**DATABASE MANAGEMENT SYSTEM**

PROJECT REPORT

ON

**HOTEL MANAGEMENT SYSTEM**

Submitted to-

Prof. Anika

By-

Chirag Mahawar 101603078

Navjot Singh Sandhu 101603199

COE-6

**ACKNOWLEDGEMENT**

We wish to express our sincere gratitude to Prof. Anika, Computer Science & Engineering Department, Thapar University, for providing us with an opportunity to undertake a Project on HOTEL MANAGEMENT SYSTEM.

We are grateful to Dr Anika for her valuable assistance and guidance throughout the project.

**CONTENTS**

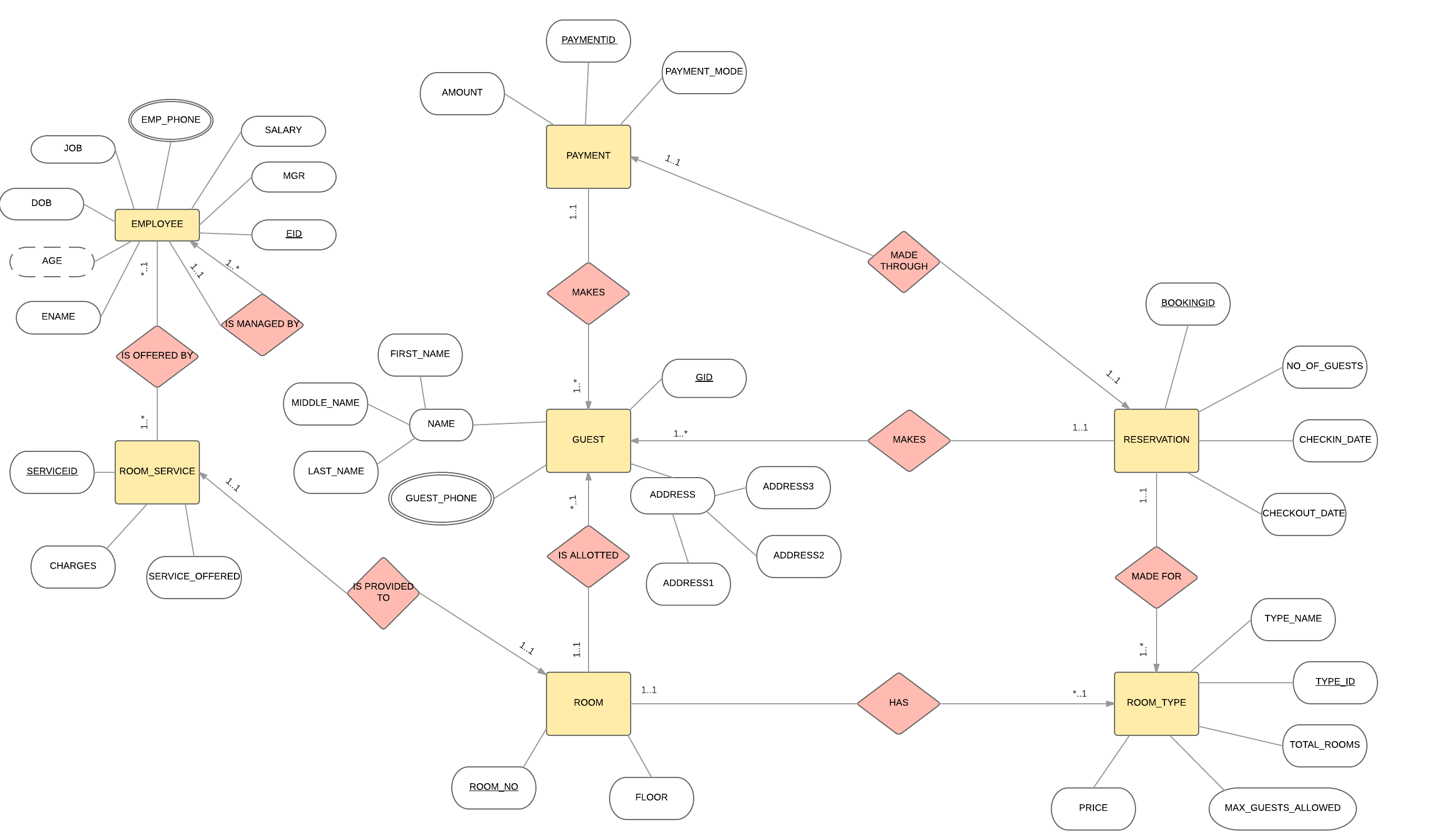
* Requirement Analysis
* ER Diagram
* ER to Table Conversion
* Functional Dependence Diagram
* Finalised Tables
* PL/SQL Codes

**REQUIREMENT ANALYSIS**

A Hotel Management System would like to keep track on the guests residing in the hotel, Reservations made by them, payment done by them, room services (like food etc.) offered to guests in their rooms. Also it would like to maintain the database of all employees working in the hotel, their managers & services offered by them.

A hotel consists of many rooms. Each room has a room type (E.g. Presidential Suite, ordinary etc.), but each room type consists of many rooms. More than one room can be allotted to a single guest but vice-versa not true. A single guest can make multiple payments but one payment is done only by one guest. A guest can also make multiple reservations for multiple rooms. After this the guest is provided with room service. The room service can be offered by more than one employee. Every employee is managed by a manager which is also an employee. Many employees can be managed by one manager but only one manager heads one employee.

**ER DIAGRAM**



**ER TO TABLE CONVERSION**

Converting ER Diagram to Tables with proper care given to composite attributes, multi-valued attributes, Derived attributes & connectivity of relationships.

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Type\_id | NOT NULL | number(10) |
| Type\_name |  | varchar2(30) |
| Total\_Rooms |  | number(10) |
| Max\_guests\_allowed |  | number(10) |
| Price |  | number(10) |

**Room\_Type**

**Payment**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Payment\_id | NOT NULL | varchar2(10) |
| Payment\_mode |  | varchar2(30) |
| Amount |  | number(10) |
| Booking\_id |  | varchar2(10) |
| Gid |  | number(10) |

**Room\_Service**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Service\_id | NOT NULL | number(10) |
| Service\_offered |  | varchar2(20) |
| Charges |  | number(10) |
| Eid |  | number(10) |
| Room\_no |  | number(10) |

**Employee**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Eid | NOT NULL | number(10) |
| Ename | NOT NULL | varchar2(30) |
| Job |  | varchar2(30) |
| DOB |  | Date |
| Salary |  | number(10) |
| Mgr |  | number(10) |

**Guest**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Gid | NOT NULL | number(10) |
| First\_name | NOT NULL | varchar2(15) |
| Middle\_name |  | varchar2(15) |
| Last\_name |  | varchar2(15) |
| Address 1 |  | varchar2(30) |
| Address 2 |  | varchar2(30) |
| Address 3 |  | varchar2(30) |

**Emp\_Phone**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Eid |  | number(10) |
| Phone |  | number(15) |

**Guest\_Phone**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Gid |  | number(10) |
| Phone |  | number(15) |

**Reservation**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Booking\_id | NOT NULL | varchar2(10) |
| No\_of\_guests | NOT NULL | number(10) |
| CheckIn\_date | NOT NULL | Timestamp |
| CheckOut\_date | NOT NULL | Timestamp |
| Type\_id |  | number(10) |
| Gid |  | number(10) |

**Room**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Room\_no | NOT NULL | number(10) |
| Floor |  | number(10) |
| Gid |  | number(10) |
| Type\_id |  | number(10) |

FUNCTIONAL DEPENDENCE DIAGRAM

**Reservation**

Type\_id

Booking\_id

CheckOut\_date

CheckIn\_date

No\_of\_Guests

Gid

**Room\_Type**

Type\_id

Price

Max\_Guests\_Allowed

Total\_rooms

Type\_name

**Room**

Room\_no

Type\_id

Gid

Floor

**Payment**

Payment\_id

Gid

Booking\_id

Amount

Payment\_mode

**Room\_Service**

Service\_id

Eid

Room\_no

Charges

Service\_offered

**Employee**

Eid

Mgr

Salary

DOB

Job

Ename

**Guest**

Gid

Address3

Address2

Address1

Last\_name

Middle\_name

First\_name

**Emp\_Phone**

Phone

Eid

**Guest\_Phone**

Phone

Gid

FINALISED TABLES

After Normalisation of tables we came up with the following tables.

**Room\_Type**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Type\_id | NOT NULL | number(10) |
| Type\_name |  | varchar2(30) |
| Total\_Rooms |  | number(10) |
| Max\_guests\_allowed |  | number(10) |
| Price |  | number(10) |

**CODE FOR ROOM\_TYPE TABLE CREATION**

create table Room\_Type(

Type\_id number(10) constraint pk\_Rt PRIMARY KEY,

Type\_name varchar2(30) constraint nn\_Rt NOT NULL,

Total\_rooms number(10),

Max\_Guests\_Allowed number(10),

Price number(10)

);

**Room**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Room\_no | NOT NULL | number(10) |
| Floor |  | number(10) |
| Gid |  | number(10) |
| Type\_id |  | number(10) |

**CODE FOR ROOM TABLE CREATION**

create table Room(

Room\_no number(10) constraint pk\_R PRIMARY KEY,

Floor number(10),

Gid varchar2(10) references Guest(Gid),

Type\_id number(10) references Room\_Type(Type\_id));

**Payment**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Payment\_id | NOT NULL | varchar2(10) |
| Payment\_mode |  | varchar2(30) |
| Amount |  | number(10) |
| Booking\_id |  | varchar2(10) |
| Gid |  | number(10) |

**CODE FOR PAYMENT TABLE CREATION**

create table Payment(

Payment\_id varchar2(10) constraint pk\_P PRIMARY KEY,

Payment\_mode varchar2(30) constraint check\_P CHECK(Payment\_mode IN('Credit Card','Debit Card','Cash','Online Banking')),

Amount number(10),

Booking\_id varchar2(10) references R1(Booking\_id),

Gid varchar2(10) references Guest(Gid));

**Room\_Service**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Service\_id | NOT NULL | number(10) |
| Service\_offered |  | varchar2(20) |
| Charges |  | number(10) |
| Eid |  | number(10) |
| Room\_no |  | number(10) |

**CODE FOR ROOM\_SERVICE TABLE CREATION**

create table Room\_Service(

Service\_id number(10) constraint pk\_Rs PRIMARY KEY,

Service\_offered varchar2(20) constraint check\_Rs CHECK(Service\_offered IN('Laundry','Food Delivery','Room Service')),

Charges number(10),

Eid number(10) references Employee(Eid),

Room\_no number(10) references Room(Room\_no)

);

**Employee**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Eid | NOT NULL | number(10) |
| Ename | NOT NULL | varchar2(30) |
| Job |  | varchar2(30) |
| DOB |  | Date |
| Salary |  | number(10) |
| Mgr |  | number(10) |

**CODE FOR EMPLOYEE TABLE CREATION**

create table Employee(

Eid number(10) constraint pk\_E PRIMARY KEY,

Ename varchar2(30) constraint nn\_E NOT NULL,

Job varchar2(30) constraint check\_E CHECK(Job IN('Manager','Laundrymen','Food Service','Janitor','Cook','Receptionist','Security')),

DOB date,

Salary number(10) CHECK(Salary>0),

Mgr number(10)

);

**Guest**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Gid | NOT NULL | number(10) |
| First\_name | NOT NULL | varchar2(15) |
| Middle\_name |  | varchar2(15) |
| Last\_name |  | varchar2(15) |
| Address 1 |  | varchar2(30) |
| Address 2 |  | varchar2(30) |
| Address 3 |  | varchar2(30) |

**CODE FOR GUEST TABLE CREATION**

create table Guest(

Gid varchar2(10) constraint pk\_R1 primary key ,

First\_name varchar2(15) constraint nn\_G NOT NULL,

Middle\_name varchar2(15),

Last\_name varchar2(15),

Address1 varchar2(30),

Address2 varchar2(30),

Address3 varchar2(30));

**Emp\_Phone**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Eid |  | number(10) |
| Phone |  | number(15) |

**CODE FOR EMP\_PHONE TABLE CREATION**

create table Emp\_phone(

Eid number(10) constraint pk\_EP primary key,

Phone number(15));

**Guest\_Phone**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Gid |  | number(10) |
| Phone |  | number(15) |

**CODE FOR GUEST\_PHONE TABLE CREATION**

create table Guest\_phone(

Eid number(10) constraint pk\_GP primary key,

Phone number(15));

As seen from the Reservation table FD Diagram, it contained transitive dependency & so in order to normalise the Reservation in 3NF it is decomposed in Reservation(R1,R2) as shown.

**R1**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Booking\_id | NOT NULL | varchar2(10) |
| CheckIn\_date | NOT NULL | Timestamp |
| CheckOut\_date | NOT NULL | Timestamp |
| Type\_id |  | number(10) |
| Gid |  | number(10) |
| Payment\_id |  | varchar2(10) |

**CODE FOR R1 TABLE CREATION**

create table R1(

Booking\_id varchar2(10) constraint pk\_R1 primary key ,

Gid varchar2(10)references Guest(Gid),

CheckIn\_date timestamp,

CheckOut\_date timestamp,

Type\_id number(10) references Room\_Type(Type\_id));

**R2**

|  |  |  |
| --- | --- | --- |
| **Name** | **Null?** | **Type** |
| Gid | NOT NULL | number(10) |
| NO\_of\_guests |  | number(10) |

**CODE FOR R2 TABLE CREATION**

create table R2(

Gid varchar2(10) constraint pk\_R2 primary key,

No\_of\_guests number(10));

PL/SQL CODES

PL/SQL CODES FOR INSERTION OF DATA INTO TABLES

* **EMPLOYEE**

DECLARE

id number(10);

name varchar2(30);

job varchar2(30);

dob date;

salary number(10);

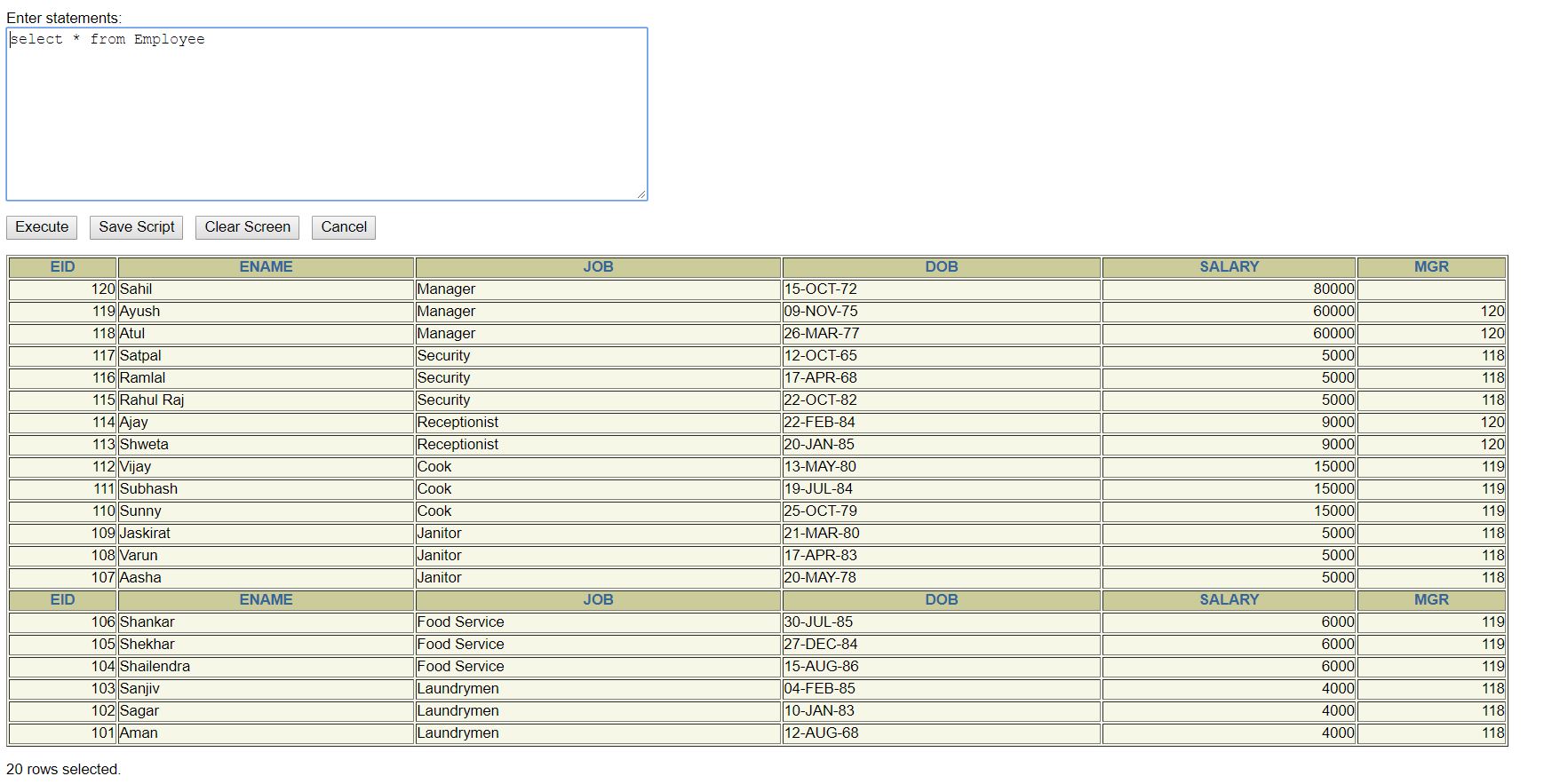
mgr number(10);

BEGIN

insert into Employee values(&id,'&name','&job','&dob',&salary,&mgr);

COMMIT;

END;



* **GUEST**

DECLARE

gid number(10);

f\_name varchar2(15);

m\_name varchar2(15);

l\_name varchar2(15);

address1 varchar2(30);

address2 varchar2(30);

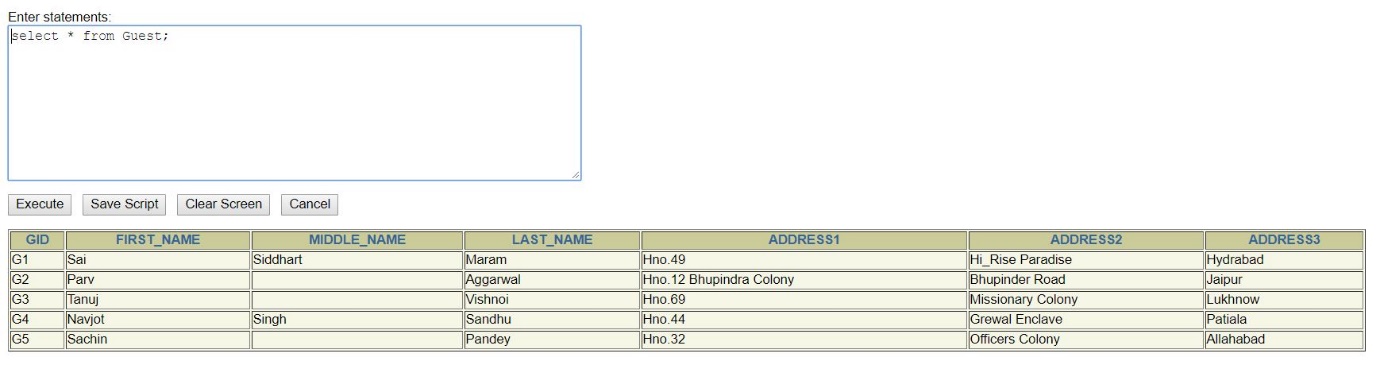
address3 varchar2(30);

BEGIN

insert into Guest values(&gid,'&f\_name','&m\_name','l\_name','&address1','&address2','&address3');

COMMIT;

END;



* **EMP\_PHONE**

DECLARE

eid number(10);

phone number(15);

BEGIN

insert into Emp\_phone values(&eid,&phone);

COMMIT;

END;

* **GUEST\_PHONE**

DECLARE

gid number(10);

phone number(15);

BEGIN

insert into Guest\_phone values(&gid,&phone);

COMMIT;

END;

* **ROOM\_SERVICE**

DECLARE

id number(10);

service varchar2(20);

charges number(10);

eid number(10);

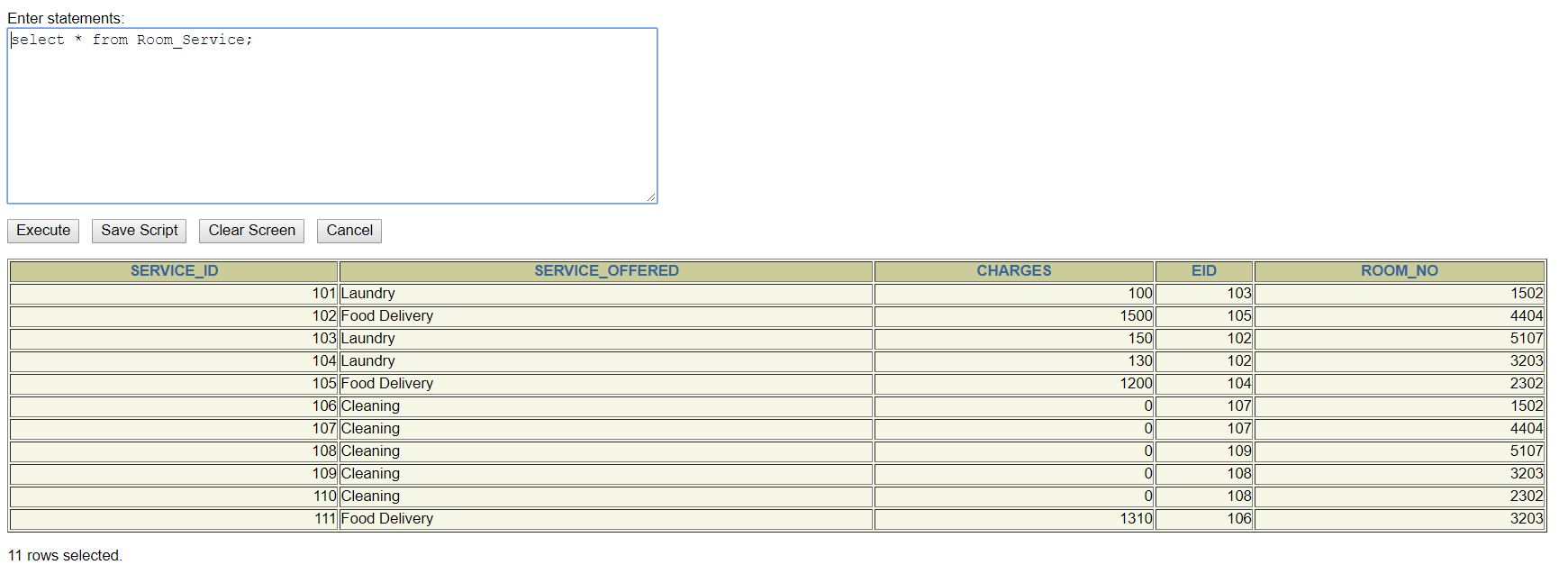
room\_no number(10);

BEGIN

insert into Room\_Service values(&id,'&service',&charges,&eid,&room\_no);

COMMIT;

END;



* **PAYMENT**

DECLARE

id varchar2(10);

mode varchar2(30);

amount number(10);

bid varchar2(10);

gid number(10);

BEGIN

insert into Payment values(&id,'&mode',&amount,&bid,&gid);

COMMIT;

END;

* **ROOM**

DECLARE

room\_no number(10);

floor number(30);

gid number(10);

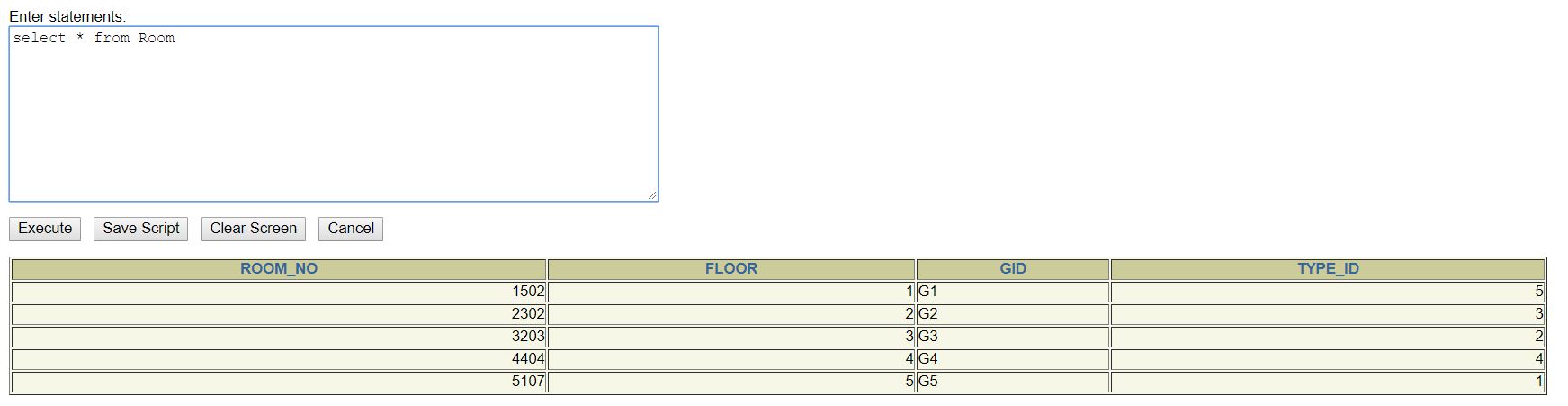
typeid number(10);

BEGIN

insert into Room values(&room\_no,&floor,&gid,&typeid);

COMMIT:

END;



* **ROOM\_TYPE**

DECLARE

typeid number(10);

typename varchar2(30);

total\_rooms number(10);

max\_guests number(10);

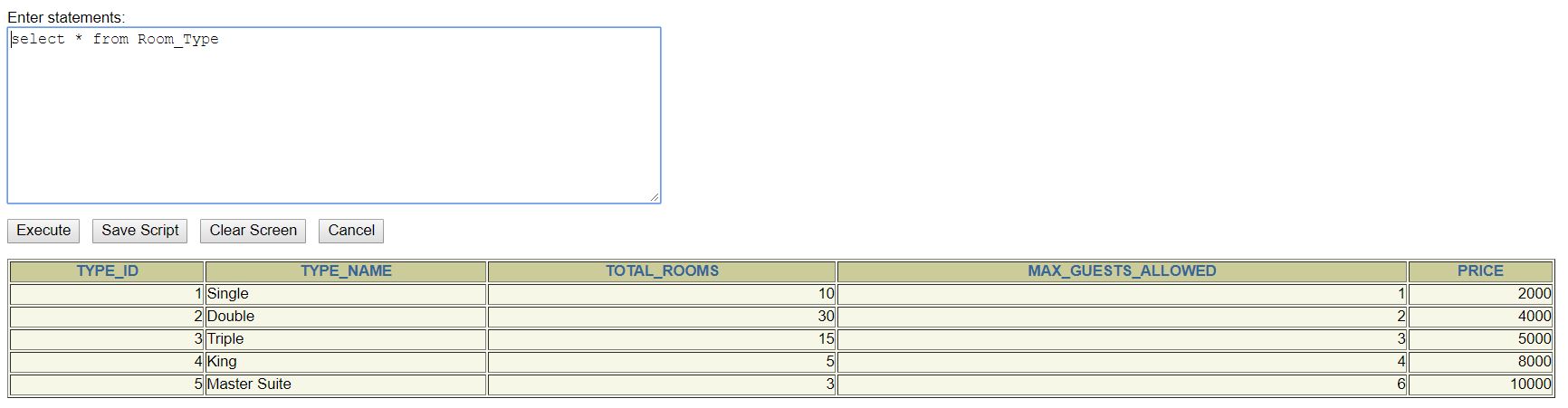
price number(10);

BEGIN

insert into Room\_Type values(&typeid,'&typename',&total\_rooms,&max\_guests,&price);

COMMIT;

END;



* **R1**

DECLARE

id varchar2(10);

checkin timestamp;

checkout timestamp;

typeid number(10);

gid number(10);

BEGIN

insert into R1 values(&id,'&checkin','&checkout',&typeid,&gid);

COMMIT;

END;

* **R2**

DECLARE

id varchar2(10);

no\_of\_guests number(10);

BEGIN

insert into R2 values(&id,&no\_of\_guests);

COMMIT;

END;

STORED FUNCTION TO DISPLAY GRADE OF EMPLOYEE USING CURSOR & CONDITIONAL CONTROL

CREATE OR REPLACE FUNCTION Find\_Grade RETURN NUMBER IS

Grade varchar2(10);

CURSOR c\_emp IS select \* from Employee;

invalid\_salary EXCEPTION;

BEGIN

FOR item in c\_emp

LOOP

IF item.Salary<0 THEN

RAISE invalid\_salary;

ELSIF item.Salary>70000 THEN

Grade:='A';

ELSIF item.Salary BETWEEN 50000 AND 70000 THEN

Grade:='B';

ELSIF item.Salary BETWEEN 20000 AND 50000 THEN

Grade:='C';

ELSIF item.Salary BETWEEN 10000 AND 20000 THEN

Grade:='D';

ELSE

Grade:='E';

END IF;

DBMS\_OUTPUT.PUT\_LINE('Grade of '||item.Eid||' , '||item.Ename||' is - '||grade);

END LOOP;

EXCEPTION

WHEN invalid\_salary THEN

DBMS\_OUTPUT.PUT\_LINE('Salary of Employee is invalid. Please see the Employees Database.');

RETURN 0;

END;

PL/SQL CODE TO CALL Find\_Grade FUNCTION

set serveroutput on;

set verify off;

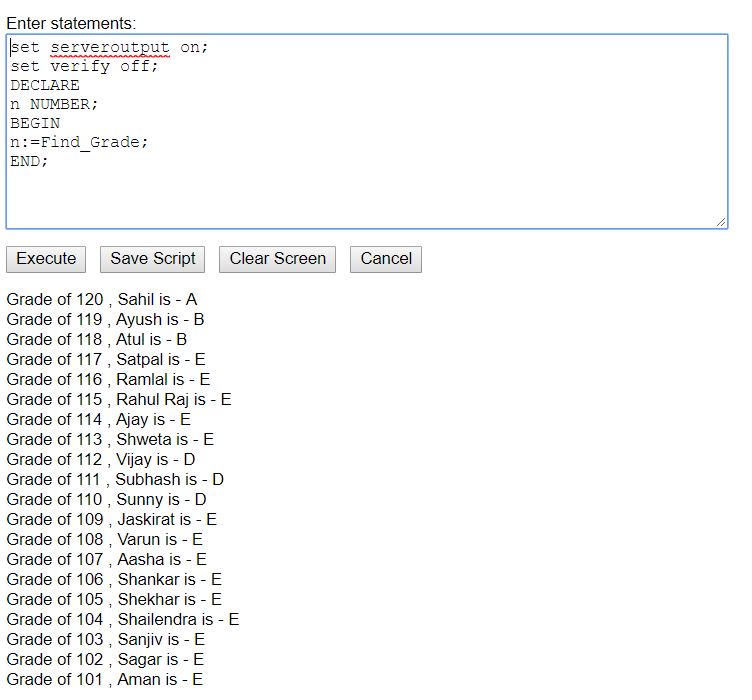
DECLARE

n NUMBER;

BEGIN

n:=Find\_Grade;

END;



STORED PROCEDURE TO CALCULATE TOTAL AMOUNT PAYABLE FOR A GUEST

CREATE OR REPLACE PROCEDURE Find\_Bill

(gid1 IN varchar2,amt OUT number) IS

price number(10);

charges1 number(10);

CURSOR c\_charges IS select sum(Charges) from Room\_Service,Room where Room.Room\_no=Room\_Service.Room\_no AND Room.Gid=gid1;

CURSOR c\_amt IS select Amount from Payment where Payment.Gid=gid1;

BEGIN

open c\_charges;

open c\_amt;

fetch c\_charges into charges1;

fetch c\_amt into price;

close c\_amt;

close c\_charges;

amt:=charges1+price;

END;

PL/SQL CODE TO CALL FIND\_BILL PROCEDURE

set serveroutput on;

set verify off;

DECLARE

gid1 varchar2(10);

amt number(10);

BEGIN

gid1:='&Enter\_Guest\_ID';

IF gid1 IS NULL THEN

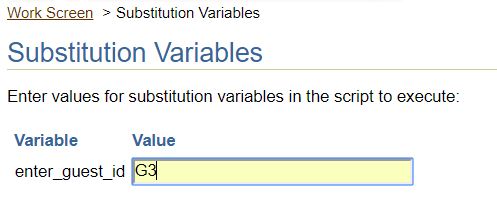
raise\_application\_error(-20001,'invaid Guest ID');

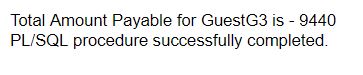
END IF;

Find\_Bill(gid1,amt);

DBMS\_OUTPUT.PUT\_LINE('Total Amount Payable for Guest'||gid1||' is - '||amt);

END;





TRIGGER TO RAISE ERROR WHILE INSERTION OF INVALID SALARY

create or replace trigger t1

before insert or update of salary on employee

for each row

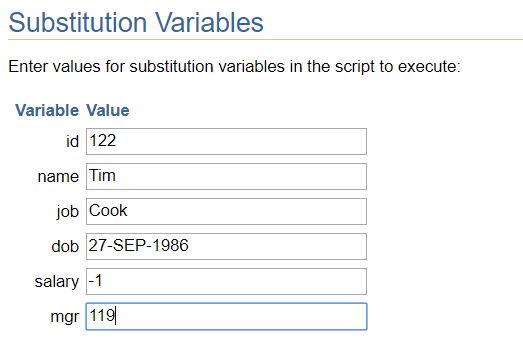
begin

if :new.salary<0 then

raise\_application\_error(-20000,'invaid salary');

end if;

end;



C:\Users\Chirag mahawar\Desktop\13.JPG

PL/SQL CODE TO DISPLAY MANAGER OF EACH EMPLOYEE USING SELF-JOIN

DECLARE

CURSOR c\_mgr IS select e1.Eid AS EMP\_ID,e1.Ename AS EMP\_NAME,e2.Eid AS MGR\_ID,e2.Ename AS MGR\_NAME from Employee e1,Employee e2 where e1.Mgr=e2.Eid;

BEGIN

for each in c\_mgr

LOOP

DBMS\_OUTPUT.PUT\_LINE('The Manager of '||each.EMP\_ID||' , '||each.EMP\_NAME||' is '||each.MGR\_ID||' , '||each.MGR\_NAME);

END LOOP;

END;

