

01.01. Se dau relațiile:

Persoană (id_pers, nume, telefon, adresa)

Carte (id_carte, titlu, nr_pagini, nr_exemplare, gen)

Împrumut (id_carte, id_imp, datai, datar, nr_zile)

Autor (id_carte, id_aut)

Pentru evidența unei biblioteci se folosește schema de mai sus. O carte poate avea mai mulți autori. Coloanele *id_imp* și *id_aut* iau valori din domeniul *id_pers*. La efectuarea unui împrumut se completează *datai* (data împrumut), coloana *datar* (data restituirii) ia valoarea NULL, iar coloana *nr_zile* reprezintă termenul de restituire (câte zile poate dura împrumutul). La restituire se va completa *datar*. Se presupune că o persoană împrumută o anumită carte o singură dată într-o zi (*datai* este data împrumutului).

Să se scrie următoarele instrucțiuni:

- creare tabelă pentru relația Persoană;
- creare tabelă pentru relația Carte;
- creare tabelă pentru relația Împrumut;
- creare tabelă pentru relația Autor;
- să se declare cheile primare și străine;
- modificare definiție tabelă Carte pentru a adăuga atributul *rezumat*.

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

- Coloana *gen* trebuie să ia valori din lista: {'Autobiografic', 'Copii', 'Fantastic', 'Tehnic'}.
- Dacă *gen* este 'Tehnic', atunci *nr_exemplare* este mai mare decât 10.

01.03. Să se exprime în SQL următoarele interogări:

- Să se găsească detaliile persoanelor cu număr de telefon ce începe cu '+44', ordonat crescător după *adresa*.
- Să se găsească detaliile împrumuturilor cu restituire întârziată în ordinea descrescătoare după *nr_zile* și crescător după *datai*.

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

- Să se găsească numele, adresa și numărul de telefon pentru persoanele ce au depășit termenul de restituire cu mai mult de 7 zile.
- Să se găsească perechi de autori (*id_aut1*, *id_aut2*) care au scris cărți de același gen. O pereche este unică în rezultat.

01.05. 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 cărților care au genul cărții cu titlul 'India'.
- b) Să se găsească titlul și autorii cărților cu mai mulți autori.

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

- a) Să se găsească numele persoanelor cu întârzierea cea mai mare în restituirea cărților împrumutate și valoarea în număr zile a întârzierii.
- b) Să se găsească numărul de pagini minim, mediu și maxim pentru fiecare gen.

01.07. Să se scrie instrucțiunile pentru actualizarea BD:

- a) Să se introducă în BD faptul că Mircea Cărtărescu cu numărul de telefon 021-0134567 a scris cartea cu titlul Visul, gen Fantastic, 294 pagini, 100 exemplare.
- b) Să se șteargă cărțile care nu sunt împrumutate.
- c) Să se actualizeze *nr_zile* pentru împrumuturile nerestituite ce au *nr_zile* mai mic decât diferența între data curentă și data împrumutului, crescând *nr_zile* cu diferența între data curentă și data împrumutului.

01.08. Să se definească trigger pentru:

- a) A asigura că la adăugarea unui împrumut, *nr_zile* este sub 15 dacă numele împrumutătorului este 'John Doe'.
- b) A împiedica adăugarea unui autor al unei cărți deja împrumutate.
- c) Presupunând vederea:

```
CREATE VIEW Carti_Tehnice AS
```

```
SELECT titlu, nr_pagini, nr_exemplare, nume AS autor, telefon
```

```
FROM Carte NATURAL JOIN Autor JOIN Persoană ON id_pers = id_aut
```

```
WHERE gen = 'Tehnic';
```

Să se definească un trigger instead-of pentru a permite adăugare prin această vedere (o carte poate avea mai mulți autori).