Proiect BAZE DE DATE

PRIMA PARTE

* Codul folosit pentru crearea tabelelor:

-- Creați tabelele aferente BD de la proiect (ținând cont de corecturile efectuate asupra schemei BD).

CREATE TABLE restaurante(

id\_restaurant CHAR(10) CONSTRAINT pk\_restaurante PRIMARY KEY,

program VARCHAR2(40) NOT NULL,

capacitate NUMBER(5) CHECK (capacitate>0),

tip\_bucatarie VARCHAR2(20),

servicii VARCHAR2(100)

);

CREATE TABLE angajati\_rest(

id\_angajat CHAR(10) CONSTRAINT pk\_angajati PRIMARY KEY,

functie VARCHAR2(20) NOT NULL,

salariu NUMBER(7,2)CHECK (salariu>0)

);

CREATE TABLE locatii\_rest(

id\_locatie CHAR(10) CONSTRAINT pk\_locatii PRIMARY KEY,

id\_restaurant CHAR(10) NOT NULL,

adresa VARCHAR2(20),

telefon VARCHAR2(20),

tip\_locatie VARCHAR2(20),

CONSTRAINT fk\_locatii\_restaurante FOREIGN KEY(id\_restaurant) REFERENCES restaurante(id\_restaurant)

);

CREATE TABLE clienti\_rest(

id\_client CHAR(10) CONSTRAINT pk\_clienti PRIMARY KEY,

metoda\_plata VARCHAR2(20)NOT NULL,

cerinte VARCHAR2(50)

);

CREATE TABLE comenzi\_clienti(

id\_comanda CHAR(10) CONSTRAINT pk\_comenzi\_clienti PRIMARY KEY,

id\_client CHAR(10) NOT NULL,

status VARCHAR2(20)NOT NULL CHECK(status IN('In procesare','Aprobat','Anulat')),

data\_comenzii DATE,

CONSTRAINT fk\_comenzi\_clienti FOREIGN KEY(id\_client) REFERENCES clienti\_rest(id\_client)

);

CREATE TABLE tranzactii(

id\_tranzactie CHAR(10) CONSTRAINT pk\_tranzactii PRIMARY KEY,

id\_comanda CHAR(10) NOT NULL,

id\_client CHAR(10) NOT NULL,

data\_tranzactiei DATE,

status\_tranzactie VARCHAR2(20) NOT NULL,

taxe NUMBER(3,3),

CONSTRAINT fk\_tranzactii\_comanda FOREIGN KEY(id\_comanda) REFERENCES comenzi\_clienti(id\_comanda),

CONSTRAINT fk\_tranzactii\_client FOREIGN KEY(id\_client) REFERENCES clienti\_rest(id\_client)

);

CREATE TABLE facturi(

id\_factura CHAR(10) CONSTRAINT pk\_facturi PRIMARY KEY,

id\_comanda CHAR(10) NOT NULL,

data\_factura DATE,

total\_fact NUMBER(4,4),

metoda\_plata VARCHAR2(20),

CONSTRAINT fk\_facturi\_comanda FOREIGN KEY(id\_comanda)REFERENCES comenzi\_clienti(id\_comanda)

);

CREATE TABLE produse\_rest(

id\_produs CHAR(10) CONSTRAINT pk\_produse PRIMARY KEY,

nume\_produs VARCHAR2(50),

pret NUMBER(10,2) CHECK (pret >= 0)

);

CREATE TABLE detalii\_comenzi (

id\_comanda CHAR(10) NOT NULL,

id\_produs CHAR(10) NOT NULL,

cantitate NUMBER(10,2) CHECK (cantitate > 0),

pret\_unitar NUMBER(10,2) CHECK (pret\_unitar >= 0),

CONSTRAINT pk\_detalii\_comenzi PRIMARY KEY (id\_comanda, id\_produs),

CONSTRAINT fk\_detalii\_comenzi\_comenzi FOREIGN KEY (id\_comanda) REFERENCES comenzi\_clienti(id\_comanda),

CONSTRAINT fk\_detalii\_comenzi\_produse FOREIGN KEY (id\_produs) REFERENCES produse\_rest(id\_produs)

);

* Am observat,după realizarea diagramei,lipsa legăturilor între :tabela *restaurante* și *angajati\_res*t,între *restaurante* și *produse\_rest* și între *angajati\_rest* și *comenzi*.

ALTER TABLE angajati\_rest ADD (id\_restaurant CHAR(10));

-- Crearea unei chei externe pentru legătura cu tabela *restaurante*

ALTER TABLE angajati\_rest ADD CONSTRAINT fk\_angajati\_restaurant FOREIGN KEY (id\_restaurant) REFERENCES restaurante(id\_restaurant);

ALTER TABLE produse\_rest ADD (id\_restaurant CHAR(10));

-- Crearea unei chei externe pentru legătura cu tabela *restaurante*

ALTER TABLE produse\_rest ADD CONSTRAINT fk\_produse\_restaurant FOREIGN KEY (id\_restaurant) REFERENCES restaurante(id\_restaurant);

ALTER TABLE comenzi\_clienti ADD (id\_angajat CHAR(10));

-- Crearea unei chei externe pentru legătura cu tabela *angajati\_rest*

ALTER TABLE comenzi\_clienti ADD CONSTRAINT fk\_comenzi\_angajat FOREIGN KEY (id\_angajat) REFERENCES angajati\_rest(id\_angajat);

* Am creat modificări de cheie externă pentru a crea legături între:tabela restaurante și angajati\_rest,între restaurante și produse\_rest și între angajati\_rest și comenzi.
* Comenzile folosite pentru introducerea datelor în tabelele create anterior:

--Adăugați date în tabelele create.

1. Pentru tabela restaurante:

SELECT \* FROM restaurante

INSERT INTO restaurante VALUES

('R123','Luni-Marti 9:00-15:00',250,'orientala','catering');

INSERT INTO restaurante VALUES

('R124','Marti 19:00-21:00',400,'traditionala','evenimente private,muzica live');

INSERT INTO restaurante VALUES

('R125','Marti-Duminica 9:00-16:00',600,'frantuzeasca','fine-dining,sala de conferinte')

INSERT INTO restaurante VALUES

('R126','Joi 9:00-12:00',180,'internationala','wi-fi gratuit,take-away')

INSERT INTO restaurante VALUES

('R127','Luni-Marti 9:00-17:00',400,'italiana','promotii la deserturi')

INSERT INTO restaurante VALUES

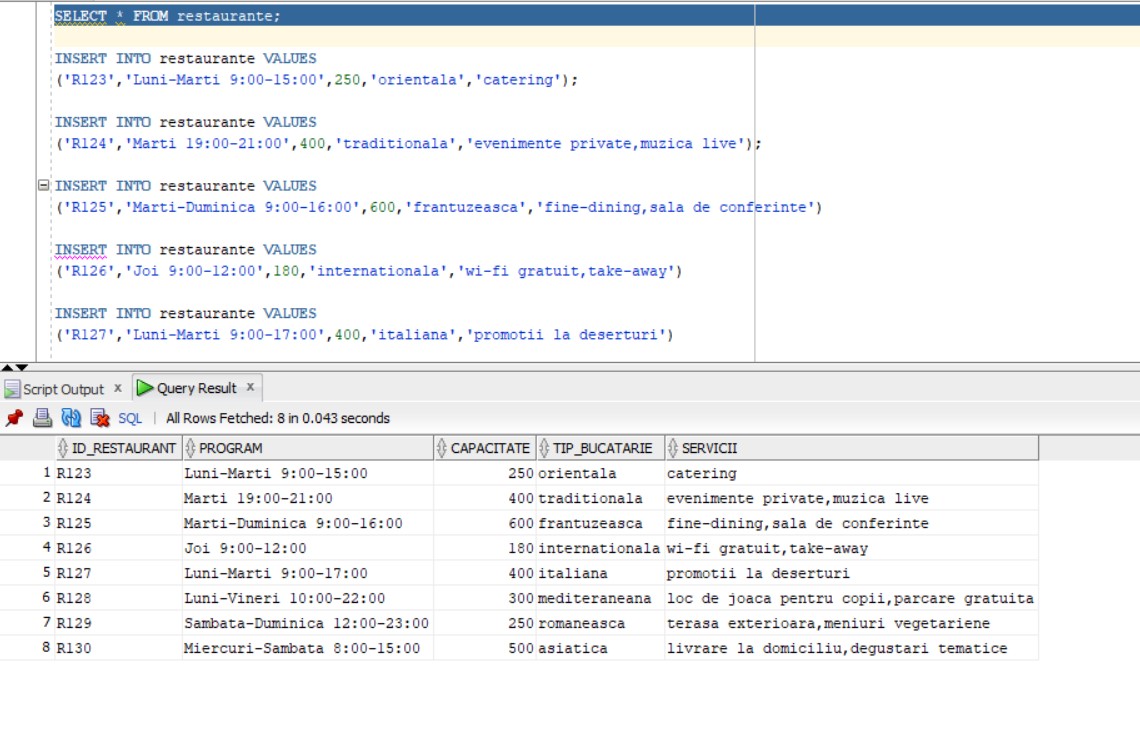
('R128','Luni-Vineri 10:00-22:00',300,'mediteraneana','loc de joaca pentru copii,parcare gratuita');

INSERT INTO restaurante VALUES

('R129','Sambata-Duminica 12:00-23:00',250,'romaneasca','terasa exterioara,meniuri vegetariene');

INSERT INTO restaurante VALUES

('R130','Miercuri-Sambata 8:00-15:00',500,'asiatica','livrare la domiciliu,degustari tematice');



2.Pentru tabela angajati\_rest:

INSERT INTO angajati\_rest VALUES

('1000','ospatar',3000);

INSERT INTO angajati\_rest VALUES

('1001','manager sala',3600);

INSERT INTO angajati\_rest VALUES

('1002','bucatar',3300);

INSERT INTO angajati\_rest VALUES

('1003','ucenic de bucatar',2700);

INSERT INTO angajati\_rest VALUES

('1004','bucatar sef',3600);

INSERT INTO angajati\_rest VALUES

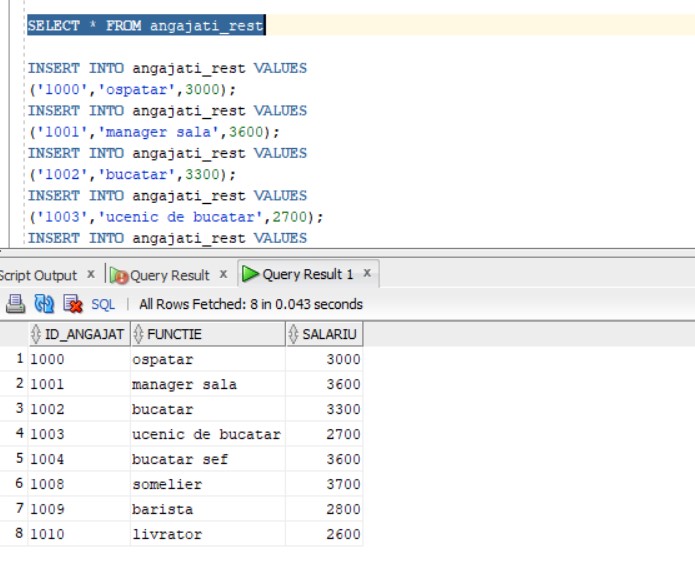
('1008', 'somelier', 3700);

INSERT INTO angajati\_rest VALUES

('1009', 'barista', 2800);

INSERT INTO angajati\_rest VALUES

('1010', 'livrator', 2600);



3.Pentru tabela locatii\_rest:

INSERT INTO locatii\_rest VALUES

('L001', 'R123', 'Str. Ion Creanga 45', '0723456789', 'urban');

INSERT INTO locatii\_rest VALUES

('L002', 'R124', 'Bd. Unirii 12', '0734567890', 'mall');

INSERT INTO locatii\_rest VALUES

('L003', 'R125', 'Calea Victoriei 23', '0745678901', 'centrul orasului');

INSERT INTO locatii\_rest VALUES

('L004', 'R126', 'Str. Primaverii 14', '0756789012', 'rural');

INSERT INTO locatii\_rest VALUES

('L005', 'R127', 'Bd. Independentei 99', '0767890123', 'suburbie');

INSERT INTO locatii\_rest VALUES

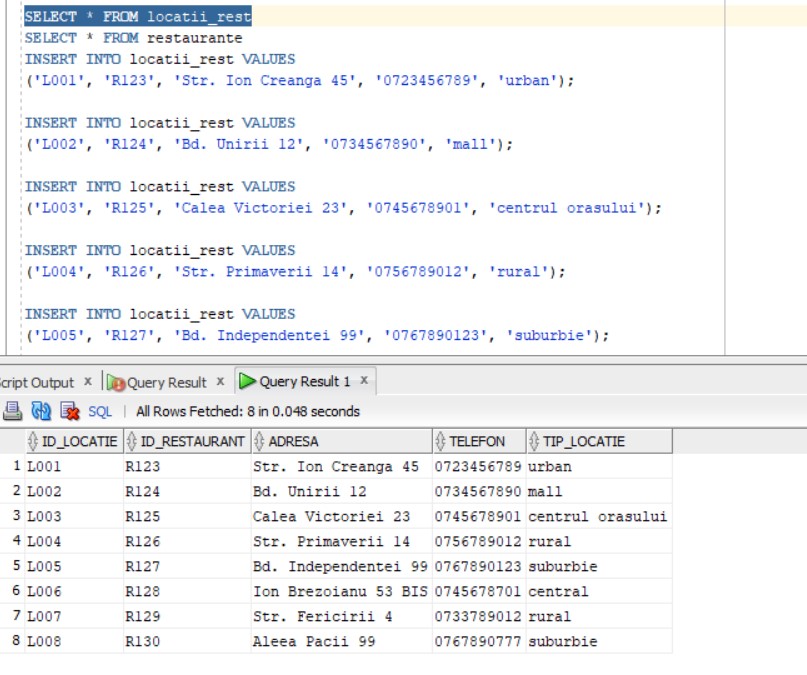
('L006', 'R128', 'Ion Brezoianu 53 BIS', '0745678701', 'central');

INSERT INTO locatii\_rest VALUES

('L007', 'R129', 'Str. Fericirii 4', '0733789012', 'rural');

INSERT INTO locatii\_rest VALUES

('L008', 'R130', 'Aleea Pacii 99', '0767890777', 'suburbie');



4.Pentru tabela clienti\_rest:

INSERT INTO clienti\_rest VALUES

('0001','cash','doua portii de papanasi');

INSERT INTO clienti\_rest VALUES

('0002','online','paine fara gluten');

INSERT INTO clienti\_rest VALUES

('0003','in avans','extra sos');

INSERT INTO clienti\_rest VALUES

('0004','cash','extra tacamuri');

INSERT INTO clienti\_rest VALUES

('0005','bon de masa','fara castraveti');

INSERT INTO clienti\_rest VALUES

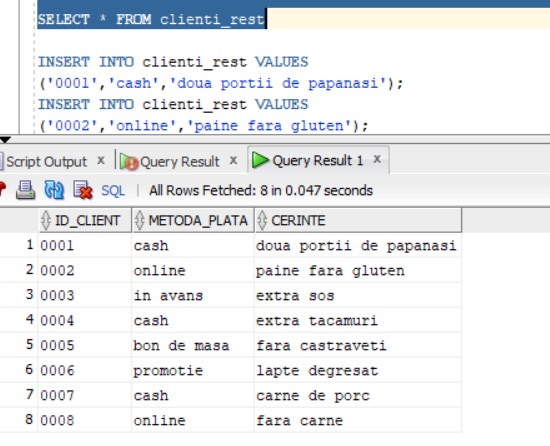
('0006', 'promotie', 'lapte degresat');

INSERT INTO clienti\_rest VALUES

('0007', 'cash', 'carne de porc');

INSERT INTO clienti\_rest VALUES

('0008', 'online','fara carne');



5.Pentru tabela comenzi\_clienti:

INSERT INTO comenzi\_clienti VALUES

('C001', '0001', 'In procesare', TO\_DATE('12/12/2021', 'DD/MM/YYYY'));

INSERT INTO comenzi\_clienti VALUES

('C002', '0002', 'In procesare', TO\_DATE('21/12/2021', 'DD/MM/YYYY'));

INSERT INTO comenzi\_clienti VALUES

('C003', '0003', 'Aprobat', TO\_DATE('11/11/2021', 'DD/MM/YYYY'));

INSERT INTO comenzi\_clienti VALUES

('C004', '0004', 'Anulat', TO\_DATE('17/05/2021', 'DD/MM/YYYY'));

INSERT INTO comenzi\_clienti VALUES

('C005', '0005', 'Aprobat', TO\_DATE('05/10/2021', 'DD/MM/YYYY'));

INSERT INTO comenzi\_clienti VALUES

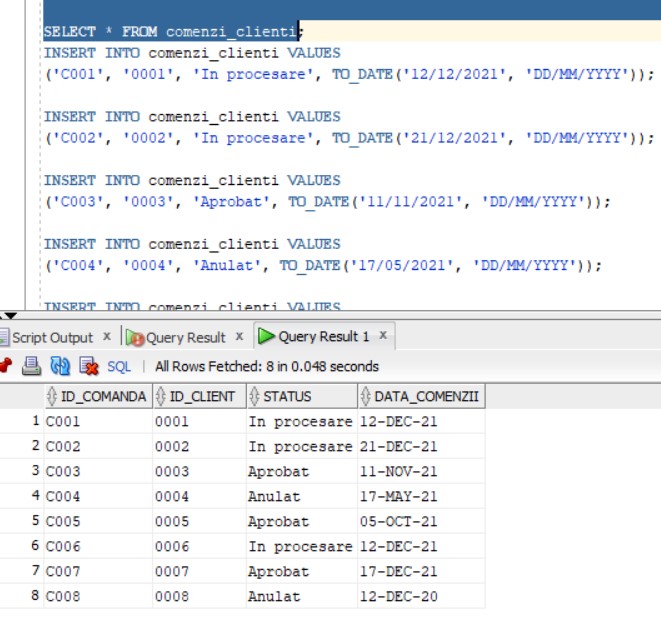
('C006', '0006', 'In procesare', TO\_DATE('12/12/2021', 'DD/MM/YYYY'));

INSERT INTO comenzi\_clienti VALUES

('C007', '0007', 'Aprobat', TO\_DATE('17/12/2021', 'DD/MM/YYYY'));

INSERT INTO comenzi\_clienti VALUES

('C008', '0008', 'Anulat', TO\_DATE('12/12/2020', 'DD/MM/YYYY'));



6.Pentru tabela tranzactii:

INSERT INTO tranzactii VALUES

('T001', 'C001', '0001', TO\_DATE('12/12/2021', 'DD/MM/YYYY'), 'Aprobat', 0.135);

INSERT INTO tranzactii VALUES

('T002', 'C002', '0002', TO\_DATE('15/12/2021', 'DD/MM/YYYY'), 'Respins', 0.150);

INSERT INTO tranzactii VALUES

('T003', 'C003', '0003', TO\_DATE('18/12/2021', 'DD/MM/YYYY'), 'Aprobat', 0.200);

INSERT INTO tranzactii VALUES

('T004', 'C004', '0004', TO\_DATE('20/12/2021', 'DD/MM/YYYY'), 'In asteptare', 0.075);

INSERT INTO tranzactii VALUES

('T005', 'C005', '0005', TO\_DATE('22/12/2021', 'DD/MM/YYYY'), 'Aprobat', 0.320);

INSERT INTO tranzactii VALUES

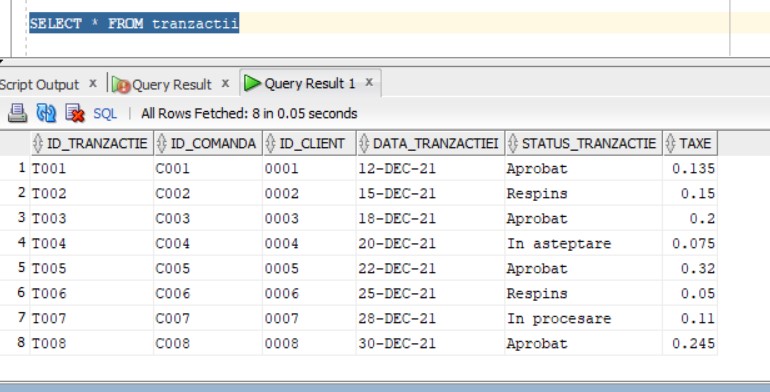
('T006', 'C006', '0006', TO\_DATE('25/12/2021', 'DD/MM/YYYY'), 'Respins', 0.050);

INSERT INTO tranzactii VALUES

('T007', 'C007', '0007', TO\_DATE('28/12/2021', 'DD/MM/YYYY'), 'In procesare', 0.110);

INSERT INTO tranzactii VALUES

('T008', 'C008', '0008', TO\_DATE('30/12/2021', 'DD/MM/YYYY'), 'Aprobat', 0.245);



7.Pentru tabela facturi:

INSERT INTO facturi VALUES

('F001','C001',TO\_DATE('12/12/2021', 'DD/MM/YYYY'),0.122,'cash');

INSERT INTO facturi VALUES

('F002','C002',TO\_DATE('15/12/2021', 'DD/MM/YYYY'),0.134,'online');

INSERT INTO facturi VALUES

('F003','C003', TO\_DATE('18/12/2021', 'DD/MM/YYYY'),0.782,'in avans');

INSERT INTO facturi VALUES

('F004','C004', TO\_DATE('20/12/2021', 'DD/MM/YYYY'),0.122,'cash');

INSERT INTO facturi VALUES

('F005','C005',TO\_DATE('22/12/2021', 'DD/MM/YYYY'),0.182,'bon de masa');

INSERT INTO facturi VALUES

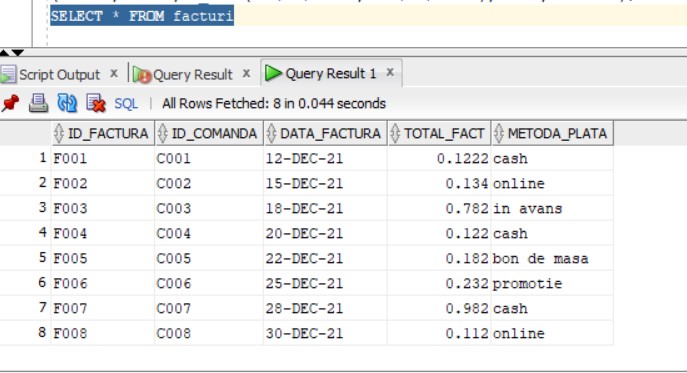
('F006','C006',TO\_DATE('25/12/2021', 'DD/MM/YYYY'),0.232,'promotie');

INSERT INTO facturi VALUES

('F007','C007',TO\_DATE('28/12/2021', 'DD/MM/YYYY'),0.982,'cash');

INSERT INTO facturi VALUES

('F008','C008',TO\_DATE('30/12/2021', 'DD/MM/YYYY'),0.112,'online');



8.Pentru tabela produse\_rest:

INSERT INTO produse\_rest VALUES

('P001', 'Pizza Margherita', 25.50);

INSERT INTO produse\_rest VALUES

('P002', 'Supa de legume', 12.30);

INSERT INTO produse\_rest VALUES

('P003', 'Lasagna Bolognese', 30.00);

INSERT INTO produse\_rest VALUES

('P004', 'Tiramisu', 15.00);

INSERT INTO produse\_rest VALUES

('P005', 'Spaghetti Carbonara', 27.50);

INSERT INTO produse\_rest VALUES

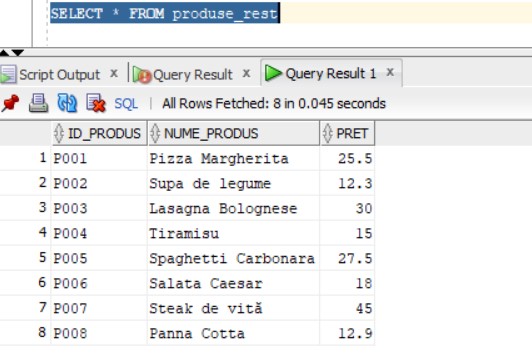
('P006', 'Salata Caesar', 18.00);

INSERT INTO produse\_rest VALUES

('P007', 'Steak de vită', 45.00);

INSERT INTO produse\_rest VALUES

('P008', 'Panna Cotta', 12.90);



9.Pentru tabela detalii\_comenzi:

INSERT INTO detalii\_comenzi VALUES

('C001', 'P001', 2, 25.50);

INSERT INTO detalii\_comenzi VALUES

('C001', 'P002', 1, 12.30);

INSERT INTO detalii\_comenzi VALUES

('C002', 'P003', 3, 30.00);

INSERT INTO detalii\_comenzi VALUES

('C002', 'P004', 2, 15.00);

INSERT INTO detalii\_comenzi VALUES

('C003', 'P005', 1, 27.50);

INSERT INTO detalii\_comenzi VALUES

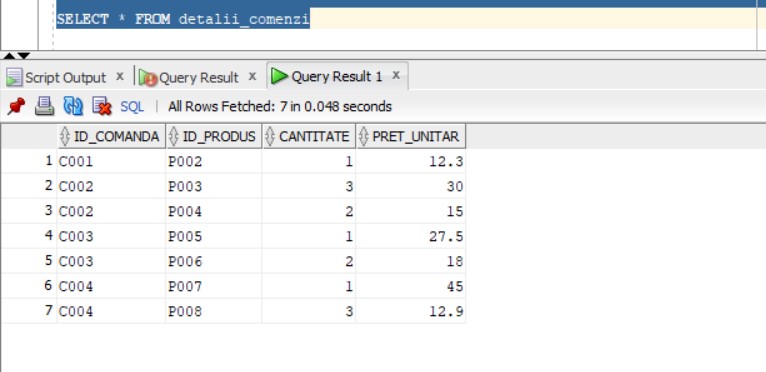
('C003', 'P006', 2, 18.00);

INSERT INTO detalii\_comenzi VALUES

('C004', 'P007', 1, 45.00);

INSERT INTO detalii\_comenzi VALUES

('C004', 'P008', 3, 12.90);



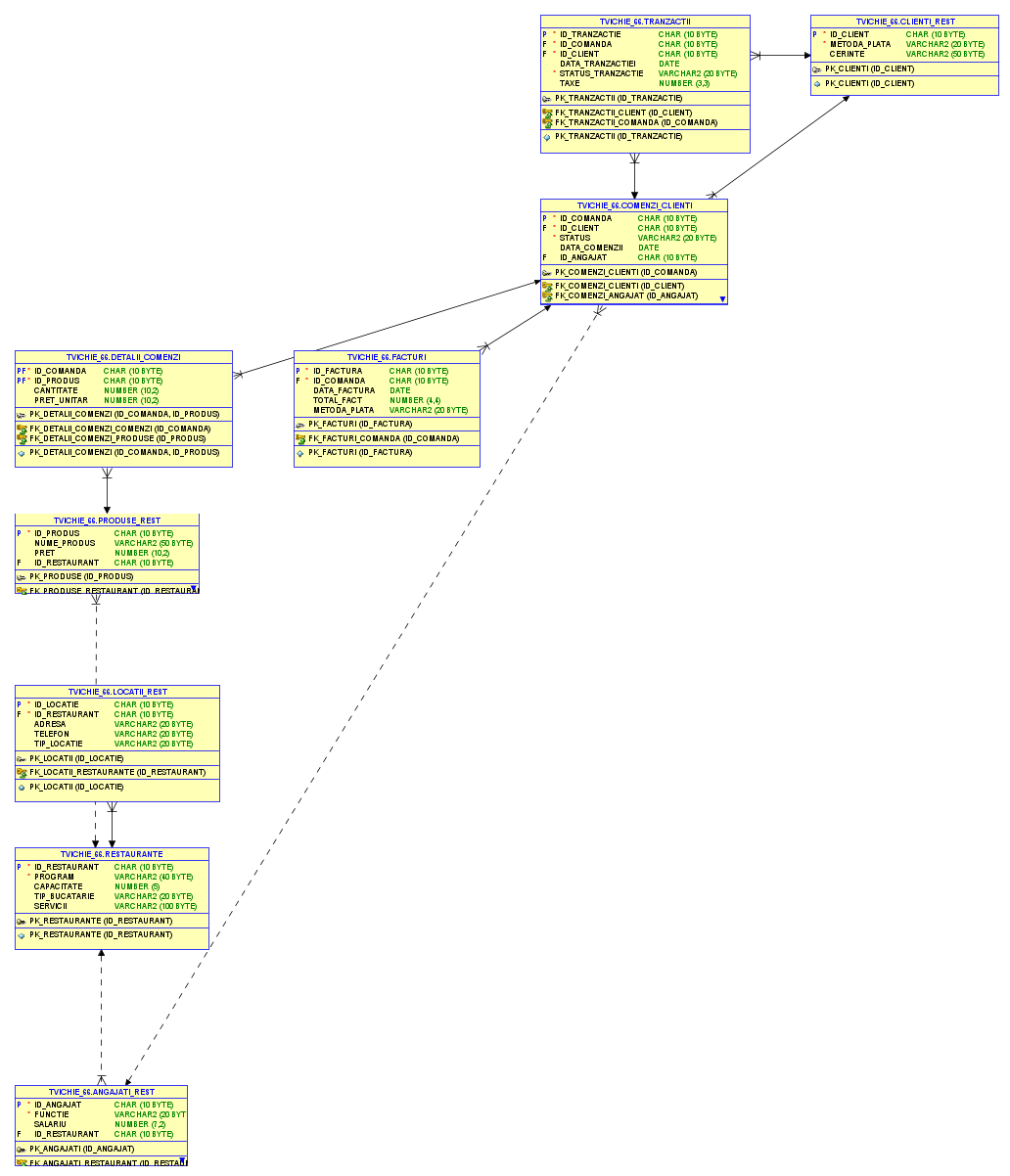
* Descrierea problemei economice alese:

Problema economică aleasă se bazează pe gestionarea eficientă a restaurantelor și a relațiilor acestora cu clienții,toate acestea fiind în contextul de afaceri din industria ospitalității.Aceasta implică aspecte,precum:organizarea fluxului de comenzi,gestionarea resurselor și a părții financiare,cât și managementul întreprinderii,împreună cu serviciile oferite clienților.

Industria ospitalității se confruntă cu diverse provocări economice, cum ar fi creșterea costurilor operaționale (de exemplu, salariile angajaților, aprovizionarea cu produse alimentare), fluctuarea cererii de servicii (în funcție de sezon sau de evenimente speciale) și cerințele clienților pentru servicii personalizate. Pentru a face față acestor provocări, restaurantele trebuie să adopte soluții eficiente pentru gestionarea comenzilor, a angajaților și a stocurilor, pentru a maximiza profitabilitatea și a asigura satisfacția clienților.

Baza de date ajută la urmărirea tuturor comenzilor plasate de clienți, gestionarea produselor din meniu, urmărirea tranzacțiilor financiare și facturarea corectă a acestora. De asemenea, permite managementul personalului, inclusiv al angajaților din diferite funcții, și coordonarea acestora în funcție de necesitățile restaurantului. În acest fel, datele sunt centralizate și pot fi accesate rapid pentru a evalua performanța restaurantului și a lua decizii economice informate.Astfel,prin centralizarea datelor și actualizarea lor în timp real,se pot evita pierderi financiare substanțiale și se poate eficientiza activitatea restaurantului.

* Diagrama realizată:



PARTEA A DOUA A PROIECTULUI

C. Implementarea operaţiilor de actualizare a datelor: INSERT, UPDATE, DELETE.

D. Implementarea interogărilor (SELECT): joncţiuni, funcţii de grup şi condiţii asupra acestora, funcţii numerice, de tip caracter, pentru dată şi timp, expresii folosind DECODE şi CASE, operatori UNION, MINUS, INTERSECT, subcereri, prelucrarea cererilor ierarhice

C.

--Adaugare in tabela produse\_rest un produs ale cărui date sunt introduse de utilizator de la tastatură:

--ALTER TABLE produse\_rest

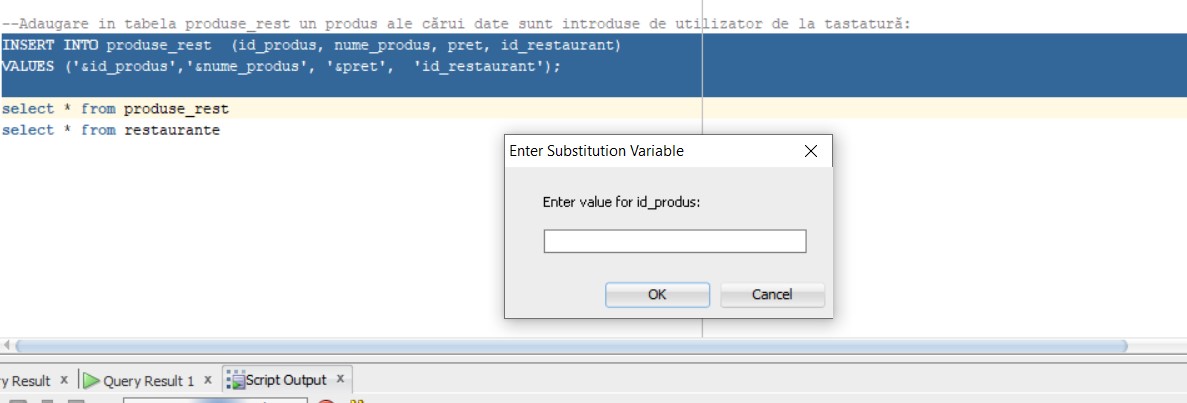
--MODIFY pret NUMBER(15,2);

--ALTER TABLE produse\_rest

--MODIFY id\_restaurant CHAR(15);

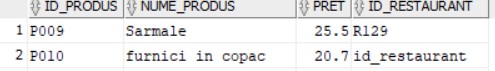
--ALTER TABLE produse\_rest DISABLE CONSTRAINT FK\_PRODUSE\_RESTAURANT; --ALTER TABLE produse\_rest ENABLE CONSTRAINT FK\_PRODUSE\_RESTAURANT;

(am făcut aceste modificări pentru a remedia erorile interogării)

INSERT INTO produse\_rest (id\_produs, nume\_produs, pret, id\_restaurant) VALUES ('&id\_produs','&nume\_produs', '&pret', 'id\_restaurant');

INSERT INTO restaurante (id\_restaurant, nume\_restaurant) VALUES ('VALOARE\_LIPSA', 'Restaurant Necunoscut');

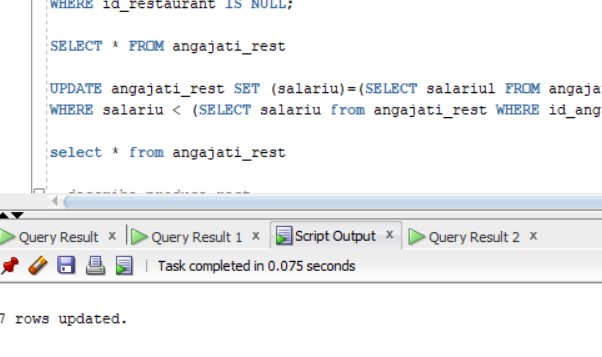
INSERT INTO produse\_rest (id\_produs, nume\_produs, pret, id\_restaurant) VALUES ('P009', 'Sarmale', 25.50, 'R129');



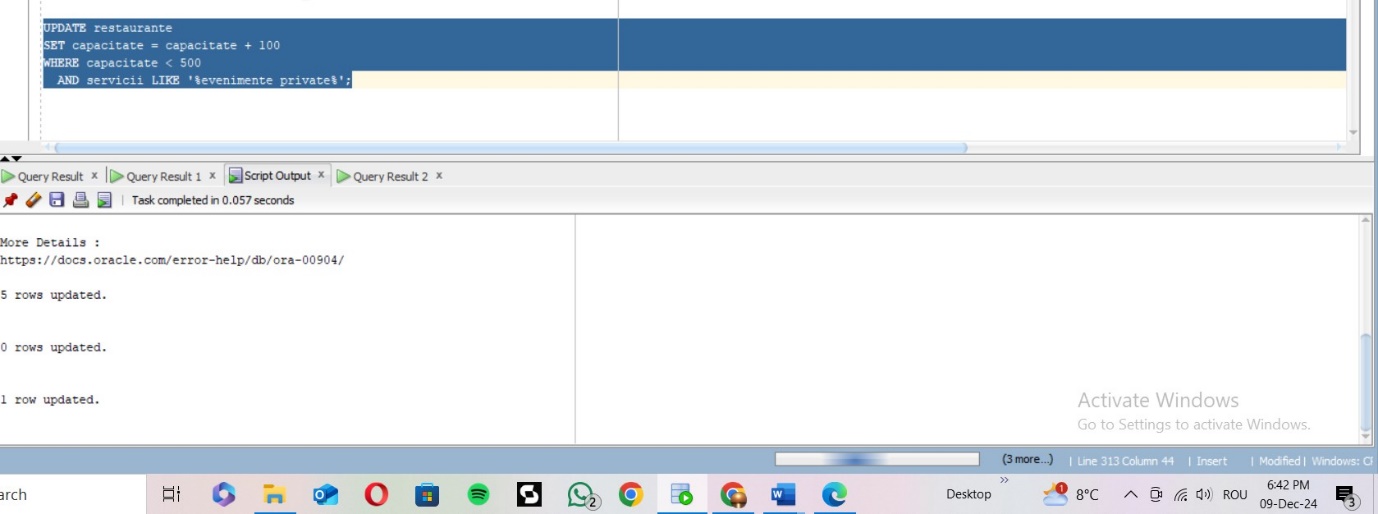
UPDATE produse\_rest SET id\_restaurant = 'R123' WHERE id\_produs = 'P001'; -- înlocuiește P001 cu ID-ul produsului real

UPDATE angajati\_rest SET id\_restaurant = 'R123' WHERE id\_angajat = '1001';

UPDATE angajati\_rest SET id\_restaurant = 'R124' WHERE id\_restaurant IS NULL;

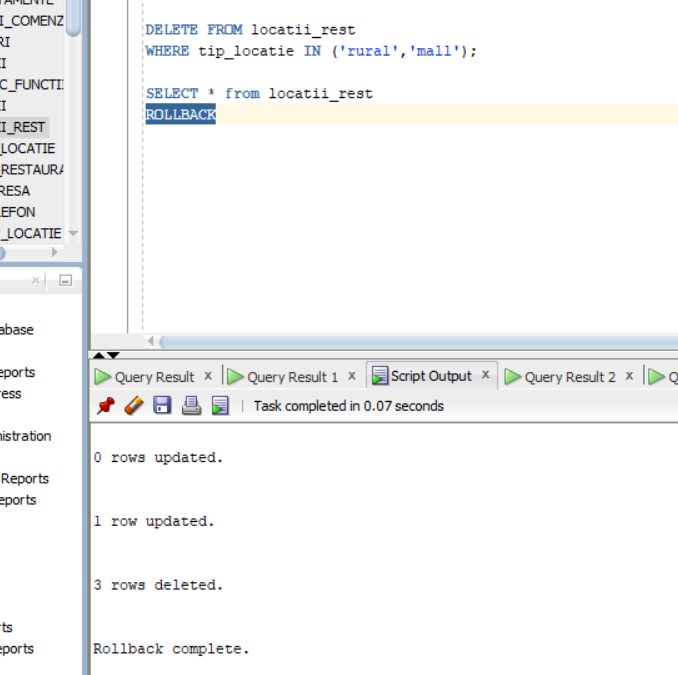


UPDATE angajati\_rest SET (salariu)=(SELECT salariul FROM angajati WHERE id\_angajat=1000) WHERE salariu < (SELECT salariu from angajati\_rest WHERE id\_angajat = 1004 ) AND id\_restaurant ='R124';

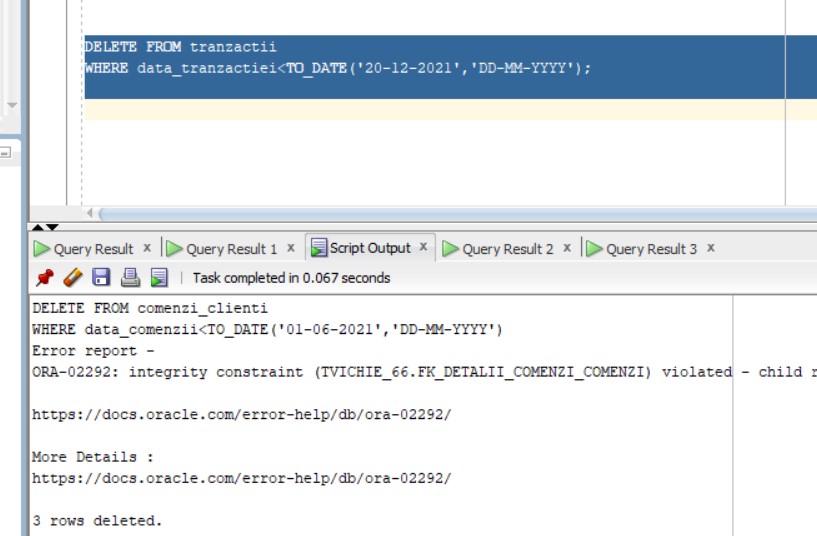
DELETE FROM locatii\_rest

WHERE tip\_locatie IN ('rural','mall');

ROLLBACK;



DELETE FROM tranzactii WHERE data\_tranzactiei<TO\_DATE('20-12-2021','DD-MM-YYYY');

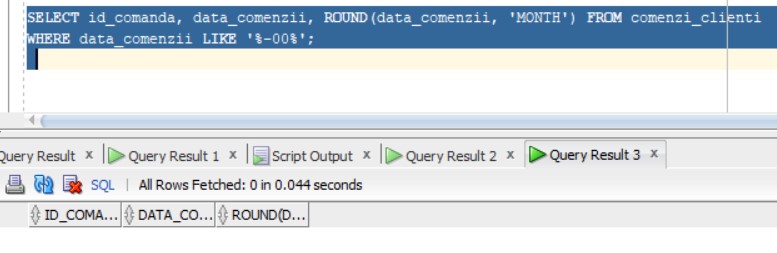


--D.Implementarea interogărilor (SELECT): joncţiuni, funcţii de grup şi condiţii asupra --acestora, funcţii numerice, de tip caracter, pentru dată şi timp, expresii folosind ----DECODE şi CASE, operatori UNION, MINUS, INTERSECT, subcereri, prelucrarea cererilor ierarhice.

(min 5 operatii cu jonctiuni si functii scalare diverse si min 5 operatii utilizand grupari variate)

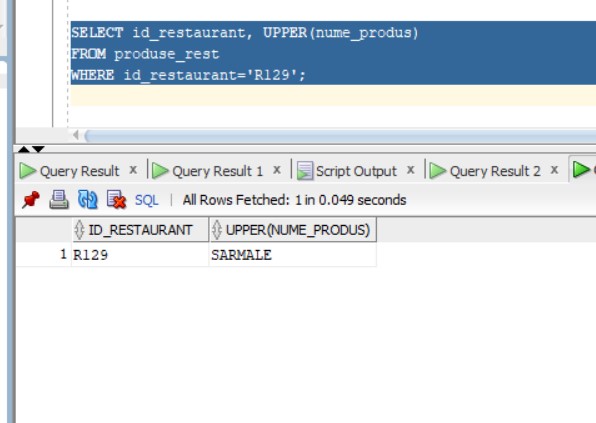
Functii numerice:

SELECT id\_comanda, data\_comenzii, ROUND(data\_comenzii, 'MONTH') FROM comenzi\_clienti WHERE data\_comenzii LIKE '%-00%';

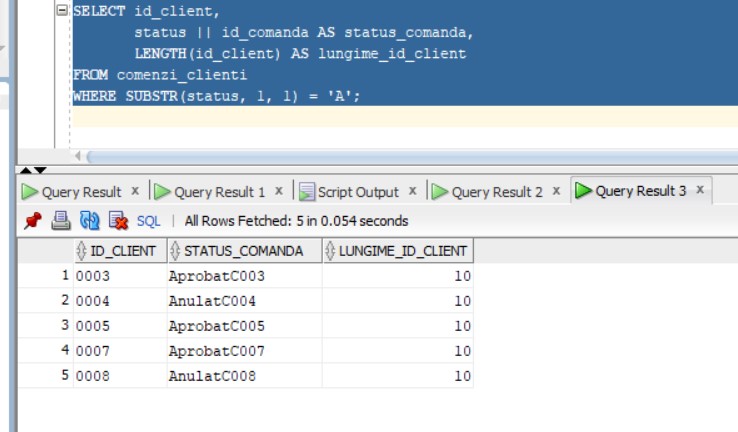


Functii de tip caracter:

SELECT id\_restaurant, UPPER(nume\_produs) FROM produse\_rest WHERE id\_restaurant='R129';

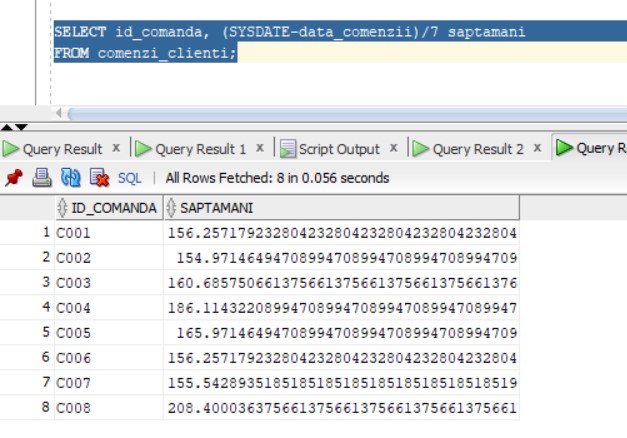


SELECT id\_client, status || id\_comanda AS status\_comanda, LENGTH(id\_client) AS lungime\_id\_client FROM comenzi\_clienti WHERE SUBSTR(status, 1, 1) = 'A';



Functie pentru data si timp:

SELECT id\_comanda, (SYSDATE-data\_comenzii) FROM comenzi\_clienti;



Functie pentru NULL:

SELECT id\_comanda,

NVL2(

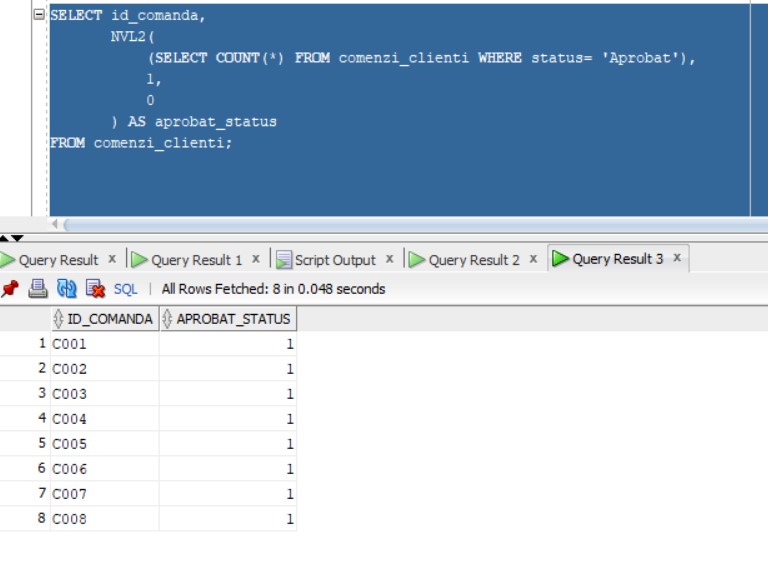
(SELECT COUNT(\*) FROM comenzi\_clienti WHERE status= 'Aprobat'),

1,

0

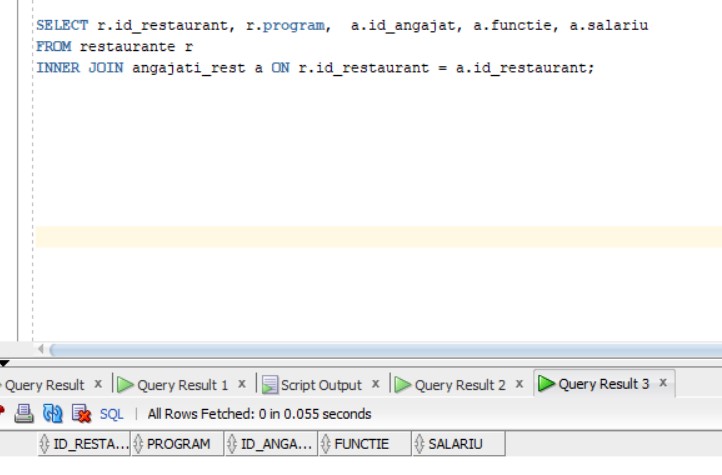
) AS aprobat\_status

FROM comenzi\_clienti;



Jonctiuni:

SELECT r.id\_restaurant, r.program, a.id\_angajat, a.functie, a.salariu FROM restaurante r INNER JOIN angajati\_rest a ON r.id\_restaurant = a.id\_restaurant;



SELECT a.id\_angajat, a.functie FROM angajati\_rest a INNER JOIN restaurante r ON a.id\_restaurant = r.id\_restaurant WHERE r.tip\_bucatarie = 'Italian'

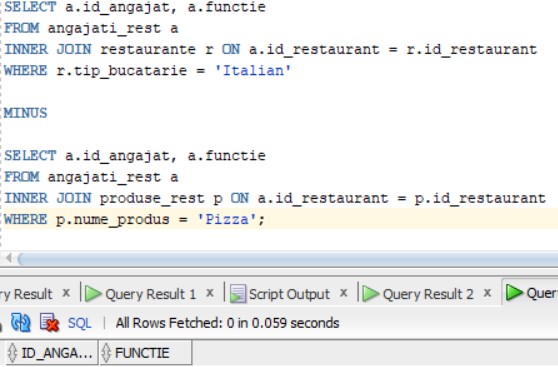
UNION

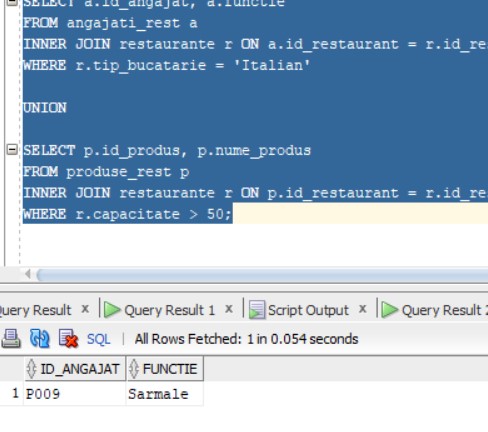
SELECT p.id\_produs, p.nume\_produs FROM produse\_rest p INNER JOIN restaurante r ON p.id\_restaurant = r.id\_restaurant WHERE r.capacitate > 50;

SELECT a.id\_angajat, a.functie FROM angajati\_rest a INNER JOIN restaurante r ON a.id\_restaurant = r.id\_restaurant WHERE r.tip\_bucatarie = 'Italian'

MINUS

SELECT a.id\_angajat, a.functie FROM angajati\_rest a INNER JOIN produse\_rest p ON a.id\_restaurant = p.id\_restaurant WHERE p.nume\_produs = 'Pizza';

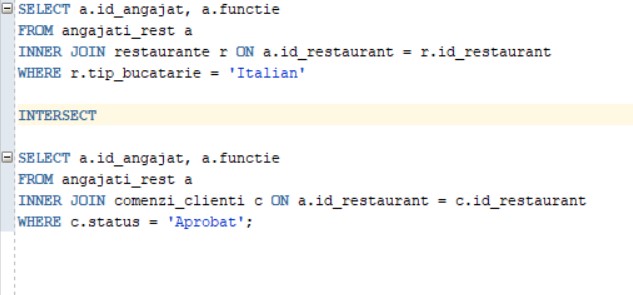




SELECT a.id\_angajat, a.functie FROM angajati\_rest a INNER JOIN restaurante r ON a.id\_restaurant = r.id\_restaurant WHERE r.tip\_bucatarie = 'Italian'

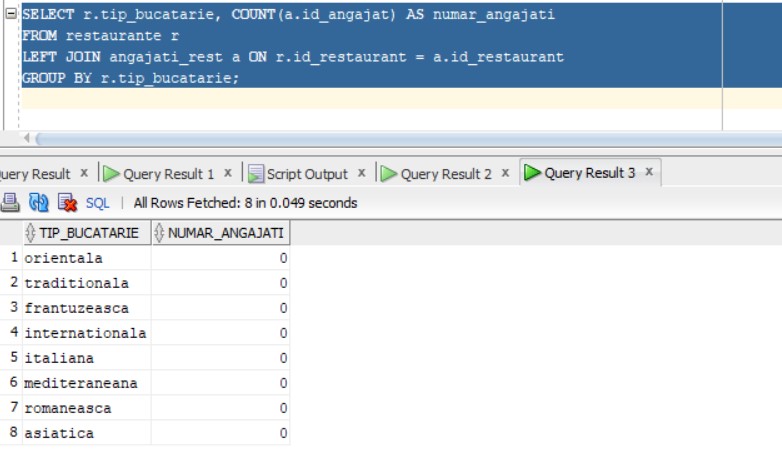
INTERSECT

SELECT a.id\_angajat, a.functie FROM angajati\_rest a INNER JOIN comenzi\_clienti c ON a.id\_restaurant = c.id\_restaurant WHERE c.status = 'Aprobat';

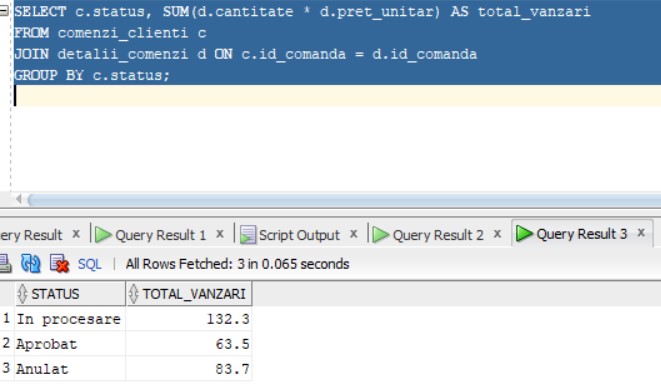


Functii de grup:

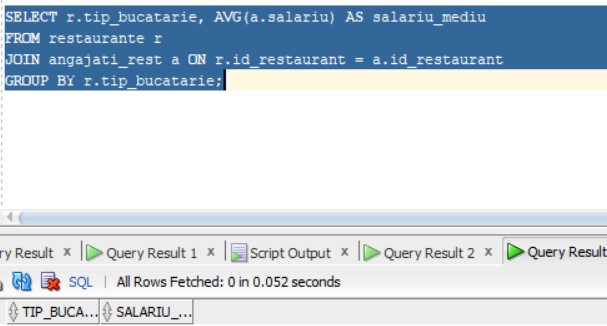
SELECT r.tip\_bucatarie, COUNT(a.id\_angajat) AS numar\_angajati FROM restaurante r LEFT JOIN angajati\_rest a ON r.id\_restaurant = a.id\_restaurant GROUP BY r.tip\_bucatarie;



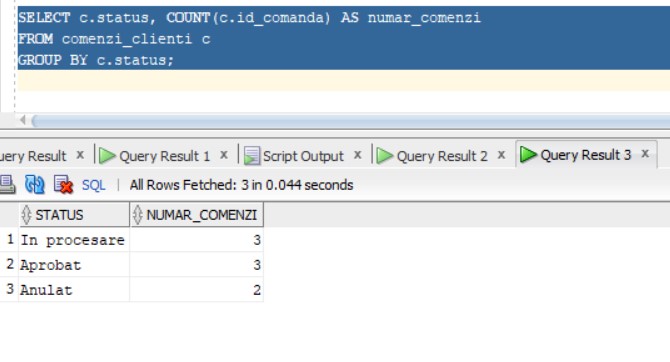
SELECT c.status, SUM(d.cantitate \* d.pret\_unitar) AS total\_vanzari FROM comenzi\_clienti c JOIN detalii\_comenzi d ON c.id\_comanda = d.id\_comanda GROUP BY c.status;



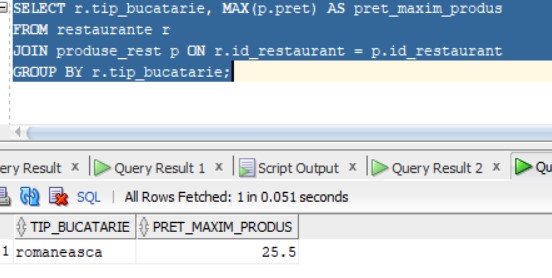
SELECT r.tip\_bucatarie, AVG(a.salariu) AS salariu\_mediu FROM restaurante r JOIN angajati\_rest a ON r.id\_restaurant = a.id\_restaurant GROUP BY r.tip\_bucatarie;



SELECT c.status, COUNT(c.id\_comanda) AS numar\_comenzi FROM comenzi\_clienti c GROUP BY c.status;



SELECT r.tip\_bucatarie, MAX(p.pret) AS pret\_maxim\_produs FROM restaurante r JOIN produse\_rest p ON r.id\_restaurant = p.id\_restaurant GROUP BY r.tip\_bucatarie;



PARTEA A TREIA

--1. Efectuați min. 5 operații de interogare a datelor utilizând subcereri

SELECT id\_angajat, functie, salariu

FROM angajati\_rest

WHERE id\_restaurant IN (

SELECT id\_restaurant

FROM restaurante

WHERE capacitate > 100

);

A screenshot of a computer

Description automatically generated

SELECT id\_comanda, data\_comenzii, status

FROM comenzi\_clienti

WHERE id\_client IN (

SELECT id\_client

FROM clienti\_rest

WHERE metoda\_plata = 'Card'

);

A screenshot of a computer program

Description automatically generated

SELECT id\_comanda, data\_comenzii, status

FROM comenzi\_clienti

WHERE id\_angajat = (

SELECT id\_angajat

FROM angajati\_rest

WHERE functie = 'Manager' AND id\_restaurant = 'R001'

);

A screenshot of a computer

Description automatically generated

SELECT DISTINCT pr.nume\_produs

FROM produse\_rest pr

JOIN detalii\_comenzi dc ON pr.id\_produs = dc.id\_produs

WHERE dc.id\_comanda IN (

SELECT id\_comanda

FROM tranzactii

WHERE data\_tranzactiei = TO\_DATE('2024-12-01', 'YYYY-MM-DD')

);

A screenshot of a computer program

Description automatically generated

SELECT DISTINCT r.id\_restaurant, r.tip\_bucatarie, r.capacitate

FROM restaurante r

WHERE r.id\_restaurant IN (

SELECT a.id\_restaurant

FROM angajati\_rest a

WHERE a.salariu > (

SELECT AVG(salariu)

FROM angajati\_rest

)

);

A screenshot of a computer

Description automatically generated

--2.Efectuați min. 5 operații de interogare a datelor utilizând cereri ierarhice (în cazul în care baza de date nu se pretează pentru parcurgeri ierarhice, suplimentați numărul subcererilor de la punctul 1)

SELECT id\_restaurant, 'Restaurant' AS nivel, NULL AS id\_parinte

FROM restaurante

UNION ALL

SELECT id\_locatie, 'Locație', id\_restaurant

FROM locatii\_rest

UNION ALL

SELECT id\_angajat, 'Angajat', id\_restaurant

FROM angajati\_rest

START WITH id\_restaurant = 'R001' -- Pornim de la un anumit restaurant

CONNECT BY PRIOR id\_restaurant = id\_restaurant;

A screenshot of a computer

Description automatically generated

SELECT id\_locatie, 'Locație' AS nivel, NULL AS id\_parinte

FROM locatii\_rest

UNION ALL

SELECT id\_comanda, 'Comandă', id\_locatie

FROM comenzi\_clienti

UNION ALL

SELECT id\_client, 'Client', id\_comanda

FROM clienti\_rest

START WITH id\_locatie = 'L001'

CONNECT BY PRIOR id\_locatie = id\_locatie;

SELECT status, 'Status Comandă' AS nivel, NULL AS id\_parinte

FROM comenzi\_clienti

UNION ALL

SELECT status\_tranzactie, 'Status Tranzacție', status

FROM tranzactii

START WITH status = 'Aprobat'

CONNECT BY PRIOR status = status;