

Proiect Baze de Date
Calomfirescu Victor – 144

Content

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare:	5
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului:.....	5
3. Descrierea entităților și cheile primare:.....	6
4. Descrierea relațiilor și cardinalitățile:	6
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.	6
6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.....	8
7. Diagrama conceptuală	8
8. Schemele relaționale corespunzătoare diagramei conceptuale proiectate la pct 7	9
9. Realizarea normalizării până la forma normală 3 (FN1-FN3).	11
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).	14
11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea	14
12. Cereri SQL Complexe	16
Cerere 1. Subcerere corelată pe 3 tabele. RESPECTA PUNCTUL a).....	17
Cerere 2. Subcereri nesincronizate în FROM . RESPECTA PUNCTUL b).....	17
Cerere 3. Grupări, funcții grup și HAVING cu subcerere nesincronizată RESPECTA PUNCTUL c).....	19
Cerere 4. ORDER BY, COALESCE (NVL) și DECODE (expresie CASE) RESPECTA PUNCTUL d)	20
Cerere 5. Funcții pe șiruri și date, expresie CASE, bloc WITH. RESPECTA PUNCTELE e) și f)	21
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri	23
14. Crearea unei vizualizări complexe	28
15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația <i>outer-join</i> pe minimum 4 tabele, o cerere ce utilizează operația <i>division</i> și o cerere care implementează analiza top- <i>n</i>	29

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare:

Compania (clientul) închiriază online unul sau mai multe birouri (sau spații de lucru) dintr-un centru de co-working, pentru o perioadă de timp, cu scopul de a fi folosite de utilizatori (angajați sau colaboratori ai companiei). Compania semnează contractul și efectuează plățile. Utilizatorul este persoana care folosește efectiv spațiul de lucru. Birourile (spațiile de lucru) se pot încadra în diverse tipuri, au un anumit număr de locuri (1–4), pot dispune de diverse facilități se pot afla în diferite cladiri. Cladirile sunt de un anumit tip, au o adresa, si un proprietar. Plata se face prin diverse metode (cash, card, transfer bancar). Doar compania are dreptul să modifice perioada de închiriere, nu și utilizatorii.

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului:

- Valorile cheilor primare (Companie_ID, Inchiriere_ID, Birou_ID, Facilitate_ID, Tip_Birouri_ID, Cladiri_ID, Numar_Locuri_ID, Metoda_plata_ID, Utilizator_ID, Plati_ID) nu pot fi NULL
- Există un număr limitat de cladiri, MAX 10, fiecare cladire are maxim 10 etaje.
- Birourile pot fi in diferite cladiri, iar birourile sunt numerotate de la 1 la 100 (de exemplu, biroul 21 se află pe etajul 2).
- Un birou are maximum 4 locuri.
- Birourile sunt de mai multe tipuri, de exemplu:
 - 'Birou deschis',
 - 'Birou dedicat',
 - 'Birou privat',
 - 'Birou pentru echipă',
 - 'Sală de conferință'
- Facilitățile posibile includ:
 - 'Espressor de cafea',
 - 'Imprimanta',
 - 'TV',
 - 'Proiector',
 - 'Microfoane profesionale',
 - 'Tablă smart',
 - 'Aparatură de conferință video',
 - 'Parcare',
 - 'Serviciu de recepție';
- Plățile se pot face prin: cash, card, transfer bancar.
- Doar compania poate modifica data de început și data de final a închirierii.

3. Descrierea entităților și cheile primare:

- Companii care are cheie primara: Companii _ID
- Inchirieri care are cheie primara: Inchirieri _ID
- Plati care are cheie primara: Plati_ID
- Metoda_de_Plata care are cheie primara: Metoda_plata_ID
- Cladiri care cheia primara: Cladiri_ID
- Birouri care are cheie primara: Birouri _ID
- Tip_Birouri care are cheie primara: Tip_ Birouri _ID
- Numar_Locuri care are cheie primara: Numar_ Locuri _ID
- Lista_Facilitati care are cheie primara: Facilitate_ID
- Utilizatori care are cheie primara: Utilizatori _ID
- Inchiriere_birouri: nu are cheie primara si este un tabel asociativ ce leaga entitatile: Utilizator, Inchiriere si Birou.
- Facilitati_Birouri: nu are cheie primara si are un tabel asociativ ce leaga entitatile: Birouri si Lista_Facilitati.

4. Descrierea relațiilor și cardinalitățile:

- O Companie poate face mai multe Inchirieri (1 : M).
- O Companie poate face mai multe Plati (1 : M).
- O Inchiriere poate include mai multe rezervări de birouri și utilizatori (1 : M).
- O Inchiriere poate fi achitată prin mai multe Plati (1 : M).
- O Metoda_de_Plata este asociată mai multor Plati (1 : M).
- Un Utilizator poate participa la mai multe Inchirieri (1 : M).
- Un Birou poate apărea în mai multe Inchirieri (1 : M).
- Un Birou are mai multe facilități (1 : M).
- O Inchiriere are mai multe Plati (1:M)
- O Cladire are mai multe birouri. (1:M)
- O Facilitate se poate regăsi în mai multe birouri (1 : M).
- Un Tip_Birouri poate fi asociat mai multor Birouri (1 : M).
- Un Numar_Locuri poate fi asociat mai multor Birouri (1 : M).

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

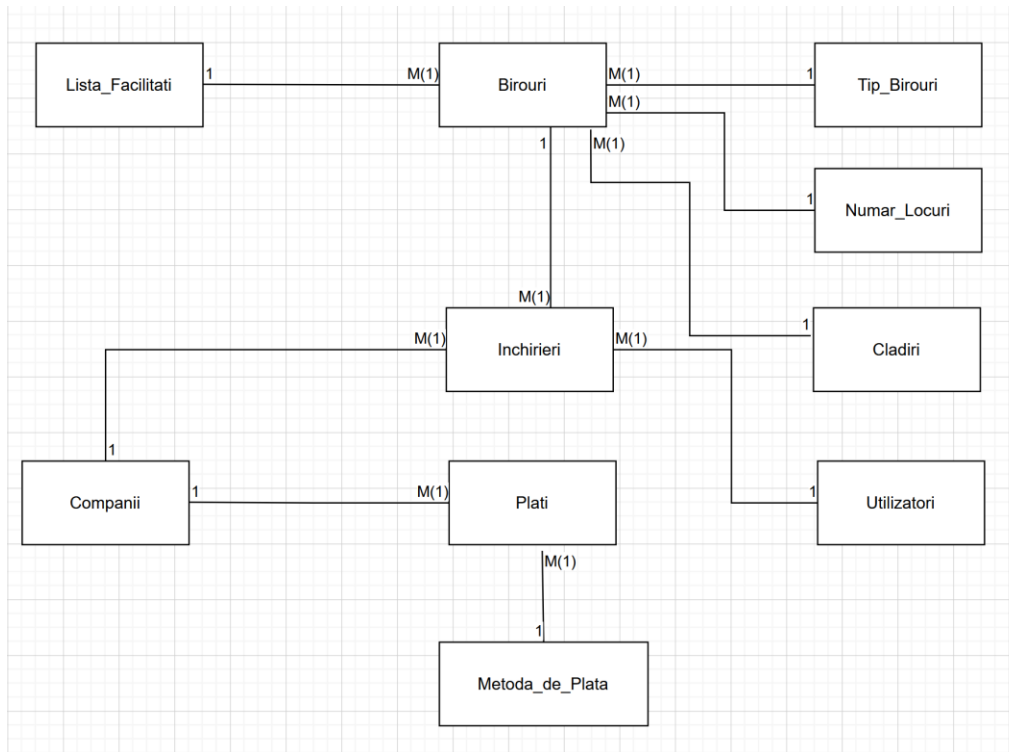
Descrierea atributelor:

- **Companii:**
 - Nume: VARCHAR2(255), diferit de NULL
 - Email: VARCHAR2(255), diferit de NULL
 - Telefon: VARCHAR2(50), diferit de NULL
- **Inchirieri:**

- Data_inchirierii: DATE, diferit de NULL (data contractului)
- Data_start: DATE, diferit de NULL
- Data_final: DATE, diferit de NULL
- **Metoda_de_Plata:**
 - Metoda_Plata: VARCHAR2(50), diferit de NULL (valori: cash, card, transfer bancar)
- **Plati:**
 - Suma_de_Plata: NUMBER(12,2), diferit de NULL, > 0
 - Data_plata: DATE, diferit de NULL
- **Utilizatori:**
 - Nume: VARCHAR2(100), diferit de NULL
 - Prenume: VARCHAR2(100), diferit de NULL
 - Email: VARCHAR2(255), diferit de NULL
 - Telefon: VARCHAR2(50), diferit de NULL
- **Cladiri:**
 - Denumire: VARCHAR2(200), diferit de NULL
 - Adresa: VARCHAR2(400), diferit de NULL
 - Proprietar: VARCHAR2(255), diferit de NULL
- **Birouri:**
 - Etaj: NUMBER, diferit de NULL, valori intre 1 si 10
 - Numar: NUMBER, diferit de NULL, valori intre 1 si 100
- **Lista_Facilitati:**
 - Descriere_Facilitate: VARCHAR2(200), diferit de NULL (ex.: aparat de cafea, proiector, TV, microfoane profesionale, imprimantă etc.)
- **Tip_Birouri:**
 - Tip: VARCHAR2(50), diferit de NULL
 - Posibile valori: Open Desk, Dedicated Desk, Private Office, Team Office, Conference Room
 - Pret_Loc: NUMBER(10,2), diferit de NULL, > 0
- **Numar_Locuri:**
 - Numar_Locuri: NUMBER, diferit de NULL, valori posibile intre 1 si 4.

Inchiriere_Birouri si Facilitati_Birouri nu au attribute, ele fiind tabele asociative.

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Diagrama conceptuala

Diagrama conceptuala:

Companii = {Companie_ID, Nume, Email, Telefon}

Inchirieri = {Inchiriere_ID, Companie_ID, Data_inchirierii, Data_start, Data_final}

Metoda_de_Plata = {Metoda_plata_ID, Metoda_Plata}

Plati = {Plata_ID, Inchiriere_ID, Companie_ID, Metoda_plata_ID, Suma_de_Plata, Data_plata}

Utilizatori = {Utilizator_ID, Nume, Prenume, Email, Telefon}

Birouri = {Birou_ID, Tip_Birouri_ID, Numar_Locuri_ID, Cladire_ID, Etaj, Numar}

Lista_Facilitati = {Facilitate_ID, Descriere_Facilitate}

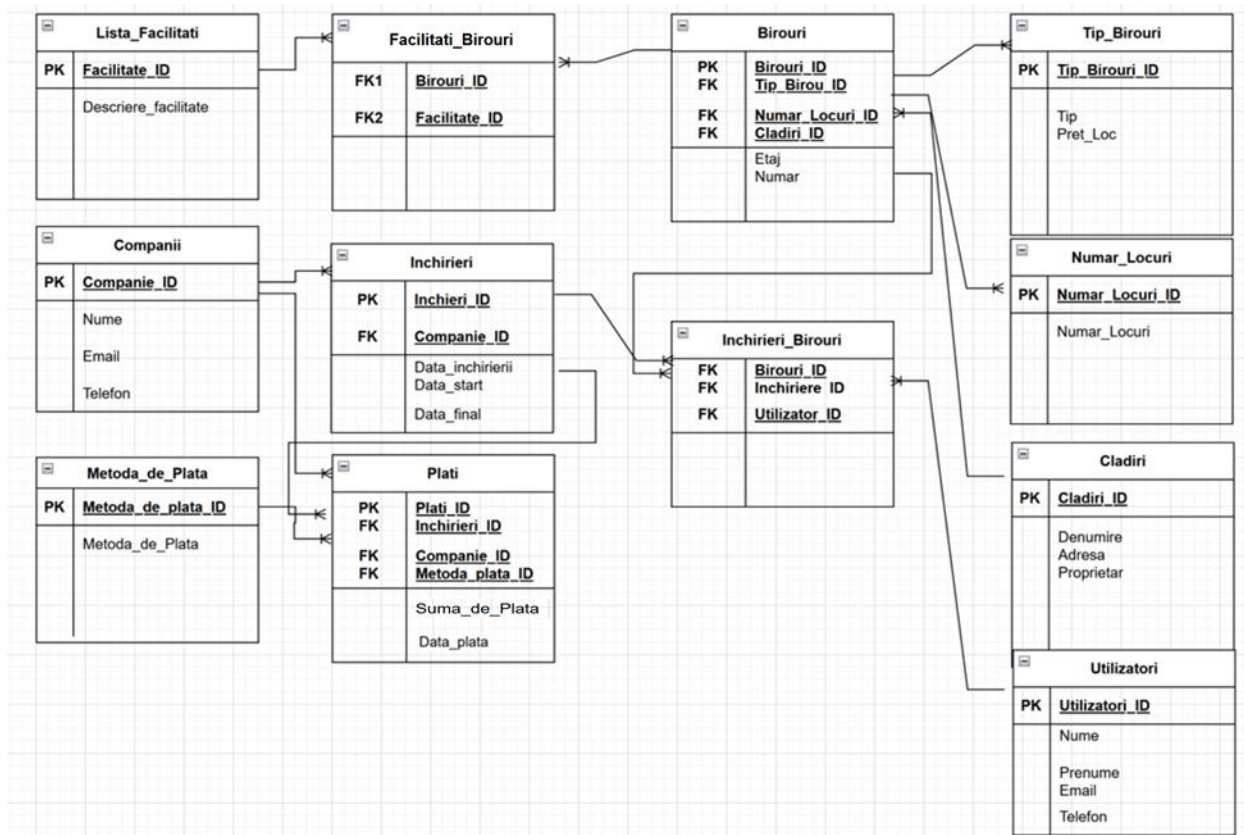
Cladiri = {Cladire_ID, Denumire, Adresa, Proprietar}

Tip_Birouri = {Tip_Birouri_ID, Tip, Pret_Loc}

Numar_Locuri = {Numar_Locuri_ID, Numar_Locuri}

Inchirieri_Birouri = {Birouri_ID, Inchirieri_ID, Utilizator_ID}

Facilitati_Birouri = {Birouri_ID, Facilitate_ID}



8. Schemele relaționale corespunzătoare diagramei conceptuale proiectate la pct 7

Schema relationala:

- a) Companii:
 - i) Companie_ID: NUMBER
 - ii) Nume: VARCHAR2(255)
 - iii) Email: VARCHAR2(255)
 - iv) Telefon: VARCHAR2(50)
- b) Inchirieri:
 - i) Inchirieri_ID: NUMBER
 - ii) Companie_ID: NUMBER
 - iii) Data_inchirierii: DATE
 - iv) Data_start: DATE
 - v) Data_final: DATE
- c) Metoda_de_Plata:
 - i) Metoda_plata_ID: NUMBER
 - ii) Metoda_de_Plata: VARCHAR2(50)
- d) Plati:

- i) Plata_ID: NUMBER
 - ii) Inchirieri_ID: NUMBER
 - iii) Companie_ID: NUMBER
 - iv) Metoda_plata_ID: NUMBER
 - v) Suma_de_Plata: NUMBER (12,2)
 - vi) Data_plata: DATE
- e) Utilizatori:
 - i) Utilizatori_ID: NUMBER
 - ii) Nume: VARCHAR2(100)
 - iii) Prenume: VARCHAR2(100)
 - iv) Email: VARCHAR2(255)
 - v) Telefon: VARCHAR2(50)
- f) Birouri:
 - i) Birou_ID: NUMBER
 - ii) Tip_Birouri_ID: NUMBER
 - iii) Numar_Locuri_ID: NUMBER
 - iv) Cladiri_ID: NUMBER
 - v) Etaj: NUMBER
 - vi) Numar: NUMBER
- g) Lista_Facilitati:
 - i) Facilitate_ID: NUMBER
 - ii) Descriere_Facilitate: VARCHAR2(200)
- h) Tip_Birouri:
 - i) Tip_Birouri_ID: NUMBER
 - ii) Tip: VARCHAR2(50)
 - iii) Pret_Loc: NUMBER(10,2)
- i) Numar_Locuri:
 - i) Numar_Locuri_ID: NUMBER
 - ii) Numar_Locuri: NUMBER
- j) Cladiri:
 - i) Cladiri_ID: NUMBER
 - ii) Denumire: VARCHAR2(200)
 - iii) Adresa: VARCHAR2(400)
 - iv) Proprietar: VARCHAR2(255)
- k) Inchirieri_Birouri:
 - i) Birou_ID: NUMBER
 - ii) Inchirieri_ID: NUMBER
 - iii) Utilizator_ID: NUMBER
- l) Facilitati_Birouri
 - i) Birou_ID: NUMBER
 - ii) Facilitate_ID: NUMBER

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

Datele Nenormalizate					
Data_inchiriere	Data_Start	Data_final	Companie	Birouri	Utilizatori
45301	45303	45305	ACME SRL, office@acme.ro, 0722-123-456	Monolith, Str Unirii 1, Globex Corp, 1 loc, Open Desk	Adriana Alex, Adriana.Alex@ acme.ro, 0722-111-222
45327	45332	45333	Beta Group, contact@beta.com, 0744-987-654	Tech Tower, Bvd Revolutiei 5, Initech, 2 locuri, Dedicated Desk	Ana Ionescu, Ana.Ionescu@ beta.com, 0744-222-333

Forma normala 3 (FN 3= FN2 + fara dependinte functionale)

Companie_ID	Nume Companie	email companie	Tel companie	Utilizator ID	Nume Utilizator	Prenume Utilizator	emai utilizator	telefon utilizator
1	ACME SRL	office@acme.ro	0722-123-456	1	Alex	Adriana	adriana.alex@acme.ro	0722-111-222
2	Beta Group	contact@beta.com	0744-987-654	2	Ionescu	Ana	ana.ionescu@beta.com	0744-222-333

Inchiriere_ID	Companie_ID	Data_inchiriere	Data_Start	Data_final
1	1	1/10/2024	1/12/2024	1/14/2024
2	2	2/5/2024	2/10/2024	2/11/2024

Cladire_ID	Denumire	Adresa	Proprietar
1	Monolith	Str Unirii 1	Globex Corp
2	Tech Tower	Bvd Revolut	Initech

Tip_Birouri_ID	Tip_Birouri	Pret_Loc	Numar_Locuri_ID	Numar_Locuri
1	Open Desk	20	1	1
2	Dedicated Desk	35	2	2
3	Private Office	75	3	3
4	Team Office	150	4	4
5	Conference Room	300	5	4
			6	3

Pentru a lega toate informatiile avem nevoie de tabele de asociere

Inchiriere_ID	Birou_ID	Utilizator_ID
1	101	1
2	102	2

Tip_Birouri_ID	Numar_Locuri_ID	Cladiri_ID	Etaj
1	1	1	1
2	2	1	1

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

- Instalez Oracle de pe link-ul din fisier.
- Se deschide DataGrip si se creeaza un nou proiect
- Se creaza un nou data source oracle, conectat cu baza de date deja instalata
- Se alege Host: localhost, Port 1521, SID: XE si ne autentificam cu userul si parola specifica serverului oracle deja instalat, Verificam conectiunea pe "Test Connection" si daca este "Succesful" apasam pe OK.
- Apasam pe "+" si cream "query console" si de aici incepem introducerea codului de mai jos din fisierul **144_Calomfirescu_Victor – Creare_Inserare.txt**.

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea

Crearea efectiva si inserarea datelor conform fisierului 144_Calomfirescu_Victor - Creare_Inserare.txt

(minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

CLADIRI_ID	DENUMIRE	ADRESA	PROPRIETAR
1	1 Monolith	Str. Unirii 1	Globex Corp
2	2 Tech Tower	Bvd. Revoluției 5	Initech
3	3 Innovation Hub	Str. Libertății 10	Umbrella Corp
4	4 Enterprise Center	Str. Căii Victoriei 15	Wayne Enterprises
5	5 Summit Plaza	Bvd. Aviatorilor 99	Stark Tower Inc

BIROU_ID	TIP_BIROURI_ID	NUMAR_LOCURI_ID	CLADIRI_ID	ETAJ	NUMAR
1	1	1	1	1	10
2	2	2	2	1	20
3	3	3	3	1	30
4	4	3	4	1	2
5	5	5	4	1	3
6	6	4	4	2	1
7	7	4	4	2	2
8	8	2	2	2	2
9	9	5	4	3	1
10	10	3	3	1	32

COMPANIE_ID	NUME	EMAIL	TELEFON
1	1 ACME SRL	office@acme.ro	0722-123-456
2	2 Beta Group	contact@beta.com	0744-987-654
3	3 Cobalt SA	info@cobalt.biz	0733-111-222
4	4 Delta Consulting	hello@delta.eu	0755-333-444
5	5 Omega Solutions	info@omega.ro	0799-111-222
6	6 Gamma LLC	contact@gamma.com	0788-333-444
7	21 Extra Solving (Inactive)	extra@solving.com	0722-555-787

BIROU_ID	FACILITATE_ID
1	4
2	4
3	5
4	5
5	5
6	5
7	5
8	6
9	6
10	6
11	6
12	6
13	6
14	6
15	6
16	6
17	6
18	6
19	6
20	6
21	6
22	7
23	7
24	7
25	7
26	7
27	8
28	8
29	9

INCHIRIERE_ID	COMPANIE_ID	DATA_INCHIRIERII	DATA_START	DATA_FINAL
1	1	2024-01-10	2024-01-12	2024-01-14
2	2	2024-02-05	2024-02-10	2024-02-11
3	3	2024-03-15	2024-03-20	2024-03-22
4	4	2024-04-01	2024-04-03	2024-04-05
5	5	2024-05-05	2024-05-07	2024-05-08
6	6	2024-06-01	2024-06-05	2024-06-06
7	7	2024-06-01	2024-06-02	2024-06-03
8	8	2024-07-01	2024-07-02	2024-07-03
9	9	2024-08-15	2024-08-16	2024-08-18

INCHIRIERE_ID	BIROU_ID	UTILIZATOR_ID
1	1	1
2	1	2
3	2	3
4	2	4
5	3	1
6	3	2
7	4	1
8	4	2
9	5	2
10	5	3
11	6	1
12	6	6
13	7	2
14	7	5

FACILITATE_ID	DESCRIERE_FACILITATE
1	Espressor de cafea
2	Imprimantă
3	TV
4	Proiector
5	Microfoane profesionale
6	Tablă smart
7	Aparatură de conferință video
8	Parcare
9	Serviciu de recepție

NUMAR_LOCURI_ID	NUMAR_LOCURI
1	1
2	2
3	3
4	4

PLATI_ID	INCHIRIERE_ID	COMPANIE_ID	METODA_PLATA_ID	SUMA_DE_PLATA	DATA_PLATA
1	1	1	2	20.00	2024-01-13
2	2	1	1	10.00	2024-01-14
3	3	2	3	120.00	2024-02-10
4	4	3	3	200.00	2024-03-21
5	5	4	1	150.00	2024-04-04
6	6	5	4	300.00	2024-05-07
7	7	6	6	50.00	2024-07-02
8	8	6	6	55.00	2024-07-03
9	9	7	5	100.00	2024-08-16
10	10	7	5	90.00	2024-08-17
11	11	7	5	120.00	2024-08-18

console_5 PLATI TIP_BIROURI console_6 console_7 console_8

Tx: Manual ✓ DDL Q

WHERE ORDER BY

	TIP_BIROURI_ID	TIP	PRET_LOC
1	1	Birou deschis	20.00
2	2	Birou dedicat	35.00
3	3	Birou privat	75.00
4	4	Birou pentru echipă	150.00
5	5	Sală de conferințe	300.00

console_5 PLATI TIP_BIROURI UTILIZATORI console_6 console_7 console_8 console_9

Tx: Manual ✓ DDL Q CSV

WHERE ORDER BY

	UTILIZATOR_ID	NUME	PRENUME	EMAIL	TELEFON
1	1	Alex	Adriana	adriana.alex@acme.ro	0722-111-222
2	2	Ionescu	Ana	ana.ionescu@beta.com	0744-222-333
3	3	Marin	George	george.marin@cobalt.biz	0733-333-444
4	4	Dumitrescu	Ioana	ioana.dumitrescu@delta.eu	0755-444-555
5	5	Vasilescu	Mihai	mihai.vasilescu@acme.ro	0722-555-666
6	6	Petrescu	Andrei	andrei.petrescu@example.com	0766-234-567
7	7	Stan	Alexandra	alexandra.stan@example.com	0766-345-678

console_5 METODA_DE_PLATA console_6 console_7

Tx: Manual ✓ DDL Q

WHERE ORDER BY

	METODA_PLATA_ID	METODA_PLATA
1	1	cash
2	2	card
3	3	transfer bancar

12. Cereri SQL Complexe

/*Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

- subcereri sincronizate în care intervin cel puțin 3 tabele
- subcereri nesincronizate în clauza FROM
- grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING)
- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
- utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
- utilizarea a cel puțin 1 bloc de cerere (clauza WITH)*/

Conform fisierului 144_Calomfirescu_Victor – exemple 12-15.txt

Cerere 1. Subcerere corelată pe 3 tabele. RESPECTA PUNCTUL a)

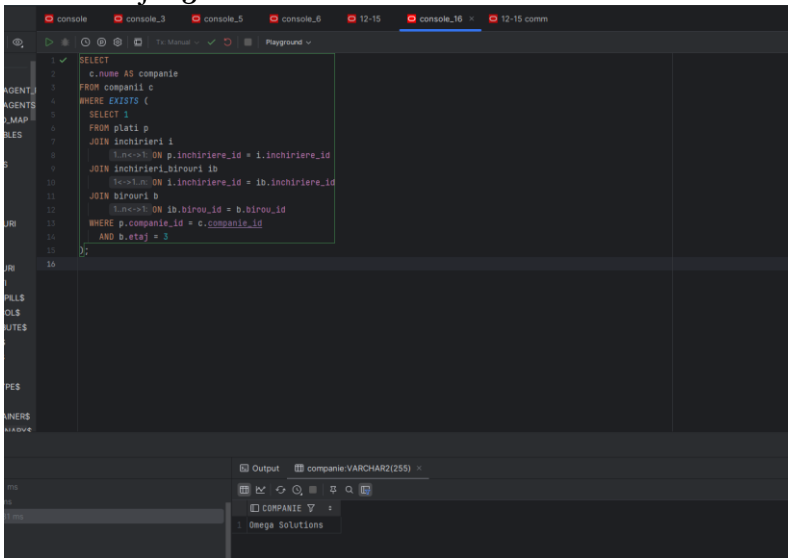
/*

Descriere: Afișează numele companiilor care au făcut cel puțin o plată pentru un birou situat la etajul 3

Acest cod rezolva cerinta a)

*/

```
SELECT
  c.nume AS companie
FROM companii c
WHERE EXISTS (
  SELECT 1
  FROM plati p
  JOIN inchirieri i
    ON p.inchiriere_id = i.inchiriere_id
  JOIN inchirieri_birouri ib
    ON i.inchiriere_id = ib.inchiriere_id
  JOIN birouri b
    ON ib.birou_id = b.birou_id
  WHERE p.companie_id = c.companie_id
  AND b.etaj = 3
);
```



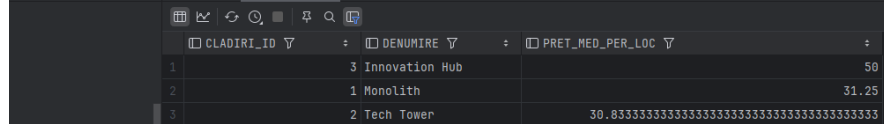
Cerere 2. Subcereri nesincronizate în FROM . RESPECTA PUNCTUL b)

/*

Descriere: Calculează pentru fiecare clădire prețul mediu per loc și afișează lista de clădiri cu acest preț, sortată descrescător.

$$* /$$

) stats



Cerere 3. Grupări, funcții grup și HAVING cu subcerere nesincronizată RESPECTA PUNCTUL c)

/* =====

Descriere: Afișează companiile și totalul cheltuielilor lor,
dar numai pe acelea al căror total depășește
media totală a cheltuielilor tuturor companiilor.

Acest cod rezolva cerinta c)

*/

```
SELECT
  c.companie_id,
  c.nume,
  SUM(p.suma_de_plata) AS total_incasari
FROM companii c
JOIN plati p
  ON c.companie_id = p.companie_id
GROUP BY c.companie_id, c.nume
HAVING SUM(p.suma_de_plata) > (
  SELECT AVG(sub.total)
    FROM (
      SELECT SUM(p2.suma_de_plata) AS total
        FROM plati p2
       GROUP BY p2.companie_id
    ) sub
)
ORDER BY total_incasari DESC
;
```

The screenshot shows a SQL IDE with a query editor and a results panel. The query is as follows:

```

1 SELECT
2   c.companie_id,
3   c.nume,
4   SUM(p.suma_de_plata) AS total_incasari
5 FROM companii c
6 JOIN plati p
7   ON c.companie_id = p.companie_id
8 GROUP BY c.companie_id, c.nume
9 HAVING SUM(p.suma_de_plata) > (
10  SELECT AVG(sub.total)
11  FROM (
12    SELECT SUM(p2.suma_de_plata) AS total
13    FROM plati p2
14    GROUP BY p2.companie_id
15  ) sub
16 )
17 ORDER BY total_incasari DESC
18 ;
19

```

The results panel shows the following data:

COMPANIE_ID	NUME	TOTAL_INCASARI
5	Omega Solutions	310
4	Delta Consulting	300

Cerere 4. ORDER BY, COALESCE (NVL) și DECODE (expresie CASE) RESPECTA PUNCTUL d)

/* =====

Descriere: Afişează fiecare plată cu ID-ul, numele companiei, suma (o dacă e NULL), descrierea metodei de plată prin DECODE și data plății, ordonate descrescător după dată.

Acest cod rezolva cerinta d)

*/

```

SELECT
  p.plati_id,
  c.nume AS companie,
  NVL(p.suma_de_plata, 0) AS suma,
  DECODE(
    mp.metoda_plata,
    'cash', 'Plata numerar',
    'card', 'Plata cu card',
    'transfer bancar', 'Plata prin transfer',
    'Altă metodă'
  ) AS metoda_prietenoasa,
  p.data_plata
FROM plati p

```

```

JOIN companii c
  ON p.companie_id = c.companie_id
JOIN metoda_de_plata mp
  ON p.metoda_plata_id = mp.metoda_plata_id
ORDER BY p.data_plata DESC
;

```

The screenshot shows a SQL playground with a query editor on the left and a results table on the right. The query is as follows:

```

2  p.plati_id,
3  c.nume AS companie,
4  NVL(p.suma_de_plata, 0) AS suma,
5  DECODE(
6    mp.metoda_plata,
7    'cash', 'Plata numerar',
8    'card', 'Plata cu card',
9    'transfer bancar', 'Plata prin transfer',
10   'Altă metodă'
11  ) AS metoda_prietenosă,
12  p.data_plata
13 FROM plati p
14 JOIN companii c
15   ON p.companie_id = c.companie_id
16 JOIN metoda_de_plata mp
17   ON p.metoda_plata_id = mp.metoda_plata_id
18 ORDER BY p.data_plata DESC

```

The results table, titled 'Result 8', contains 11 rows of data. The columns are: PLATI_ID, COMPANIE, SUMA, METODA_PRIETENOASA, and DATA_PLATA.

PLATI_ID	COMPANIE	SUMA	METODA_PRIETENOASA	DATA_PLATA
11	Omega Solutions	120	Plata numerar	2024-08-18
10	Omega Solutions	90	Plata cu card	2024-08-17
9	Omega Solutions	100	Plata prin transfer	2024-08-16
8	Gamma LLC	55	Plata cu card	2024-07-03
7	Gamma LLC	50	Plata numerar	2024-07-02
6	Delta Consulting	300	Plata numerar	2024-05-07
5	ACME SRL	150	Plata cu card	2024-04-04
4	Cobalt SA	200	Plata cu card	2024-03-21
3	Beta Group	120	Plata prin transfer	2024-02-10
2	ACME SRL	10	Plata numerar	2024-01-14
1	ACME SRL	20	Plata cu card	2024-01-13

Cerere 5. Funcții pe șiruri și date, expresie CASE, bloc WITH. RESPECTA PUNCTELE e) și f)

/* =====

Descriere: Pentru fiecare birou, află data ultimei
închirieri, numărul de zile scurse de atunci
și clasifică-l ('Recent', 'Moderate', 'Vechi'),
afișând și un cod și un tip prescurtat.

Acest cod rezolva cerinta e) și f)

*/

```

WITH ultimele_inchirieri AS (
    SELECT
        b.birou_id,
        MAX(i.data_final) AS data_ultima
    FROM birouri b
    JOIN inchirieri_birouri ib
        ON b.birou_id = ib.birou_id
    JOIN inchirieri i
        ON ib.inchiriere_id = i.inchiriere_id
    GROUP BY b.birou_id
)
SELECT
    'Birou ' || TO_CHAR(b.birou_id) AS cod_birou, -- funcție pe șir
    UPPER(SUBSTR(t.tip,1,3)) AS tip_scurt, -- funcții pe șir
    ui.data_ultima AS data_final,
    TRUNC(SYSDATE - ui.data_ultima) AS zile_trecute, -- funcții pe date
    CASE
        -- expresie CASE
        WHEN SYSDATE - ui.data_ultima < 300 THEN 'Recent'
        WHEN SYSDATE - ui.data_ultima <= 400 THEN 'Moderate'
        ELSE 'Vechi'
    END AS categorie
FROM birouri b
JOIN ultimele_inchirieri ui
    ON b.birou_id = ui.birou_id
JOIN tip_birouri t
    ON b.tip_birouri_id = t.tip_birouri_id
ORDER BY zile_trecute DESC
;
SELECT
    b.Birou_ID,
    'Etaj ' || b.Etaj || ' #' || b.Numar AS DescriereBirou,
    ( -- subcerere corelată
        SELECT COUNT(*)
        FROM Plati p
        JOIN Inchirieri_birouri ib ON p.Inchiriere_ID = ib.Inchiriere_ID
        WHERE ib.Birou_ID = b.Birou_ID
        AND strftime('%Y', p.Data_plata) = '2024'
    ) AS NrPlati2024
FROM Birouri b;

```

```

1 WITH ultimele_inchirieri AS (
2     b.birou_id,
3     MAX(i.data_final) AS data_ultima
4 FROM birouri b
5 JOIN inchirieri_birouri ib
6     ON b.birou_id = ib.birou_id
7 JOIN inchirieri i
8     ON ib.inchiriere_id = i.inchiriere_id
9 GROUP BY b.birou_id
10 )
11
12 SELECT
13     'Birou ' || TO_CHAR(b.birou_id) AS cod_birou,
14     UPPER(SUBSTR(t.tip,1,3)) AS tip_scurt,
15     ui.data_ultima AS data_final,
16     TRUNC(SYSDATE - ui.data_ultima) AS zile_trecute,
17     CASE
18         WHEN SYSDATE - ui.data_ultima < 300 THEN 'Recent'
19         WHEN SYSDATE - ui.data_ultima <= 400 THEN 'Moderate'
20         ELSE 'Vechi'
21     END AS categorie
22 FROM birouri b
23 JOIN ultimele_inchirieri ui
24     ON b.birou_id = ui.birou_id
25 JOIN tip_birouri t
26     ON b.tip_birouri_id = t.tip_birouri_id
27 ORDER BY zile_trecute DESC
28 ;

```

COD_BIROU	TIP_SCURT	DATA_FINAL	ZILE_TRECUTE	CATEGORIE
Birou 4	BIR	2024-02-11	487	Vechi
Birou 3	BIR	2024-05-08	400	Vechi
Birou 5	SAL	2024-06-03	374	Moderate
Birou 2	BIR	2024-06-03	374	Moderate
Birou 1	BIR	2024-06-06	371	Moderate

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri

/* -----

Operațiunea 1: Marcarea companiilor inactive - UPDATE cu subcerere corelată (EXISTS)

Aflăm companiile care nu au niciună plată înregistrată și adăugăm sufixul " (Inactive)" la câmpul Nume.

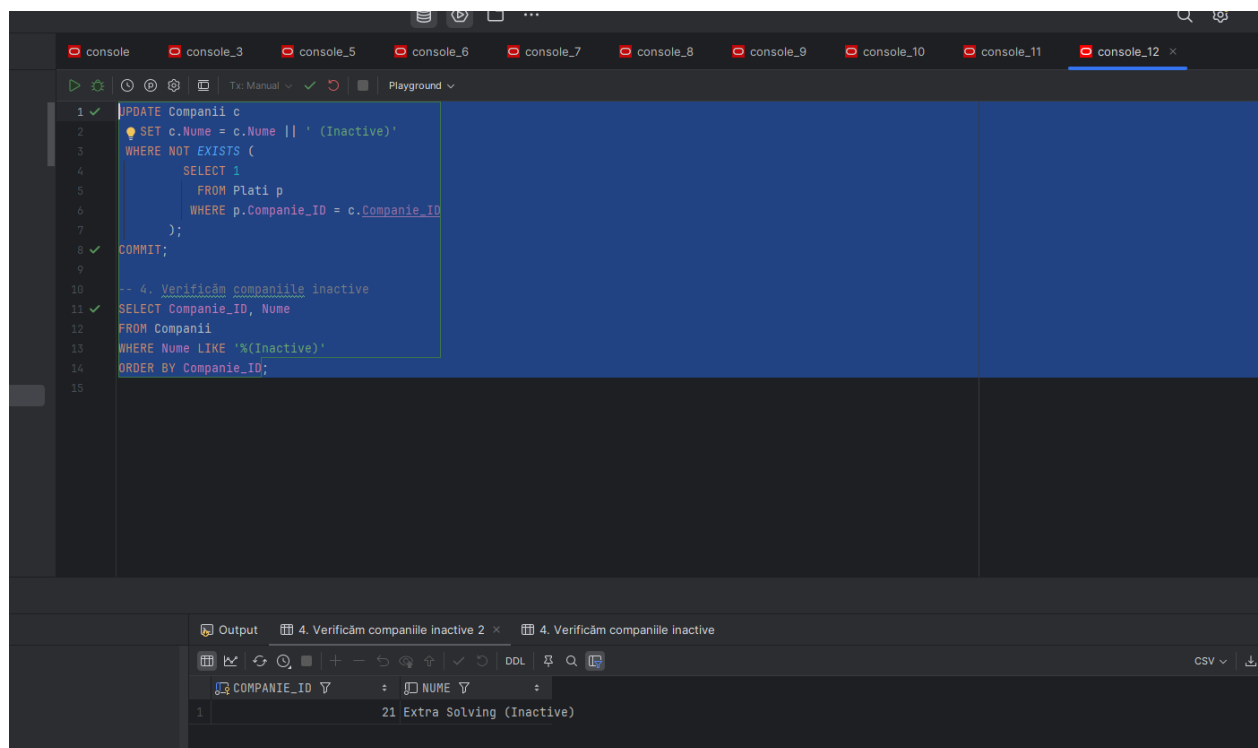
----- */

```

UPDATE Companii c
SET c.Nume = c.Nume || ' (Inactive)'
WHERE NOT EXISTS (
    SELECT 1
    FROM Plati p
    WHERE p.Companie_ID = c.Companie_ID
);
COMMIT;

```

```
-- 4. Verificăm companiile inactive
SELECT Companie_ID, Nume
FROM Companii
WHERE Nume LIKE '%(Inactive)'
ORDER BY Companie_ID;
```



/* -----

Operațiunea 2: Majorare preț loc pentru tipuri populare. UPDATE cu subcerere nesincronizată în FROM

Pentru fiecare tip de birou care a fost închiriat de mai mult de 5 ori (adică acel tip apare în tabelul INCHIRIERI_BIROURI de mai mult de 5 ori), creștem prețul per loc cu 10%.

----- */

```
UPDATE tip_birouri t
  SET t.pret_loc = t.pret_loc * 1.10
WHERE t.tip_birouri_id IN (
  SELECT tip_birouri_id
  FROM (
    SELECT b.tip_birouri_id,
```

```

        COUNT(*) AS cnt
    FROM birouri b
    JOIN inchirieri_birouri ib
        ON b.birou_id = ib.birou_id
    GROUP BY b.tip_birouri_id
) sub
WHERE sub.cnt > 5
);
COMMIT;
SELECT
    b.birou_id,
    b.etaj,
    b.numar AS numar_birou,
    t.tip AS tip_birou,
    t.pret_loc AS tarif_per_loc
FROM birouri b
JOIN tip_birouri t
    ON b.tip_birouri_id = t.tip_birouri_id
ORDER BY b.birou_id;

```

The screenshot shows a SQL IDE with a query editor and a table view. The query editor contains the following SQL code:

```

1 UPDATE tip_birouri t
2   SET t.pret_loc = t.pret_loc * 1.10
3   WHERE t.tip_birouri_id IN (
4     SELECT tip_birouri_id
5     FROM (
6       SELECT b.tip_birouri_id,
7             COUNT(*) AS cnt
8       FROM birouri b
9       JOIN inchirieri_birouri ib
10        ON b.birou_id = ib.birou_id
11      GROUP BY b.tip_birouri_id
12    ) sub
13    WHERE sub.cnt > 5
14  );
15 COMMIT;
16
17 SELECT
18   b.birou_id,
19   b.etaj,
20   b.numar AS numar_birou,
21   t.tip AS tip_birou,
22   t.pret_loc AS tarif_per_loc
23 FROM birouri b
24 JOIN tip_birouri t
25   ON b.tip_birouri_id = t.tip_birouri_id
26 ORDER BY b.birou_id;
27

```

The table view shows the results of the query, with columns: BIROU_ID, ETAJ, NUMAR_BIROU, TIP_BIROU, and TARIF_PER_LOC. The data is as follows:

BIROU_ID	ETAJ	NUMAR_BIROU	TIP_BIROU	TARIF_PER_LOC
1	1	1	10 Birou deschis	20.00
2	1	1	20 Birou dedicat	35.00
3	2	2	30 Birou privat	75.00
4	2	2	31 Birou privat	75.00
5	3	3	40 Sală de conferințe	300.00
6	1	1	50 Birou	150.00

/* -----

Operațiunea 3: Ștergerea închirierilor vechi. DELETE cu subcerere (IN)

Ștergem toate rândurile din INCHIRIERI_BIROURI care au asociate închirieri finalizate înainte de 1 ianuarie 2024, apoi ștergem și înregistrările din INCHIRIERI.

----- */

-- 1) Ștergem asocierile birou-închiriere pentru închirieri finalizate înainte de 2024

DELETE FROM Inchirieri_birouri ib

WHERE ib.Inchiriere_ID IN (

SELECT i.Inchiriere_ID

FROM Inchirieri i

WHERE i.Data_final < DATE '2024-01-01'

);

-- 2) Arătăm starea după ștergerea asocierilor

```
SELECT *  
FROM Inchirieri_birouri  
ORDER BY Inchiriere_ID, Birou_ID, Utilizator_ID;
```

--3) Ștergem închirierile finalizate înainte de 2024

```
DELETE FROM Inchirieri i  
WHERE i.Data_final < DATE '2024-01-01';  
COMMIT;  
SELECT *  
FROM Inchirieri  
ORDER BY Inchiriere_ID;
```

The screenshot shows a SQL IDE interface with a script editor and a table view. The script editor contains the following SQL code:

```
-- 1) Ștergem asocierile birou-inchiriere pentru inchirieri finalizate înainte de 2024  
DELETE FROM Inchirieri_birouri ib  
WHERE ib.Inchiriere_ID IN (  
    SELECT i.Inchiriere_ID  
    FROM Inchirieri i  
    WHERE i.Data_final < DATE '2024-01-01'  
);  
  
-- 2) Arătăm starea după ștergerea asocierilor  
SELECT *  
FROM Inchirieri_birouri  
ORDER BY Inchiriere_ID, Birou_ID, Utilizator_ID;  
  
--3) Ștergem închirierile finalizate înainte de 2024  
DELETE FROM Inchirieri i  
WHERE i.Data_final < DATE '2024-01-01';  
COMMIT;  
SELECT *  
FROM Inchirieri  
ORDER BY Inchiriere_ID;
```

The table view shows the following data:

INCHIRIERE_ID	COMPANIE_ID	DATA_INCHIRIERII	DATA_START	DATA_FINAL
1	1	2024-01-10	2024-01-12	2024-01-14
2	2	2024-02-05	2024-02-10	2024-02-11
3	3	2024-03-15	2024-03-20	2024-03-22
4	4	2024-04-01	2024-04-03	2024-04-05
5	5	2024-05-05	2024-05-07	2024-05-08
6	1	2024-06-01	2024-06-05	2024-06-06
7	1	2024-06-01	2024-06-02	2024-06-03
8	5	2024-07-01	2024-07-02	2024-07-03
9	6	2024-08-15	2024-08-16	2024-08-18

14. Crearea unei vizualizări complexe

/*Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.*/

-- 1) Crearea unei view complexe

CREATE VIEW Vw_BirouriDetaliate AS

SELECT

b.Birou_ID,

c.Denumire AS NumeCladire,

t.Tip AS TipBirou,

nl.Numar_Locuri,

-- coloană calculată

(t.Pret_Loc * nl.Numar_Locuri) AS CostTotalLocuri

FROM Birouri b

JOIN Cladiri c ON b.Cladiri_ID = c.Cladiri_ID

JOIN Tip_Birouri t ON b.Tip_Birouri_ID = t.Tip_Birouri_ID

JOIN Numar_Locuri nl ON b.Numar_Locuri_ID = nl.Numar_Locuri_ID;

-- OPERATIE LMD permisa pe acest view

-- putem interoga (SELECT) liniștit

SELECT *

FROM Vw_BirouriDetaliate

WHERE CostTotalLocuri < 200;

-- OPERATIE LMD nepermisa

-- încercare de INSERT, va eșua

INSERT INTO Vw_BirouriDetaliate (Birou_ID, NumeCladire, TipBirou, Numar_Locuri, CostTotalLocuri)

VALUES (11, 'Skyline Plaza', 'Open Desk', 1, 20);

	BIROU_ID	NUMECLADIRE	TIPBIROU	NUMAR_LOCURI	COSTTOTALLOCURI
1	1	Monolith	Birou deschis	1	20
2	2	Monolith	Birou dedicat	2	70
3	8	Tech Tower	Birou dedicat	2	70
4	3	Monolith	Birou privat	3	225
5	4	Monolith	Birou privat	4	300
6	10	Innovation Hub	Birou privat	3	225
7	6	Tech Tower	Birou pentru echipă	4	600
8	7	Tech Tower	Birou pentru echipă	4	600
9	5	Monolith	Sală de conferințe	4	1200
10	9	Innovation Hub	Sală de conferințe	4	1200

15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația *outer-join* pe minimum 4 tabele, o cerere ce utilizează operația *division* și o cerere care implementează analiza top-*n*.

-- 1) OUTER JOIN pe minim 4 tabele
 -- Scop: afișează toate birourile, cu facilități (dacă există),
 -- tipul biroului (dacă e definit), numărul de locuri (dacă e definit)
 -- și numele clădirii (necesar, deci INNER JOIN pe clădiri).
 -- Notă: coloanele TIP_BIROURI_ID și NUMAR_LOCURI_ID din BIROURI trebuie să accepte NULL

```
SELECT
  bir.birou_id,
  cld.denumire AS nume_cladire,
  bir.etaj,
  bir.numar AS numar_birou,
  lfct.descriere_facilitate,
  tbir.tip,
  nrl.numar_locuri
```

```

FROM birouri bir
LEFT OUTER JOIN facilitati_birouri fbir
    ON fbir.birou_id = bir.birou_id
LEFT OUTER JOIN lista_facilitati lfct
    ON lfct.facilitate_id = fbir.facilitate_id
LEFT OUTER JOIN tip_birouri tbir
    ON tbir.tip_birouri_id = bir.tip_birouri_id
LEFT OUTER JOIN numar_locuri nrl
    ON nrl.numar_locuri_id = bir.numar_locuri_id
JOIN cladiri cld
    ON cld.cladiri_id = bir.cladiri_id -
ORDER BY bir.birou_id
;

```

The screenshot shows a SQL playground interface with a query editor on the left and an output table on the right. The query is a complex JOIN statement involving several tables: birouri, facilitati_birouri, lista_facilitati, tip_birouri, numar_locuri, and cladiri. The output table displays the results of the query, showing columns for BIROU_ID, NUME_CLADIRE, ETAJ, NUMAR_BIROU, DESCRIERE_FACILITATE, TIP, and NUMAR_LOCURI. The results are ordered by BIROU_ID.

Query:

```

-- 1) OUTER JOIN pe minim 4 tabele
-- Scop: afisează toate birourile, cu facilități (dacă există),
--       tipul biroului (dacă e definit), numărul de locuri (dacă e definit)
--       și numele clădirii (necesar, deci INNER JOIN pe cladiri).
-- Notă: coloanele TIP_BIROURI_ID și NUMAR_LOCURI_ID din BIROURI trebuie să accepte NULL
SELECT
    bir.birou_id,
    cld.denumire AS nume_cladire,
    bir.etaj,
    bir.numar AS numar_birou,
    lfct.descriere_facilitate,
    tbir.tip,
    nrl.numar_locuri
FROM birouri bir
LEFT OUTER JOIN facilitati_birouri fbir
    ON fbir.birou_id = bir.birou_id
LEFT OUTER JOIN lista_facilitati lfct
    ON lfct.facilitate_id = fbir.facilitate_id
LEFT OUTER JOIN tip_birouri tbir
    ON tbir.tip_birouri_id = bir.tip_birouri_id
LEFT OUTER JOIN numar_locuri nrl
    ON nrl.numar_locuri_id = bir.numar_locuri_id
JOIN cladiri cld
    ON cld.cladiri_id = bir.cladiri_id
ORDER BY bir.birou_id
;

```

Output Table:

BIROU_ID	NUME_CLADIRE	ETAJ	NUMAR_BIROU	DESCRIERE_FACILITATE	TIP	NUMAR_LOCURI
1	1 Monolith	1	10	Imprimantă	Birou deschis	
2	2 Monolith	1	20	Espresson de cafea	Birou dedicat	
3	2 Monolith	1	20	Imprimantă	Birou dedicat	
4	3 Monolith	2	20	Espresson de cafea	Birou privat	
5	3 Monolith	2			Birou privat	

40 rows

-- 2) DIVISION: tipuri de birouri care NU sunt folosite în nicio închiriere
 -- Scop: găsește toate tipurile pentru care nu există niciun birou în inchirieri_birouri
 --nu merge sa scot check-ul si nu pot introduce tipuri noi de birouri
 --dar am rulat in alt mediu fara checkuri si functioneaza

```
INSERT INTO Tip_Birouri (Tip, Pret_Loc)
VALUES ('Open Desk_1', 200000);
COMMIT;
```

```
SELECT *
FROM Tip_Birouri tb
WHERE NOT EXISTS (
  SELECT 1
  FROM Inchirieri_birouri ib
  JOIN Birouri b
    ON ib.Birou_ID = b.Birou_ID
  WHERE b.Tip_Birouri_ID = tb.Tip_Birouri_ID
)
ORDER BY tb.Tip_Birouri_ID
;
```

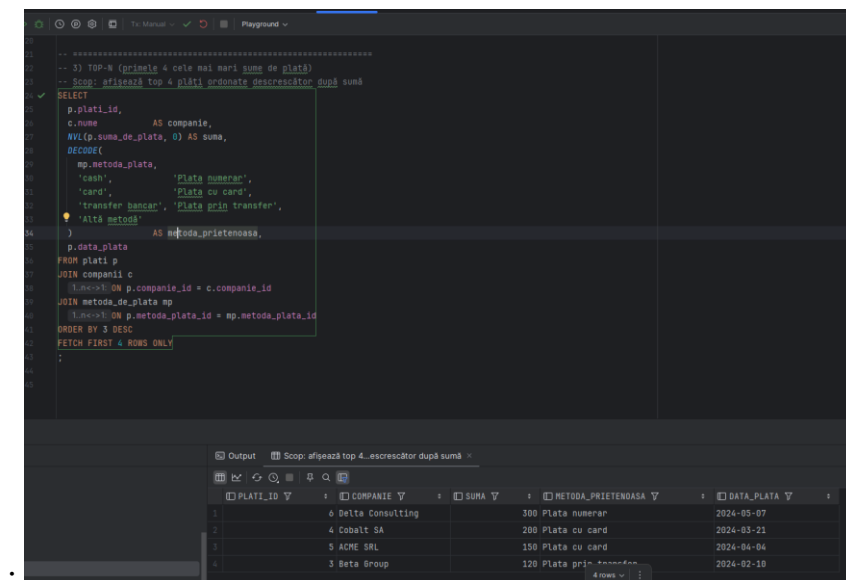
-- 3) TOP-N (primele 4 cele mai mari sume de plată)
 -- Scop: afișează top 4 plăți ordonate descrescător după sumă
 -- (se poate adapta pentru paginare cu OFFSET...FETCH NEXT)

```
SELECT
  p.plati_id,
  c.numere AS companie,
  NVL(p.suma_de_plata, 0) AS suma,
  DECODE(
```

```

mp.metoda_plata,
'cash',      'Plata numerar',
'card',      'Plata cu card',
'transfer bancar', 'Plata prin transfer',
'Altă metodă'
)          AS metoda_prietenosa,
p.data_plata
FROM plati p
JOIN companii c
  ON p.companie_id = c.companie_id
JOIN metoda_de_plata mp
  ON p.metoda_plata_id = mp.metoda_plata_id
ORDER BY 3 DESC
FETCH FIRST 4 ROWS ONLY

```



```

-- 1) Top 4 (primile 4 cele mai mari sume de plati)
-- Scop: afișează top 4 plăți ordonate descrescător după sumă
SELECT
  p.plati_id,
  c.nume          AS companie,
  NVL(p.suma_de_plata, 0) AS suma,
  DECODE(
    mp.metoda_plata,
    'cash',      'Plata numerar',
    'card',      'Plata cu card',
    'transfer bancar', 'Plata prin transfer',
    'Altă metodă'
  )          AS metoda_prietenosa,
  p.data_plata
FROM plati p
JOIN companii c
  ON p.companie_id = c.companie_id
JOIN metoda_de_plata mp
  ON p.metoda_plata_id = mp.metoda_plata_id
ORDER BY 3 DESC
FETCH FIRST 4 ROWS ONLY
;

```

PLATI_ID	COMPANIE	SUMA	METODA_PRIETENOASA	DATA_PLATA
6	Delta Consulting	300	Plata numerar	2024-05-07
4	Cobalt SA	200	Plata cu card	2024-01-21
5	ACME SRL	150	Plata cu card	2024-04-04
3	Beta Group	120	Plata prin transfer	2024-02-10