VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

A project report on

HOTEL MANAGEMENT SYSTEMS

Submitted towards jth component of the course

Database Management Systems

by

NASEER THANVEER (17BCA0004)

ARYAN BHATNAGAR(17BCA0005)

AVINASH KUMAR(17BCA0007)

handled by

Prof Parimala M

School of Information Technology and Engineering

VIT University

March 2018

PROBLEM DESCRIPTION

- Our database contains information of all customers staying in our hotel.
- We store each customer's details like customer's name (first name, last name), customer ID, address, phone no. and their email address if available.
- The customer makes a reservation for a room by giving his check-in date and check-out date which are stored in the database.
- The customer pays the money(payment) for the type of room he booked by selecting the mode of payment (cash, card). This payment is included in the reservation.
- Other than customer's details and his reservation details our database also stores information of the rooms in our hotel.
- This information includes details like room type, room fare and total number of rooms present in that type.
- A room is then allotted for a reservation.

INDEX

Chapter 1: Analysis

1.1 ENTITIES

1) Customer
2) Payment
3) Reservation
4) Room

1.2 ATTRIBUTES

1) Customer
   a) Customer ID
   b) Name
   c) Address
   d) Phone Number
   e) Email ID

2) Payment
   a) Payment ID
   b) Amount

3) Reservation
   a) Reservation ID
   b) Check-in
   c) Check-out

4) Room
   a) Room No
   b) Room type
   c) Room fare
   d) Total No. of Room

1.3 RELATIONSHIP

- A customer **makes** a reservation for a room
- A customer **pays** payment.
- A reservation **includes** a payment
- A room is **allotted** for a reservation.

1.4 CARDINALITY

1. Customer makes reservation
   1.1. a customer can make many reservations.
   1.2. One reservation is done by one customer.

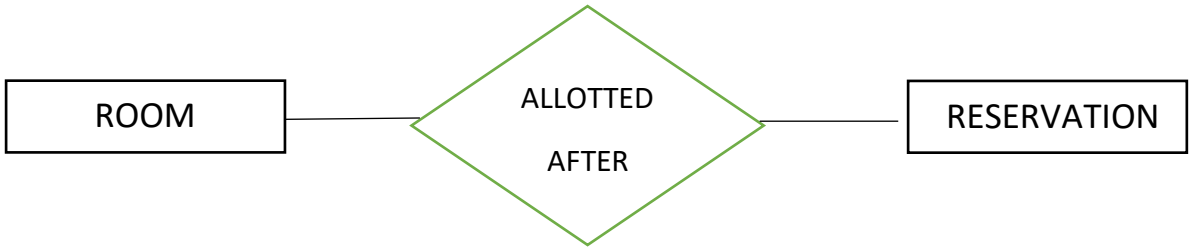
   Hence, the cardinality is 1:N

2. Customer pays payment
   2.1. One customer can make many payments.
   2.2. One payment is done by one customer.

   Hence, the cardinality is 1:N

3. Reservation includes payment
   3.1. One reservation includes one payment.
   Hence, the cardinality is 1:1

4. Rooms allotted after reservation.
   4.1. One room is allotted for only one reservation.
   Hence, the cardinality is 1:1

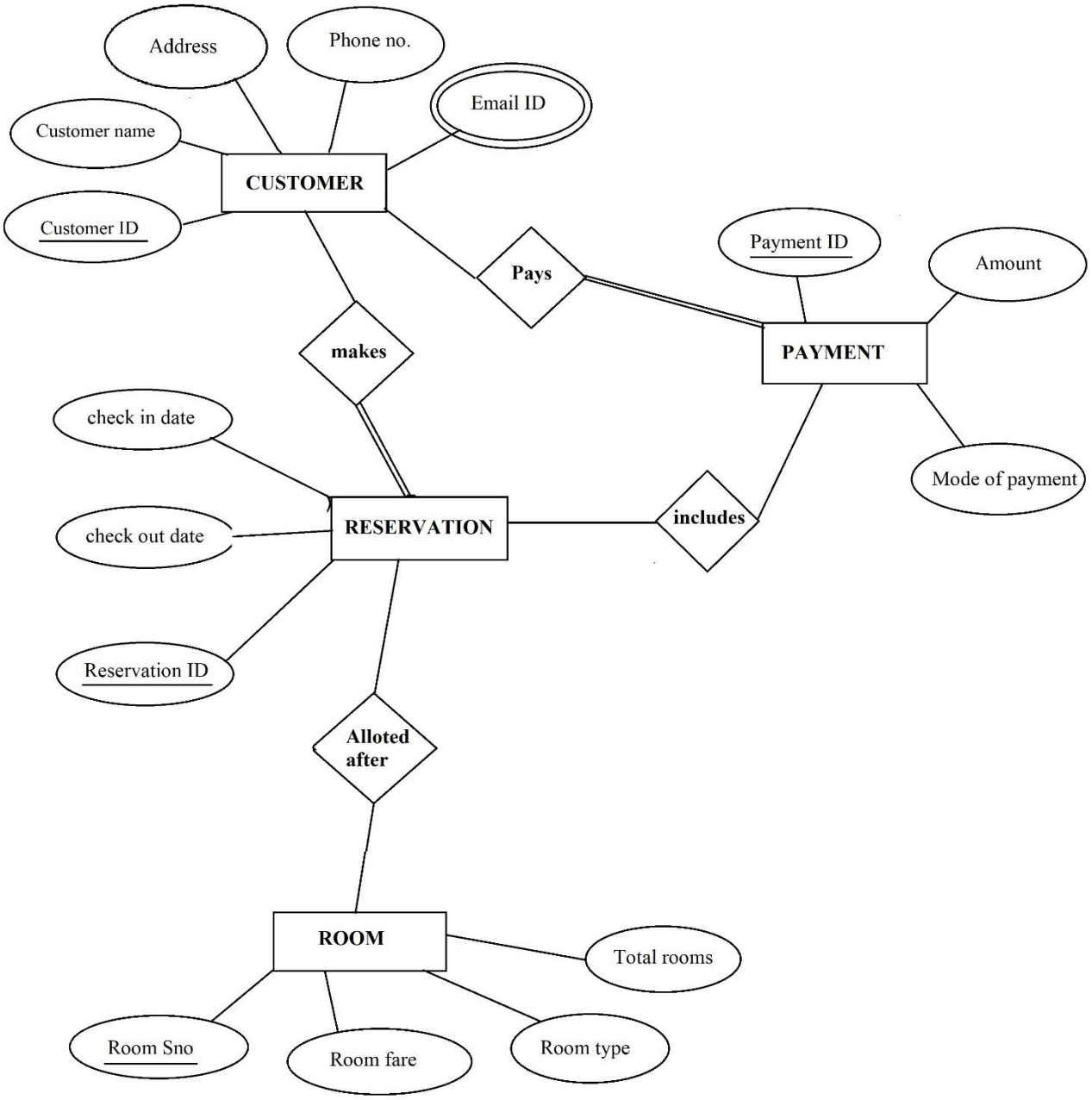


1.5 PARTICIPATION

- Customer and Reservation cannot exist without each other hence, total participation.
- Customer and Payment cannot exist without each other hence, total participation.
- Reservation and Room can exist without each other hence, partial participation.
- Reservation and Payment can exist without each other hence, partial participation.

Chapter 2: DESIGN

2.1 ENTITY RELATIONSHIP DIAGRAM



2.2 MAPPING OF ER TO TABLE

Step 1 – Regular Entity Types

- For each regular entity type E
  - create relation R that includes all simple attributes of E and simple component attributes of composite attributes of E
  - choose one of the key attributes of E as the primary key

Step 2 – Weak Entity Types

- For each weak entity type W with owner entity type E
  - create relation R including all simple attributes and simple component attributes of W
  - include primary key of E's corresponding relation as a foreign key in W
  - primary key of R is combination of foreign key from E and W's partial key

Step 3 –Binary 1:1 Relationship Types

- For each binary 1:1 relationship type R
  - identify relations corresponding to participating entities, say T and S
  - if there is total participation, make that S
  - include primary key of T as a foreign key in S
  - include all simple attributes and simple components of composites of R as attributes of S

Step 4 -Regular 1:N Binary Relationship Types

- For each regular 1:N binary relationship type R
  - identify relations, say S and T, corresponding to participating entity types
  - if S is at "N side" of relationship type then include primary key of T in S
  - include any simple attributes or simple components of R as attributes of S

Step 5 – Binary M:N Relationship Types

- For each binary N:M relationship type K
  - create new relation R to represent K
  - if S and T are the relations corresponding to the participating entity types then include their primary keys as foreign keys in R
  - combination of foreign keys will be primary key of R
  - include any simple attributes or simple components of K as attributes of R

Step 6 – Multi-valued attributes

- For each multi-valued attribute A
  - create a relation R that contains the attribute A and the primary key of the relation S that corresponds to the entity type that contained A

After following all 6 steps we come up with the following tables for our database:-

Customer	(Customer ID, Cname, Address, Phone, email ID)
Reservation	(Reservation ID, Check-in date, Check-out date, Customer ID)
Payment	(Payment ID, Amount, Mode of Payment, Reservation ID, Customer ID)
Room	(Room No, Room fare, Room type, Total No. of Rooms, Reservation ID)
Email	(Email ID, Customer ID)

Chapter 3: IMPLEMENTATION

1) Create with constraint

- create table customer(C_ID number(4) primary key,

C_name varchar2(20),

Address varchar2(50),

phone number(10));

```
SQL> create table customer(
  2   C_ID number(4) primary key,
  3  C_name varchar2(20),
  4  Address varchar2(50),
  5  phone number(10));
```

- create table Reservation(R_ID number(5) primary key ,

check_in date,

check_out date,

C_id number(5),

constraint f1 foreign key (C_id) references customer(C_ID));

```
SQL> create table Reservation(
  2  R_ID number(5) primary key,
  3  check_in date,
  4  check_out date,
  5  C_id number(5),
  6  constraint fk foreign key(C_id)
  7  References Customer(C_ID));

Table RESERVATION created.
```

- create table Payments(P_ID number(5) primary key,

Amount number(10),

Pay_mode varchar(10),

R_id number(5),

C_id number(5),

constraint f2 foreign key(C_id) references customer(C_ID),

constraint f3 foreign key(R_id) references Reservation(R_ID));

```
SQL> create table Payments(P_ID number(5) primary key,
  2  Ammount number(10,2),
  3  Pay_mode varchar(10),
  4  R_id number(5),
  5  C_id number(5),
  6  constraint f2 foreign key(C_id) references customer(C_ID),
  7  constraint f3 foreign key(R_id) references Reservation(R_ID));

Table PAYMENTS created.
```

- create table Room(Sno number(5) primary key ,

Fare number(10),

R_type varchar(10),

Total_Rooms number(5),

R_id number(5),

constraint f4 foreign key(R_id) references Reservation(R_ID));

```
SQL> create table Room(Sno number(5) primary key ,
  2  Fare number(10,2),
  3  R_type varchar(10),
  4  Total_Rooms number(5),
  5  R_id number(5),
  6  constraint f4 foreign key(R_id) references reservation(R_ID));

Table ROOM created.
```

- create table Email(

Email_id varchar2(15),

C_id number(5),

constraint p1 primary key(Email_id,C_id),

constraint s6 foreign key(C_id) references customer(C_ID));

```
SQL> create table Email(
  2   Email_id varchar2(15),
  3  C_id number(5),
  4  constraint p1 primary key(Email_id,C_id),
  5  constraint s6 foreign key(C_id) references customer(C_ID));

Table EMAIL created.
```

2) Insert

- insert into customer values(0001,'Aryan','H.no 225,R K puram,Delhi',9128373666,'aryn@gmail.com');

  insert into customer values(0002,'Avinash','H.no 226, Rk puram,Delhi',9128373656,'arn12@gmail.com');

  insert into customer values(0003,"thanveer","H.no 235, sk puram,Delhi",9158373666,"ary67n@gmail.com");

  insert into customer values(0004,'Hifaazz','H.no 25, Tk puram,Delhi',9124473666,'aww22@gmail.com')

  insert into customer values(0005,'Aryan','H.no 225, pk puram,Delhi',9128373666,'arssyn@gmail.com');

```
SQL> insert into customer values(0001,'Aryan','H.no 225, Rk puram,Delhi',9128373666,'aryn@gmail.com');
1 row inserted.
```

```
SQL> insert into customer values(0002,'Avinash','H.No 223,RKPuram,Delhi',9704907348,'avi@gmail.c0m');

1 row inserted.
```

```
SQL> insert into customer values(0003,'Thanveer','H.No 158,Chittoor,Andhra_Pradesh',9252024548,'thanveer@gmail.com'
1 row inserted.

SQL> insert into customer values(0004,'Hifaazz','H.no 25, Tk puram,Delhi',9124473666,'aww22@gmail.com');
1 row inserted.

SQL> insert into customer values(0005,'snehil','H.no 225, pk puram,Delhi',9128373666,'arssyn@gmail.com');

1 row inserted.
```

- insert into Reservation values(100,'10-02-2018','12-03-2018',0001);

  insert into Reservation values(101,'11-02-2018','12-03-2018',0002);

  insert into Reservation values(102,'09-02-2018','12-03-2018',0003);

  insert into Reservation values(103,'08-02-2018','12-03-2018',0004);

  insert into Reservation values(104,'07-02-2018','12-03-2018',0005);

```
SQL> insert into Reservation values(100,'10-02-2018','12-03-2018',0001);

1 row inserted.

SQL> insert into Reservation values(101,'11-02-2018','12-03-2018',0002);

1 row inserted.

SQL> insert into Reservation values(102,'09-02-2018','12-03-2018',0003);

1 row inserted.

SQL> insert into Reservation values(103,'08-02-2018','12-03-2018',0004);

1 row inserted.

SQL> insert into Reservation values(104,'07-02-2018','12-03-2018',0005);

1 row inserted.
```

- insert into Payments values(9292,2300,’Debit card’,100,0001);

  insert into Payments values(9232,2500,’card’,101,0002);

  insert into Payments values(9234,5000,’cash’,102,0003);

  insert into Payments values(4234,50000,’cheque’,104,0004);

  insert into Payments values(4221,50000,’cheque’,103,0005);

```
SQL> insert into Payments values(9292,2300,'Debit card',100,0001);
1 row inserted.
```

```
SQL> insert into Payments values(9232,2500,'card',101,0002);
1 row inserted.


SQL> insert into Payments values(9234,5000,'cash',102,0003);
1 row inserted.


SQL> insert into Payments values(4234,50000,'cheque',104,0004);
1 row inserted.


SQL> insert into Payments values(4221,50000,'cheque',103,0005);
1 row inserted.
```

- insert into Room values(01,10000,’delux A/c’,580,101);

  insert into Room values(02,15000,’luxary A/c’,780,103);

  insert into Room values(03,1000,’normal’,980,102);

  insert into Room values(05,5500,’A/c’,1110,104);

```
SQL>  insert into Room values(01,10000,'delux A/c',580,101);
1 row inserted.


SQL> insert into Room values(02,15000,'luxury A/c',780,102);
1 row inserted.

SQL> insert into Room values(03,1000,'normal',980,102);

1 row inserted.
```

- insert into Email values(‘ak@gmail.com’,0001);

  insert into Email values(‘dd@gmail.com’,0001):

```
SQL> insert into Email values('&Email_id',&C_id);
Enter value for Email_id: ak@gmail.com
Enter value for C_id: 0001
old:insert into Email values('&Email_id',&C_id)
new:insert into Email values('ak@gmail.com',0001)

1 row inserted.

SQL> /
Enter value for Email_id: dd@gmail.com
Enter value for C_id: 0001
old:insert into Email values('&Email_id',&C_id)
new:insert into Email values('dd@gmail.com',0001)

1 row inserted.
```

3) Alter

alter table customer add(Email varchar2(20));

```
SQL> alter table customer add(Email varchar2(20));

Table CUSTOMER altered.
```

4) Delete

delete * from customer where C_ID=0001;

```
SQL> delete * from customer where C_ID=0001;
1 row deleted.
```

5) Update

update customer set email='arrhy@gmail.com' where C_ID=0001;

```
SQL> update customer set email='arrhy@gmail.com' where C_ID=0001;

1 row updated.
```

6) Select…Where

select * from Reservation where R_ID=100;

```
SQL> select * from Reservation where R_ID=100;
      R_ID CHECK_IN CHECK_OU       C_ID
---------- -------- -------- ----------
       100 10-02-18 12-03-18          1
```

7) Order by

select * from Reservation order by check_in;

```
SQL> select * from Reservation order by check_in;
      R_ID CHECK_IN CHECK_OU       C_ID
---------- -------- -------- ----------
       100 10-02-18 12-03-18          1
       102 02-03-18 14-03-18          3
       103 05-03-18 15-03-18          4
       104 09-03-18 11-03-18          5
       101 09-03-18 12-03-18          2
```

8) Pattern matching

select * from Payments where pay_mode like ‘c%’;

```
SQL> select * from Payments where pay_mode like 'c%';
      P_ID     AMOUNT PAY_MODE         R_ID       C_ID
---------- ---------- ---------- ---------- ----------
      9232       2500 card              101          2
      9234       5000 cash              102          3
      4234      50000 cheque            104          4
      4221      50000 cheque            103          5
```

9) Aggregate functions

select max(fare) from Room;

```
SQL> select max(fare) from Room;

 MAX(FARE)
----------
     10000
```

10) Date functions

select * to char(check_in,'dd/mm/yyyy') from Reservation;

```
SQL> select to_char(check_in,'dd/mm/yyyy') from Reservation;
TO_CHAR(CH
----------
10/02/2018
09/03/2018
02/03/2018
05/03/2018
09/03/2018
```

11) Numerical functions

Select sqrt(Fare) from Room;

```
SQL> Select sqrt(Fare) from Room;
SQRT(FARE)
----------
70.7106781
83.6660027
        50
77.4596669
       100
```

12) String functions

Select length(C_name) from customer;

```
LENGTH(C_NAME)
--------------
             5
             7
             8
             6
             3
```

13) Group by…Having

Select count(*) as “count of pay mode” from Payments group by pay_mode having count(*)>1;

```
SQL> Select count(*) as "count of pay mode" from Payments group by pay_mode having count(*)>1;
count of pay mode
-----------------
                2
```

14) Join

Select c.C_name,c.phone,r.R_ID from customer c,Reservation r where c.C_ID=r.C_id;

```
SQL> Select c.C_name,c.phone,r.R_ID from customer c,Reservation r where c.C_ID=r.C_id;

C_NAME                    PHONE       R_ID
-------------------- ---------- ----------
Aryan                9128373666        100
Avinash              9128373656        101
thanveer             9158373666        102
Hifazz               9158373336        103
Sid                  9158323336        104
```

15) Subquery

Select * from Reservation where C_Id=(select C_ID from customer where C_name=’Aryan’);

```
SQL> Select * from Reservation where C_Id=(select C_ID from customer where C_name='Aryan');

     R_ID CHECK_IN CHECK_OU       C_ID
---------- -------- -------- ----------
       100 10-02-18 12-03-18          1
```

Chapter 4: CONCLUSION

This project is designed to meet the requirements of Hotel Management. Hotel management system now-a-day have the advantage of modernization. Our project has the info for every customer staying in our hotel on a daily basis. While developing this project we have learnt a lot about hotel management and got a vast and wider outlook on the subject Database Management System. We successfully achieved most of the objectives with the approach we adopted. The project has more potential and can be made bettered in many ways. It can be more user friendly, errors can be minimized if faced and performance can be improved. Overall As a student, it helped me understand SQL in a better way and made it really interesting. Therefore the project, is highly interesting and only expands horizon of my knowledge, and helps us grow in the field of applications.