

CONSULTES MULTITAUULA - JOIN

PRODUCTE CARTESIÀ – CROSS JOIN

JOIN

- ✓ ISO SQL/87

- **CONDICIÓ JOIN AL WHERE-** Àlies de les taules

- ✓ ISO SQL/92

- **JOIN ON (condició JOIN)**

- **SIMPLIFICACION JOIN ON en EQUI-JOINS**

- ✓ **NATURAL JOIN**

Clau forana i clau principal amb el mateix nom i tipus de dades.

+: Sintaxi més simple i no fa falta àlies

-: Vigilar que no hi hagi més columnes amb el mateix nom a les dues taules. No queda clar camp de JOIN

- ✓ **JOIN USING:** Queda clar el camp del JOIN.

SELF-JOIN (AUTO-UNIÓ).

UNIÓ DE TRES TAULES

NO EQUI-JOIN (Unió no igualitària)

INTRODUCCIÓ

- Una **base de dades** és un conjunt de **TAULES RELACIONADES** entre elles.
- Les relacions entre les taules s'estableixen amb la **CLAU FORANA**.

EMPLOYEES			DEPARTMENTS		
EMPLOYEE_ID	LAST_NAME	DEPT_ID	DEPARTMENT_ID	DEPT_NAME	LOCATION_ID
100	King	90	10	Administration	1700
101	Kochhar	90	20	Marketing	1800
...			...		
202	Fay	20	110	Accounting	1700
205	Higgins	110	190	Contracting	1700
206	Gietz	110			

Les **CONSULTES MULTITAUULA** permeten obtenir i combinar **INFORMACIÓ DE MÉS D'UNA TAULA**.

Les taules que intervenen en la consulta estaran relacionades entre elles amb una **CLAU FORANA**.

PRODUCTE CARTESIÀ

Quin és el resultat d'un SELECT amb dues taules al FROM ?

- Uneix cada fila de la primera taula amb **TOTES LES FILES** de la segona taula.
- El resultat són **TOTES LES POSSIBLES COMBINACIONS** entre les files de les dues taules.
- Les files del resultat seran les files de la primera taula x les files de la segona taula.

S'anomena **PRODUCTE CARTESIÀ** de les dues taules.

EXEMPLE

```
SELECT *  
FROM COUNTRIES, REGIONS
```

PRODUCTE CARTESIÀ

COUNTRIES: 4 files

COUNTRY_ID	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2
Fila(s) 1 - 4 de 4		

REGIONS: 4 files

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa
Fila(s) 1 - 4 de 4	

CAMPS TAULA COUNTRIES

COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
CA	Canada	2	1	Europe
CA	Canada	2	2	Americas
CA	Canada	2	3	Asia
CA	Canada	2	4	Middle East and Africa
DE	Germany	1	1	Europe

US	United States of America	2	4	Middle East and Africa
16 filas devueltas en 0,02 segundos Descargar				

CROSS JOIN – UNIÓ CREUADA

Formalment, el producte cartesià és la **unió creuada** (**CROSS JOIN**) entre dues taules.

```
SELECT *  
FROM COUNTRIES CROSS JOIN REGIONS
```

16 files

COUNTRY_ID	COUNTRY_NAME	REGION_ID	REGION_ID	REGION_NAME
CA	Canada	2	1	Europe
CA	Canada	2	2	Americas
CA	Canada	2	3	Asia
CA	Canada	2	4	Middle East and Africa
DE	Germany	1	1	Europe
DE	Germany	1	2	Americas
DE	Germany	1	3	Asia
DE	Germany	1	4	Middle East and Africa
UK	United Kingdom	1	1	Europe
UK	United Kingdom	1	2	Americas
UK	United Kingdom	1	3	Asia
UK	United Kingdom	1	4	Middle East and Africa
US	United States of America	2	1	Europe
US	United States of America	2	2	Americas
US	United States of America	2	3	Asia
US	United States of America	2	4	Middle East and Africa

16 filas devueltas en 0,01 segundos [Descargar](#)

JOIN – EQUI-JOIN

- El resultat del PRODUCTE CARTESIÀ rara vegada interessa.
- Buscarem la **UNIÓ (JOIN)** entre les files de dues taules en funció dels valors de la **clau forana** d'una i la **clau principal** de l'altra.

EQUI-JOIN

JOIN en el què s'uneixen les files de dues taules que tinguin el **mateix valor** per a la clau forana d'una i la clau principal de l'altra.

```
clauForana taula1 = clauPrincipal taula2
```

UNIÓ - JOIN

Diferents sintaxis per al JOIN, pels **diferents estàndards SQL**:
SQL-ISO-87, SQL-92, SQL-99

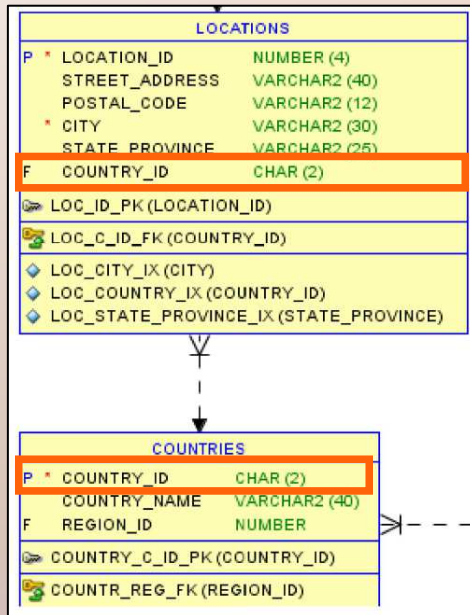
SQL-ISO-87

- Abans de l'aparició de la comanda **JOIN**.
- Les condicions del JOIN s'afegeixen a la clàusula **WHERE**, amb la resta de condicions de la consulta.
- És a dir, la **condició del JOIN** s'afegeix **com una condició més del WHERE**.

Alguna documentació ho cita com a anècdota històrica 😊

EQUI-JOIN AMB WHERE

RELACIÓ ENTRE COUNTRIES I LOCATIONS (*ubicació*)



Al traduir la relació 1:N entre COUNTRIES i LOCATIONS, s'afegeix la clau principal de COUNTRIES, **COUNTRY_ID**, com a clau forana a la taula LOCATIONS, amb el mateix nom que té com a clau principal, COUNTRY_ID.

Per unir cada LOCATION amb el seu COUNTRY:

```
SELECT *  
FROM LOCATIONS, COUNTRIES  
WHERE COUNTRY_ID = COUNTRY_ID
```



ORA-00918: columna definida de forma ambigua

MySQL diu:

#1052 - Column 'COUNTRY_ID' in where clause is ambiguous

La columna **COUNTRY_ID** està **repetida** en les dues taules. Els SGBDs no són capaços desfer l'ambigüitat en la comparació i generen un error.

IDENTIFICAR ELS CAMPS

Per desfer l'ambigüitat es pot posar com a prefix de les columnes el nom de la taula a la que pertanyen, per **qualificar-les** inequívocament:

```
SELECT *  
FROM LOCATIONS, COUNTRIES  
WHERE LOCATIONS.COUNTRY_ID = COUNTRIES.COUNTRY_ID
```

CAMPS TAULA LOCATION

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1800	460 Bloor St. W.	ON M5S 1X8	Toronto	Ontario	CA	CA	Canada	2
2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK	UK	United Kingdom	1
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US	US	United States of America	2
1500	2011 Interiors Blvd	99236	South San Francisco	California	US	US	United States of America	2
1700	2004 Charade Rd	98199	Seattle	Washington	US	US	United States of America	2

5 filas devueltas en 0,02 segundos [Descargar](#)

CAMPS TAULA COUNTRIES

Identificar els camps amb el noms de la taula fa que el codi de les consultes sigui més llarg d'escriure i més difícil de llegir !!

ÀLIES DE LES TAULES

Un àlies és un **nom curt** (el més curt possible) que s'assigna a cada taula a la clàusula FROM. La consulta és més curta, fàcil d'escriure i d'entendre.

```
SELECT *  
FROM LOCATIONS L, COUNTRIES C  
WHERE L.COUNTRY_ID = C.COUNTRY_ID
```

Àlies típic: inicial de la taula

CAMPS TAULA LOCATION

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE
1800	460 Bloor St. W.	ON M5S 1X8	Toronto	Ontario
2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford
1400	2014 Jabberwocky Rd	26192	Southlake	Texas
1500	2011 Interiors Blvd	99236	South San Francisco	California
1700	2004 Charade Rd	98199	Seattle	Washington

5 filas devueltas en 0,02 segundos [Descargar](#)

CAMPS TAULA COUNTRIES

COUNTRY_ID	COUNTRY_ID	COUNTRY_NAME	REGION_ID
CA	CA	Canada	2
UK	UK	United Kingdom	1
US	US	United States of America	2
US	US	United States of America	2
US	US	United States of America	2

NOTA: Formalment, els àlies de les taules s'escriuen amb **AS**, com els àlies dels camps dels SELECT, però no s'acostumen a posar.

PROJECCIÓ DEL JOIN

Seleccionem els camps de les dues taules que volem mostrar:

```
SELECT L.CITY, L.STATE_PROVINCE, L.COUNTRY_ID, C.COUNTRY_NAME  
FROM LOCATIONS L, COUNTRIES C  
WHERE L.COUNTRY_ID = C.COUNTRY_ID
```

CITY	STATE_PROVINCE	COUNTRY_ID	COUNTRY_NAME
Toronto	Ontario	CA	Canada
Oxford	Oxford	UK	United Kingdom
Southlake	Texas	US	United States of America
South San Francisco	California	US	United States of America
Seattle	Washington	US	United States of America

5 filas devueltas en 0,02 segundos [Descargar](#)

NOTA

No fa falta posar prefix a tots els camps del SELECT, només aquells que puguin generar ambigüitat (repetits en les dues taules).

Si es posa a tots els camps, queda més clar **de quina taula prové cada camp**.

EQUI-JOIN AMB SELECCIÓ

EXAMPLE

Mostrar el nom de la ciutat i el nom del país de totes les ciutats que tinguin com a codi de país 'US'

```
SELECT L.CITY, C.COUNTRY_NAME
FROM   COUNTRIES C, LOCATIONS L
WHERE  C.COUNTRY_ID = L.COUNTRY_ID
      AND
      UPPER(L.COUNTRY_ID) LIKE 'US'
```

CITY	COUNTRY_NAME
Southlake	United States of America
South San Francisco	United States of America
Seattle	United States of America
3 filas devueltas en 0,05 segundos Descargar	

SQL-ISO-92

- Nova sintaxi per als JOINS.
- **S'indica la condició del JOIN en la clàusula FROM.**
- Al WHERE només queden les condicions del SELECT.

SINTAXI

```
SELECT  
FROM taula1 t1 JOIN taula2 t2 ON (condició JOIN)  
WHERE [condicions SELECT]
```

Més entenedor per separar les condicions del JOIN de les condicions del SELECT.

La clàusula ON indica les condicions per les que es fa el JOIN.

EQUI JOIN: JOIN ON

EXEMPLE

Modifiquem l'exemple de COUNTRIES i LOCATIONS posant la condició del JOIN al JOIN enlloc de al WHERE:

```
SELECT L.CITY, L.STATE_PROVINCE, L.COUNTRY_ID, C.COUNTRY_NAME  
FROM LOCATIONS L JOIN COUNTRIES C ON (COUNTRY_ID = C.COUNTRY_ID)
```

CITY	STATE_PROVINCE	COUNTRY_ID	COUNTRY_NAME
Toronto	Ontario	CA	Canada
Oxford	Oxford	UK	United Kingdom
Southlake	Texas	US	United States of America
South San Francisco	California	US	United States of America
Seattle	Washington	US	United States of America

5 filas devueltas en 0,02 segundos [Descargar](#)

EXAMPLE JOIN-ON

EXAMPLE

Mostrar el nom de la ciutat i el nom del país de totes les ciutats que tinguin com a codi de país 'US' (*exemple diapositiva 12 amb where*)

```
SELECT L.CITY, C.COUNTRY_NAME
FROM   COUNTRIES C JOIN LOCATIONS L
      ON (C.COUNTRY_ID = L.COUNTRY_ID )
WHERE  UPPER(L.COUNTRY_ID) LIKE 'US'
```

La condició del JOIN queda separada de la resta de condicions del SELECT.

CITY	COUNTRY_NAME
Southlake	United States of America
South San Francisco	United States of America
Seattle	United States of America
3 filas devueltas en 0,05 segundos Descargar	

EXAMPLE JOIN-ON

EXAMPLE

Consulta que mostri el nom, salari i nivell de salari (grade_level) de cada empleat que tingui nivell 'A' o 'E':

```
SELECT E.LAST_NAME, E.SALARY, J.GRADE_LEVEL, J.LOWEST_SAL, J.HIGHEST_SAL
FROM   EMPLOYEES E JOIN JOB_GRADES J
      ON (E.SALARY BETWEEN J.LOWEST_SAL AND J.HIGHEST_SAL)
WHERE  UPPER(J.GRADE_LEVEL) IN ('A', 'E')
```

LAST_NAME	SALARY	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
Matos	2600	A	1000	2999
Vargas	2500	A	1000	2999
King	24000	E	15000	24999
Kochhar	17000	E	15000	24999
De Haan	17000	E	15000	24999
5 rows returned in 0.03 seconds Download				

NATURAL JOIN

DEFINICIÓ GENÈRICA DE NATURAL JOIN

Uneix les files d'una taula amb les files de l'altra que tinguin el **mateix valor** per a les columnes que tenen el **mateix nom** i són del mateix tipus a les dues taules.

El **camp d'unió** només surten **1 cop** al resultat (no es repeteixen).

DEFINICIÓ DE NATURAL JOIN EN TERMES DE CLAU FORANA

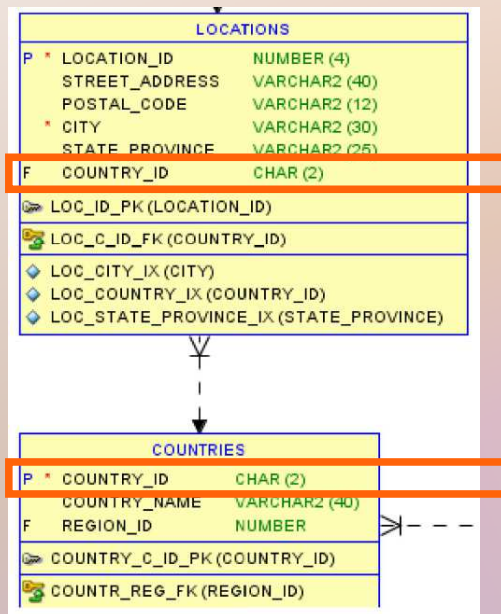
Quan dues taules estan vinculades amb una clau forana que té el mateix nom que la clau principal, el **NATURAL JOIN** uneix les files d'una taula amb les files de l'altra que tenen el mateix valor de la clau principal i de la clau forana.

La clau forana només es mostra un cop al resultat.

El **NATURAL JOIN** és una **simplificació** del **JOIN ON** en EQUI JOINS quan la clau forana i la clau principal tenen el mateix nom.

NATURAL JOIN

EXEMPLE: RELACIÓ ENTRE COUTRIES I LOCATIONS



Un NATURAL JOIN entre COUNTRIES i LOCATIONS, unirà cada fila de LOCATIONS amb la fila de COUNTRIES que tingui el mateix valor per al camp COUNTRY_ID (columna que és diu igual i és del mateix tipus en les dues taules).

Són “*equivalents*”:

```
SELECT *  
FROM LOCATIONS NATURAL JOIN COUNTRIES
```

```
SELECT *  
FROM LOCATIONS L JOIN COUNTRIES C  
ON (L.COUNTRY_ID=C.COUNTRY_ID)
```

NATURAL JOIN

LOCATIONS

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID
1800	460 Bloor St. W.	ON M5S 1X8	Toronto	Ontario	CA
2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1700	2004 Charade Rd	98199	Seattle	Washington	US

5 filas devueltas en 0,00 segundos [Descargar](#)

COUNTRIES

COUNTRY_ID	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

4 filas devueltas en 0,01 segundos [Descargar](#)

NATURAL JOIN

```
SELECT *  
FROM LOCATIONS NATURAL JOIN COUNTRIES
```

CAMP JOIN	CAMPS TAULA LOCATIONS					CAMPS COUNTRIES	
COUNTRY_ID	LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_NAME	REGION_ID
CA	1800	460 Bloor St. W.	ON M5S 1X8	Toronto	Ontario	Canada	2
UK	2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	United Kingdom	1
US	1400	2014 Jabberwocky Rd	26192	Southlake	Texas	United States of America	2
US	1500	2011 Interiors Blvd	99236	South San Francisco	California	United States of America	2
US	1700	2004 Charade Rd	98199	Seattle	Washington	United States of America	2

5 filas devueltas en 0,01 segundos [Descargar](#)

SELECT amb **JOIN ON** mostra dos cops COUNTRY_ID (un per cada taula) en el lloc que ocupen dintre de la taula.

SELECT amb **NATURAL JOIN** mostra el camp del JOIN (COUNTRY_ID) només un cop al principi.

NATURAL JOIN

COMPTE

El NATURAL JOIN combina les dues taules per **TOTS ELS CAMPS** que tinguin el mateix nom i siguin del mateix tipus.

Ens hem d'assegurar que no hi ha altres camps a les dues taules que es diguin igual i siguin del mateix tipus per què també passarien a formar part de la condició del JOIN.

NATURAL JOIN

CONTRA EXAMPLE

Mostrar l'id i el cognom dels treballadors i l'id i el nom del departament en el que treballen.

JOIN ON

```
SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID, DEPARTMENT_NAME  
FROM EMPLOYEES E JOIN DEPARTMENTS D  
ON (E.DEPARTMENT_ID = D.DEPARTMENT_ID)
```

19 filas devueltas en 0,02 segundos [Descargar](#)

NATURAL JOIN

```
SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID, DEPARTMENT_NAME  
FROM EMPLOYEES NATURAL JOIN DEPARTMENTS
```

12 rows returned in 0.00 seconds

Per què el NATURAL JOIN retorna menys files que el JOIN ON ???

NATURAL JOIN

CONTRA EXEMPLE

Hi ha dos camps que tenen el mateix nom en les dues taules: **DEMPARMENT_ID** i **MANAGER_ID**.

El **JOIN ON** mostra els empleats amb les dades del seu departament:

```
E.DEPARTMENT_ID=D.DEPARTMENT_ID
```

El **NATURAL JOIN** els empleats amb el departament que tingui el mateix ID i el mateix MANAGER_id

```
E.DEPARTMENT_ID=D.DEPARTMENT_ID  
AND  
E.MANAGER_ID=D. MANAGER_ID
```

NATURAL JOIN

CONTRA EXEMPLE

Recordar que el **NATURAL JOIN** mostra al principi de la taula els camps del JOIN.

Si mostrem tots els camps:

```
SELECT *  
FROM EMPLOYEES NATURAL JOIN DEPARTMENTS
```

CAMPS DEL JOIN

MANAGER_ID	DEPARTMENT_ID	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
201	20	202	Pat	Fay
124	50	141	Trenna	Rajs
124	50	142	Curtis	Davies
124	50	143	Randall	Matos
124	50	144	Peter	Vargas
103	60	104	Bruce	Ernst
103	60	107	Diana	Lorentz
149	80	174	Ellen	Abel
149	80	176	Jonathon	Taylor
100	90	101	Neena	Kochhar

NATURAL JOIN

AVANTAGES NATURAL JOIN

La sintaxi és més **senzilla** i **no fa falta àlies** de les taules per què no s'ha de posar la condició del JOIN ON.

INCONVENIENTS

No queda explícit el camp pel que es fa el JOIN. Si no anem amb cura pot fer join per més d'un camp.

PERSONALMENT

No protegeix les consultes davant de possibles futurs canvis de noms dels camps. Què passaria si canviem el nom d'un i és igual que un altre camp de l'altra taula del NATURAL JOIN ?

JOIN USING

Indica amb la clàusula **USING** el camp o camps pels que es fa el JOIN.

SINTAXI

```
SELECT  
FROM taula1 JOIN taula2 USING (condició JOIN)  
WHERE [condicions SELECT]
```

Indicant **explícitament** els camps del JOIN. **Evita el problema del NATURAL JOIN** quan hi ha més d'un camp amb el mateix nom en les dues taules.

EXEMPLE

Mostrar l'id i el cognom dels treballadors i l'id i el nom del departament en el que treballen.

```
SELECT EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID, DEPARTMENT_NAME  
FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
```

19 rows returned in 0.03 seconds

RESUM EQUI-JOINS

EXAMPLE

Per cada departament, mostra el nom del departament i la ciutat on és.

Condició del join al WHERE

```
SELECT DEPARTMENT_NAME, CITY  
FROM DEPARTMENTS D, LOCATIONS L  
WHERE D.LOCATION_ID=L.LOCATION_ID
```

JOIN ON

```
SELECT DEPARTMENT_NAME, CITY  
FROM DEPARTMENTS D JOIN LOCATIONS L  
ON (D.LOCATION_ID=L.LOCATION_ID)
```

NATURAL JOIN

```
SELECT DEPARTMENT_NAME, CITY  
FROM DEPARTMENTS NATURAL JOIN LOCATIONS
```

JOIN USING

```
SELECT DEPARTMENT_NAME, CITY  
FROM DEPARTMENTS JOIN LOCATIONS USING(LOCATION_ID)
```

RESUM EQUI-JOINS

EXEMPLE

Per cada departament, mostra el nom del departament i la ciutat on és.

DEPARTMENT_NAME	CITY
Marketing	Toronto
Sales	Oxford
IT	Southlake
Shipping	South San Francisco
Administration	Seattle
Executive	Seattle
Accounting	Seattle
Contracting	Seattle
Public Relations	Seattle
9 rows returned in 0.02 seconds Download	

JOIN USING I NATURAL JOIN

ATENCIÓ

Els camps del JOIN amb un NATURAL JOIN o amb un USING només hi són un cop a la taula resultat del SELECT. **Si intentem qualificar-los donarà un error.**

```
SELECT L.COUNTRY_ID, CITY, COUNTRY_NAME  
FROM COUNTRIES C NATURAL JOIN LOCATIONS L  
WHERE UPPER(COUNTRY_ID) LIKE 'US'
```



ORA-25155: la columna utilizada en la unión NATURAL no puede tener un cualificador

```
SELECT L.COUNTRY_ID, CITY, COUNTRY_NAME  
FROM COUNTRIES C JOIN LOCATIONS L USING(COUNTRY_ID)  
WHERE UPPER(COUNTRY_ID) LIKE 'US'
```



ORA-25154: la parte de columna de la cláusula USING no puede tener un cualificador

RESUM JOINS AMB MYSQL

EXAMPLE

Mostrar el número i el nom dels empleats i nom del departament en el que treballen.

```
SELECT E.EMP_NO, E.COGNOM, D.DNOM  
FROM EMP E, DEPT D  
WHERE E.DEPT_NO = D.DEPT_NO
```

```
SELECT E.EMP_NO, E.COGNOM, D.DNOM  
FROM EMP E JOIN DEPT D  
ON (E.DEPT_NO = D.DEPT_NO)
```

```
SELECT EMP_NO, COGNOM, DNOM  
FROM EMP NATURAL JOIN DEPT
```

```
SELECT EMP_NO, COGNOM, DNOM  
FROM EMP JOIN DEPT USING (DEPT_NO)
```

EMP_NO	COGNOM	DNOM
7369	SANCHEZ	INVESTIGACIÓ
7499	ARROYO	VENDES
7521	SALA	VENDES
7566	JIMENEZ	INVESTIGACIÓ
7654	MARTÍN	VENDES
7698	NEGRO	VENDES
7782	CEREZO	COMPTABILITAT
7788	GIL	INVESTIGACIÓ
7839	REY	COMPTABILITAT
7844	TOVAR	VENDES
7876	ALONSO	INVESTIGACIÓ
7900	JIMENO	VENDES
7902	FERNÁNDEZ	INVESTIGACIÓ
7934	MUÑOZ	COMPTABILITAT

UNIÓ DE 3 TAULES

EXAMPLE

Consulta que mostri el nom de l'empleat, el nom del departament en el que treballa i la ciutat no és el departament.

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME, l.city
FROM EMPLOYEES E JOIN DEPARTMENTS D ON (E.DEPARTMENT_ID= D.DEPARTMENT_ID)
      JOIN LOCATIONS L      ON (D.LOCATION_ID=L.LOCATION_ID)
```

LAST_NAME	DEPARTMENT_NAME	CITY
Hartstein	Marketing	Toronto
Fay	Marketing	Toronto
Zlotkey	Sales	Oxford
Abel	Sales	Oxford
Taylor	Sales	Oxford
Hunold	IT	Southlake
Ernst	IT	Southlake
Lorentz	IT	Southlake
Mourgos	Shipping	South San Francisco
Rajs	Shipping	South San Francisco
Davies	Shipping	South San Francisco
Matos	Shipping	South San Francisco
Vargas	Shipping	South San Francisco
Whalen	Administration	Seattle

Si hi ha 20 empleats,
per què es mostren 19 files al resultat?

UNIÓ DE 3 TAULES

EXAMPLE

Consulta que mostri el nom de l'empleat, el nom del departament en el que treballa i la ciutat no és el departament.

Solucions amb els 3 tipus de JOIN i les condicions del JOIN la WHERE

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME, L.CITY
FROM   EMPLOYEES E, DEPARTMENTS D
WHERE  E.DEPARTMENT_ID = D.DEPARTMENT_ID
       AND
       D.LOCATION_ID=L.LOCATION_ID
```

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME, L.CITY
FROM   EMPLOYEES E JOIN DEPARTMENTS D
       ON (E. LOCATIONS = D.DEPARTMENT_ID)
       JOIN LOCATIONS L
       ON (D.LOCATION_ID=L.LOCATION_ID)
```

```
SELECT LAST_NAME, DEPARTMENT_NAME, CITY
FROM   EMPLOYEES NATURAL JOIN DEPARTMENTS
       NATURAL JOIN LOCATIONS
```

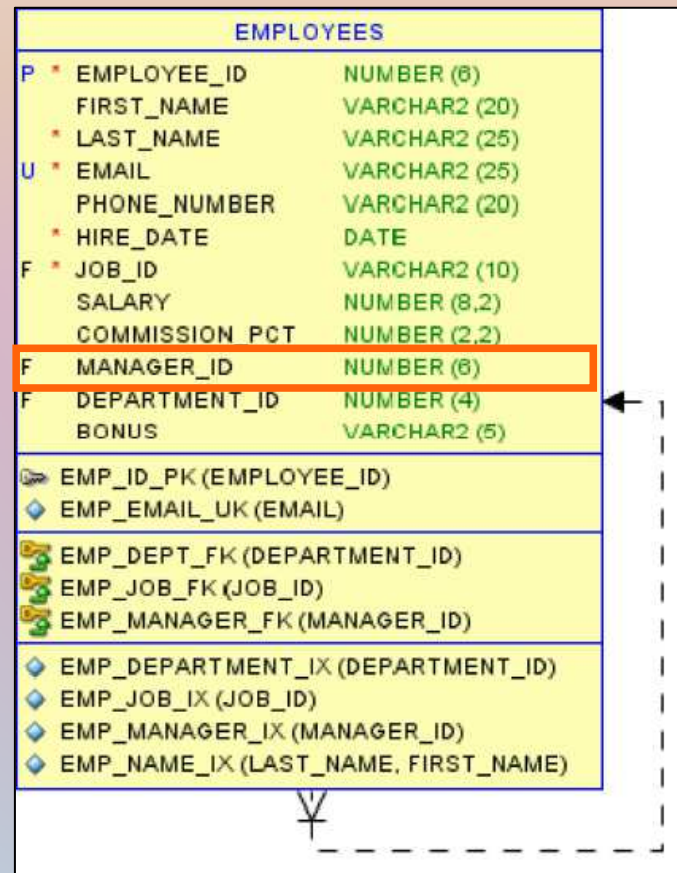
```
SELECT LAST_NAME, DEPARTMENT_NAME, CITY
FROM   EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)
       JOIN LOCATIONS USING (LOCATION_ID)
```


SELF-JOIN (AUTO-UNIÓ)

RELACIONS REFLEXIVES: quan una està relacionada amb ella mateixa.

EXAMPLE

Mostrar el nom de cada treballador i el del seu cap



SELF-JOIN (AUTO-UNIÓ)

EXAMPLE

Mostrar el nom de cada treballador i del seu cap.

```
SELECT EMPLEAT.LAST_NAME AS "EMPLEAT", MANAGER.LAST_NAME AS "CAP"  
FROM EMPLOYEES EMPLEAT JOIN EMPLOYEES MANAGER  
ON (EMPLEAT.MANAGER_ID = MANAGER.EMPLOYEE_ID);
```

EMPLEAT	CAP
Kochhar	King
De Haan	King
Zlotkey	King
Mourgos	King
Hartstein	King
Whalen	Kochhar
Higgins	Kochhar
Hunold	De Haan
Gietz	Higgins
Abel	Zlotkey
Taylor	Zlotkey
Grant	Zlotkey

Es pot utilitzar NATURAL JOIN
o JOIN USING ?

SELF-JOIN (AUTO-UNIÓ)

EXAMPLE (Solució amb WHERE)

Mostrar el nom de cada treballador i del seu cap

```
SELECT EMPLEAT.LAST_NAME AS "EMPLEAT", MANAGER.LAST_NAME AS "CAP"  
FROM EMPLOYEES EMPLEAT, EMPLOYEES MANAGER  
WHERE EMPLEAT.MANAGER_ID = MANAGER.EMPLOYEE_ID
```

EMPLEAT	CAP
Kochhar	King
De Haan	King
Zlotkey	King
Mourgos	King
Hartstein	King
Whalen	Kochhar
Higgins	Kochhar
Hunold	De Haan
Gietz	Higgins
Abel	Zlotkey
Taylor	Zlotkey
Grant	Zlotkey

NO EQUI-JOIN (Unió no igualitària)

La **condició** del JOIN ON pot ser **qualsevol**.
No hi ha cap obligació de que sigui una condició d'igualtat.

EXAMPLE

Mostrar el nom, salari i nivell de salari (*grade_level*) de cada empleat.

El GRADE_LEVEL és a la taula **JOB_GRADES**, que no té cap camp en comú amb la taula EMPLOYEES.

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

S'haurà de comprovar que el salari de l'empleat estigui entre el salari més petit i més alt de cada nivell: **BETWEEN -AND**

EXAMPLE JOIN-ON NO EQUI JOIN

EXAMPLE

Mostrar el nom, salari i nivell de salari (*grade_level*) de cada empleat:

```
SELECT E.LAST_NAME, E.SALARY, J.GRADE_LEVEL  
FROM   EMPLOYEES E JOIN JOB_GRADES J  
      ON (E.SALARY BETWEEN J.LOWEST_SAL AND J.HIGHEST_SAL)
```

LAST_NAME	SALARY	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
Vargas	2500	A	1000	2999
Matos	2600	A	1000	2999
Davies	3100	B	3000	5999
Rajs	3500	B	3000	5999
Lorentz	4200	B	3000	5999
Whalen	4400	B	3000	5999
Mourgos	5800	B	3000	5999
Fay	6000	C	6000	9999
Ernst	6000	C	6000	9999
Grant	7000	C	6000	9999
Hay más de 10 filas disponibles. Aumente el selector de filas para ver más.				
10 filas devueltas en 0,03 segundos Descargar				

Condició del join sobre camps amb **diferent nom** i condició diferent d'igualtat.

NO EQUI-JOIN → JOIN ON

NO EQUI-JOIN (Unió no igualitària)

EXAMPLE

Es vol saber el GRADE_LEVEL per al salari de cada empleat.

Solució amb la condició del JOIN al WHERE:

```
SELECT E.LAST_NAME, E.SALARY, J.GRADE_LEVEL, J.LOWEST_SAL, J.HIGHEST_SAL
FROM   EMPLOYEES E, JOB_GRADES J
WHERE  E.SALARY BETWEEN J.LOWEST_SAL AND J.HIGHEST_SAL
```

LAST_NAME	SALARY	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
Vargas	2500	A	1000	2999
Matos	2600	A	1000	2999
Davies	3100	B	3000	5999
Rajs	3500	B	3000	5999
Lorentz	4200	B	3000	5999
Whalen	4400	B	3000	5999
Mourgos	5800	B	3000	5999
Fay	6000	C	6000	9999
Ernst	6000	C	6000	9999