

Lucrare laborator 6

Manipularea datelor. Instrucțiunile INSERT, SELECT, UPDATE, DELETE

Funcționalitatea bazelor de date, după cum s-a remarcat din lucrările de laborator anterioare, este dată de posibilitățile de manipulare a datelor prin afișare, ștergere, inserare și modificarea în tabelele componente.

În limbajul SQL, problemele de manipulare a datelor sunt materializate prin patru instrucțiuni și anume:

- instrucțiunea select pentru extragerea informațiilor dorite din tabele
- instrucțiunea insert pentru inserarea de noi înregistrări în tabele
- instrucțiunea update pentru modificarea înregistrărilor existente
- instrucțiunea delete pentru ștergerea datelor nedorite din tabele.

1. Instrucțiunea INSERT

Inserează date într-un tabel.

Formatul general al instrucțiunii este:

```
INSERT [IGNORE] [INTO] tbl_name [(col_name,...)]  
VALUES ((expression),...),(...),...
```

sau

```
INSERT [IGNORE] [INTO] tbl_name [(col_name,...)]  
SELECT ...
```

După cum se poate vedea, instrucțiunea insert are două formate generale. Prin utilizarea primului format, se pot introduce valori specificate de utilizator în tabelul selectat de tbl_name iar cu ajutorul celui de-al doilea format se pot introduce valori extrase din alt tabel prin comanda select în tabelul tbl_name.

În situația introducerii de valori multiple iar valorile încalcă regula de unicitate pentru un câmp de tip cheie primară (valorile trebuie să fie unice), inserarea va fi întreruptă dacă nu se va utiliza opțiunea IGNORE. Deci, dacă în tabelul x vom avea declarat câmpul "camp1" ca fiind cheie primară și pentru câmpul amintit se doresc introduse valorile 1, 2, 3, 1, 4, cea de-a doua valoare 1 va încalca regula unicitate pentru cheia primară. Prin utilizarea opțiunii IGNORE cea de-a doua valoare 1 va fi ignorată la inserare, iar în caz contrar inserarea va fi întreruptă după introducerea valorii 3.

Exemplu:

Se consideră un tabel cu numele vanzari2003 în care se ține evidența vânzărilor unor companii pentru lunile anului 2003. Tabelul va fi compus din câmpurile companie (numele companiei), vanzari (suma încasată) și luna (luna în care s-a făcut încasarea). Pentru acest tabel vom dori inserarea sumelor de 5000000000, 3000000000 și 5000000000 pentru companiile companie 4, companie 5 și respectiv companie 1 pentru luna mai. Pentru aceasta vom folosi comanda

```
INSERT INTO vanzari2003 (companie, vanzari, luna) VALUES
('companie 4', 5000000000, 5),
('companie 5', 3000000000, 5),
('companie 1', 5000000000, 5);
```

Exemplu:

Ca o continuare a exemplului anterior, vom considera necesara construirea unui al doilea tabel cu numele vanzari2002 in care se doreste inserarea valorilor de la campurile companie si luna din tabelul vanzari2003. Pentru aceasta vom folosi a doua forma a instructiunii INSERT in felul urmat:

```
INSERT INTO vanzari2002 (companie, luna) SELECT companie, luna FROM
vanzari2003
```

2. Instructiunea SELECT

Instructiunea SELECT afiseaza unul sau mai multe campuri din unul sau mai multe tabele.

Formatul general al aceste instructiuni este:

```
SELECT select_expression, ... FROM table_references
    [WHERE where_definition]
    [GROUP BY {col_name} , ...]
    [HAVING where_definition]
    [ORDER BY {col_name} [ASC | DESC], ...]
```

Unde:

- *select_expression*, reprezinta coloanele sau expresii de calcul intre coloanele ce se doresc afisate;
- *table_references*, reprezinta tabelul sau tabellele la care se face referire in optiunea select;
- folosirea clauzei *WHERE* da posibilitatea utilizatorului sa selecteze numai rezultatele ce indeplinesc conditia sau conditiile logice din *where_definition*;

Exemplu:

```
SELECT nume, prenume FROM student WHERE an_studiu = 2;
```

Aceasta comanda va avea ca rezultat afisarea campurilor nume si prenume din tabelul student unde anul de studiu este 2.

- folosirea optiunii *GROUP BY* da posibilitatea utilizatorului sa obtina o grupare a rezultatelor dupa campul sau campurile specificate in *col_name*;

Exemplu:

```
SELECT * FROM materii;
```

Comanda de mai sus va afisa toate campurile din tabelul materii. Comanda va avea urmatorul rezultat:

nr_leg	materie	capitol
1	matematica	cap 1

2	matematica	cap 2
3	fizica	cap 1
4	fizica	cap 3
5	matematica	cap 3
6	chimie	cap 1

SELECT materie, capitol FROM materii GROUP BY materie;
 Comanda de mai sus are ca rezultat selectarea campurilor materie se capitol din tabelul materii cu grupare dupa materii. Rezultatul va fi sub urmatoarea forma:

materie	capitol
chimie	cap 1
fizica	cap 1
matematica	cap 1

sau

SELECT materie, capitol FROM materii GROUP BY materie, capitol;
 Comanda de mai sus are ca rezultat selectarea campurilor materie se capitol din tabelul materii cu grupare dupa materii si capitol. Rezultatul va fi sub urmatoarea forma:

materie	capitol
chimie	cap 1
fizica	cap 3
fizica	cap 1
matematica	cap 2
matematica	cap 3
matematica	cap 1

De aceasta data se poate observa o grupare a campurilor in primul rand dupa materii si in al doilea rand dupa capitol ordinea fiind data de reprezentarea in optiunea BGROUP BY.

- folosirea optiunii *HEAVING* da posibilitatea utilizatorului sa obtina o rerafinare a rezultateilor selectate urmarind conditiile logice precizate in where_definition

Exemplu:

Ca un exemplu pentru aceasta optiune vom considera un tabel cu numele vanzari2003, in care sunt introduse date despre mai multe companii cu incasarile pe fiecare luna. Sa consideram ca tabelul are urmatoarele valori pe care le putem afla din comanda

SELECT * FROM vanzari2003
 ce va avea ca rezultat

companie	vanzari	luna
companie 1	12000000	1
companie 2	130000000	1
companie 3	100000	1
companie 1	100000000	2

companie 1	2000000	3
companie 2	300000000	2
companie 3	12000000	2

Dorim sa scoatem din tabel toate companiile care au suma vanzarilor pe toate lunile (introduse in tabel) mai mare de 100000000. Acest lucru se poate face cu o comanda de forma:

```
SELECT Companie, SUM(vanzari) FROM vanzari2003
GROUP BY companie
HAVING (SUM(vanzari) >= 100000000);
```

Rezultatul comenzii va fi urmatorul:

companie	SUM(vanzari)
companie 1	114000000
companie 2	430000000

- folosirea optiunii *ORDER BY* da posibilitatea utilizatorului sa obtina o ordonare a rezultatelor ce se doresc a fi afisate, ordonarea facandu-se dupa coloanele specificate in col_name. Cuvintele cheie ASC si DESC se folosesc pentru ordonare ascendenta respectiv descendenta. Daca nici una din optiuni nu este specificata atunci ASC este luat implicit.

Exemplu:

Considerand enuntul de la exemplul de mai sus, vom incerca ordonarea rezultatelor de la comanda `SELECT * FROM vanzari2003` cu comanda.

```
SELECT * FROM vanzari2003 ORDER BY companie;
```

Rezultatul va fi urmatorul

companie	vanzari	luna
companie 1	12000000	1
companie 1	100000000	2
companie 1	2000000	3
companie 2	130000000	1
companie 2	300000000	2
companie 3	100000	1
companie 3	12000000	2

Un alt exemplu ar fi extinderea ordonarii dupa campul companie in ordine crescatoare si dupa campul luna in ordine descrescatoare prin comanda:

```
SELECT * FROM vanzari2003 ORDER BY companie ASC, luna DESC;
ce va avea ca efect:
```

companie	vanzari	luna
companie 1	2000000	3
companie 1	100000000	2
companie 1	12000000	1
companie 2	300000000	2
companie 2	130000000	1
companie 3	12000000	2
companie 3	100000	1

3. Instructiunea UPDATE

Instructiunea UPDATE modifica inregistrarile existente ale unui tabel. Formatul general al instructiunii este

```
UPDATE [IGNORE] tbl_name
      SET col_name1=expr1 [, col_name2=expr2, ...]
      [WHERE where_definition]
```

unde:

- tbl_name este numele tabelului in care se doreste facuta modificarea
- col_name este numele coloanei ce se doreste modificata
- clauza WHERE este optionala si ajuta la rafinarea dupa logica din where_definition a inregistrarilor ce se doresc a fi modificate
- optiunea IGNORE are acelasi rol ca la instructiunea INSERT fiind necesara ignorarii posibilitatilor de violare a campurilor de tip cheie primara.

Exemplu:

Pentru exemplificarea acestei instructiuni vom considera relua tabelul vanzari2003 in care se va dori incrementarea cu o unitate campului luna. Pentru aceasta vom considera urmatoarea comanda:

```
UPDATE vanzari2003 SET luna = luna + 1;
```

In situatia in care in SET sunt specificate mai multe operatii, acestea vor fi evaluate in ordine de la stanga la dreapta.

Exemplu:

```
UPDATE vanzari2003 SET luna = luna*2, luna = luna + 1;
```

In acest exemplu, in tabelul vanzari2003 campul luna va fi prima data dublat iar apoi incrementat.

4. Instructiunea DELETE

Instructiunea DELETE are ca efect stergerea inregistrarilor dintr-un tabel. Formatul general al instructiunii este:

DELETE FROM table_name [WHERE where_definition]

unde:

- table_name este numele tabelului din care se doreste stergerea
- clauza WHERE este folosita pentru rafinarea inregistrarilor ce se doresc a fi sterse. Se foloseste ca la instructiunea SELECT.

Exemplu:

Pentru a exemplifica efectul acestei comenzi se consideram tabelul materii amintit la instructiunea SELECT in care se va dori stergerea campurilor ce contin materia matematica.

Pentru rezolvarea acestei probleme se va folosi urmatoarea linie de comanda:

```
DELETE FROM materii WHERE materie = 'matematica';
```

In situatia in care se doreste stergerea tuturor inregistrarilor clauza WHERE va dispere din linia de comanda ca in exemplul urmator

```
DELETE FROM materii;
```

Tema de laborator

Sa se construiasca o baza de date pentru gestiunea incasarilor si cheltuielilor distribuite pe luni si ani pentru un grup de firme.

Cheltuielile vor fi exprimate in lei.

1. Se vor introduce date in tabel pentru cel putin 3 firme pe ultimii 2 ani.
2. Sa se afiseze datele introduse
3. Sa se afiseze suma totala a incasarilor si cheltuielilor pe ultimii ultimul an pentru fiecare firma
4. Sa se converteasca cheltuielile in dolari, considerand cursul valutar la 30000 lei.
5. Sa se stearga toate incasarile firmei 1 pentru primul an.