

Probleme suplimentare – Tema Nr. 3

Exercițiile din această temă se referă la modelul *STUDENT*. Fișierul de creare a tabelelor și inserare de date în acestea este *create_insert_student.sql*.

! Înainte de a trece la rezolvarea exercițiilor, pornind de la schemele relaționale ale tabelelor, realizați **diagrama conceptuală a modelului**. Schemele relaționale sunt:

LOCATION (loc_id#, bldg_code, room, capacity)

FACULTY (f_id#, f_last, f_first, f_mi, loc_id, f_phone, f_rank, f_super, f_pin, f_image BLOB)

STUDENT (s_id#, s_last, s_first, s_mi, s_address, s_city, s_state, s_zip, s_phone, s_class, s_dob, s_pin, f_id, time_enrolled)

TERM (term_id#, term_desc, status, start_date)

COURSE (course_no#, course_name, credits)

COURSE_SECTION (c_sec_id#, course_no, term_id, sec_num, f_id, c_sec_day, c_sec_time, c_sec_duration, loc_id, max_enrl)

ENROLLMENT (s_id#, c_sec_id#, grade)

Obs: Tabelul *FACULTY* conține cadrele didactice ale facultății.

1. Să se determine, printr-o singură cerere, codul și numele studenților coordonați de profesorul Brown, precum și codul și numele cursurilor ținute de acesta. Etichetați coloanele „Cod” și „Student sau curs”.

Adăugați o coloană denumită „tip”, care precizează tipul valorii de pe coloana a doua („student” sau „curs”).

2. Determinați studenții (cod, nume) care au urmat un curs de Baze de date, dar nu și unul de Programare în C++.
3. Determinați studenții (cod, nume) care au obținut nota C la cel puțin un examen sau care au cel puțin o notă necunoscută.
4. Afișați locațiile care au capacitate maximă (codul locației, codul clădirii, capacitatea).
5. Executați comenzile următoare:

```
CREATE TABLE t (id NUMBER PRIMARY KEY);
```

```
INSERT INTO t VALUES(1);
```

```
INSERT INTO t VALUES(2);
```

```
INSERT INTO t VALUES(4);
```

```
INSERT INTO t VALUES(6);
```

```
INSERT INTO t VALUES(8);
```

```
INSERT INTO t VALUES(9);
```

Determinați id-ul minim, respectiv maxim, disponibil.

De exemplu, pentru valorile introduse, cererea va întoarce valorile 3 și 7.

6. Să se obțină un rezultat de forma: *cod profesor, nume profesor, student, curs*. Pentru fiecare profesor, coloanele *student* și *curs* vor afișa „Da” dacă există vreun student coordonat, respectiv vreun curs prezentat de acel profesor, și „Nu” altfel. În cazul afirmativ, se va specifica între paranteze numărul de studenți coordonați, respectiv numărul de cursuri ținute.
7. Determinați perechile posibile de semestre a căror descriere (*term_desc*) diferă doar pe ultimul caracter.
8. Determinați studenții care au urmat cel puțin două cursuri al căror cod (*course_no*) diferă pe al cincilea caracter.
9. Determinați perechile de coduri de cursuri care s-au ținut pe același semestru. Perechile se vor considera neordonate (dacă se determină (x,y) , nu se va include în rezultat și (y, x)). Codul mai mare va fi pe prima coloană.
10. Să se determine codul, numele cursului, denumirea semestrului și numărul de locuri (*max_enrl*) pentru cursurile al căror număr de locuri este mai mic decât numărul de locuri corespunzător oricărui curs ținut în locația 1.
11. Determinați cursurile cu număr minim de locuri. Se vor afișa numele cursului și numărul de locuri.
12. Pentru fiecare profesor, să se afișeze numele acestuia și numărul mediu de locuri corespunzătoare cursurilor sale.
13. Determinați profesorii care coordonează cel puțin 3 studenți. Afișați numele profesorului și numărul de studenți coordonați.
14. Determinați, pentru fiecare curs, capacitatea maximă a locațiilor în care s-a desfășurat. Se vor afișa numele cursului, capacitatea maximă și codul locației corespunzătoare.
15. Pentru fiecare semestru din 2007, să se afle valoarea medie a numărului de locuri la cursurile din semestrul respectiv.