Hotel Booking Management System

*Introducere :*

În prezent, companiile hoteliere au posibilitatea de a-și gestiona clienții foarte bine datorită introducerii sistemelor de gestionare a rezervarilor si a clientilor lor , unde acestia retin date depre clineti , agenti de turism , camere si plati .

*Reguli de funtionare :*

O rezervare este facuta la un singur hotel , pe cand un hotel poate avea mai multe rezervari .

O rezervare poate tine de una sau mai multe camere , pe cand o camera trebuie sa tina de o rezervare .

Unei rezervari trebuie sa ii apartina o plata , iar o plata trebuie sa fie facuta pentru o rezervare.

La un hotel trebuie exista mai multe camere , dar o camera trebuie sa existe intr-un hotel .

Mai multi clienti, pot solicita una sau mai multe rezervari, apeland la mai multi agenti , dar doar o rezervare este facuta in aceeasi perioada , a unui singur client , luanduse in calcul doar un agent .

Fiecare camera poate avea unul sau mai multi angajati , iar acestia pot lucra in una sau mai multe camere.

Fiecare angajat are o pozitie , iar pe o pozitie pot fi mai multi angajati .

Fiecare camera apartine unui tip , iar un tip de trebuie sa apartina unei camere.

Un tip de camera poate avea unul sau mai multe tarife ..Un tarif trebuie sa apartina unei camera .

Fiecare plata trebuie sa aiba detalii, iar detaliile pot apartine mai multor plati.

*Entitati:*

Entitatile dependente sunt : REZERVARE, PLATA, CAMERA, TARIFE, ANGAJATI, restul sunt entitati independente.

CLIENT(GUEST) = persoana ce poate solicita una sau mai multe rezervari intr-un hotel, cheia primară este *guest\_id*

ANGENT DE TURSIM(RESERVATION AGENT)=perosana prin care clientii pot fac rezervari , cheia primară este *resAgent\_id*

REZERVARE(BOOKING)=este un document prin care clientul , cat si hotelul , are dovada inchirierii camerei si a platii pentru aceasta , cheie primare compusa : booking\_id, hotel\_id,guest\_id, resAgent\_id

HOTEL=locul unde clientii pot face rezervari , cheie primara hotel\_id

CAMERA(ROOM)=locul din hotel pe care clientii il inchiriaza pe o perioada de timp , iar angajatii isi profeseaza pozitia in hotel , cheie primara compusa , hotel\_id,staff\_id,room\_type\_id,booking\_id;

TIP CAMERA(ROOMTYPE)=camerele din hotel pot fi de mai multe tipuri , cheie primara room\_type\_id

TARIFE(RATES) = costul unei camera difera in in functie de tipul camerei , cheie primara compusa rate\_id, room\_type\_id

PLATA(PAYMENT)=transferul de bani dintre client si hotel , efectuat in urma unei rezervari , cheie primara compusa:payment\_id, booking\_if ,payment\_type\_id, payment\_status\_id;

DETALII PLATA(PAYMENTDETAILS)=o plata poate avea mai multe caracteristici, cheie primara compusa payment\_type\_id, payment\_status\_id;

ANGAJATI(STAFF)= persoana care lucreaza in hotel pe diferite pozitii pentru a deservi camerele . cheie primara compusa staff\_id, position\_id

POZITIE(POSITION)=meseria pe care o are un angajat , cheie primara position\_id

*Relatii :*

*CLIENT\_solicita\_AGENT\_pentru\_REZERVARE* = relaţie de tip 3 care leagă entităţile CLIENT, AGENT si REZERVARE , reflectând modul prin care rezervarile sunt efectuate

. Denumirea relației va fi *SOLICITA.*

REZERVARE\_apartine\_HOTEL=relatie care leaga entitatile REZERVARE si HOTEL, reflectând legătura dintre acestea (ce rezervari sunt intr-un hotel). Ea are

cardinalitatea minimă 0:1 şi cardinalitatea maximă M:1

REZERVARE\_tine\_de\_CAMERA= relatie care leaga entitatile REZERVARE si Camera, reflectând legătura dintre acestea (ce camera sunt rezervate ) ). Ea are

cardinalitatea minimă 1:0 şi cardinalitatea maximă 1:M .

PLATA\_apartine de \_REZERVARE= relatie care leaga entitatile REZERVARE si PLATA, reflectând legătura dintre acestea (ce rezervari sunt platite ). cardinalitatea maximă

1:1

HOTEL\_exista\_CAMERE= relatie care leaga entitatile HOTEL si CAMERA , reflectând legătura dintre acestea (ce camera exista in hotel). Ea are cardinalitatea minimă 1:1 şi cardinalitatea maximă 1:M

CAMERA\_lucreaza\_ANGAJAT= relatie care leaga entitatile CAMERA si ANGAJAT , reflectând legătura dintre acestea (ce angajati lucreaza pentru acea camera ). Ea are cardinalitatea minimă 0:0 şi cardinalitatea maximă M:M

ANGAJAt\_are\_POZITIE= relatie care leaga entitatile ANGAJAT si POZITIE , reflectând legătura dintre acestea (pe ce post lucreaza angajatul). Ea are cardinalitatea minimă 0:1 şi cardinalitatea maximă M:1

CAMERA\_apartine\_TIP CAMERA==relatie care leaga entitatile CAMERA si TIP CAMERA , reflectând legătura dintre acestea (ce tip de camera este ). Ea are cardinalitatea 1:1

TIP CAMERA\_are\_TARIF= relatie care leaga entitatile TIP CAMERA si TARIF , reflectând legătura dintre acestea (care este costul camerei in functie de tip ). Ea are cardinalitatea minimă 0:1 şi cardinalitatea maximă 1:M

PLATA\_are\_DETALII PLATA= relatie care leaga entitatile PLATA si DELATII PLATA, reflectând legătura dintre acestea (ce fel de plata este efectuata ). Ea are cardinalitatea minimă 0:1 şi cardinalitatea maximă 1:M

*Atribute:*

Entitatea dependentă TARIF are atributele:

* rate\_id=variabila de tip intreg care reprezintă codul unui tarif
* room\_type\_id= variabila de tip intreg , care reprezintă codul unui tip de camera . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TIP CAMERA
* rate=variabila de tip numar, de lungime maximă 5, cu 2 zecimale reprezinta pretul unei camerei

Entitatea independentă TIP CAMERA are atributele:

* room\_type\_id= variabila de tip intreg , care reprezintă codul unui tip de camera
* room\_type= variabilă de tip caracter, de lungime maximă 20 , care reprezintă tipul unei camera
* prim\_facility= variabilă de tip caracter, de lungime maximă 20, care reprezintă facilitatea dominanta

Entitatea independentă POZITIE are atributele:

* position\_id= variabila de tip intreg ,care reprezintă codulunei meserii
* position = variabilă de tip caracter, de lungime maximă 20, care reprezintă numele meseriei

Entitatea dependenta ANGAJAT are atributele:

* staff\_id=variabila de tip intreg ,, care reprezintă codul unui angajat
* position\_id= variabila de tip intreg , care reprezintă codul unei pozitii . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul POZITIE
* name= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele angajatului
* email= variabilă de tip caracter, de lungime maximă 20, care reprezintă emailul angajatului
* phone\_number= variabilă de tip caracter, de lungime maximă 15 care reprezintă numarul de telefon al angajatului

Entitatea dependenta CAMERA are atributele:

* room\_id= variabila de tip intreg ,, care reprezintă codul unei camere
* hotel\_id= variabila de tip intreg, care reprezintă codul hotelului din care face parte. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul HOTEL
* room\_type\_id= variabila de tip intreg , care reprezintă codul unui tip de camera. . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TIP CAMERA
* booking\_id= variabila de tip intreg , care reprezintă codul unei rezervari . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul REZERVARE
* room\_number= variabilă variabila de tip intreg ,de lungime maximă 3, care reprezintă numarul camerei
* floor= variabilă variabila de tip intreg ,de lungime maximă 2, care reprezintă etajul

Entitatea independentă HOTEL are atributele:

* hotel\_id= variabila de tip intreg care reprezintă codul hotelului
* hotel\_name= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele hotelului
* hotel\_code= variabila de tip intreg ,de lungime maximă 4, care reprezinta codul postalal hotelului
* city=variabilă de tip caracter, de lungime maximă 20, care reprezintă numele orasului in care se afla hotelul
* state=variabilă de tip caracter, de lungime maximă 20, care reprezintă numele statului in care se afla hotelul
* main\_phone\_number= variabilă de tip caracter, de lungime maximă 15, care reprezintă numarul de telefon al hotelului

Entitatea dependenta REZERVARE are atributele:

* booking\_id= variabila de tip intreg , care reprezintă codul unei rezervari
* room\_id= variabila de tip intreg , care reprezintă codul unei camera. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CAMERA
* hotel\_id= variabila de tip intreg , care reprezintă codul hotelului din care face parte. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul HOTEL
* resAgent\_id= variabila de tip intreg , care reprezintă codul unui agent de turism . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul AGENT DE TURISM
* guest\_id= variabila de tip intreg , care reprezintă codul unui client. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CLIENT
* room\_count= variabilă variabila de tip intreg ,de lungime maximă 2, care reprezintă numarul de camera dorite
* date\_from= variabilă de tip dată calendaristică, care reprezintă data inceperii sejurului
* date\_to= variabilă de tip dată calendaristică, care reprezintă data incheierii sejurului

Entitatea independent CLIENT are atributele:

* guest\_id= variabila de tip intreg , care reprezintă codul unui client.
* name= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele clientului
* email= variabilă de tip caracter, de lungime maximă 20, care reprezintă emailul clientului
* phone\_number= variabilă de tip caracter, de lungime maximă 15, care reprezintă numarul de telefon al clientului

Entitatea independent AGENT DE TURISM are atributele:

* resAgent\_id= variabila de tip intreg care reprezintă codul unui agent de turism .
* name= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele agentului
* email= variabilă de tip caracter, de lungime maximă 20, care reprezintă emailul agentului
* phone\_number= variabilă de tip caracter, de lungime maximă 15, care reprezintă numarul de telefon al agentului

Entitatea dependenta PLATA are atributele:

* payment\_id=variabila de tip intreg , care reprezintă codul unei plati
* booking\_id= variabila de tip intreg , care reprezintă codul unei rezervari. . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul REZERVARE
* payment\_type\_id= variabila de tip intreg care reprezintă codul unui tip de plata . . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul DETALII PLATA.
* date=variabilă de tip dată calendaristică, care reprezintă data efectuarii platii

Entitatea independent DETALII PLATA are atributele:

* payment\_type\_id= variabila de tip intreg care reprezintă codul unui tip de plata .
* payment\_type= variabilă de tip caracter, de lungime maximă 20, care reprezintă tipul platii
* payment\_status= variabilă variabila de tip bool , care reprezintă daca o rezervare a mai este activa sau nu

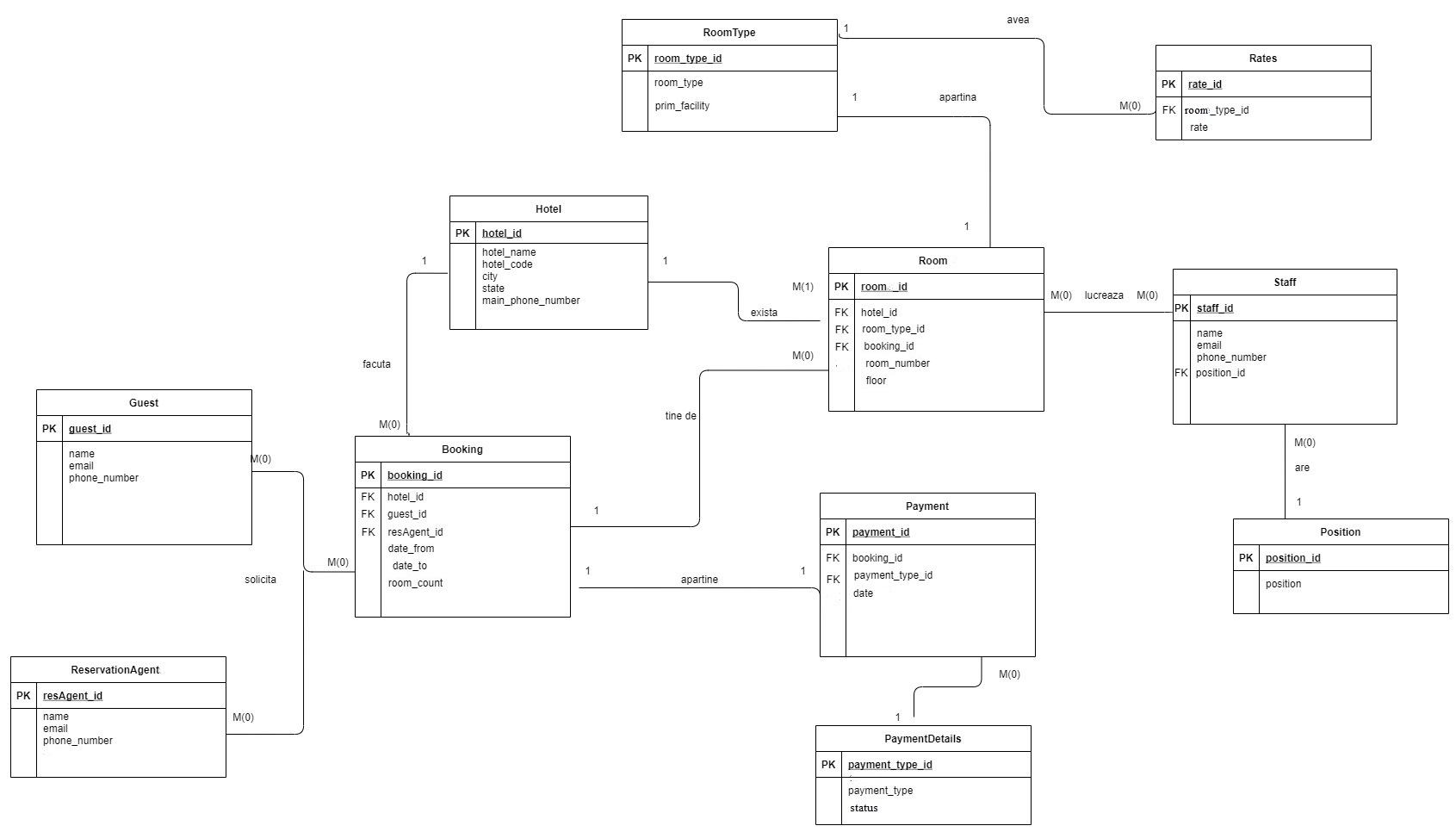
*Relatia client\_solicita\_agent\_pentru\_rezervare* are atributele:

* resAgent\_id= variabila de tip intreg , care reprezintă codul unui agent de turism . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul AGENT DE TURISM
* guest\_id= variabila de tip intreg , care reprezintă codul unui client. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CLIENT
* booking\_id= variabila de tip intreg , care reprezintă codul unei rezervari . Atributul trebuie să corespundă la o valoare a cheii primare din tabelul REZERVARE

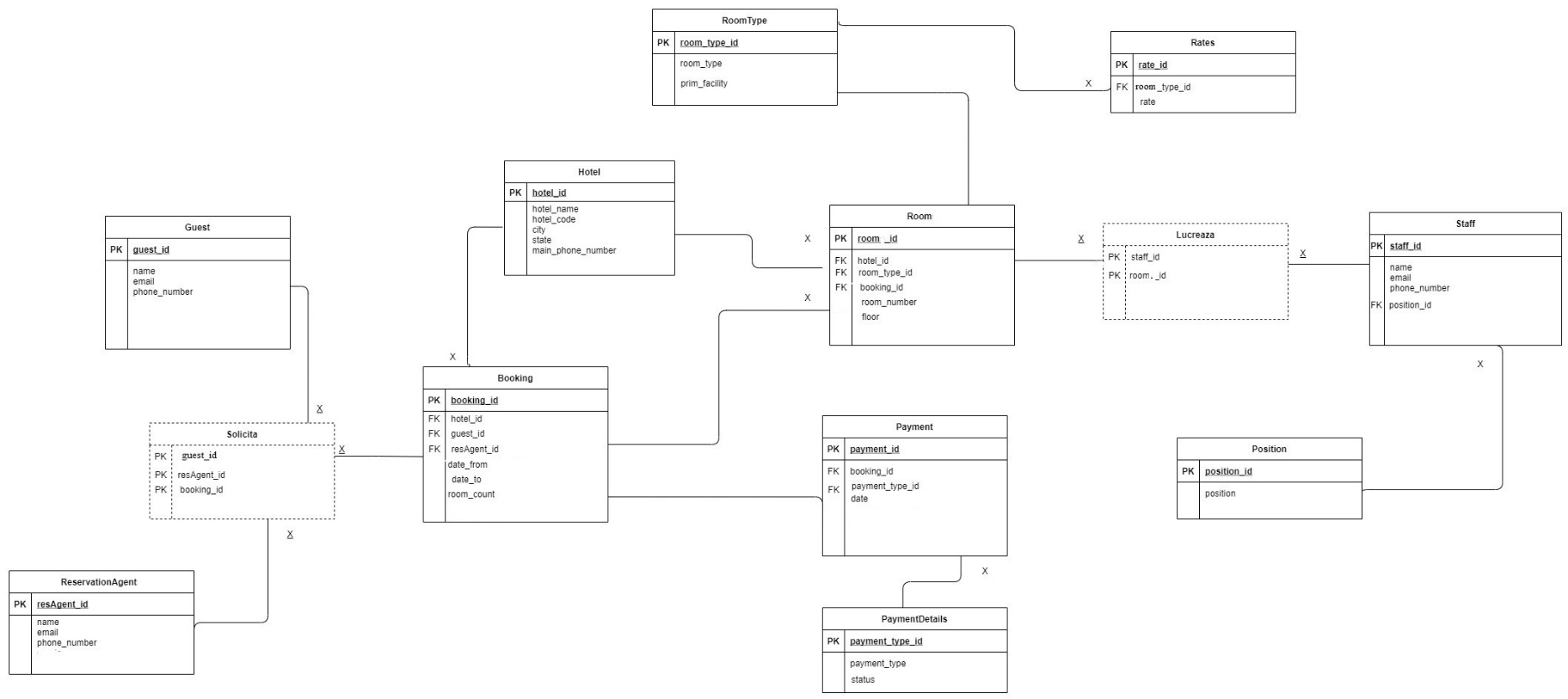
*Relatia* Camera\_lucreaza\_Angajatare atributele:

* staff\_id=variabila de tip intreg , care reprezintă codul unui angajat al camerei Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ANGAJAT
* room\_id= variabila de tip intreg , care reprezintă codul unei camera. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CAMERA

*Diagrama ER:*

**

–*Diagrama Conceptuala:*

**

*Scheme relaționale:*

TARIF(rate\_id#,room\_type\_id#, rate)

TIP CAMERA (room\_type\_id#,room\_type,prim\_facility)

POZITIE (position\_id#,position )

ANGAJAT (staff\_id#,position\_id#,name,,email,phone\_number)

CAMERA (room\_id#,hotel\_id#,room\_type\_id#,booking\_id#,room\_number,floor)

HOTEL (hotel\_id,hotel\_name,hotel\_code,city,state,main\_phone\_number)

REZERVARE (booking\_id#, room\_id#, hotel\_id#, resAgent\_id#, guest\_id#, room\_count, booking\_status,date\_from,date\_to)

CLIENT (guest\_id#, name,email, phone\_number)

AGENT DE TURISM (resAgent\_id#,name,email,phone\_number)

PLATA (payment\_id#,booking\_id#,payment\_type\_id#,date)

DETALII PLATA (payment\_type\_id#,payment\_type,payment\_status)

SOLICITA(resAgent\_id#,guest\_id#,booking\_id#)

LUCREAZA(staff\_id#,room\_id#)

*Crearea tabelelor:*

CREATE TABLE Guest(

guest\_id int Primary Key,

g\_name VARCHAR(20) NOT NULL ,

email VARCHAR(20) ,

phone\_number varchar(15) );

CREATE TABLE ReservationAgent(

resAgent\_id int PRIMARY KEY,

ra\_name VARCHAR(20) NOT NULL ,

email VARCHAR(20) ,

phone\_number varchar(15) );

CREATE TABLE Hotel(

hotel\_id int PRIMARY KEY,

hotel\_name VARCHAR(20) NOT NULL ,

hotel\_code number(4) not null,

city VARCHAR(20) not null ,

state VARCHAR(20) not null ,

main\_phone\_number varchar(15) );

CREATE TABLE RoomType(

room\_type\_id int PRIMARY KEY,

room\_type VARCHAR(20) ,

prim\_facility VARCHAR(20));

CREATE TABLE SPosition(

position\_id int PRIMARY KEY,

sposition VARCHAR(20) );

CREATE TABLE PaymentDetails(

payment\_type\_id int PRIMARY KEY,

payment\_type VARCHAR(20) ,

payment\_status number(1));

CREATE TABLE Rate(

rate\_id int PRIMARY KEY,

room\_type\_id int ,

FOREIGN KEY (room\_type\_id) REFERENCES RoomType(room\_type\_id),

rate decimal(5,2) );

CREATE TABLE Staff(

staff\_id int PRIMARY KEY,

position\_id int ,

FOREIGN KEY (position\_id) REFERENCES SPosition(position\_id),

s\_name VARCHAR(20) NOT NULL ,

email VARCHAR(20) ,

phone\_number varchar(15) );

CREATE TABLE Room(

room\_id int PRIMARY KEY,

hotel\_id int ,

FOREIGN KEY (hotel\_id) REFERENCES Hotel(hotel\_id),

room\_type\_id int ,

FOREIGN KEY (room\_type\_id) REFERENCES RoomType(room\_type\_id),

room\_number number(3),

floor number(2));

CREATE TABLE Booking(

booking\_id int PRIMARY KEY,

hotel\_id int ,

FOREIGN KEY (hotel\_id) REFERENCES Hotel(hotel\_id),

resAgent\_id int,

FOREIGN KEY (resAgent\_id) REFERENCES ReservationAgent(resAgent\_id),

guest\_id int,

FOREIGN KEY (guest\_id) REFERENCES Guest(guest\_id),

room\_count number(2) ,

data\_from DATE ,

data\_to DATE ,

CHECK (data\_to > data\_from) );

CREATE TABLE Payment(

payment\_id int PRIMARY KEY,

booking\_id int ,

FOREIGN KEY (booking\_id) REFERENCES Booking(booking\_id),

pdate DATE ,

payment\_type\_id int ,

FOREIGN KEY ( payment\_type\_id) REFERENCES PaymentDetails(payment\_type\_id));

ALTER TABLE Room

ADD booking\_id int;

ALTER TABLE Room

ADD FOREIGN KEY (booking\_id )

REFERENCES Booking(booking\_id );

commit;

Create table Solicita(

booking\_id int not null,

guest\_id int not null ,

resAgent\_id int not null,

FOREIGN KEY (booking\_id) REFERENCES Booking(booking\_id),

FOREIGN KEY (resAgent\_id) REFERENCES ReservationAgent(resAgent\_id),

FOREIGN KEY (guest\_id) REFERENCES Guest(guest\_id) ,

PRIMARY KEY( booking\_id, guest\_id, resAgent\_id )

);

Create table Lucreaza(

room\_id int ,

FOREIGN KEY (room\_id) REFERENCES Room(room\_id),

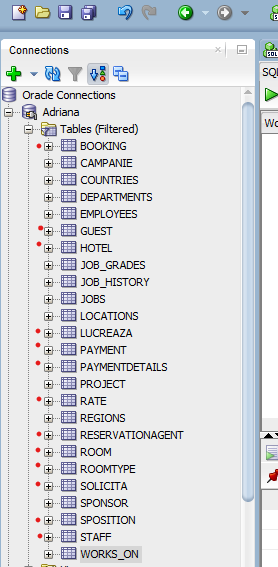
staff\_id int,

FOREIGN KEY (staff\_id) REFERENCES Staff(staff\_id),

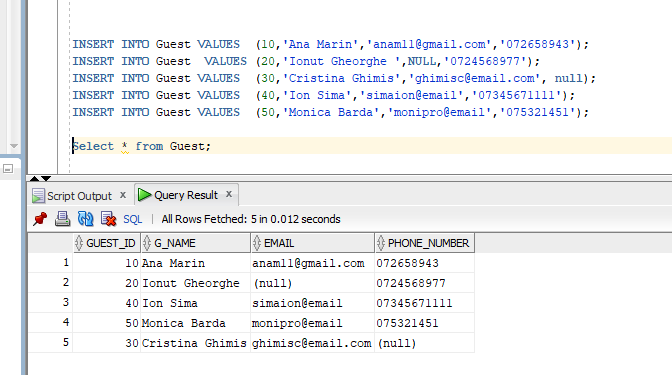
PRIMARY KEY( staff\_id,room\_id)

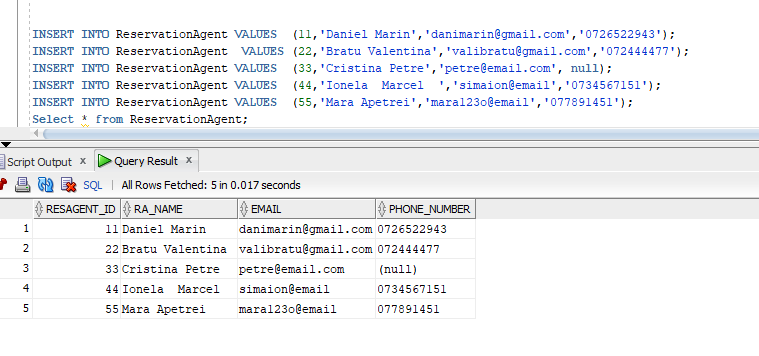
);

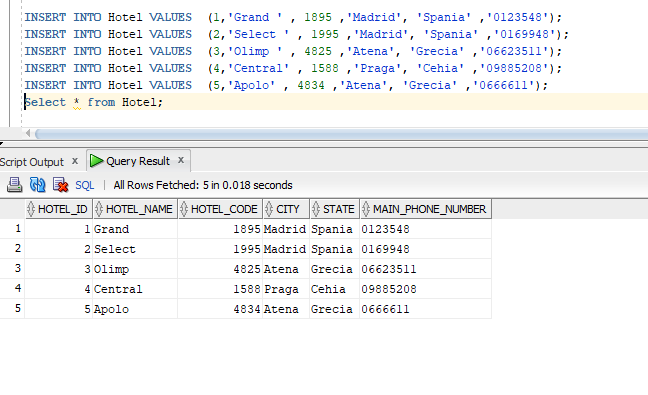
commit;

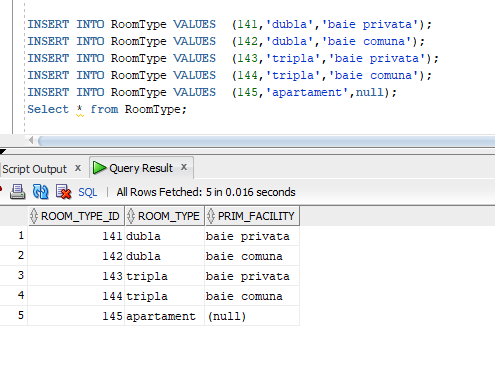


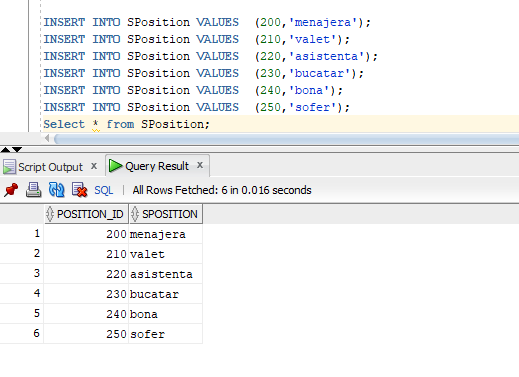
*Inserarea datelor in tabele :*

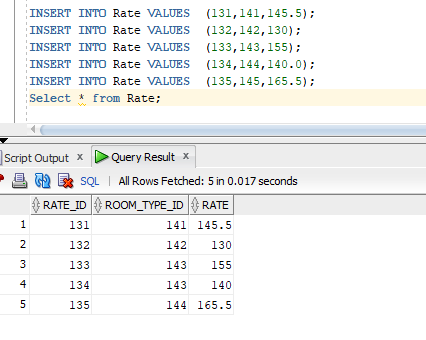


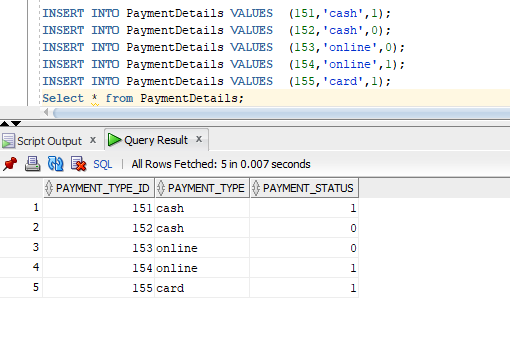


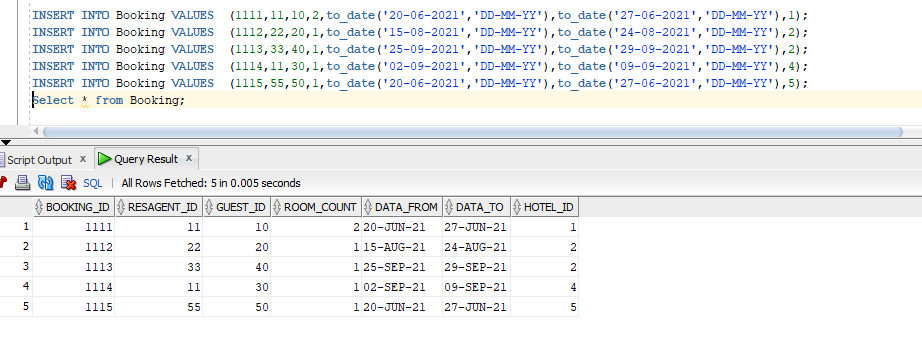


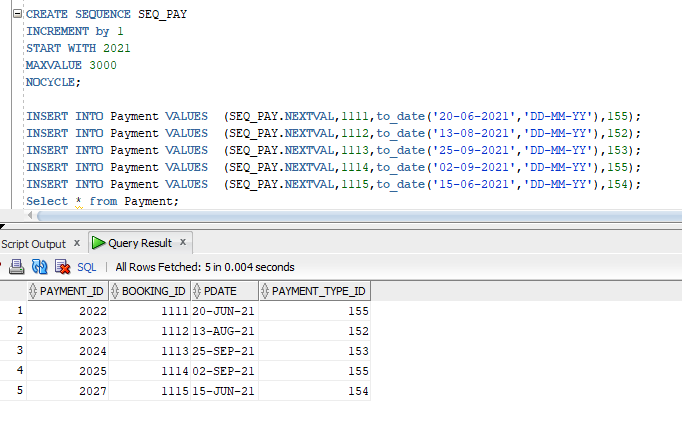


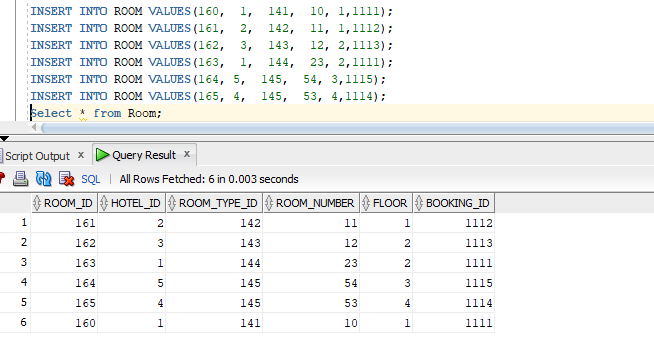


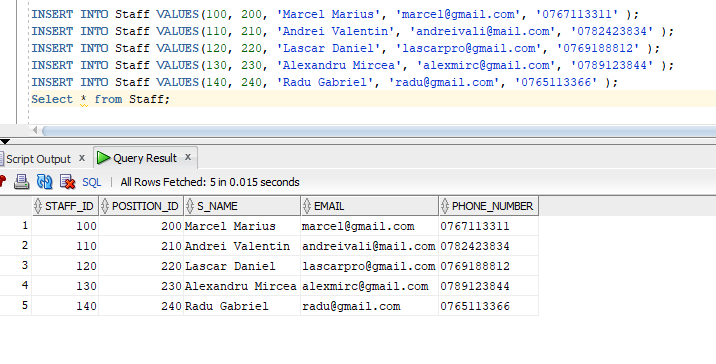


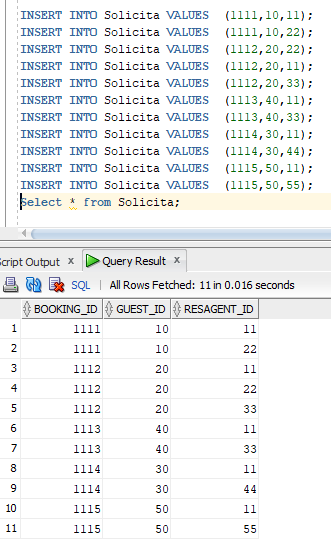


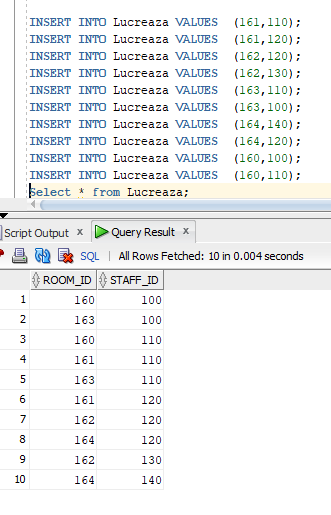












Ex11:

--1 Numele angajatilor care se ocupa de rezervarile facute dupa data de 15-AUG-2021

Select s\_name as Nume\_angajat

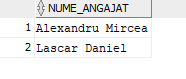
from staff s inner join lucreaza l

on s.staff\_id = l.staff\_id inner join room r

on r.room\_id = l.room\_id inner join booking b

on b.booking\_id = r.booking\_id

where b.data\_from>'15-AUG-2021';



-- 2 Sa se selecteze codul rezervarilor , codul clinetului si sa se verifice daca acesta are numar de telefon , daca are se va afisa "are numar de telefon

-- pentru rezervarile care au un tarif mai mare de 140

select booking\_id as cod\_rezervare, b.guest\_id as cod\_client , DECODE(phone\_number, null, 'Nu are numar de telefon ', 'are numar de telefon') as Verificare\_nr\_de\_contact

from booking b , guest g

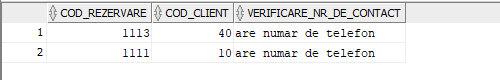
where b.booking\_id in (Select booking\_id from room v

where v.room\_type\_id in ( select room\_type\_id from RoomType r

where r.room\_type\_id in ( select room\_type\_id from Rate where rate>140)

))

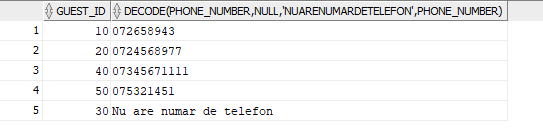
and b.guest\_id =g.guest\_id ;



--verificare daca merge decode

select guest\_id ,DECODE(phone\_number, null, 'Nu are numar de telefon ', phone\_number)

from guest;



--3 Sa se selecteze numele clientilor si emailul pentru cei ce au ales metoda de plata pentru rezervarea facuta , mai tarziu ( status=0 )

select unique( g\_name ) as nume\_client, nvl(email ,'Nu are email') as email

from guest g inner join solicita s

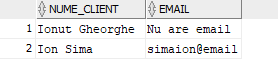
on s.guest\_id=g.guest\_id inner join booking b

on s.booking\_id=b.booking\_id inner join payment p

on p.booking\_id =b.booking\_id inner join paymentdetails pd

on p.payment\_type\_id=pd.payment\_type\_id

where pd.payment\_status=0;



--4 Sa se afiseze descrescator codul agentilor si numarul de rezervari (daca are mai mult de 2 sau o singura rezervare) pentru agentii ale caror

--rezervari facute au data de inceput mai mare decat data curenta si sejurl ( data de inceput - data incheierii) este mai mare de 5 zile

select resagent\_id ,case when count(booking\_id)>=2 THEN 'are mau mult de 2 rezervari facute ' else 'are o singura rezervare' end as Rezervari

from solicita

group by resagent\_id

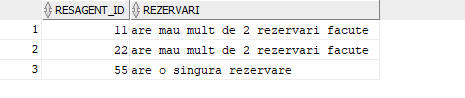
having resagent\_id in (select resagent\_id

from booking

where to\_char(data\_from,'dd/mm/yyyy') > TO\_CHAR(SYSDATE, 'MM-DD-YYYY') and data\_to-data\_from > 5

)

order by count(booking\_id) desc;



---5 sa se selecteze codul hotelului din Spania si tipul de camerei al camerelor care au fost rezervate cu

--tariful minim

with n1 as (

select hotel\_id

from hotel

where upper(state) like 'SPANIA' ) ,

n2 as (

select min(rate) as tarifminim ,room\_type\_id

from rate

group by room\_type\_id )

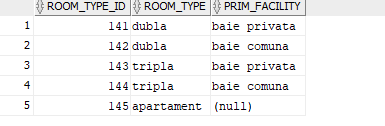
select hotel\_id , room\_type\_id from n1, n2

where tarifminim =(select min(rate) from rate)and (hotel\_id,room\_type\_id) in ( select hotel\_id, room\_type\_id from room) ;



**Ex 12:**

select \* from roomType;



Update roomType

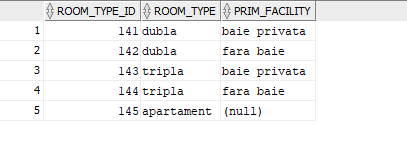
set prim\_facility ='fara baie'

where room\_type\_id in (select room\_type\_id

from room r inner join booking b

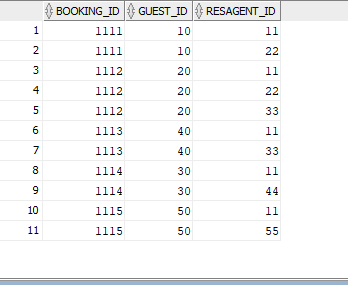
on b.booking\_id=r.booking\_id) and lower(prim\_facility)='baie comuna';

select \* from roomType;



rollback;

select \* from solicita;

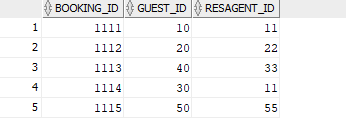


delete from solicita

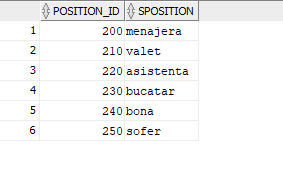
where (booking\_id,guest\_id,resAgent\_id ) not in (select booking\_id,guest\_id,resAgent\_id from booking );

select \* from solicita;

rollback;



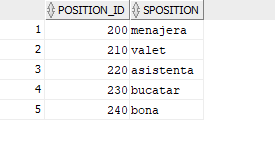
select \* from Sposition;



delete from Sposition

where position\_id not in ( select position\_id from staff);

select \* from Sposition;



rollback;

**Ex 16**:

--Sa se afiseze codul agentului,codul rezervarii, unde statsul platii este 0 , iar tipul de plata este cash.

SELECT DISTINCT b.resagent\_id ,b.booking\_id

FROM booking b

WHERE NOT EXISTS

(SELECT 1

FROM payment p inner join paymentDetails pd

on pd.payment\_type\_id =p.payment\_type\_id

where payment\_status=0 and LOWER(payment\_type) = 'cash'

AND NOT EXISTS

(SELECT 'x'

FROM booking b1

WHERE b1.booking\_id=b.booking\_id and b1.booking\_id=p.booking\_id));



-sa se afiseze codul angajatilor care lucreaza in toate camerele care au fost rezervate dupa luna a 9-a;

select staff\_id

FROM lucreaza

WHERE room\_id in

(SELECT room\_id

FROM room r inner join booking b

on b.booking\_id=r.booking\_id

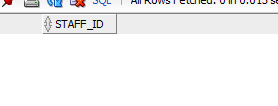
where TO\_CHAR(b.data\_from,'mm')>'09' )

GROUP BY staff\_id

HAVING COUNT(staff\_id)=

(SELECT COUNT(staff\_id)

FROM staff);



--sa se selecteze codul clientului, numele hotelului, tara si tipul camerei de acestia

Select distinct s.guest\_id , hotel\_name, state ,rt.room\_type

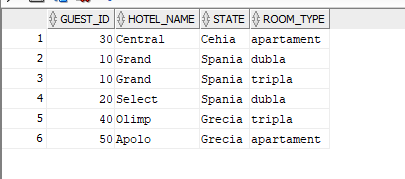
from roomType rt

left outer join room r on rt.room\_type\_id=r.room\_type\_id

left outer join hotel h on r.hotel\_id=h.hotel\_id

left outer join booking b on r.booking\_id=b.booking\_id

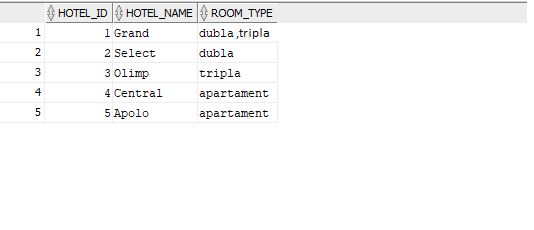
left outer join solicita s on s.booking\_id=b.booking\_id;



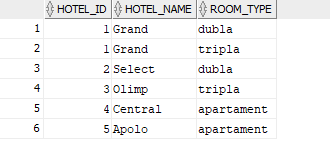
**Ex 9:**

*Forma normală 1:*

Contraexemplu:



FN1:



În contraexemplu, în relație fiecărui hotel\_id ii corespundeau mai multe tipuri de camera (room\_type) , astfel încât acele valori erau divizibile. Pentru a ajunge în FN1, am separat tipurile de camera.

*Forma normală 2- Forma normală 3:*

Am folosit un exemplu fictiv pentru acest exercițiu.

Contraexemplu:

Lucreaza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| camera\_id# | tip\_camera | angajat\_id# | functie\_angajat | salariu |
| 1 | dubla | 100 | menajera | 600 |
| 1 | dubla | 110 | bucatar | 500 |
| 2 | tripla | 110 | bucatar | 500 |
| 2 | tripla | 120 | bona | 400 |
| 3 | apartament | 100 | menajera | 600 |

Transformarea in FN2:

Lucreaza\_a

|  |  |  |  |
| --- | --- | --- | --- |
| camera\_id# | angajat\_id# | functie\_angajat | salariu |
| 1 | 100 | menajera | 600 |
| 1 | 110 | bucatar | 500 |
| 2 | 110 | bucatar | 500 |
| 2 | 120 | bona | 400 |
| 3 | 100 | menajera | 600 |

Lucreaza\_b

|  |  |
| --- | --- |
| camera\_id# | Tip\_camera |
| 1 | dubla |
| 2 | tripla |
| 3 | apartament |

-am aplicat regula lui Casey-Delobel pentru FN2 :

Astfel avem:

{camera \_id#} -> {tip\_camera} –codul camerei determina functional tipul acesteia

{ cod\_salariat#, angajat\_id# -> {functie\_angajat, salariu}

Pentru a aduce la FN3 se aplică regula Casey-Delobel. Relaţia Lucreaza\_a se descompune, prin eliminarea dependenţelor funcţionale tranzitive, în proiecţiile:

Transformarea in FN3:

Lucreaza\_2a

|  |  |  |
| --- | --- | --- |
| camera\_id# | angajat\_id# | functie\_angajat |
| 1 | 100 | menajera |
| 1 | 110 | bucatar |
| 2 | 110 | bucatar |
| 2 | 120 | bona |
| 3 | 100 | menajera |

Lucreaza\_2b

|  |  |
| --- | --- |
| functie\_angajat | salariu |
| menajera | 600 |
| bucatar | 500 |
| bona | 400 |

Lucreaza\_2a(camera\_id#, angajat\_id, functie\_angajat)

Lucreaza\_2b(, functie\_angajat , salariu)

**Ex 17:**

SQL:

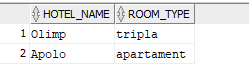
--sa se afiseze numele hotelului , tipul camerei pentru hotelurile din Grecia care au camerele la un etaj > 1.

select hotel\_name, rt.room\_type

from hotel h join room r on r.hotel\_id = h.hotel\_id

join roomtype rt on r.room\_type\_id=rt.room\_type\_id

where state='Grecia' and floor>1;



Expresie algebrica:

R1 = SELECT(HOTEL, state=Grecia)

R2 = PROJECT(R1,hotel\_id, hotel\_name)

R3 = SELECT(ROOM, floor>1)

R4=PROJECT(R3,room\_type\_id,hotel\_id)

R5=SEMIJOIN(R2,R4,hotel\_id)

R6=SEMIJOIN(R4, RoomType,room\_type\_id)

Rezultat=R7=PROJECT(R5,R6,hotel\_name, room\_type)

Arbore algebric:

