THE RITZ-CARLTON®

customer view

PROJECT DEFINITION

The Ritz-Carleton is a hotel that you can stay in . The hotel has different kinds of rooms which is single, double, and suite. Also, it provides some services, such as restaurant, and fitness center . This database will help the receptionists to book a room for the customer; and whatever services they want.

VIEW DESCRIPTION:

The Ritz-Carlton's database provides guests with a variety of services so that the time and effort are short and easy to deal with.
The hotel offers rooms, room type, duration of stay, reservations at the restaurant and fitness center ,all via the Database through some guest information such as ID, first and last name, phone number, etc.

DATA REQUIREMENTS:

Staff

The data required on members of the staff includes a unique staff numbers, name (first and last name), position, gender, date of birth (DOB), and name of the supervisor supervise an allocated group of staff (up to maximum of 10 an any one time).

Room for rent

The data stored on room for rent includes unique room ID, and a room number, room floor, room type witch indicate whether it is single, double, or suite booking a reservation is assigned to member of staff whether, it is checking in/out, a member of staff may mange maximum of 30 rooms an where is this relation.

Reservation

Once a guest decide to book a room a reservation is made, the reservation may or may not include a number of service. The information in the reservation includes booking number, the duration time, and type of payment.

Hotel guest

When a prospective guest make a reservation in the Ritz-Carlton hotel, the data stored includes the guest number, name (first and last name), telephone number, email, and some data on the desired room during book and order the services .

Branch

Each branch has one manger, and each manger manages one branch, the branch data includes, branch code, and the branch address (Street, city, and postcode), telephone number (up to maximum of three).

Services

Each serves may or may not be reserved by reserve many reservation, and each reservation may or may not have services. Services have service type, and unique service number. The service offers, fitness center, and restraint food.

TRANSACTION REQUIREMENTS:

Data Entry:

1. Enter type of room
2. Enter Booking date.
3. Enter payment method.

Data Update/Deletion:

1. Update/delete information of customers.

2.Update/delete service type.

3. Update the prices.
4. Update/delete the clients.
5. Update/delete data on staff.
6. Update/delete data on branches

Data Queries:

1.List name of customer

2. List number of branch.
3. Display phone number of branch
4. List number the room are reserved
5. Display service type

6. Display a price of room.

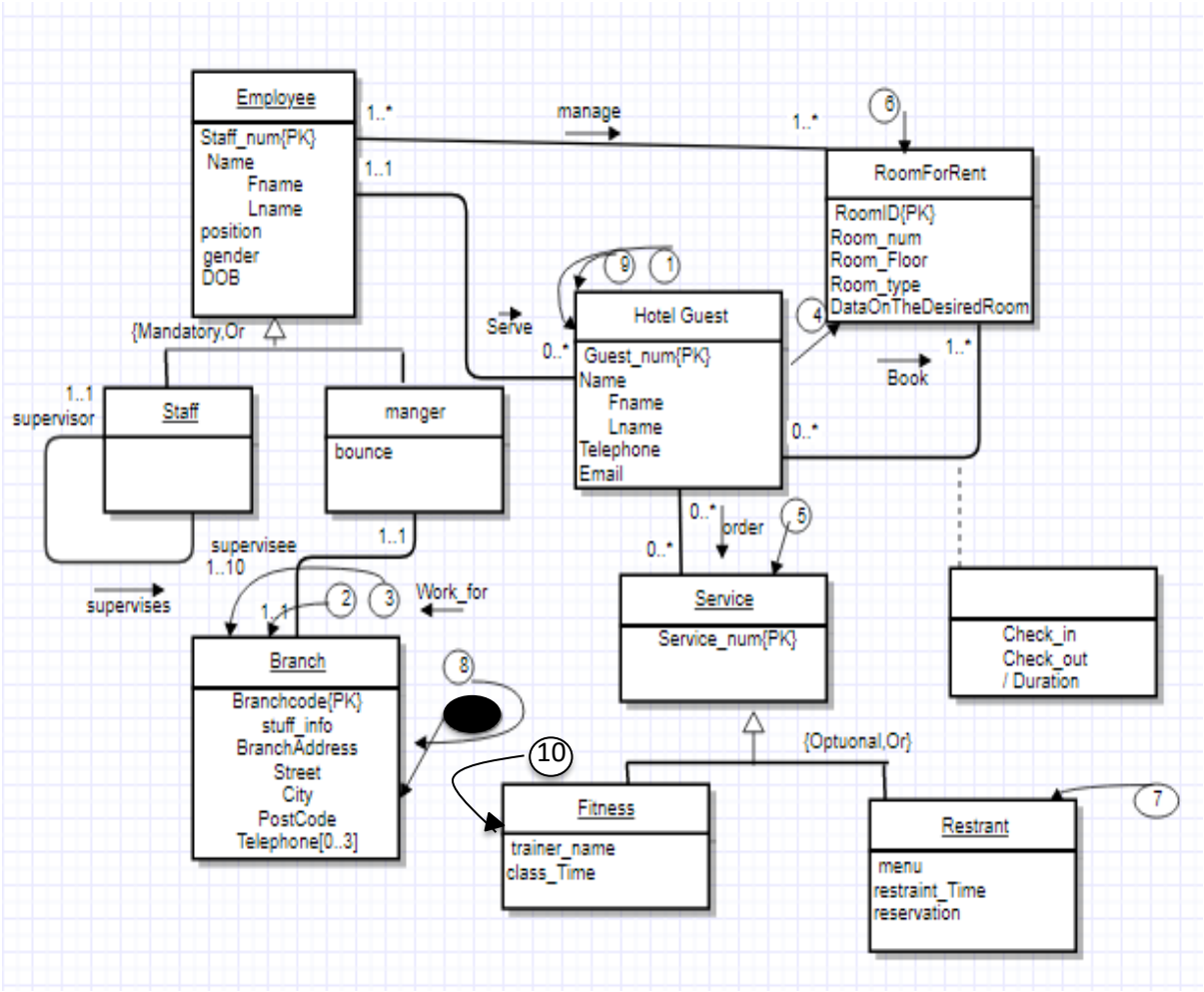
7. Display time of hotel's restaurant.

8. Display email of hotel's

9. Display the details of client by third national ID.

10. Display Class Time .

TRANSACTION PATHWAY :



RELATION SCHEMA :

Staff(staff_num,supervisorID,Fname,Lname,position,gender,DOB)

Manger(staff_num,bounce, Fname ,Lname ,position , gender ,DOB)

Branch(Branchcode, staff_num, street,city,postcode)

Telephone(Telephone , Branchcode)

RoomEorRent(room_id ,room_num,room_floor,room_type,Room_ description_room,)

MangedRoom (staff_num, Room_id)

Book(chect_in,chec_out,duration,guest_num,room_id)

Hotelguest (guest_num ,Employee_id,fname,Lname, Telephone ,email)

Service(Service_num)

Fitness(Service_num ,trainer_name, class_time)

Resturant(Service_num, menu, restaurant_time , reservation)

NORMALIZATION :

Manger

FD's:

Staff_num	Fname	Lname	position	Bounce	gender	DOB

Staff_num → Fname, Lname, position, gender,DOB

position → Bounce

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the primary key only it is in the 2nd Normal form.

third Normal Form:

Staff(Staff_num ,Fname, Lname, position, gender,DOB)

PositionManager(position, Bounce)

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Staff

FD's:

Staff_num	Fname	Lname	position	SuperviserID	gender	DOB

Staff_num → Fname, Lname, position, gender,DOB,SupervisorID

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the primary key only it is in the 2nd Normal form.

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

MangedRoom

FD's:

Staff_num	Room_id

Staff_num → Fname, Lname, position, gender,DOB,SupervisorID

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the primary key only it is in the 2nd Normal form.

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

RoomForRent

roomId	Room_Num	Room_floor	roomtype	BookDescription	

roomId → Room_Num, Room_floor, roomtype, BookDescription, Staff_num

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the primary key only it is in the 2nd Normal form.

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

HotelGustes

Guest_num	Fname	Lname	telephone	email	Employee_id

Guest _num → Fname, Lname, telephone, email, Employee_id

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the primary key only it is in the 2nd Normal form.

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Branch

FD's:

BranchCode	Staff_num	Street	City	postCode

BranchCode → staff_num , BranchCode,Street,City,postCode

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the .primary key only it is in the 2nd Normal form

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Normalization:

Telephone

FD's:

Telephone	BranchCode

Telephone , BranchCode ⟶ **Telephone, BranchCode.**

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the .primary key only it is in the 2nd Normal form

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Normalization:

Book

FD's:

Guest_num	Room_ID	Check_in	Check_out	Duration

Room_ID,guest_num ⟶ check_in,check_out

Duration ⟶ check_in,check_out

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the .primary key only it is in the 2nd Normal form

third Normal Form:

Book(Room_ID ,guest_num, check_in,check_out)

DurationTime (check_in,check_out ,Duration)

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Normalization:

Service

FD’s:

Service_num

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the .primary key only it is in the 2nd Normal form

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Fitness

Service_num	Trainer_name	Class_time

FD's:

Service_num $\longrightarrow$ **Trainer_name, Class_time**

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the primary key only it is in the 2nd Normal form .

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Restaurant

Service_num	menu	Restraint_time	reservation

FD's:

Service_num $\longrightarrow$ **menu, Restraint_time**, **reservation**

First Normal Form:

since the relation satisfies that:

The intersection of each row and column contains one and only one value

it is in the first Normal form

second Normal Form:

since the relation satisfies that:

Every non-primary key attribute is fully functionally dependent on the Primary key. and since we have only one FD and its source is the primary key only it is in the 2nd Normal form .

third Normal Form:

since the relation satisfies that: No non-primary key attribute is transitively dependent on the primary key. and since we have only one FD and its source is the primary key only it is in the 3rd Normal form

Data Dictionary showing description of all entities:

Entity Name	Description	Aliases	Occurrence
Employee	Is the person who work for the branch, and manage the rooms, he also help the guest to book a room.		Many Employee mangers many rooms For Rent, and many Employee serve many Hotel guest.
HotelGuest	Is a person who is interested to book a room from the hotel.		The guest may have more than one room, the guest may order many services or maybe not, and the guest is served by one employee.
RoomForRent	It a room that is ready to be occupied by a guest		The room for rent is managed my one employee
Service	The hotel services are offered for the hotel guest, it must be ordered by the guest.		The services are ordered my no guest or by one guest, or many guest.
Branch	It is part of a long chain of hotel		Each branch has one manager

Data Dictionary showing description of all relationships:

Entity Name	Multiplicity	Relationship	Entity Name	Multiplicity
Employee	1..* 1..1	Mang Serve	RoomForRent Hotel Guest	1..* 0..*
Staff	1..1	supervises	staff	1..10
Manger	1..1	marge	Branch	1..1
Hotel Guest	0..* 0..*	Book Order	RoomForRent Service	1..* 0..*

Data Dictionary showing description of all attributes:

Entity Name	Attribute	Description	Data Type	Length	Nulls	Multi-Valued
Employee	Staff_num Name Fname Lname position gender DOB	Uniquely identifies a member of Employee. Name of Employee First name of Employee member Last name of Employee member Job title of a member of Employee Gender of Employee Date of birth of Employee	Integer Variable character Variable character Variable character Variable character Character Date	10 15 15 15 10 10 -	NO NO NO NO NO NO NO	NO NO NO NO NO NO NO
Room For Rent	RoomID Room_num Room_Floor Room_type DataOnTheDesiredRoom	Uniquely identifies a Room For Rent Number of Room Floor of Room For Rent Type of Room For Rent Data on the desired room For Rent	Variable character Integer Integer Variable character Variable character	10 15 10 30 30	NO NO NO NO NO	NO NO NO NO NO
Hotel Guest	Guest_num{PK} Name Fname Lname Telephone Email	Uniquely identifies a Hotel Guest Name of Hotel Guest First name of Hotel Guest last name of Hotel Guest Telephone number of Hotel Guest Email for Hotel Guest	Character Variable character Variable character Variable character Integer Variable character	10 15 15 15 10 30	NO No No No No No	NO No No No yas No

Service	Service_num Service_Type	Uniquely identifies a Service The type of the Service	Character Varchar	10 20	No No	No No
Branch	Branch code stuff_info BranchAddress Street City PostCode Telephone[0..3]	Uniquely identifies a Branch Information of staff who work for the branch The address for a branch The Street of Branch City where the branch it is in The PostCodeofBranch Telephone number of Branch	Character Variable character Variable character Variable character Variable character Variable character integer	10 20 30 20 20 20 10	No No No No No No No	No No No No No No Yes

Creating Tables:

```
create table Staff (

Staff_num numeric NOT null,

Fname varchar (15)not null,

Lname  varchar (15)not null,

position  varchar (15)not null,

gender  char (15)not null,

DOB DATE NOT NULL ,

PRIMARY KEY (Staff_Num));


insert into Staff values(324556,'maha','ibrahim','Staff','f',date'1419-4-10');


insert into Staff values(459556,'khaled','fahad','manger','m',date'1420-6-3');
```

```
CREATE table Manger (
Staff_num numeric NOT null,
bounce DECIMAL (10) NOT NULL ,
fName VARCHAR (15) NOT NULL ,
lName VARCHAR (15) NOT NULL ,
position VARCHAR (10) NOT NULL ,
gender CHAR (10) NOT NULL ,
DOB DATE NOT NULL ,
PRIMARY KEY (Staff_Num));
insert into Manger values(324556,5,'maha','ibrahim','Staff','f',date'1419-4-10');
insert into Manger values(459556,5,'khaled','fahad','manger','m',date'1420-6-3');

CREATE table Branch (
Branch_code varchar(10) not null,
Staff_num numeric NOT null,
Street VARCHAR (20) NOT NULL ,
City VARCHAR (20) NOT NULL ,
postCode VARCHAR (20) NOT NULL ,
PRIMARY KEY (Branch_code)
FOREIGN KEY (BranchCode)REFERENCES Branch( BranchCode));

insert into Branch values('00R3',324556,'altahliah','riyadh','6298');
insert into Branch values('00R9',459556 ,'King salman','riyadh','0988');
```

```
CREATE table Telephone (
Branch_code char not null,
Telephone  numeric NOT null,
PRIMARY KEY (Telephone)
FOREIGN KEY (BranchCode)REFERENCES Branch( BranchCode));
insert into Telephone values('00R3',0558125581);
insert into Telephone values('00R9',0554433903);

CREATE table RoomFor Rent (
Room_id varchar (10) not null,
Room_num INTEGER (15) NOT NULL ,
Room_Floor INTEGER (10) NOT NULL ,
Room_Type VARCHAR (30) NOT NULL ,
Room_describtion  VARCHAR (30)
PRIMARY KEY (RoomID));
insert into RoomFor Rent values('35D7',1,1,'Single','hot path');
insert into RoomFor Rent values('67D7',2,2,'double','breaqkfast');

CREATE table MangedRoom (
Room_id varchar (10) not null,
Staff_num numeric NOT null,
PRIMARY KEY (RoomID,Staff_num));


insert into MangedRoom  values('35D7',324556);
insert into MangedRoom  values('67D7',459556);
```

```
CREATE table Book (
Check_in DATE NOT NULL ,
Check_Out DATE NOT NULL ,
Duratione VARCHAR (10) NOT NULL ,
guest_num numeric (10) not null,
Room_id varchar (10) not null,
PRIMARY KEY (Room_id,guest_num));

insert into Book  values(DATE'1439-5-3',DATE'1439-5-10','7', 2345,'35D7');
insert into Book  values(DATE'1439-5-1',DATE'1439-5-3','2', 5672,'67D7');


CREATE table Hotelguest(
guest_num numeric (10) not null,
Staff_num numeric NOT null,
Fname varchar (15)not null,
Lname  varchar (15)not null,
Telephone  numeric NOT null,
email varchar(15) NOT NULL,
PRIMARY KEY (guest_num,Staff_num));

insert into Hotelguest  values(2345,324556,'maha','ibrahim',0558125581,'maha11@gmail');
insert into Hotelguest  values(5672,459556,'khaled','fahad',0554433903,'khaled0@gmail');


CREATE table service(
```

```
service_num numeric (10) not NULL,
PRIMARY KEY (service_num));

insert into service  values(01);
insert into service  values(02);

CREATE table Fitness(
service_num numeric (10) not NULL,
trainer_name varchar (15) not null,
class_time varchar(10) not null,
PRIMARY KEY (service_num));

insert into Fitness  values(01,'faisal','10:00AM');
insert into Fitness  values(02,'saleh','7:00PM');

CREATE table Resturant(
service_num numeric (10) not NULL,
menue varchar (30) not null,
Resturant_time varchar(20) not null,
reservation varchar (10) not null,
PRIMARY KEY (service_num));

insert into Resturant  values(01,'italian','8:00AM-10:00PM','true');
insert into Resturant  values(02,'chinese','8:00AM-10:00PM','true');
```