```
mysql> select nom_empl
-> from Empleats
-> where sou > (
-> select sou from Empleats where nom_Empl='Clara'
-> );
+-----+
| nom_empl |
+-----+
| Maria    |
| Pere     |
| Jordi    |
+-----+
3 rows in set (0.00 sec)
```

Subconsultes simples



El que fem es simplement obtenir el valor de la subconsulta i utilitzar-lo a la consulta que està més externa.

Podem fer servir qualsevol dels operadors que ja sabem: $>$, \geq , $<$, \leq , $=$, \neq

Subconsultes simples



4.8.1.- Subconsultes simples



Mostra el títol de les pel·lícules que es van fer el mateix any que la pel·lícula Alien

Mostra el títol i les votacions de les pel·lícules amb més vots que Titanic

Mostra el títol i les votacions de les pel·lícules amb més vots que Titanic però menys que Star Wars de dos maneres diferents

Subconsultes simples



**ROUND 2
FIGHT!**

Les subconsultes simples les podem utilitzar de moltes maneres ja que cal recordar que només són un valor.

Per exemple, podem utilitzar-les per modificar dades

```
update empleats  
set sou=(select sou from empleats where nom_Empl='Pere')  
where ciutat_dpt='Barcelona';
```


Subconsultes simples



També podem utilitzar les funcions de grup

```
select titulo from peliculas  
where votos = (select max(votos) from peliculas);
```

```
TITULO  
-----  
Star Wars
```

```
select titulo from peliculas  
where votos = (select min(votos) from peliculas);
```

```
TITULO  
-----  
What's Up, Tiger Lily?  
Class  
Soldier's Story, A  
Henry: Portrait of a Serial Killer  
King Ralph  
Kicking and Screaming  
  
6 rows selected.
```



4. Consultes

4.7. Subconsultes

4.7.3. Subconsultes múltiples

Subconsultes múltiples



També podem utilitzar subconsultes que retornin varies files PERÒ NOMÉS UNA COLUMNA.

Aquestes subconsultes s'anomenen de múltiples files i les tractarem amb uns operadors nous

Subconsultes múltiples



- IN: Aquest operador ja el coneixíem. L'utilitzàvem per buscar dins d'un conjunt de valors. El funcionament serà:

```
SELECT ...  
FROM TAULA  
WHERE Atribut IN (subconsulta)
```

Ex: Nom dels empleats assignats en algun projecte que el preu sigui superior a 5000



```
select nom_Empl  
from empleats  
where codi_proj IN (  
    select codi_proj  
    from projectes  
    where preu>5000  
);
```

Subconsultes múltiples



- NOT IN: El funcionament és el contrari:

```
SELECT ...  
FROM TAULA  
WHERE Atribut NOT IN (subconsulta)
```

Ex: Nom dels empleats **NO** assignats
en algún projecte que el preu
sigui superior a 5000



```
select nom_Empl  
from empleats  
where codi_proj NOT IN (  
    select codi_proj  
    from projectes  
    where preu>5000  
);
```

Subconsultes múltiples



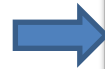
- ANY: La traducció seria algun. Aquest operador serveix per validar si hi ha algun registre que compleixi una condició. Per tant, és similar a l'operador IN, però té alguna diferència:
 - L'operador ANY va acompanyat d'un operador de comparació: =, >, <, >=....
 - Per tant, podem utilitzar coses com > ANY que voldrà dir que sigui més gran que algun
 - Així, podríem establir que més o menys, l'operador IN seria equivalent a = ANY

Subconsultes múltiples



- ANY: Exemple

Ex: Dades dels empleats que cobrin més que algun dels empleats que siguin dissenyadors



```
select nom_Empl, nom_Dpt, sou
from empleats
where sou > ANY (
    select sou
    from empleats
    where nom_dpt='DISS'
);
```

Subconsultes múltiples



- ALL: La traducció seria tots. Aquest operador és similar a ANY però ara la comparació ha de ser certa per tots els elements de la subconsulta:
 - Amb ANY era cert si es complia per un sol element. Amb ALL s'ha de complir per tots

Subconsultes múltiples



- ALL: Exemple

Ex: Dades dels empleats que cobrin més que **TOTS** els empleats que siguin dissenyadors



```
select nom_Empl, nom_Dpt, sou
from empleats
where sou > ALL (
    select sou
    from empleats
    where nom_dpt='DISS'
);
```

Subconsultes múltiples



Mostra el títol de les pel·lícules que tinguin més vots que qualsevol de les pel·lícules fetes a l'any 1995 de dos maneres diferents

```
+-----+  
| titulo |  
+-----+  
| Star Wars |  
| Pulp Fiction |  
| Blade Runner |  
| Titanic |  
+-----+
```



4. Consultes

4.7. Subconsultes

4.7.4. Aspectes avançats

Aspectes avançats



- El resultat d'una consulta, en realitat, es pot utilitzar de moltes maneres:
 - Select
 - From
 - where
 - having
- Cal tenir present que aquestes formes de treball són avançades.

Aspectes avançats



- SELECT

- Imagina que volem portar un control de quants productes de cada categoria tenim en diferents moments en una botiga, o en el nostre cas:

```
SELECT SYSDATE() HOY,  
(SELECT COUNT(*) FROM profe) as profes,  
(SELECT COUNT(*) FROM estudiant) as estudiants;
```

```
mysql> SELECT SYSDATE() HOY,  
-> (SELECT COUNT(*) FROM profe) as profes,  
-> (SELECT COUNT(*) FROM estudiant) as estudiants;  
+-----+-----+-----+  
| HOY          | profes | estudiants |  
+-----+-----+-----+  
| 2021-03-07 17:00:50 |      10 |         280 |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

Aspectes avançats



- SELECT

- Existeix una forma de treball consistent en fer consultes anidades que poden passar paràmetres a la subconsulta:

```
select *,  
       (select count(*) from matricula where estudiant_id=id)  
from estudiant;
```

id	nom	anyo	casa_id	(select count(*) from matricula where estudiant_id=id)
1	Waino Poulos	7	2	8
2	Daren Ortiz	3	4	8
3	Elbert Moore	6	2	8
4	Kaylee Grady	4	1	8
5	Elna Jacobs	1	2	9
6	Alisha Hartmann	4	2	8
7	Mya Dooley	1	4	9
8	Luigi Mante	2	4	8
9	Ella Stroman	4	2	8
10	Moses Prohaska	5	3	8
11	Ray Hermann	7	4	8
12	Charmen DuBois	2	2	9

Aspectes avançats



- FROM
 - El resultat d'una consulta, es pot tractar com origen d'una altra consulta
 - La condició és que a aquest origen de dades li posem un àlies

```
Select nom FROM  
(select * FROM profe) b;
```

```
mysql> Select nom FROM  
-> (select * FROM profe) b;  
+-----+  
| nom |  
+-----+  
| Severus Snape |  
| Minerva McGonagall |  
| Filius Flitwick |  
| Pomona Sprout |  
| Remus Lupin |  
| Aurora Sinistra |  
| Rolanda Hooch |  
| Charity Burbage |  
| Cuthbert Binns |  
| Albus Dumbledore |  
+-----+  
10 rows in set (0.00 sec)
```

Aspectes avançats



- FROM

- Per exemple, imagina que volem saber per a cada empleat, quants companys de projecte treballen amb ell

```
select *  
from empleats e,  
     (select codi_proj, count(*) as total  
      from empleats  
      group by codi_proj) b  
where e.codi_proj=b.codi_proj;
```

Codi_empl	Nom_empl	Cognom_empl	sou	Nom_dpt	Ciutat_dpt	Codi_proj	codi_proj	total
1	Maria	Puig	2000	DIR	Girona	1	1	2
2	Pere	Mas	2500	DIR	Barcelona	4	4	3
3	Anna	Ros	1850	DISS	Lleida	3	3	2
4	Jordi	Roca	1800	DISS	Barcelona	4	4	3
5	Clara	Blanca	1500	PROG	Tarragona	1	1	2
6	Laura	Tort	1420	PROG	Tarragona	3	3	2
7	Roger	Salt	1200	NULL	NULL	4	4	3

7 rows in set (0.01 sec)

Aspectes avançats



- WHERE/HAVING
 - L'ús de subconsultes en el where i en el having és molt freqüent.

```
Select materia
FROM classes
where profe_id= (
    Select id from profe
    where casa_id = (
        Select id FROM casa
        where nom='Gryffindor'
    )
);
```

```
mysql> Select materia
-> FROM classes
-> where profe_id= (
->     Select id from profe
->     where casa_id = (
->         Select id FROM casa
->         where nom='Gryffindor'
->     )
-> );
+-----+
| materia |
+-----+
| Transfiguracio |
+-----+
1 row in set (0.00 sec)
```

Aspectes avançats



- Les subconsultes també s'utilitzen en operacions com
 - Insert
 - Delete
 - Update
 - Create
 - ...

Aspectes avançats



- INSERT

- A banda de l'ús que podem pensar, una altra utilitat consisteix en recuperar l'ID autoincrement d'una taula relacionada
- Imagina que volem donar d'alta al profe “Jordi” i també indicar que dona la matèria BBDD

```
INSERT INTO profe (nom, casa_id)
VALUES ('Jordi',NULL);
```

```
Insert into classes (materia, profe_id)
values ('BBDD', (Select id from profe where
nom='Jordi'));
```

Aspectes avançats



- CREATE
 - Es poden crear taules realitzant consultes.
 - Per exemple, per fer una còpia de certes dades o per exportar-les
 - També per tenir resultats calculats

```
CREATE TABLE NousAlumnes as (  
  select materia, nom  
  from estudiant join matricula on (matricula.estudiant_id=estudiant.id)  
    join classes on (matricula.classe_id=classes.id)  
  where estudiant.anyo=1  
  order by materia;  
);
```