

# Sentències SQL Bàsiques IV

## (Visualització de dades de varies taules)



# Objectius

---



- Escriure sentències SELECT per accedir a dades de més d' una taula utilitzant unions d' igualtat i de no igualtat.
- Visualitzar dades que no compleixen condició d'unió utilitzant unions externes.
- Unir una taula amb si mateixa utilitzant una autounió.
- Totes els exemples estan basats en la base de dades **SCOTT**


# Obtenció de dades de varies taules

EMP

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



EMPNO	DEPTNO	LOC
7839	10	NEW YORK
7698	30	CHICAGO
7782	10	NEW YORK
7566	20	DALLAS
7654	30	CHICAGO
...		
14 rows selected.		

# Unió de taules

```
SELECT table1.column, table2.column  
FROM   table1
```

```
[CROSS JOIN table2] |  
[NATURAL JOIN table2] |  
[JOIN table 2 USING (column_name) |  
[JOIN table2  
    ON (table1.column_name = table2.column_name)] |  
[LEFT | RIGHT | FULL OUTER JOIN table2  
    ON (table1.column_name = table2.column_name)];
```

```
[WHERE condition(s)]  
[ORDER BY {column, expr, alias} [ASC|DESC]]  
[LIMIT {row_count} | FETCH {FIRST|NEXT}  
    [row_count] {ROW|ROWS} ONLY ]
```

# Què és una unió d'igualtat?

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		
14 rows selected.		

Clau forana

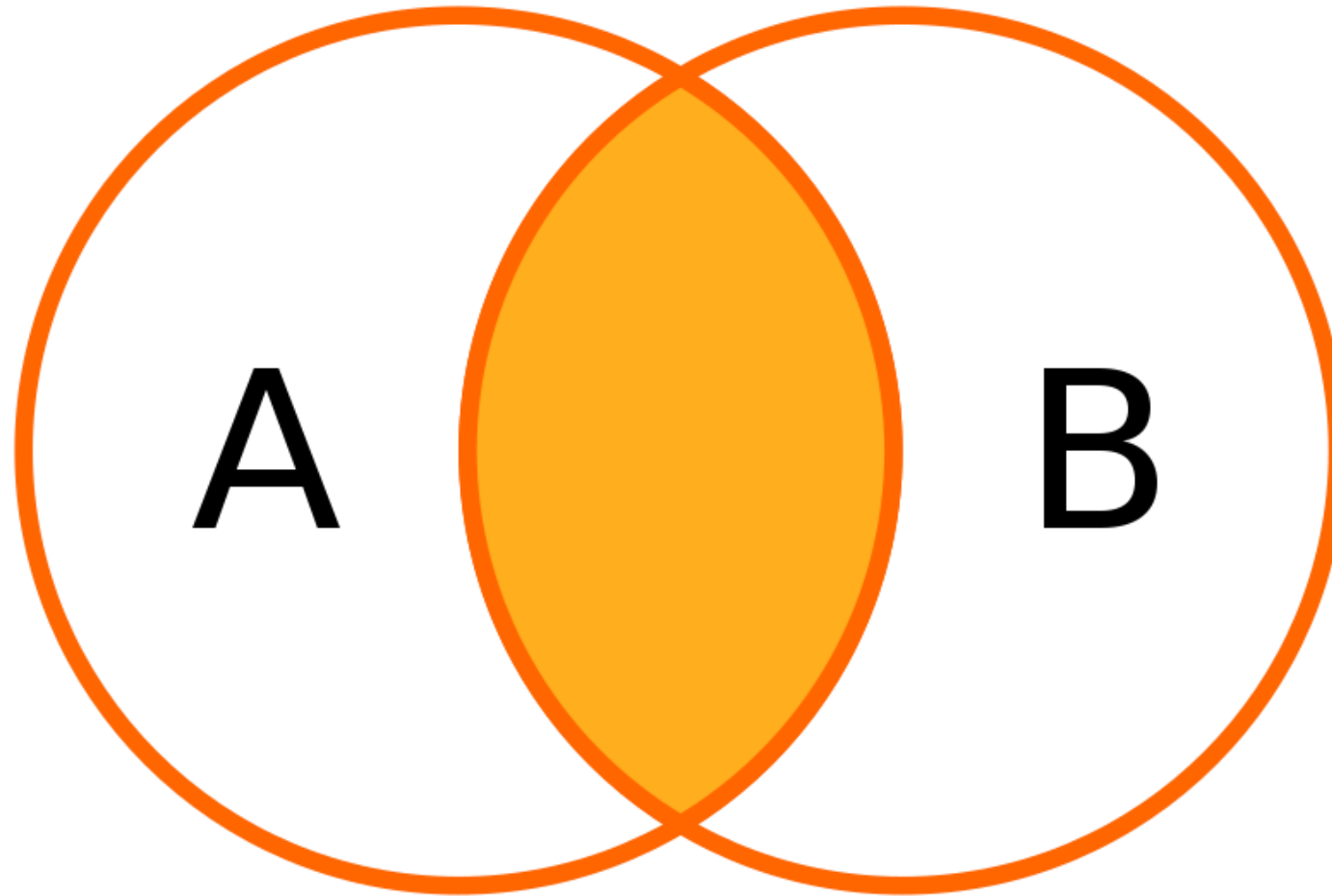
DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		
14 rows selected.		

Clau primària

# Què és una unió d'igualtat?

---



# Recuperació de registres amb unions d'igualtat

---

Podem utilitzar tres sintaxis diferents:

**NATURAL JOIN**

**INNER JOIN <TAULA> USING**

**INNER JOIN <TAULA> ON**

# Recuperació de registres amb unions d'igualtat

## NATURAL JOIN

```
SQL> SELECT *  
      FROM emp  
      NATURAL JOIN dept;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS

...

14 rows selected.



# Recuperació de registres amb unions d'igualtat

## USING

```
SQL> SELECT *  
      FROM emp  
      INNER JOIN dept USING (deptno) ;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS

...

14 rows selected.

# Recuperació de registres amb unions d'igualtat

## INNER JOIN (Combinació interna)

```
SQL> SELECT *  
      FROM emp  
      INNER JOIN dept ON emp.deptno = dept.deptno;
```

**IMPORTANT**

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS

...

14 rows selected.

# Creació d'unions amb la clàusula **ON**

---

- Per especificar condicions arbitràries o especificar columnes per unir, s'utilitza la clàusula **ON**.
- La condició d'unió es separa d'altres condicions de recerca.
- La clàusula **ON** facilita la compressió del codi.

```
SQL> SELECT      *  
      FROM        emp  
      INNER JOIN dept ON emp.deptno = dept.deptno;
```

# Condicions de cerca adicional amb clàusula WHERE

EMP			DEPT		
EMPNO	ENAME	DEPTNO	DEPTNO	DNAME	LOC
7839	KING	10	10	ACCOUNTING	NEW YORK
7698	BLAKE	30	30	SALES	CHICAGO
7782	CLARK	10	10	ACCOUNTING	NEW YORK
7566	JONES	20	20	RESEARCH	DALLAS
7654	MARTIN	30	30	SALES	CHICAGO
7499	ALLEN	30	30	SALES	CHICAGO
7844	TURNER	30	30	SALES	CHICAGO
7900	JAMES	30	30	SALES	CHICAGO
7521	WARD	30	30	SALES	CHICAGO
7902	FORD	20	20	RESEARCH	DALLAS
7369	SMITH	20	20	RESEARCH	DALLAS
...			...		
14 rows selected.			14 rows selected.		

```
SQL> SELECT *
      FROM emp
      INNER JOIN dept ON
emp.deptno = dept.deptno
WHERE emp.deptno=10;
```

# Qualificació de noms de columna ambigus

---

- Utilitzar **prefixos de taula** per qualificar noms de columnes que estiguin en diverses taules.
- Millorar el rendiment usant prefixos de taules.
- Distingir columnes, d'idèntic nom però pertanyents a diferents taules, usant **àlies de columnes**.

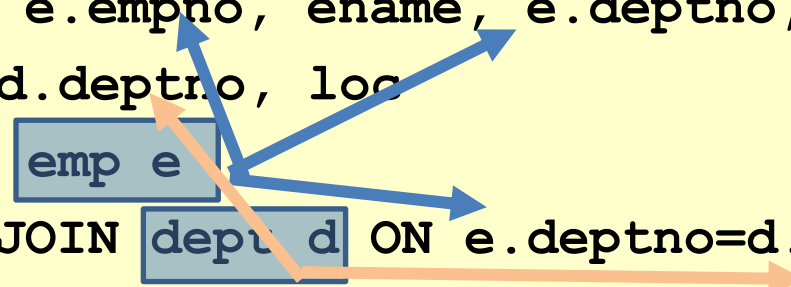
# Ús d'ALIAS de taula

## Prefixos de taula

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
           dept.deptno, dept.loc  
FROM emp  
INNER JOIN dept ON emp.deptno = dept.deptno;
```

## Alias de taula

```
SQL> SELECT e.empno, ename, e.deptno,  
           d.deptno, loc  
FROM emp e  
INNER JOIN dept d ON e.deptno=d.deptno;
```



# Unió de més de dues taules

CUSTOMER

NAME	CUSTID
-----	-----
JOCKSPORTS	100
TKB SPORT SHOP	101
VOLLYRITE	102
JUST TENNIS	103
K+T SPORTS	105
SHAPE UP	106
WOMENS SPORTS	107
...	...
9 rows selected.	

ORD

CUSTID	ORDID
-----	-----
101	610
102	611
104	612
106	601
102	602
106	604
106	
...	...
21 rows selected.	

ITEM

ORDID	ITEMID
-----	-----
610	3
611	1
612	1
601	1
602	1
...	...
64 rows selected.	



Per unir n taules es necessita un mínim de n-1 condicions d'unió.

Ex: per tres taules, mínim dos unions

# Creació d'unions en tres sentits amb la clàusula JOIN

## o INNER JOIN-ON

BD: HR

```
SELECT employee_id, city, department_name
FROM employees e JOIN departments d
ON (e.department_id = d.department_id)
JOIN locations l
ON d.location_id = l.location_id;
```

```
SELECT employee_id, city, department_name
FROM employees e INNER JOIN departments d
ON (e.department_id = d.department_id)
INNER JOIN locations l
ON d.location_id = l.location_id;
```

**JOIN** i **INNER JOIN** són clàusules idèntiques. Depenent del SGBD accepten una o les dues. En Postgresql ambdues son vàlides.



# Unions de no igualtat

EMPLOYEES

EMPNO	ENAME	SAL
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
...		
14 rows selected.		

JOBS

JOB	MIN_SAL	MAX_SAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

“el salari a la taula  
EMPLOYEES, està entre  
qualsevol parell de rangs  
de Salari mínim i màxim.

# Recuperació de registres amb unions de no igualtat

BD: HR

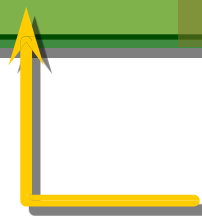
```
SELECT e.first_name, e.salary, j.job_title
FROM   employees e
JOIN   jobs j ON e.salary BETWEEN j.min_salary AND j.max_salary
WHERE  j.job_id = 1;
```

ENAME	SAL	JOB_TITLE
JAMES	950	Public Accountant
SMITH	800	Public Accountant
ADAMS	1100	Public Accountant
...		

14 rows selected.

# Unions externes (OUTER)

EMP			DEPT	
ENAME	DEPTNO		DEPTNO	DNAME
-----	-----		-----	-----
KING	10		10	ACCOUNTING
BLAKE	30		30	SALES
CLARK	10		10	ACCOUNTING
JONES	20		20	RESEARCH
...			...	
			40	OPERATIONS



No hi ha empleats en el  
departament OPERATIONS

# Unions **INNER** en front a **OUTER**

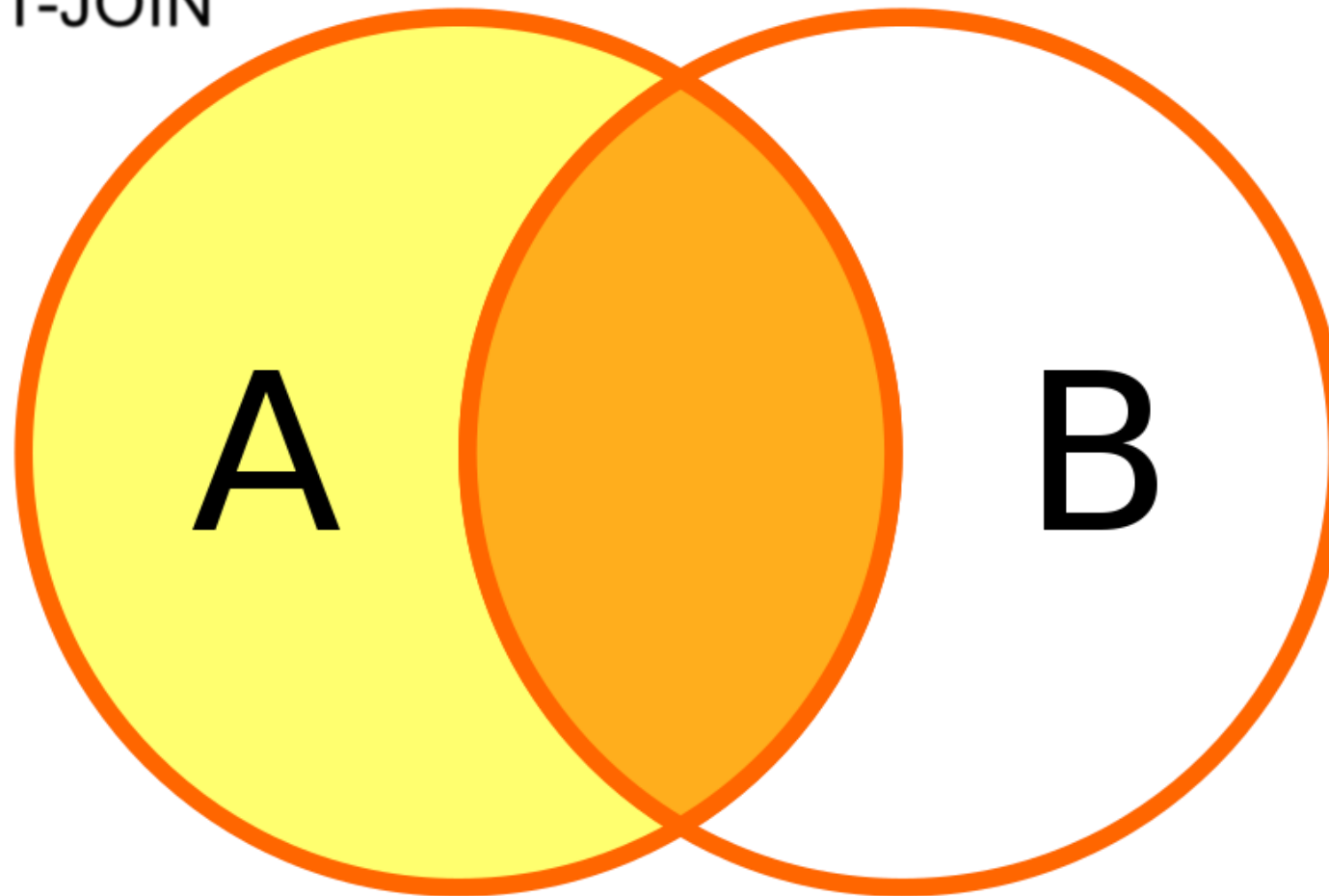
---

- La unió que retorna només files coincidents és una unió interna. (**INNER**)
- 
- La unió que retorna files coincidents, així com les files no coincidents de la taula esquerra o dreta, és unió externa esquerra o dreta. (**LEFT/RIGHT OUTER**)
- La unió que retorna files d'una unió interna i d'una unió esquerra i dreta és unió externa completa. (**OUTER**)

# Unions externes

---

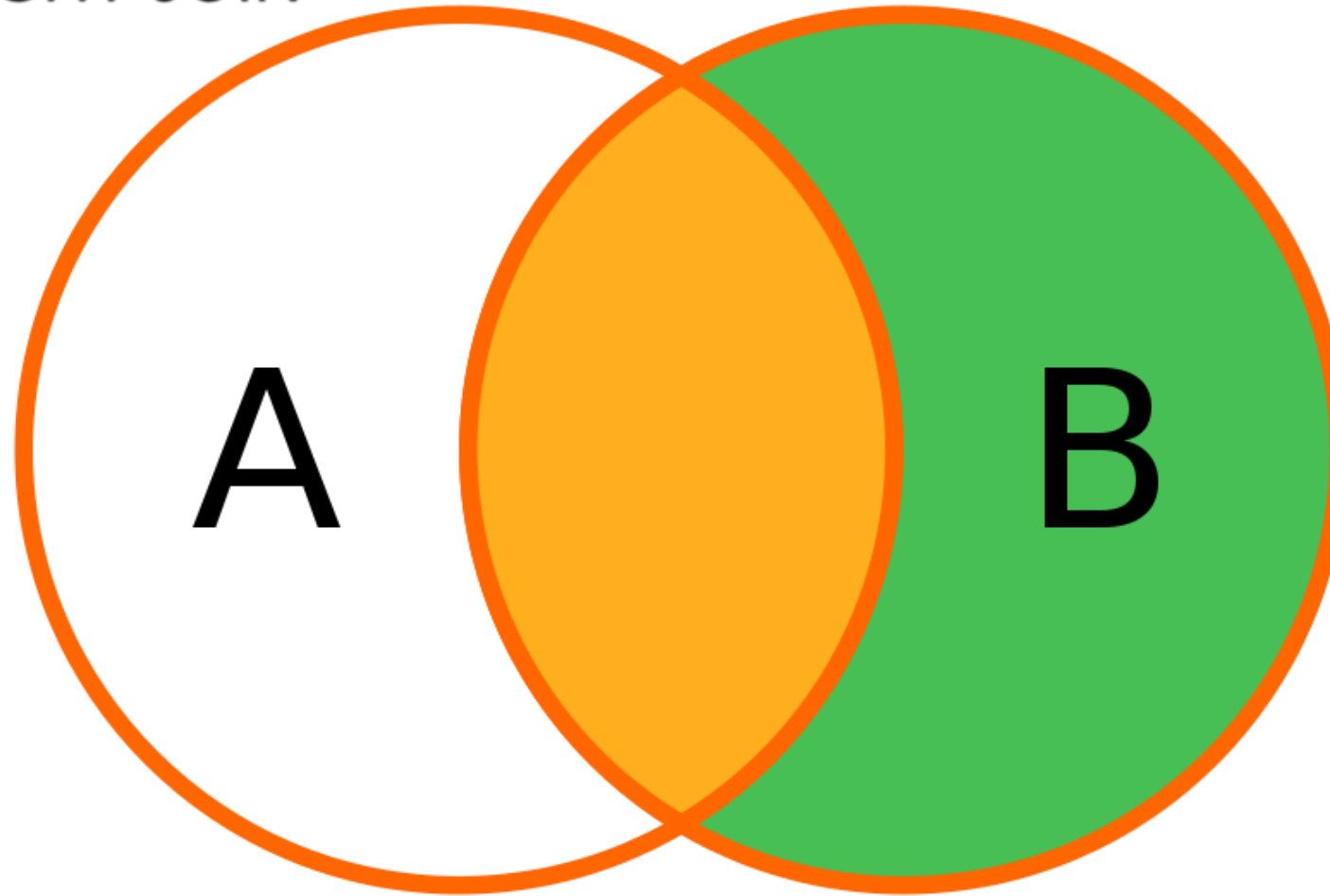
LEFT-JOIN



# Unions externes

---

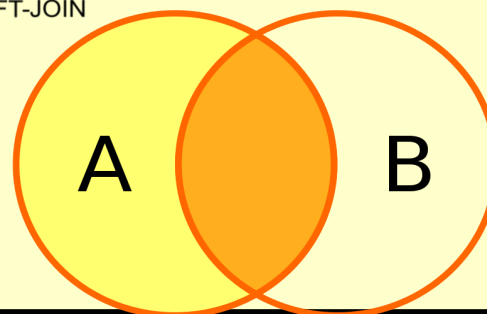
RIGHT-JOIN



# LEFT/RIGHT [OUTER] JOIN

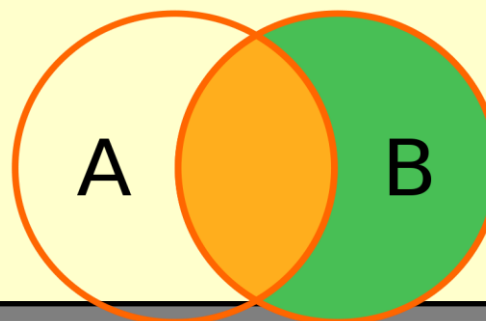
```
SELECT e.ename, e.deptno, d.dname
FROM emp e
LEFT JOIN dept d
ON (e.deptno = d.deptno);
```

LEFT-JOIN



(Desasignem departament 20 a la taula EMP al usuari SMITH)

```
SELECT e.ename, d.deptno, d.dname
FROM emp e
RIGHT JOIN dept d
ON (e.deptno = d.deptno);
```



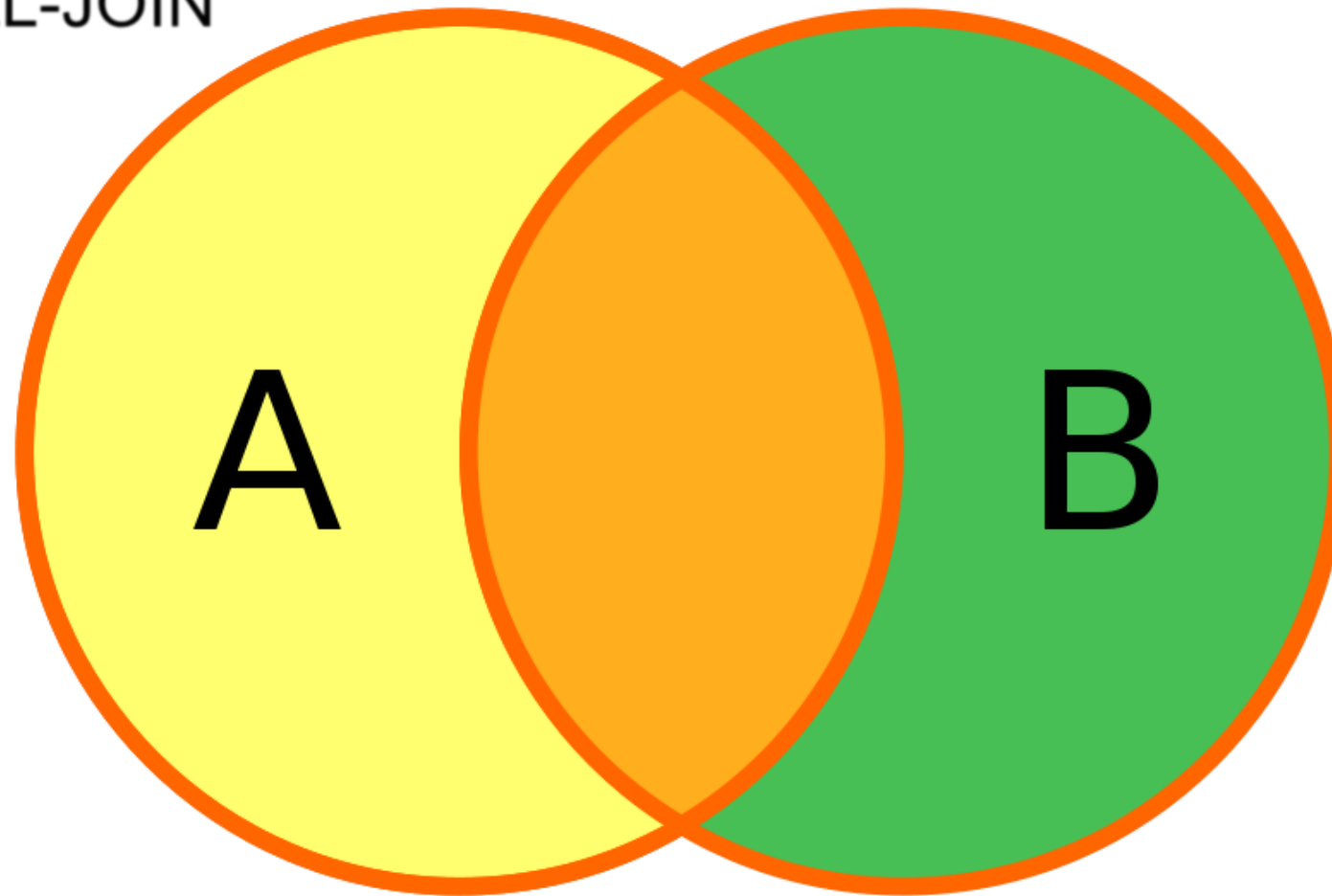
	ename text	deptno integer	dname text
2	WARD	30	SALES
3	JONES	20	RESEARCH
4	MARTIN	30	SALES
5	BLAKE	30	SALES
6	CLARK	10	ACCOUNTI...
7	SCOTT	20	RESEARCH
8	KING	10	ACCOUNTI...
9	TURNER	30	SALES
10	ADAMS	20	RESEARCH
11	JAMES	30	SALES
12	FORD	20	RESEARCH
13	MILLER	10	ACCOUNTI...
14	SMITH	[null]	[null]

	ename text	deptno integer	dname text
2	WARD	30	SALES
3	JONES	20	RESEARCH
4	MARTIN	30	SALES
5	BLAKE	30	SALES
6	CLARK	10	ACCOUNTI...
7	SCOTT	20	RESEARCH
8	KING	10	ACCOUNTI...
9	TURNER	30	SALES
10	ADAMS	20	RESEARCH
11	JAMES	30	SALES
12	FORD	20	RESEARCH
13	MILLER	10	ACCOUNTI...
14	[null]	40	OPERATIONS

# Unions Externes

---

FULL-JOIN





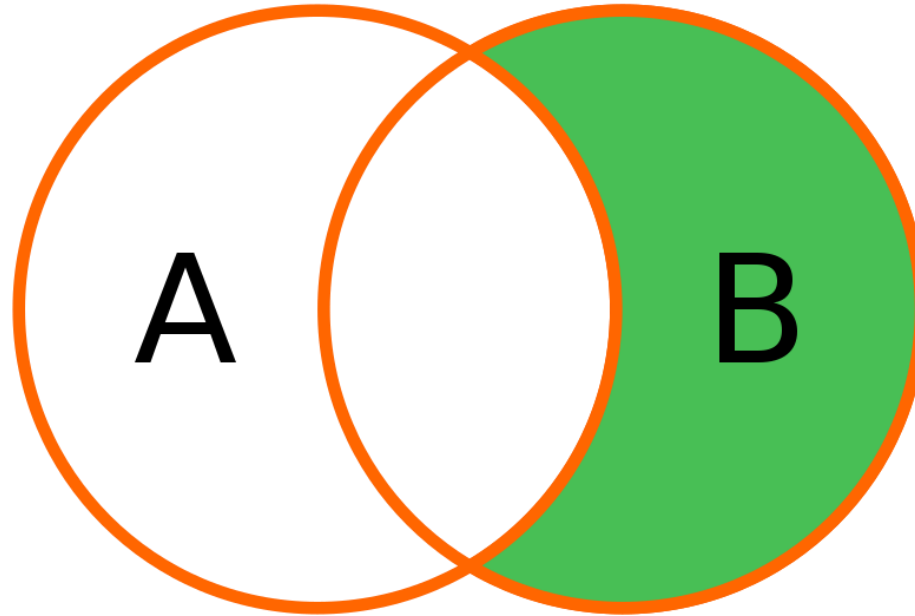
# FULL [OUTER] JOIN

- Recupera totes les files de la taula EMPLOYEES encara que no hi hagi cap coincidència en DEPARTMENTS i a la inversa.

```
SELECT e.ename, e.deptno, d.dname FROM emp e
FULL JOIN dept d
ON (e.deptno = d.deptno);
```

	ename text	deptno integer	dname text
1	ALLEN	30	SALES
2	WARD	30	SALES
3	JONES	20	RESEARCH
4	MARTIN	30	SALES
5	BLAKE	30	SALES
6	CLARK	10	ACCOUNTI...
7	SCOTT	20	RESEARCH
8	KING	10	ACCOUNTI...
9	TURNER	30	SALES
10	ADAMS	20	RESEARCH
11	JAMES	30	SALES
12	FORD	20	RESEARCH
13	MILLER	10	ACCOUNTI...
14	SMITH	[null]	[null]
15	[null]	40	OPERATIONS

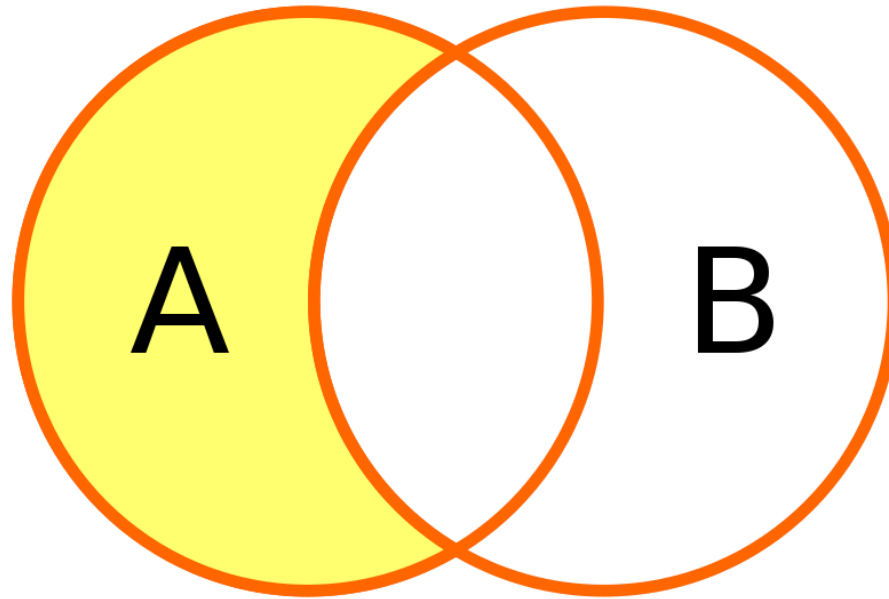
# RIGHT [OUTER] JOIN



	ename text	deptno integer	dname text
1	[null]	[null]	OPERATIONS

```
SELECT e.ename, e.deptno, d.dname
FROM emp e
RIGHT OUTER JOIN dept d ON (e.deptno = d.deptno)
WHERE (e.deptno IS NULL);
```

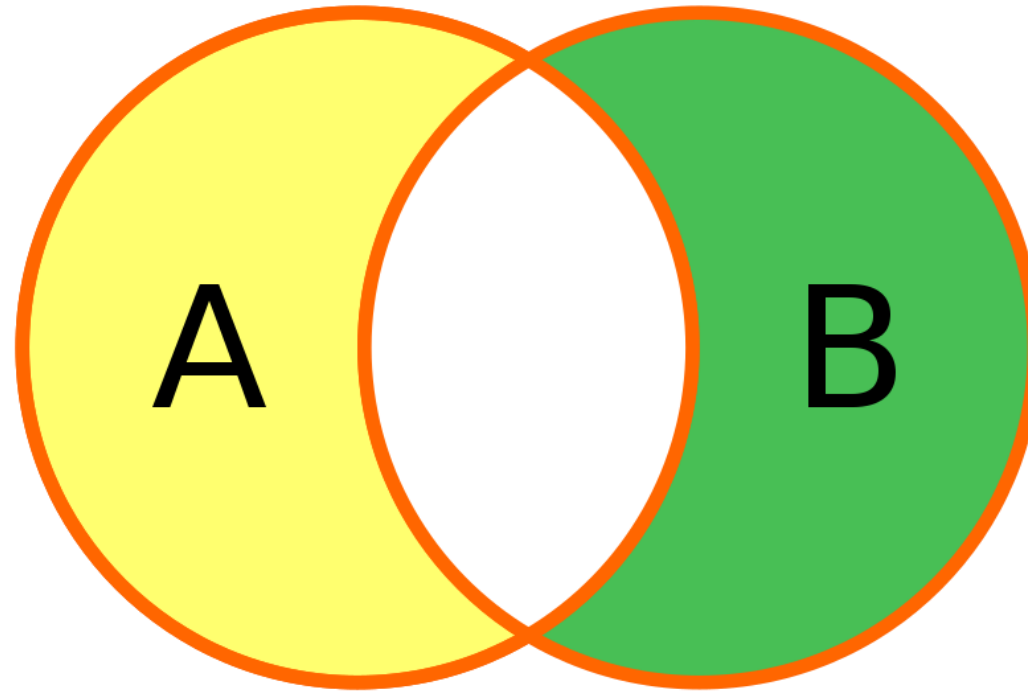
# LEFT [OUTER] JOIN



	ename text	deptno integer	dname text
1	SMITH	[null]	[null]

```
SELECT e.ename, e.deptno, d.dname
FROM emp e
LEFT OUTER JOIN dept d ON (e.deptno = d.deptno)
WHERE (d.deptno IS NULL);
```

# FULL [OUTER] JOIN



	ename text	deptno integer	dname text
1	SMITH	[null]	[null]
2	[null]	[null]	OPERATI...

```
SELECT e.ename, e.deptno, d.dname
FROM emp e
FULL OUTER JOIN dept d ON (e.deptno = d.deptno)
WHERE (e.deptno IS NULL OR d.deptno IS NULL);
```

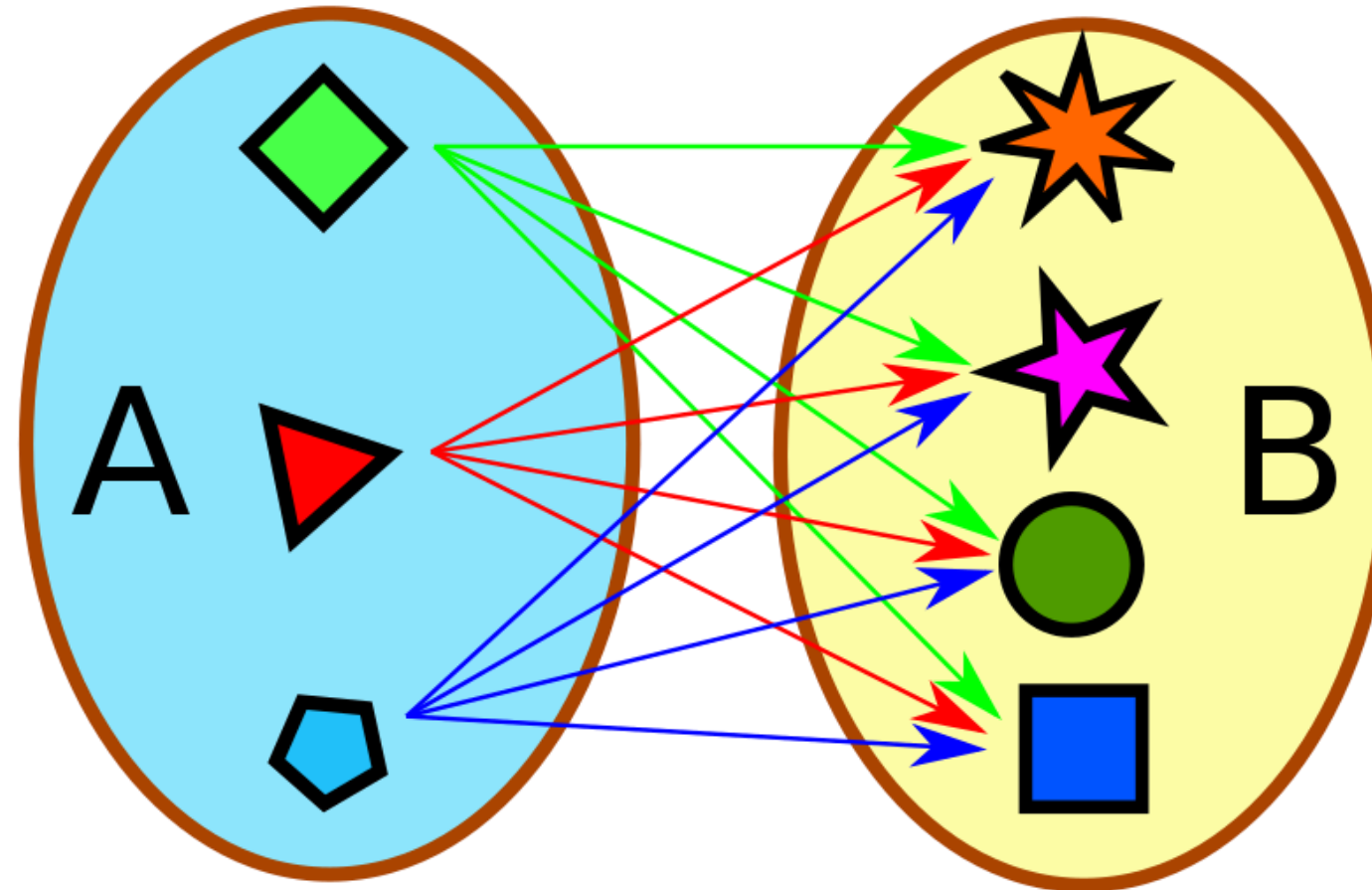
# Producte cartesià

---

Producte cartesià es forma quan:

- Una condició de join s'omet.
- Una condició de join no és vàlida.
- Totes les files de la primera taula s' uneixen a totes les files de la segona.
- Per evitar un producte cartesià, incloure una condició de join vàlida en la clàusula

# Producte cartesià



# Generació d'un producte cartesià


EMP (20 filas)

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

DEPT (8 filas)

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON


**SELECT** ename, dname  
**FROM** emp, dept

“Producto  
Cartesiano:   
20\*8=160 filas”

ENAME	DNAME
-----	-----
KING	ACCOUNTING
BLAKE	ACCOUNTING
...	
KING	RESEARCH
BLAKE	RESEARCH
...	
160 rows selected.	

# Autounions

EMP (WORKER)			EMP (MANAGER)	
EMPNO	ENAME	MGR	EMPNO	ENAME
7839	KING		7839	KING
7698	BLAKE	7839	7839	KING
7782	CLARK	7839	7839	KING
7566	JONES	7839	7839	KING
7654	MARTIN	7698	7698	BLAKE
7499	ALLEN	7698	7698	BLAKE



“MGR a la taula WORKER es igual a  
EMPNO a la taula MANAGER”

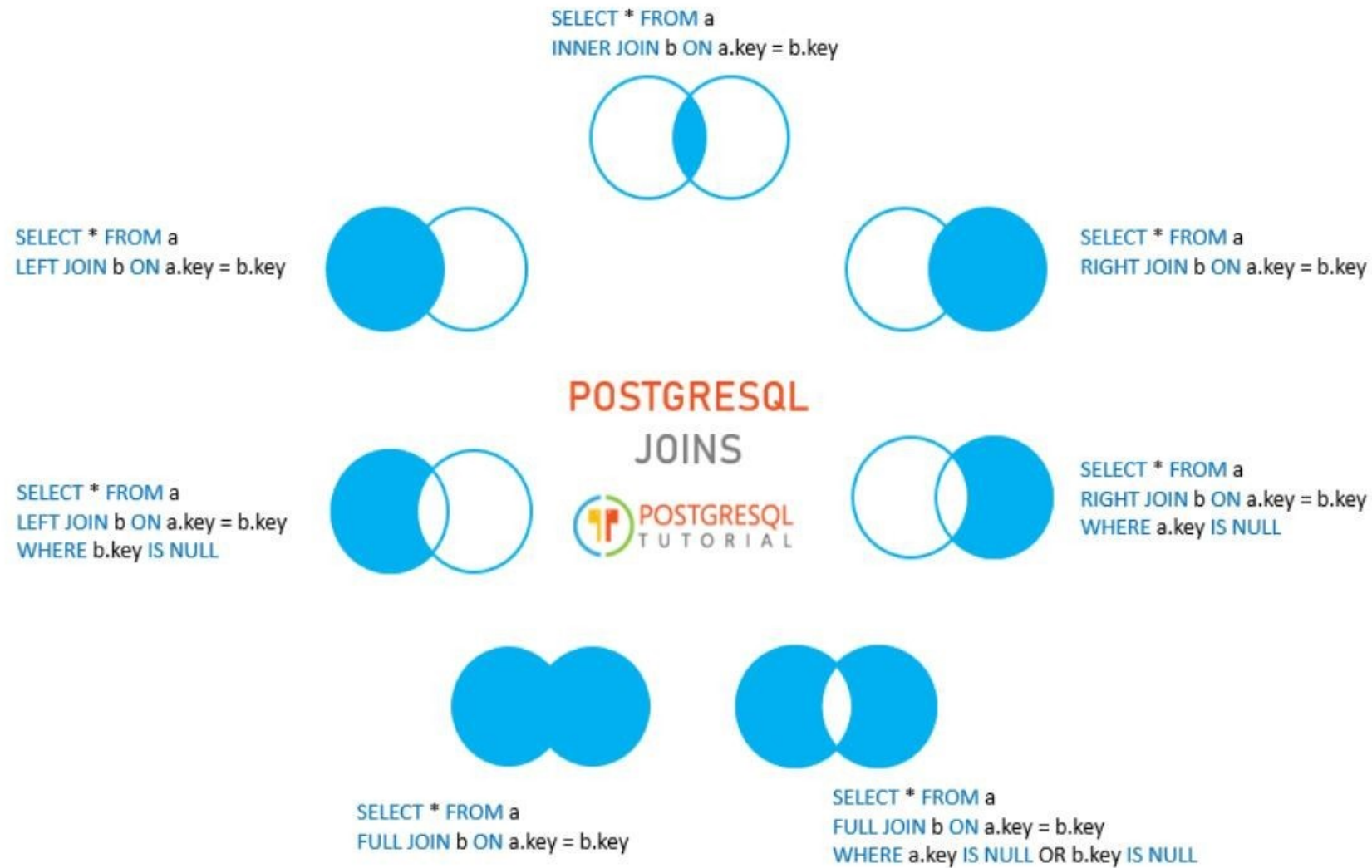


# Unió d'una taula amb ella mateixa

```
SQL> SELECT worker.ename||' works for '||manager.ename  
2   FROM emp worker  
3   JOIN emp manager ON worker.mgr = manager.empno;
```

```
WORKER.ENAME || 'WORKSFOR' || MANAG  
-----  
BLAKE works for KING  
CLARK works for KING  
JONES works for KING  
MARTIN works for BLAKE  
...  
13 rows selected.
```

# Resum



# Activitat A04

## Consulta amb diverses taules

---



- Connectat al moodle i descarrega't la pràctica "A04 Consulta amb diverses taules".
  - Hauràs de fer servir la base de dades HR i PAGILA
  - Entrega en format PDF.
  - Temps 60m
- 
- Si acabes abans continua millorant les teves habilitats de SQL en el següent link: [https://sqlzoo.net/wiki/SELECT\\_basics](https://sqlzoo.net/wiki/SELECT_basics)

# WEBGRAFIA

---

- SQL Tutorial, W3schools, Setembre 2022, <https://www.w3schools.com/sqL/default.asp>
- PostgreSQL Tutorial from scratch, Setembre 2022, <https://www.postgresqltutorial.com/>
- Exercicis Online de SQL, W3schools, Setembre 2022, [https://www.w3schools.com/SQL/sql\\_exercises.asp](https://www.w3schools.com/SQL/sql_exercises.asp)
- PostgreSQL Exercices, Practice,Solution, W3resource,Setembre 2022, <https://www.w3resource.com/postgresql-exercises/>
- PostgreSQL Documentation, PostgreSQL, Setembre 2022, <https://www.postgresql.org/docs/>