11.01. Se dau relaţiile:

Proprietar (id_proprietar, nume, 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* păstrează câte o înregistrare pentru fiecare apartament, fiecare lună a unui an calendaristic. Coloana *cantitate* reprezintă consumul de apă (m^3), 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 ≤ X este mai mică decât suma valorilor din Consum cu (an, luna) < (X.an, X.luna)).

Să se scrie următoarele instrucţiuni:
a) creare tabelă pentru relaţia Proprietar;
b) creare tabelă pentru relaţia Apartament;
c) creare tabelă pentru relaţia Consum;
d) creare tabelă pentru relaţia Chitanță;
e) să se declare cheile primare şi străine;
f) modificare definiţie tabelă Proprietar pentru a adăuga atributul *telefon*.

11.02. Să se exprime următoarele constrângeri (la nivel atribut sau tuplă):
a) Atributul an poate lua valori între 2017 și 2019.
b) Dacă *cantitate* este zero atunci numărul de persoane trebuie sa fie zero.

11.03. Să se exprime în SQL următoarele interogări:
a) Să se găsească numele și email pentru proprietarii al căror email conține 'yahoo' ordonat după nume.
b) Să se găsească apartamentele cu suprafața între 60 și 80 mp în ordinea descrescătoare a adresei.

11.04. Să se exprime în SQL următoarele interogări folosind operatorul JOIN:
a) Să se găsească perechi de apartamente (id_ap1, id_ap2) cu condiția să aparțină la același proprietar. O pereche este unică în rezultat.
b) Să se genereze lista de cheltuieli pentru anul 2018, luna august, pentru apartamentele de la adresa 'Turda Băița 17' (adresa, nr_ap, proprietar, valoare).

11.05. 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 adresa, numărul de apartament și luna pentru consumul cu cea mai mare valoare pentru anul 2018.
b) Să se găsească numele proprietarilor ce dețin mai multe apartamente.

11.06. Să se exprime în SQL următoarele interogări folosind funcţii de agregare:
a) Să se găsească valoarea medie de consum per an pentru fiecare apartament.
b) Să se găsească numele restanțierilor la data ’21-Sep-2018’.

11.07. Să se scrie instrucţiunile pentru actualizarea BD:
a) Să se introducă în BD faptul că apartamentul cu identificatorul 4 are pentru luna 9 anul 2018, 3 persoane, 23 m^3 apă consumată și valoarea 150 și în data 01-OCT-2018 s-a încasat chitanța cu numărul 14 în valoare de 150.
b) Să se şteargă apartamentele care nu au consum.
c) Să se actualizeze în relaţia Consum pentru a reflecta faptul că la apartamentul cu identificator 111, pentru lunile 8 și 9 anul 2018 numărul de persoane este 1 și valoarea este 75.

11.08. Să se definească trigere pentru:
a) A asigura că la adăugarea unui consum valoarea să se actualizeze după formula: valoare = nr_pers*ceva+cantitate*altceva, unde ceva și altceva sunt citite dintr-o tabelă auxiliară (Tarif).
b) A asigura că nu se permite modificarea unui consum lunar dacă există chitanță cu data ulterioară an_luna consum.
c) Presupunând vederea:

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

Să se definească un triger instead-of pentru a permite adăugare prin această vedere.