LABORATOR SQL RECAPITULARE – 1

1. Indicaţi valoarea de adevăr a următoarelor afirmaţii:

1) Comenzile SQL*Plus accesează baza de date.
F
2) Funcţiile grup se aplică asupra unei mulţimi de înregistrări şi întorc un singur rezultat.
A
3) Funcţiile grup includ în calcule valorile null.
F

2. Alegeţi afirmaţia adevărată:
a. Cheia primară a unui tabel nu poate fi dezactivată ulterior adăugării ei, ci doar eliminată.
b. Un tabel poate avea declarată o singură constrângere de cheie primară.
c. Cheia primară a unui tabel nu poate fi compusă din mai multe coloane ale acestuia.
d. Pentru a putea adăuga o constrângere de cheie primară pe o coloana a unui tabel, coloana respectivă trebuie să conţină valori fără duplicate sau valori null.

B

3. Alegeţi afirmaţia incorectă:
a. Constrângerea de cheie externă implementează o relaţie de tip one-to-many între două tabele.
b. Constrângerea de cheie externă se adaugă tabelului "copil" şi trebuie să refere o cheie unică sau primară din tabelul "părinte".
c. Ştergerea unei linii din tabelul "părinte" implică întotdeauna ştergerea liniilor corespunzătoare acesteia din tabelul "copil", dacă relaţia dintre cele două tabele este implementată cu ajutorul unei constrângeri de cheie externă.
d. Coloana din tabelul "copil" pe care este declarată o constrângere de cheie externă poate conţine valori null sau valori menţinute în coloana referită din tabelul "părinte".

C

4. O constrângere de validare
a. poate fi declarată doar la crearea tabelului.
b. poate fi declarată doar ulterior creării tabelului.
c. defineşte explicit o condiţie ce trebuie satisfăcută doar de anumite linii ale tabelului.
d. defineşte explicit o condiţie ce trebuie satisfăcută de fiecare linie a tabelului.

D

5. O vizualizare simplă (extrage date dintr-un singur tabel, nu conţine funcţii şi grupări de date)
a. nu reflectă întotdeauna actualizările realizate asupra tabelului de bază.
b. stochează datele obţinute prin execuţia cererii din definiţia ei.
c. determină ştergerea unei linii din tabelul de bază, atunci când linia respectivă este ştearsă din vizualizare.
d. nu permite actualizarea tabelului de bază prin intermediul său.

C

6. O subcerere care întoarce cel puţin două linii nu poate fi utilizată într-o comandă SELECT în clauza
a. SELECT
b. FROM
c. WHERE
d. HAVING

A

7. Execuţia comenzii următoare

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

determină execuţia subcererii sale de un număr de ori egal cu

a. 1
b. 0
c. numărul de autori de naţionalitate Română din tabelul "autor"
d. numărul de linii din tabelul "carte"

A

8. Dacă în tabelul "angajat" sunt menţinute informaţii despre angajați, respectiv despre departamentul şi jobul pe care lucrează în prezent, iar în tabelul "istoric_angajat" informaţii despre departamentele şi joburile pe care au lucrat aceștia în trecut, atunci comanda următoare

```
SELECT id_angajat, cod_departament, cod_job
FROM    angajat
INTERSECT
SELECT cod_angajat, cod_departament, cod_job
FROM    istoric_angajat;
```

obţine angajaţii care în prezent lucrează

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

C

9. Se dau următoarele trei tabele:

```
FACTURA(id_factura#, data_facturare)
CONTINE(cod_factura#, cod_produs#, cantitate)
PRODUS(id_produs#, denumire, pret_unitar)
```

Comanda următoare

```
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;
```

obţine

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.

C

10. Se dă următorul tabel:
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 tabelul "student" cu informaţiile date.
b. determina adăugarea unei linii în tabelul "student" cu informaţiile date, iar pentru coloana "cod_camin" utilizează valoarea null.
c. determină adăugarea unei linii în tabelul "student" cu informaţiile 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.
D

11. Adăugaţi un comentariu tabelului *emp_****.

```
COMMENT ON TABLE emp_*** IS 'Informaţii despre angajati';
```

12. Folosind vizualizarea *user_tab_comments* afişaţi comentariul adăugat tabelului *emp_****.

```
SELECT *
FROM user_tab_comments
WHERE table_name = upper(emp_***);
```

13. Modificaţi formatul datei calendaristice setat la nivel de sesiune astfel încât datele calendaristice să respecte următoarea formă 01.10.2011 16:10:05.

Indicaţie: Folosiţi comanda

```
ALTER SESSION SET NLS_DATE_FORMAT = 'formatul dorit'
```

```
ALTER SESSION
SET NLS_DATE_FORMAT = 'DD.MM.YYYY HH24:MI:SS';
```

14. Rulaţi următoarea cerere SQL:

```
SELECT EXTRACT(YEAR FROM SYSDATE)
FROM   dual;
```

15. Modificaţi cererea anterioară astfel încât să obţineţi ziua, respectiv luna datei curente.

16. Afişaţi numele tuturor tabelelor personale create (nume_tabel_***).

Indicaţie: Folosiţi vizualizarea *user_tables*.

```
SELECT table_name
FROM   user_tables
WHERE  table_name LIKE UPPER('%_test');
```

17. Generaţi automat un script SQL care să conţină comenzi de ştergere a tuturor tabelelor personale create.

Indicaţie: Folosiţi comenzile SPOOL …/sterg_tabele.sql şi SPOOL OFF.

```
spool c:/test.sql
```

```
SELECT 'DROP TABLE ' || table_name || ';'
FROM   user_tables
```

```
WHERE  table_name LIKE upper('%_test');

spool off
```

18. Verificaţi informaţiile din fişierul generat.

19. Ce informaţii suplimentare sunt incluse în acest fişier dacă folosim SQL*Plus?

```
SQL> SELECT 'DROP TABLE ' || table_name || ';'
  2  FROM    user_tables
  3  WHERE   table_name LIKE upper('%_test');

'DROPTABLE'||TABLE_NAME||';'
----------------------------------------
DROP TABLE CARTE_TEST;
DROP TABLE CITITOR_TEST;
DROP TABLE DEPARTAMENT_TEST;
DROP TABLE DEPARTMENT_TEST;
DROP TABLE DEPT_TEST;
DROP TABLE DOMENIU_TEST;
DROP TABLE ECO_TEST;
DROP TABLE EMP0_TEST;
DROP TABLE EMP1_TEST;
DROP TABLE EMP2_TEST;
DROP TABLE EMP3_TEST;

'DROPTABLE'||TABLE_NAME||';'
----------------------------------------
DROP TABLE EMP_TEST;
DROP TABLE IMPRUMUTA_TEST;
DROP TABLE PROJECTS_TEST;
DROP TABLE SALARIAT_TEST;
DROP TABLE WORK_TEST;

16 rows selected.

SQL> spool off
```

20. Verificaţi ce efect are utilizarea comenzii SET FEEDBACK OFF.

```
Set feedback off
spool c:/test.sql

SELECT 'DROP TABLE ' || table_name || ';'
FROM   user_tables
WHERE  table_name LIKE upper('%_test');

spool off
set feedback on
```

21. Asiguraţi-vă că antetul tabelului rezultat nu se multiplică.

Indicaţie: Utilizaţi comanda SET PAGESIZE 0

```
set pagesize 0
set feedback off
spool c:/test.sql
```

```
SELECT 'DROP TABLE ' || table_name || ';'
FROM   user_tables
WHERE  table_name LIKE upper('%_test');

spool off
set feedback on
set pagesize 10
```

22. Fără să rulaţi scriptul creat daţi exemplu de un caz în care execuţia acestui script va determina erori. Indicaţi o metodă de rezolvare a acestui caz.

23. Folosind tabelul *departments* generaţi automat script-ul SQL de inserare a înregistrărilor în acest tabel.

```
SELECT
 'INSERT INTO departments VALUES
  (' || department_id || ', ''' || department_name ||
   ''', ' || location_id || ');'
  AS "Inserare date"
FROM   departments;
```