

**1)**În ce categorie de servicii oferite de serverele de baze de date se încadrează *managementul copiilor de siguranță (backup management)*:

- a)** servicii de gestiune a accesului la date;
- b)** servicii de gestiune a sesiunilor de lucru ale clienților;
- c)** servicii de jurnalizare-recuperare;
- d)** servicii de monitorizare și control;

**2)**Elaborarea *planului de execuție* în cadrul procesării frazelor SQL:

- a)** presupune determinarea căilor de access;
- b)** presupune analiza semantică și sintactică;
- c)** presupune optimizarea structurilor de stocare;
- d)** presupune accesarea înregistrărilor din tabele și indecși;
- e)** presupune accesarea datelor ce descriu starea structurilor (tabele, indecși) implicate;

**3)**În cadrul instanței Oracle, *procesul de background SMON* (system monitor) are ca responsabilități:

- a)** gestiunea operațiilor de scriere și citire a fișierelor jurnal,
- b)** gestiunea arhivării fișierelor jurnal;
- c)** controlul concurențial asupra datelor;
- d)** verificarea integrității fișierelor de date la pornirea instanței;

**4)***Fișierele de date* ale unei baze de date Oracle:

- a)** arhivează jurnalul tranzacțional al bazei de date;
- b)** sunt organizate în secțiuni de dimensiune constantă numite segmente de date;
- c)** sunt gestionate din punctul de vedere al alocării de spațiu către obiectele SQL prin intermediul tablespace-urilor;
- d)** pot găzdui și date temporare și UNDO privitoare la starea și procesările curente ale bazei de date;

**5)**Într-o bază de date Oracle, *stocarea datelor din tabelele de bază* (tabelele SQL) ale unei scheme relaționale:

- a)** nu se poate face aleator, întotdeauna ordinea de inserare este dictată de cheia primară,
- b)** se poate face strict ordonat după cheia primară, dacă forma de stocare presupune structuri de tip index;
- c)** se poate face într-o manieră relativ ordonată, respectând un criteriu de grupare;
- d)** se poate face, în anumite condiții specifice, folosind mai multe tipuri de segmente fizice (index și table);
- e)** de regulă se face relativ aleator, funcție de ordinea de inserare și modul dinamic de alocare a spațiului de stocare;

**6)***Planurile de execuție* ale cererilor SQL presupun operații de tipul:

- a)** selecție - exprimată prin predicate de access;
- b)** filtru - exprimat prin excluderea duplicatelor;
- c)** proiecție - exprimat prin liste ale coloanelor din tabelele de bază;
- d)** sortare - exprimată prin criteriile de grupare (GROUP BY) a datelor;
- e)** scanare totală indecși (full index scan) - pornind de la criteriile de ordonare a datelor (ORDER BY);

**7)**Forma *HASH* de stocare fizică a *tabelor* este caracterizată prin:

- a)** favorizarea accesării frecvente a tabelor folosind predicate bazate pe valori-exacte;
- b)** ineficientă din punctul de vedere al accesului pe baza coloanei implicate în repartizarea fizică a înregistrărilor pe baza algoritmului de hash-ing;
- c)** eficientă din punctul de vedere al stabilității volumului de spațiu necesar;
- d)** eficientizarea scanărilor după coloane multiple;

**8)**Opțiunea de optimizare(hint) *FIRST\_ROWS* specifică ORACLE presupune:

- a)** obținerea cât mai rapidă a primei/primelor linii din rezultatul final;
- b)** obținerea cât mai eficientă (din punctul de vedere al consumului de memorie) a primei/primelor linii din rezultatul final din prisma costurilor de execuție;

- c)** obținerea cât mai rapidă a primei linii din ultimul bloc al rezultatului final;
- d)** execuția cât mai rapidă a primei linii sau primului nod din planul de execuție;

9) Fiind dată următoarea structură a unei fraze de interogare:

```
SELECT CODPR, NRFACT, PRETUNIT  
FROM LINIIFACT  
WHERE CODPR = 2211  
AND NRFACT BETWEEN 4300 AND 4400  
AND CANTITATE > 50  
ORDER BY NRFACT, CODPR, PRETUNIT, CANTITATE;
```

Ce *structura de indexare* credeti ca este mai potrivita ?

- a)** CREATE INDEX LINIIFACT\_IDX ON LINIIFACT(CODPR, NRFACT, CANTITATE, PRETUNIT);
- b)** CREATE INDEX LINIIFACT\_IDX ON LINIIFACT(NRFACT, CODPR, CANTITATE, PRETUNIT);
- c)** CREATE INDEX LINIIFACT\_IDX ON LINIIFACT(CANTITATE, NRFACT, CODPR, PRETUNIT);
- d)** CREATE INDEX LINIIFACT\_IDX ON LINIIFACT(CODPR, CANTITATE, NRFACT, PRETUNIT);

**10)** Comanda

```
CREATE MATERIALIZED VIEW LOG ON <log_view_name>  
WITH ROWID,  
SEQUENCE(VALLINIE)  
INCLUDING NEW VALUES
```

are drept scop:

- a)** creare unei tabele virtuale temporare;
- b)** crearea unei tabele virtuale globale;
- c)** crearea unui jurnal în scopul actualizării tranzactionale a unei tabele virtuale materializate;
- d)** transformarea unei tabele virtuale convenționale într-o tabelă virtuală materializată;