



# Manipulació de dades

**NF2 - Llenguatge SQL: DML (insert, update, delete)**

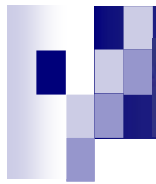
**UF2 – Llenguatge SQL: DML i DDL**

## Que veurem?

1. Introducció
2. Inserció de dades
  - I. Omplir registres a partir de files d'una consulta
3. Actualització de registres
4. Esborrat de registres
5. Transaccions
  - i. Commit
  - ii. Commit explícit
  - iii. Rollback
  - iv. Savepoint
6. Resum

## 1. INTRODUCCIÓ

- ❑ Una de les parts fonamentals de SQL és la manipulació de dades (DML).
- ❑ El DML ho formen les instruccions capaces de consultar i modificar les dades d'una base de dades.
- ❑ Al conjunt d'instruccions DML que s'executen de manera consecutiva se'ls agrupa en **transaccions**.
- ❑ Aquestes transaccions es poden acceptar (COMMIT) o anul·lar (ROLLBACK).
- ❑ Acceptar vol dir que els canvis fets a la BD es consoliden, anul·lar fa que els canvis fets es tirin enrere i les dades quedin en l'estat que estaven abans d'iniciar la transacció.



## 2. INSERCIÓ DE DADES

Afegir dades a una taula es realitza amb la comanda **INSERT**.

```
INSERT INTO nom_taula [( columna1 [ , columna2 ... ])]  
VALUES ( valor1 [, valor2]);
```

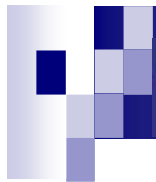
- ***nom\_taula*** indica la taula a la qual volem afegir files i els ***valor1...*** que segueixen a **VALUES** són els valors que donem als diferents camps que conformen una fila de la taula.
- Si no s'especifica la llista de camps, ***columna1..*** , la llista de valors ha de seguir l'ordre de les columnes de la taula segons va ser creada (és l'ordre de columnes quan executem la instrucció *DESC nom\_taula*).
- Els camps no referenciats explícitament a la llista de camps, ***columna1...***, s'emplenen amb el seu valor per defecte (DEFAULT) o bé amb NULL.
- Si algun camp té restricció de tipus NOT NULL, es produirà un error quan s'executi l'INSERT si no emplenen el camp amb algun valor.

## 2. INSERCIÓ DE DADES

**Exemple**, suposem que tenim un taula clients amb els camps: dni, nom, cognom1, cognom2, localitat i adreça; suposem que aquest és l'ordre de creació dels camps d'aquesta taula i que la localitat té com a valor per defecte *Palència* i l'adreça no té valor per defecte.

```
INSERT INTO clients  
VALUES ( '11111111', 'Pere', 'Vera', 'Crepo', DEFAULT, NULL);  
ó  
INSERT INTO clients (dni, nom, cognom1, cognom2)  
VALUES ( '11111111', 'Pere', 'Vera', 'Crepo');
```

En aquest cas les següents instruccions són equivalents equivalents perquè a la 2ona instrucció els camps no indicats s'omplen amb el seu valor per defecte.



## 2. INSERCIÓ DE DADES

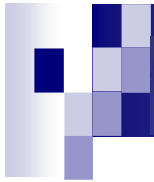
### Omplir registres a partir de files d'una consulta

- Hi ha un tipus de consulta, anomenada d'**addició de dades**, que permet afegir dades a una taula a partir del resultat d'una consulta.
- Aquest farciment es basa en una consulta **SELECT** que recupera les dades que volem afegir a la taula.

```
INSERT INTO nom_taula ( camp1, camp2, ..., campN)
SELECT CampCompatibleCamp1, CampCompatibleCamp2, ..., CampCompatibleCampN
FROM taula (s)
[... Altres clàusules del SELECT...]
```

### Exemple

```
INSERT INTO clients2015 (dni, nom, localitat, adreça)
SELECT dni, nom, localitat, adreça
FROM clients
WHERE problemes=0;
```



### 3. ACTUALITZACIÓ DE REGISTRES

La modificació de dades dels registres s'implementa amb la instrucció **UPDATE**.

```
UPDATE nom_taula  
SET Columna1=valor1 [, columna2=valor2 ... ] [WHERE condicio]
```

Es modifiquen les columnes referenciades en l'apartat **SET** amb els valors indicats.

La clàusula **WHERE** permet especificar quin registres seran modificats.

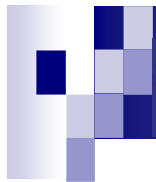
#### Exemples:

```
UPDATE clients SET provincia='Ourense'  
WHERE provincia='Ourense';
```

```
UPDATE productes SET preu=preu 1*16;
```

```
UPDATE partits SET data=NEXT_DAY (SYSDATE, 'Dimarts')  
WHERE data=SYSDATE;
```

```
UPDATE empleats SET lloc_treball=(SELECT lloc_treball FROM empleats WHERE id_empleat=12)  
WHERE seccio=23;
```



## 4. ESBORRAT DE DADES

L'esborrat de dades dels registres s'implementa amb la instrucció **DELETE**.

```
DELETE [FROM] nom_taula  
[WHERE condicio]
```

Elimina els registres de la taula que compleix la condició indicada.

**Exemples:**

```
DELETE FROM clients WHERE provincia='Ourense';
```

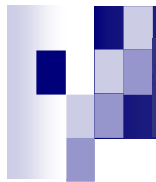
```
DELETE FROM empleats  
WHERE id_empleat IN (SELECT id_empleat FROM errors_greus);
```

L'esborrat de dades no pot provocar que es violin regles d'integritat, si ho fa es produirà un error i no es permetre esborrar les dades.



## 5. TRANSACCIONS

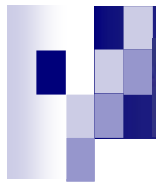
- ❑ Una transacció està formada per una sèrie d'instruccions DML.
- ❑ Una transacció comença amb la primera instrucció DML que s'executa i finalitza amb alguna d'aquestes circumstàncies:
  - Una operació **COMMIT** o **ROLLBACK**
  - Una instrucció DDL (com ALTER TABLE per exemple)
  - Una instrucció DCL (com GRANT)
  - L'usuari abandona la sessió Caiguda del sistema
- ❑ Cal tenir en compte que qualsevol instrucció DDL o DCL dóna lloc a un COMMIT implícit, és a dir totes les instruccions DML executades fins a aquest instant passen a ser definitives.



## 5. TRANSACCIONS

### COMMIT

- La instrucció COMMIT fa que els canvis realitzats per la transacció siguin definitius, irrevocables.
- Només s'ha d'utilitzar si estem d'acord amb els canvis, convé assegurar-se molt abans de realitzar el COMMIT ja que les instruccions executades poden afectar a milers de registres.
- A més el tancament correcte de la sessió dona lloc a un COMMIT, encara que sempre convé executar explícitament aquesta instrucció a fi d'assegurar-nos del que fem.
- Hi ha entorns en que es pot trobar activada l'opció **AUTOCOMMIT**, això fa que qualsevol canvi que fem provoqui un commit automàtic. Això pot passar en eines que ens fem servir per accedir a la BD, com «MySQL Workbench», però mai en els programes que construïm que utilitzen una BD.



## 5. TRANSACCIONS

### COMMIT IMPLÍCIT

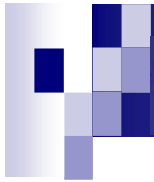
Les següents instruccions SQL executen un COMMIT sense necessitat d'explicitar-ho:

QUIT	DISCONNECT	CREATE VIEW	ALTER
EXIT	CREATE TABLE	DROP VIEW	REVOKE
CONNECT	DROP TABLE	GRANT	AUDIT
			NOAUDIT

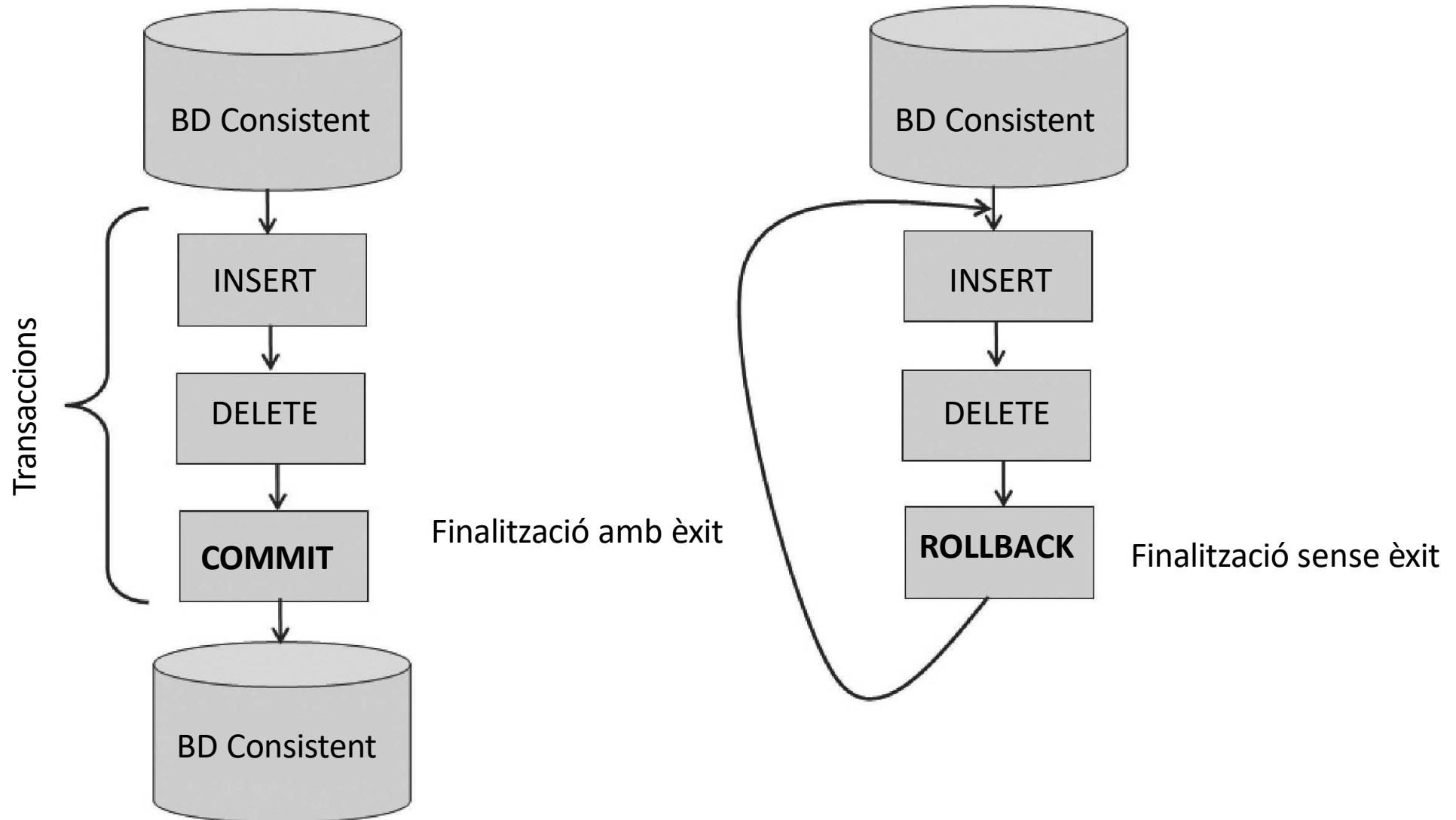
## 5. TRANSACCIONS

### ROLLBACK

- Aquesta instrucció avorta la transacció, retorna a la situació de les taules de la base de dades des de l'últim COMMIT.
- Anul·la definitivament els canvis, per la qual cosa convé també assegurar-se d'aquesta operació.
- Un abandó de sessió incorrecte o un problema de comunicació o de caiguda del sistema donen lloc a un ROLLBACK implícit.



## 5. TRANSACCIONS



## 5. TRANSACCIONS

### SAVEPOINT

Aquesta instrucció permet establir punts de consolidació. El problema de fer ROLLBACK o COMMIT és que un COMMIT consolida tot el bloc de transacció i un ROLLBACK el desfà tot. SAVEPOINT permet assenyalar punts al llarg de la transacció per poder marcar blocs d'actualitzacions. Així podem fer COMMIT o ROLLBACK parcials, en comptes de fer-ho de tot el bloc de transacció.

START TRANSACTION;

SAVEPOINT SavePoint1;

INSERT INTO Product VALUES(1005, 'Product-5', 5000, 500);

INSERT INTO Product VALUES(1006, 'Product-6', 6000, 600);

SAVEPOINT SavePoint2;

INSERT INTO Product VALUES(1007, 'Product-7', 7000, 700);

INSERT INTO Product VALUES(1008, 'Product-8', 8000, 800);

SAVEPOINT SavePoint3;

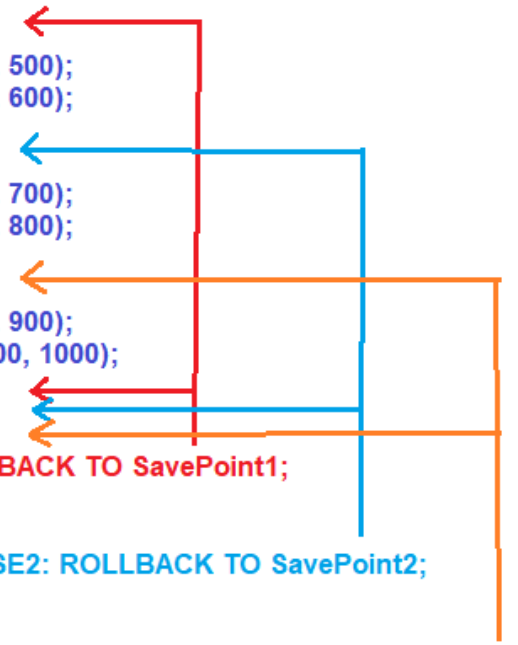
INSERT INTO Product VALUES(1009, 'Product-9', 9000, 900);

INSERT INTO Product VALUES(1010, 'Product-10', 10000, 1000);

CASE1: ROLLBACK TO SavePoint1;

CASE2: ROLLBACK TO SavePoint2;

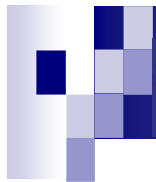
CASE3: ROLLBACK TO SavePoint3;





## 6. RESUM (I)

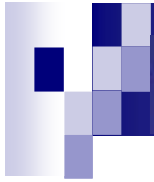
INSERT	Inserció d'una fila	<b>INSERT INTO</b> NomTaula [(columna [, columna] ...)] <b>VALUES</b> (valor [, valor] ...);
	Inserció amb SELECT	<b>INSERT INTO</b> NomTaula1 [(columna [, columna] ...)] <b>SELECT</b> {columna [, columna] ...   *} <b>FROM</b> NomTaula2 [CLÀUSULES DE SELECT];
UPDATE	Modificació files	<b>UPDATE</b> <NomTaula> <b>SET</b> columna1 = valor1, ..., columnan = valorn <b>WHERE</b> condició;
	Modificació de files fent servir SELECT	<b>UPDATE</b> <NomTaula> <b>SET</b> columna1 = valor1, columna2 = valor2, ... <b>WHERE</b> columna3=( <b>SELECT</b> ....); <b>UPDATE</b> <NomTaula> <b>SET</b> (columna1, columna2, ...)=( <b>SELECT</b> col1, col2, ...) <b>WHERE</b> condició; <b>UPDATE</b> <NomTaula> <b>SET</b> columna1 = ( <b>SELECT</b> col1 ... ), columna2 = ( <b>SELECT</b> col2 ...) <b>WHERE</b> condició;



## 6. RESUM (II)

<b>DELETE</b>	Esborrar de files	<b>DELETE [FROM] NomTaula WHERE</b> condició;
<b>TRANSACCIONS</b>	Validar els canvis	<b>COMMIT;</b>
	Desfer canvis	<b>ROLLBACK;</b>





**Preguntes!!!!**