

# Proiect baze de date

HOTEL – Robu Florentina-Corina

## Cuprins:

1. Descrierea modelului real
2. Prezentarea constrângerilor
3. Descrierea entităților
4. Descrierea relațiilor
5. Descrierea atributelor
6. Diagrama ER
7. Diagrama conceptuală
8. Enumerarea schemelor relaționale
9. Normalizările FN1-FN3
10. Secvență
11. Creare tabelor în SQL și inserarea datelor
12. Cererile SQL
13. Operațiile de actualizare și ștergere

## 1. DESCRIEREA MODELULUI REAL

Modelul de date proiectat gestionează activitatea unui hotel. Clienții hotelului interacționează cu angajați. Aceștia sunt perfecționați pentru orice probleme , de la defecțiuni în camere până la încasarea plăților. Hotelul are și un mic restaurant de la care clienții pot comanda atât în camera, cât și în incinta localului. Baza de date reține date despre fiecare client, de fiecare data când se cazează aici. Poți obține informații și despre mâncarea comandată, cât și despre camera în care a stat, perioada de cazare și eventualele interacțiuni cu angajații.

## 2. PREZENTAREA CONSTRÂNGERILOR

- Exista mai mulți furnizori pentru același tip de produs(ex: Supa de pui)
- Clienții mereu vor dispune de mese în incinta restaurantului hotelului (numărul lor depășește capacitatea de cazare).
- Deși fiecare angajat se pricepe la orice problema întâmpinată de client, un client poate interacționa cu mai mulți angajați în funcție de disponibilitatea acestora în momentul respectiv
- Fiecare rezervare trebuie să conțină o singură camera
- Fiecare comandă poate conține mai multe produse, și fiecare produs poate fi continuat în mai multe comenzi.

## 3. DESCRIEREA ENTITĂȚILOR

Client = persoana ce dorește cazare în hotel în schimbul unei sume de bani.  
Cheia primară este id\_client

Angajat = persoana care oferă servicii în schimbul unui salariu. Cheia primară este id\_angajat

Plată = suma de bani oferită în urma unor servicii sau bunuri materiale. Cheia primară este id\_plată

Camera = loc dat spre închiriere unui client pe timpul unei perioade stabilite. Cheia primara este numar\_camera

Rezervare = păstrarea unei camere pentru clientul ce face rezervarea , cu scopul de a fi libera în perioada dorita de acesta. Cheia primara este id\_rezervare

Comanda = plasarea unui cereri de a fi se aduce mâncarea dorita. Cheia primara este id\_comanda

Produs = bun material rezultat în urma gătirii unor alimente. Cheia primara este id\_produs

Furnizor = persoana fizica ce livrează marfa. Cheia primara este id\_furnizor

#### **4. DESCRIEREA RELATIILOR**

Clientul realizează rezervarea : Un client poate realiza mai multe rezervări, și trebuie să realizeze măcar 1. O rezervare poate fi realizata de 1 singur client și neapărat de 1. Cardinalitatea maxima este de n:1, și cea minima 1:1.

Rezervarea conține camere : O rezervare poate conține o singura camera și trebuie să conțină măcar 1. O camera poate fi conținută în mai multe rezervări și nu neapărat în vreuna. Cardinalitatea maxima este de n:1 și cea minima 0:1.

Rezervarea are plata : O rezervare poate avea o singura plata și trebuie să aibă măcar 1. O plata poate fi pe o singura rezervare și neapărat pe una. Cardinalitatea maxima și cea minima este de 1:1.

Clientul interacționează cu angajați : Un client poate interacționează cu mai mulți angajați și nu trebuie să interacționeze neapărat cu vreunul. Angajatul poate interacționează cu mai mulți clienți , dar nu neapărat cu vreunul( doar ce s-a angajat). Cardinalitatea maxima este de n:n, iar cea minima 0:0.

Clientul face comanda : Un client poate face mai multe comenzi și nu trebuie să facă neapărat vreo comanda. O comanda poate fi făcută de un singur client și neapărat de măcar unul. Cardinalitatea maxima este de n:1, și cea minima de 1:1.

Comanda conține produse : O comanda poate conține mai multe produse și neapărat 1 produs. Un produs poate fi conținut de mai multe comenzi și nu neapărat de vreuna. Cardinalitatea maxima este de n:n, iar cea minima 0:1.

Produsul este adus de furnizor : Un produs poate fi adus de mai mulți furnizori și neapărat de 1. Un furnizor poate aduce mai multe produse și măcar 1. Cardinalitatea maxima este de n:n, iar cea minima 1:1.

## 5. DESCRIEREA ATRIBUTELOR

### Angajat :

- Id\_angajat : variabila de tip număr întreg de maxim 3 cifre, unica, nu poate lipsi. Ex: 12
- Nume\_angajat : variabila de tip varchar2 de lungime 20, nu poate fi null. Reprezintă numele de familie al angajatului. Ex : Ionescu
- Data\_angajarii : variabila de tip date, nu poate lipsi. Reprezintă data în care s-a angajat la hotelul actual angajatul. Ex : 10-IAN-00.

### Client :

- Id\_client : variabila de tip număr întreg de maxim 4 cifre, unica, nu poate lipsi. Ex: 4329
- Nume\_client : variabila de tip varchar2 de lungime 20, nu poate fi null. Reprezintă numele de familie al clientului. Ex : Ionescu
- Email: variabila de tip varchar2 de lungime maxim 20, nu poate lipsi, este unic, reprezintă emailul clientului. EX : myemail@gmail.com
- Numar\_telefon : variabila de tip varchar2 de lungime 10, nu poate lipsi, este unic, reprezintă numărul de telefon al clientului. EX: 0766783086

### Camera :

- Numar\_camera : variabila de tip number cu maxim 4 cifre, unica, nu poate lipsi. EX: 248
- Tip\_camera : variabila de tip varchar2, maxim 20 , nu poate lipsi. Reprezintă tipul camerei . EX : single
- Preț : variabila de tip number de maxim 3 cifre, nu poate lipsi. Reprezintă prețul de închiriere pe noapte al camerei. EX: 470

### Plata :

- Id\_plata: variabila de tip number cu maxim 4 cifre, unica, nu poate lipsi. EX: 24
- Modalitate : variabila de tip varchar2 de maxim 10. Reprezintă modalitatea de plata. EX : cash / card

- Statut : variabila de tip varchar2 de maxim 20, nu poate lipsi. Reprezintă statutul plății. EX : neplatita / plătită

#### Rezervare :

- Id\_rezervare: variabila de tip number cu maxim 4 cifre, unica, nu poate lipsi. EX: 24
- Data\_cazare : variabila de tip date, nu poate lipsi. Reprezintă data în care s-a cazat clientul. Ex : 10-IAN-00.
- Data\_plecare : variabila de tip date, nu poate lipsi. Reprezintă data în care a părăsit clientul camera. Ex : 10-IAN-00.

#### Comanda :

- Id\_comanda : variabila de tip number cu maxim 4 cifre, unica, nu poate lipsi. EX: 24
- Data\_comanda : variabila de tip date, nu poate lipsi. Reprezintă data în care clientul a comandat mâncarea/ băuturile. EX : 10-IAN-00.
- Loc\_servire: variabila de tip varchar2 de maxim 20. Reprezintă locul de servire a mâncării/ băuturilor comandate. EX: în camera/ restaurant

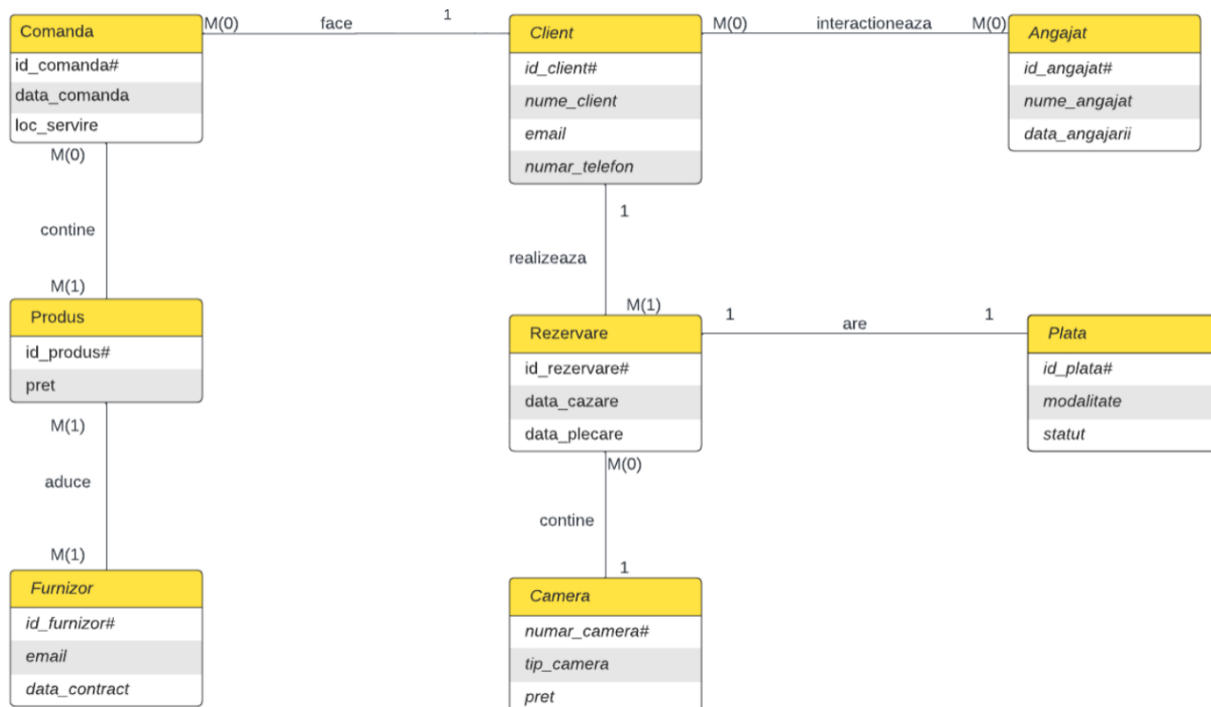
#### Produs :

- Id\_produs : variabila de tip number cu maxim 4 cifre, unica, nu poate lipsi. EX: 24
- Preț : variabila de tip number maxim 3 cifre, nu poate lipsi . Reprezintă prețul produsului comandat. EX : 200

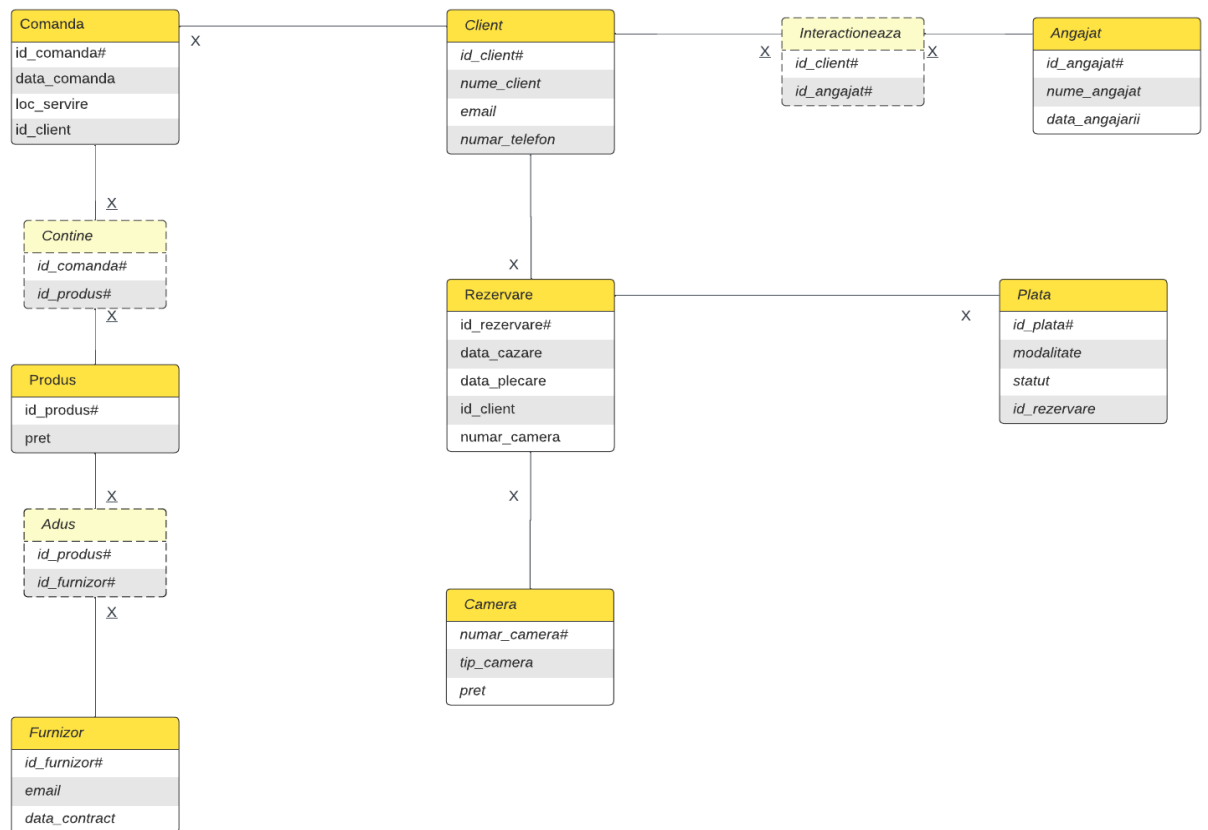
#### Furnizor :

- Id\_furnizor: variabila de tip number cu maxim 4 cifre, unica, nu poate lipsi. EX: 24
- Email : variabila de tip varchar2 de lungime maxim 20, este unic, reprezintă emailul furnizorului . EX : myemail@gmail.com
- Data\_contract : variabila de tip date, nu poate lipsi. Reprezintă data în care hotelul a făcut contractul cu respectivul furnizor. Ex : 10-IAN-00.

## **6. DIAGRAMA ENTITATE-RELATIE**



## 7. DIAGRAMA CONCEPTUALA



## 8. SCHEMELE RELATIONALE

Client (id\_client#, nume\_client, email, numar\_telefon)

Angajat (id\_angajat#, nume, data\_angajarii)

Plata (id\_plata#, modalitate, statut, id\_rezervare)

Camera (numar\_camera#, tip\_camera, pret)

Rezervare (id\_rezervare#, data\_cazare, data\_plecare, id\_client, numar\_camera)

Comanda (id\_comanda#, data\_comada, loc\_servire, id\_client)

Produs (id\_produs#, pret)

Furnizor (id\_furnizor#, email, data\_contract)

Client\_Interactioneaza\_Angajat (id\_client#, id\_angajat#)

Comanda\_contine\_produse (id\_comanda#, id\_produs#)

Produs\_adus\_furnizor (id\_produs#, id\_furnizor#)

## 9. NORMALIZARILE FN1-FN3

### - FN1

Diagrama realizată se află în FN1 deoarece fiecare tabel are o cheie primară, fiecare celulă conține o singură valoare și nu există grupuri care se repetă.

Un exemplu de non fn-1 ar fi ca în tabelul comanda să am un câmp cu produse comandate, în felul acesta câmpul ar putea conține o listă de produse.

Comanda:

Id_comanda	Data_comada	Loc_servire	Id_client	Produse_comandate
12	20-IAN-2020	camera	53	12,4,20

Rezolvarea fiind să fac tabel separat pentru produs, astfel fiecare rând din tabelul asociativ 'contine', va conține un singur produs pentru o anumită comandă.

Comanda:

Id_comanda	Data_comanda	Loc_servire	Id_client
12	20-IAN-2020	camera	53

Produs:

Id_produs	pret
12	20
4	13

Tabel asociativ Contine:

Id_comanda	Id_produs
12	12
12	4



## - FN2

Diagrama realizată se află în FN2 deoarece toate atributele care nu sunt cheie primară sunt dependente de întreaga cheie primară.

Un exemplu de non fn-2 ar fi ca în tabelul asociativ "interactioneaza" să am atributul nume\_client, acesta fiind dependent de cheia primară id\_client, dar nu și de cea id\_angajat.

Interactioneaza :

Id_client	Id_angajat	Nume_client
53	20	Robu

Soluția ar fi să șterg acest atribut și să las informațiile despre client în tabelul "Client", tabelul asociativ având doar cele 2 chei primare id\_client, și id\_angajat ce îl leagă de tabelele "Client" și "Angajat".

Interactioneaza:

Id_client	Id_angajat
53	20

Client:

Id_client	Nume_client	Numar_telefon	email
53	Robu	0766783086	Robu@gmail.ro

## - FN3

Diagrama realizată se află în FN3 deoarece fiecare atribut care nu este cheie primară depinde de întreaga cheie primară și numai de ea.

Un exemplu de non fn-3 ar fi sa am in tabelul Produs si numele furnizorului, caci pretul unui produs e dependent de furnizorul care il aduce, astfel un atribut non-cheie (furnizor) ar fi dependent de alt atribut non-cheie(pret), ce se leaga de cheia primara id\_produs.

Produs:

Id_produs	pret	Nume_furnizor
10	8	furnizor1

Solutia fiind sa fac un tabel separat pentru furnizor.

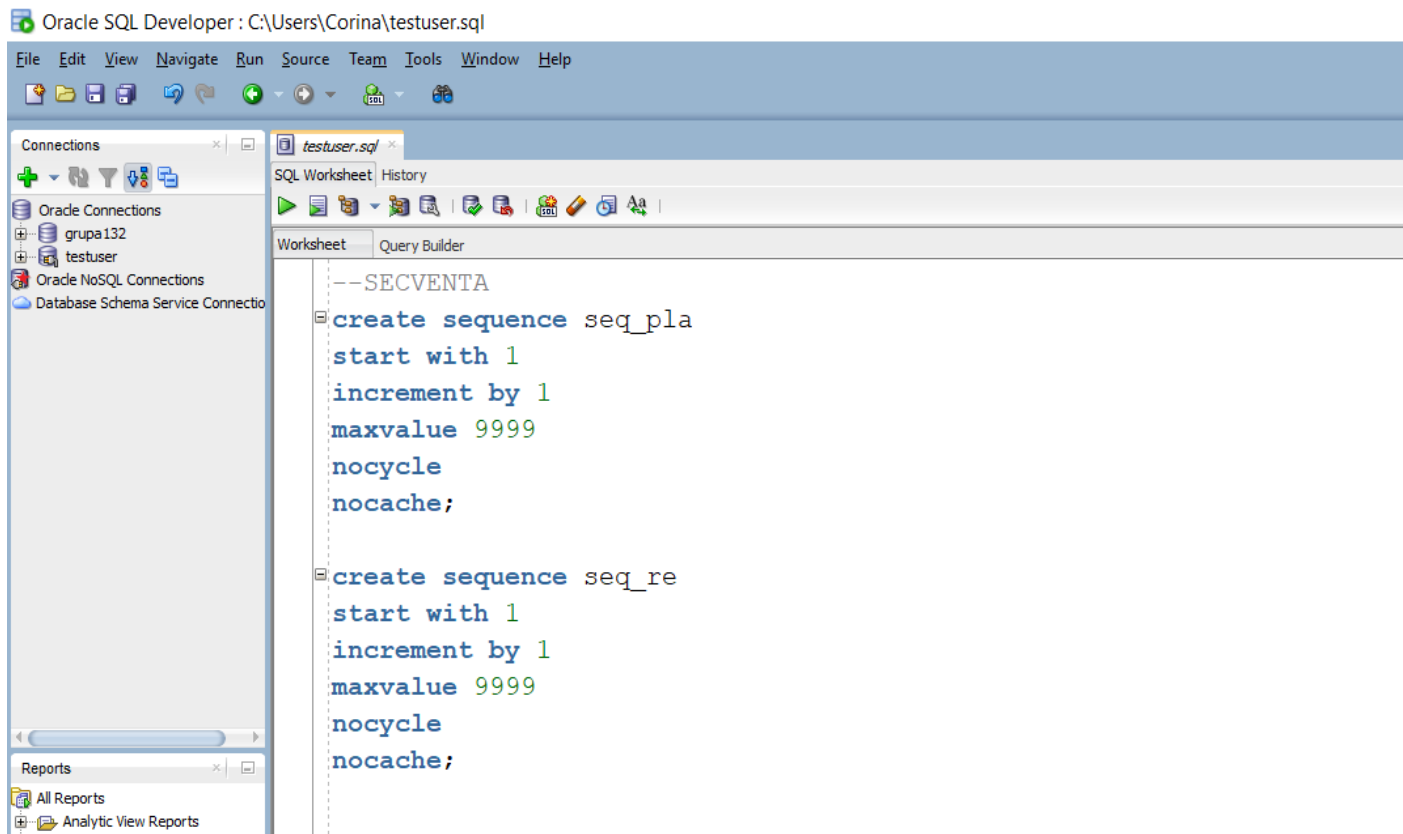
Produs:

Id_produs	pret
10	8

Furnizor:

Id_furnizor	email	Data_contract
4	Furnizor1@gmail.com	10-OCT-2019

## 10. SECVENTA



```
create sequence seq_pla
```

```
start with 1
```

```
increment by 1
```

```
maxvalue 9999
```

```
nocycle
```

```
nocache;
```

```
create sequence seq_re
```

```
start with 1
```

```
increment by 1
```

```
maxvalue 9999
```

```
nocycle
```

```
nocache;
```

## 11. CREEAREA TABELELOR IN SQL SI INSERAREA DATELOR

- Tabel rezervare

```
create table rezervare (id_rezervare number(4),
data_cazare date constraint null_re_ca not null,
data_plecare date constraint null_re_ple not null,
id_client number(4),
numar_camera number(4),
constraint pk_rez primary key(id_rezervare),
constraint fk_re_cli foreign key(id_client)
references client(id_client) ON DELETE CASCADE,
constraint fk_re_ca foreign key(numar_camera)
references camera(numar_camera)ON DELETE CASCADE);
```

```
insert into rezervare
values(seq_re.nextval, '04-MAR-23', '07-MAR-23', 53, 421);
```

```
insert into rezervare
values(seq_re.nextval, '04-MAR-23', '07-MAR-23', 53, 500);
```

```
insert into rezervare
values(seq_re.nextval, '01-MAI-23', '02-MAR-23', 12, 111 );
```

```
insert into rezervare
values(seq_re.nextval, '05-MAI-23', '07-MAR-23', 214, 23 );
```

```
insert into rezervare
```

```
values(seq_re.nextval, '10-MAI-22', '17-IUN-23', 300, 111);
```

```
insert into rezervare
```

```
values(seq_re.nextval, '04-MAI-23', '07-Mai-23', 800, 701);
```

```
insert into rezervare
```

```
values(seq_re.nextval, '04-MAR-23', '07-MAR-23', 800, 832);
```

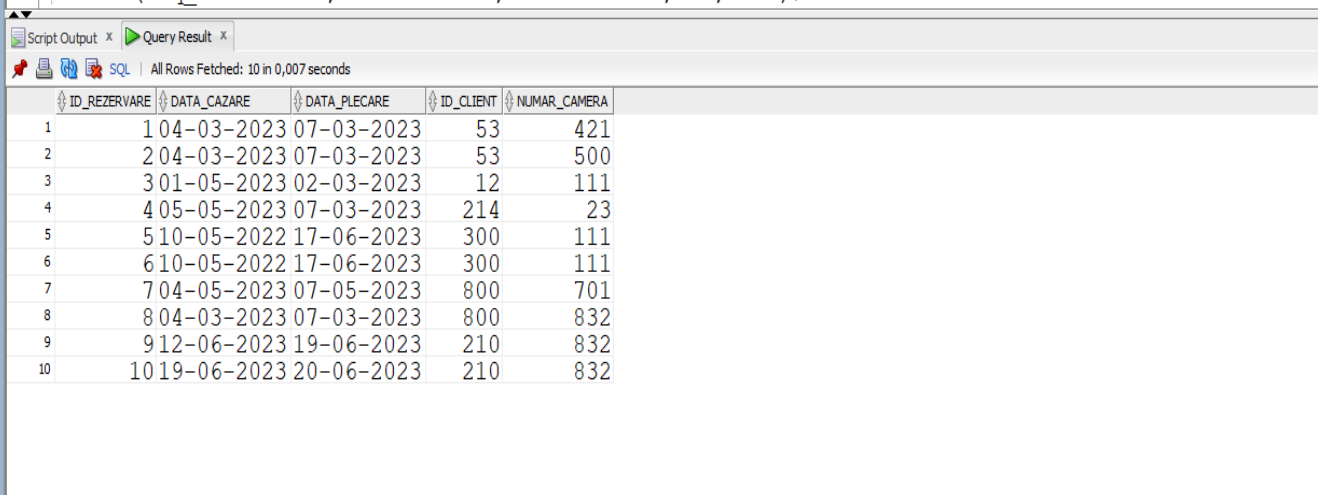
```
insert into rezervare
```

```
values(seq_re.nextval, '12-IUN-23', '19-IUN-23', 210, 832);
```

```
insert into rezervare
```

```
values(seq_re.nextval, '19-IUN-23', '20-IUN-23', 210, 832);
```

```
select * from rezervare;
```



The screenshot shows a SQL query result window with the following data:

ID_REZERVARE	DATA_CAZARE	DATA_PLECARE	ID_CLIENT	NUMAR_CAMERA
1	04-03-2023	07-03-2023	53	421
2	04-03-2023	07-03-2023	53	500
3	01-05-2023	02-03-2023	12	111
4	05-05-2023	07-03-2023	214	23
5	10-05-2022	17-06-2023	300	111
6	10-05-2022	17-06-2023	300	111
7	04-05-2023	07-05-2023	800	701
8	04-03-2023	07-03-2023	800	832
9	12-06-2023	19-06-2023	210	832
10	19-06-2023	20-06-2023	210	832

- Tabel client

```
create table client (id_client number(4),
```

```
nume_client varchar2(20) constraint null_cli_nu not null,
```

```
numar_telefon varchar2(10) constraint null_cli_num not null,  
email varchar2(20) constraint null_cli_e not null ,  
constraint pk_cli primary key(id_client),  
constraint unq_cli_e unique(email),  
constraint unq_cli_num unique(numar_telefon)  
);
```

```
insert into client  
values(12,'Ionescu', '0711111111','ion@gmail.io');
```

```
insert into client  
values(53,'Robu', '0766783086','robu@gmail.ro');
```

```
insert into client  
values(214,'Spiridon', '0748253119','spiridon@gmail.sp');
```

```
insert into client  
values(300,'Popescu', '0722277789','popescu@gmail.po');
```

```
insert into client  
values(531,'Enache', '0723021298','enache@gmail.en');
```

```
insert into client  
values(800,'Vasile', '0723021291','vasile@gmail.va');
```

```
insert into client  
values(210,'Barbu', '0733021298','barbu@gmail.ba');
```

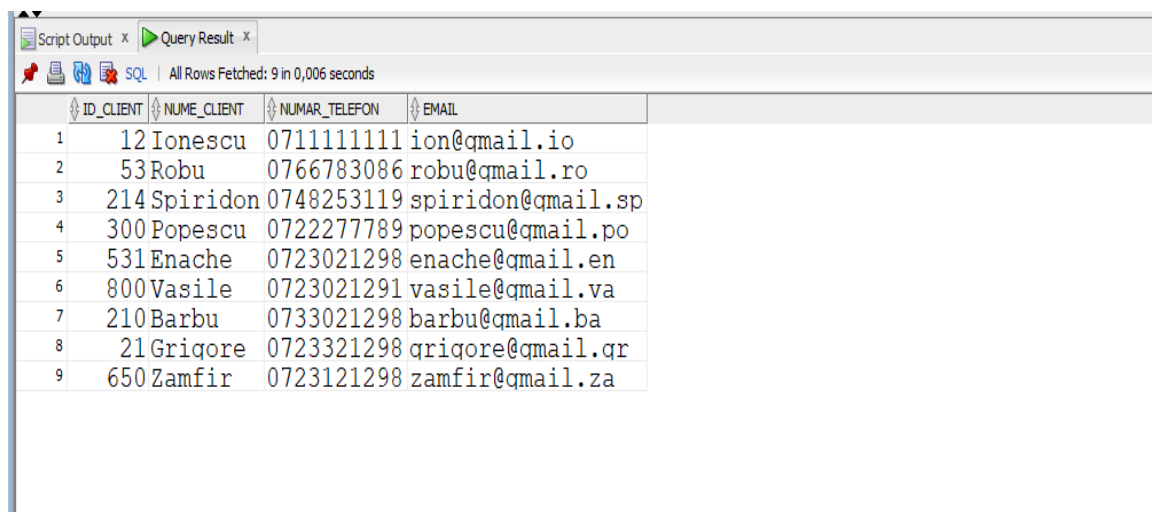
insert into client

values(650,'Zamfir', '0723121298','zamfir@gmail.za');

insert into client

values(21,'Grigore', '0723321298','grigore@gmail.gr');

select \* from client;



The screenshot shows a SQL query result window with a table containing 9 rows. The columns are ID\_CLIENT, NUME\_CLIENT, NUMAR\_TELEFON, and EMAIL. The data is as follows:

ID_CLIENT	NUME_CLIENT	NUMAR_TELEFON	EMAIL
1	12 Ionescu	0711111111	ion@gmail.io
2	53 Robu	0766783086	robu@gmail.ro
3	214 Spiridon	0748253119	spiridon@gmail.sp
4	300 Popescu	0722277789	popescu@gmail.po
5	531 Enache	0723021298	enache@gmail.en
6	800 Vasile	0723021291	vasile@gmail.va
7	210 Barbu	0733021298	barbu@gmail.ba
8	21 Grigore	0723321298	grigore@gmail.gr
9	650 Zamfir	0723121298	zamfir@gmail.za

- Tabel camera

create table camera( numar\_camera number(4),  
tip\_camera varchar2(20) constraint null ca\_tip not null,  
pret number(3) constraint null ca\_pre not null,  
constraint pk\_ca primary key(numar\_camera)  
);

insert into camera

values(111,'single', 350);

insert into camera  
values(23,'single', 280);

insert into camera  
values(51,'single', 300);

insert into camera  
values(421,'dubla', 390);

insert into camera  
values(500,'dubla', 420);

insert into camera  
values(502,'dubla', 425);

insert into camera  
values(701,'tripla', 590);

insert into camera  
values(810,'apartament', 630);

insert into camera  
values(832,'apartament', 685);

select \* from camera;



Script Output x Query Result x			
All Rows Fetched: 9 in 0,011 seconds			
	NUMAR_CAMERA	TIP_CAMERA	PRET
1	111	single	350
2	23	single	280
3	51	single	300
4	421	dubla	390
5	500	dubla	420
6	502	dubla	425
7	701	tripla	590
8	810	apartament	630
9	832	apartament	685

- Tabel plata

```
create table plata( id_plata number(4),
modalitate varchar2(10),
statut varchar2(20) constraint null_pla_sta not null,
id_rezervare number(4),
constraint pk_pla primary key(id_plata),
constraint unq_rez unique(id_rezervare),
constraint fk_pla_re foreign key(id_rezervare)
references rezervare(id_rezervare) on delete cascade);
```

```
insert into plata
values(seq_pla.nextval, 'card', 'platita',2 );
```

```
insert into plata
values(seq_pla.nextval, 'card', 'platita', 5);
```

```
insert into plata  
values(seq_pla.nextval,null, 'neplatita', 4);
```

```
insert into plata  
values(seq_pla.nextval, 'cash', 'platita', 1);
```

```
insert into plata  
values(seq_pla.nextval,null, 'neplatita', 3);
```

```
insert into plata  
values(seq_pla.nextval,null, 'neplatita', 7);
```


```
insert into plata  
values(seq_pla.nextval,null, 'neplatita', 6);
```

```
insert into plata  
values(seq_pla.nextval, 'card' , 'platita', 8);
```

```
insert into plata  
values(seq_pla.nextval, 'cash', 'platita', 9);
```

```
select * from plata;
```

Script Output x Query Result x

 | All Rows Fetched: 9 in 0,006 seconds

	ID_PLATA	MODALITATE	STATUT	ID_REZERVARE
1	1	card	platita	2
2	2	card	platita	5
3	3 (null)		neplatita	4
4	4	cash	platita	1
5	5 (null)		neplatita	3
6	6 (null)		neplatita	7
7	7 (null)		neplatita	6
8	8	card	platita	8
9	9	cash	platita	9

- Tabel angajat

```
create table angajat (id_angajat number(3),
nume varchar2(20) constraint null_an_nu not null,
data_angajarii date constraint null_an_da not null,
constraint pk_an primary key(id_angajat));
```

```
insert into angajat
values(7, 'Ungureanu', '16-FEB-19');
```

```
insert into angajat
values(11, 'Spiridon', '11-FEB-20');
```

```
insert into angajat
values(21, 'Robu', '10-IAN-21');
```

```
insert into angajat
```

```
values(23, 'Ionescu', '23-MAR-23');
```

```
insert into angajat
```

```
values(28, 'Popescu', '2-DEC-19');
```

```
insert into angajat
```

```
values(20, 'Marcu', '2-MAI-23');
```

```
insert into angajat
```

```
values(49, 'Mircea', '2-DEC-22');
```

```
insert into angajat
```

```
values(27, 'Sbora', '16-IUN-23');
```

```
insert into angajat
```

```
values(13, 'Georgescu', '26-IUN-23');
```

```
insert into angajat
```

```
values(2, 'Enache', '5-IUN-23');
```

```
select * from angajat;
```

Script Output x Query Result x			
SQL   All Rows Fetched: 10 in 0,005 seconds			
ID_ANGAJAT	NUME	DATA_ANGAJARII	
1	7 Unquareanu	16-02-2019	
2	11 Spiridon	11-02-2020	
3	21 Robu	10-01-2021	
4	23 Ionescu	23-03-2023	
5	28 Popescu	02-12-2019	
6	20 Marcu	02-05-2023	
7	49 Mircea	02-12-2022	
8	27 Sboru	16-06-2023	
9	13 Georgescu	26-06-2023	
10	2 Enache	05-06-2023	

- Tabel comanda

```
create table comanda( id_comanda number(4),
data_comanda date constraint null_co_da not null,
loc_servire varchar2(20),
id_client number(4),
constraint pk_co primary key(id_comanda),
constraint fk_co_cli foreign key(id_client)
references client(id_client) ON DELETE CASCADE);
```

```
insert into comanda
values(23, '12-IAN-19', 'camera', 12);
```

```
insert into comanda
values(31, '14-IAN-19', 'camera', 12);
```

```
insert into comanda
values(63, '15-MAI-19', 'restaurant', 53);
```

insert into comanda

values(132, '12-AUG-20', 'camera', 300);

insert into comanda

values(137, '15-AUG-20', 'camera', 300);

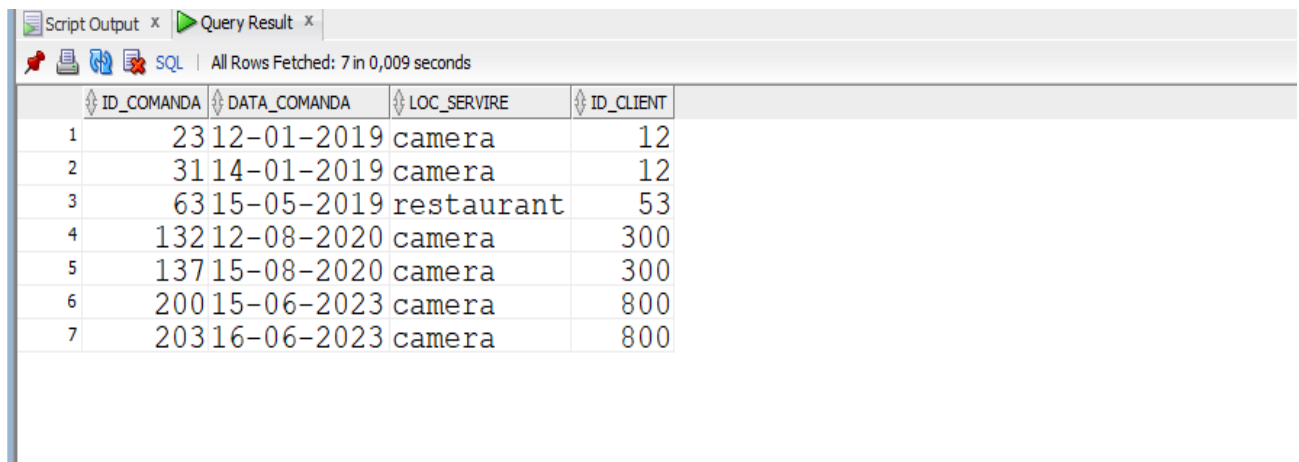
insert into comanda

values(200, '15-IUN-23', 'camera', 800);

insert into comanda

values(203, '16-IUN-23', 'camera', 800);

select \* from comanda;



The screenshot shows a SQL query result window with the following data:

ID_COMANDA	DATA_COMANDA	LOC_SERVIRE	ID_CLIENT
1	23-12-2019	camera	12
2	31-01-2019	camera	12
3	15-05-2019	restaurant	53
4	12-08-2020	camera	300
5	15-08-2020	camera	300
6	15-06-2023	camera	800
7	16-06-2023	camera	800

- Tabel produs

```
create table produs( id_produs number(4),  
pret number(3) constraint null_pro_pre not null,  
constraint pk_pro primary key(id_produs));
```

```
insert into produs  
values(10, 55);
```

```
insert into produs  
values(12, 40);
```

```
insert into produs  
values(15, 25);
```

```
insert into produs  
values(16, 38);
```

```
insert into produs  
values(20, 75);
```

```
insert into produs  
values(22, 10);
```

```
insert into produs  
values(23, 23);
```

```
insert into produs  
values(25, 30);
```

```
insert into produs  
values(31, 175);
```

insert into produs

values(37, 40);

insert into contine

values (200, 31);

insert into contine

values (203, 31);

select \* from produs;



The screenshot shows a database query result window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying a table with 10 rows and 2 columns: 'ID\_PRODUS' and 'PRET'. The data is as follows:

	ID_PRODUS	PRET
1	10	55
2	12	40
3	15	25
4	16	38
5	20	75
6	22	10
7	23	23
8	25	30
9	31	175
10	37	40

- Tabel furnizor

create table furnizor(id\_furnizor number(4),

email varchar2(20) constraint null\_fur\_e not null,

data\_contract date constraint null\_fur\_da not null,

constraint pk\_fur primary key(id\_furnizor),

constraint unq\_fur\_e unique(email));



```
insert into furnizor  
values(30, 'furnizor1@gmail.ro', '01-IAN-06');
```

```
insert into furnizor  
values(50, 'furnizor2@gmail.ro', '01-MAI-09');
```

```
insert into furnizor  
values(67, 'furnizor3@gmail.ro', '21-FEB-16');
```

```
insert into furnizor  
values(81, 'furnizor4@gmail.ro', '31-DEC-20');
```

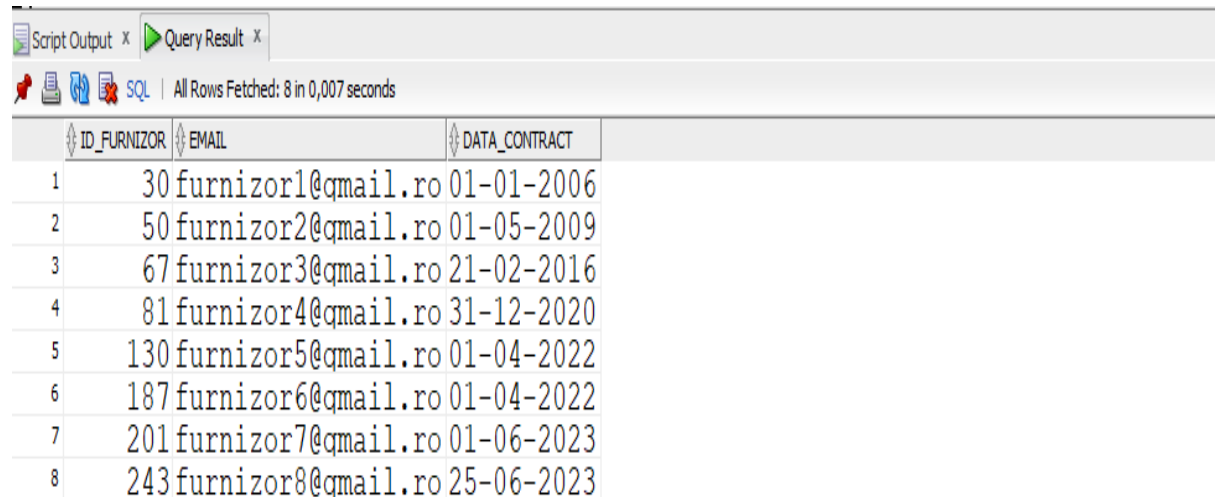
```
insert into furnizor  
values(130, 'furnizor5@gmail.ro', '01-APR-22');
```

```
insert into furnizor  
values(187, 'furnizor6@gmail.ro', '01-APR-22');
```

```
insert into furnizor  
values(201, 'furnizor7@gmail.ro', '01-IUN-23');
```

```
insert into furnizor  
values(243, 'furnizor8@gmail.ro', '25-IUN-23');
```

```
select * from furnizor;
```



ID_FURNIZOR	EMAIL	DATA_CONTRACT
1	30 furnizor1@gmail.ro	01-01-2006
2	50 furnizor2@gmail.ro	01-05-2009
3	67 furnizor3@gmail.ro	21-02-2016
4	81 furnizor4@gmail.ro	31-12-2020
5	130 furnizor5@gmail.ro	01-04-2022
6	187 furnizor6@gmail.ro	01-04-2022
7	201 furnizor7@gmail.ro	01-06-2023
8	243 furnizor8@gmail.ro	25-06-2023

- Tabel interactioneaza

```
create table interactioneaza( id_client number(4),  
id_angajat number(4),  
constraint pk_in primary key(id_client, id_angajat),  
constraint fk_in_cli foreign key(id_client)  
references client(id_client) ON DELETE CASCADE,  
constraint fk_in_an foreign key(id_angajat)  
references angajat(id_angajat) ON DELETE CASCADE);
```

```
insert into interactioneaza  
values (12, 11);
```

```
insert into interactioneaza  
values (12, 21);
```

insert into interactioneaza  
values (12, 28);

insert into interactioneaza  
values (214, 11);

insert into interactioneaza  
values (12, 7);

insert into interactioneaza  
values (300, 11);

insert into interactioneaza  
values (300, 7);

insert into interactioneaza  
values (300, 21);

insert into interactioneaza  
values (53, 28);

insert into interactioneaza  
values (53, 23);

insert into interactioneaza  
values (21, 2);

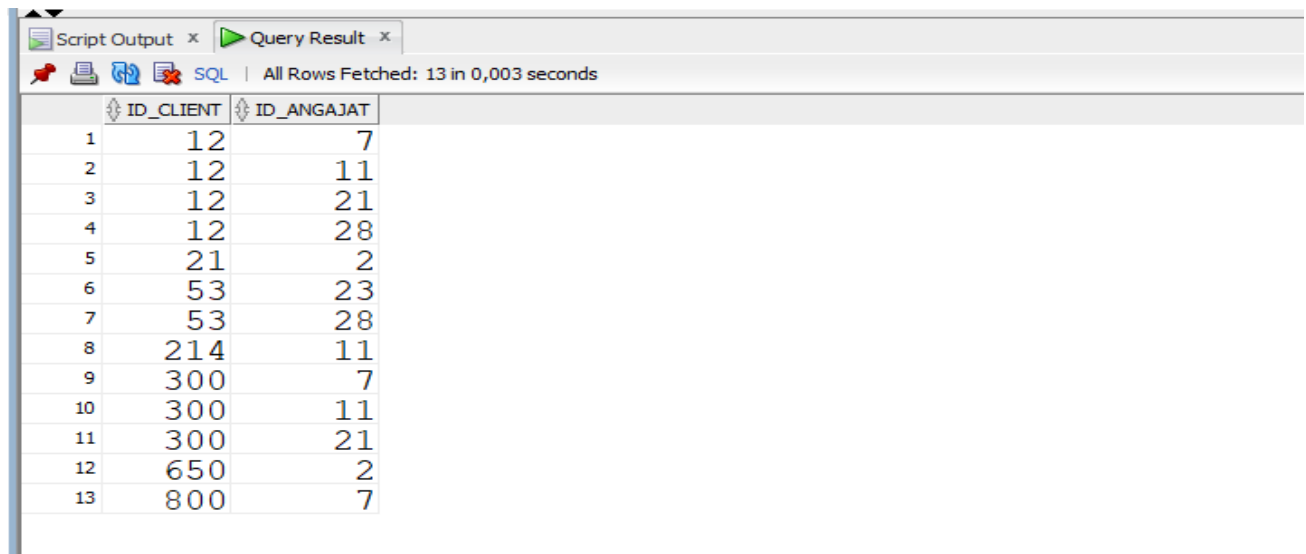
insert into interactioneaza

values (650, 2);

insert into interactioneaza

values (800, 7);

select \* from interactioneaza;



	ID_CLIENT	ID_ANGAJAT
1	12	7
2	12	11
3	12	21
4	12	28
5	21	2
6	53	23
7	53	28
8	214	11
9	300	7
10	300	11
11	300	21
12	650	2
13	800	7

- Tabel contine

```
create table contine(id_comanda number(4),  
id_produș number(4),  
constraint pk_con primary key(id_comanda, id_produș),  
constraint fk_con_co foreign key(id_comanda)  
references comanda(id_comanda) ON DELETE CASCADE,  
constraint fk_con_pro foreign key(id_produș)  
references produș(id_produș)ON DELETE CASCADE);
```

insert into contine

values (23, 10);

insert into contine  
values (23, 20);

insert into contine  
values (23, 15);

insert into contine  
values (31, 12);

insert into contine  
values (31, 15);

insert into contine  
values (31, 16);

insert into contine  
values (31, 20);

insert into contine  
values (63, 15);

insert into contine  
values (63, 20);

insert into contine  
values (137, 15);

insert into contine

values (137, 25);

insert into contine

values (137, 31);

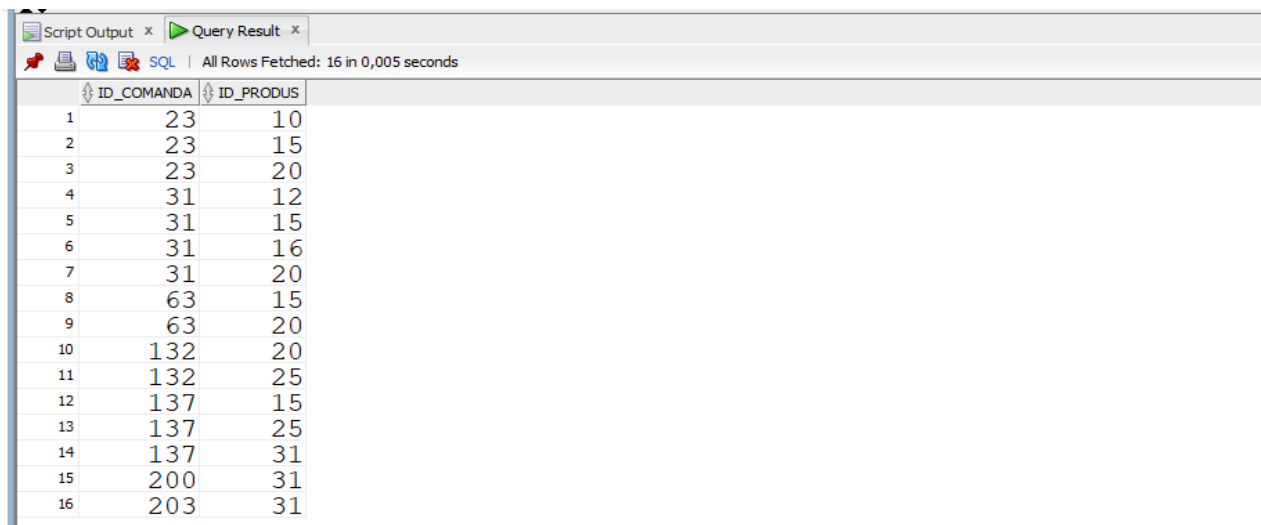
insert into contine

values (132, 20);

insert into contine

values (132, 25);

select \* from contine;



The screenshot shows a SQL query result window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying a table with two columns: 'ID\_COMANDA' and 'ID\_PRODUS'. The table contains 16 rows of data. The status bar indicates 'All Rows Fetched: 16 in 0,005 seconds'.

	ID_COMANDA	ID_PRODUS
1	23	10
2	23	15
3	23	20
4	31	12
5	31	15
6	31	16
7	31	20
8	63	15
9	63	20
10	132	20
11	132	25
12	137	15
13	137	25
14	137	31
15	200	31
16	203	31

- Tabel adus

create table adus(id\_produs number(4),

id\_furnizor number(4),

```
constraint pk_a primary key(id_produ, id_furnizor),  
constraint fk_a_pro foreign key(id_produ)  
references produ(id_produ) ON DELETE CASCADE,  
constraint fk_a_fur foreign key(id_furnizor)  
references furnizor(id_furnizor) ON DELETE CASCADE);
```

```
insert into adus  
values (10, 30);
```

```
insert into adus  
values (10, 50);
```

```
insert into adus  
values (12, 30);
```

```
insert into adus  
values (12, 50);
```

```
insert into adus  
values (12, 81);
```

```
insert into adus  
values (20, 67);
```

```
insert into adus  
values (20, 130);
```

```
insert into adus  
values (16, 50);
```

```
insert into adus  
values (16, 130);
```

```
insert into adus  
values (12, 67);
```

```
insert into adus  
values (15, 130);
```

```
insert into adus  
values (22, 67);
```

```
insert into adus  
values (31, 201);
```

```
select * from adus;
```

	ID_PRODUS	ID_FURNIZOR
1	10	30
2	10	50
3	12	30
4	12	50
5	12	67
6	12	81
7	15	130
8	16	50
9	16	130
10	20	67
11	20	130
12	22	67
13	31	201



## 12. CERERILE SQL

-- subcereri corelate in care intervin 3 tabele + folosire case

-- (numele clientilor , alaturi de numarul lor de rezervari si comenzi , precizand tipul clientului - fidel sau nu)

```
SELECT distinct c.num_e_client,
```

```
    (SELECT COUNT(r.id_client) FROM Rezervare r WHERE r.id_client = c.id_client) AS  
numar_rezervari,
```

```
    (SELECT COUNT(co.id_client) FROM Comanda co WHERE co.id_client = c.id_client) AS  
numar_comenzi,
```

```
    CASE
```

```
        WHEN (SELECT COUNT(r.id_client) FROM Rezervare r WHERE r.id_client = c.id_client)  
>= 2
```

```
        THEN 'Client fidel :)'
```

```
        ELSE 'Putine rezervari :('
```

```
    END AS tip_client
```

```
FROM Client c, Rezervare r
```

```
where c.id_client = r.id_client
```

```
order by 2 desc, 3 desc;
```

Worksheet Query Builder

```
-- 1
-- subcereri corelate in care intervin 3 tabele + folosire case
-- (numele clientilor , alaturi de numarul lor de rezervari si comenzi , precizand tipul clientului - fidel sau nu, pentru clientii
-- ce au facut macar 1 rezervare)
SELECT distinct c.num_client,
      (SELECT COUNT(r.id_client) FROM Rezervare r WHERE r.id_client = c.id_client) AS numar_rezervari,
      (SELECT COUNT(co.id_client) FROM Comanda co WHERE co.id_client = c.id_client) AS numar_comenzi,
      CASE
        WHEN (SELECT COUNT(r.id_client) FROM Rezervare r WHERE r.id_client = c.id_client) >= 2
        THEN 'Client fidel :)'
        ELSE 'Putine rezervari :('
      END AS tip_client
FROM Client c, Rezervare r
where c.id_client = r.id_client
order by 2 desc, 3 desc;
```

Script Output x Query Result x

SQL All Rows Fetched: 6 in 0,005 seconds

	NUM_CLIENT	NUMAR_REZERVARI	NUMAR_COMENZI	TIP_CLIENT
1	Popescu	2	2	Client fidel :)
2	Vasile	2	2	Client fidel :)
3	Robu	2	1	Client fidel :)
4	Barbu	2	0	Client fidel :)
5	Ionescu	1	2	Putine rezervari :(
6	Spiridon	1	0	Putine rezervari :(

-- subcerere nesincronizata in clauza de from(2) + 2 functii pe date calendaristice (sysdate+extract)

-- (afiseaza clientii care au facut mai mult de 1 rezervare in ultima jumatate de an si numarul lor de rezervari)

SELECT c.num\_client, r.numar\_rezervari

FROM Client c

JOIN (

SELECT id\_client, COUNT(\*) AS numar\_rezervari, data\_cazare

FROM Rezervare

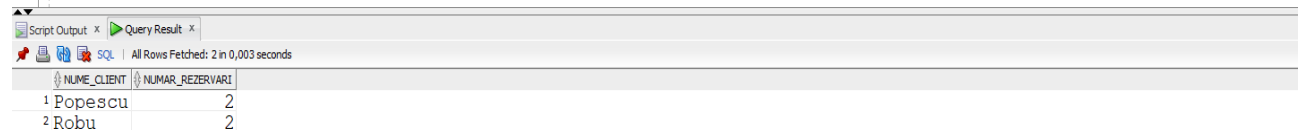
GROUP BY id\_client, data\_cazare

order by 2 desc, 3 asc

) r ON c.id\_client = r.id\_client

WHERE r.numar\_rezervari >= 2 and extract(month from sysdate) - extract(month from r.data\_cazare) <= 6 ;

```
-- 2
-- subcerere nesincronizata in clauza de from(2) + 2 functii pe date calendaristice (sysdate+extract)
-- (afiseaza clientii care au facut mai mult de 1 rezervare in ultima jumatate de an si numarul lor de rezervari)
SELECT c.numa_client, r.numar_rezervari
FROM Client c
JOIN (
  SELECT id_client, COUNT(*) AS numar_rezervari, data_cazare
  FROM Rezervare
  GROUP BY id_client, data_cazare
  order by 2 desc, 3 asc
) r ON c.id_client = r.id_client
WHERE r.numar_rezervari >= 2 and extract(month from sysdate) - extract(month from r.data_cazare) <= 6 ;
```



	NUME_CLIENT	NUMAR_REZERVARI
1	Popescu	2
2	Robu	2

-- ordonari + nvl si decode(4) + 2 functii pe siruri de caractere(concat, to\_char)

-- (clientii , alaturi de data comenzilor acestora si codul produselor comandate - inclusiv clientii ce nu au dat comenzi)

```
SELECT concat(c.id_client, c.numa_client) as client, NVL(to_char(co.data_comanda), 'Neprecizata')
AS data_comanda,
```

```
DECODE(ccp.id_produs, Null, 'Nedefinit', ccp.id_produs) AS id_produs
```

```
FROM Client c
```

```
LEFT JOIN Comanda co ON c.id_client = co.id_client
```

```
LEFT JOIN Contine ccp ON co.id_comanda = ccp.id_comanda
```

```
ORDER BY co.data_comanda ASC, ccp.id_produs desc;
```

```
-- 4
-- ordonari + nvl si decode(4) + 2 functii pe siruri de caractere(concat, to_char)
-- (clientii , alaturi de data comenzilor acestora si codul produselor comandate - inclusiv clientii ce nu au dat comenzi)
SELECT concat(c.id_client, c.num_client) AS client, NVL(to_char(co.data_comanda), 'Neprecizata') AS data_comanda,
DECODE(ccp.id_produs, Null, 'Nedefinit', ccp.id_produs) AS id_produs
FROM Client c
LEFT JOIN Comanda co ON c.id_client = co.id_client
LEFT JOIN Contine ccp ON co.id_comanda = ccp.id_comanda
ORDER BY co.data_comanda ASC, ccp.id_produs desc;
```

	CLIENT	DATA_COMANDA	ID_PRODUS
1	12Ionescu	12-01-2019	20
2	12Ionescu	12-01-2019	15
3	12Ionescu	12-01-2019	10
4	12Ionescu	14-01-2019	20
5	12Ionescu	14-01-2019	16
6	12Ionescu	14-01-2019	15
7	12Ionescu	14-01-2019	12
8	53Robu	15-05-2019	20
9	53Robu	15-05-2019	15
10	300Popescu	12-08-2020	25
11	300Popescu	12-08-2020	20
12	300Popescu	15-08-2020	31
13	300Popescu	15-08-2020	25
14	300Popescu	15-08-2020	15
15	800Vasile	15-06-2023	31
16	800Vasile	16-06-2023	31
17	210Barbu	Neprecizata	Nedefinit
18	531Enache	Neprecizata	Nedefinit
19	21Griqore	Neprecizata	Nedefinit
20	650Zamfir	Neprecizata	Nedefinit

-- grupări de date cu subcereri nesincronizate in care intervin 3 tabele,

-- functii grup(count, avg), filtrare la nivel de grupuri(having)

----- (numarul de produse(nu distincte) comandat de fiecare client ce a comandat

-- un numar de produse mai mare de numarul mediu de produse comandate-- si nr total al clientului respectiv de comenzi)

```
SELECT c.num_client, sub.numar_produce, sub.numar_comenzi
```

```
FROM Client c
```

```
JOIN (
```

```
SELECT co.id_client, COUNT(p.id_produs) AS numar_produce, count(distinct
co.id_comanda) as numar_comenzi
```

```
FROM Comanda co
```

```
JOIN Contine p ON co.id_comanda = p.id_comanda
```

```
GROUP BY co.id_client
```

```
HAVING COUNT(p.id_produs) >= (SELECT avg(COUNT(p.id_produs))
```

```
FROM Comanda co JOIN Contine p ON co.id_comanda =
p.id_comanda
```

```
GROUP BY co.id_client) -- media numarului de produse
```

```
) sub ON c.id_client = sub.id_client
```

```
ORDER BY sub.numar_produce DESC, sub.numar_comenzi desc;
```

```
-- 5
-- grupări de date cu subcereri nesincronizate in care intervin 3 tabele, |
-- functii grup(count, avg), filtrare la nivel de grupuri(having)
----- (numarul de produse(nu distincte) comandat de fiecare client ce a comandat
-- un numar de produse mai mare de numarul mediu de produse comandate-- si nr total al clientului respectiv de comenzi
=SELECT c.num_client, sub.numar_produce, sub.numar_comenzi
FROM Client c
JOIN (
= SELECT co.id_client, COUNT(p.id_produ) AS numar_produce, count(distinct co.id_comanda) as numar_comenzi
FROM Comanda co
JOIN Contine p ON co.id_comanda = p.id_comanda
GROUP BY co.id_client
HAVING COUNT(p.id_produ) >= (SELECT avg(COUNT(p.id_produ))
FROM Comanda co JOIN Contine p ON co.id_comanda = p.id_comanda
GROUP BY co.id_client) -- media numarului de produse
) sub ON c.id_client = sub.id_client
ORDER BY sub.numar_produce DESC, sub.numar_comenzi desc;
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0,004 seconds

	NUME_CLIENT	NUMAR_PRODUSE	NUMAR_COMENZI
1	Ionescu	7	2
2	Popescu	5	2

-- folosire with+nvl

-- (afisarea camerelor insotite de numarul lor de rezervari si plati neachitate)

WITH camere\_populare AS (

SELECT numar\_camera, COUNT(\*) AS nr\_rezervari

FROM Rezervare

GROUP BY numar\_camera

ORDER BY nr\_rezervari DESC

), camere\_platite AS (

SELECT r.numar\_camera, COUNT(\*) AS nr\_plati

FROM Rezervare r

INNER JOIN Plata p ON r.id\_rezervare = p.id\_rezervare

WHERE p.statut = 'platita'

GROUP BY r.numar\_camera

)

SELECT cp.numar\_camera, cp.nr\_rezervari, nvl(cp.nr\_rezervari, 0) - nvl(cpp.nr\_plati, 0) AS  
nr\_neplatite

FROM camere\_populare cp

LEFT JOIN camere\_platite cpp ON cp.numar\_camera = cpp.numar\_camera

order by 2 desc;

```
-- 6
-- folosire with+nvl (6)
-- (afisarea camerelor insotite de numarul lor de rezervari si plati neachitate)
WITH camere_populare AS (
    SELECT numar_camera, COUNT(*) AS nr_rezervari
    FROM Rezervare
    GROUP BY numar_camera
    ORDER BY nr_rezervari DESC
), camere_platite AS (
    SELECT r.numar_camera, COUNT(*) AS nr_plati
    FROM Rezervare r
    JOIN Plata p ON r.id_rezervare = p.id_rezervare
    WHERE p.statut = 'platita'
    GROUP BY r.numar_camera
)
SELECT cp.numar_camera, cp.nr_rezervari, nvl(cp.nr_rezervari, 0) - nvl(cpp.nr_plati, 0) AS nr_neplatite
FROM camere_populare cp
LEFT JOIN camere_platite cpp ON cp.numar_camera = cpp.numar_camera
order by 2 desc,3;
```

Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0,004 seconds

	NUMAR_CAMERA	NR_REZERVARI	NR_NEPLATITE
1	832	3	1
2	111	3	2
3	421	1	0
4	500	1	0
5	23	1	1
6	701	1	1

### 13. OPERATILE DE ACTUALIZARE SI SUPRIMARE

-- update

-- crestere cu 10% a celor mai comandate camere

UPDATE Camera

SET pret = pret \* 1.1

WHERE numar\_camera IN (

select numar\_camera from (

SELECT r.numar\_camera, COUNT(\*) AS numar\_rezervari

FROM Rezervare r

GROUP BY r.numar\_camera

HAVING COUNT(\*) = (

```

SELECT MAX(numar_rezervari)
FROM (
    SELECT numar_camera, COUNT(*) AS numar_rezervari
    FROM Rezervare
    GROUP BY numar_camera
    ) rezervari)
));

```

Inainte:

Script Output x Query Result x			
SQL   All Rows Fetched: 9 in 0,011 seconds			
	NUMAR_CAMERA	TIP_CAMERA	PRET
1	111	single	350
2	23	single	280
3	51	single	300
4	421	dubla	390
5	500	dubla	420
6	502	dubla	425
7	701	tripla	590
8	810	apartament	630
9	832	apartament	685

Dupa:

```

----- ex 13
-- update
-- crestere cu 10% a celor mai comandate camere
UPDATE Camera
SET pret = pret * 1.1
WHERE numar_camera IN (
    select numar_camera from (
        SELECT r.numar_camera, COUNT(*) AS numar_rezervari
        FROM Rezervare r
        GROUP BY r.numar_camera
        HAVING COUNT(*) = (
            SELECT MAX(numar_rezervari)
            FROM (
                SELECT numar_camera, COUNT(*) AS numar_rezervari
                FROM Rezervare
                GROUP BY numar_camera
            ) rezervari)) );
select * from camera;
--rollback;

```

	NUMAR_CAMERA	TIP_CAMERA	PRET
1	111	single	385
2	23	single	280
3	51	single	300
4	421	dubla	390
5	500	dubla	420
6	502	dubla	425
7	701	tripla	590
8	810	apartament	630
9	832	apartament	754

-- delete

-- sterge angajatii care nu au interactionat cu clienti si sunt angajati de mai mult de 30 de zile

DELETE FROM Angajat

WHERE id\_angajat NOT IN (SELECT DISTINCT id\_angajat FROM Interactioneaza)  
and (sysdate - data\_angajarii) > 30;

Inainte:

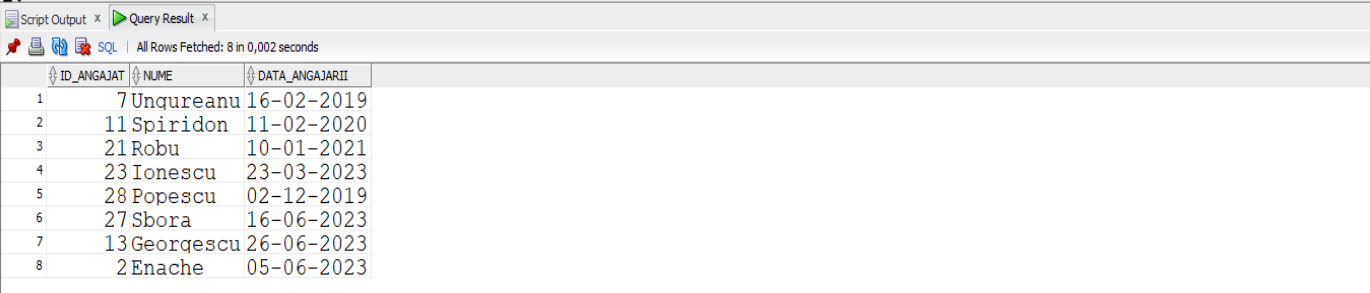
ID_ANGAJAT	NUME	DATA_ANGAJARII
7	Unquareanu	16-02-2019
11	Spiridon	11-02-2020
21	Robu	10-01-2021
23	Ionescu	23-03-2023
28	Popescu	02-12-2019
20	Marcu	02-05-2023
49	Mircea	02-12-2022
27	Sbora	16-06-2023
13	Georgescu	26-06-2023
2	Enache	05-06-2023



Dupa:

```
-- delete
-- sterge angajatii care nu au interactionat cu clienti si sunt angajati de mai mult de 30 de zile
DELETE FROM Angajat
WHERE id_angajat NOT IN (SELECT DISTINCT id_angajat FROM Interactioneaza)
and (sysdate - data_angajarii) > 30;

rollback;
select * from angajat;
```



ID_ANGAJAT	NUME	DATA_ANGAJARII
7	Unquareanu	16-02-2019
11	Spiridon	11-02-2020
21	Robu	10-01-2021
23	Ionescu	23-03-2023
28	Popescu	02-12-2019
27	Sbora	16-06-2023
13	Georgescu	26-06-2023
2	Enache	05-06-2023

-- delete

-- sterge clientii care nu au achitat, si s-au cazat de mai mult de 3 saptamani

DELETE FROM Client

WHERE id\_client IN (

SELECT r.id\_client

FROM Rezervare r

JOIN Plata p ON r.id\_rezervare = p.id\_rezervare

WHERE p.statut = 'neplatita'

AND sysdate - r.data\_cazare > 21

);

Inainte:

ID_CLIENT	NUME_CLIENT	NUMAR_TELEFON	EMAIL
1	12 Ionescu	0711111111	ion@gmail.io
2	53 Robu	0766783086	robu@gmail.ro
3	214 Spiridon	0748253119	spiridon@gmail.sp
4	300 Popescu	0722277789	popescu@gmail.po
5	531 Enache	0723021298	enache@gmail.en
6	800 Vasile	0723021291	vasile@gmail.va
7	210 Barbu	0733021298	barbu@gmail.ba
8	21 Griqore	0723321298	griqore@gmail.qr
9	650 Zamfir	0723121298	zamfir@gmail.za

Dupa:

```
-- delete
-- sterge clientii care nu au achitat, si s-au cazat de mai mult de 3 saptamani
DELETE FROM Client
WHERE id_client IN (
    SELECT r.id_client
    FROM Rezervare r
    JOIN Plata p ON r.id_rezervare = p.id_rezervare
    WHERE p.statut = 'neplatita'
    AND sysdate - r.data_cazare > 21
);

select * from client;
```

ID_CLIENT	NUME_CLIENT	NUMAR_TELEFON	EMAIL
1	53 Robu	0766783086	robu@gmail.ro
2	531 Enache	0723021298	enache@gmail.en
3	210 Barbu	0733021298	barbu@gmail.ba
4	21 Griqore	0723321298	griqore@gmail.qr
5	650 Zamfir	0723121298	zamfir@gmail.za