

01.01. Se dau relațiile:

Client(id_c, nume, adresa)

Factura(id_f, data, nr_slideuri, cost_slide, nr_zile, total, id_c)

Localitate(id_l, denumire)

Difuzare(id_f, id_l, datai, datas)

ce reprezintă schema bazei de date pentru o firmă de publicitate. Se presupune că valorile din coloanele *cost_slide* și *total* sunt exprimate în EUR. Un client difuzează un anumit număr de slideuri în mai multe localități. Perioadele de difuzare pot să difere între localități. Pentru calculul *total* contează numărul total de zile (*nrzile*). ($\text{total} = \text{nr_slideuri} * \text{cost_slide} * \text{nr_zile}$)

Să se scrie următoarele instrucțiuni:

- creare tabelă pentru relația Client;
- creare tabelă pentru relația Factura;
- creare tabelă pentru relația Localitate;
- creare tabelă pentru relația Difuzare;
- să se declare cheile primare și străine;
- modificare definiție tabelă Factura pentru a adăuga atributul *tva*.

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

- Atributul *nr_slideuri* (tabela Factura) trebuie să aibă o valoare cuprinsă între 1 și 40.
- Dacă *nume* (tabela Client) începe cu ,firmă' atunci *adresa* trebuie să conțină ,sediul social'.

01.03. Să se exprime în SQL următoarele interogări:

- Să se afișeze detaliile pentru facturile cu totalul mai mare decât 750 ordonat descrescător după *data* și crescător după *total*.
- Să se găsească denumirea localităților care se termină cu ,a'.

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

- Să se găsească numele clienților care au facturi emise în anul 2019.
- Să se găsească perechi de localități (*id_l1*, *id_l2*) cu condiția că pentru ambele localități difuzarea s-a facturat pe aceeași factură. 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 denumirea localităților unde au existat difuzări în luna septembrie 2019.
- b) Să se găsească numele clientului care are factura cu total maxim în anul 2019.

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

- a) Să se găsească pentru fiecare localitate și fiecare zi din luna septembrie 2019 numărul total de slide-uri.
- b) Să se găsească valoare medie totală pentru facturile emise în 2018.

01.07. Să se scrie instrucțiunile pentru actualizarea BD:

- a) Să se adauge factura cu numărul 123 pentru clientul cu id_c = 1 (,Georgescu Mircea') și celelalte detalii: data = ,01-OCT-2019', nr_slideuri = 4, cost_slide = 4, nr_zile = 10, total = 160.
- b) Să se modifice total pentru facturile cu nr_zile peste 10 scăzând 5% din total.
- c) Să se șteargă facturile ce nu au specificată difuzare.

01.08. Să se definească trigere pentru:

- a) A actualiza numărul de zile și totalul facturii, la adăugarea sau ștergerea în tabela Difuzare.
- b) A asigura că orice difuzare are data de început și data de sfârșit consecvente (ulterioare datei facturii și între ele relația datai < datas).
- c) Presupunând vederea:

```
CREATE VIEW Facturi_Ionescu_Paul AS
SELECT id_f, data, nr_slide, cost_slide, nr_zile, total, denumire AS localitate,
       datai, datas
FROM Client NATURAL JOIN Factura NATURAL JOIN
     Difuzare NATURAL JOIN Localitate
WHERE nume = 'Ionescu Paul';
```

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