

# Rezolvarea Temei de Casa Proiectarea Bazelor de date

## 1. (1p)

a) Să se dea un exemplu de atribut repetitiv (multivaloare) al unei entități în modelul entitate-legătură.

**R:** Considerand baza de date a unei firme de asigurari auto, pentru entitatea Client acesta poate asigura mai multe masini pe numele sau (dar o masina este atribuita unei singure persoane), deci atributul masina este unul repetitiv.

b) Să se dea un exemplu de atribut repetitiv (multivaloare) al unei relații mulți-la-mulți în modelul entitate-legătură.

**R:** Presupunand existenta unei entitati Server si o entiate Aplicatie, un Server poate gazdui pe el zero sau mai multe aplicatii (pe porturi diferite) iar o Aplicatie poate rula pe mai multe instante de servere diferite.

c) Să se arate cum se transformă attributele de mai sus la crearea design-ului logic al unei baze de date relaționale.

**R:** Attributele multivaloare (Server, Aplicatie, Masina) devin tabele dependente continand chei straine care fac referinta la cheile primare ale entitatilor.

## 2. (1p)

a) Să se dea un exemplu de relație de tip 3 (între mai mult de două entități) în modelul entitate-legătură.

**R:** O relatie de tip 3 poate fi relatia dintre Jucator-Pozitie-Echipa. Un jucator poate juca mai multe pozitii, o echipa are mai multe pozitii. Aceasta relatie nu s-ar putea sparge deoarece s-ar pierde informatia.

b) Să se dea un exemplu de trei sau mai multe entități care nu formează o relație de tip 3, ci, relația aparentă de tip 3, „se sparge” de fapt în relații mulți-la-mulți (între câte două entități).

**R:** Trei entitati care nu formeaza o relatie de tip 3 sunt entitatile Actor-Personaj-Film. Actorul joaca un personaj in mai multe filme. Relatia se sparge in actorul interpreteaza personajul, personajul apare in film, actorul apare in film. Relatia se poate sparge deoarece intr-un film un singur actor poate juca un personaj si putem reconstrui relatia initiala.

3. (1p)

a) Să se dea un exemplu de tabel relațional care este în FN1, dar nu în FN2. Să se aducă tabelul în FN2.

**R:**

FN1:

id_angajat	nume	job	id_job	locatie	pozitie	salariu
1	Marcel	bucatar	1	titan	sef	60
1	Marcel	chelner	2	titan	sef	60
2	Marinel	ingrijitor	3	dristor	trainee	10
3	Bogdan	bucatar	1	tineretului	ajutor	30
4	Cristi	chelner	2	titan	ajutor	30

FN2:

job	id_job	pozitie	salariu
bucatar	1	sef	60
chelner	2	trainee	10
ingrijitor	3	ajutor	30

id_angajat	nume	id_job	locatie	pozitie
1	Marcel	1	titan	sef
1	Marcel	2	titan	sef
2	Marinel	3	dristor	trainee
3	Bogdan	1	tineretului	ajutor
4	Cristi	2	titan	ingrijitor

b) Să se dea un exemplu de tabel relațional care este în FN2, dar nu în FN3.  
Să se aducă tabelul în FN3.

**R:**

FN2:

id_angajat	nume	id_job	locatie	pozitie
1	Marcel	1	titan	sef
1	Marcel	2	titan	sef
2	Marinel	3	dristor	trainee

3	Bogdan	1	tineretului	ajutor
4	Cristi	2	titan	ingrijitor

FN3:

id_angajat	nume	locatie	id_angajat	id_job	pozitie
1	Marcel	titan	1	1	sef
2	Marinel	dristor	1	2	sef
3	Bogdan	tineretului	2	3	trainee
4	Cristi	titan	3	1	ajutor
			4	2	ingrijitor

4. (0,5p) Să se dea un exemplu de tabel relațional în care există o dependență multivaloare (multidependență) între atributele (coloanele) sale, care nu este dependență funcțională.

**R:**

id_tehnician	unealta	salariu
1	ciocan	30
1	surubelnita	30
2	surubelnita	55
3	ciocan	55

5. (1p)

a) Să se illustreze printr-un exemplu structura unui index de tip arbore B\* și modul în care o interogare SQL folosește acest index.

**R:** De exemplu consideram ca avem un index arbore B\* pe tabelul pentru un aeroport, indexul fiind construit pe coloana nr\_zbor si dorim sa aflam ora de plecare a unui avion. Vom putea obtine ora\_plecare in acelasi numar de pasi pentru fiecare avion cautand nr\_zbor din radacina.

b) Să se illustreze printr-un exemplu structura unui index de tip bitmap și modul în care o interogare SQL folosește acest index.

**R:** Pentru un tabel apartament unde avem o coloana indexata tip bitmap nr\_camere si o alta coloana bitmap etaj putem efectua o interogare

```
SELECT * FROM apartament WHERE nr_camere = 4 AND etaj = 1
```

care va returna intrarile care respecta operatia logica 0100 AND 0001.

6. (0,5p) Să se dea un exemplu de vedere (vizualizare) VIEW\_EX astfel încât comanda INSERT INTO VIEW\_EX VALUES să producă o eroare. Să se explice cauza erorii.

**R:**    CREATE VIEW VIEW\_EX AS

```
SELECT cod_client, cod_masina, an_fabricatie
```

```
FROM masina
```

```
WHERE an_fabricatie > 2000
```

```
INSERT INTO VIEW_EX VALUES(3, 44, 2001)
```

În aceasta vizualizare ne-ar apărea doar intrările pentru mașini din acest mileniu. Încercând să introducem în această vizualizare o nouă mașină cu cod\_masina 33 acest lucru ar putea eșua deoarece cod\_masina este PK și poate aparține unei mașini ce nu apare în vizualizare (<2000). De asemenea dacă tabelul din care este creat vizualizarea conține mai multe

atribute si acestea nu au valori default setate atunci inserarea va produce o eroare deoarece nu oferim valori pentru acestea.

7. (0,5p) Să se illustreze printr-un exemplu modul în care o vedere (vizualizare) poate fi folosită pentru a asigura securitatea într-o bază de date.

**R:** Vizualizarile nu stocheaza date propriu-zise ci doar ofera o afisare a acestora, asadar putem folosi vizualizarile pentru a limita informatiile pe care utilizatorii le pot vedea sau accesa. Sa presupunem ca avem un tabel in care tinem clientii cu comenzi, vechimea lor, parola si adresa personala. Putem crea o vizualizare din acest tabel unde tot ce ar fi accesibil este numele clientului si numarul total de comenzi.

8. (1p) Să se arate printr-un exemplu în ce condiții este posibil ca două select-uri identice consecutive (fără nici o altă comandă între ele), efectuate în aceeași sesiune de lucru, pe același tabel, pot produce rezultate diferite.

**R:** SELECT \*

FROM masina

ORDER BY DBMS\_RANDOM.RANDOM

Daca rulam de doua ori succesive interogarea vom obtine aceleasi dar rezultatele intr-o alta ordine. (Cred ca ceva similar s-ar putea obtine folosind valori autoincrement.)

9. (0,5p) Să se dea un exemplu care să illustreze interblocarea.

**R:** Prin incercarea de a folosi aceiasi resursa de catre mai multi utilizatori se poate ajunge la interblocare. Un exemplu ar fi:

<p>Tranzactia A:</p> <p>UPDATE masina SET taxa = 5000 WHERE an_fabricatie &lt; 2000</p> <p>UPDATE client</p>	<p>Tranzactia B:</p> <p>UPDATE client SET tip_asigurare = "premium" WHERE cod_client = 40</p> <p>UPDATE masina</p>
--	--

SET numar_telefon = "0729571924" WHERE cod_client = 42	SET carburant = "benzina" WHERE cod_masina = 33 AND cod_client = 40
---	---

10. (1p) Să se illustreze printr-un exemplu utilizarea unui trigger pentru a realiza o constrangere de integritate care nu ar putea fi implementată folosind un CONSTRAINT din definiția unui tabel.

**R:**

```
CREATE TRIGGER trigger_client_major
```

```
  BEFORE INSERT ON client
```

```
  FOR EACH ROW
```

```
BEGIN
```

```
  IF(ADD_MONTHS(NEW.client_datanastere, 18 * 12) > sysdate ) THEN
```

```
    RAISE_APPLICATION_ERROR(-20001, 'Clientul trebuie sa fie major' );
```

```
  END IF;
```

```
END;
```

11. (1p) Să se illustreze printr-un exemplu de program PL/SQL multi-bloc modul de propagare a excepțiilor. Vor fi ilustrate cel puțin situațiile în care o excepție este tratată sau nu în blocul curent și în care controlul programului va fi transmis blocului următor din secvență sau blocului exterior.

12. (1p) Să se illustreze prin exemple folosirea instrucțiunii RAISE pentru a ridica atât o excepție predefinită cât și o excepție definită de utilizator. În cazul excepțiilor predefinite, să se explice cum anume folosirea instrucțiunii RAISE schimbă funcționalitatea programului (față de cazul când această instrucțiune nu există).