

Laborator 1

Recapitulare

Notare

- Examen in sesiune 50%
- Laborator: 50% (Partial 1 25%, Partial 2 25%)
- Bonus laborator: max 1 pct

Indicați valoarea de adevăr a următoarelor afirmații:

- Comenzile SQL*Plus accesează baza de date.
- Funcțiile grup se aplică asupra unei mulțimi de înregistrări și întorc un singur rezultat.
- Funcțiile grup includ în calcule valorile null.

Alegeți afirmația adevărată:

- a. Cheia primară a unei tabele nu poate fi dezactivată ulterior adăugării ei, ci doar eliminată.
- b. O tabelă poate avea declarată o singură constrângere de cheie primară.
- c. Cheia primară a unei tabele nu poate fi compusă din mai multe coloane ale acestuia.
- d. Pentru a putea adăuga o constrângere de cheie primară pe o coloana a unei tabele, coloana respectivă trebuie să conțină valori fără duplicate sau valori null.

Alegeți afirmația incorectă:

- a. Constrângerea de cheie externă implementează o relație de tip one-to-many între două tabele.
- b. Constrângerea de cheie externă se adaugă tabeli “copil” și trebuie să refere o cheie unică sau primară din tabela “părinte”.
- c. Ștergerea unei linii din tabela “părinte” implică întotdeauna ștergerea liniilor corespunzătoare acesteia din tabela “copil”, dacă relația dintre cele două tabele este implementată cu ajutorul unei constrângeri de cheie externă.
- d. Coloana din tabela “copil” pe care este declarată o constrângere de cheie externă poate conține valori null sau valori menținute în coloana referită din tabela “părinte”.

O constrângere de validare

- a. poate fi declarată doar la crearea tabelului.
- b. poate fi declarată doar ulterior creării tabelului.
- c. definește explicit o condiție ce trebuie satisfăcută doar de anumite linii ale tabelului.
- d. definește explicit o condiție ce trebuie satisfăcută de fiecare linie a tabelului.

O vizualizare simplă (extrage date dintr-o singură tabelă, nu conține funcții și grupări de date)

- a. nu reflectă întotdeauna actualizările realizate asupra tabelului de bază.
- b. stochează datele obținute prin execuția cererii din definiția ei.
- c. determină ștergerea unei linii din tabelul de bază, atunci când linia respectivă este ștearsă din vizualizare.
- d. nu permite actualizarea tabelului de bază prin intermediul său

O subcerere care întoarce cel puțin două linii nu poate fi utilizată într-o comandă SELECT în clauza

- a. SELECT
- b. FROM
- c. WHERE
- d. HAVING

Execuția comenzii următoare determină execuția subcererii sale de un număr de ori egal cu

```
SELECT titlu  
FROM carte  
WHERE cod_autor NOT IN (  
    SELECT id_autor  
    FROM autor  
    WHERE nationalitate = 'Romana');
```

- a. 1
- b. 0
- c. numărul de autori de naționalitate Română din tabelul “autor”
- d. numărul de linii din tabelul “carte”

Dacă în tabela “angajat” sunt menținute informații despre angajați, respectiv despre departamentul și jobul pe care lucrează în prezent, iar în tabela “istoric_angajat” informații despre departamentele și joburile pe care au lucrat aceștia în trecut, atunci comanda următoare obține angajații care în prezent lucrează

```
SELECT id_angajat,  
cod_departament, cod_job
```

```
FROM angajat
```

```
INTERSECT
```

```
SELECT cod_angajat,  
cod_departament, cod_job
```

```
FROM istoric_angajat;
```

- a. într-un departament în care au lucrat și în trecut.
- b. pe un job pe care au lucrat și în trecut.
- c. în același departament și pe același job pe care au lucrat și în trecut.
- d. într-un departament și pe un job pe care nu au mai lucrat în trecut

Se dau următoarele trei tabele:

FACTURA(id_factura#, data_facturare)

CONTINE(cod_factura#, cod_produs#, cantitate)

PRODUS(id_produs#, denumire, pret_unitar). Comanda următoare obține

```
SELECT cod_factura,  
SUM(cantitate*pret_unitar)
```

```
FROM contine a, produs b, factura c  
WHERE a.cod_produs = b.id_produs AND  
a.cod_factura = c.id_factura AND  
TO_CHAR(data_facturare, 'yyyy') =  
TO_CHAR(sysdate, 'yyyy')  
GROUP BY cod_factura;
```

- a. valoarea totală a tuturor facturilor emise în anul curent.
- b. valoarea totală a fiecărei facturi emise la o dată egală cu data curentă.
- c. valoarea totală a fiecărei facturi emise în anul curent.
- d. valoarea totală a tuturor facturilor emise la o dată egală cu data sistemului.

Se dă următoarea tabelă:

STUDENT(id_student#, nume, prenume, an_nastere, oras, cod_camin);

Comanda următoare

INSERT INTO student VALUES (100, 'Popescu', 'Andrei', 1980, 'Bucuresti');

- a. determină adăugarea unei linii în tabela “student” cu informațiile date.
- b. determina adăugarea unei linii în tabela “student” cu informațiile date, iar pentru coloana “cod_camin” utilizează valoarea null.
- c. determină adăugarea unei linii în tabela “student” cu informațiile date, iar pentru coloana “cod_camin” utilizează valoarea null doar dacă această coloană nu are definită o valoare implicită.
- d. nu are efect deoarece se termină cu o eroare.

Folosind tabelul departments generați automat script-ul SQL de inserare a înregistrărilor în acest tabel. (2 variante)

- Comanda select
- Comanda spool

1. Sa se afiseze pentru fiecare department nr. de persoane angajate in fiecare an si procentul din nr. total de angajti.
2. Sa se afiseze pentru fiecare tara si oras numarul de departamente care au mai mult de 5 angajati.
3. Sa se afiseze pentru fiecare angajat salariul si salariul majorat in functie de vechime astfel:
vechime ≤ 1 an \rightarrow 5%, vechime ≤ 5 ani \rightarrow 10 %, altfel 15%.
4. Sa se afiseze orasele in care nu s-a angajat nicio persoana in ultimul an.

5. Sa se afiseze denumirile joburilor si o coloana denumita statistica ce va contine:

- suma salariilor, pentru job-urile care incep cu litera S;
- media generala a salariilor, pentru job-ul avand salariul maxim;
- salariul minim, pentru fiecare din celelalte job-uri.
- salariul minim posibil pentru joburile care nu sunt inca ocupate de un angajat.

Includeti in coloana un string care sa precizeze semnificatia (ex 'media: 5000')

6. Sa se afiseze numele angajatilor stabilindu-se pentru fiecare angajat daca a fost „promovat”/„penalizat”/„a pastrat nivelul”. Se considera ca un angajat a fost promovat daca media dintre min_salary si max_salary pentru ultimul job detinut anterior este mai mica decat media dintre min_salary si max_salary pentru jobul detinut in prezent. Se vor afisa si angajatii care nu au schimbat jobul.