DBMS Project

Prof. Name : P M Jat
Topic : Hotel Management
Group No. : G4_3

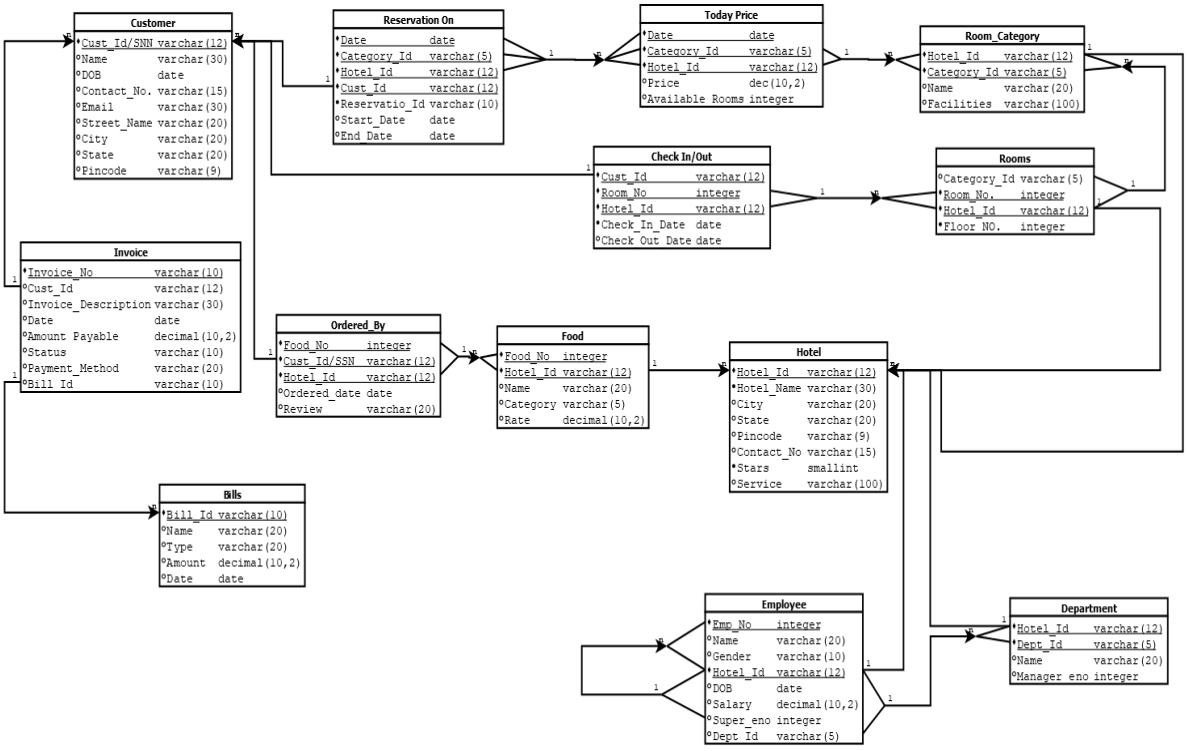
Submissions :

1) Relational Schema Diagram
2) Minimal FD Set
3) BCNF Decomposing
4) DDL Script

TEAM MEMBER DETAILS :

Name	Student Id
Bhavesh Baraiya	202101241
Smeet Agrawal	202101237
Dhruvin Chaudhary	202101239
Rushang Parmar	202101240

1) Relational Schema Diagram :



2) Minimal FD Set :

Cust_Id/SNN → {Name ,DOB ,Contact_No ,Email ,City ,State ,Pin code}

{Date ,Category_Id ,Hotel_Id ,Cust_Id} → {Reservaion_Id ,Start_Date , End_Date}

{Reservation_Id ,Hotel_Id} → {Start_Date ,End_Date, Date ,Category_Id ,Cust_Id }

Invoice_No → {Customer_Id ,Invoice_Description ,Date ,Amount Payable ,Status , Payment_Method ,Bill_id}

Bill_Id → {Name ,Type ,Amount ,Date }

{Food_No ,Cust_Id ,Hotel_Id} → {Ordered_date ,Cust_Review}

{Food_No ,Hotel_id} → {Food_name ,Category ,Rate}

{Room_No , Hotel_Id} → {Category_id ,Floor No}

{Cust_Id ,Room_no ,Hotel_Id} → {Check_In_date ,Check_Out_date}

{Date ,Category_Id ,Hotel_id} → {Price ,Available Rooms}

{Hotel_Id ,Category_id} → {Category_Name ,Facilities}

Hotel_Id → {Hotel_Name , City , State , Pin code , Contact_No , Stars , Service}

{Hotel_id , Emp_No} → { Name , Gender , DOB , Salary , Super_eno , Dept_Id}

{Hotel_id , Dept_No } → { Name , Manager eno}

3) Checking For BCNF(for every relation) :

Customer :

Cust_Id/SNN → {Name ,DOB ,Contact_No ,Email ,City ,State ,Pin code}

Key = { Cust_Id/SNN }

- Here every FDs have Key in left side so this relation is in BCNF.

Reservation On :

{Date ,Category_Id ,Hotel_Id ,Cust_Id} → {Reservaion_Id ,Start_Date , End_Date}

{Reservation_Id ,Hotel_Id} → {Start_Date ,End_Date , Date ,Category_Id ,Cust_Id }

Key ={Date ,Category_Id ,Hotel_Id ,Cust_Id} , {Reservation_Id ,Hotel_Id}.

- Here every FDs has one of the Key in left side so this relation is in BCNF.

Invoice :

Invoice_No → {Customer_Id ,Invoice_Description ,Date ,Amount Payable ,Status , Payment_Method ,Bill_id}

Key = Invoice_No

- Here every FDs have Key in left side so this relation is in BCNF.

Bill :

Bill_Id → {Name ,Type ,Amount ,Date }

Key = Bill_Id

- Here every FDs have Key in left side so this relation is in BCNF.

Ordered_By :

{Food_No ,Cust_Id ,Hotel_Id} → {Ordered_date ,Cust_Review}

Key ={Food_No ,Cust_Id ,Hotel_Id}

- Here every FDs have Key in left side so this relation is in BCNF.

Food :

{Food_No ,Hotel_id} → {Food_name ,Category ,Rate}

Key = {Food_No ,Hotel_id}

- Here every FDs have Key in left side so this relation is in BCNF.

Room :

{Room_No , Hotel_Id} → {Category_id ,Floor No}

Key = {Room_No , Hotel_Id}

- Here every FDs have Key in left side so this relation is in BCNF.

Check In/Out :

{Cust_Id ,Room_no ,Hotel_Id} → {Check_In_date ,Check_Out_date}

Key ={Cust_Id ,Room_no ,Hotel_Id}

- Here every FDs have Key in left side so this relation is in BCNF.

Today Price :

{Date ,Category_Id ,Hotel_id} → {Price ,Available Rooms}

Key = {Date ,Category_Id ,Hotel_id}

- Here every FDs have Key in left side so this relation is in BCNF.

Room Category :

{Hotel_Id ,Category_id} → {Category_Name ,Facilities}

Key = {Hotel_Id ,Category_id}

- Here every FDs have Key in left side so this relation is in BCNF.

Hotel :

Hotel_Id → {Hotel_Name , City , State , Pin code , Contact_No , Stars , Service}

Key = Hotel_Id

- Here every FDs have Key in left side so this relation is in BCNF.

Employee :

{Hotel_id , Emp_No} → { Name , Gender , DOB , Salary , Super_eno , Dept_Id}

Key = {Hotel_id , Emp_No}

- Here every FDs have Key in left side so this relation is in BCNF.

Department :

{Hotel_id , Dept_No } → { Name , Manager eno}

Key = {Hotel_id , Dept_No }

- Here every FDs have Key in left side so this relation is in BCNF.

Q-4 DDL Script :

```
create schema Hotel_Management;
set search_path to Hotel_Management;

-- ***************  Hotel   ******************

Create table Hotel(
Hotel_ID varchar(12) primary key,
Hotel_Name varchar(30),
City varchar(20),
State varchar(12),
Pincode varchar(9),
Contact_No varchar(15),
Stars smallint not null,
Service varchar(100)
);

-- ***************  Room Category  **************

Create table Room_Category(
Hotel_ID varchar(12),
Category_ID varchar(5),
Name varchar(20),
Facilities varchar(100),
Primary key(Hotel_ID,Category_ID),
Foreign key (Hotel_ID) references Hotel(Hotel_ID) on delete cascade on update
cascade
);
```

```
--**********  Room  **********

Create table Rooms(
Category_ID varchar(5),
Room_No integer,
Hotel_ID varchar(12),
Floor_No integer not null,
Primary key(Room_No,Hotel_ID),
Foreign key (Hotel_ID) references Hotel(Hotel_ID) on delete cascade on update cascade,
Foreign key (Category_ID,Hotel_ID) references Room_Category(Category_ID,Hotel_ID) on delete cascade on update cascade
);

-- **********  Today Price  **********

create table Today_Price(
Date date,
Category_ID varchar(5),
Hotel_ID varchar(12),
Price decimal(10,2),
Available_Rooms integer,
Primary key(Date,Category_Id,Hotel_ID),
Foreign key(Category_ID,Hotel_ID) references Room_Category(Hotel_ID,Category_ID)  on delete cascade on update cascade
);

-- **********  Customer  **********

create table Customer(
```

```
Cust_id varchar(12) primary key,
"Name" varchar(30),
DOB date,
Contact_No varchar(15),
Email varchar(30),
Street_Name varchar(20),
City varchar(20),
"State" varchar(20),
Pincode varchar(9)
);

-- **********  reservation On  **********

Create table Reservation_On(
"Date" date,
Category_ID varchar(5),
Hotel_ID varchar(12),
Cust_ID varchar(12),
Reservation_ID varchar(10) not null,
Start_Date date,
End_date date,
Primary key("Date",Category_ID,Hotel_ID,Cust_ID),
Foreign key("Date",Category_ID,Hotel_ID) references
Today_Price(Date,Category_ID,Hotel_ID)  on delete cascade on update cascade,
Foreign key (Cust_id) references Customer(Cust_id) on delete cascade on update
cascade
);

-- **********  Check In/Out  **********

Create table check_in_out(
```

```
Cust_ID varchar(12),
Room_No integer,
Hotel_ID varchar(12),
Check_In_Date date,
Check_Out_Date date,
Primary key(Cust_id,Room_No,Hotel_ID),
Foreign key(Room_No,Hotel_ID) references Rooms(Room_No,Hotel_ID)  on delete cascade on update cascade,
Foreign key (Cust_id) references Customer(Cust_id) on delete cascade on update cascade
);

-- **********  Department  **********

Create table Department(
Hotel_ID varchar(12),
Dept_ID varchar(5),
Name varchar(20),
Manager_eno integer,
Primary key(Hotel_ID,Dept_ID),
Foreign key (Hotel_ID) references Hotel(Hotel_ID) on delete cascade on update cascade
);

-- **********  Employee  **********

Create table Employee(
Hotel_ID varchar(12),
Emp_No integer,
"Name" varchar(20),
Gender varchar(10),
```

```
DOB date,
Salary decimal(10,2),
Super_eno integer,
Dept_ID varchar(5),
Primary key(Hotel_ID,Emp_NO),
Foreign key(Dept_ID,Hotel_ID) references Department(Dept_ID,Hotel_ID) on delete cascade on update cascade,
Foreign key(Super_eno,Hotel_ID) references Employee(Emp_No,Hotel_ID) on delete cascade on update cascade,
Foreign key(Hotel_ID) references Hotel(Hotel_ID) on delete cascade on update cascade
);

-- **********  Food  **********

Create table Food(
Food_No integer,
"Name" Varchar(20),
Hotel_ID varchar(12),
Category Varchar(5),
Rate decimal(10,2),
Primary key (Food_No,Hotel_ID),
Foreign key(Hotel_ID) references Hotel(Hotel_ID) on delete cascade on update cascade
);

-- **********  Ordered By  **********

Create table Ordered_By(
Food_NO integer,
Cust_ID varchar(12),
Hotel_ID varchar(12),
```

```
Ordered_date date,
Review varchar(20),
primary key(Food_NO,Cust_ID,Hotel_ID),
Foreign key (Cust_ID) references Customer(Cust_ID) on delete cascade on update cascade,
Foreign key(Food_No,Hotel_ID) references Food(Food_No,Hotel_ID)
);

-- **********  Bills  **********

Create table Bills(
Bill_ID varchar(10) primary key,
"Name" varchar(20),
Type varchar(20),
Amount decimal(10,2),
Date date
);
-- **********  Invoice  **********

Create table Invoice(
Invoice_No Varchar(10) Primary key,
Cust_id Varchar(12),
Invoice_Description Varchar(30),
"Date" date,
Amount_Payable decimal(10,2),
Status Varchar(10),
Payment_Method Varchar(20),
Bill_ID Varchar(10),
Foreign key (Cust_id) references Customer(Cust_id) on delete cascade on update cascade,
Foreign key (Bill_ID) references Bills(Bill_ID) on delete cascade on update cascade);
```