**DBMS - MINI PROJECT**

**TRAVEL MANAGEMENT SYSTEM**

**Submitted By:**

Name: Jolepalyam Himakar

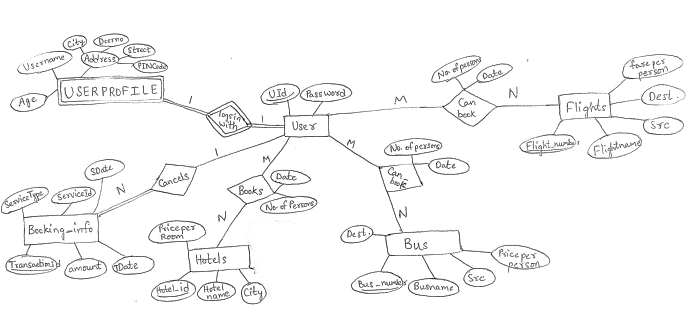
SRN: PES1UG20CS181

V Semester Section C

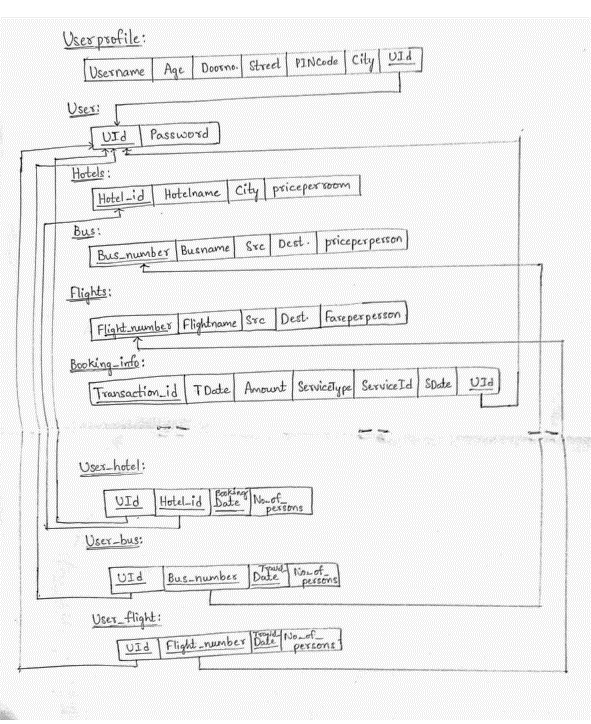
# **ABSTRACT**

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);

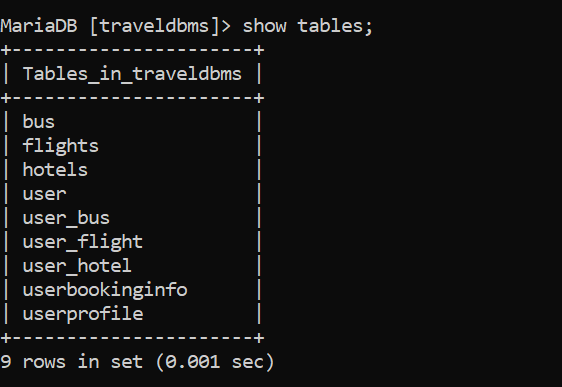
**Creating the table “USER\_FLIGHT” to store flight booking details**

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:**



# **Populating the Database:**

**Populating the table “USER”:**

**populate\_user.py**

****

**Populating the table “USERPROFILE”:**

**populate\_userprofile.py**

****

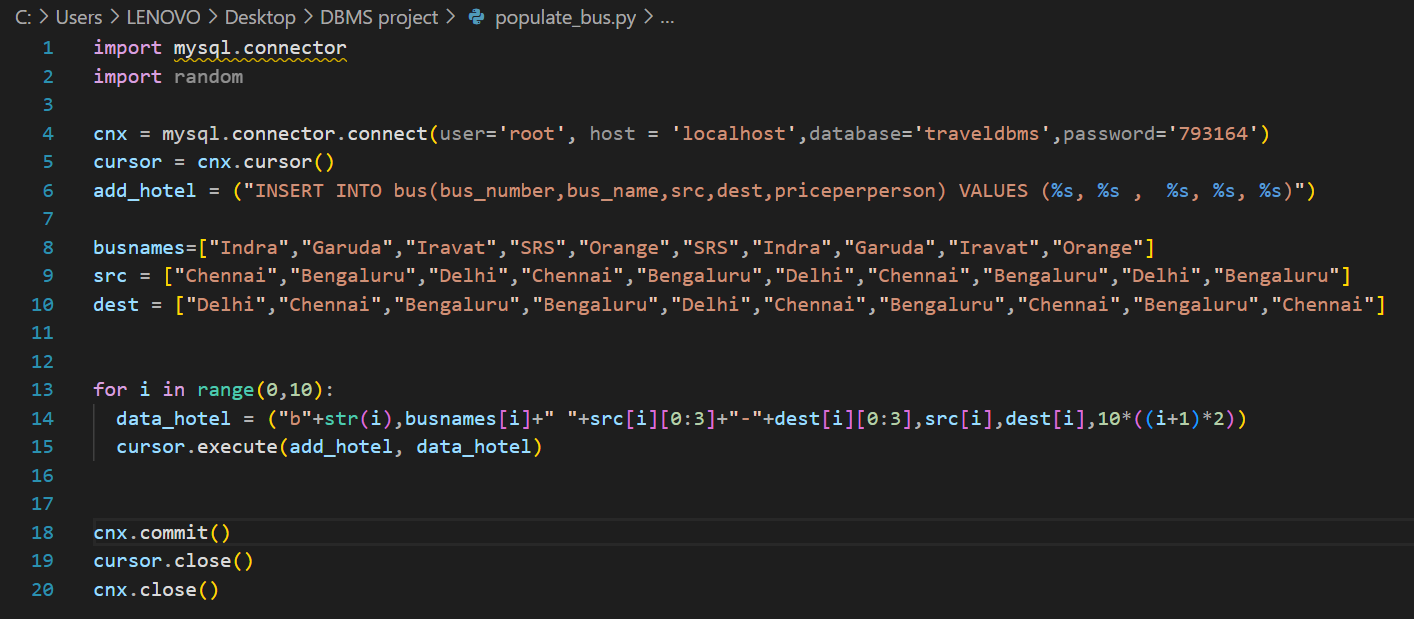
**Populating the table “HOTELS”:**

**populate\_hotels.py**

****

**Populating the table “BUS”:**

**populate\_bus.py**

****

**Populating the table “FLIGHTS”:**

**populate\_flights.py**

****

**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

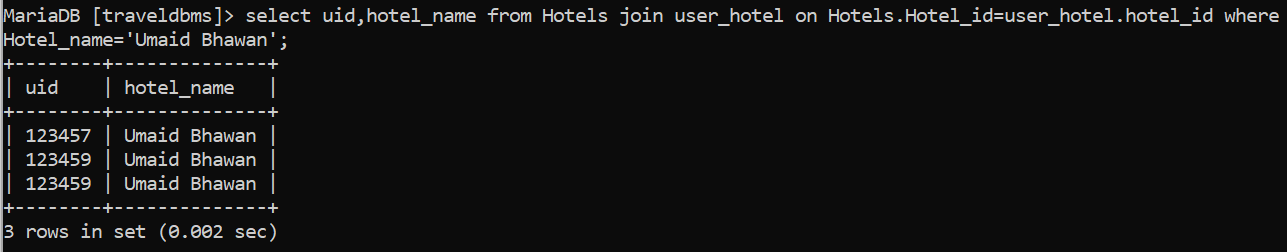
# **Queries**

**Join queries**

**Regular JOIN:**

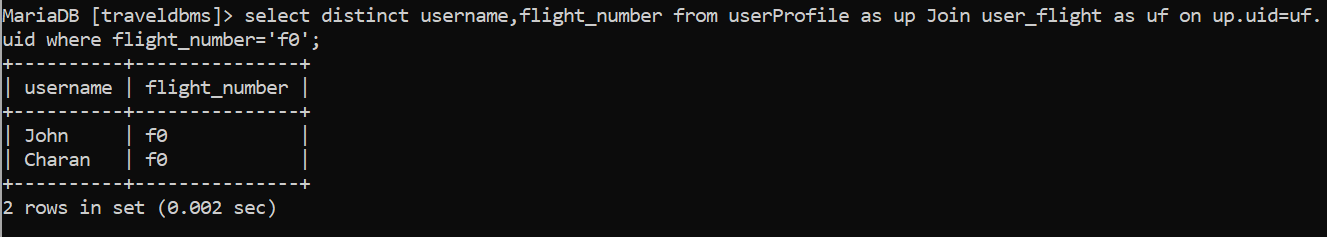
**1)Display the users who booked a room in the hotel “Umaid Bhawan”.**

**A)** select uid,hotel\_name from Hotels join user\_hotel on Hotels.Hotel\_id=user\_hotel.hotel\_id where Hotel\_name='Umaid Bhawan';

****

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

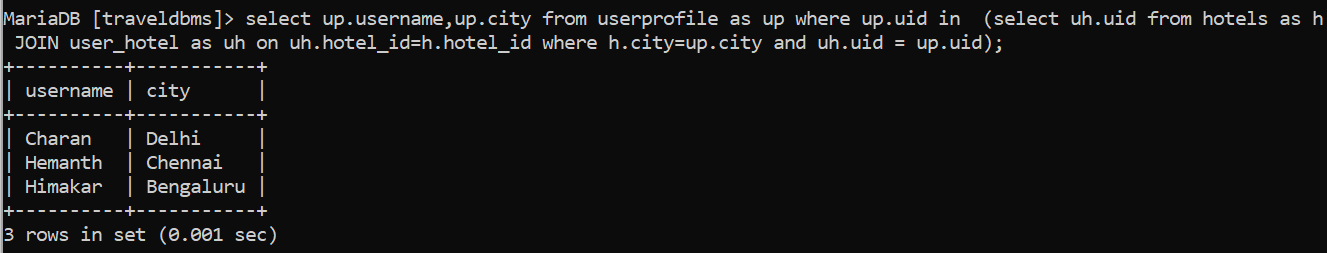
**A)** select distinct username,flight\_number from userProfile as up Join user\_flight as uf on up.uid=uf.uid where flight\_number='f0';



**CORRELATED JOIN:**

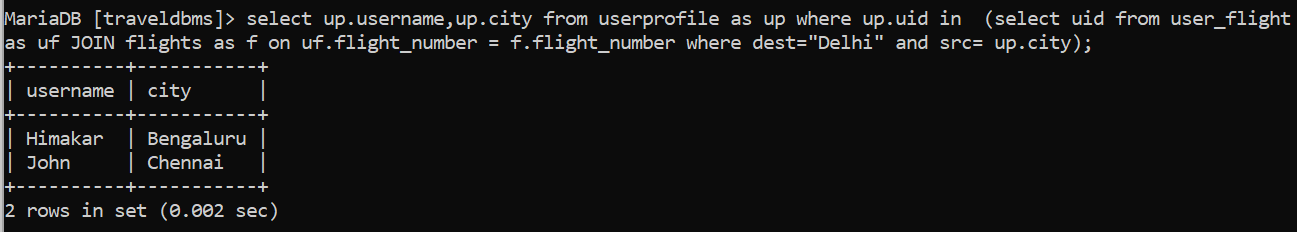
**3)Display the names of users who lives in a city and had booked a hotel located in the same city.**

**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);



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

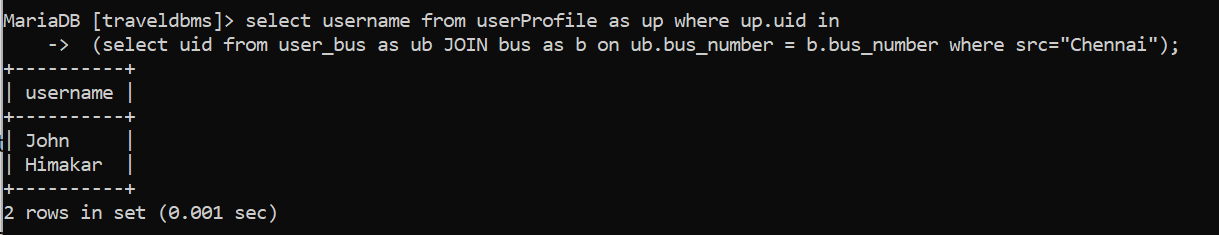
**A)** select up.username,up.city from userprofile as up where up.uid in (select uid from user\_flight as uf JOIN flights as f on uf.flight\_number = f.flight\_number where dest="Delhi" and src= up.city);



**NESTED JOIN**

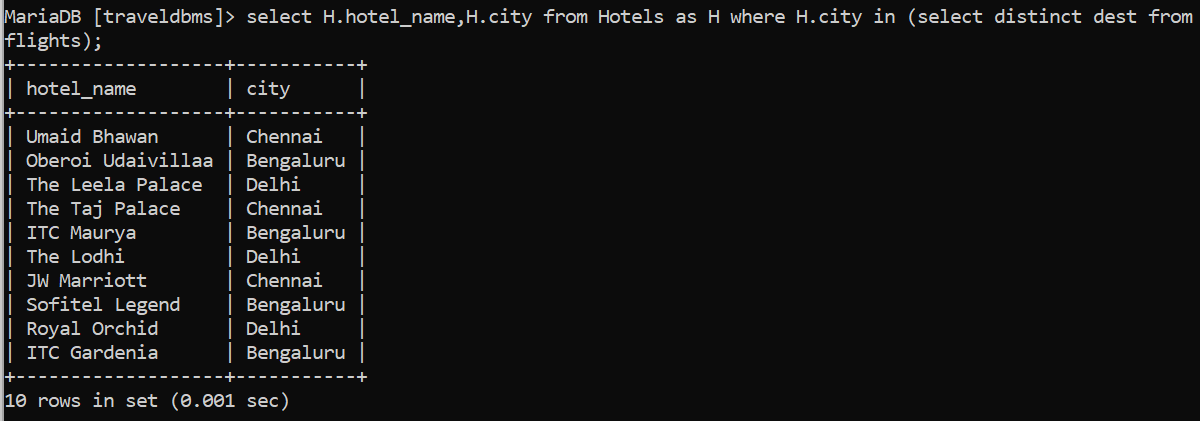
**5)Print the names of users who booked a bus ticket to "Chennai".**

**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");



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

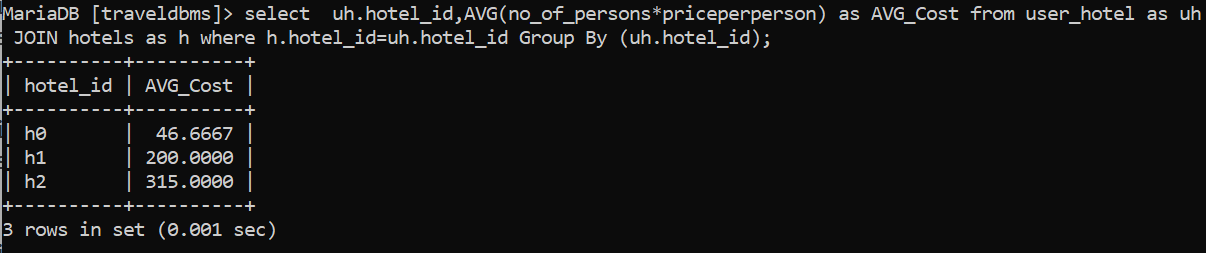
**A)** select H.hotel\_name,H.city from Hotels as H where H.city in (select distinct dest from flights);



**Aggregate Functions**

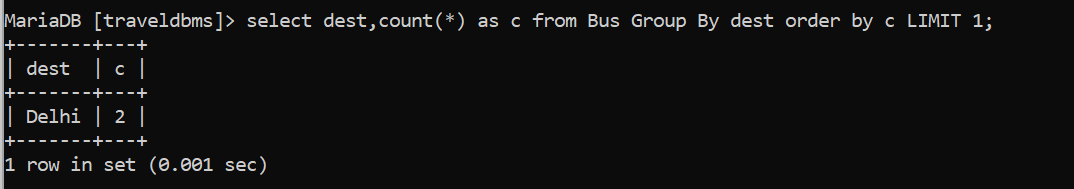
**1) Calculate the average amount spent by an user in each hotel.**

**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);

****

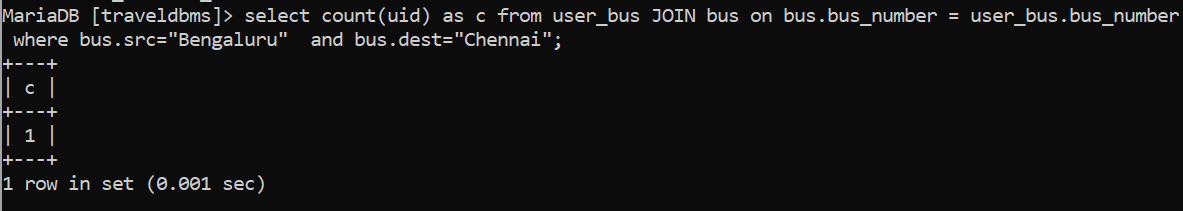
**2)** **Find the places which have minimum incoming buses.**

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

****

**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";

****

**Set Operations**

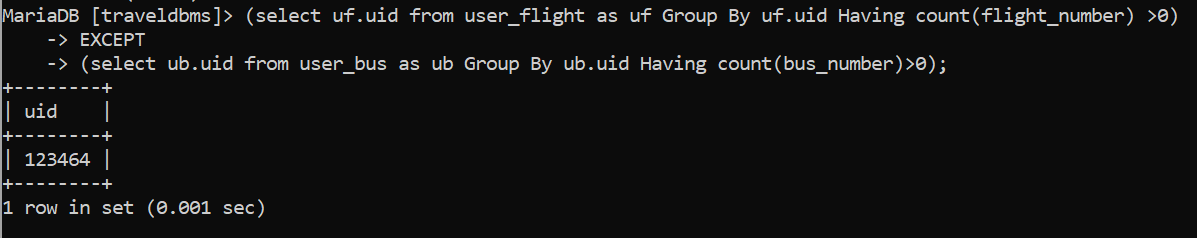
**1)Find the users who always prefer flight.**

**A)**

(select uf.uid from user\_flight as uf Group By uf.uid Having count(flight\_number) >0)

EXCEPT

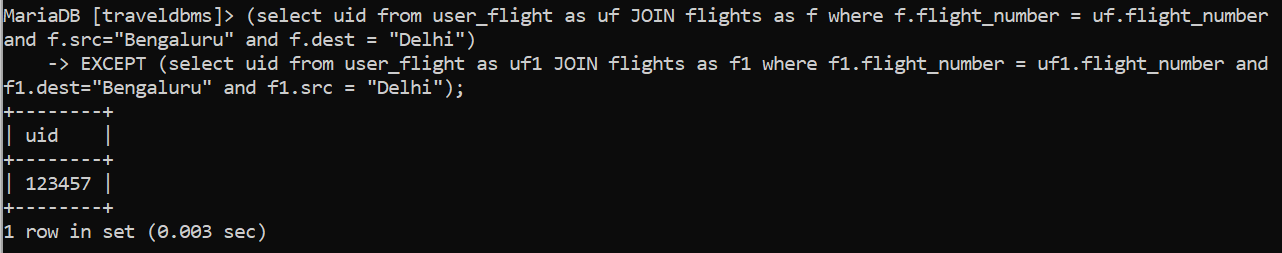
(select ub.uid from user\_bus as ub Group By ub.uid Having count(bus\_number)>0);



**2)** L**ist 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")

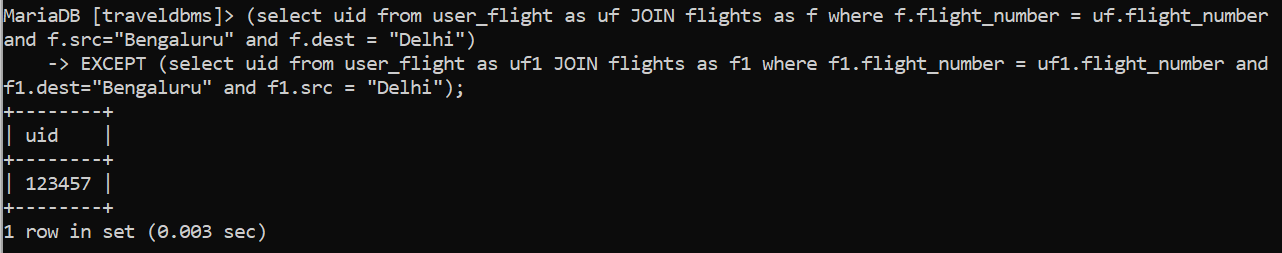
EXCEPT (select uid from user\_flight as uf1 JOIN flights as f1 where f1.flight\_number = uf1.flight\_number and f1.dest="Bengaluru" and f1.src = "Delhi");

****

**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".**

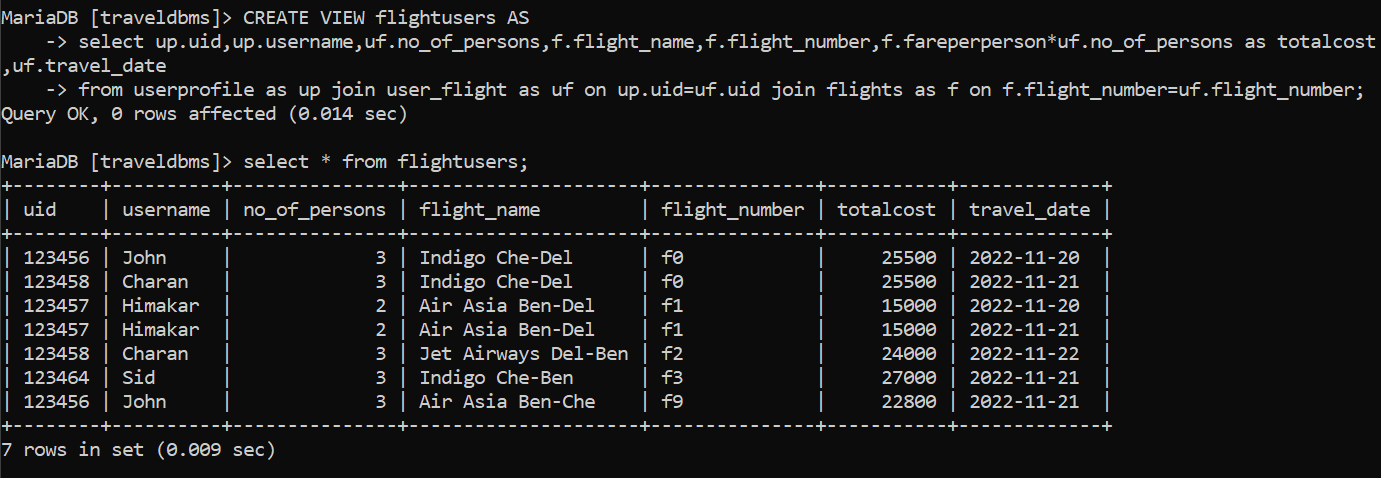
**sq:-** (select uid from user\_flight as uf JOIN flights as f where f.flight\_number = uf.flight\_number and f.src="Bengaluru" and f.dest = "Delhi")

INTERSECT (select uh.uid from user\_hotel as uh where uh.hotel\_id="h0");



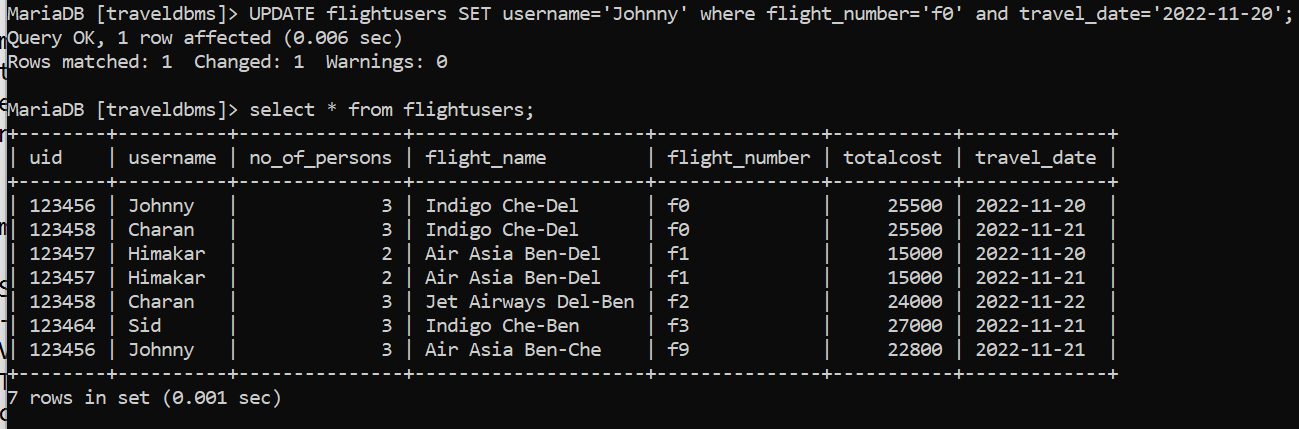
**View**

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

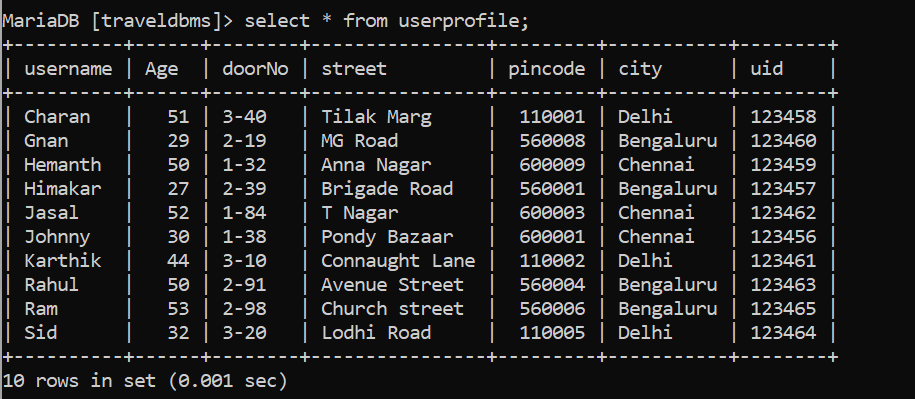
****

**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’;



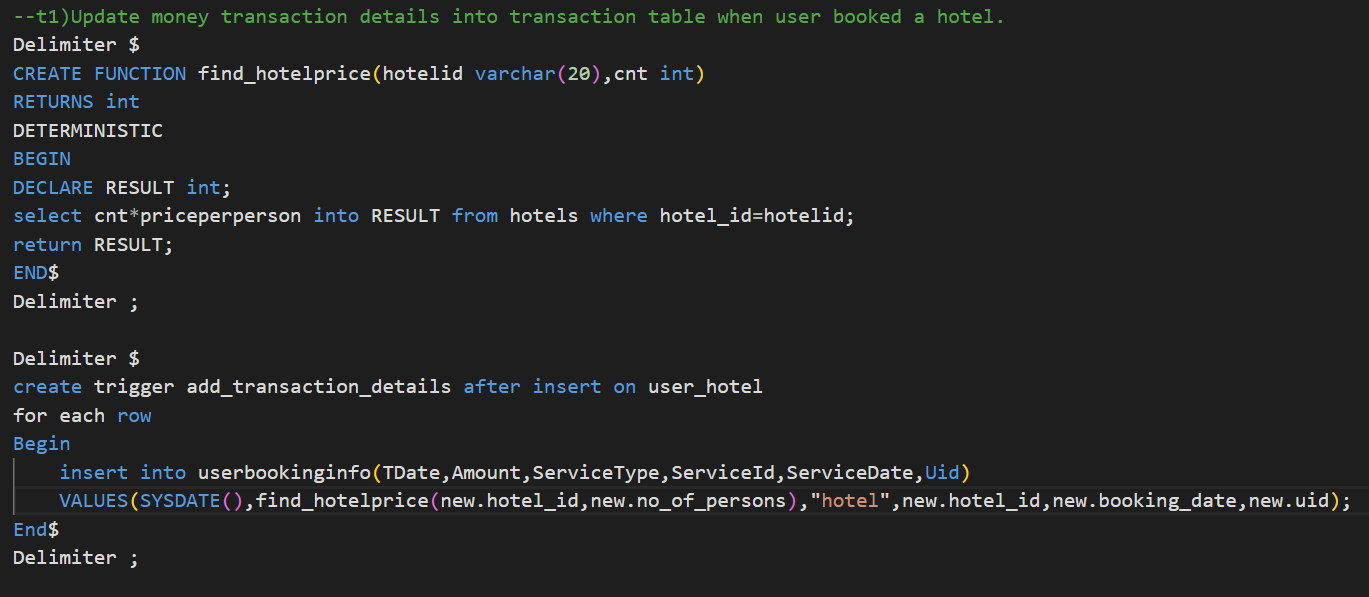
We can also observe that the username is modified in the userprofile’s table.



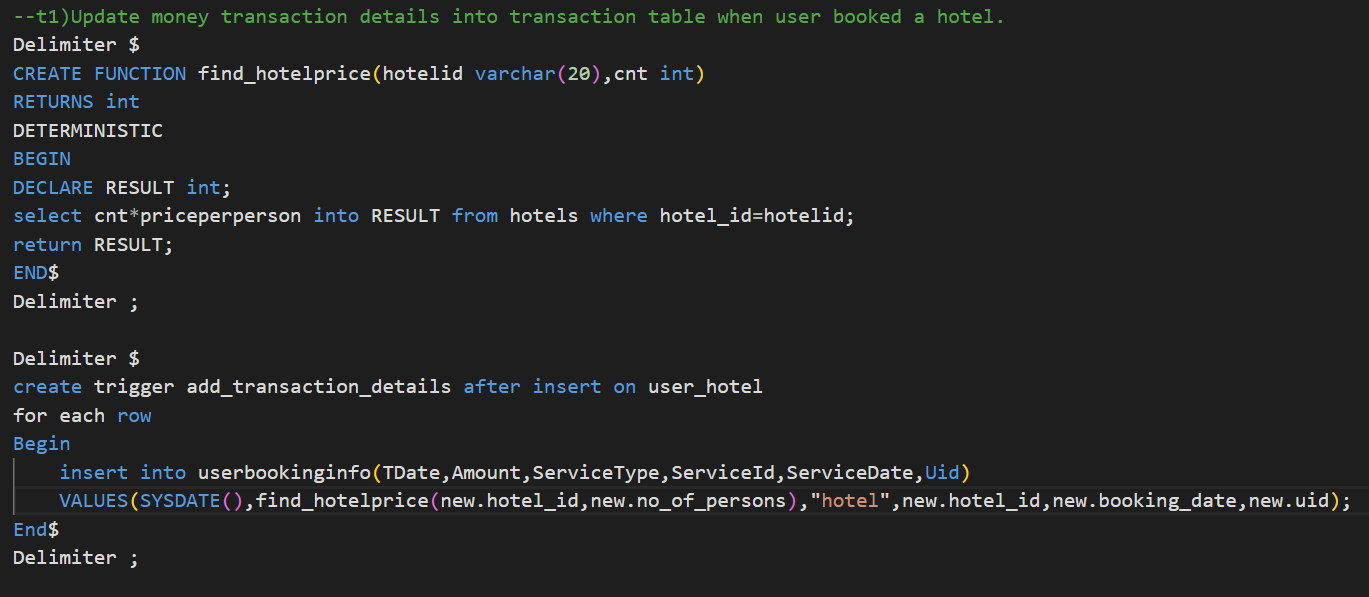
**Triggers (Functions or Procedures)**

**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 .**

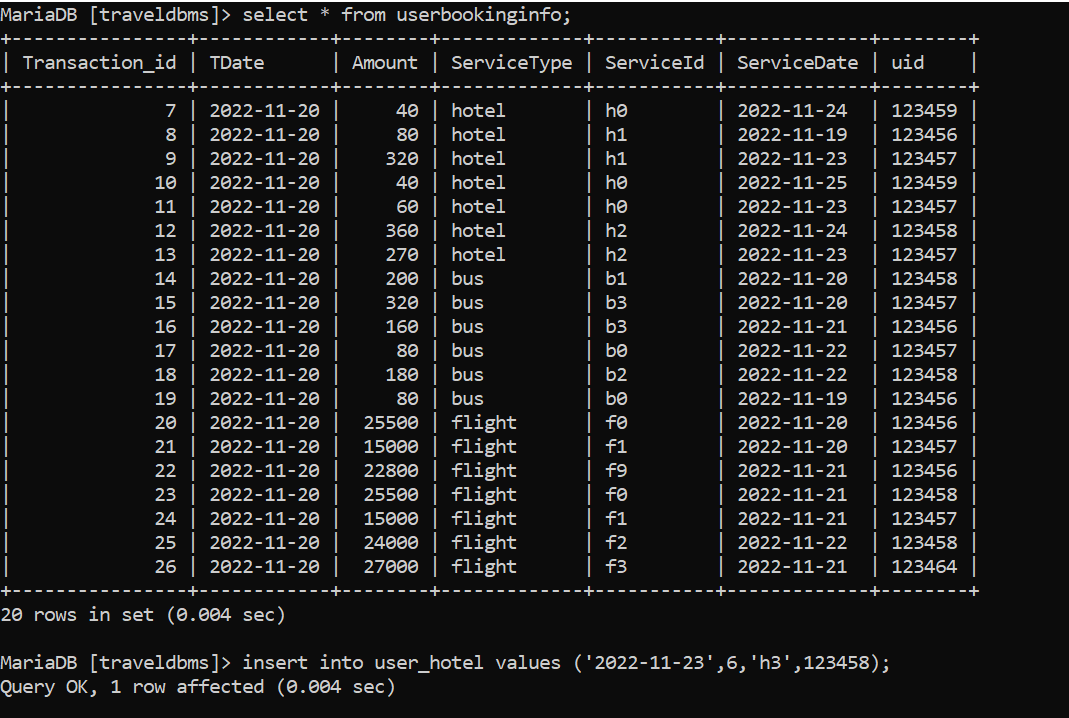


**Trigger to be called when the user booked a hotel and updates the userbookinginfo table with the transaction details.**

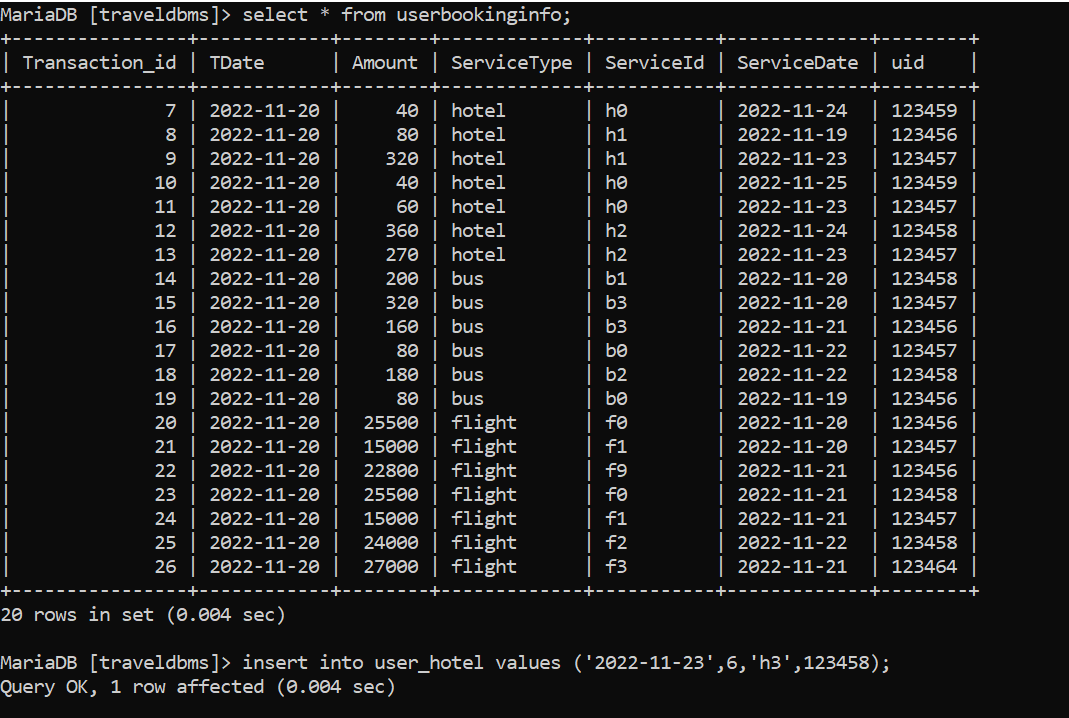


**Results screenshots:**

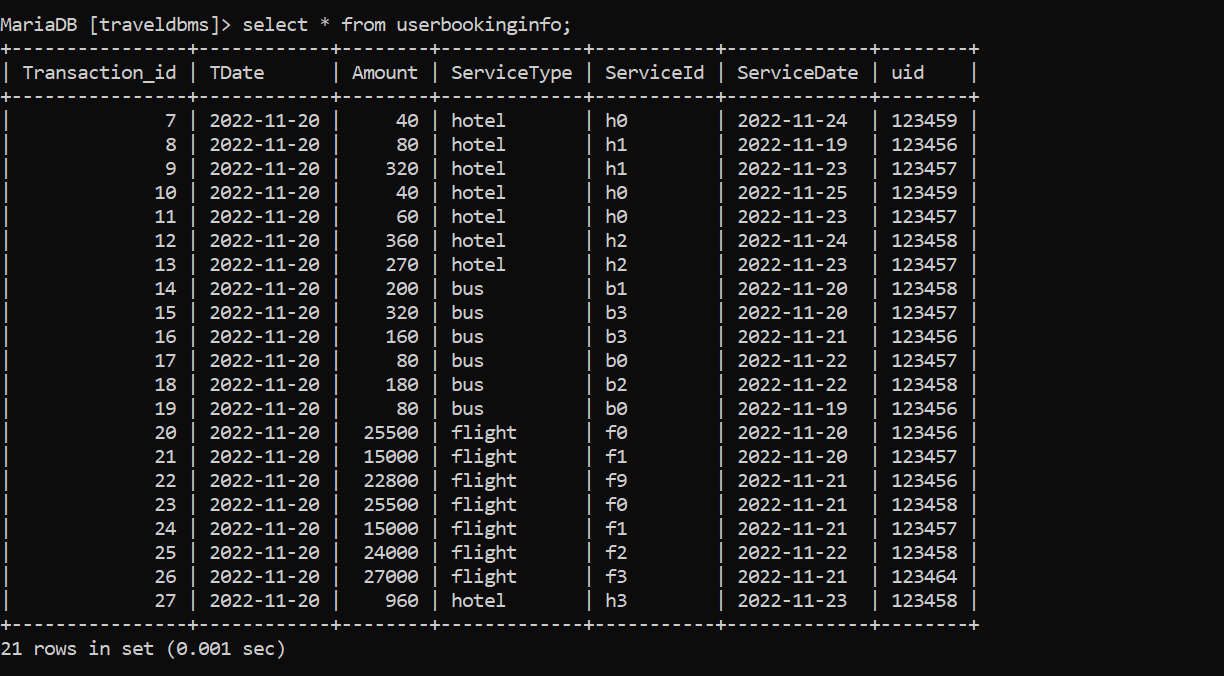
Rows in the userbookinginfo table before:



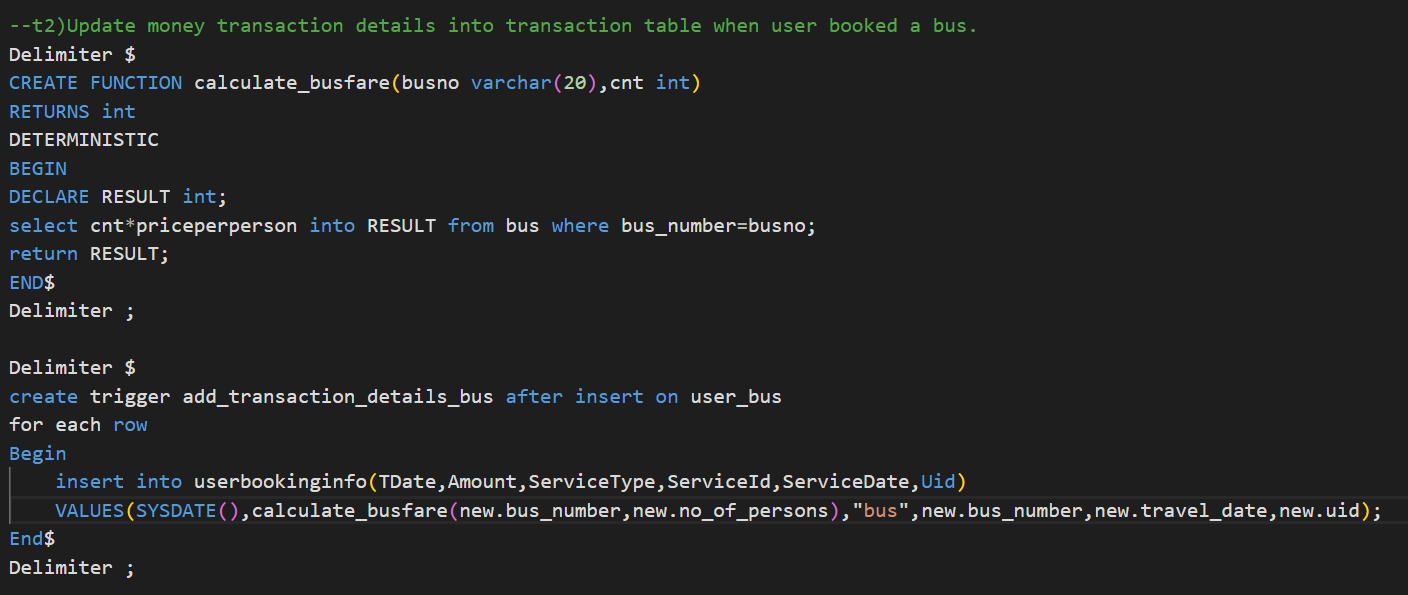
Inserting a row into user\_hotel;



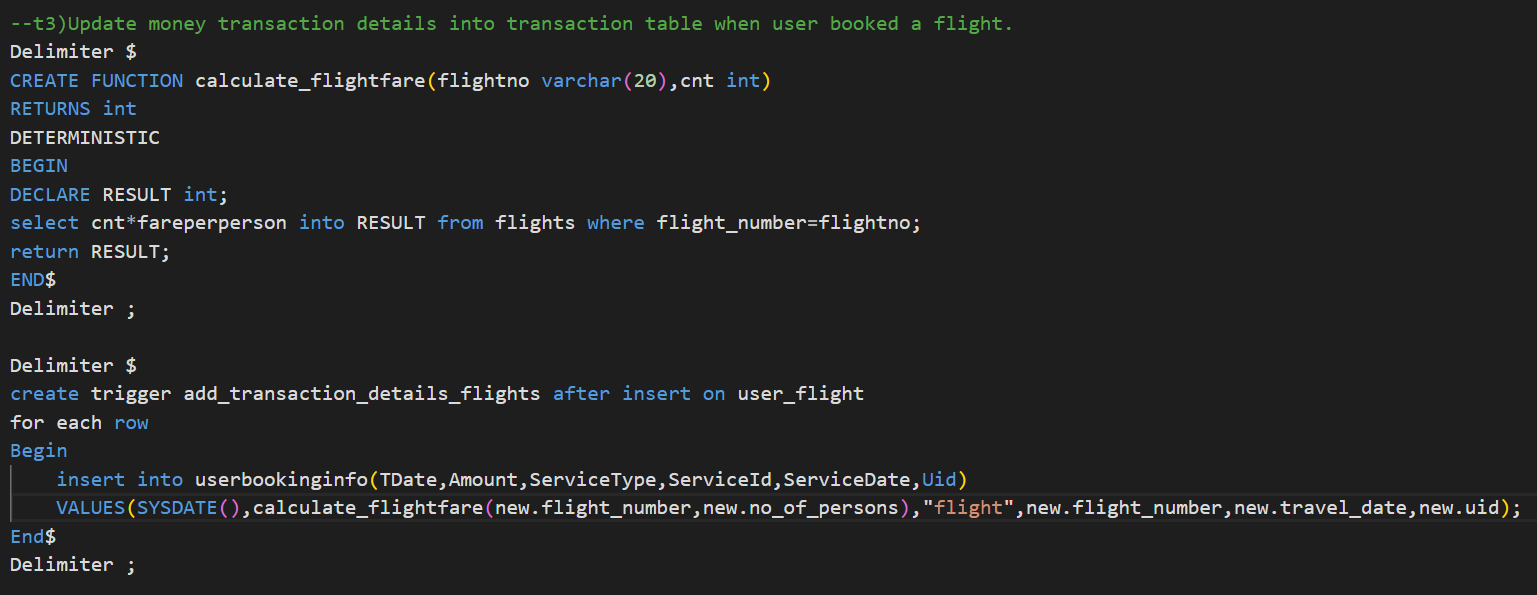
Rows in the userbookinginfo table After:



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

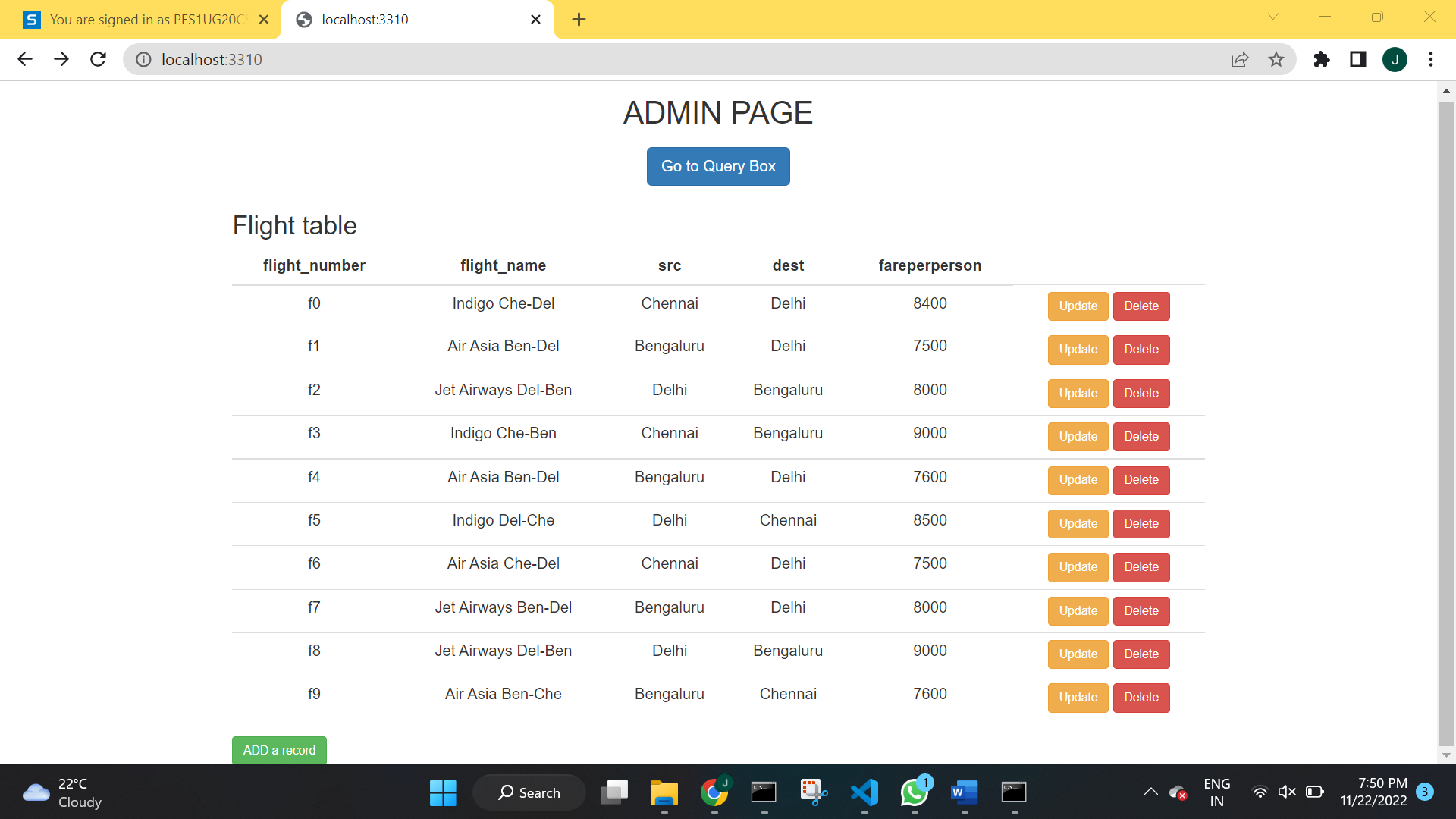


**Similarly the below code is used to update the userbookinginfo table when an user booked a bus.**



## **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.



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

**Adding a record into table with flight number f10:**

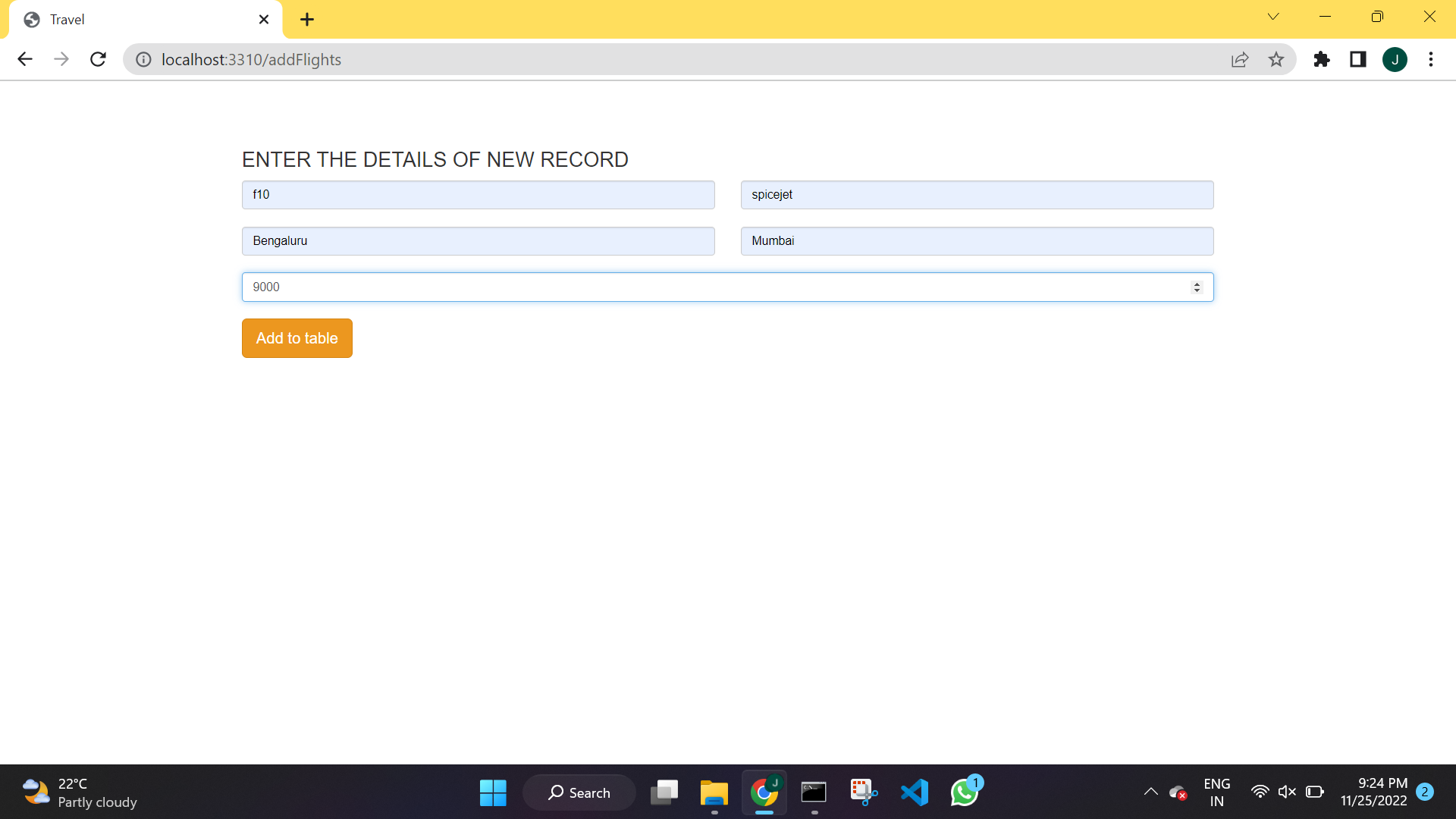
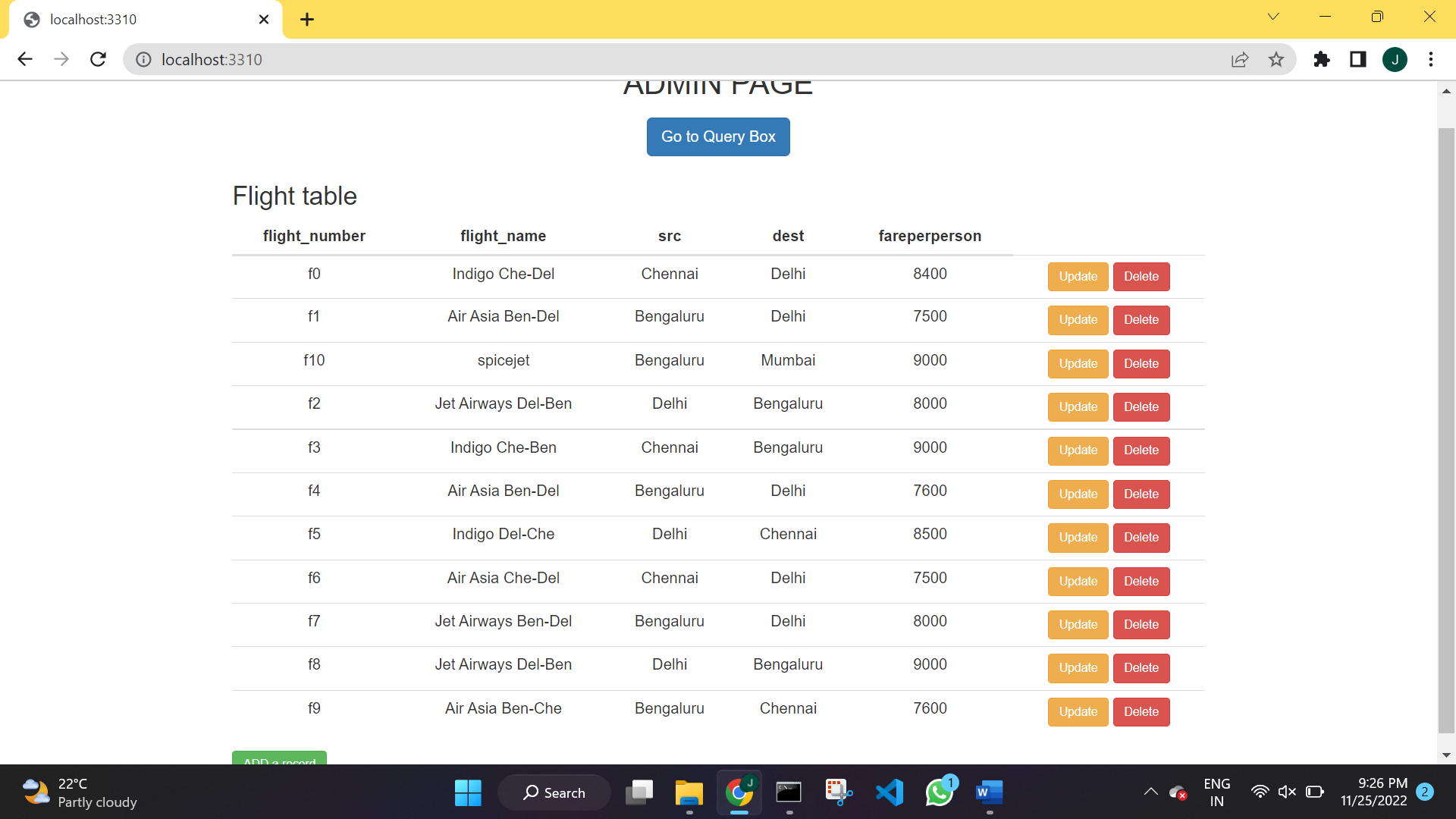
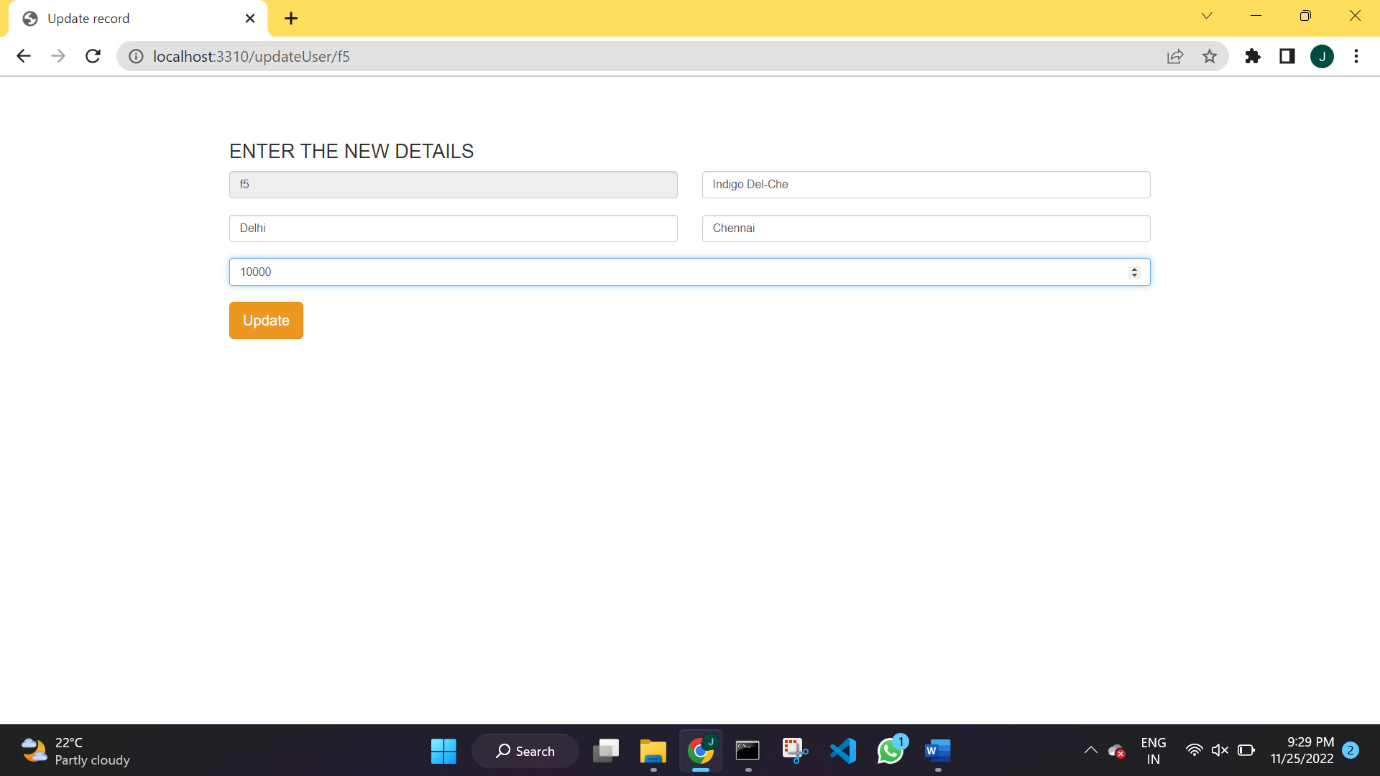


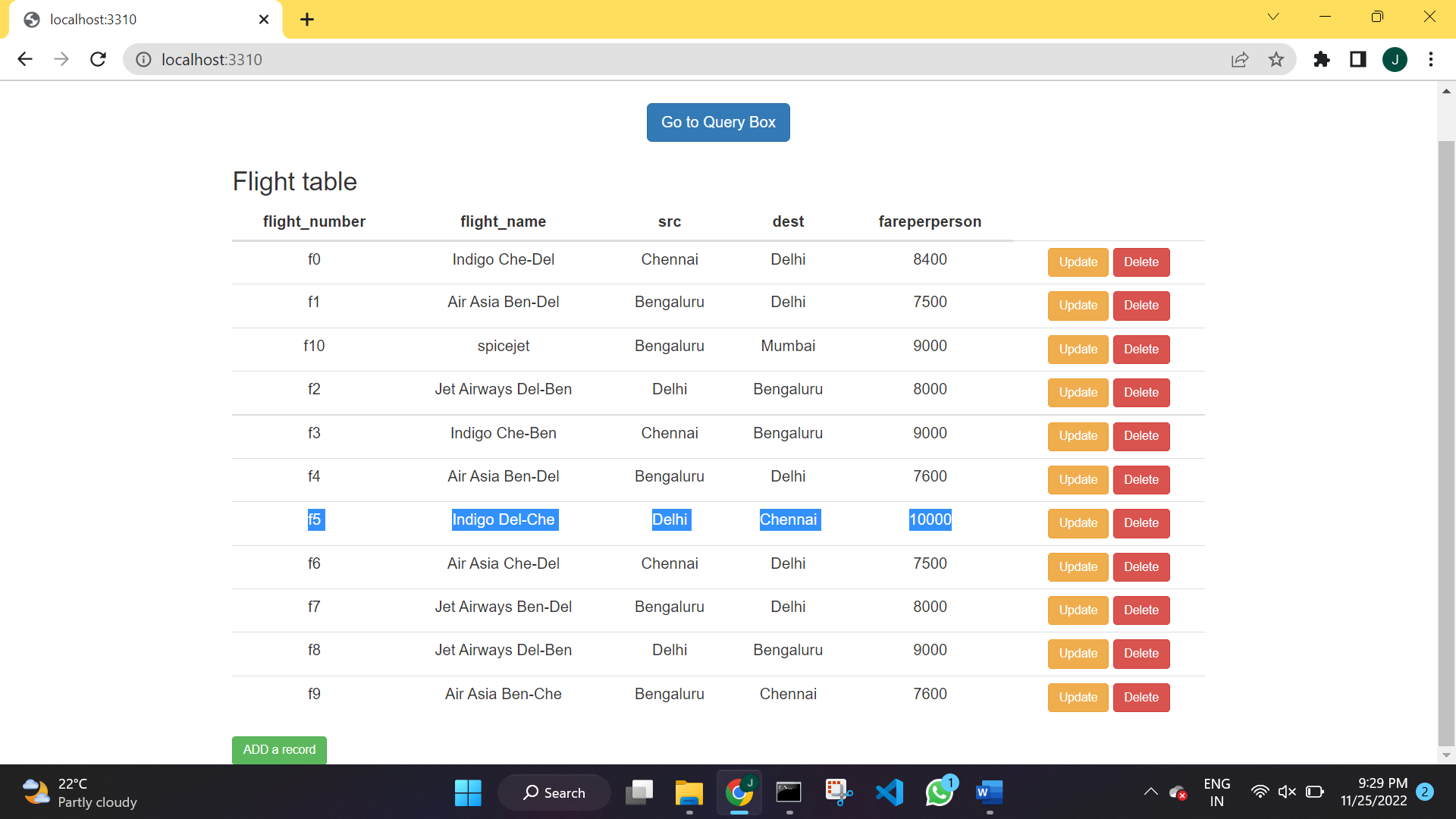
Table after adding a record:



**Updating a record in the table with flight number :**

****

Updated value



**Deleting a record with flight number f10:**

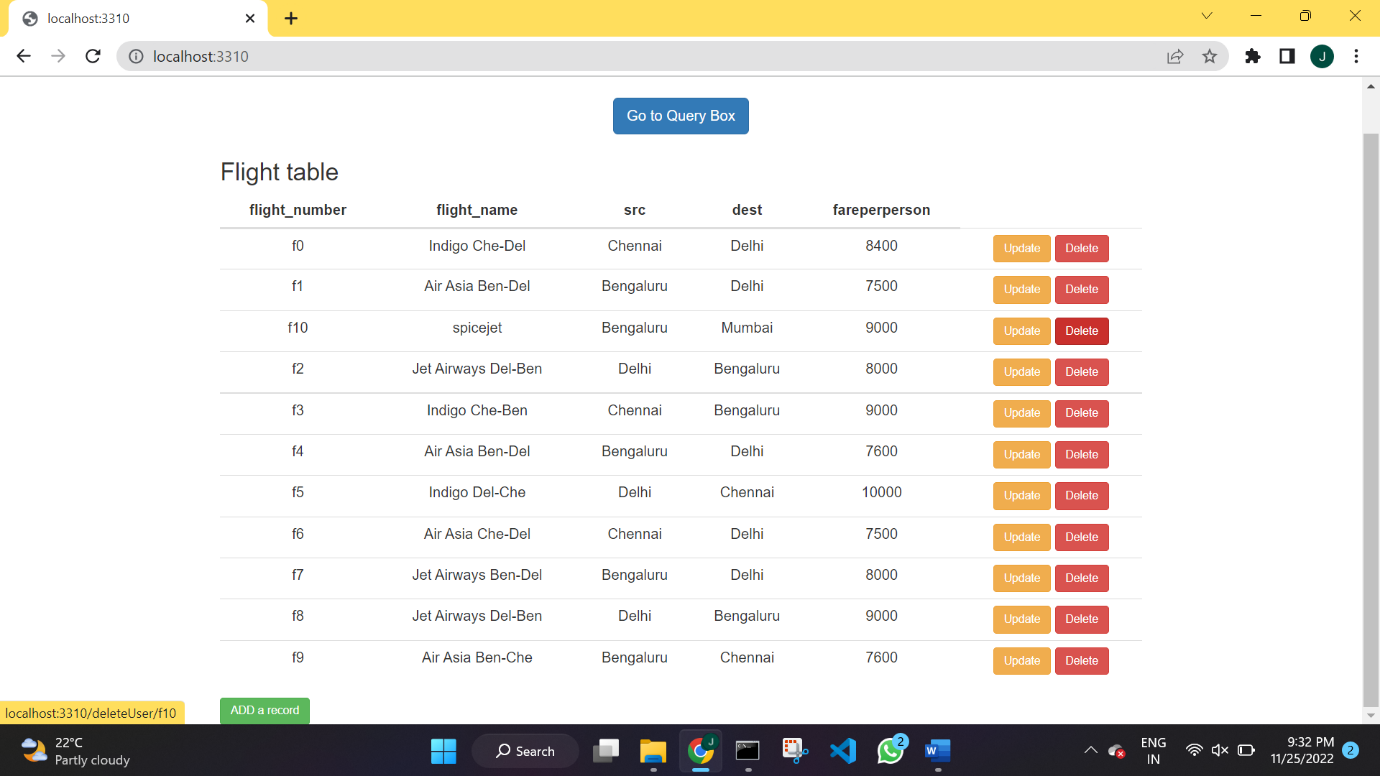
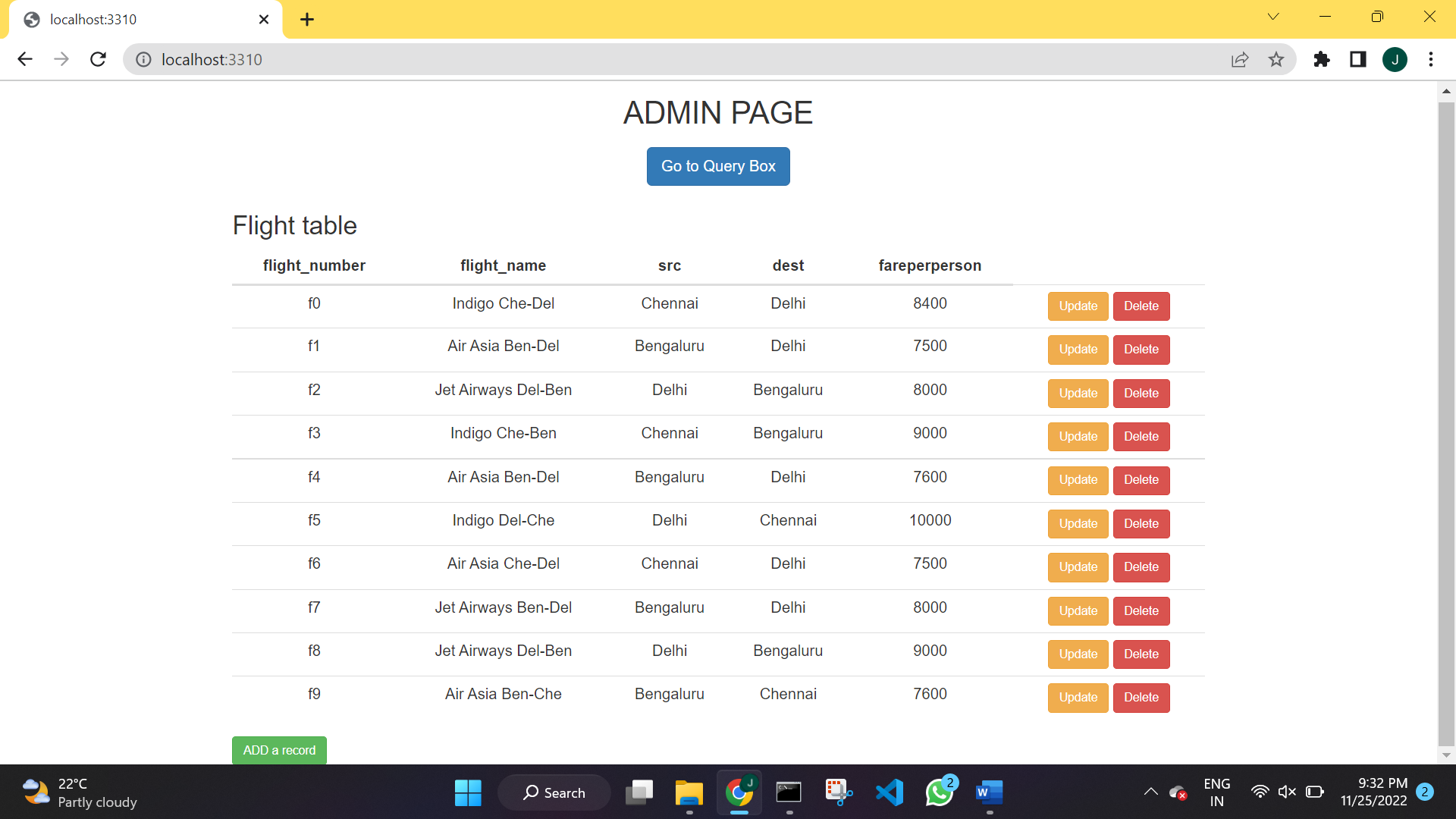
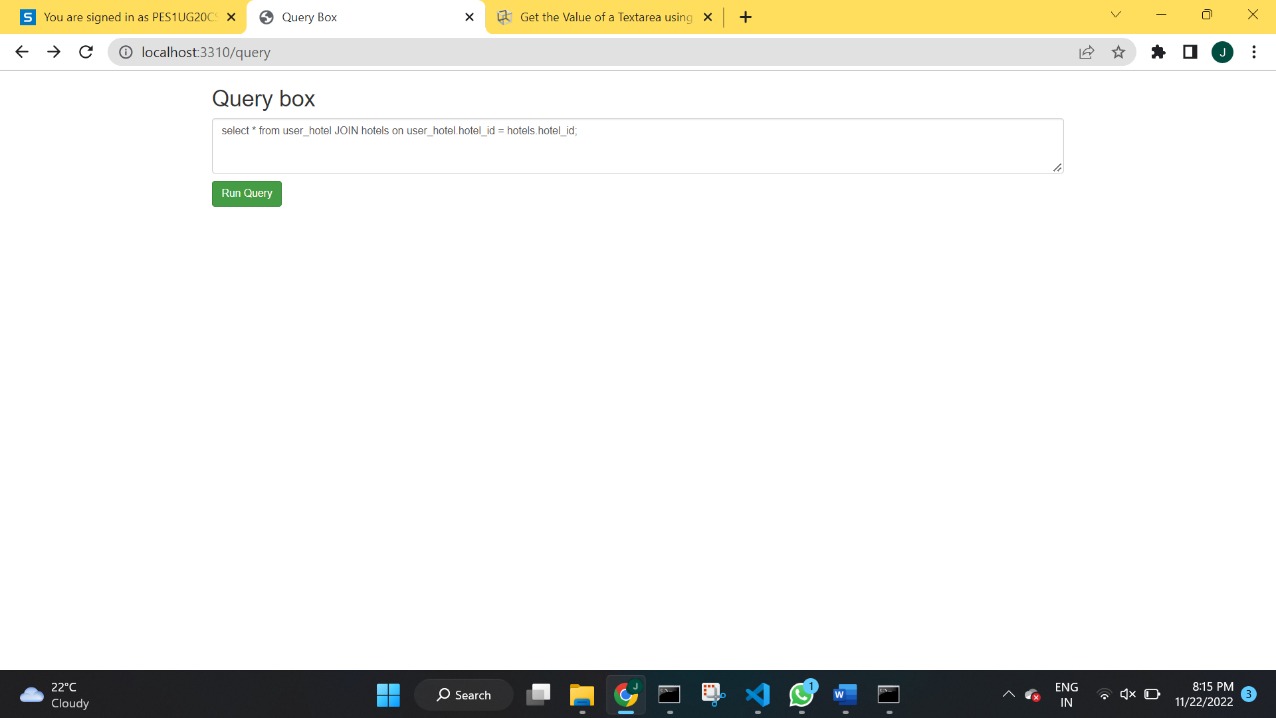
****

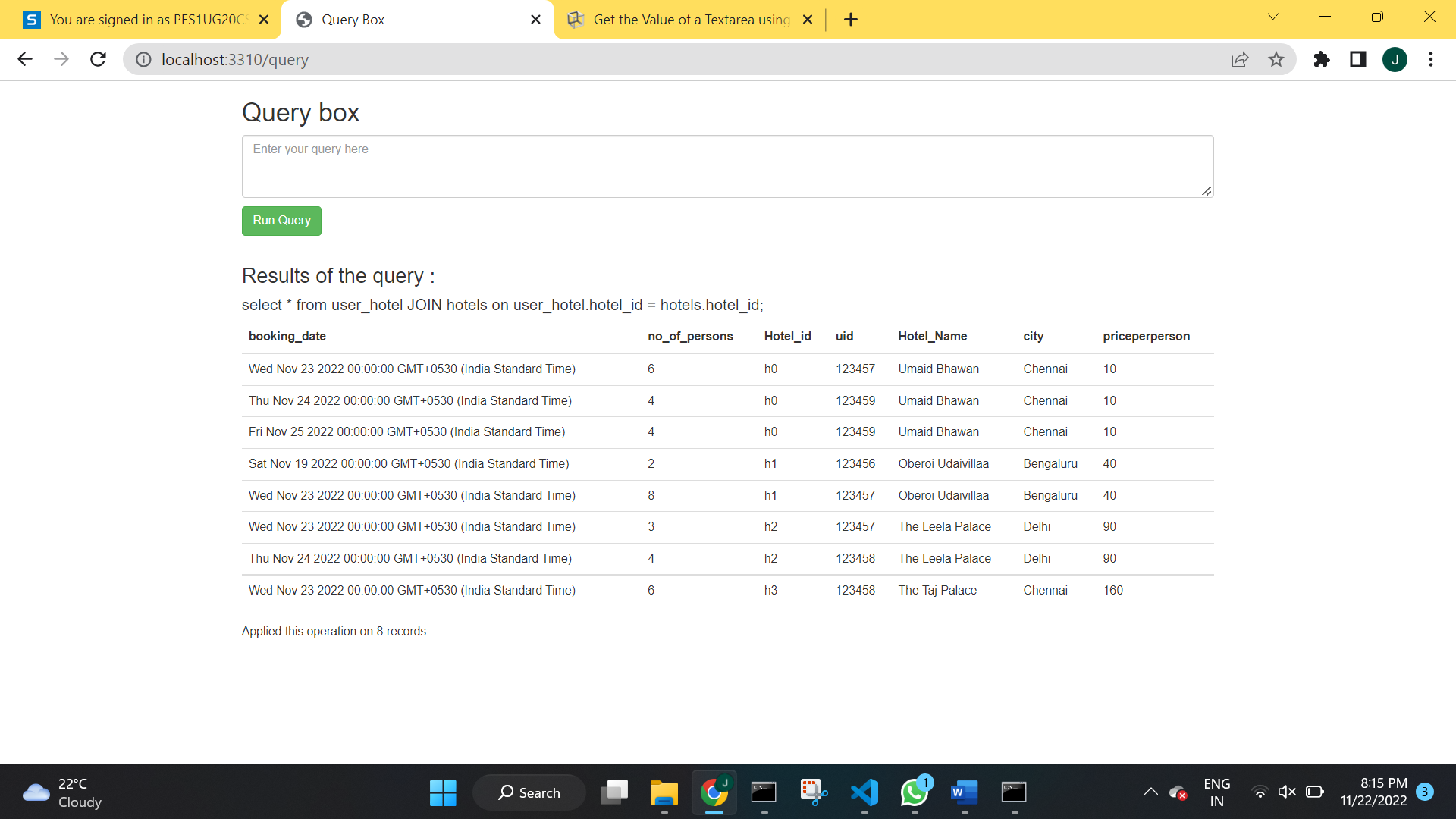
Table after deletion:



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

Query1:

****



**Query2:**

