



## TEST-GRILĂ

**1**

Care dintre următoarele opțiuni poate fi utilizată la stergerea unei tabele?

- a) PRIOR
- b) CASCADE CONSTRAINTS
- c) CONNECT BY
- d) DELETE

**2**

Fie două relații R1 și R2. Operația definită pe cele două relații, R1 și R2, care constă din construirea unei noi relații R3, prin concatenarea unor tupluri din R1 cu tupluri din R2, pe baza unei condiții specificate explicit în cadrul operației, poartă denumirea de:

- a) reuniune
- b) intersecție
- c) joncțiune
- d) produs cartezian

**3**

Fie secvența de cod C următoare:

```
#include <stdio.h>
int f1(int *a)
{
    int n=sizeof (a)/sizeof (*a),s=0;
    for(int i=0;i<n;i++){s+=a[i];}
    return s;
}
int f2(int a[])
{
    int n=sizeof (a)/sizeof (*a),s=0;
    for(int i=0;i<n;i++){s+=a[i];}
    return s;
}
int f3(int a[],int n)
{
    int s=0;
    for(int i=0;i<n;i++){s+=a[i];}
    return s;
}

int main()
{
    int a[3] = { 8, 4, 5 };
    int n = sizeof (a)/sizeof (*a);
    ...
    return 0;
}
```

Care dintre următoarele apeluri ale funcțiilor f1, f2 și f3 vor întoarce corect suma elementelor vectorului a?

- a) f1(a) și f2(a)
- b) f3(a,n)
- c) f1(a), f2(a) și f3(a,n)
- d) f1(a)

**4**

Fie tabela ANGAJATI (id\_angajat NUMBER(5) PRIMARY KEY, nume VARCHAR2(30), prenume VARCHAR2(30), data\_angajare DATE, salariul NUMBER(10,2)) cu cel puțin 100 de rânduri (înregistrări) și blocul PL/SQL:



```
SET SERVEROUTPUT ON
DECLARE
CURSOR cursor1 IS SELECT nume, salariul FROM angajati;
vnume angajati.nume%TYPE;
vsalariul angajati.salariul%TYPE;
BEGIN
OPEN cursor1;
WHILE cursor1%NOTFOUND LOOP
FETCH cursor1 INTO vnume, vsalariul;
DBMS_OUTPUT.PUT_LINE('Angajatul '||vnume|| 'are salariul '||vsalariul);
END LOOP;
END;
/
```

Care afirmatie este corecta?

- a) blocul va afisa numele tuturor angajatilor din tabela ANGAJATI
- b) blocul se executa, dar nu afiseaza nimic
- c) blocul va afisa numele primului angajat din tabela
- d) blocul contine o eroare deoarece nu este corect utilizata conditia cursor1%NOTFOUND

**5**

Fie tabela ANGAJATI (id\_angajat NUMBER(5) PRIMARY KEY, nume VARCHAR2(30), prenume VARCHAR2(30), salariul NUMBER(10,2), comision NUMBER(4,2)).

Se considera blocul PL/SQL:

```
SET SERVEROUTPUT ON
DECLARE
v_venit NUMBER;
BEGIN
SELECT SUM(salariul*(1+NVL(comision,0))) INTO v_venit FROM ANGAJATI WHERE
id_departament=50;
DBMS_OUTPUT.PUT_LINE(v_venit);
END;
/
```

Care afirmatie este corecta?

- a) blocul este eronat deoarece lipseste clauza GROUP BY
- b) blocul se va rula cu succes si se va afisa venitul total din departamentul 50 in cazul in care acesta exista si zero in cazul in care departamentul 50 nu exista
- c) in cazul in care departamentul 50 nu exista va apare un mesaj de eroare
- d) blocul se va rula cu succes si se va afisa venitul total din departamentul 50 in cazul in care acesta exista si nu se va afisa nimic in cazul in care departamentul 50 nu exista

**6**

Fie functia C definita in felul urmatoar:

```
int f(int a,int b)
{
  if (a<b)
  {
    return f(a+3,b-1);
  }
  else
  {
    return a + b;
  }
}
```

Ce rezultat va intoarce apelul f(1000,2000)?

- a) 1500
- b) 3000
- c) 3500
- d) Va genera eroare de executie si va afisa un mesaj de depasire stiva



**7**

Fie tabela ANGAJATI (id\_angajat NUMBER(5) PRIMARY KEY, nume VARCHAR2(30), prenume VARCHAR2(30), data\_angajare DATE, salariul NUMBER(10,2)) cu cel puțin 100 de randuri (inregistrari) și următoarea secvență de comenzi:

```
CREATE OR REPLACE FUNCTION f_emp_vechime
(p_data_angajare IN angajati.data_angajare%type)
return NUMBER
IS
BEGIN
RETURN to_number(round((sysdate-p_data_angajare)/365,0));
END f_emp_vechime;
/
SELECT id_angajat, nume, prenume, f_emp_vechime(data_angajare)
FROM angajati
WHERE f_emp_vechime(data_angajare)>10;
```

Care afirmație este corectă?

- a) instrucțiunea SELECT afișează angajații cu vechimea mai mare de 10 ani
- b) instrucțiunea SELECT afișează angajații cu vechimea mai mare de 10 zile
- c) funcția f\_emp\_vechime returnează numărul de zile dintre data curentă și data primită ca parametru
- d) se va genera o eroare deoarece funcția f\_emp\_vechime nu poate fi utilizată în comanda SELECT

**8**

Ce se va întâmpla la rularea următorului program C?

```
#include <stdio.h>
void f(int i);
int main()
{
    f(1);
}
void f(int i)
{
    if (i > 10)
        return ;
    printf("%d ", i);
    return f((i+=3, --i));
}
```

- a) Se va afișa: 1 4 7 10
- b) Programul va rula fără să afișeze nimic
- c) Va fi generată o eroare de depășire stivă
- d) Se va afișa: 1 3 5 7 9

**9**

Să se precizeze care dintre interogările SQL-Oracle de mai jos afișează următoarea zi de duminică față de data curentă?

- a) SELECT sysdate('SUNDAY') from dual;
- b) SELECT next\_day('SUNDAY') from sysdate;
- c) SELECT last\_day(sysdate, 'SUNDAY') from dual;
- d) SELECT next\_day(sysdate, 'SUNDAY') from dual;

**10**

Ce va afișa următorul program C?

```
#include "stdio.h"
int main()
{
    int i = 3;
    goto LOOP;
    i++;
}
```



```
for (i = 0; i < 10; i+=2)
{
    printf("%d", i);
LOOP:
    continue;
    printf("%d", i++);
}
return 0;
}
```

- a) Nu va afisa nimic
- b) 356789
- c) 579
- d) 3579

**11**

Care va fi valoarea afisata de urmatorul program C?

```
#include <stdio.h>
int main()
{
    static int s=1;
    for(int i=0;i<5;i++)
    {
        static int s=0;
        s+=i;
    }
    printf(" %d", s);
    return 0;
}
```

- a) O valoare intreaga nedefinita
- b) 10
- c) 11
- d) 1

**12**

Fie a, un masiv bidimensional de tip double cu n linii si n coloane, n intreg par.

Care instructiune dintre cele specificate in raspunsurile de mai jos poate inlocui punctele de suspensie (...) din secventa de program C urmatoare, astfel incat executarea acestuia sa permita memorarea in variabila s a valorii sumei elementelor aflate pe diagonalele matricei?

```
int s=0;
for(int i=0;i<n;i++){
    ...
}
```

- a) s+=a[n-i-1][i]+a[i][n-i-1];
- b) s+=a[i][i]+a[i][n-i-1];
- c) s+=a[i][i]+a[n-i-1][n-i-1];
- d) s+=a[i][n-i-1]+a[n-i-1][i];

**13**

Care dintre urmatoarele afirmatii despre cursorii expliciti in PL/SQL este adevarata?

- a) la deschidere se utilizeaza intotdeauna clauza NOTOPEN
- b) stocheaza informatii cu privire la procesarea instructiunilor LCT (limbajul de control al tranzactiilor)
- c) pot fi declarati ca variabile si constante
- d) pot avea parametri



**14**

Care va fi outputul generat la executia urmatorului program C?

```
#include<stdio.h>
int main()
{
    int a[5] = { 5 };
    for (int i = 0; i < 5; i++){
        if ((char)a[i] != '5'){
            printf("FAIL ");
        } else {
            printf("%d ",a[i]);
        }
    }
    return 0;
}
```

- a) FAIL FAIL FAIL FAIL 5
- b) 5 FAIL FAIL FAIL FAIL
- c) 5 5 5 5
- d) FAIL FAIL FAIL FAIL FAIL

**15**

Ce se va afisa in urma executarii urmatorului program C?

```
#include <stdio.h>
#include <string.h>
void f(char str1[], char str2[])
{
    if (*str1++ = *str2++) { f(str1, str2); }
}
int main()
{
    int k='a'-'A';
    char a[100];
    f(a,"Examen Master 2022");
    for(int i=0;i<(int)strlen(a);i++)
        a[i]=a[i]>='a'&& a[i]<='z'?a[i]-k:a[i];
    printf("%s",a);
    return 0;
}
```

- a) EXAMEN MASTER 2022
- b) EXAMENMASTER
- c) XAMEN ASTER 2022
- d) Va genera o eroare deoarece functia f utilizeaza incorect doua masive de caractere care nu sunt lvalues

**16**

Se considera tabela PRODUSE (id\_produș NUMBER(6), denumire\_produș VARCHAR2(150), descriere VARCHAR2(150), categorie VARCHAR2(50), pret\_lista NUMBER(4)) si interogarea:

```
SELECT denumire_produș, descriere
FROM produse WHERE categorie IN
(SELECT categorie FROM produse
WHERE LOWER(denumire_produș) LIKE '%laptop%')
AND pret_lista >100;
Alegeti varianta corecta:
```

- a) afiseaza produsele care au pret\_lista mai mare de 100 si se afla in aceleasi categorii cu produsele care contin in denumire cuvantul „laptop”
- b) afiseaza doar produsele care contin in denumire cuvantul „laptop”
- c) afiseaza o eroare deoarece subcererea returneaza mai multe valori
- d) afiseaza o eroare deoarece operatorul LIKE nu este corect utilizat



**17**

Ce va afisa urmatorul program C?

```
#include<stdio.h>
int main()
{
    printf("%d", 3<<2 );
    return 0;
}
```

- a) 9
- b) 2
- c) 12
- d) 6

**18**

Fie tabela ANGAJATI (id\_angajat NUMBER(5) PRIMARY KEY, nume VARCHAR2(30), prenume VARCHAR2(30), salariul NUMBER(10,2), id\_departament NUMBER(3)) cu cel puțin 200 angajati care lucreaza in peste 10 departamente si blocul PL/SQL:

```
SET SERVEROUTPUT ON
DECLARE
CURSOR cursor2 IS SELECT id_departament, COUNT(salariul) nr_angajati, AVG (salariul) sal_mediu
FROM Angajati GROUP BY id_departament ORDER BY 2 DESC;
r2 cursor2%rowtype;
BEGIN
FOR r2 IN cursor2 LOOP
DBMS_OUTPUT.PUT_LINE('In departamentul ||r2.id_departament|| ' lucreaza ||r2.nr_angajati|| ' cu salariul
mediu ||r2.sal_mediu);
EXIT WHEN cursor2%ROWCOUNT>3;
END LOOP;
END;
/
```

Care afirmatie este corecta?

- a) blocul contine o eroare deoarece nu este corect utilizata conditia cursor2%ROWCOUNT
- b) blocul va afisa primele trei departamente cu cei mai multi angajati
- c) blocul va afisa toate departamentele
- d) blocul va afisa primele patru departamente cu cei mai multi angajati

**19**

Fie tabela RAND\_COMENZI (id\_comanda NUMBER(6), id\_produs NUMBER(8), cantitate NUMBER(7), pret NUMBER(7,2)) avand 100 de randuri precum si interogarea SQL Oracle:

```
SELECT id_comanda, ROUND(AVG(cantitate*pret)) VAL_MEDIE
FROM rand_comenzi
GROUP BY id_comanda
HAVING COUNT(id_produs) >=3;
```

Care din urmatoarele afirmatii este adevarata?

- a) interogarea contine o eroare deoarece clauza HAVING este incorect utilizata
- b) se afiseaza valoarea medie a produselor daca acestea apar pe cel puțin trei comenzi
- c) se afiseaza valoarea medie a comenzilor daca acestea contin cel puțin trei produse
- d) interogarea contine o eroare deoarece functia ROUND este incorect utilizata

**20**

Care va fi outputul generat de urmatorul program C?

```
#include <stdio.h>
int f()
{
    static int v = 10;
    return v--;
```



}

```
int main()
{
    for(f(); f(); f())
        printf("%d ", f());
    return 0;
}
```

- a) Va rula la infinit si va genera o eroare de executie pentru depasire stiva
- b) 8 5 2
- c) Va genera eroare de compilare deoarece bucla for este incorecta sintactic
- d) 8 6 4 2

### 21

In modelul relational pentru baze de date:

- a) cheia primara nu trebuie sa fie unica
- b) proiectia, negatia si produsul cartezian sunt operatori din algebra relationala
- c) sunt utilizati operatori din algebra relationala
- d) selectia, conjunctia si jonctiunea sunt operatori din calculul relational

### 22

Care va fi numarul de afisari ale cuvântului Examen la rularea programului C de mai jos?

```
#include <stdio.h>
int main()
{
    unsigned int i = 10;
    for (; i; i >>= 1)
        printf("Examen\n");
    return 0;
}
```

- a) 0
- b) 4
- c) 10
- d) 3

### 23

Fie tabela ANGAJATI (id\_angajat NUMBER(5) PRIMARY KEY, nume VARCHAR2(30), prenume VARCHAR2(30), salariul NUMBER(10,2), id\_departament NUMBER) avand 200 randuri (inregistrari).

Precizati ce rezultat va furniza interogarea urmatoare:

```
SELECT * FROM angajati a WHERE salariul > (SELECT MAX(salariul) FROM angajati WHERE
id_departament=a.id_departament)
ORDER BY salariul;
```

- a) afiseaza angajatii care au salariul cel mai mare in departamentul in care lucreaza
- b) se executa, dar nu afiseaza nimic
- c) afiseaza toti angajatii indiferent de salariu deoarece conditia din subcerere este adevarata intotdeauna
- d) va apare o eroare deoarece lipseste clauza GROUP BY in subcerere

### 24

Ce va afisa urmatorul program C?

```
#include<stdio.h>
int main()
{
    char p[] = "ropot";
    char t;
    int l = sizeof (p)/sizeof (p[0])-1;
    for (int i= 0, j = l-1; i < j; i++)
    {
```



```
t = p[i];
p[i] = p[j];
p[j] = t;
}
printf("%s", p);
return 0;
}
```

- a) ropot
- b) Nu va afisa nimic
- c) tropo
- d) topor

25

Care din urmatoarele functii SQL nu poate fi folosita in instructiuni de atribuire specifice PL/SQL?

- a) AVG
- b) TO\_DATE
- c) INSTR
- d) NVL

26

Ce va afisa urmatorul program C?

```
#include <stdio.h>
int g(int n,int v) {
    if (n) {
        return g(n / 10,v) * 10 + n % 10;
    }
    else {
        return v==n;
    }
}
int main()
{
    int n = 6779;
    printf("%d",g(n,n));
}
```

- a) 9776
- b) 6779
- c) 1
- d) 0

27

Ce valoare contine variabila C de tip double a, in urma atribuirii:  $a = 9 * 3 / 2 * 2 / 3$ ?

- a) 9
- b) 8
- c) 0
- d) Atribuirea va genera o eroare deoarece nu este permisa o expresie aritmetica formata doar din constante

28

Fie tabela ANGAJATI (id\_angajat NUMBER(5) PRIMARY KEY, nume VARCHAR2(30), prenume VARCHAR2(30), salariul NUMBER(10,2), data\_angajare DATE, id\_departament NUMBER(3)) avand 200 randuri (inregistrari).

Alegeti varianta corecta referitoare la urmatoarea comanda SQL:

```
UPDATE angajati SET salariul=salariul *1.1
```

```
WHERE EXTRACT(month from data_angajare) = EXTRACT(month from sysdate)
```

```
AND id_departament IN (SELECT id_departament FROM angajati WHERE salariul > (SELECT AVG(salariul) FROM angajati));
```





- a) se majoreaza doar salariile angajatilor care au in prezent salariul mai mare decat salariul mediu
- b) se majoreaza doar salariile persoanelor angajate in luna curenta, indiferent de zi sau an, daca acestea lucreaza in prezent intr-un departament in care toti angajatii au salariul mai mare decat salariul mediu
- c) instructiunea este eronata deoarece in tabela nu exista coloana sysdate
- d) se majoreaza salariile persoanelor angajate in luna curenta, indiferent de zi sau an, daca acestea lucreaza in prezent intr-un departament in care exista angajati care au salariul mai mare decat salariul mediu

**29**

Fie tabela ANGAJATI (id\_angajat NUMBER(5) PRIMARY KEY, nume VARCHAR2(20), prenume VARCHAR2(30), salariul NUMBER(10,2), id\_departament NUMBER) avand 200 randuri (inregistrari).

Se considera urmatorul bloc PL/SQL:

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
CURSOR cursor1 IS SELECT id_angajat, nume FROM angajati ORDER BY salariul DESC;
```

```
vid angajati.id_angajat%TYPE;
```

```
vnume CHAR (20);
```

```
BEGIN
```

```
OPEN cursor1;
```

```
FETCH cursor1 INTO vid,vnume;
```

```
DBMS_OUTPUT.PUT_LINE('Angajatul ||vnume);
```

```
CLOSE cursor1;
```

```
END;
```

```
/
```

Precizati care dintre urmatoarele afirmatii este corecta:

- a) afiseaza numele tuturor angajatilor ordonati descrescator in functie de salariu
- b) lucreaza cu un cursor implicit
- c) blocul PL/SQL genereaza o eroare deoarece lipseste structura repetitiva
- d) se afiseaza numele angajatului cu cel mai mare salariu

**30**

Ce se va intampla la executia urmatorului program C?

```
#include<stdio.h>
```

```
struct st
```

```
{
```

```
    int x;
```

```
    struct st *next;
```

```
};
```

```
int main()
```

```
{
```

```
    struct st temp;
```

```
    temp.x = 100;
```

```
    temp.next = &temp;
```

```
    printf("%d", temp.next->next->x);
```

```
    return 0;
```

```
}
```

- a) Va fi generata o eroare de compilare deoarece recursivitatea in descrierea unei structuri nu este permisa
- b) Va fi afisata valoarea 100
- c) Va fi afisata valoarea 0
- d) Va fi afisata o valoare intreaga nedefinita

**Barem**

|    |   |
|----|---|
| 1  | b |
| 2  | c |
| 3  | b |
| 4  | b |
| 5  | d |
| 6  | c |
| 7  | a |
| 8  | d |
| 9  | d |
| 10 | c |
| 11 | d |
| 12 | b |
| 13 | d |
| 14 | d |
| 15 | a |
| 16 | a |
| 17 | c |
| 18 | d |
| 19 | c |
| 20 | b |
| 21 | c |
| 22 | b |
| 23 | b |
| 24 | c |
| 25 | a |
| 26 | b |
| 27 | b |
| 28 | d |
| 29 | d |
| 30 | b |