

DBMS - MINI PROJECT

TRAVEL MANAGEMENT SYSTEM

Submitted By:

Name: Jolepalyam Himakar

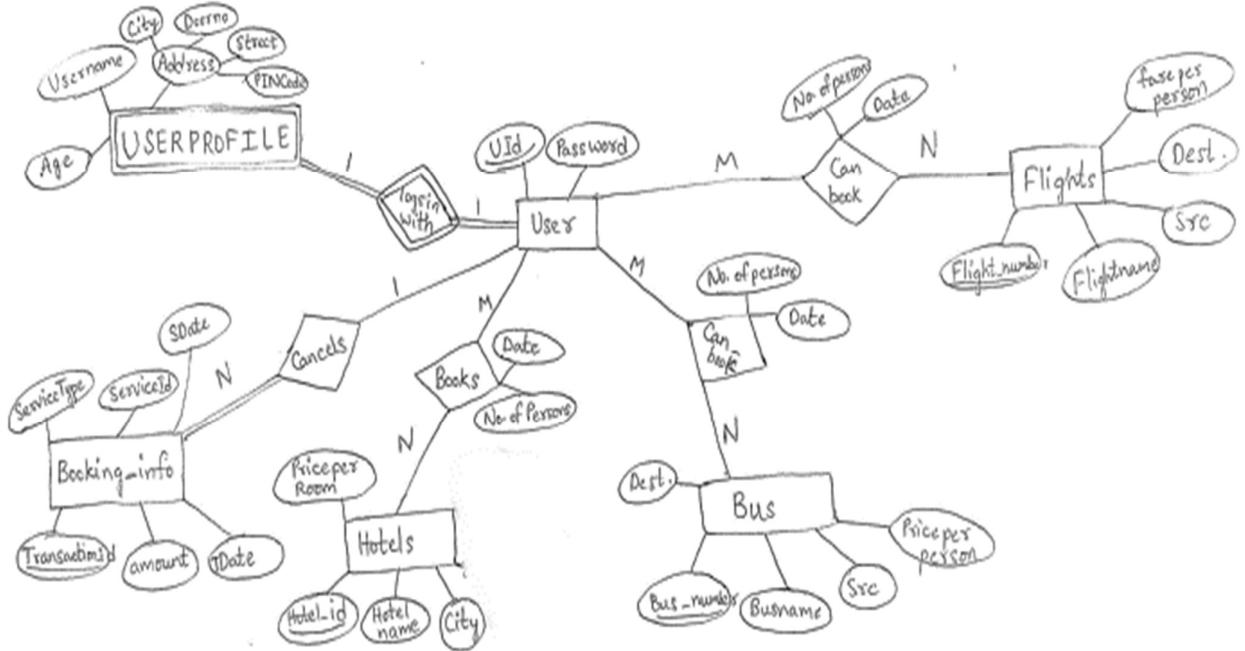
SRN: PES1UG20CS181

V Semester Section C

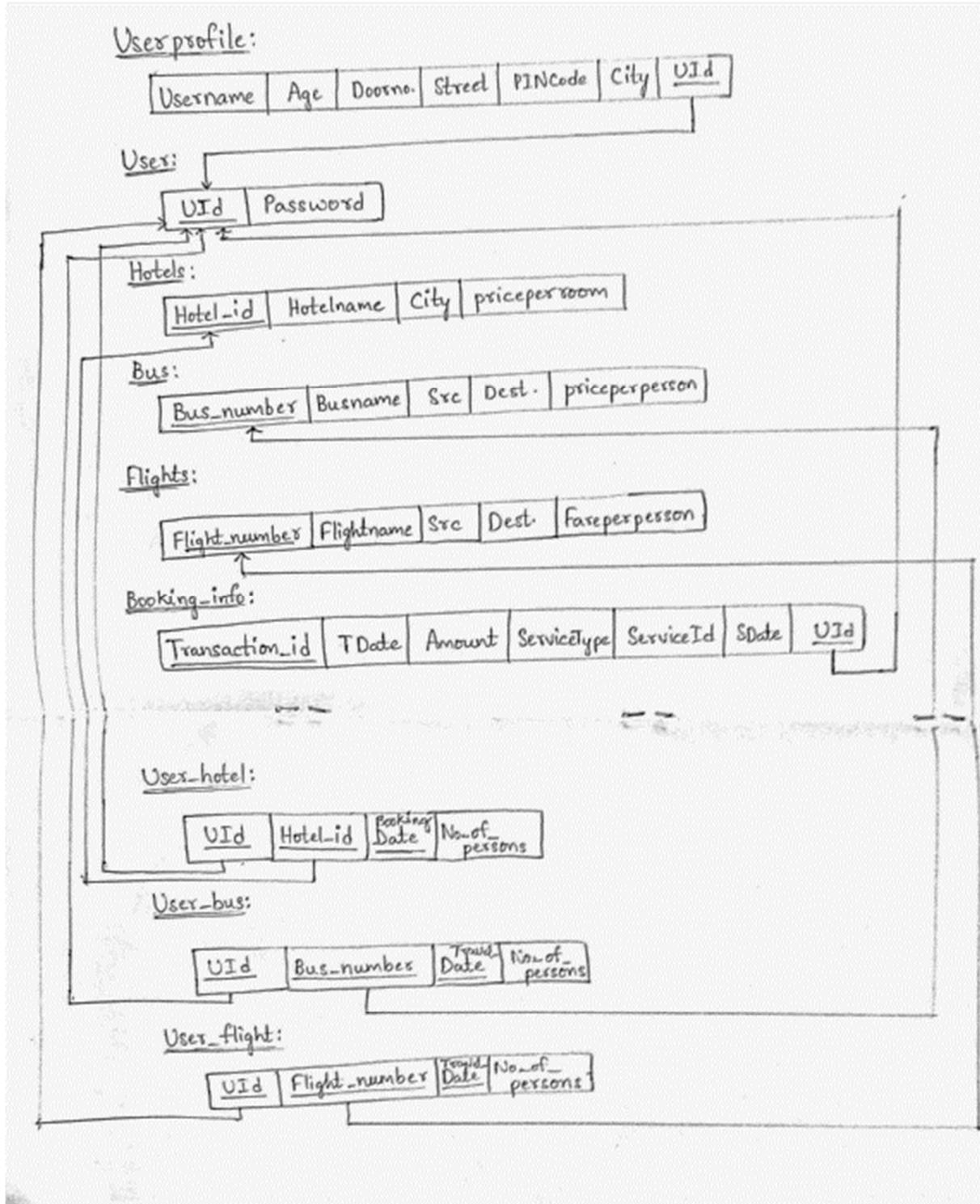
ABSTRACT

The purpose of this project is to provide the user to book bus or flight tickets to travel major cities in India as they wish to. Users can book hotels .Users can also see all his previous booking details. Admin can add new data into the table or update the existing data about the services provided by the travel agency.

ER Diagram



Relational Schema



DDL statements - Building the database

Creating the database:

```
create DATABASE if not exists traveldbms;
```

Using the created database:

```
use traveldbms;
```

Creating the table “USER” to store login credentials of user

```
create table if not exists USER(uid int(6) primary key not null, password  
varchar(15) not null);
```

Creating the table “USERPROFILE” to store user personal details

```
create table if not exists Userprofile(username varchar(30) not null,Age int ,  
doorNo varchar(7) not null , street varchar(20) not null,pincode int(6),city  
varchar(15) not null,uid int(6) not null,primary key(username,uid),foreign  
key(uid) references USER(uid) on delete cascade);
```

Creating the table “HOTELS” to store hotel details

```
create table if not exists HOTELS(Hotel_id varchar(10) primary key,Hotel_Name  
varchar(20) not null,city varchar(15) not null,priceperperson int not null);
```

Creating the table “BUS” to store bus details

```
create table if not exists BUS(Bus_Number varchar(10) primary key,Bus_name  
varchar(20) not null,src varchar(20) not null,dest varchar(20) not  
null,priceperperson int not null);
```

Creating the table “FLIGHTS” to store flight details.

```
create table if not exists flights(flight_number varchar(10) primary  
key,flight_name varchar(20) not null,src varchar(20) not null,dest varchar(20)  
not null,fareperperson int not null);
```

Creating the table “USER_HOTEL” to store hotel booking details

```
create table if not exists user_hotel(booking_date date not null,no_of_persons  
int(2),Hotel_id varchar(10) not null,uid int(6) not null,primary  
key(Hotel_id,uid,booking_date),foreign key(uid) references USER(uid) on  
delete cascade,foreign key(Hotel_id) references HOTELS(Hotel_id) on delete  
cascade);
```

Creating the table “USER_BUS” to store bus booking details

```
create table if not exists user_bus(travel_date date not null,no_of_persons int(2),Bus_Number varchar(10) not null,uid int(6) not null,primary key(Bus_Number,uid,travel_date),foreign key(uid) references USER(uid) on delete cascade,foreign key(Bus_Number) references BUS(Bus_Number) on delete cascade);
```

Creating the table “USER_FLIGHT” to store flight booking details

```
create table if not exists user_flight(travel_date date not null,no_of_persons int(2),flight_number varchar(10) not null,uid int(6) not null,primary key(flight_number,uid,travel_date),foreign key(uid) references USER(uid) on delete cascade,foreign key(flight_number) references flights(flight_number) on delete cascade);
```

Creating the table “USERBOOKINGINFO” to store the details of all the transactions done by an user

```
create table if not exists UserBookingInfo(Transaction_id int not null AUTO_INCREMENT,TDate date,Amount int not null,ServiceType varchar(10) not null,ServiceId varchar(10) not null,ServiceDate date not null,uid int(6) not null,foreign key(uid) references USER(uid) on delete cascade,primary key(Transaction_id,uid));
```

Using show tables command after creating tables:

```
MariaDB [traveldbms]> show tables;
+-----+
| Tables_in_traveldbms |
+-----+
| bus
| flights
| hotels
| user
| user_bus
| user_flight
| user_hotel
| userbookinginfo
| userprofile
+-----+
9 rows in set (0.001 sec)
```

Populating the Database:

Populating the table “USER”:

populate_user.py

```
C: > Users > LENOVO > Desktop > DBMS project > populate_user.py > ...
1 import mysql.connector
2 import random
3
4 cnx = mysql.connector.connect(user='root', host = 'localhost',database='traveldbms',password='793164')
5 cursor = cnx.cursor()
6 add_user = ("INSERT INTO user VALUES (%s, %s)")
7
8 k = 123456
9 # Insert new user
10 for i in range(0,10):
11     data_user = (k+i,str(random.randint(10000,11000)))
12     cursor.execute(add_user, data_user)
13
14 cnx.commit()
15 cursor.close()
16 cnx.close()
```

Populating the table “USERPROFILE”:

populate_userprofile.py

```
C: > Users > LENOVO > Desktop > DBMS project > populate_userprofile.py > ...
1 import mysql.connector
2 import random
3 cnx = mysql.connector.connect(user='root', host = 'localhost',database='traveldbms',password='793164')
4 cursor = cnx.cursor()
5 add_userprofile = ("INSERT INTO userprofile(username,age,doorNo,street,pincode,city,uid) VALUES (%s, %s,%s, %s,%s, %s,%s)")
6
7 usernames=["John","Himakar","Charan","Hemanth","Gnan","Karthik","Jasal","Rahul","Sid","Ram"]
8 doornos = ["1-38","2-39","3-48","1-32","2-19","3-10","2-84","2-91","3-28","2-98"]
9 streets = ["Pondy Bazaar","Brigade Road","Tilak Marg","Anna Nagar","MG Road","Connaught Lane",
10 "T Nagar","Avenue Street","Lodhi Road","Church street"]
11 pincodes = ["600001","560001","110001","600009","560008","110002","600003","560004","110005","560006"]
12 cities = ["Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Bengaluru"]
13 uids=[i for i in range(123456,123466)]
14
15 for i in range(0,10):
16     data_user = (usernames[i],random.randint(24,60),doornos[i],streets[i],pincodes[i],cities[i],uids[i])
17     cursor.execute(add_userprofile,data_user)
18
19 cnx.commit()
20 cursor.close()
21 cnx.close()
```

Populating the table “HOTELS”:

populate_hotels.py

```
C: > Users > LENOVO > Desktop > DBMS project > populate_hotel.py > ...
1 import mysql.connector
2 import random
3
4 cnx = mysql.connector.connect(user='root', host = 'localhost',database='traveldbms',password='793164')
5 cursor = cnx.cursor()
6 add_hotel = ("INSERT INTO hotels(hotel_id,hotel_name,city,priceperperson) VALUES (%s, %s ,%s, %s)")
7
8 hotelnames=["Umaid Bhawan","Oberoi Udaivilaa","The Leela Palace","The Taj Palace","ITC Maurya","The Lodhi",
9 "JW Marriott","Sofitel Legend","Royal Orchid","ITC Gardenia"]
10 cities = ["Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Bengaluru"]
11
12
13 for i in range(0,10):
14     data_hotel = ("h"+str(i),hotelnames[i],cities[i],10*(i+1)**2)
15     cursor.execute(add_hotel, data_hotel)
16
17
18 cnx.commit()
19 cursor.close()
20 cnx.close()
```

Populating the table “BUS”:

populate_bus.py

```
C: > Users > LENOVO > Desktop > DBMS project > populate_bus.py > ...
1 import mysql.connector
2 import random
3
4 cnx = mysql.connector.connect(user='root', host = 'localhost',database='traveldbms',password='793164')
5 cursor = cnx.cursor()
6 add_hotel = ("INSERT INTO bus(bus_number,bus_name,src,dest,priceperperson) VALUES (%s, %s , %s, %s, %s)")
7
8 busnames=["Indra","Garuda","Iravat","SRS","Orange","SRS","Indra","Garuda","Iravat","Orange"]
9 src = ["Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Bengaluru"]
10 dest = ["Delhi","Chennai","Bengaluru","Bengaluru","Delhi","Chennai","Bengaluru","Chennai","Bengaluru","Chennai"]
11
12
13 for i in range(0,10):
14     data_hotel = ("b"+str(i),busnames[i]+ " "+src[i][0:3]+"-"+dest[i][0:3],src[i],dest[i],10*((i+1)*2))
15     cursor.execute(add_hotel, data_hotel)
16
17
18 cnx.commit()
19 cursor.close()
20 cnx.close()
```

Populating the table “FLIGHTS”:

populate_flights.py

```
C: > Users > LENOVO > Desktop > DBMS project > populate_flight.py > ...
1 import mysql.connector
2 import random
3
4 cnx = mysql.connector.connect(user='root', host = 'localhost',database='traveldbms',password='793164')
5 cursor = cnx.cursor()
6 add_hotel = ("INSERT INTO flights(flight_number,flight_name,src,dest,fareperperson) VALUES (%s, %s , %s, %s, %s)")
7
8 flightnames=["Air Asia","Indigo","Jet Airways","Spice Jet"]
9 src = ["Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Chennai","Bengaluru","Delhi","Bengaluru"]
10 dest = ["Delhi","Delhi","Bengaluru","Bengaluru","Delhi","Chennai","Delhi","Bengaluru","Chennai"]
11 prices = [8500 , 7500 , 8000 , 9000 , 7600 , 8500 , 7500 , 8000 , 9000 , 7600 ]
12
13 for i in range(0,10):
14     data_hotel = ("f"+str(i),random.choice(flightnames)+" "+src[i][0:3]+"-"+dest[i][0:3],src[i],dest[i],prices[i])
15     cursor.execute(add_hotel, data_hotel)
16
17
18 cnx.commit()
19 cursor.close()
20 cnx.close()
```

Populating the table “USER_HOTEL”:

populate_user_hotel.sql

```
insert into user_hotel values ('2022-11-24',4,'h0',123459);
insert into user_hotel values ('2022-11-19',2,'h1',123456);
insert into user_hotel values ('2022-11-23',8,'h1',123457);
insert into user_hotel values ('2022-11-24',4,'h0',123459);
insert into user_hotel values ('2022-11-23',6,'h0',123457);
insert into user_hotel values ('2022-11-24',4,'h2',123458);
insert into user_hotel values ('2022-11-23',3,'h2',123457);
```

Populating the table “USER_BUS”:

populate_user_bus.sql

```
insert into user_bus values("2022-11-20",5,'b1',123458);
insert into user_bus values("2022-11-20",4,'b3',123457);
insert into user_bus values("2022-11-21",2,'b3',123456);
insert into user_bus values("2022-11-22",4,'b0',123457);
insert into user_bus values("2022-11-22",3,'b2',123458);
insert into user_bus values("2022-11-19",4,'b0',123456);
```

Populating the table “USER_FLIGHT”:

populate_user_flight.sql

```
insert into user_flight VALUES("2022-11-20",3,'f0',123456);
insert into user_flight VALUES("2022-11-20",2,'f1',123457);
insert into user_flight VALUES("2022-11-21",3,'f9',123456);
insert into user_flight VALUES("2022-11-21",3,'f0',123458);
insert into user_flight VALUES("2022-11-21",2,'f1',123457);
insert into user_flight VALUES("2022-11-22",3,'f2',123458);
insert into user_flight VALUES("2022-11-21",3,'f3',123464);
```

Populating the table “USERBOOKINGINFO”:

This table stores the user transaction details. So this table will be automatically populated with help of triggers whenever the users book a hotel or bus or flight .

Tools Used:

Backend : MYSQL (MariaDB) , NodeJS

Frontend: ejs templates , Bootstrap CSS

Queries

Join queries

Regular JOIN:

1)Display the users who booked a room in the hotel “Umaid Bhawan”.

A) select uid,hotel_name from Hotels join user_hotel on Hotels.Hotel_id=user_hotel.hotel_id where Hotel_name='Umaid Bhawan';

```
MariaDB [traveldbms]> select uid,hotel_name from Hotels join user_hotel on Hotels.Hotel_id=user_hotel.hotel_id where
Hotel_name='Umaid Bhawan';
+-----+-----+
| uid | hotel_name |
+-----+-----+
| 123457 | Umaid Bhawan |
| 123459 | Umaid Bhawan |
| 123459 | Umaid Bhawan |
+-----+-----+
3 rows in set (0.002 sec)
```

2)List the names of users who booked the flight with the given flight number ‘f0’.

A) select distinct username,flight_number from userProfile as up Join user_flight as uf on up.uid=uf.uid where flight_number='f0';

```
MariaDB [traveldbms]> select distinct username,flight_number from userProfile as up Join user_flight as uf on up.uid=uf.
uid where flight_number='f0';
+-----+-----+
| username | flight_number |
+-----+-----+
| John     | f0          |
| Charan   | f0          |
+-----+-----+
2 rows in set (0.002 sec)
```

CORRELATED JOIN:

3)Display the names of users who lives in a city and had booked a hotel located in the same city.

A) select up.username,up.city from userProfile as up where up.uid in (select uh.uid from hotels as h JOIN user_hotel as uh on uh.hotel_id=h.hotel_id where h.city=up.city and uh.uid = up.uid);

```
MariaDB [traveldbms]> select up.username,up.city from userprofile as up where up.uid in (select uh.uid from hotels as h JOIN user_hotel as uh on uh.hotel_id=h.hotel_id where h.city=up.city and uh.uid = up.uid);
+-----+-----+
| username | city   |
+-----+-----+
| Charan   | Delhi  |
| Hemanth  | Chennai|
| Himakar  | Bengaluru|
+-----+-----+
3 rows in set (0.001 sec)
```

4)List the names of users who booked a flight ticket to "Delhi" from his home city.

A) select up.username,up.city from userprofile as up where up.uid in (select uid from user_flight as uf JOIN flights as f on uf.flight_number = f.flight_number where dest="Delhi" and src= up.city);

```
MariaDB [traveldbms]> select up.username,up.city from userprofile as up where up.uid in (select uid from user_flight as uf JOIN flights as f on uf.flight_number = f.flight_number where dest="Delhi" and src= up.city);
+-----+-----+
| username | city   |
+-----+-----+
| Himakar  | Bengaluru|
| John     | Chennai|
+-----+-----+
2 rows in set (0.002 sec)
```

NESTED JOIN

5)Print the names of users who booked a bus ticket to "Chennai".

A) select username from userProfile as up where up.uid in (select uid from user_bus as ub JOIN bus as b on ub.bus_number = b.bus_number where src="Chennai");

```
MariaDB [traveldbms]> select username from userProfile as up where up.uid in
    -> (select uid from user_bus as ub JOIN bus as b on ub.bus_number = b.bus_number where src="Chennai");
+-----+
| username |
+-----+
| John    |
| Himakar |
+-----+
2 rows in set (0.001 sec)
```

6)Display the names of the hotels located in a city to which users can reach by flight.

A) select H.hotel_name,H.city from Hotels as H where H.city in (select distinct dest from flights);

```
MariaDB [traveldbms]> select H.hotel_name,H.city from Hotels as H where H.city in (select distinct dest from flights);
+-----+-----+
| hotel_name | city   |
+-----+-----+
| Umaid Bhawan | Chennai |
| Oberoi Udaivilaa | Bengaluru |
| The Leela Palace | Delhi |
| The Taj Palace | Chennai |
| ITC Maurya | Bengaluru |
| The Lodhi | Delhi |
| JW Marriott | Chennai |
| Sofitel Legend | Bengaluru |
| Royal Orchid | Delhi |
| ITC Gardenia | Bengaluru |
+-----+-----+
10 rows in set (0.001 sec)
```

Aggregate Functions

1) Calculate the average amount spent by an user in each hotel.

A) select uh.hotel_id,AVG(no_of_persons*priceperperson) as AVG_Cost from user_hotel as uh JOIN hotels as h where h.hotel_id=uh.hotel_id Group By (uh.hotel_id);

```
MariaDB [traveldbms]> select uh.hotel_id,AVG(no_of_persons*priceperperson) as AVG_Cost from user_hotel as uh
JOIN hotels as h where h.hotel_id=uh.hotel_id Group By (uh.hotel_id);
+-----+-----+
| hotel_id | AVG_Cost |
+-----+-----+
| h0       | 46.6667 |
| h1       | 200.0000 |
| h2       | 315.0000 |
+-----+-----+
3 rows in set (0.001 sec)
```

2) Find the places which have minimum incoming buses.

A) select dest,count(*) as c from Bus Group By dest order by c LIMIT 1;

```
MariaDB [traveldbms]> select dest,count(*) as c from Bus Group By dest order by c LIMIT 1;
+-----+---+
| dest | c |
+-----+---+
| Delhi | 2 |
+-----+---+
1 row in set (0.001 sec)
```

3)Calculate the number of users travelled from “Bengaluru” to “Chennai” by bus.

A) select count(uid) as c from user_bus JOIN bus on bus.bus_number = user_bus.bus_number where bus.src="Bengaluru" and bus.dest="Chennai";

```
MariaDB [traveldbms]> select count(uid) as c from user_bus JOIN bus on bus.bus_number = user_bus.bus_number
where bus.src="Bengaluru" and bus.dest="Chennai";
+---+
| c |
+---+
| 1 |
+---+
1 row in set (0.001 sec)
```

Set Operations

1)Find the users who always prefer flight.

A)

```
(select uf.uid from user_flight as uf Group By uf.uid Having count(flight_number) >0)
EXCEPT
(select ub.uid from user_bus as ub Group By ub.uid Having count(bus_number)>0);
```

```
MariaDB [traveldbms]> (select uf.uid from user_flight as uf Group By uf.uid Having count(flight_number) >0)
    -> EXCEPT
    -> (select ub.uid from user_bus as ub Group By ub.uid Having count(bus_number)>0);
+-----+
| uid   |
+-----+
| 123464 |
+-----+
1 row in set (0.001 sec)
```

2) List the passengers who have travelled from Bengaluru to Delhi by flight and did not returned to Bengaluru from Delhi by flight.

A) (select uid from user_flight as uf JOIN flights as f where f.flight_number = uf.flight_number and f.src="Bengaluru" and f.dest = "Delhi")

EXCEPT (select uid from user_flight as uf1 JOIN flights as f1 where f1.flight_number = uf1.flight_number and f1.dest="Bengaluru" and f1.src = "Delhi");

```
MariaDB [traveldbms]> (select uid from user_flight as uf JOIN flights as f where f.flight_number = uf.flight_number
and f.src="Bengaluru" and f.dest = "Delhi")
    -> EXCEPT (select uid from user_flight as uf1 JOIN flights as f1 where f1.flight_number = uf1.flight_number and
f1.dest="Bengaluru" and f1.src = "Delhi");
+-----+
| uid   |
+-----+
| 123457 |
+-----+
1 row in set (0.003 sec)
```

3) Display the userid's of the passengers who have travelled from “Bengaluru” to “Delhi” by flight and also stayed in hotel with hotel_id "h0".

sq:- (select uid from user_flight as uf JOIN flights as f where f.flight_number = uf.flight_number and f.src="Bengaluru" and f.dest = "Delhi")

INTERSECT (select uh.uid from user_hotel as uh where uh.hotel_id="h0");

```
MariaDB [traveldbms]> (select uid from user_flight as uf JOIN flights as f where f.flight_number = uf.flight_number
and f.src="Bengaluru" and f.dest = "Delhi")
    -> EXCEPT (select uid from user_flight as uf1 JOIN flights as f1 where f1.flight_number = uf1.flight_number and
f1.dest="Bengaluru" and f1.src = "Delhi");
+-----+
| uid   |
+-----+
| 123457 |
+-----+
1 row in set (0.003 sec)
```

View

Creating a view to show all important details of user and the flight booked by that user together.

```
MariaDB [traveldbms]> CREATE VIEW flightusers AS
    -> select up.uid,up.username,uf.no_of_persons,f.flight_name,f.flight_number,f.fareperperson*uf.no_of_persons as totalcost
,uf.travel_date
    -> from userprofile as up join user_flight as uf on up.uid=uf.uid join flights as f on f.flight_number=uf.flight_number;
Query OK, 0 rows affected (0.014 sec)

MariaDB [traveldbms]> select * from flightusers;
+-----+-----+-----+-----+-----+-----+
| uid | username | no_of_persons | flight_name | flight_number | totalcost | travel_date |
+-----+-----+-----+-----+-----+-----+
| 123456 | John | 3 | Indigo Che-Del | f0 | 25500 | 2022-11-20 |
| 123458 | Charan | 3 | Indigo Che-Del | f0 | 25500 | 2022-11-21 |
| 123457 | Himakar | 2 | Air Asia Ben-Del | f1 | 15000 | 2022-11-20 |
| 123457 | Himakar | 2 | Air Asia Ben-Del | f1 | 15000 | 2022-11-21 |
| 123458 | Charan | 3 | Jet Airways Del-Ben | f2 | 24000 | 2022-11-22 |
| 123464 | Sid | 3 | Indigo Che-Ben | f3 | 27000 | 2022-11-21 |
| 123456 | John | 3 | Air Asia Ben-Che | f9 | 22800 | 2022-11-21 |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.009 sec)
```

Query: Update the username to Johnny for the user who booked the flight with flight number f0 for the date “20-11-2022”

A) UPDATE flightusers SET username=”Johnny” where flight_number='f0' and travel_date='2022-11-20';

```
MariaDB [traveldbms]> UPDATE flightusers SET username='Johnny' where flight_number='f0' and travel_date='2022-11-20';
Query OK, 1 row affected (0.006 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [traveldbms]> select * from flightusers;
+-----+-----+-----+-----+-----+-----+
| uid | username | no_of_persons | flight_name | flight_number | totalcost | travel_date |
+-----+-----+-----+-----+-----+-----+
| 123456 | Johnny | 3 | Indigo Che-Del | f0 | 25500 | 2022-11-20 |
| 123458 | Charan | 3 | Indigo Che-Del | f0 | 25500 | 2022-11-21 |
| 123457 | Himakar | 2 | Air Asia Ben-Del | f1 | 15000 | 2022-11-20 |
| 123457 | Himakar | 2 | Air Asia Ben-Del | f1 | 15000 | 2022-11-21 |
| 123458 | Charan | 3 | Jet Airways Del-Ben | f2 | 24000 | 2022-11-22 |
| 123464 | Sid | 3 | Indigo Che-Ben | f3 | 27000 | 2022-11-21 |
| 123456 | Johnny | 3 | Air Asia Ben-Che | f9 | 22800 | 2022-11-21 |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)
```

We can also observe that the username is modified in the userprofile’s table.

```
MariaDB [traveldbms]> select * from userprofile;
+-----+-----+-----+-----+-----+-----+
| username | Age | doorNo | street | pincode | city | uid |
+-----+-----+-----+-----+-----+-----+
| Charan | 51 | 3-40 | Tilak Marg | 110001 | Delhi | 123458 |
| Gnan | 29 | 2-19 | MG Road | 560008 | Bengaluru | 123460 |
| Hemanth | 50 | 1-32 | Anna Nagar | 600009 | Chennai | 123459 |
| Himakar | 27 | 2-39 | Brigade Road | 560001 | Bengaluru | 123457 |
| Jasal | 52 | 1-84 | T Nagar | 600003 | Chennai | 123462 |
| Johnny | 30 | 1-38 | Pondy Bazaar | 600001 | Chennai | 123456 |
| Karthik | 44 | 3-10 | Connaught Lane | 110002 | Delhi | 123461 |
| Rahul | 50 | 2-91 | Avenue Street | 560004 | Bengaluru | 123463 |
| Ram | 53 | 2-98 | Church street | 560006 | Bengaluru | 123465 |
| Sid | 32 | 3-20 | Lodhi Road | 110005 | Delhi | 123464 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.001 sec)
```

Triggers (Functions or Procedures)

Updating the userbookinginfo table when an user booked a hotel:

Function to find the total amount to be paid by the user for booking a hotel .

```
Delimiter $  
CREATE FUNCTION find_hotelprice(hotelid varchar(20),cnt int)  
RETURNS int  
DETERMINISTIC  
BEGIN  
DECLARE RESULT int;  
select cnt*priceperperson into RESULT from hotels where hotel_id=hotelid;  
return RESULT;  
END$  
Delimiter ;
```

Trigger to be called when the user booked a hotel and updates the userbookinginfo table with the transaction details.

```
Delimiter $  
create trigger add_transaction_details after insert on user_hotel  
for each row  
Begin  
| insert into userbookinginfo(TDate,Amount,ServiceType,ServiceId,ServiceDate,Uid)  
| VALUES(SYSDATE(),find_hotelprice(new.hotel_id,new.no_of_persons),"hotel",new.hotel_id,new.booking_date,new.uid);  
End$  
Delimiter ;
```

Results screenshots:

Rows in the userbookinginfo table before:

```
MariaDB [traveldbms]> select * from userbookinginfo;  
+-----+-----+-----+-----+-----+-----+  
| Transaction_id | TDate | Amount | ServiceType | ServiceId | ServiceDate | uid |  
+-----+-----+-----+-----+-----+-----+  
| 7 | 2022-11-20 | 40 | hotel | h0 | 2022-11-24 | 123459 |  
| 8 | 2022-11-20 | 80 | hotel | h1 | 2022-11-19 | 123456 |  
| 9 | 2022-11-20 | 320 | hotel | h1 | 2022-11-23 | 123457 |  
| 10 | 2022-11-20 | 40 | hotel | h0 | 2022-11-25 | 123459 |  
| 11 | 2022-11-20 | 60 | hotel | h0 | 2022-11-23 | 123457 |  
| 12 | 2022-11-20 | 360 | hotel | h2 | 2022-11-24 | 123458 |  
| 13 | 2022-11-20 | 270 | hotel | h2 | 2022-11-23 | 123457 |  
| 14 | 2022-11-20 | 200 | bus | b1 | 2022-11-20 | 123458 |  
| 15 | 2022-11-20 | 320 | bus | b3 | 2022-11-20 | 123457 |  
| 16 | 2022-11-20 | 160 | bus | b3 | 2022-11-21 | 123456 |  
| 17 | 2022-11-20 | 80 | bus | b0 | 2022-11-22 | 123457 |  
| 18 | 2022-11-20 | 180 | bus | b2 | 2022-11-22 | 123458 |  
| 19 | 2022-11-20 | 80 | bus | b0 | 2022-11-19 | 123456 |  
| 20 | 2022-11-20 | 25500 | flight | f0 | 2022-11-20 | 123456 |  
| 21 | 2022-11-20 | 15000 | flight | f1 | 2022-11-20 | 123457 |  
| 22 | 2022-11-20 | 22800 | flight | f9 | 2022-11-21 | 123456 |  
| 23 | 2022-11-20 | 25500 | flight | f0 | 2022-11-21 | 123458 |  
| 24 | 2022-11-20 | 15000 | flight | f1 | 2022-11-21 | 123457 |  
| 25 | 2022-11-20 | 24000 | flight | f2 | 2022-11-22 | 123458 |  
| 26 | 2022-11-20 | 27000 | flight | f3 | 2022-11-21 | 123464 |  
+-----+-----+-----+-----+-----+  
20 rows in set (0.004 sec)
```

Inserting a row into user_hotel;

```
MariaDB [traveldbms]> insert into user_hotel values ('2022-11-23',6,'h3',123458);  
Query OK, 1 row affected (0.004 sec)
```

Rows in the userbookinginfo table After:

```
MariaDB [traveldbms]> select * from userbookinginfo;
+-----+-----+-----+-----+-----+-----+-----+
| Transaction_id | TDate      | Amount | ServiceType | ServiceId | ServiceDate | uid   |
+-----+-----+-----+-----+-----+-----+-----+
|      7 | 2022-11-20 |    40 | hotel       | h0        | 2022-11-24 | 123459 |
|      8 | 2022-11-20 |    80 | hotel       | h1        | 2022-11-19 | 123456 |
|      9 | 2022-11-20 |   320 | hotel       | h1        | 2022-11-23 | 123457 |
|     10 | 2022-11-20 |    40 | hotel       | h0        | 2022-11-25 | 123459 |
|     11 | 2022-11-20 |    60 | hotel       | h0        | 2022-11-23 | 123457 |
|     12 | 2022-11-20 |   360 | hotel       | h2        | 2022-11-24 | 123458 |
|     13 | 2022-11-20 |   270 | hotel       | h2        | 2022-11-23 | 123457 |
|     14 | 2022-11-20 |   200 | bus         | b1        | 2022-11-20 | 123458 |
|     15 | 2022-11-20 |   320 | bus         | b3        | 2022-11-20 | 123457 |
|     16 | 2022-11-20 |   160 | bus         | b3        | 2022-11-21 | 123456 |
|     17 | 2022-11-20 |    80 | bus         | b0        | 2022-11-22 | 123457 |
|     18 | 2022-11-20 |   180 | bus         | b2        | 2022-11-22 | 123458 |
|     19 | 2022-11-20 |    80 | bus         | b0        | 2022-11-19 | 123456 |
|     20 | 2022-11-20 | 25500 | flight      | f0        | 2022-11-20 | 123456 |
|     21 | 2022-11-20 | 15000 | flight      | f1        | 2022-11-20 | 123457 |
|     22 | 2022-11-20 | 22800 | flight      | f9        | 2022-11-21 | 123456 |
|     23 | 2022-11-20 | 25500 | flight      | f0        | 2022-11-21 | 123458 |
|     24 | 2022-11-20 | 15000 | flight      | f1        | 2022-11-21 | 123457 |
|     25 | 2022-11-20 | 24000 | flight      | f2        | 2022-11-22 | 123458 |
|     26 | 2022-11-20 | 27000 | flight      | f3        | 2022-11-21 | 123464 |
|     27 | 2022-11-20 |   960 | hotel       | h3        | 2022-11-23 | 123458 |
+-----+-----+-----+-----+-----+-----+-----+
21 rows in set (0.001 sec)
```

Similarly the below code is used to update the userbookinginfo table when an user booked a bus.

```
--t2)Update money transaction details into transaction table when user booked a bus.
Delimiter $
CREATE FUNCTION calculate_busfare(busno varchar(20),cnt int)
RETURNS int
DETERMINISTIC
BEGIN
DECLARE RESULT int;
select cnt*priceperperson into RESULT from bus where bus_number=busno;
return RESULT;
END$
Delimiter ;

Delimiter $
create trigger add_transaction_details_bus after insert on user_bus
for each row
Begin
insert into userbookinginfo(TDate,Amount,ServiceType,ServiceId,ServiceDate,Uid)
VALUES(SYSDATE(),calculate_busfare(new.bus_number,new.no_of_persons),"bus",new.bus_number,new.travel_date,new.uid);
End$
Delimiter ;
```

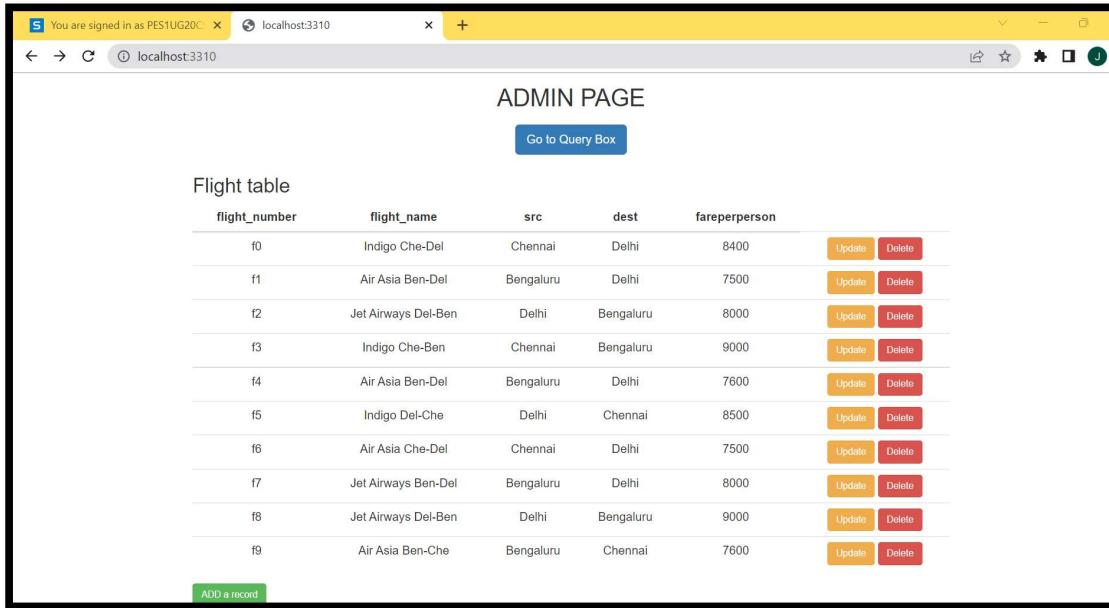
Similarly the below code is used to update the userbookinginfo table when an user booked a bus.

```
--t3)Update money transaction details into transaction table when user booked a flight.
Delimiter $
CREATE FUNCTION calculate_flightfare(flightno varchar(20),cnt int)
RETURNS int
DETERMINISTIC
BEGIN
DECLARE RESULT int;
select cnt*fareperperson into RESULT from flights where flight_number=flightno;
return RESULT;
END$
Delimiter ;

Delimiter $
create trigger add_transaction_details_flights after insert on user_flight
for each row
Begin
insert into userbookinginfo(TDate,Amount,ServiceType,ServiceId,ServiceDate,Uid)
VALUES(SYSDATE(),calculate_flightfare(new.flight_number,new.no_of_persons),"flight",new.flight_number,new.travel_date,new.uid);
End$
Delimiter ;
```

Developing the Frontend

The first page looks like this. From here admin can open a query box to run any query or can he can do CRUD operations on the flights table.

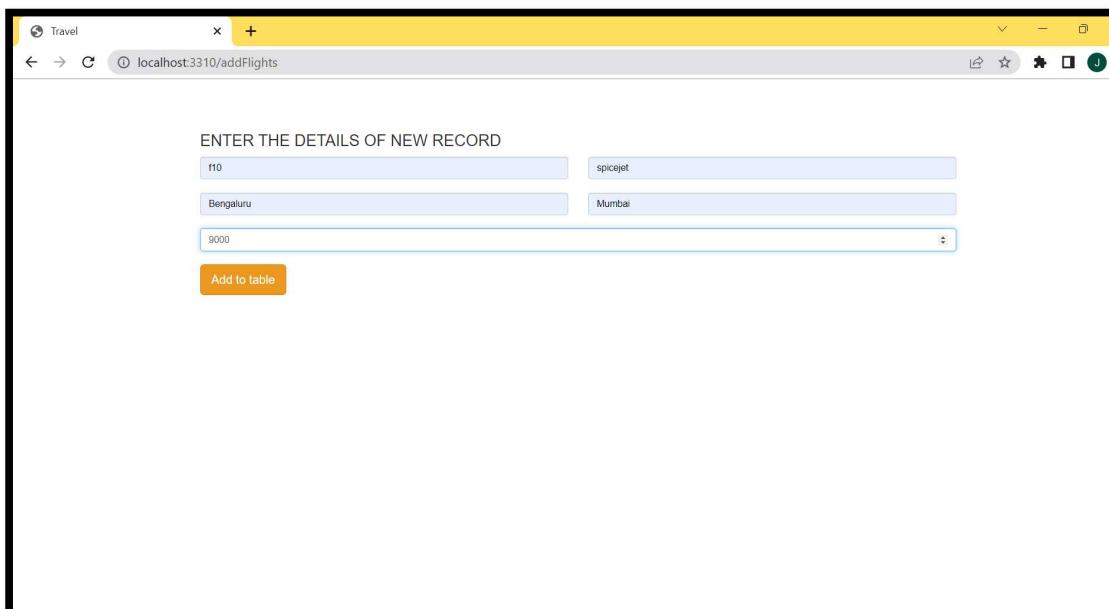


A screenshot of a web browser showing an 'ADMIN PAGE'. The title bar says 'You are signed in as PES1UG20...' and 'localhost:3310'. Below the title bar is a navigation bar with back, forward, refresh, and search buttons. The main content area has a yellow header 'ADMIN PAGE' and a blue button 'Go to Query Box'. Below this is a section titled 'Flight table' containing a table with 10 rows of flight data. The columns are labeled: flight_number, flight_name, src, dest, fareperperson. Each row has two buttons at the end: 'Update' (orange) and 'Delete' (red). At the bottom left of the table is a green button 'ADD a record'.

flight_number	flight_name	src	dest	fareperperson	Update	Delete
f0	Indigo Che-Del	Chennai	Delhi	8400	<button>Update</button>	<button>Delete</button>
f1	Air Asia Ben-Del	Bengaluru	Delhi	7500	<button>Update</button>	<button>Delete</button>
f2	Jet Airways Del-Ben	Delhi	Bengaluru	8000	<button>Update</button>	<button>Delete</button>
f3	Indigo Che-Ben	Chennai	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f4	Air Asia Ben-Del	Bengaluru	Delhi	7600	<button>Update</button>	<button>Delete</button>
f5	Indigo Del-Che	Delhi	Chennai	8500	<button>Update</button>	<button>Delete</button>
f6	Air Asia Che-Del	Chennai	Delhi	7500	<button>Update</button>	<button>Delete</button>
f7	Jet Airways Ben-Del	Bengaluru	Delhi	8000	<button>Update</button>	<button>Delete</button>
f8	Jet Airways Del-Ben	Delhi	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f9	Air Asia Ben-Che	Bengaluru	Chennai	7600	<button>Update</button>	<button>Delete</button>

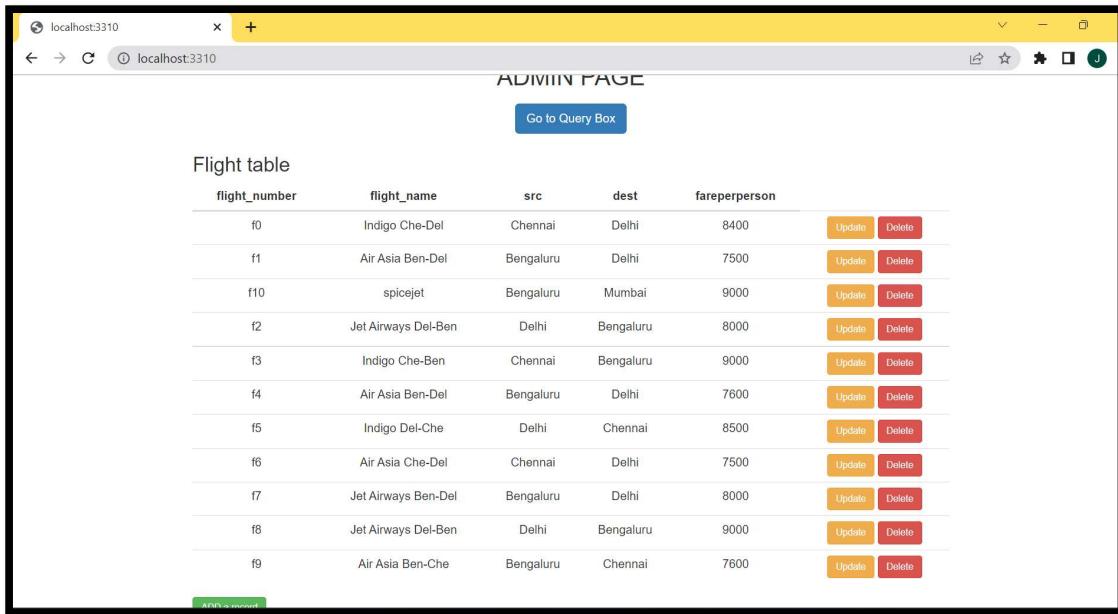
1. Addition, Modification and Deletion of records from flights table.

Adding a record into table with flight number f10:



A screenshot of a web browser showing a form titled 'ENTER THE DETAILS OF NEW RECORD'. The title bar says 'Travel' and 'localhost:3310/addFlights'. The form has four input fields: 'flight_number' (containing 'f10'), 'flight_name' (containing 'spicejet'), 'src' (containing 'Bengaluru'), 'dest' (containing 'Mumbai'), and a 'fareperperson' dropdown menu set to '9000'. At the bottom is an orange button 'Add to table'.

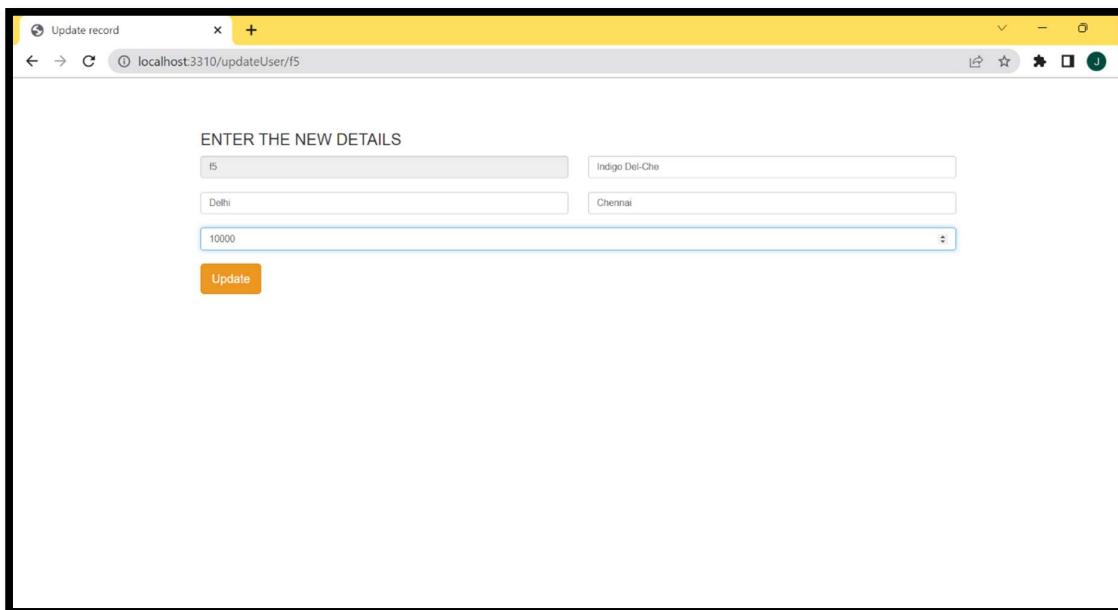
Table after adding a record:



The screenshot shows a web browser window titled "ADMIN PAGE" at "localhost:3310". The main content is a table titled "Flight table" with the following data:

flight_number	flight_name	src	dest	fareperperson	Update	Delete
f0	Indigo Che-Del	Chennai	Delhi	8400	<button>Update</button>	<button>Delete</button>
f1	Air Asia Ben-Del	Bengaluru	Delhi	7500	<button>Update</button>	<button>Delete</button>
f10	spicejet	Bengaluru	Mumbai	9000	<button>Update</button>	<button>Delete</button>
f2	Jet Airways Del-Ben	Delhi	Bengaluru	8000	<button>Update</button>	<button>Delete</button>
f3	Indigo Che-Ben	Chennai	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f4	Air Asia Ben-Del	Bengaluru	Delhi	7600	<button>Update</button>	<button>Delete</button>
f5	Indigo Del-Che	Delhi	Chennai	8500	<button>Update</button>	<button>Delete</button>
f6	Air Asia Che-Del	Chennai	Delhi	7500	<button>Update</button>	<button>Delete</button>
f7	Jet Airways Ben-Del	Bengaluru	Delhi	8000	<button>Update</button>	<button>Delete</button>
f8	Jet Airways Del-Ben	Delhi	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f9	Air Asia Ben-Che	Bengaluru	Chennai	7600	<button>Update</button>	<button>Delete</button>

Updating a record in the table with flight number :



The screenshot shows a web browser window titled "Update record" at "localhost:3310/updateUser/f5". The form is titled "ENTER THE NEW DETAILS" and contains the following fields:

<input type="text" value="f5"/>	<input type="text" value="Indigo Del-Che"/>
<input type="text" value="Delhi"/>	<input type="text" value="Chennai"/>
<input type="text" value="10000"/>	
<input type="button" value="Update"/>	

Updated value

A screenshot of a web browser window titled "localhost:3310". The page displays a "Flight table" with the following data:

flight_number	flight_name	src	dest	fareperperson	Update	Delete
f0	Indigo Che-Del	Chennai	Delhi	8400	<button>Update</button>	<button>Delete</button>
f1	Air Asia Ben-Del	Bengaluru	Delhi	7500	<button>Update</button>	<button>Delete</button>
f10	spicejet	Bengaluru	Mumbai	9000	<button>Update</button>	<button>Delete</button>
f2	Jet Airways Del-Ben	Delhi	Bengaluru	8000	<button>Update</button>	<button>Delete</button>
f3	Indigo Che-Ben	Chennai	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f4	Air Asia Ben-Del	Bengaluru	Delhi	7600	<button>Update</button>	<button>Delete</button>
f5	Indigo Del-Che	Delhi	Chennai	10000	<button>Update</button>	<button>Delete</button>
f6	Air Asia Che-Del	Chennai	Delhi	7500	<button>Update</button>	<button>Delete</button>
f7	Jet Airways Ben-Del	Bengaluru	Delhi	8000	<button>Update</button>	<button>Delete</button>
f8	Jet Airways Del-Ben	Delhi	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f9	Air Asia Ben-Che	Bengaluru	Chennai	7600	<button>Update</button>	<button>Delete</button>

At the bottom left, there is a green button labeled "ADD a record". At the top right, there is a blue button labeled "Go to Query Box".

Deleting a record with flight number f10:

A screenshot of a web browser window titled "localhost:3310". The page displays a "Flight table" with the following data:

flight_number	flight_name	src	dest	fareperperson	Update	Delete
f0	Indigo Che-Del	Chennai	Delhi	8400	<button>Update</button>	<button>Delete</button>
f1	Air Asia Ben-Del	Bengaluru	Delhi	7500	<button>Update</button>	<button>Delete</button>
f10	spicejet	Bengaluru	Mumbai	9000	<button>Update</button>	<button>Delete</button>
f2	Jet Airways Del-Ben	Delhi	Bengaluru	8000	<button>Update</button>	<button>Delete</button>
f3	Indigo Che-Ben	Chennai	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f4	Air Asia Ben-Del	Bengaluru	Delhi	7600	<button>Update</button>	<button>Delete</button>
f5	Indigo Del-Che	Delhi	Chennai	10000	<button>Update</button>	<button>Delete</button>
f6	Air Asia Che-Del	Chennai	Delhi	7500	<button>Update</button>	<button>Delete</button>
f7	Jet Airways Ben-Del	Bengaluru	Delhi	8000	<button>Update</button>	<button>Delete</button>
f8	Jet Airways Del-Ben	Delhi	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f9	Air Asia Ben-Che	Bengaluru	Chennai	7600	<button>Update</button>	<button>Delete</button>

At the bottom left, there is a yellow button labeled "localhost:3310/deleteUser/f10". At the bottom center, there is a green button labeled "ADD a record". At the top right, there is a blue button labeled "Go to Query Box".

Table after deletion:

The screenshot shows a web browser window titled "ADMIN PAGE". Below the title is a button labeled "Go to Query Box". The main content is a table titled "Flight table" with the following columns: flight_number, flight_name, src, dest, fareperperson. Each row contains a set of flight details and two buttons: "Update" and "Delete". At the bottom left of the table is a green button labeled "ADD a record". The browser's address bar shows "localhost:3310". The taskbar at the bottom of the screen displays various application icons.

flight_number	flight_name	src	dest	fareperperson	Update	Delete
f0	Indigo Che-Del	Chennai	Delhi	8400	<button>Update</button>	<button>Delete</button>
f1	Air Asia Ben-Del	Bengaluru	Delhi	7500	<button>Update</button>	<button>Delete</button>
f2	Jet Airways Del-Ben	Delhi	Bengaluru	8000	<button>Update</button>	<button>Delete</button>
f3	Indigo Che-Ben	Chennai	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f4	Air Asia Ben-Del	Bengaluru	Delhi	7600	<button>Update</button>	<button>Delete</button>
f5	Indigo Del-Che	Delhi	Chennai	10000	<button>Update</button>	<button>Delete</button>
f6	Air Asia Che-Del	Chennai	Delhi	7500	<button>Update</button>	<button>Delete</button>
f7	Jet Airways Ben-Del	Bengaluru	Delhi	8000	<button>Update</button>	<button>Delete</button>
f8	Jet Airways Del-Ben	Delhi	Bengaluru	9000	<button>Update</button>	<button>Delete</button>
f9	Air Asia Ben-Che	Bengaluru	Chennai	7600	<button>Update</button>	<button>Delete</button>

2. There should be a window to accept and run any SQL statement and display the result.

Query1:

The screenshot shows a web browser window titled "Query Box". The address bar shows "localhost:3310/query". The main content area is titled "Query box" and contains a text input field with the following SQL query: "Select * from user_hotel JOIN hotels on user_hotel.hotel_id = hotels.hotel_id;". Below the input field is a green "Run Query" button. The browser's address bar also shows "localhost:3310" and a message "You are signed in as PES1UG20C". The taskbar at the bottom of the screen displays various application icons.

You are signed in as PES1UG20C

Query Box

localhost:3310/query

Query box

Enter your query here

Run Query

Results of the query :

```
select * from user_hotel JOIN hotels on user_hotel.hotel_id = hotels.hotel_id;
```

booking_date	no_of_persons	Hotel_Id	uid	Hotel_Name	city	priceperperson
Wed Nov 23 2022 00:00:00 GMT+0530 (India Standard Time)	6	h0	123457	Umaid Bhawan	Chennai	10
Thu Nov 24 2022 00:00:00 GMT+0530 (India Standard Time)	4	h0	123459	Umaid Bhawan	Chennai	10
Fri Nov 25 2022 00:00:00 GMT+0530 (India Standard Time)	4	h0	123459	Umaid Bhawan	Chennai	10
Sat Nov 19 2022 00:00:00 GMT+0530 (India Standard Time)	2	h1	123456	Oberoi Udaivilaa	Bengaluru	40
Wed Nov 23 2022 00:00:00 GMT+0530 (India Standard Time)	8	h1	123457	Oberoi Udaivilaa	Bengaluru	40
Wed Nov 23 2022 00:00:00 GMT+0530 (India Standard Time)	3	h2	123457	The Leela Palace	Delhi	90
Thu Nov 24 2022 00:00:00 GMT+0530 (India Standard Time)	4	h2	123458	The Leela Palace	Delhi	90
Wed Nov 23 2022 00:00:00 GMT+0530 (India Standard Time)	6	h3	123458	The Taj Palace	Chennai	160

Applied this operation on 8 records

Query2:

You are signed in as PES1UG20C

Query Box

localhost:3310/query

Query box

show tables;

Run Query

