

06.01. Se dau relațiile:

Proprietar (id\_proprietar, nume, prenume, email)

Apartament (id\_ap, adresa, nr\_ap, suprafata, id\_proprietar)

Consum (id\_ap, an, luna, nr\_pers, cantitate, valoare, pret\_apa)

Chitanță (nr, data, id\_ap, valoare)

Pentru evidența cheltuielilor unei asociații de proprietari se folosește schema de mai sus. Tabela *Consum* (cheltuieli) păstrează câte o înregistrare pentru fiecare apartament, fiecare lună a unui an calendaristic. Coloana *cantitate* reprezintă consumul de apă (m<sup>3</sup>), iar coloana *valoare* reprezintă suma de plată. Tabela *Chitanță* păstrează pentru fiecare apartament valorile încasate. Adresa conține localitate, strada, număr. Se definește restanțier la data X, un proprietar de apartament pentru care suma valorilor din Chitanță cu data  $\leq X$  este mai mică decât suma valorilor din Consum cu  $(an, luna) < (X.an, X.luna)$ .

Să se scrie următoarele instrucțiuni:

- creare tabelă pentru relația Proprietar;
- creare tabelă pentru relația Apartament;
- creare tabelă pentru relația Consum;
- creare tabelă pentru relația Chitanță;
- să se declare cheile primare și străine;
- modificare definiție tabelă Proprietar pentru a modifica lungimea coloanei *nume* la 100 caractere.

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

- Coloana *data* trebuie să ia valori ulterioare anului 2023.
- Dacă *nr\_ap* în Apartament are valoare pară atunci *suprafata* trebuie să fie între 70 și 80.

06.03. Să se exprime în SQL următoarele interogări:

- Să se găsească detaliile proprietarilor al căror nume conține 'p' sau conține 'e' ordonat descrescător după *nume* și crescător după *prenume*.
- Să se găsească detaliile chitanțelor din anul 2024 ordonat descrescător după *valoare*.

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

- Să se genereze lista de cheltuieli pentru luna ianuarie 2024, pentru apartamentele de la adresa 'Cluj-Napoca strada Observatorului nr. 9' (nr\_ap, proprietar, valoare).
- Să se găsească perechi de apartamente (id\_ap1, id\_ap2) cu condiția să aibă aceeași valoare Consum în același an și lună. O pereche este unică în rezultat.

06.05. 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 consumului cu valoarea maximă pentru luna ianuarie 2024.
- b) Să se găsească numele proprietarilor ce au aceeași valoare consum în luna martie 2024 ca și proprietarul apartamentului cu id\_ap = 1.

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

- a) Să se găsească valoarea minimă, medie și maximă încasată per trimestru pentru fiecare apartament (trimestru, id\_ap, minim, mediu, maxim).
- b) Să se găsească numele restanțierilor la data '01-OCT-2024' de la adresa 'Turda strada Baladei nr. 3'.

06.07. Să se scrie instrucțiunile pentru actualizarea BD:

- a) Să se introducă în BD faptul că apartamentul cu identificatorul 4 are pentru luna 9 anul 2024, 3 persoane, 23 m<sup>3</sup> apă consumată și valoarea 150 și în data 01-OCT-2024 s-a încasat chitanța cu numărul 14 în valoare de 150.
- b) Să se șteargă apartamentele care nu au consum.
- c) Să se actualizeze tabela Consum pentru a reflecta faptul că apartamentul cu identificator 1, pentru lunile 1 și 2 anul 2024 are numărul de persoane 0 și valoarea 35.

06.08. Să se definească o procedură stocată care va introduce în tabela *Excepții* acele linii din tabela *Chitanță* cu condiția anul și luna din data chitanței depășește cu 1 anul și luna consum pentru un anumit apartament, și valoarea chitanței este mai mare decât valoarea consumului lunar sau valoarea chitanței este mai mică decât valoarea consumului lunar. Tabela *Excepții* va avea aceleași coloane ca și tabela *Chitanță* plus o coloană ce indică natura excepției și diferența valorilor.

06.09. Să se definească trigere pentru:

- a) A asigura că nu se permite ștergerea unei chitanțe sau modificarea detaliilor unei chitanțe dacă există consum cu *an\_luna* consum ulterior datei chitanței.

- b) Presupunând vederea:

```
CREATE VIEW Consum_Ap_Turda AS
SELECT id_proprietar, nume, email, id_ap, adresa, nr_ap, suprafata, an, luna,
       nr_pers, cantitate, valoare, pret_apa
FROM Proprietar NATURAL JOIN Apartament NATURAL JOIN Consum
WHERE adresa LIKE 'Turda%';
```

Să se definească un trigere instead-of pentru a permite adăugare prin această vedere. (un apartament poate avea 12 consumuri într-un an calendaristic, un proprietar poate deține mai multe apartamente)