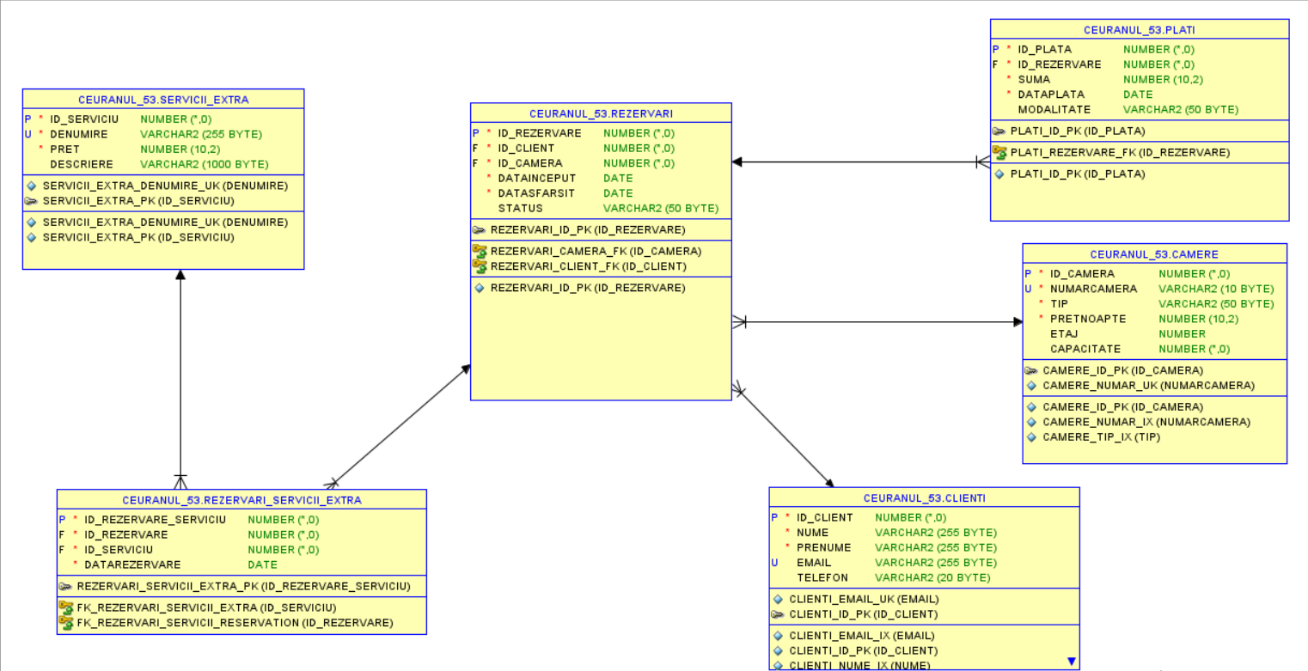
**Proiect BAZE DE DATE**

**Nota 5:**

1. Descrierea problemei economice. Schema conceptuală a bazei de date.



Proiectul este despre proiectarea și implementarea unei baze de date pentru gestiunea activității unei pensiuni.

Se consideră activitatea de evidență a funcționării unei pensiuni.

Descrierea proiectului:

Relațiile între aceste entități sunt esențiale pentru funcționalitatea sistemului nostru. În primul rând, rezervările sunt conectate cu camerele, clienții și plățile. Aceasta înseamnă că fiecare rezervare este asociată cu o cameră specifică, un client și o plată corespunzătoare. Astfel, putem urmări cu exactitate detaliile rezervărilor și să asigurăm o eficientă în gestionarea plăților.

De asemenea, avem și relația între plățile pentru servicii extra și rezervări. Aceasta înseamnă că fiecare plată pentru servicii extra este legată de o rezervare specifică și un serviciu extra oferit în cadrul acelei rezervări. Acest lucru ne permite să monitorizăm și să gestioneze cu precizie plata pentru serviciile extra oferite în timpul rezervării.

În tabela 'REZERVĂRI', avem câteva coloane importante precum: 'id\_rezervare' pentru identificarea unică a rezervării, 'client' pentru conectarea cu entitatea 'Clienți', 'camera' pentru asocierea rezervării cu o cameră specifică, 'datainceput' și 'datasfarsit' pentru a înregistra intervalul de timp al rezervării și 'status' pentru a indica dacă rezervarea este 'confirmată' sau nu.

Această bază de date ERD ne permite să gestionăm eficient rezervările, să urmărim detaliile specifice ale clienților, camerelor, plăților și serviciilor extra, precum și să oferim un sistem solid de înregistrare și monitorizare a tuturor aspectelor legate de rezervări în domeniul hospitalității.

Prin utilizarea acestei baze de date și a relațiilor între entități, suntem capabili să oferim un sistem robust și eficient de gestionare a rezervărilor, care va contribui la creșterea eficienței și a satisfacției clienților noștri.

**Nota 6:**

În acest caz, avem următoarele entități în baza de date ERD: 'Rezervări', 'Clienți', 'Camere', 'Plăți', 'Servicii Extra' și 'Plăți Servicii Extra'. Relațiile între aceste entități sunt următoarele:

* Rezervările sunt legate de camere, clienți și plăți. Asta înseamnă că fiecare rezervare este asociată cu o cameră particulară, un client și o plată corespunzătoare.
* Plățile pentru servicii extra sunt legate de rezervări și servicii extra. Asta înseamnă că fiecare plată pentru servicii extra este asociată cu o rezervare specifică și un serviciu extra oferit în cadrul acelei rezervări.

**Tabela ’CAMERE’**:

CREATE TABLE CAMERE

(

ID\_Camera INT NOT NULL,

NumarCamera VARCHAR2(10) NOT NULL,

Tip VARCHAR2(50) NOT NULL,

PretNoapte DECIMAL(10, 2) NOT NULL,

Etaj NUMBER,

Capacitate INT

);

CREATE INDEX CAMERE\_NUMAR\_IX ON CAMERE (NUMARCAMERA);

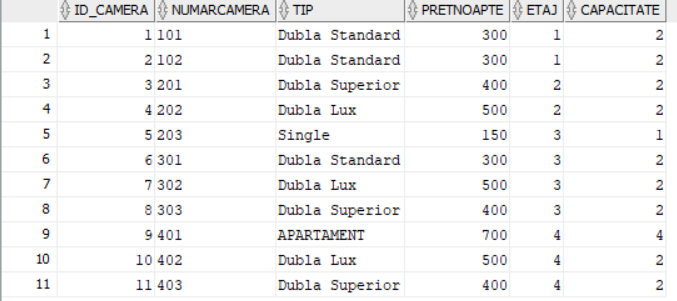
CREATE INDEX CAMERE\_TIP\_IX ON CAMERE (TIP);

ALTER TABLE CAMERE

ADD CONSTRAINT CAMERE\_ID\_PK PRIMARY KEY (ID\_Camera);

ALTER TABLE CAMERE

ADD CONSTRAINT CAMERE\_NUMAR\_UK UNIQUE (NumarCamera);



**Tabela ‘CLIENTI’**:

CREATE TABLE CLIENTI

(

ID\_Client INT NOT NULL,

Nume VARCHAR2(255),

Prenume VARCHAR2(255),

Email VARCHAR2(255),

Telefon VARCHAR2(20)

);

= Această comandă CREATE TABLE este folosită pentru a crea un tabel nou într-o bază de date SQL. Tabelul, denumit CLIENTI, va avea cinci coloane cu tipurile de date și restricțiile specificate.

CREATE INDEX CLIENTI\_NUME\_IX ON CLIENTI (NUME);

CREATE INDEX CLIENTI\_PRENUME\_IX ON CLIENTI (PRENUME);

CREATE INDEX CLIENTI\_EMAIL\_IX ON CLIENTI (EMAIL);

**ALTER TABLE** CLIENTI

CLIENTI\_ID\_PK PRIMARY KEY (ID\_Client);

1. =Această comandă adaugă o constrângere de cheie primară pentru coloana ID\_Client. Cheia primară este o restricție care asigură că fiecare valoare în coloana ID\_Client este unică și nu nulă în toate rândurile tabelului CLIENTI. Numele acestei constrângeri este CLIENTI\_ID\_PK.

**ALTER TABLE** CLIENTI

ADD CONSTRAINT CLIENTI\_EMAIL\_UK UNIQUE (EMAIL);

=Prin această comandă se adaugă o constrângere de unicitate pe coloana EMAIL, impunând ca toate valorile în coloana EMAIL să fie unice în întreg tabelul. Cu alte cuvinte, niciun client nu poate avea același email ca alt client. Numele acestei constrângeri este CLIENTI\_EMAIL\_UK.

ALTER TABLE CLIENTI

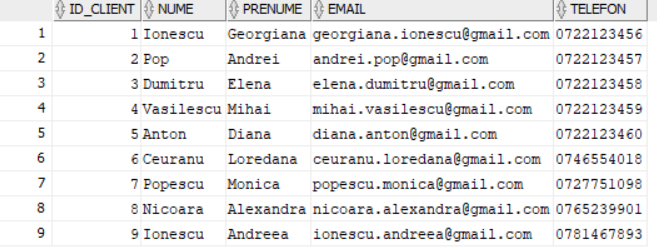
ADD CONSTRAINT CLIENTI\_NUME\_NN

CHECK ("NUME" IS NOT NULL);

ALTER TABLE CLIENTI

ADD CONSTRAINT CLIENTI\_PRENUME\_NN

CHECK ("PRENUME" IS NOT NULL);



**Daca dorim sa stergem aceasta tabela, vom folosi DROP TABLE CLIENTI CASCADE CONSTRAINTS;**

**Tabela ‘PLATI’**:

CREATE TABLE PLATI

(

ID\_Plata INT NOT NULL,

ID\_Rezervare INT NOT NULL,

Suma DECIMAL(10, 2) NOT NULL,

DataPlata DATE NOT NULL,

Modalitate VARCHAR2(50)

);

-- Se presupune că tabela REZERVARI a fost creată anterior

-- și ID\_Rezervare din PLATI se referă la ID\_Rezervare din tabelul REZERVARI

ALTER TABLE PLATI

ADD CONSTRAINT PLATI\_ID\_PK PRIMARY KEY (ID\_Plata);

ALTER TABLE PLATI

ADD CONSTRAINT PLATI\_REZERVARE\_FK FOREIGN KEY (ID\_Rezervare)

REFERENCES REZERVARI (ID\_Rezervare);

A screenshot of a computer

Description automatically generated

**Tabela ‘REZERVARI’:**

CREATE TABLE REZERVARI

(

ID\_Rezervare INT NOT NULL,

ID\_Client INT NOT NULL,

ID\_Camera INT NOT NULL,

DataInceput DATE NOT NULL,

DataSfarsit DATE NOT NULL,

Status VARCHAR2(50)

);

ALTER TABLE REZERVARI

ADD CONSTRAINT REZERVARI\_CLIENT\_FK FOREIGN KEY (ID\_Client)

REFERENCES CLIENTI (ID\_Client);

ALTER TABLE REZERVARI

ADD CONSTRAINT REZERVARI\_CAMERA\_FK FOREIGN KEY (ID\_Camera)

REFERENCES CAMERE (ID\_Camera);

A screenshot of a computer

Description automatically generated

**Tabela ‘Servicii extra’** :

CREATE TABLE SERVICII\_EXTRA

(

ID\_Serviciu INT NOT NULL,

Denumire VARCHAR2(255) NOT NULL,

Pret DECIMAL(10, 2) NOT NULL,

Descriere VARCHAR2(1000)

);

ALTER TABLE SERVICII\_EXTRA

ADD CONSTRAINT SERVICII\_EXTRA\_PK PRIMARY KEY (ID\_Serviciu);

ALTER TABLE SERVICII\_EXTRA

ADD CONSTRAINT SERVICII\_EXTRA\_DENUMIRE\_UK UNIQUE (Denumire);



**Tabela ‘Rezervari servicii extra’ :**

CREATE TABLE REZERVARI\_SERVICII\_EXTRA (

ID\_Rezervare\_Serviciu INT PRIMARY KEY,

ID\_Rezervare INT NOT NULL,

ID\_Serviciu INT NOT NULL,

DataRezervare DATE NOT NULL,

CONSTRAINT FK\_Rezervari\_Servicii\_Reservation FOREIGN KEY (ID\_Rezervare)

REFERENCES REZERVARI(ID\_Rezervare),

CONSTRAINT FK\_Rezervari\_Servicii\_Extra FOREIGN KEY (ID\_Serviciu)

REFERENCES SERVICII\_EXTRA(ID\_Serviciu)

);



**Nota 7:**

1. Adaugati un client nou in tabela ‘clienti’.

INSERT INTO CLIENTI (ID\_Client, Nume, Prenume, Email, Telefon) VALUES

(10, 'Marinescu', 'Ioana', 'marinescu.ioana@gmail.com', '0722000000');

A screenshot of a computer

Description automatically generated

2. Actualizati adresa de email a noului client.

UPDATE CLIENTI SET Email = 'ioana23marinescu@gmail.com' WHERE ID\_Client = 10;



3. Eliminati clientul cu id-ul 2.

DELETE FROM CLIENTI WHERE ID\_Client = 2;

A screenshot of a computer

Description automatically generated

4. Adaugati o Rezervarea rezervare pentru servicii extra in data de 1 mai pentru

ID-ul 5 .

INSERT INTO REZERVARI\_SERVICII\_EXTRA (ID\_Rezervare\_Serviciu, ID\_Rezervare, ID\_Serviciu, DataRezervare)

VALUES(4, 5, 1, TO\_DATE('2023-05-01', 'YYYY-MM-DD'));

A screenshot of a computer

Description automatically generated

5. Adaugati o noua rezervare in tabela ‘rezervari’, a clientului cu id-ul 10.

INSERT INTO REZERVARI (ID\_Rezervare, ID\_Client, ID\_Camera, DataInceput, DataSfarsit, Status) VALUES

(10, 10, 11, TO\_DATE('2023-07-01', 'YYYY-MM-DD'), TO\_DATE('2023-07-05', 'YYYY-MM-DD'), 'Confirmat');

A screenshot of a computer

Description automatically generated

6. Adaugati in tabela ‘plati’ clientul cu id-ul 10.

INSERT INTO PLATI (ID\_Plata, ID\_Rezervare, Suma, DataPlata, Modalitate) VALUES

(10,10, 1600.00, TO\_DATE('2023-07-01', 'YYYY-MM-DD'), 'Card');

A screenshot of a computer

Description automatically generated

7. Actualizati adresa de email a clientului nr 6 (nume student).

UPDATE CLIENTI SET Email = 'ceuranu.loredana17@gmail.com' WHERE ID\_Client = 6;

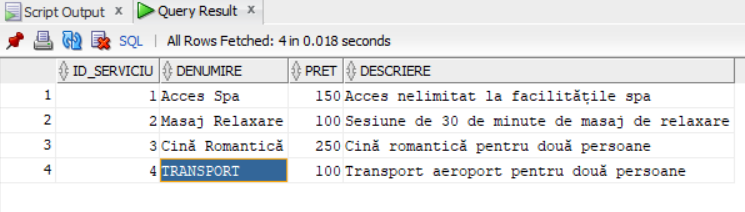
A screenshot of a computer

Description automatically generated

8.Sa se adauge in tabela ‘Servicii extra’ un nou serviciu de TRANSPORT.

INSERT INTO SERVICII\_EXTRA (ID\_Serviciu, Denumire, Pret, Descriere) VALUES

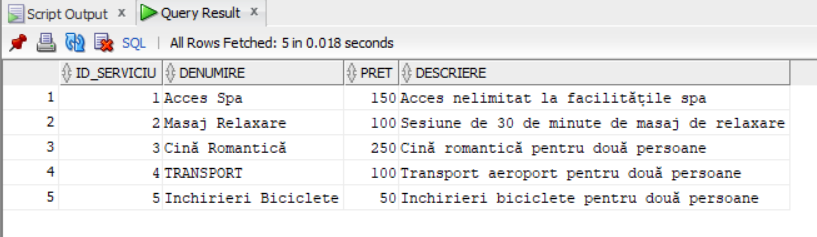
(4, 'TRANSPORT', 100.00, 'Transport aeroport pentru două persoane');



9. Sa se adauge in tabela ‘Servicii extra’ un nou serviciu de inchirieri biciclete.

INSERT INTO SERVICII\_EXTRA (ID\_Serviciu, Denumire, Pret, Descriere) VALUES

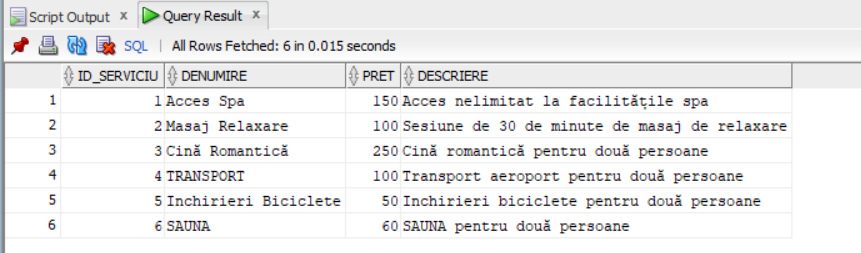
(5, 'Inchirieri Biciclete', 50.00, 'Inchirieri biciclete pentru două persoane');



10. Sa se adauge in tabela ‘Servicii extra’ un nou serviciu de ‘Sauna’.

INSERT INTO SERVICII\_EXTRA (ID\_Serviciu, Denumire, Pret, Descriere) VALUES

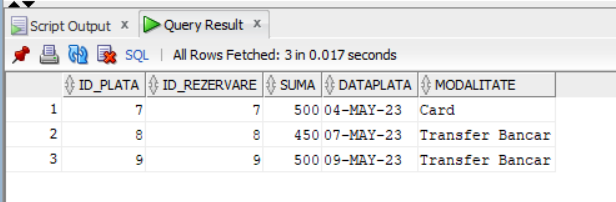
(6, ‘SAUNA’, 60.00, ‘SAUNA pentru doua persoane’);



**Nota 8:**

1.Sa se afiseze platile cu ID-urile 7,8,9.

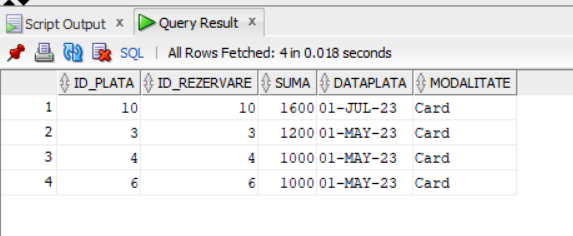
SELECT\*FROM PLATI WHERE ID\_PLATA IN (7,8,9);



2. Sa se afiseze platile cu suma mai mare sau egala cu 1000 de lei.

SELECT\* FROM PLATI

WHERE SUMA>=1000;



3.Sa se afiseze platile cu suma mai mica de 1000 de lei.

SELECT\* FROM PLATI

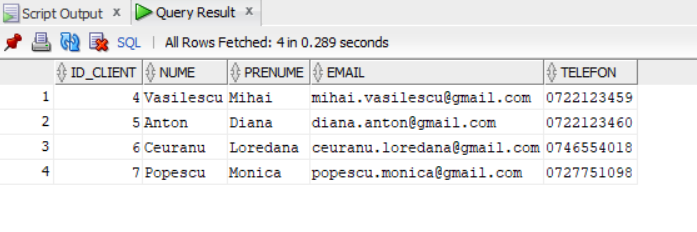
WHERE SUMA<1000;

A screenshot of a computer

Description automatically generated

4. Sa se afiseze clientii cu id-urile cuprinse intre 4 si 7.

SELECT \* FROM CLIENTI WHERE ID\_Client BETWEEN 4 AND 7;



5. Scrieți o interogare care să afișeze numărul total de rezervări pentru fiecare cameră. Includeți doar rezervările cu statutul „Confirmat” care au o dată de începere în mai 2023.

SELECT

ID\_Camera,

COUNT(\*) AS total\_reservations

FROM

REZERVARI

WHERE

Status = 'Confirmat'

AND EXTRACT(MONTH FROM DataInceput) = 5

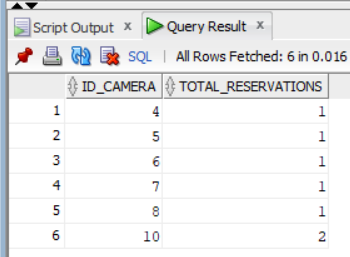
AND EXTRACT(YEAR FROM DataInceput) = 2023

GROUP BY

ID\_Camera

ORDER BY

ID\_Camera ASC;



6. Scrie o interogare care să afișeze detaliile rezervărilor care au o dată de început între 10 și 15 aprilie 2023. Include doar rezervările cu statusul 'Confirmat'.

SELECT \*FROM REZERVARI

WHERE

Status = 'Confirmat'

AND DataInceput >= TO\_DATE('2023-04-10', 'YYYY-MM-DD')

AND DataInceput <= TO\_DATE('2023-04-15', 'YYYY-MM-DD');

A screenshot of a computer

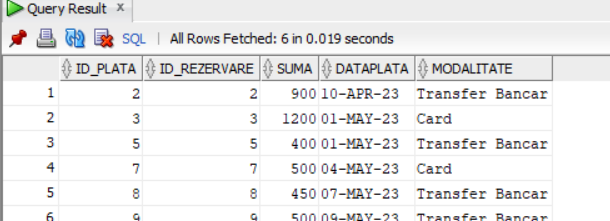
Description automatically generated

7. Sa se afiseze din tabela ‘plati’ sumele cuprinse intre 400 si 1200, mai putin sumele 600 si 1200.

SELECT \* FROM PLATI WHERE Suma BETWEEN 400 AND 1200

MINUS

SELECT \* FROM PLATI WHERE Suma IN (600.00, 1000.00);



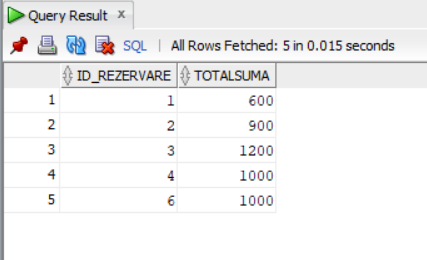
8. Selectați ID-urile rezervărilor și suma totală plătită pentru fiecare rezervare din tabelul PLATI. Grupând rezultatele după ID-urile rezervărilor, afișați doar acele înregistrări pentru care suma totală plătită este mai mare de 500.00

SELECT ID\_Rezervare, SUM(Suma) AS TotalSuma

FROM PLATI

GROUP BY ID\_Rezervare

HAVING SUM(Suma) > 500.00;



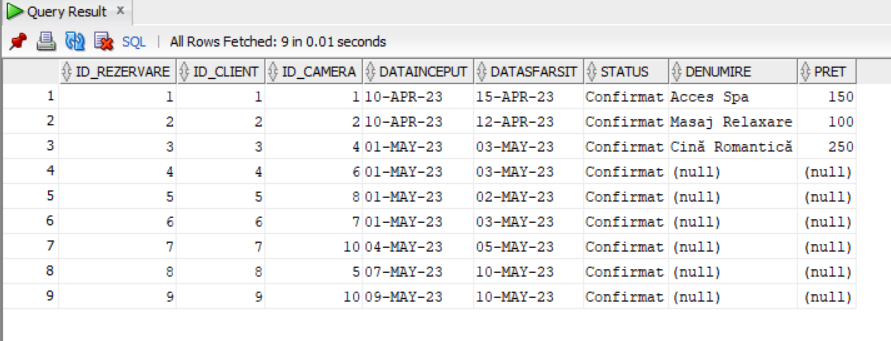
9. Selectați toate informațiile din tabelul REZERVARI (R) împreună cu denumirea și prețul fiecărui serviciu extra asociat rezervărilor, în cazul în care există. Utilizați o operație LEFT JOIN pentru a asocia rezervările cu serviciile extra, iar rezultatul să includă toate înregistrările din tabelul REZERVARI, indiferent dacă există sau nu corespondențe în tabelul REZERVARI\_SERVICII\_EXTRA sau SERVICII\_EXTRA

SELECT R.\*, SE.Denumire, SE.Pret

FROM REZERVARI R

LEFT JOIN REZERVARI\_SERVICII\_EXTRA RSE ON R.ID\_Rezervare = RSE.ID\_Rezervare

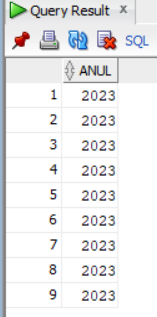
LEFT JOIN SERVICII\_EXTRA SE ON RSE.ID\_Serviciu = SE.ID\_Serviciu;



10. Utilizând funcția EXTRACT, extrage anul dintr-o dată.

SELECT EXTRACT(YEAR FROM DataInceput) AS Anul

FROM REZERVARI;



11. Afișați toate detaliile despre rezervări (REZERVARI), inclusiv informațiile despre clienți (CLIENTI), în cazul în care există o asociere între rezervări și clienți. Utilizați un LEFT JOIN pentru a include toate înregistrările din REZERVARI, indiferent dacă există sau nu o corespondență în tabelul CLIENTI. Dacă există o corespondență, să fie afișate numele și prenumele clienților pentru rezervările respective.

SELECT R.\*, C.Nume, C.Prenume

FROM REZERVARI R

LEFT JOIN CLIENTI C ON R.ID\_Client = C.ID\_Client;

A screenshot of a computer

Description automatically generated

12. Selectați numărul camerei (NumarCamera) și tipul camerei (Tip) din tabelul CAMERE. Adăugați o coloană suplimentară numită RoomCategory, care să afișeze o traducere a tipului camerei în engleză utilizând funcția DECODE. Traducerile sunt definite astfel: 'Dubla Standard' -> 'Standard Room', 'Dubla Superior' -> 'Superior Room', 'Dubla Lux' -> 'Luxury Room', 'Single' -> 'Single Room', 'APARTAMENT' -> 'Apartment'."

SELECT NumarCamera, Tip,

DECODE(Tip,

'Dubla Standard', 'Standard Room',

'Dubla Superior', 'Superior Room',

'Dubla Lux', 'Luxury Room',

'Single', 'Single Room',

'APARTAMENT', 'Apartment'

) AS TIP

FROM CAMERE;

A screenshot of a computer

Description automatically generated

13. Selectați toate înregistrările din tabelul CAMERE (c1) pentru care prețul pe noapte (PretNoapte) este mai mare decât oricare dintre prețurile pe noapte ale celorlalte camere (c2), cu excepția camerei curente. Utilizați clauza ANY împreună cu o subinterogare pentru a compara prețurile pe noapte ale diferitelor camere.

SELECT \*

FROM CAMERE c1

WHERE c1.PretNoapte > ANY (SELECT c2.PretNoapte FROM CAMERE c2 WHERE c2.ID\_Camera <> c1.ID\_Camera);

A screenshot of a computer

Description automatically generated

14. Se cere să se scrie un query SQL folosind INNER JOIN pentru a obține o listă cu detaliile rezervărilor și plăților care corespund unor perioade de suprapunere între datele de început și sfârșit ale rezervărilor și datele de plată.

SELECT

r.ID\_Rezervare,

r.ID\_Client,

r.ID\_Camera,

r.DataInceput AS DataInceputRezervare,

r.DataSfarsit AS DataSfarsitRezervare,

r.Status AS StatusRezervare,

p.ID\_Plata,

p.Suma,

p.DataPlata,

p.Modalitate

FROM REZERVARI r

INNER JOIN PLATI p ON r.ID\_Rezervare = p.ID\_Rezervare

WHERE p.DataPlata BETWEEN r.DataInceput AND r.DataSfarsit;

A screenshot of a computer

Description automatically generated

15. . Obțineți informații detaliate despre plăți și rezervări, inclusiv formatarea datelor pentru a afișa data plății (**DataPlata**) și datele de început și sfârșit ale rezervărilor (**DataInceput**, **DataSfarsit**). În plus, suma plăților (**Suma**) este formatată pentru a afișa două zecimale și formatare de mii. Rezultatele sunt obținute prin combinarea datelor din tabelele **PLATI** și **REZERVARI**, unde se potrivește **ID\_Rezervare**.

SELECT

p.ID\_Plata,

TO\_CHAR(p.DataPlata, 'DD-MON-YYYY') AS DataPlataFormatata,

TO\_CHAR(p.Suma, '999,999.99') AS SumaFormatata,

r.ID\_Rezervare,

TO\_CHAR(r.DataInceput, 'DD-MON-YYYY') AS DataInceputFormatata,

TO\_CHAR(r.DataSfarsit, 'DD-MON-YYYY') AS DataSfarsitFormatata

FROM

PLATI p

JOIN

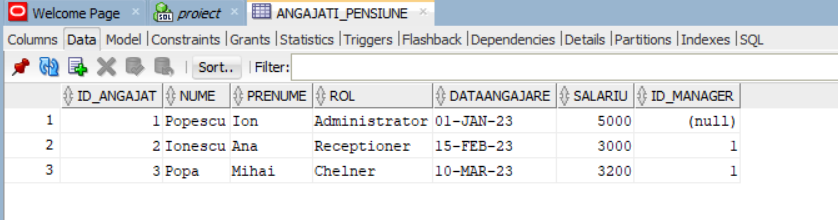
REZERVARI r ON p.ID\_Rezervare = r.ID\_Rezervare;

A screenshot of a computer

Description automatically generated

**Nota 9:**

Intai am creat tabela ‘Angajati\_Pensiune’

****

Realizează o interogare asupra tabelului 'ANGAJATI\_PENSIUNE' pentru a obține o listă ordonată a angajaților, împreună cu nivelul lor ierarhic din organizație. Începe cu angajatii care nu au un manager (adică 'ID\_Manager' este NULL), interpretându-i ca fiind la cel mai înalt nivel al ierarhiei. Folosește o interogare ierarhică pentru a conecta angajații prin verificarea dacă id-ul unui angajat ('Id\_Angajat') corespunde cu id-ul managerului altor angajați ('ID\_Manager'). În rezultate, redenumește coloana 'LEVEL' ca 'nv' și sortează lista mai întâi după nivelul ierarhic și apoi alfabetic după numele angajaților.

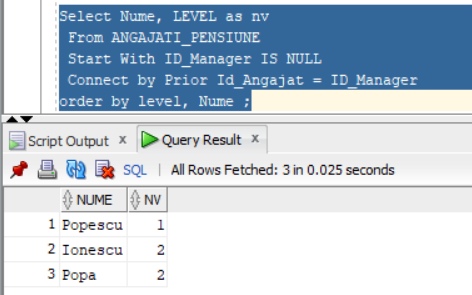
Select Nume, LEVEL as nv

From ANGAJATI\_PENSIUNE

Start With ID\_Manager IS NULL

Connect by Prior Id\_Angajat = ID\_Manager

order by level, Nume ;

****

**Nota 10:**

2. CREATE VIEW VIEW\_PLATI\_REZERVARI AS

SELECT

r.ID\_Rezervare,

r.ID\_Client,

r.ID\_Camera,

r.DataInceput,

r.DataSfarsit,

r.Status,

p.ID\_Plata,

p.Suma,

p.DataPlata AS DataPlataPlati,

p.Modalitate

FROM REZERVARI r

LEFT JOIN PLATI p ON r.ID\_Rezervare = p.ID\_Rezervare;

CREATE INDEX IDX\_VIEW\_PLATI\_REZERVARI

ON VIEW\_PLATI\_REZERVARI (ID\_Rezervare, DataPlataPlati);

CREATE SEQUENCE SEQ\_ID\_CLIENT

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

