06.01. Se dau relaţiile:

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 instrucţiuni:

a) creare tabelă pentru relaţia Persoana;
b) creare tabelă pentru relaţia Deviz;
c) creare tabelă pentru relaţia Piesa;
d) creare tabelă pentru relaţia Piesa_Deviz;
e) să se declare cheile primare şi străine;
f) modificare definiţie tabelă Piesa_Deviz pentru a adăuga atributul sursa.

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

a) În tabela Persoana coloana *email* trebuie să conțină caracterul ‚@’.
b) În tabela Deviz data_introducere, data_constatare și data_finalizare trebuie să fie consecvente.

06.03. Să se exprime în SQL următoarele interogări:

a) Să se găsească detaliile pentru devizele care au constatare și sunt nefinalizate la data ‚01-SEP-2023’, ordonat după data introducere.
b) Să se găsească detaliile pieselor care au *cantitate_stoc* sub 5 ordonat crescător după *cantitate_stoc* și descrescător după *descriere*.

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

a) Să se găsească (id_d, descriere, fabricant, pret_c și pret_r) pentru piesele cu prețul de catalog mai mare decât prețul real.
b) Să se găsească perechi de piese (id_p1, id_p2) care apar pe același deviz în aceeași cantitate. O pereche este unică în rezultat.

06.05. Să se exprime în SQL fără funcții de agregare următoarele interogări folosind cel puţin o interogare imbricată şi operatori de genul EXISTS, IN, ALL, ANY:
a) Să se găsească detaliile devizelor care au folosit piesa cu descrierea ‚șurub'.
b) Să se găsească descrierea și fabricantul pentru piesa cu prețul real cel mai mare.

06.06. Să se exprime în SQL următoarele interogări folosind funcţii de agregare:
a) Să se găsească pentru fiecare depanator numărul de devize (nume_depanator, câte_devize) cu data_finalizare în luna septembrie 2023.
b) Să se găsească pentru fiecare piesă folosită la devize cu data_finalizare în luna septembrie 2023 cantitatea totală (descriere, fabricant, cantitate_totală).

06.07. Să se scrie instrucţiunile pentru actualizarea BD:
a) Să se adauge devizul cu identificatorul 123 din data 30 Septembrie 2023, aparatul 'TV Samsung' cu simptome 'imagine desincronizata' clientul cu identificator 11, depanator alocat cu identificator 17.
b) Să se șteargă piesele care nu sunt folosite la nici un deviz.
c) Să se modifice *total* scăzând cu 5% pentru devizul cu id 111.

06.08. 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.

06.09. Să se definească triggere 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.nume as client, id_depanator, b.nume 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)