

SUBCONSULTES



CFGs DAM



1. QUÈ ÉS UNA SUBCONSULTA?

- Una subconsulta és un tipus de consulta que ens permet construir consultes més complexes.
- Una subconsulta (o subquery) es defineix com una **consulta (query) dins d'una altra consulta** i obté unes dades intermèdies (una o més dades) que s'utilitzen en la consulta principal.
- De forma simplificada, és una sentència SELECT que s'utilitza dins d'altra sentència SELECT.

1. QUÈ ÉS UNA SUBCONSULTA?

- Per entendre millor el funcionament, comencem amb un exemple senzill:
 - Tenim una base de dades d'un videoclub i suposem que volem trobar les pel·lícules que tinguen una taxa de lloguer superior a la mitjana del lloguer. Ho podem fer en dos passos:
 - 1) Busquem la tarifa mitjana de lloguer mitjançant la sentència SELECT i la funció mitjana (AVG).
 - 2) Utilitzem el resultat de la primera consulta en la segona sentència SELECT per trobar les pel·lícules que volem.
- Amb aquesta consulta obtindrem el preu mitjà del lloguer (PAS 1):

```
SELECT AVG (taxa_lloguer) FROM pelicules;
```

- I a l'executar-la, ens retornarà un valor, per exemple:

avg
2.98

1. QUÈ ÉS UNA SUBCONSULTA?

- Ara ja podem obtenir les pel·lícules amb una taxa de lloguer superior a la mitjana:

```
SELECT film_id, titol, taxa_lloguer  
FROM pelicules  
WHERE taxa_lloguer > 2.98;
```

- Però aquest codi que requereix dos passos, no és massa elegant... Volem una manera de passar el resultat de la primera consulta a la segona consulta en una **única consulta**. La solució és utilitzar una subconsulta.

1. QUÈ ÉS UNA SUBCONSULTA?

- En resum:
- Una subconsulta és una consulta “anidada” dins d’una altra consulta (com ara **SELECT**, **INSERT**, **DELETE** o **UPDATE**).
 - És a dir, que és possible col·locar una **subquery** dins de les consultes de tipus **SELECT**, **INSERT**, **UPDATE** i **DELETE**.
 - La **subquery** és sempre una sentència **SELECT** i ha d’anar **tancada entre parèntesis**.
- Ens centrarem només en les subconsultes dins de la sentència **SELECT**.

1. QUÈ ÉS UNA SUBCONSULTA?

- Així, per construir una subconsulta, posem la segona consulta entre claudàtors i la fem servir a la clàusula WHERE de la consulta principal com a expressió:

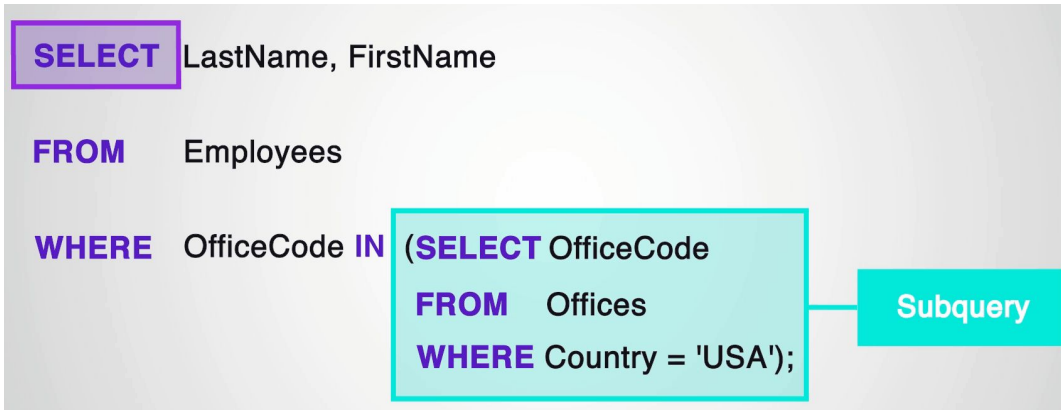
SELECT llista_columnes

FROM nom_taula

WHERE columna operador (SELECT columna
FROM nom_taula
WHERE condicio);

1. QUÈ ÉS UNA SUBCONSULTA?

- La **consulta dins dels parentesi** s'anomena **subconsulta o consulta interna** i s'executa abans que s'execute la consulta principal.
- La consulta que conté la subconsulta es coneix com a **consulta principal o consulta externa** i **utilitza el resultat de la subconsulta**.



1. QUÈ ÉS UNA SUBCONSULTA?

- Seguint amb l'exemple del videoclub, ho podríem obtindre amb la següent consulta:

```
SELECT film_id, titol, taxa_lloguer
```

```
FROM pelicules
```

```
WHERE taxa_lloguer > (SELECT AVG (taxa_lloguer)  
                      FROM pelicules);
```


2. TIPUS DE SUBCONSULTES

- Hi ha diversos tipus de subconsultes: d'una sola fila, de diverses files o correlacionades.
- Atenent al nombre de files i columnes que retornen:
 - Subconsultes que retornen un valor únic (una fila amb una columna).
 - Subconsultes que retornen una única tupla o fila amb més d'una columna.
 - Subconsultes que retornen un conjunt de files (0, 1 o diverses files).

2.1. SUBCONSULTES QUE RETORNEN UN VALOR ÚNIC

- No es consideren les que retornen diverses columnes d'una sola fila, ja que es produirà un error d'execució!
- Una **subquery que retorna un valor simple** pot ser col·locada en qualsevol lloc en què puga haver-hi una expressió.
 - Es poden situar en la clàusula FROM, WHERE o HAVING de la consulta principal però és **molt habitual trobar-les en la clàusula WHERE**.
 - El valor que retorna la subconsulta l'utilitzarem com un valor normal en una expressió, una comparació (=, <, >, >=, <=, != o <>).

2.1. SUBCONSULTES QUE RETORNEN UN VALOR ÚNIC

- Exemple: Imaginem que volem seleccionar el departament, salari i nom de l'empleat amb salari mínim. Per a això farem servir la següent consulta:

```
SELECT empname AS nom, deptno AS departament, sal AS salari
FROM emp
WHERE sal = (SELECT MIN(sal) FROM emp);
```

- que retornarà:

	nom character varying (10)	departament smallint	salari numeric (9,2)
1	SMITH	20	800.00

- De la mateixa manera podríem obtindre l'empleat amb salari màxim (canviant la funció MIN per MAX), o els empleats que tenen un salari superior al salari mitjà de tota l'empresa (canviant = per > i utilitzant la funció AVG), o altres.

2.2. SUBCONSULTES QUE RETORNEN 1 FILA AMB DIVERSOS VALORS

- Seria el cas de les subconsultes que retornen una única fila amb més d'una columna.
 - El número de columnes que utilitzem en l'expressió de l'esquerra ha de coincidir en el número de columnes que retorna la subconsulta.
 - Si la subconsulta retorna més d'una fila, es produirà un error d'execució.
“ERROR: more than one row returned by a subquery used as an expression”.

2.2. SUBCONSULTES QUE RETORNEN 1 FILA AMB DIVERSOS VALORS

- En aquest tipus de consultes normalment utilitzem els operadors = i != (o <>), però podeu utilitzar tots els **operadors de comparació** (=, >, <, > =, <=, <>, !=).
 - Cal recordar que al comparar el valor de dues columnes, **només es pot comparar si no hi ha cap valor nul en cap d'eixos atributs**.
 - Per exemple l'expressió (4) = (null) retorna com a resultat el valor desconegut, ja que no es pot dir que l'expressió siga certa ni falsa (null representa valor desconegut).

2.2. SUBCONSULTES QUE RETORNEN 1 FILA AMB DIVERSOS VALORS

- Exemple: Imaginem que volem obtenir el salari i la comissió que guanya un empleat que guanye una 4^a part del sou màxim i la comissió més alta

```
SELECT empname, job
FROM emp
WHERE (sal, comm) = (SELECT MAX(sal)/4, MAX(comm) FROM emp);
```

- En aquest cas si que hi ha un empleat que compleix la condició, per tant, ens mostra una fila i les columnes que hem demanat:

	empname character varying (10)	job character varying (10)
1	MARTIN	SALESMAN

2.2. SUBCONSULTES QUE RETORNEN 1 FILA AMB DIVERSOS VALORS

- En PostgreSQL aquestes subconsultes també es coneixen com **row subqueries**, i com hem dit, es tracta d'una subconsulta que retorna una sola fila i més d'un valor de columna.
- Podem utilitzar aquesta altra sintaxi:

SELECT llista_columnes

FROM nom_taula

WHERE ROW (llista_cols) operador (subconsulta);

2.2. SUBCONSULTES QUE RETORNEN 1 FILA AMB DIVERSOS VALORS

- Veiem un exemple un poc simple, però que serveix per entendre el funcionament:

```
SELECT empname, sal
FROM emp
WHERE ROW(sal, deptno) IN (SELECT sal, deptno
                           FROM emp
                           WHERE sal BETWEEN 1000 and 1500)
ORDER BY empname;
```

- Ens retorna:

	empname character varying (10)	sal numeric (9,2)
1	ADAMS	1100.00
2	MARTIN	1250.00
3	MILLER	1300.00
4	TURNER	1500.00
5	WARD	1250.00

Com la subcosulta retorna més d'una fila, utilitzem un **operador de conjunts (IN)**, que veurem en més profunditat en el següent apartat

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES

- Aquest tipus retornen més d'una fila com un conjunt de resultats. Se situa, generalment en les clàusules **WHERE** o **HAVING**.
- Atès que retornen múltiples files, hem d'utilitzar algun **operador de comparació de conjunts** (**IN**, **EXISTS**, **ALL**, **ANY**).
- **IN**: comprova si el valor pertany a un conjunt; retorna true si el valor és igual que un element de la llista.
- **EXISTS**: comprova l'existència de files retornades per una subconsulta; retorna true si la consulta retorna al menys una fila (existeix un element)
- **ANY**: recupera dades comparant un valor amb un conjunt de valors retornats per una subconsulta; compara el valor amb cada element de la llista.
- **ALL**: consulta les dades comparant un valor amb una llista de valors retornats per una subconsulta; retorna true si el valor és igual a tots els valors de la llista.

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES

- Els operadors **ANY** i **ALL** s'utilitzen combinats amb els operadors de comparació habituals:
 - = ANY: Igual a algun dels valors de la llista (equivalent a IN).
 - != ANY: Diferent d'algun dels valors de la llista.
 - < ANY: Més xicotet que el més gran dels valors de la llista.
 - > ANY: Major que el menor dels valors de la llista.
- = ALL: Igual a amb tots els valors de la llista.
- != ALL: No igual a cap valor de la llista (equivalent als NOT IN);
- > ALL: Major que el major de tots els valors de la llista.
- < ALL: Més xicotet que el menor dels valors de la llista.

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES



- **NOTA:** Cal dir que aquests operadors tenen dues formes de funcionar: comparació de valors i comparació de files.
 - Si comparem **per valors** es poden **utilitzar tots els operadors de comparació** (>, >=, ...),
 - mentre que si comparem **per files** només podem **utilitzar els comparadors = i !=** (<>).

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador ALL**
- L'operador **ALL** compara el valor amb **TOTS els valors retornats** per la subconsulta. Sintaxi:

SELECT llista_columnnes

FROM nom_taula

WHERE expressio operador **ALL** (subquery)

la subquery ha de retornar exactament 1 columna; l'expressió de l'esquerra s'avalua i es compara amb cada fila del resultat de la subconsulta mitjançant l'operador, que ha de donar un resultat booleà.

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador ALL**

Exemple: La següent consulta selecciona el departament (el codi) amb el salari mitjà més alt.

```
SELECT deptno, ROUND(AVG(sal),2) AS "salari mitjà"  
FROM emp  
GROUP BY deptno  
HAVING AVG(sal)>=ALL (SELECT AVG(sal) FROM emp  
                      GROUP BY deptno);
```

- Retorna:

	deptno smallint	salari mitjà numeric
1	10	2916.67

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador ALL**
- **Exemple:** La subconsulta troba el salari mitjà de cada departament i, a continuació, la consulta principal selecciona el departament amb el salari mitjà més alt.
- Utilitzem **ALL** en la consulta ja que el departament seleccionat ha de tenir un salari mitjà superior o igual a tots els salaris mitjans dels altres departaments.
- Utilitzem **HAVING** i **no WHERE** perquè en la comparació necessitem utilitzar una funció agregada (**AVG** en aquest cas) i **WHERE** no ho admet.

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador ALL**
- **Exemple:** Seleccionar els noms dels empleats i el seu salari sempre que tinguen menor salari que els empleats del departament 10:

```
SELECT empname, sal
FROM emp
WHERE sal < ALL (SELECT sal
                  FROM emp
                  WHERE deptno = 10);
```

- Retorna:

	empname character varying (10)	sal numeric (9,2)
1	SMITH	800.00
2	WARD	1250.00
3	MARTIN	1250.00
4	ADAMS	1100.00
5	JAMES	950.00

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador ANY/SOME**
- L'operador **ANY** compara el valor amb cada valor retornat per la subconsulta. Per tant, ANY (que ha de seguir un operador de comparació) retorna TRUE si la comparació és TRUE per a QUALESEVOL dels valors de la columna que retorna la subconsulta. Sintaxi:

SELECT llista_columnnes

FROM nom_taula

WHERE expressio operador **ANY / SOME** (subquery)

SOME és un sinònim de **ANY**. **IN** equival a **= ANY**

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador ANY**

Exemple: La consulta següent selecciona qualsevol empleat que treballa a la ubicació de NEW YORK. La subconsulta troba l'identificador del departament a la ubicació de NEW YORK i, a continuació, la consulta principal selecciona els empleats que treballen en qualsevol d'aquests departaments.

NOTA: Hem utilitzat **ANY** en aquesta consulta, perquè és probable que la subconsulta trobe més d'un departament a la ubicació de NEW YORK. També podríem haver utilitzat **SOME** i el resultat seria el mateix.

```
SELECT empname, deptno CodiDepartament
FROM emp
WHERE deptno = ANY (SELECT deptno
                    FROM dept
                    WHERE loc='NEW YORK');
```

NOTA 2: Si utilitzem la paraula clau **ALL** en lloc de **ANY**, obtindríem dades només si algun empleat treballa a tots els departaments de la ubicació de NEW YORK (justament en aquesta base de dades cada departament només està ubicant a un lloc, però el més habitual és que un departament tinguera més d'una ubicació i en eixe cas, **ALL** no ens serviria per al que volem).

- **Retorna:**

	empname character varying (10)	codidepartament... smallint
1	CLARK	10
2	KING	10
3	MILLER	10

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador ANY**

Exemple: Conèixer el nom i salari de tots els empleats que cobren el mateix o més que qualsevol dels empleats del departament 10:

```
SELECT empname, sal
FROM emp
WHERE sal >= ANY (SELECT sal
                  FROM emp
                  WHERE deptno = 10);
```

- Retorna:

	empname character varying (10)	sal numeric (9,2)
1	ALLEN	1600.00
2	JONES	2975.00
3	BLAKE	2850.00
4	CLARK	2450.00
5	KING	5000.00
6	TURNER	1500.00
7	FORD	3000.00
8	MILLER	1300.00
9	ESCOTT	3000.00

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador IN**
- Una subconsulta pot retornar zero o més files. Per utilitzar una subconsulta, podem utilitzar l'operador **IN** a la clàusula WHERE. Sintaxi:

SELECT llista_columnnes

FROM nom_taula

WHERE expressio **IN** (subquery)

la subquery ha de retornar exactament 1 columna; l'expressió de l'esquerra s'avalua i es compara amb cada fila del resultat de la subconsulta, que ha de donar un resultat booleà.

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador IN**

Exemple: La consulta següent selecciona els empleats que treballen a la ubicació CHICAGO. La subconsulta troba l'identificador del departament amb ubicació en CHICAGO (pot retornar diverses files) i, a continuació, la consulta principal selecciona els empleats que treballen a qualsevol d'aquests departaments.

```
SELECT empname, deptno Codi_Departament
FROM emp
WHERE deptno IN (SELECT deptno FROM dept
                 WHERE loc='CHICAGO');
```

- Retorna:

	empname character varying (10)	codi_departament smallint
1	ALLEN	30
2	WARD	30
3	MARTIN	30
4	BLAKE	30
5	TURNER	30
6	JAMES	30

Anàlogament podríem utilitzar l'operador **NOT IN** si el que volem és avaluar si l'expressió és distinta a tots els valors de la columna que retorna la subconsulta

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador IN**

Exemple: La següent consulta selecciona aquells empleats que no treballen en aquell departament on treballen els caps de departament d'ID entre 7780 i 7790. La subconsulta troba l'identificador del departament que dirigeix l'empleat l'identificador del qual està entre 7780 i 7790 i, a continuació, la consulta principal selecciona els empleats que no treballen en cap d'aquests departaments.

```
SELECT empname,deptno
FROM emp
WHERE deptno NOT IN (SELECT deptno FROM emp
                     WHERE mgr
                     BETWEEN 7780 AND 7790);
```

- Retorna:

	empname character varying (10)	deptno smallint
1	ALLEN	30
2	WARD	30
3	MARTIN	30
4	BLAKE	30
5	TURNER	30
6	JAMES	30

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- Podríem dir que les instruccions que inclouen una subconsulta normalment tenen un d'aquests formats:

```
WHERE expresión [NOT] IN (subconsulta)
WHERE expresión operador_comparación [ANY | ALL] (subconsulta)
WHERE [NOT] EXISTS (subconsulta)
```

- Ara veurem l'operador EXISTS

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador EXISTS**
- L'operador **EXISTS** és un operador booleà que prova l'existència de files en una subconsulta, és a dir, que accepta un argument que siga una subconsulta. Sintaxi:

SELECT llista_columnes

FROM nom_taula

WHERE EXISTS (subquery)

La subconsulta s'avalua per determinar si retorna alguna fila. Si retorna com a mínim una fila, el resultat de **EXISTS** és cert; si la subconsulta no retorna cap fila, el resultat de **EXISTS** és fals.

L'operador **EXISTS** només es "preocupa" pel nombre de files retornades de la subconsulta, no pel contingut de les files. Per tant, les columnes que apareixen a la clàusula **SELECT** de la subconsulta no són importants.

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES. EXEMPLES.

- **Subconsultes amb l'operador EXISTS**

Exemple: A la següent consulta es troben empleats de la taula d'empleats que tenen almenys una persona al seu càrrec:

```
SELECT empno, empname, job, deptno
FROM emp E
WHERE EXISTS (SELECT * FROM emp
              WHERE mgr = E.empno);
```

- Retorna:

	empno [PK] smallint	empname character varying (10)	job character varying (10)	deptno smallint
1	7566	JONES	MANAGER	20
2	7698	BLAKE	MANAGER	30
3	7782	CLARK	MANAGER	10
4	7839	KING	PRESIDENT	10
5	7902	FORD	ANALYST	20
6	7788	ESCOTT	ANALYST	20

2.3. SUBCONSULTES QUE RETORNEN MÚLTIPLES FILES.

- **Subconsultes amb l'operador EXISTS**
- L'operador **EXISTS** s'utilitza sovint amb una **subconsulta correlacionada**. Les consultes vistes fins ara fan que el resultat de la columna general depenga dels valors de la subquery. Però també és possible fer que el resultat de la consulta interna depenga de valors proporcionats per la consulta externa.
- Això vol dir que la consulta interna, s'executa una vegada per cada fila seleccionada per la consulta principal.

2.4. SUBCONSULTES CORRELACIONADES

- En altres paraules, una **subconsulta correlacionada** és una subconsulta que conté una referència a una taula que també apareix a la consulta externa. PostgreSQL avalua de dins cap a fora.
- **Exemple de consulta correlacionada:** Volem seleccionar els empleats d'un departament que tinguin un salari superior a la mitjana de l'esmentat departament.
 - En primer lloc hem de fer una selecció de tots els empleats en la consulta externa.
 - Després s'executa la subquery (en cada fila, i es calcula la mitjana dels empleats del departament en què es troba l'empleat que s'està seleccionant en l'anterior (E.empno)).

2.4. SUBCONSULTES CORRELACIONADES

- Exemple de consulta correlacionada:

```
SELECT empno, deptno, sal
FROM emp E
WHERE sal > (SELECT AVG(sal)
             FROM emp T
             WHERE E.deptno = T.deptno);
```

- Retorna:

	empno [PK] smallint	deptno smallint	sal numeric (9,2)
1	7499	30	1600.00
2	7566	20	2975.00
3	7698	30	2850.00
4	7839	10	5000.00
5	7902	20	3000.00
6	7788	20	3000.00

2.4. SUBCONSULTES CORRELACIONADES

- Ja que les columnes seleccionades en la subconsulta són secundàries i el que importa és el nombre de files que retorna, és molt comú utilitzar l'operador **EXISTS** de la següent forma:
- ... **EXISTS** (**SELECT 1 FROM** taula **WHERE** condicio);
 - On 1 indica que la condició és certa o true.
 - També es habitual fer **SELECT ***.

2.4. SUBCONSULTES CORRELACIONADES

- S'entén millor a la següent consulta:

```
SELECT nom, cognom  
FROM clients  
WHERE EXISTS ( SELECT 1  
                FROM pagaments  
                WHERE pagaments.client_id = clients.customer_id );
```

- Resultat:

Jared	Ely
Mary	Smith
Patricia	Johnson
Linda	Williams
Barbara	Jones
Elizabeth	Brown
Jennifer	Davis
Maria	Miller

La consulta funciona com un INNER JOIN amb la columna client_id. Tanmateix, retorna com a màxim una fila per a cada fila de la taula de clients, tot i que hi ha algunes files corresponents a la taula de pagaments.