

10.01. Se dau relațiile:

Persoana(id, nume, email, adresa)

Deviz(id_d, data_introducere, aparat, simptome, defect, data_constatare,
data_finalizare, durata, manopera_ora, total, id_client, id_depanator)

Piesa(id_p, descriere, fabricant, cantitate_stoc, pret_c)

Piesa_Deviz(id_d, id_p, cantitate, pret_r)

ce reprezintă schema bazei de date pentru un atelier de reparații. O persoană poate avea rolurile de client (id_client) respectiv depanator (id_depanator). Coloana *aparat* conține denumirea aparatului, coloana *durata* conține durata în ore necesară pentru reparare, coloana *total* este valoarea totală a devizului, obținută prin însumarea costului pieselor cu manopera. Inițial *data_constatare* și *data_finalizare* au valoarea NULL. O piesă are un preț de catalog (Piesa.pret_c) și un preț real (Piesa_Deviz.pret_r). Pentru Piesa combinația (descriere, fabricant) este unică.

Să se scrie următoarele instrucțiuni:

- creare tabelă pentru relația Persoana;
- creare tabelă pentru relația Deviz;
- creare tabelă pentru relația Piesa;
- creare tabelă pentru relația Piesa_Deviz;
- să se declare cheile primare și străine;
- modificare definiție tabelă Persoana pentru a modifica lungimea coloanei adresa la 200 caractere.

10.02. Să se exprime următoarele constrângeri (la nivel atribut sau tuplă):

- În tabela Persoana coloana *email* trebuie să conțină caracterul '@' și o adresă de email trebuie să aibă lungimea minimă 10 caractere (inclusiv caracterul '@').
- În tabela Deviz dacă *data_finalizare* este completată, atunci *data_constatare* nu poate fi NULL.

10.03. Să se exprime în SQL următoarele interogări:

- Să se găsească detaliile pentru devizele care au constatare și sunt nefinalizate la data '01-MAR-2024', ordonat descrescător după *data_constatare* și crescător după *data_introducere*.
- Să se găsească detaliile pieselor care au *pret_c* mai mare decât 10 descrescător după *pret_c*.

10.04. Să se exprime în SQL următoarele interogări folosind operatorul JOIN:

- Să se găsească nume depanator, descriere, fabricant, pret_c, pret_r pentru piesele cu prețul de catalog mai mare decât prețul real.
- Să se găsească perechi de devize (id_d1, id_d2) cu condiția să aibă cel puțin o piesă în comun folosită în aceeași cantitate. O pereche este unică în rezultat.

10.05. Să se exprime în SQL **fără funcții de agregare** următoarele interogări folosind cel puțin o interogare imbricată și operatori de genul EXISTS, IN, ALL, ANY:

- a) Să se găsească detaliile devizelor pentru care s-au folosit piese a căror descriere conține 'șurub' în cantitate minimă.
- b) Să se găsească descrierea și fabricantul pentru piesele cu prețul de catalog identic cu prețul real al piesei cu $id_p = 1$ folosită la devizul cu $id_d = 1$.

10.06. Să se exprime în SQL următoarele interogări folosind funcții de agregare:

- a) Să se găsească pentru fiecare depanator total minim, total mediu și total maxim pentru devizele cu *data_introducere* în anul 2024.
- b) Să se găsească pentru fiecare deviz suma valorii reale (valoare = cantitate * pret_r).

10.07. Să se scrie instrucțiunile pentru actualizarea BD:

- a) Să se adauge devizul cu identificatorul 123 din data 30 Septembrie 2024, aparatul 'TV Samsung' cu simptome 'image desincronizata' clientul cu identificator 11, depanator alocat cu identificator 17.
- b) Să se șteargă liniile din Piesa_Deviz dacă anul finalizării devizului este precedent anului 2023.
- c) Să se crească *manopera_ora* cu 10 și proporțional *total* pentru devizele finalizate în luna septembrie 2024.

10.08. Să se definească o procedură stocată care va introduce în tabela *Excepții* acele linii din tabela *Piesa_Deviz* ce respectă condiția $pret_r > pret_c$ (pentru piesa respectivă) sau $data_constatare = data_finalizare$ (pentru devizul respectiv). Tabela *Excepții* va avea aceleași coloane ca și tabela *Piesa_Deviz* plus o coloană ce indică natura excepției.

10.09. Să se definească trigger pentru:

- a) A actualiza *cantitate_stoc* pentru piesă la adăugarea sau ștergerea piesei pentru un deviz.

- b) Presupunând vederea:

```
CREATE VIEW PieseDeviz123 AS
```

```
SELECT data_introducere, aparat, simptome, defect, data_constatare,  
       data_finalizare, durata, manopera_ora, total, id_client,  
       a.numa as client, id_depanator, b.numa as depanator,  
       descriere, p.id_p, fabricant, cantitate_stoc, pret_c, cantitate, pret_r
```

```
FROM Persoana a, Persoana b, Deviz d, Piesa_Deviz c, Piesa p
```

```
WHERE d.id_d = 123 AND
```

```
      a.id = d.id_client AND b.id = d.id_depanator AND
```

```
      c.id_d = d.id_d AND p.id_p = c.id_p;
```

Să se definească un trigger instead-of pentru a permite adăugare prin această vedere. (la un deviz pot fi utilizate mai multe piese)