

# Sentències SQL Bàsiques VI

## (Manipulació de dades)



**Institut Rafael  
Campalans**

Plaça del Remei, 1  
17160 Anglès

**Cicle:** DAM

**Curs:** 2022/2023

**Mòdul:** 02 Bases de Dades

# Objectius

---



- Descriure cada sentència DML.
- Inserir files noves en una taula.
- Modificar files d' una taula.
- Eliminar files d' una taula.
- Explicar el control de transaccions i la seva importància.

# Llenguatge de manipulació de dades (DML)

---

- Una sentència DML s'executa quan:
  - Afegeix nous registres a una taula.
  - Modifica registres existents.
  - Elimina registres existents.
- Una transacció consisteix en una col·lecció de sentències DML que formen una unitat lògica de treball.

# Afegir una nova fila en una taula

50	DEVELOPMENT	DETROIT
----	-------------	---------


Nou Registro

DEPT

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

“...insertar una nova fila a la taula DEPT...”

DEPT



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

# La sentència INSERT

---

- Afegim noves files a una taula, mitjançant de la sentència **INSERT**.

```
INSERT INTO    table [(column [, column...])]  
VALUES         (value [, value...])  
[RETURNING *|column, column...];
```

# Inserció de noves files

---

- Inserir una nova fila contenint els valors per a cada columna
- Opcionalment, es poden llistar les columnes a la clàusula INSERT

```
SQL> INSERT INTO dept (deptno, dname, loc)
      2  VALUES      (50, 'DEVELOPMENT', 'DETROIT');
1 row created.
```

- Col·locar els valors en l'ordre que les columnes tenen a la taula per defecte
- Escriure els valors de dades de tipus caràcter i data entre cometes simples

# Inserció de valors NULL

- Mètode Implícit: Ometre la columna a la llista

```
SQL> INSERT INTO dept (deptno, dname )  
      2 VALUES          (60, 'MIS');  
1 row created.
```

- Mètode Explícit: Especificar la paraula clau NULL.

```
SQL> INSERT INTO dept  
      2 VALUES          (70, 'FINANCE', NULL);  
1 row created.
```

# Inserció de valors especials

---

- La funció CURRENT\_DATE registra la data i hora actual.

```
SQL> INSERT INTO emp (empno, ename, job,  
2          mgr, hiredate, sal, comm,  
3          deptno)  
4  VALUES      (7196, 'GREEN', 'SALESMAN',  
5              7782, CURRENT_DATE, 2000, NULL,  
6              10);  
  
1 row created.
```

- Per inserir dates farem servir el format 'YYYY-MM-DD'



# Inserció de múltiples files

- Podem afegir diverses files amb una mateixa sentència **INSERT**

```
INSERT INTO    table [(column [, column...])]  
VALUES         (value [, value...]),  
               (value [, value...]),  
...  
[RETURNING *|column, column...];
```

```
INSERT INTO  
    links (url, name)  
VALUES  
    ('https://www.google.com', 'Google'),  
    ('https://www.yahoo.com', 'Yahoo'),  
    ('https://www.bing.com', 'Bing');
```

# Copia de files des d'una altra taula

---

- Escriure la comanda **INSERT** amb una subconsulta (subquery)
- No utilitzar la clàusula **VALUES**

```
SQL> INSERT INTO managers(id, name, salary, hiredate)
      2          SELECT empno, ename, sal, hiredate
      3          FROM    emp
      4          WHERE   job = 'MANAGER';

3 rows created.
```


- Observar que coincideixi el nombre de columnes en la clàusula **INSERT** amb les de la subconsulta

# Canvi de les dades d'una taula

EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		10
7566	JONES	MANAGER		20
...				

“...Modificació  
d'una fila a la  
Taula EMP...”



EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		20
7566	JONES	MANAGER		20
...				

# La sentència UPDATE

---

- Modifiquem registres existents amb la sentència **UPDATE**

```
UPDATE      table
SET          column = value [, column = value]
[WHERE      condition]
[RETURNING *|column, column,...];
```

- Pot afectar més d'una fila alhora

# Actualització de les files d'una taula

- Les files a modificar s' especifiquen per mitjà de la clàusula **WHERE**

```
SQL> UPDATE    emp
      2  SET      deptno = 20
      3  WHERE    empno = 7782;
1 row updated.
```

- Si s' omet la clàusula **WHERE** es modifiquen totes les files de la taula

**IMPORTANT**

```
SQL> UPDATE    employee
      2  SET      deptno = 20;
14 rows updated.
```

# Actualitzar files en base a una altra taula

---

- Utilitzar subconsultes en sentències UPDATE, per modificar registres d' una taula, basats en valors d' una altra taula.

```
SQL> UPDATE employee
      2 SET deptno = (SELECT deptno
      3                  FROM emp
      4                  WHERE empno = 7788)
      5 WHERE job = (SELECT job
      6                  FROM emp
      7                  WHERE empno = 7788);
2 rows updated.
```

# Actualitzar files en base a una altra taula

---

- Podem utilitzar la condició de join per actualitzar dades d'una taula a partir de dades d'una altra taula

```
UPDATE emp e
  SET sal = min_salary * 1.10
FROM   jobs j
WHERE  e.job = j.job and
       e.deptno = 10;
```

2 rows updated.

# Actualització de files: Error de integritat

---

- El departament número 55 no existeix

```
SQL> UPDATE    emp
      2  SET      deptno = 55
      3  WHERE    deptno = 10;
```

```
UPDATE emp
      *
ERROR at line 1:
ORA-02291: integrity constraint (USR.EMP_DEPTNO_FK)
violated - parent key not found
```




# Eliminació d'un fila en una taula

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	DEVELOPMENT	DETROIT
60	MIS	
...		

“...Borrem una fila de la taula DEPT...”

DEPT



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

# La sentència DELETE

---

- Pot eliminar les files existents d' una taula, mitjançant de la sentència **DELETE**

```
DELETE FROM table  
[WHERE      condition]  
[RETURNING *|column, column, ...];
```

# Supressió de files en una taula

---

- Les files a eliminar s'especifiquen per mitjà de la clàusula WHERE.

```
SQL> DELETE FROM dept  
2 WHERE      dname = 'DEVELOPMENT';  
1 row deleted.
```

- Si omet la clàusula WHERE s'esborren totes les files de la taula.

```
SQL> DELETE FROM dept;  
4 rows deleted.
```

# Supressió de files basat en una altra taula (subquery)

- Utilitza subconsultes en sentències DELETE, per eliminar les files d' una taula, basats en valors d' una altra taula.
- 

```
SQL> DELETE FROM      employee
2   WHERE              deptno =
3                       (SELECT  deptno
4                             FROM    dept
5                             WHERE   dname = 'SALES') ;
6 rows deleted.
```

# Supressió de files basat en una altra taula (join)

---

- Podem eliminar registres d' una taula a partir de les condicions de JOIN d' una segona taula.

```
DELETE FROM contacts
USING blacklist
WHERE contacts.phone = blacklist.phone;
```

# Supressió de files error de restricció de integritat

---

- No pots eliminar una fila que conté una clau primaria, utilitzada com a clau externa en una altra taula

```
SQL> DELETE FROM      dept
      2  WHERE          deptno = 10;
```

```
DELETE FROM dept
          *
ERROR at line 1:
02292: integrity constraint (USR.EMP_DEPTNO_FK)
violated - child record found
```

# Transaccions de Bases de Dades

---

Contenen una de les sentències següents:

- Conjunt de sentències **DML** que decideixen un canvi consistent sobre les dades. (Llenguatge de modificació de dades INSERT/UPDATE/DELETE)
- Una sentència **DDL**. (Llenguatge de definició de dades CREATE/DROP)
- Una sentència **DCL** (Llenguatge de control de les dades: GRANT,REVOKE).

# Transaccions de BD Implícites

---

PostgreSQL utilitza l'**AUTOCOMMIT** per a la validació de les sentències executades. Des de psql:

```
\echo :AUTOCOMMIT  
\set    AUTOCOMMIT [on|off]
```

Amb **AUTOCOMMIT OFF** s'iniciarà una transacció implícita quan executem qualsevol sentència DML, DDL, DCL, haurem de tancar-la amb COMMIT o ROLLBACK.

Amb **AUTOCOMMIT ON** s'iniciarà una transacció implícita quan executem qualsevol sentència DML, DDL, DCL, aquesta transacció es tancarà després de l'execució de la sentència amb COMMIT si tot ha funcionat o ROLLBACK si hi ha algun error.



# Transaccions de BD explícites

---

- Per iniciar una nova transacció utilitzarem la comanda:

**BEGIN [WORK|TRANSACTION]**

- Tancarem la transacció amb:

**COMMIT [WORK|TRANSACTION]**  
**ROLLBACK [WORK|TRANSACTION]**

(\*) La paraula clau TRANSACTION és només una alternativa “cosmètica” al WORK. No cal especificar cap de les paraules clau.

# Avantatges de les TRANSACCIONS

---

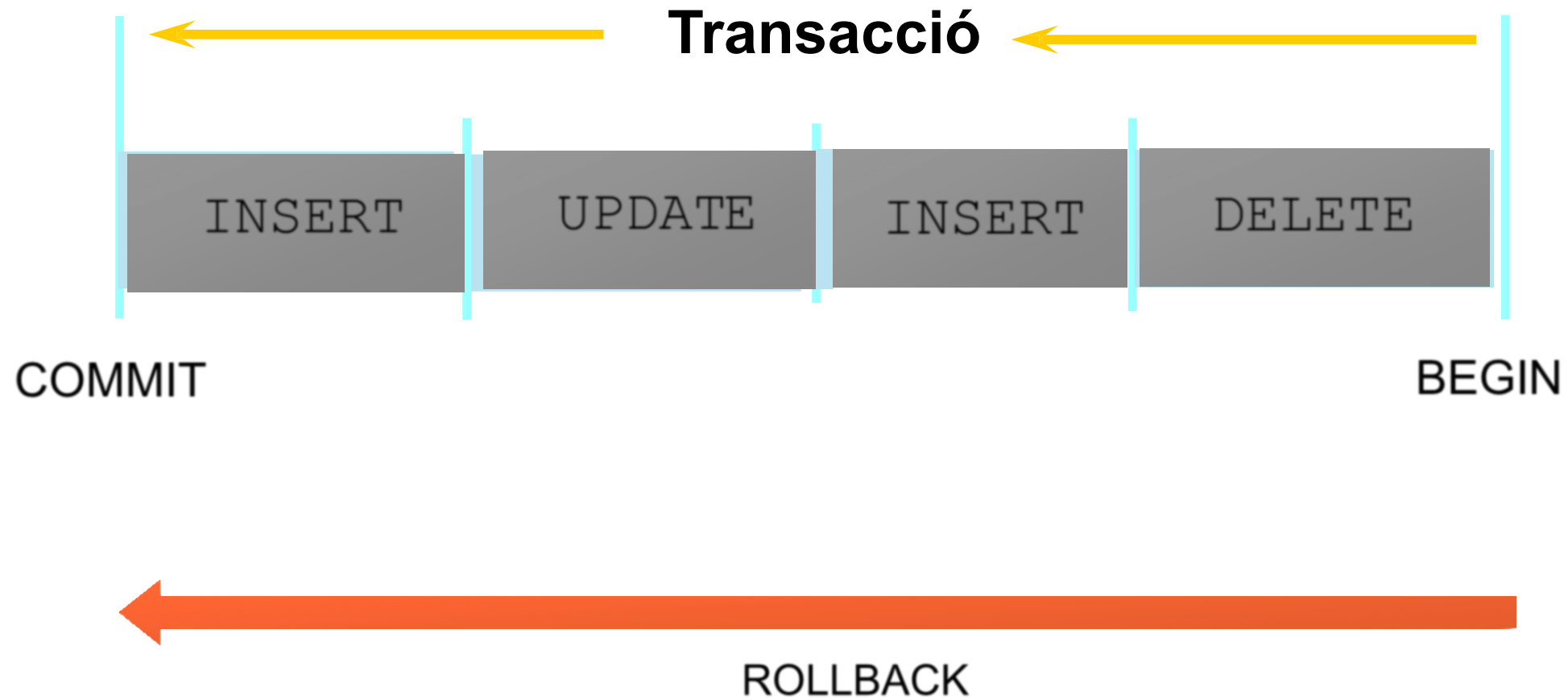
Asseguren la consistència de les dades.

Poden visualitzar els canvis sobre les dades abans de fer-les permanents.

Agrupen lògicament tasques relacionades entre si.

# Control de transaccions

---



# Estat de les dades abans del COMMIT/ROLLBACK

---

- Pot recuperar l'estat anterior de les dades.
- L'usuari actual pot revisar els resultats d'operacions DML usant la sentència SELECT.
- Altres usuaris no poden veure els resultats de les sentències DML executades per l'usuari actual.
- Les files afectades són bloquejades, altres usuaris no poden canviar les dades pertanyents a aquestes files.

# Validació de dades (COMMIT)

---

Realitzar els canvis:

```
SQL> UPDATE   emp
      2  SET      deptno = 10
      3  WHERE    empno = 7782;
1 row updated.
```

```
SQL> COMMIT;
Commit complete.
```

# Estat de les dades després del COMMIT

---

- Els canvis en les dades són escrits a la base de dades.
- L'estat anterior de les dades es perd permanentment.
- Tots els usuaris poden veure els resultats.
- S'alliberen els bloquejos aplicats a les files afectades; aquestes files estan ara disponibles perquè altres usuaris les facin servir.

# Desfer canvis (ROLLBACK)

---

- Realitzar los canvis:

```
SQL> DELETE FROM employee;  
14 row deleted.
```

- Desfer els canvis amb Rollback:

```
SQL> ROLLBACK;  
Rollback complete.
```

# Estat de les dades després de ROLLBACK

---

- Els canvis de dades es desfan.
- Es restaura l' estat anterior de les dades.
- S'alliberen tots els bloquejos sobre les files afectades.

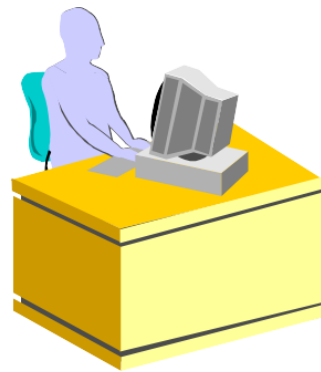


# Consistència en lectura

---

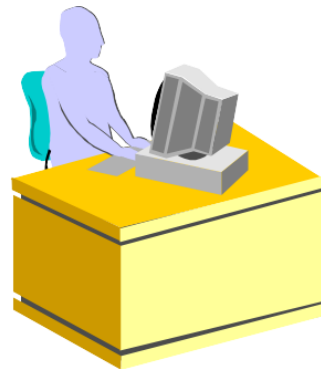
- La consistència en lectura garanteix una vista consistent de les dades en qualsevol circumstància.
- Els canvis realitzats per un usuari no creen conflictes amb els canvis realitzats per un altre usuari.
- Assegura que sobre les mateixes dades les lectures no esperen a les escriptures i viceversa.

# Implementació de la consistència en lectura



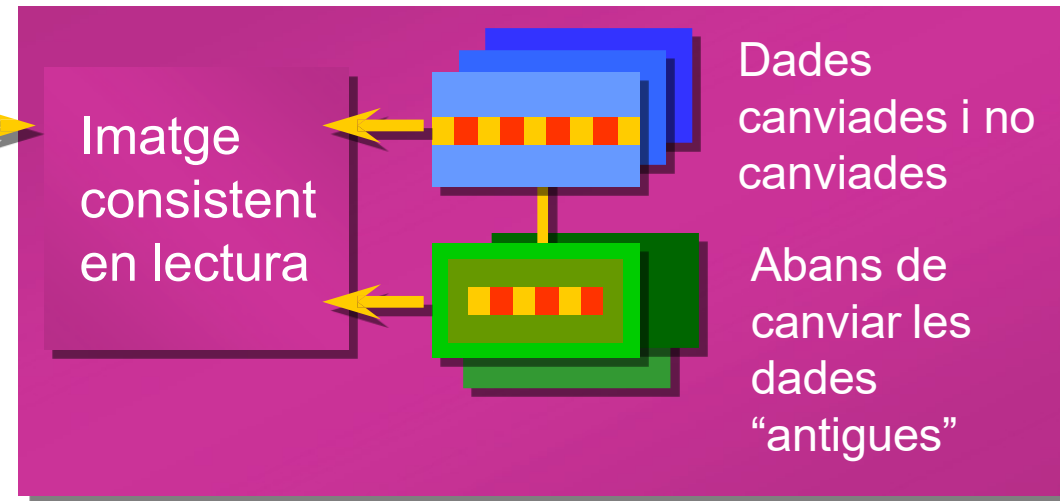
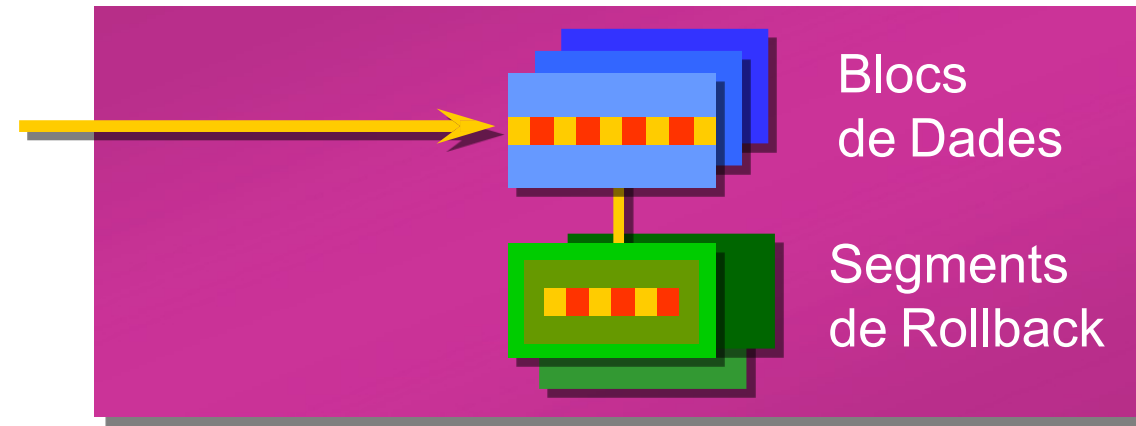
Usuari A

```
update emp  
set sal = 2000  
where ename =  
'SCOTT'
```



Usuari B

```
select *  
from emp
```



# Bloquejos

---

- Prevenen la interacció destructiva entre transaccions concurrents.
- No requereixen accions de l'usuari.
- Automàticament utilitzen el nivell més baix de restricció.
- Es mantenen mentre dura la transacció.
  
- Tenen bàsicament dues maneres:
  - Explícit
  - Implícit

# Bloqueig implícit

---

Dos tipus de bloqueig:

**Exclusiu:** bloqueja altres usuaris

**Compartit:** Accés a altres usuaris

Alt nivell de simultaneïtat de dades:

DML: compartir taula, exclusiu de fila

Consultes: no requereixen bloquejos

DDL: protegeix definicions d' objectes

Bloquejos mantinguts fins a COMMIT o ROLLBACK

# Resum

---

Comanda	Descripció
INSERT	Agrega una fila nova a la taula.
UPDATE	Modifica files d'una taula.
DELETE	Elimina files d'una taula.
BEGIN	Inicia una nova transacció.
COMMIT	Fa permanent tots els canvis pendants
ROLLBACK	Descarta tots els canvis pendants

# 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/>