

CONNEXIÓ DE TAULES



CFGs DAM



1. INTRODUCCIÓ.

- En una base de dades relacional, les dades normalment es distribueixen en més d'una taula.
- Per seleccionar dades completes, sovint cal consultar dades de diverses taules.



- Fins ara hem vist consultes sobre una única taula de la base de dades, però, **com ho fem si volem combinar files de dos o més taules?**

2. UNIÓ DE TAULES

- La resposta és amb la sentència **JOIN** (unir, combinar).
- Les sentències **JOIN** combinen dades guardades en diferents taules i els presenten de forma filtrada en una taula de resultats.
 - Més concretament, es combinen columnes d'una o més taules en funció dels valors de les **columnes comunes entre taules relacionades**.
 - Les columnes comunes solen ser les columnes de claus primàries de la primera taula i les columnes de claus externes de la segona taula.

2. UNIÓ DE TAULES

- PostgreSQL admet unió interna, unió externa (esquerra, dreta, i unió exterior completa), unió creuada, unió natural i un tipus especial d'unió anomenat "autounió".
- La sintaxi general és:

SELECT columnnes

FROM nom_taula1 [**tipus_JOIN**] **JOIN** nom_taula2

ON columna_taula1 = columna_taula2

WHERE condicio;

2. UNIÓ DE TAULES. TAULES I DADES D'EXEMPLE.

- Per entendre millor el funcionament, treballarem amb les següents taules:

```
CREATE TABLE cistella_a (  
    a INT PRIMARY KEY,  
    fruita_a VARCHAR (100) NOT NULL  
);
```

```
INSERT INTO cistella_a (a, fruita_a)  
VALUES  
    (1, 'Poma'),  
    (2, 'Taronja'),  
    (3, 'Plàtan'),  
    (4, 'Cireres');
```

```
CREATE TABLE cistella_b (  
    b INT PRIMARY KEY,  
    fruita_b VARCHAR (100) NOT NULL  
);
```

```
INSERT INTO cistella_b (b, fruita_b)  
VALUES  
    (1, 'Taronja'),  
    (2, 'Poma'),  
    (3, 'Síndria'),  
    (4, 'Pera');
```

2.1. COMBINACIÓ INTERNA: INNER JOIN

- **(INNER) JOIN** és l'operació JOIN per defecte i la més utilitzada comunament. La paraula clau **INNER** és **opcional**.
- Retorna registres que tenen **valors coincidents en ambdues taules**.
- Amb aquesta operació cada registre a la taula A és combinat amb els corresponents de la taula B que satisfaguen les condicions que s'especifiquen en el predicat del JOIN.
 - Qualsevol registre de la taula A o de la taula B que no tinga un corresponent en l'altra taula és exclòs, i només apareixeran els que tinguen correspondència en l'altra taula.

2.1. COMBINACIÓ INTERNA: INNER JOIN

- La sintaxi general amb INNER és (forma explícita):

SELECT columnnes

FROM nom_taula1 INNER JOIN nom_taula2

ON columna_taula1 = columna_taula2

WHERE condicio;

2.1. COMBINACIÓ INTERNA: INNER JOIN

- Vegem un exemple:
- La sentència següent uneix la primera taula (cistella_a) amb la segona taula (cistella_b) fent coincidir els valors de les columnes fruita_a i fruita_b:

```
SELECT a,fruita_a, b, fruita_b
FROM cistella_a INNER JOIN cistella_b
ON fruita_a = fruita_b
```

- I s'obté com a resultat:

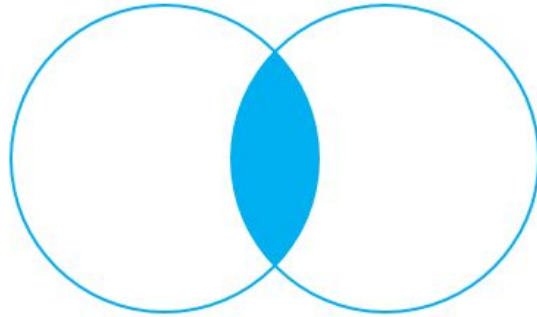
	 a integer	 fruita_a character varying (100)	 b integer	 fruita_b character varying (100)
1	1	Poma	2	Poma
2	2	Taronja	1	Taronja

2.1. COMBINACIÓ INTERNA: INNER JOIN

- **Què ha fet internament?** **INNER JOIN** examina cada fila de la primera taula (cistella_a) i compara el valor de la columna fruita_a amb el valor de la columna fruita_b de cada fila de la 2^a taula (cistella_b).
- Si aquests valors són iguals, es crea una nova fila que conté columnes de les dues taules i afegeix a aquesta nova fila el conjunt de resultats.

2.1. COMBINACIÓ INTERNA: INNER JOIN

- El següent diagrama de Venn il·lustra la unió interior:



INNER JOIN



Quan es combinen columnes amb valors nuls (NULL), cal recordar que el valor null no es combina amb un altre valor, excepte quan se li afegeixen predicats com ara IS NULL o IS NOT NULL.

2.2. COMBINACIÓ EXTERNA: OUTER JOIN

- Mitjançant aquesta operació no es requereix que un registre en una taula estiga relacionat amb altre registre de l'altra taula.
 - El registre es manté en la taula combinada encara que no existisca el corresponent en l'altra taula.
- Hi ha **tres tipus** de combinacions externes: **LEFT JOIN**, **RIGHT JOIN** i **FULL JOIN**, on es prenen tots els registres de la taula de l'esquerra, o tots els de la taula dreta, o tots els registres respectivament.

2.2.1. COMBINACIÓ EXTERNA: LEFT (OUTER) JOIN





- **LEFT (OUTER) JOIN** retorna tots els registres de la taula de l'esquerra i els registres coincidents de la taula de la dreta.
 - No és obligatori incloure OUTER, LEFT JOIN és el mateix que LEFT OUTER JOIN, de manera que podeu utilitzar-los indistintament.
- En el context de LEFT JOIN, la primera taula s'anomena taula esquerra i la segona taula es diu taula dreta.

2.2.1. COMBINACIÓ EXTERNA: LEFT (OUTER) JOIN

- Vegem un exemple: La sentència següent utilitza la clàusula de combinació esquerra per unir la taula cistella_a amb la taula cistella_b.

```
SELECT a, fruita_a, b, fruita_b  
FROM cistella_a LEFT JOIN cistella_b  
ON fruita_a = fruita_b;
```

- I s'obté com a resultat:

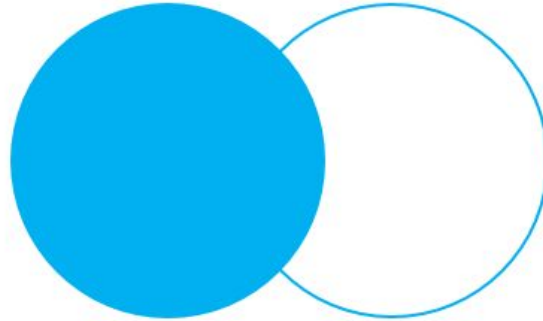
	 a integer	 fruita_a character varying (100)	 b integer	 fruita_b character varying (100)
1		1 Poma		2 Poma
2		2 Taronja		1 Taronja
3		3 Platan		[null] [null]
4		4 Cireres		[null] [null]

2.2.1. COMBINACIÓ EXTERNA: LEFT (OUTER) JOIN

- **Què ha fet internament?** **LEFT JOIN** comença a seleccionar dades de la taula esquerra. Compara els valors de la columna fruita_a amb els valors de la columna fruita_b de la taula cistella_b.
- Si aquests valors són iguals, crea una fila nova que conté columnes de les dues taules i afegeix aquesta nova fila al conjunt de resultats.
- En cas que els valors no siguin iguals, LEFT JOIN també crea una fila nova que conté columnes de les dues taules i l'afegeix al conjunt de resultats. Però omple les columnes de la taula dreta (cistella_b) amb nuls.

2.2.1. COMBINACIÓ EXTERNA: LEFT (OUTER) JOIN

- El següent diagrama de Venn il·lustra la combinació esquerra:



LEFT OUTER JOIN

2.2.2. COMBINACIÓ EXTERNA: RIGHT (OUTER) JOIN





- **RIGHT (OUTER) JOIN** retorna tots els registres de la taula dreta i els registres coincidents de la taula esquerra.
- La combinació dreta és una versió invertida de la combinació esquerra. Comença a seleccionar les dades de la taula dreta.
- El RIGHT JOIN i el RIGHT OUTER JOIN són el mateix, per tant, podeu utilitzar-los indistintament.

2.2.1. COMBINACIÓ EXTERNA: LEFT (OUTER) JOIN

- Vegem un exemple: La sentència següent utilitza la clàusula de combinació dreta per unir la taula cistella_a amb la taula cistella_b.

```
SELECT a, fruita_a, b, fruita_b
FROM cistella_a RIGHT JOIN cistella_b
ON fruita_a = fruita_b;
```

- I s'obté com a resultat:

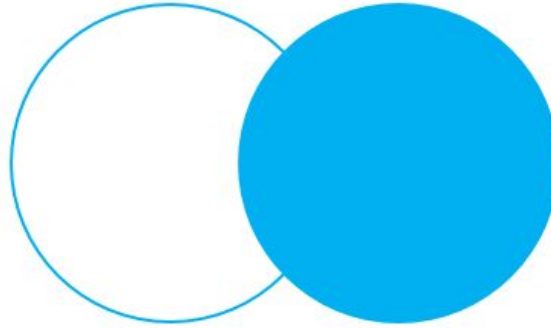
	 a integer	 fruita_a character varying (100)	 b integer	 fruita_b character varying (100)
1		2 Taronja		1 Taronja
2		1 Poma		2 Poma
3		[null] [null]		3 Sindria
4		[null] [null]		4 Pera

2.2.1. COMBINACIÓ EXTERNA: RIGHT (OUTER) JOIN

- **Què ha fet internament?** **RIGHT JOIN** compara cada valor de la columna fruita_b de cada fila de la taula dreta amb cada valor de la columna fruita_a de cada fila de la taula esquerra.
- Si aquests valors són iguals, es crea una fila nova que conté columnes de les dues taules.
- En cas que aquests valors no siguin iguals, la també crea una fila nova que conté columnes de les dues taules. Però, omple les columnes de la taula esquerra amb NULL.

2.2.1. COMBINACIÓ EXTERNA: RIGHT (OUTER) JOIN

- El següent diagrama de Venn il·lustra el RIGHT JOIN:



RIGHT OUTER JOIN

2.2.3. COMBINACIÓ EXTERNA: FULL (OUTER) JOIN

- **FULL (OUTER) JOIN** retorna tots els registres quan hi ha una coincidència en la taula esquerra o dreta.
- En altres paraules, la **combinació exterior completa** retorna un **conjunt de resultats que conté totes les files de les taules esquerra i dreta**, amb les files coincidents dels dos costats si estan disponibles. En cas que no hi haja cap coincidència, les columnes de la taula s'ompliran amb NUL.

2.2.3. COMBINACIÓ EXTERNA: FULL (OUTER) JOIN

- Seguint amb l'exemple:

```
SELECT a, fruita_a, b, fruita_b
FROM cistella_a FULL OUTER JOIN cistella_b
ON fruita_a = fruita_b;
```

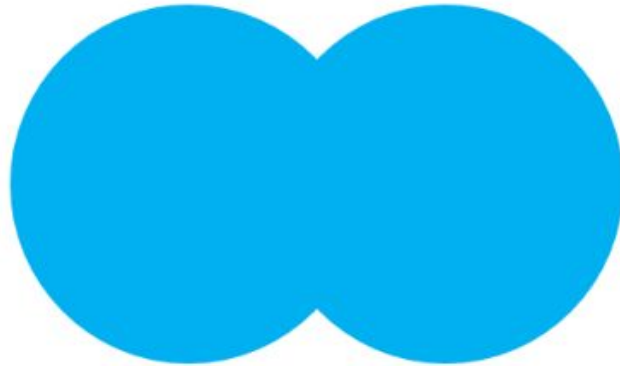


- I obtindríem com a resultat:

	 a integer	 fruita_a character varying (100)	 b integer	 fruita_b character varying (100)
1	1	Poma	2	Poma
2	2	Taronja	1	Taronja
3	3	Platan	[null]	[null]
4	4	Cireres	[null]	[null]
5	[null]	[null]	3	Sindria
6	[null]	[null]	4	Pera

2.2.3. COMBINACIÓ EXTERNA: FULL (OUTER) JOIN

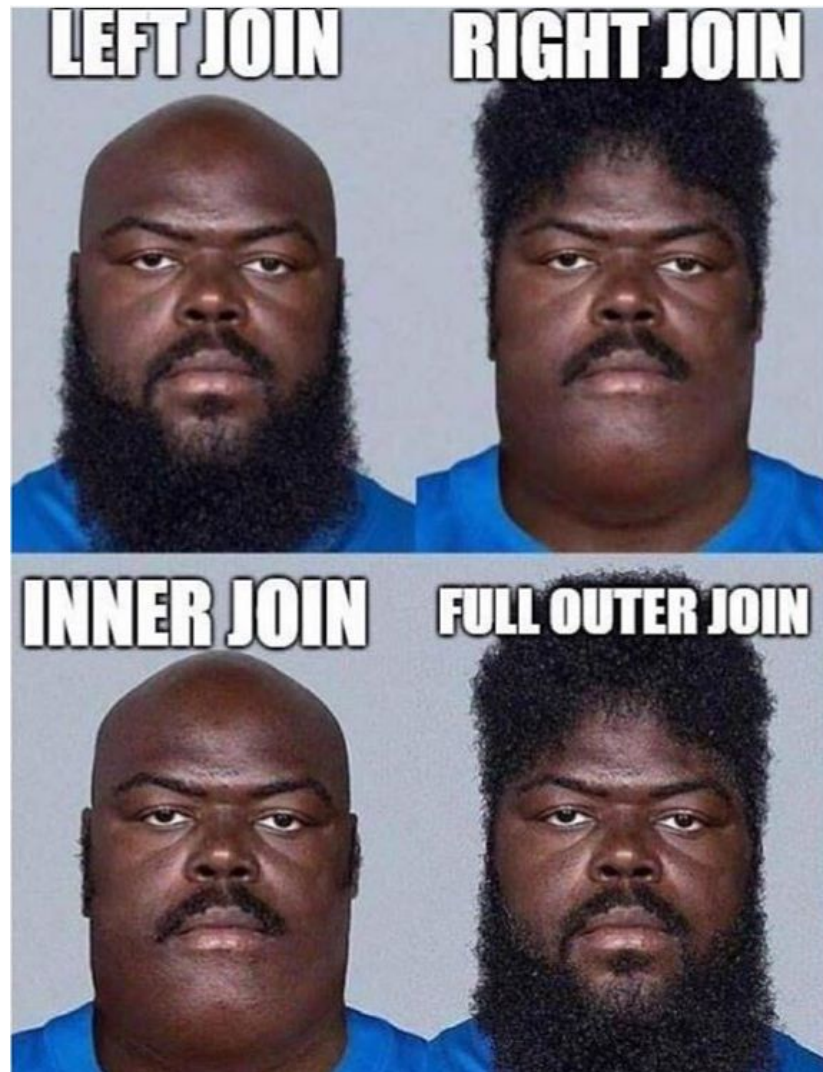
- La representació de la unió exterior completa amb diagrama de Venn seria:



FULL OUTER JOIN

2.2.3. RESUM

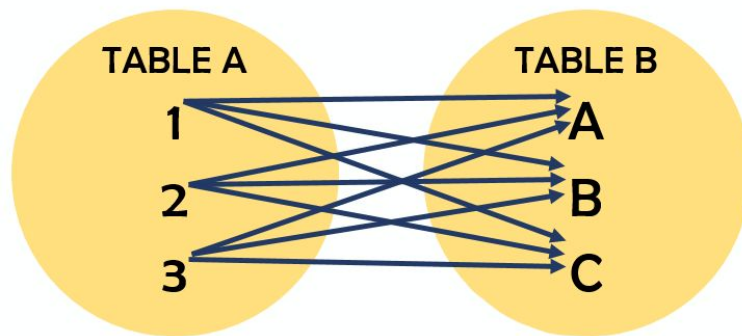
- Resum dels principals tipus de JOIN:



2.2.4. PRODUCTE CARTESIÀ: CROSS JOIN

- La clàusula **CROSS JOIN** genera una combinació aparellant de cada fila de la primera taula amb cada fila de la segona taula.
 - Aquest tipus d'unió també es coneix com a unió cartesiana o producte cartesià.

CROSS JOIN



2.2.4. PRODUCTE CARTESIÀ: CROSS JOIN

- A diferència d'altres clàusules d'unió com LEFT JOIN o INNER JOIN, la clàusula **CROSS JOIN** no té predicat d'unió. És a dir, que **les taules no tenen perquè estar relacionades per un camp (columna) en comú.**

2.2.4. PRODUCTE CARTESIÀ: CROSS JOIN

- Supposeu que esteu asseguts a una cafeteria i decidiu demanar l'esmorzar. Examinarem el menú i comenceu a pensar quina combinació de menjars i begudes pot ser més gustosa. El vostre cervell rebrà aquest senyal i començarà a generar totes les combinacions de menjars i begudes.
- Doncs, CROSS JOIN funciona de manera similar a aquest mecanisme, ja que crea totes les combinacions.

Meals	
Omlet	
Fried Egg	
Sausage	

Drinks	
Orange Juice	
Tea	
Coffee	

CROSS JOIN

Menu Combination	
	
	
	
	
	
	
	
	
	

2.2.5. UNIÓ NATURAL: NATURAL JOIN

- Una **unió natural** és una unió que crea una **unió implícita basada en els mateixos noms de columna de les taules unides**.
- **NATURAL JOIN** pot ser una unió interior, unió esquerra o unió dreta.
- Si no especifiquem una unió explícitament, per ex., INNER JOIN, LEFT JOIN, RIGHT JOIN, **PostgreSQL** utilitzarà INNER JOIN per defecte.

2.2.5. UNIÓ NATURAL: NATURAL JOIN

- A continuació es mostra la sintaxi del NATURAL JOIN de PostgreSQL:

SELECT columnes

FROM taula1

NATURAL [INNER | LEFT | RIGHT] JOIN taula2;

2.2.5. UNIÓ NATURAL: NATURAL JOIN

- Per demostrar la combinació natural de PostgreSQL, crearem dues taules: categories i productes.
 - Cada categoria té zero o molts productes i cada producte pertany a una única categoria.

```
DROP TABLE IF EXISTS categories;  
CREATE TABLE categories (  
    categoria_id serial PRIMARY KEY,  
    categoria_nom VARCHAR (255) NOT NULL  
);
```

```
DROP TABLE IF EXISTS productes;  
CREATE TABLE products (  
    producte_id serial PRIMARY KEY,  
    producte_nom VARCHAR (255) NOT NULL,  
    categoria_id INT NOT NULL,  
    FOREIGN KEY (categoria_id)  
REFERENCES categories (categoria_id)  
);
```

2.2.5. UNIÓ NATURAL: NATURAL JOIN

- Les instruccions INSERT següents insereixen algunes dades a les taules de categories i productes:

```
INSERT INTO categories (categoria_nom)  
VALUES  
    ('Smart Phone'),  
    ('Laptop'),  
    ('Tablet');
```

```
INSERT INTO products (producte_nom,  
                      categoria_id)  
VALUES  
    ('iPhone', 1),  
    ('Samsung Galaxy', 1),  
    ('HP Elite', 2),  
    ('Lenovo Thinkpad', 2),  
    ('iPad', 3),  
    ('Kindle Fire', 3);
```

2.2.5. UNIÓ NATURAL: NATURAL JOIN

- La declaració següent utilitza **NATURAL JOIN** per unir la taula de productes amb la taula de categories: **SELECT * FROM products NATURAL JOIN categories;**
- I genera el següent resultat:

	category_id integer	product_id integer	product_name character varying (255)	category_name character varying (255)
1	1	1	iPhone	Smart Phone
2	1	2	Samsung Galaxy	Smart Phone
3	2	3	HP Elite	Laptop
4	2	4	Lenovo Thinkpad	Laptop
5	3	5	iPad	Tablet
6	3	6	Kindle Fire	Tablet

2.2.5. UNIÓ NATURAL: NATURAL JOIN

- La comoditat del NATURAL JOIN és que no requereix que especifiqueu la clàusula join (ON columna_taulaA = columna_taulaB) perquè utilitza una clàusula de join implícita basada en la columna amb nom comú.
- Tot i així, **millor evitar l'ús** de NATURAL JOIN sempre que siga possible, ja que de vegades pot provocar un resultat inesperat (per exemple, si les dues taules també tenen una altra columna comuna que s'utilitza en el NATURAL JOIN).

2.2.6. UNIÓ NATURAL: SELF JOIN

- Un **SELF JOIN** és una unió que uneix una taula amb ella mateixa utilitzant el Join Interior (INNER JOIN) o Join Esquerra (LEFT OUTER JOIN).
 - Normalment utilitzem aquest tipus d'unió per consultar dades jeràrquiques o per comparar files de la mateixa taula.
- Per formar una “autounió”, heu d'especificar la mateixa taula dues vegades amb diferents àlies de taula i proporcionar el predicat d'unió després de la paraula clau ON.

SELECT columnes

FROM nom_taula t1

[**INNER** | **LEFT** | **RIGHT**] **JOIN** nom_taula t2 **ON** predicat_join;

3. ÚS D'ÀLIES

- **Aplicacions pràctiques dels àlies de taula:**
- Els àlies de taula tenen diverses aplicacions pràctiques:
- **1) Utilitzar àlies de taula en taules amb nom llarg.** Per fer més llegibles les consultes, podem reanomenar les taules en les consultes.
- Per exemple, en lloc d'utilitzar l'expressió següent en una consulta:
nom_taula_molt_llarg.nom_columna, podríem assignar a la taula *nom_taula_molt_llarg* un àlies com:
nom_taula_molt_llarg AS nomcurt
- I fer referència al nom de columna de la taula anterior mitjançant l'àlies de la taula: *nomcurt.nom_columna*

3. ÚS D'ÀLIES

- **2) Utilitzar àlies de taula en clàusules de JOIN.** Normalment, utilitzem una clàusula de JOIN per consultar dades de diverses taules que tinguen el mateix nom de columna. Si feu servir el mateix nom de columna que prové de diverses taules sense qualificar-les completament, obtindreu un error.
- Per evitar aquest error, heu de qualificar aquestes columnes mitjançant la sintaxi següent: `nom_taula.nom_columna` o bé `alias_nomtaula.nom_columna`
- **3) Utilitzar àlies en taules “reflexives”.** Quan uniu una taula amb si mateixa, heu d'utilitzar àlies de taula. Això es deu al fet que fer referència a la mateixa taula diverses vegades dins d'una consulta retornarà error.