

05.01. Se dau relațiile:

Client(id\_c, nume, adresa)

Factura(id\_f, data, nr\_pagini, cost\_pagina, nr\_zile, valoare, tva, 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\_pagina* și *valoare* sunt exprimate în EUR. Un client primește **1 factură pentru difuzarea** unui anumit număr de pagini **în una sau mai multe localități**. Perioadele de difuzare (datai, datas) pot să difere între localități ce corespund difuzării pe aceeași factură (datai = data de început difuzare, datas = data de sfârșit difuzare). Calculul *valoare* de pe factură se face cu formula:  $valoare = nr\_pagini * cost\_pagina * nr\_zile$ , unde *nr\_zile* reprezintă suma numărului de zile de difuzare pentru toate localitățile corespunzătoare facturii.

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 *valoare\_totala* ( $= valoare + tva$ ).

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

- Atributul *tva* (tabela Factura) trebuie să respecte formula  $tva = .19 * valoare$ .
- Dacă *nume* începe cu 'firma' atunci *adresa* trebuie să conțină 'sediu social'.

05.03. Să se exprime în SQL următoarele interogări:

- Să se afișeze detaliile pentru facturile cu valoare între 500 și 1000 ordonat descrescător după *data* și crescător după *valoare*.
- Să se găsească denumirea localităților care au în denumire litera 'j' ordonat alfabetic.

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

- Să se găsească numele clienților care au facturi corespunzătoare difuzării în localitățile Oradea și Aleșd.
- Să se găsească perechi de localități (*id\_l1*, *id\_l2*) cu condiția difuzarea să apară pe aceeași factură și să se suprapună perioadele de difuzare. O pereche este unică în rezultat.

05.05. 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 octombrie 2022 și factura are valoarea sub 100.
- b) Să se găsească numele clientului care are factura cu cel mai mic număr de zile în anul 2022.

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

- a) Să se găsească pentru fiecare localitate câte pagini trebuie difuzate în data curentă.
- b) Să se găsească valoare totală minimă, medie și maximă pentru facturile emise în anul 2022.

05.07. Să se scrie instrucțiunile pentru actualizarea BD:

- a) Să se adauge factura cu numărul 123 pentru clientul cu id\_c = 1 ('John Doe') și celelalte detalii: data = '02-OCT-2022', nr\_pagini = 4, cost\_pagina = 4, nr\_zile = 10, valoare = 160, tva = 30.4, valoare\_totala = 190,4 .
- b) Să se șteargă facturile ce nu au difuzări.
- c) Să se modifice *valoare*, *tva* și *valoare\_totala* pentru facturile cu nr\_zile peste 20 aplicând o reducere de 5%.

05.08. Să se definească trigere pentru:

- a) A actualiza numărul de zile și celelalte attribute implicate ale facturii (*valoare*, *tva*, *valoare\_totala*), la adăugarea sau ștergerea în tabela Difuzare.
- b) A preveni modificarea data de început și data de sfârșit difuzare dacă data curentă este ulterioară *datai*.

c) Presupunând vederea:

```
CREATE VIEW Facturi_John_Doe AS
SELECT id_f, data, nr_pagini, cost_pagina, nr_zile, valoare, tva, valoare_totala,
       denumire AS localitate, datai, datas
FROM Client NATURAL JOIN Factura NATURAL JOIN
     Difuzare NATURAL JOIN Localitate
WHERE nume = 'John Doe';
```

Să se definească un triger instead-of pentru a permite adăugare prin această vedere (pe 1 factură pot apărea difuzări în mai multe localități).