

Unitat 4.1

Llenguatge SQL. DML

DML

Les sentències DML s'utilitzen per modificar (o accedir) a les **dades** (informació dels registres) d'una base de dades.

Exemples:

- Inserció de dades.
- Modificació de dades existents (actualització).
- Eliminació de dades.
- Consulta de dades.

Inserció de dades: INSERT INTO

Per introduir registres a una taula s'utilitza INSERT INTO:

```
INSERT INTO nom_taula [atributs_a_inserir] VALUES atributs_registre;
```

On:

- atributs_a_inserir: llista entre () i separada per comes dels atributs a inserir. Opcional, per defecte s'introdueixen tots els atributs.
- atributs_registre: llista entre () i separada dels valors que es volen introduir a la taula. S'han de posar en el mateix ordre que "atributs_a_inserir" o, en cas d'introduir-se tots, en l'ordre de creació de cada atribut.

INSERT INTO

Exemples:

```
INSERT INTO AUTOR(nom, cognoms, data_naix, nacionalitat) VALUES ('Yuah', 'Noah Harari', DATE'1976-02-24', 'ISR');
```

```
INSERT INTO EDITORIAL VALUES (1, 'LaButxaca');
```

```
INSERT INTO LLIBRE (TITOL, AN, EXEMPLARS, ID_EDITORIAL, ID_SEQUELA) VALUES ('Sàpiens', 2011, 3, 1, NULL);
```

```
INSERT INTO AUTOR_LLIBRE VALUES (1,1);
```

INSERT INTO amb clau autogenerada

Si tenim una taula amb una clau primària autogenerada, s'han d'inserir els nous registres sense indicar l'atribut afectat.

```
CREATE TABLE comanda (  
    id_comanda NUMBER GENERATED AS IDENTITY,  
    DNI_client varchar2(9),  
    producte varchar2(40),  
    preu NUMBER,  
    PRIMARY KEY (id_comanda),  
    FOREIGN KEY (DNI_client) REFERENCES client(DNI)  
);
```



```
INSERT INTO comanda(DNI_client, producte, preu)  
VALUES ('00000000A', 'Samsung G80SD', 900);
```

SELECT

Per recuperar/consultar els registres d'una taula s'utilitza SELECT:

```
SELECT atributs FROM nom_taula;
```

On “atributs” és una llista dels atributs que volem recuperar.

També podem utilitzar “*” per recuperar els registres sencers, amb totes les columnes.

Per veure tots els registres de qualsevol taula:

```
SELECT * FROM nom_taula;
```

SELECT

Exemples:

```
SELECT nom FROM AUTOR;
```

```
SELECT titol, exemplars FROM LLIBRE;
```

```
SELECT NOM, COGNOMS, DATA_NAIX FROM AUTOR;
```

```
SELECT * FROM AUTOR_LLIBRE;
```

Exercici

Agafant el codi SQL de la biblioteca, crea els següents registres:

- 2 editorials.
- 2 llibres, cada un d'una editorial diferent.
- 3 autors.
- Vincula els 2 llibres als autors:
 - Un llibre ha de tenir un autor.
 - L'altre llibre n'ha de tenir dos.

Utilitza SELECT per verificar que les dades hi són.

WHERE

Serveix per indicar condicions a una consulta (filtres):

```
SELECT atributs FROM nom_taula WHERE condicions;
```

Per exemple:

```
SELECT exemplars FROM LLIBRE WHERE titol = 'Sàpiens';
```

```
SELECT nom FROM EDITORIAL WHERE id = 1;
```

```
SELECT nom || ' ' || cognoms FROM AUTOR WHERE nacionalitat = 'ESP';
```

 Operador de concatenació de *strings* (||)

WHERE: condicions

Es poden utilitzar els següents operadors condicionals:

| Operador | Descripció |
|----------|-----------------------------------|
| = | Valors iguals |
| != | Valors diferents |
| > | Valor 1 major que Valor 2 |
| < | Valor 1 menor que Valor 2 |
| >= | Valor 1 major o igual que Valor 2 |
| <= | Valor 1 menor o igual que Valor 2 |

```
SELECT titol, an FROM LLIBRE WHERE exemplars > 2;
```

```
SELECT nom FROM AUTOR WHERE nacionalitat != 'ESP';
```

WHERE: múltiples condicions

Es poden utilitzar els operadors AND i OR per combinar condicions:

- AND: S'han de complir totes les condicions de banda i banda.

```
SELECT nom || ' ' || cognoms FROM AUTOR WHERE nacionalitat = 'ESP' AND nacionalitat = 'GER';
```

- OR: Basta es compleixi alguna de les condicions.

```
SELECT titol, an FROM LLIBRE WHERE exemplars < 1 OR exemplars >= 2;
```

Es poden combinar:

```
SELECT titol, an FROM LLIBRE WHERE exemplars = 1 OR (titol = 'Sàpiens' AND an = 2011);
```

Condicció LIKE

Es pot utilitzar LIKE a un WHERE per filtrar per patrons a columnes alfanumèriques.

Caràcters importants (comodins):

- %: Substitueix tots els caràcters (0 o més)
- _: Substitueix qualsevol caràcter (però només un).

LIKE és *case-sensitive*, però utilitzant UPPER es pot fer que no ho sigui:

```
SELECT * FROM PERSONA WHERE UPPER(nom) LIKE UPPER('%an%');
```

LIKE: Exemples

```
SELECT * FROM PERSONA WHERE nom LIKE 'A%';
```

Retorna totes les persones amb nom que comenci per A. (ANDRÉS, ARNAU, AINA).

```
SELECT * FROM PERSONA WHERE nom LIKE '%E';
```

Retorna totes les persones amb nom que acabi per E. (MAITE, CARME).

```
SELECT * FROM PERSONA WHERE nom LIKE '%AU%';
```

Retorna totes les persones amb un nom que contengui AU a qualsevol posició. (PAU, AURELIO).

LIKE: Exemples

```
SELECT * FROM PERSONA WHERE nom LIKE '____';
```

Retorna totes les persones amb un nom format per 3 lletres. (PAU, PEP)

```
SELECT * FROM PERSONA WHERE nom LIKE 'MARI_';
```

Retorna totes les persones amb un nom de 5 lletres i que comenci per MARI. (MARIO, MARIA).

```
SELECT * FROM PERSONA WHERE nom LIKE '_ERN%';
```

Retorna totes las persones amb un nom que comenci de qualsevol lletra seguida de ERN i qualsevol terminació. (BERNAT, FERNANDO).

INSERT INTO a partir d'una consulta

Es pot inserir a una taula el resultat d'una consulta:

```
INSERT INTO nom_taula [atributs_a_inserir] consultaSELECT;
```

Per exemple:

```
INSERT INTO llibres_escassos (titol, an) SELECT titol, an FROM LLIBRE WHERE exemplars <= 2;
```

Si els tipus de les columnes coincideixen es pot fer de la següent forma:

```
INSERT INTO taula1 SELECT * FROM taula2;
```

UPDATE

Permet actualitzar els valors d'un registre.

```
UPDATE nom_taula SET  
    columna1 = expr1,  
    columna2 = expr2,  
    ...  
[WHERE condicions];
```

El nou valor pot ser un valor concret, una consulta o un càlcul

Per exemple:

```
UPDATE LLIBRE SET exemplars = exemplars * 2;
```

```
UPDATE LLIBRE SET exemplars = 2 WHERE exemplars = 1;
```

```
UPDATE LLIBRE SET  
    titol = (SELECT titol FROM LLIBRE WHERE id = 2),  
    an = (SELECT an FROM LLIBRE WHERE id = 2)  
WHERE id = 1;
```


DELETE

Permet eliminar registres d'una taula segons una condició.

```
DELETE FROM nom_taula [WHERE condicions];
```

Per exemple:

```
DELETE FROM LLIBRE WHERE exemplars = 0;
```

```
DELETE FROM AUTOR WHERE NOM = 'Solaire' OR COGNOMS = 'Astora';
```

```
DELETE FROM AUTOR;
```

Exercici

Altera les dades inserides a l'exercici anterior:

- Fes que el llibre que tenia 2 autors només en tenguí un.
- Canvia un llibre d'editorial. Una editorial ha de quedar sense llibres.
- Elimina l'editorial que ha quedat sense llibres (posant directament la ID).
- Un llibre (qualsevol) s'ha esgotat! Posa els seus exemplars a 0.
- Crea una taula nova LLIBRES_ESGOTATS amb la mateixa estructura que LLIBRE.
- Copia els llibres amb exemplars == 0 a la nova taula.
- Elimina els registres de AUTOR_LLIBRE dels llibres amb exemplars a 0.
- Elimina els llibres sense exemplars (= 0) de la taula LLIBRE.

Literals

Per introduir valors de diferents tipus de dades s'han d'utilitzar els seus **literals**.

Els literals més bàsics són els següents:

- Cadenes de caràcters: 'Exemple', 'Prova', 'Flor d'ametller'
- Numèrics: 0, -10, +77, 4, 25e-03, 6.34

Literals: DATE i TIMESTAMP

Aquests dos tipus es representen de la següent manera:

- DATE: `DATE 'YYYY-MM-DD'`

On YYYY és l'any, MM el més i DD el dia.

- Exemples: DATE '2023-07-19', DATE '1998-12-25'

- TIMESTAMP: `TIMESTAMP 'YYYY-MM-DD HH:MI:SS.FF'`

On HH és l'hora (format 24), MI els minuts, SS els segons i FF la fracció de segon

- Exemple: TIMESTAMP '1997-01-31 09:26:50.124'

- Funcions de conversió a i des de cadenes de caràcters

- TO_CHAR: de DATE i TIMESTAMP a cadena de caràcters.
- TO_DATE i TO_TIMESTAMP: De cadena de caràcters a DATE i TIMESTAMP.

Conversions amb dates. Exemples.

- TO_DATE:

```
TO_DATE('2020-12-12 09:09:09', 'yyyy-mm-dd hh:mi:ss');
```

```
TO_DATE('19-FEB-2004', 'DD-MON-YYYY')
```

```
TO_DATE('19-22-2004')
```

- TO_TIMESTAMP:

```
TO_TIMESTAMP('16-SEP-2022 08:10:14 AM')
```

- TO_CHAR:

```
TO_CHAR(SYSDATE, 'DD-MM-YYYY')
```

```
TO_CHAR(SYSTIMESTAMP)
```

Exercici:

Amb les mateixes dades:

- Afegeix la columna “data_arribada” a LLIBRE, ha de ser un DATE.
- Ha arribat un llibre nou i ja està prestat. Introdueix un llibre qualsevol amb la data d’arribada assignada a la data actual.