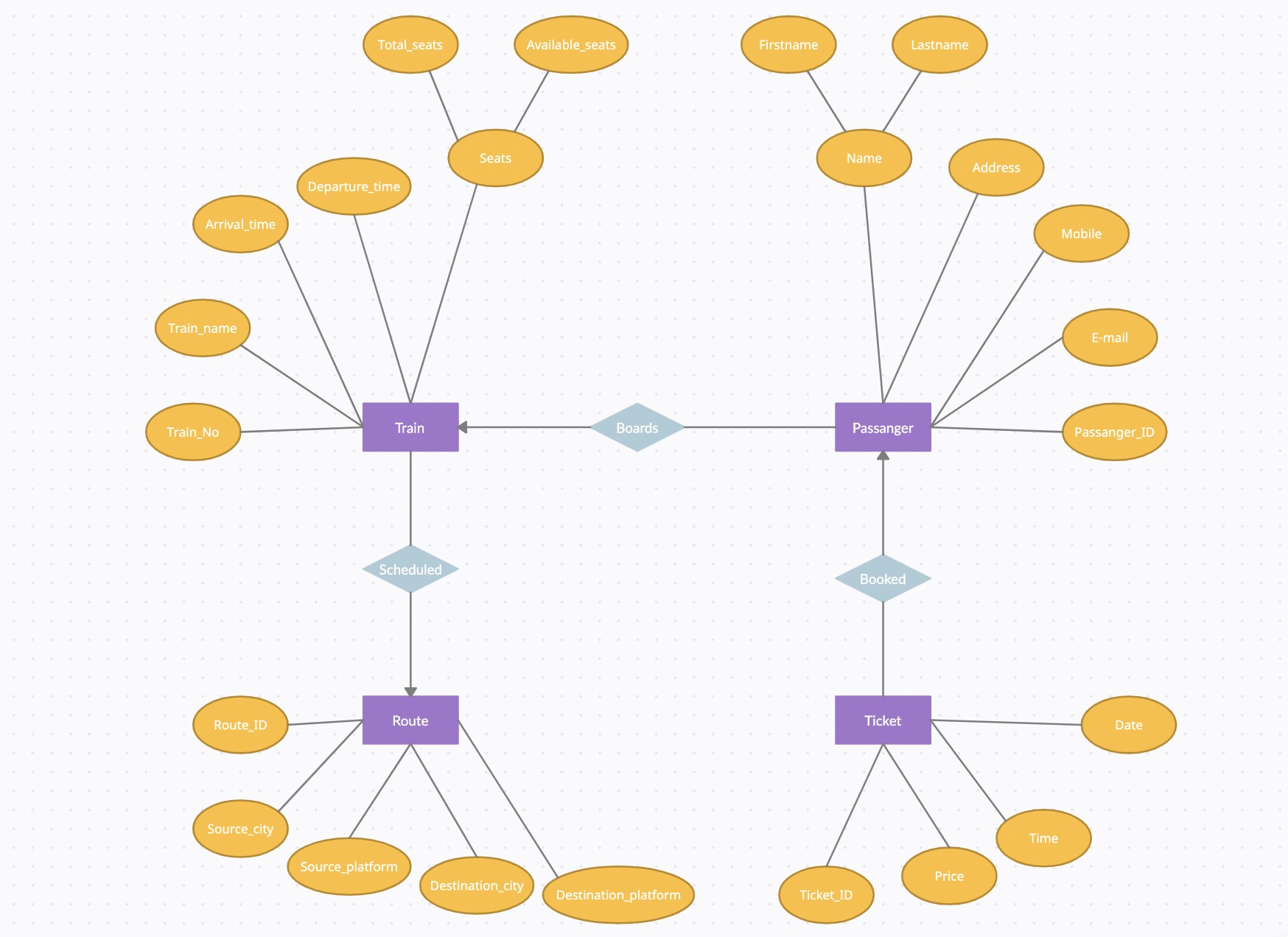
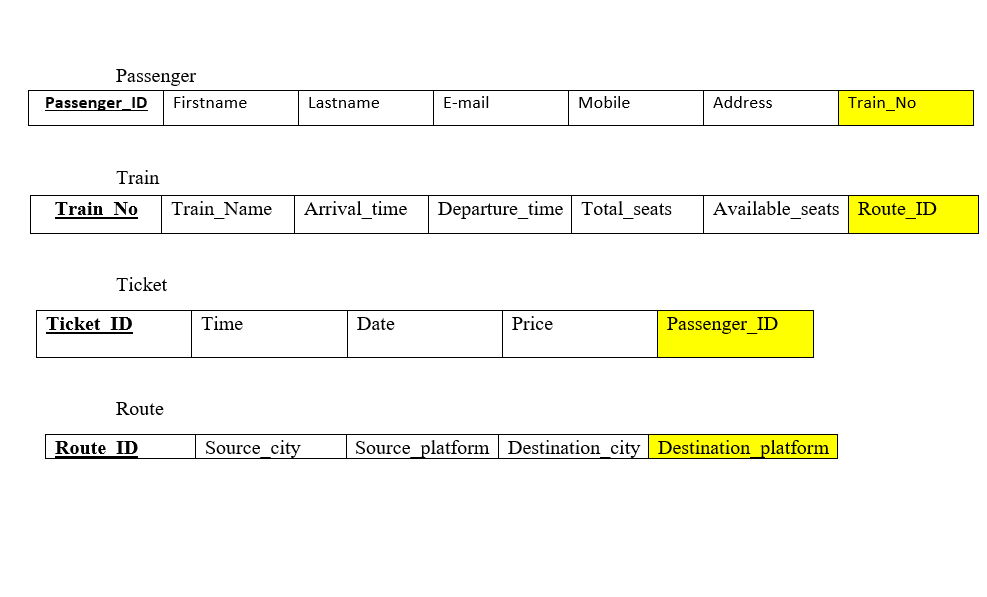
**Railway Management System**

# Problem Statement :A database to keep record of train timings and tracking the routes of trains. Moreover, keeping track of the total number of seats, the number of seats available and the price of each type of ticket while also keeping record of all the passengers and their type of ticket.

**Entity-Relationship Diagram**



# ER Diagram to Table



**Normalized Table**

Ticket Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ticket\_ID | | | Time |  | | Date | | | |  | Price |  | Passenger\_ID | | |
| Passenger Table | | | |  | |  | | | |  | |  | | | |
| Passenger\_ ID | | First  name | |  | Last  name |  | | E-mail |  | Mobile | | Address | | Train\_No | |
| Train Table | | | |  | |  | | | |  | |  | | | |
| Train\_No | Train\_ name | | | Arrival\_ time | |  | Departure\_ time | | |  | Total\_ seats | Available\_ seats | | | Route\_ ID |
| Route Table | | | |  | |  | | | |  | |  | | | |
| Route\_ID | | | Source\_city | | | Source\_ platform | | | |  | Destination\_ city | | Destination\_ platform | | |

**Creating Table**

create table Route(Route\_ID int primary key,

Source\_city varchar(20),

Source\_platform int,

Destination\_city varchar(20), Destination\_platform int);

create table Train(Train\_No int primary key,

Train\_name varchar(20),

Arrival\_time timestamp,

Departure\_time timestamp,

Total\_seats int,

Available\_seats int,

Route\_ID int,

CONSTRAINT FK\_Route\_ID FOREIGN KEY (Route\_ID) REFERENCES Route(Route\_ID));

create table Passenger(Passenger\_ID int primary key,

Firstname varchar(20),

Lastname varchar(20),

Email varchar(30),

Mobile number(10),

Address varchar(50),

Train\_No int,

CONSTRAINT FK\_Train\_No FOREIGN KEY (Train\_No) REFERENCES Train(Train\_No));

create table Ticket(Ticket\_ID int primary key,

Train\_time timestamp,

Train\_date date,

Price number(10),

Passenger\_ID int,

CONSTRAINT FK\_Passenger\_ID FOREIGN KEY (Passenger\_ID)

REFERENCES Passenger(Passenger\_ID));

# Inserting data in the Table

INSERT INTO Route VALUES(1, 'Delhi', 3, 'Patiala', 5);

INSERT INTO Route VALUES(2, 'Hyderabad', 7, 'Vishakapatnam', 1);

INSERT INTO Route VALUES(3, 'Ahmedabad', 1, 'Bhopal', 2);

INSERT INTO Route VALUES(4, 'Delhi', 5, 'Amritsar', 3);

INSERT INTO Route VALUES(5, 'Chennai', 2, 'Bengluru', 4);

Select \* from Route;

INSERT INTO Train VALUES(14507, DLI FKA EXP ','6-Jun-2022 01:05:00pm','6-Jun-2022

06:19:00pm',700,543,1);

INSERT INTO Train VALUES(12704,'Falaknuma Exp','12-May-2022 03:55:00pm','13-Jan2022 04:40:00am',1000,743,2);

INSERT INTO Train VALUES(19413,  ADI KOLKATA EXP ,'21-May-2022 09:05:00pm','22-May-2022 08:25:00am',500,197,3);

INSERT INTO Train VALUES(12497,'Shane Punjab','7-May-2022 06:40:00am','7-May-2022

02:15:00am',800,554,4);

INSERT INTO Train VALUES(12007,'MYS Shatabdi','27-May-2022 06:00:00am','27-May-2022 10:45:00am',400,154,5);

Select \* from Train;

INSERT INTO Passenger VALUES(101, 'Ananya, 'Nathi', 'ananya11@gmail.com', 9345655123, 'Kukatpally, Hyderabad', 14507);

INSERT INTO Passenger VALUES(145, 'Srishti', 'Rawat', 'srishtir@gmail.com', 9347367890, 'Kashipur, Uttrakand', 19413);

INSERT INTO Passenger VALUES(332, 'Neeraj', 'Garg', 'ngarg@gmail.com', 7128311223,

'Ashok Vihar, Punjab', 12704);

INSERT INTO Passenger VALUES(201, 'Hiten', 'Narang', 'hitenn1@gmail.com', 9876543421,

'Bhadson road, Patiala', 12007);

INSERT INTO Passenger VALUES(443, 'Nishanth', 'Verma', 'nishanth13@gmail.com', 7865234591,'Chennai', 12497);

Select \* from Passenger;

INSERT INTO Ticket VALUES(3519545764,'25-April-2022 03:20:00pm','25-April-2022',1000, 145);

INSERT INTO Ticket VALUES(2347812334,'10-July-2022 08:30:00pm','10-July-2022',1500, 101);

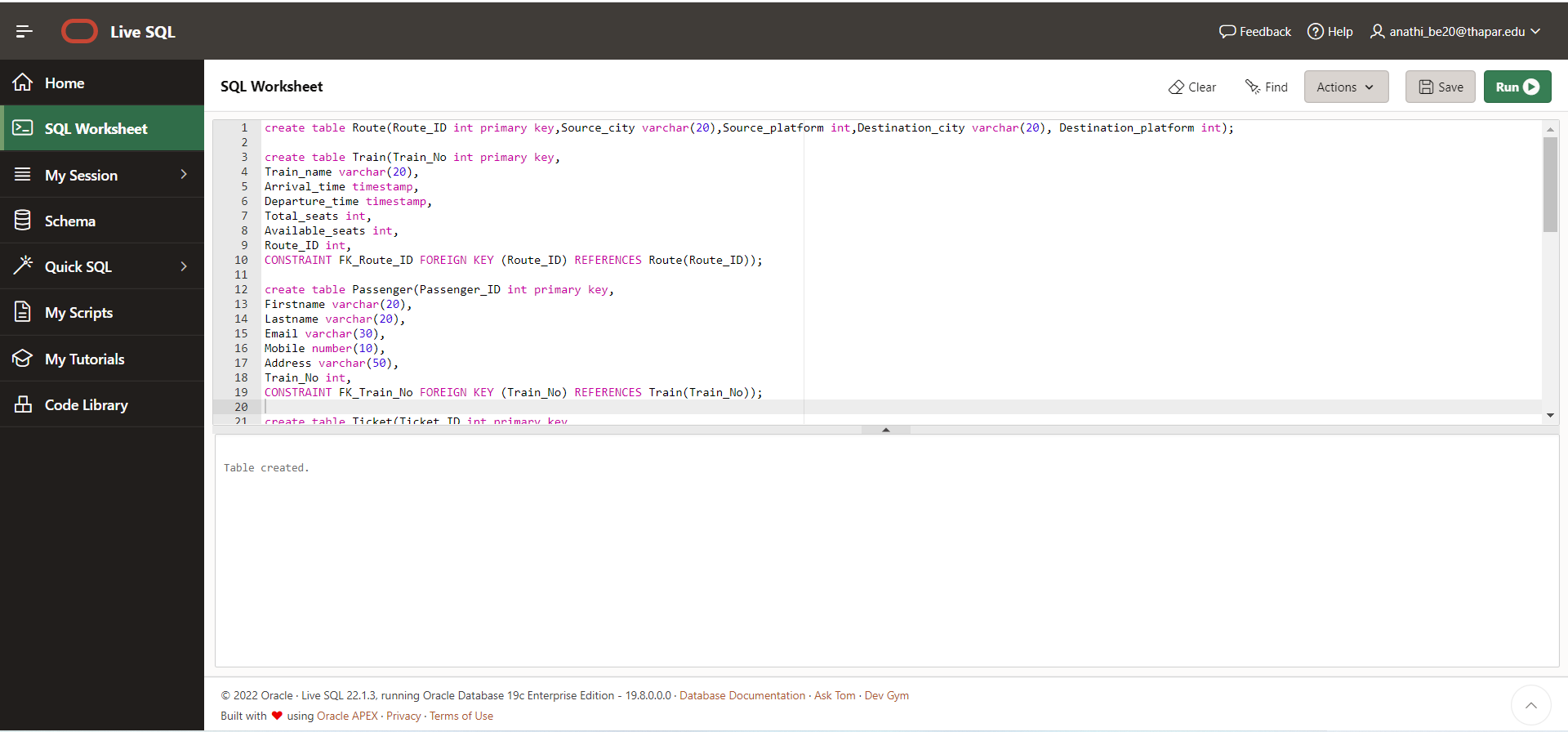
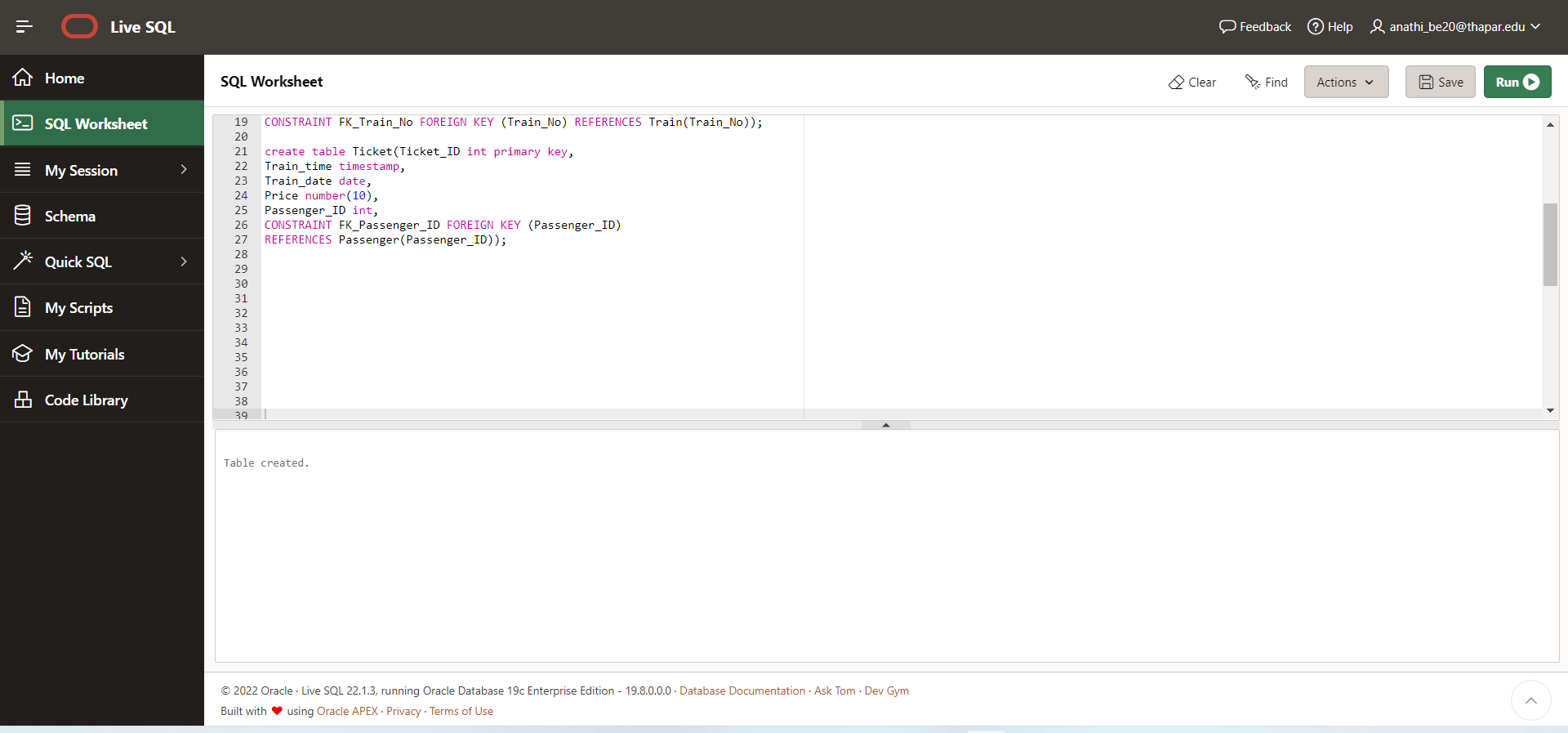
INSERT INTO Ticket VALUES(1235678321,'19-Jun-2022 10:50:00pm','19-Jun-2022',1150, 201);

INSERT INTO Ticket VALUES(156712901,'22-May-2022 11:40:30pm','22-May-2022',550, 443);

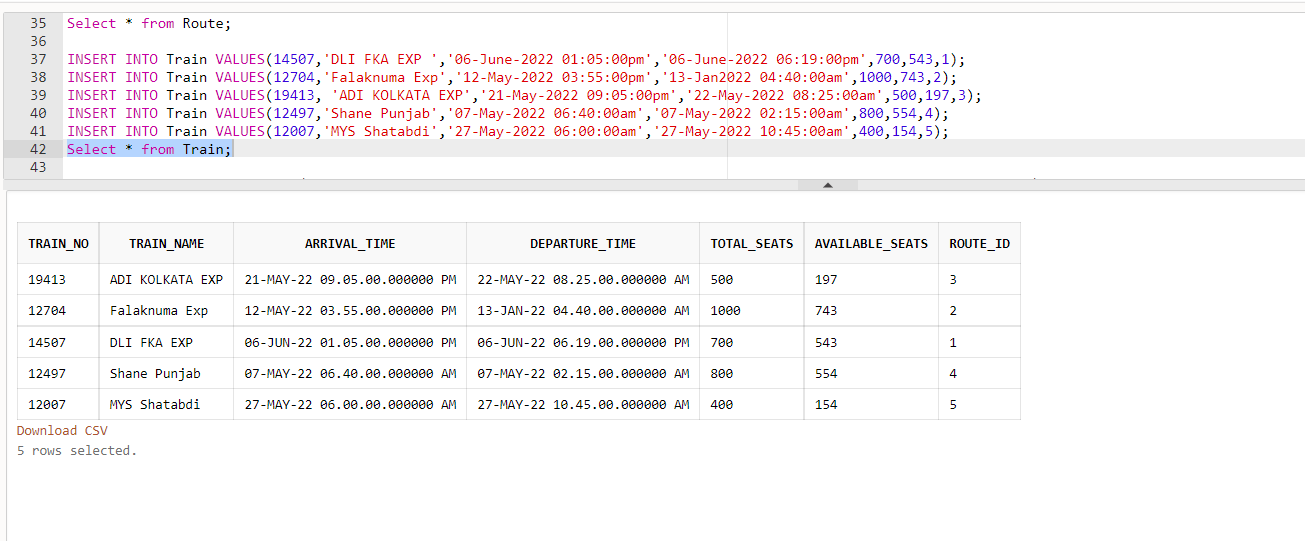
INSERT INTO Ticket VALUES(1421621421,'05-Jan-2022 09:35:00am','18-Jan-2022',900, 332);

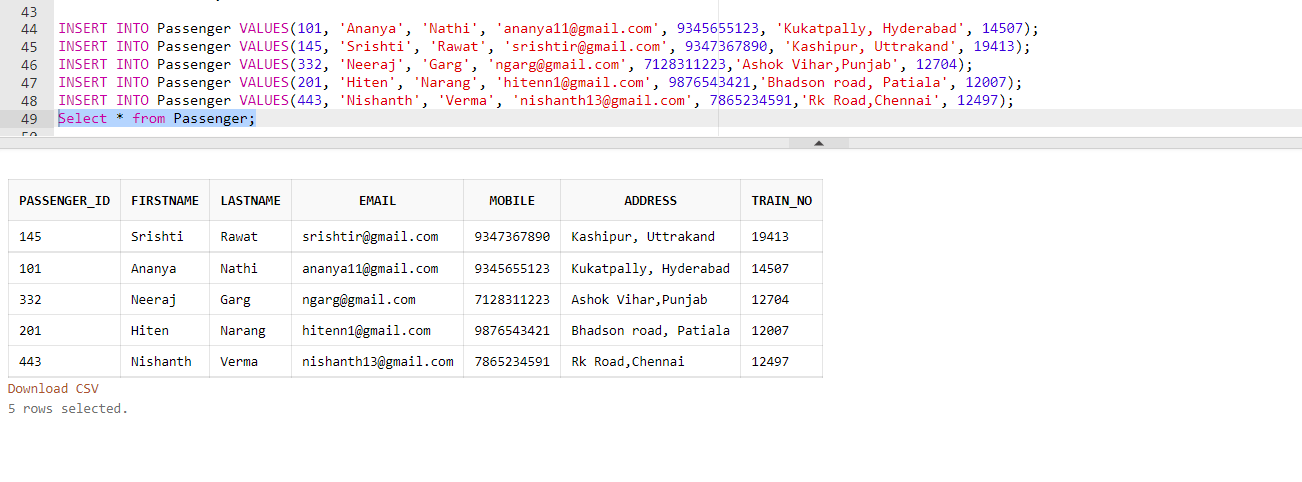
Select \* from Ticket;

**SQL Output Screenshots**







**PL/SQL Codes and Screenshots**

## 1. To display details of all passengers

Declare

P\_Firstname Passenger.Firstname%type;

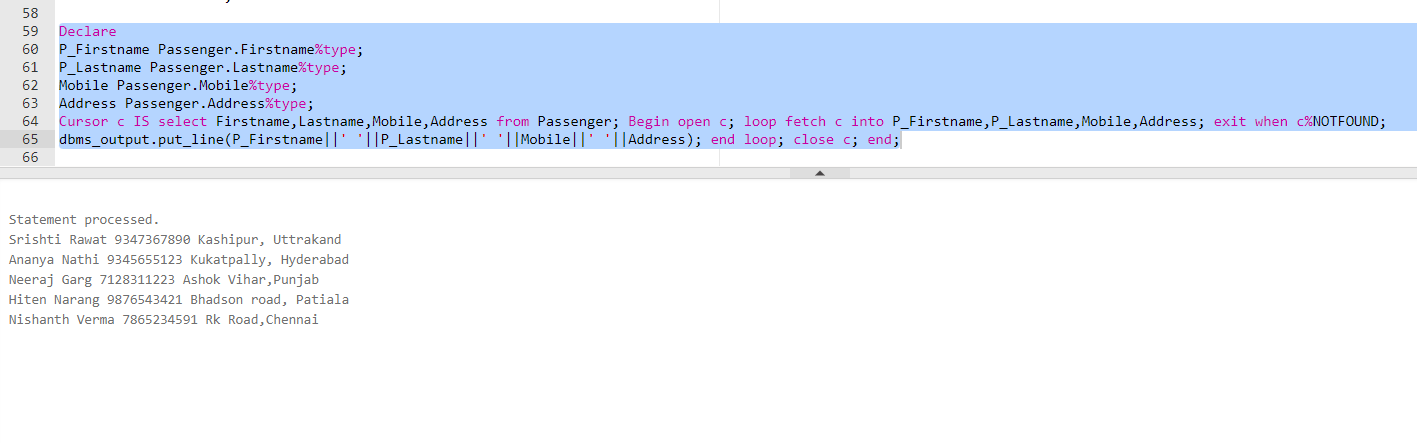
P\_Lastname Passenger.Lastname%type;

Mobile Passenger.Mobile%type;

Address Passenger.Address%type;

Cursor c IS select Firstname,Lastname,Mobile,Address from Passenger; Begin open c; loop fetch c into P\_Firstname,P\_Lastname,Mobile,Address; exit when c%NOTFOUND;

dbms\_output.put\_line(P\_Firstname||' '||P\_Lastname||' '||Mobile||' '||Address); end loop; close c; end;



## 2. To display the details of passenger whose has taken highest fair ticket

declare

P\_Firstname Passenger.Firstname%type;

P\_Lastname Passenger.Lastname%type;

Mobile Passenger.Mobile%type;

Address Passenger.Address%type;

CustomerId Passenger.Passenger\_ID%type;

Cursor c IS select P.Firstname,P.Lastname,P.Mobile,P.Address,T.Passenger\_ID from

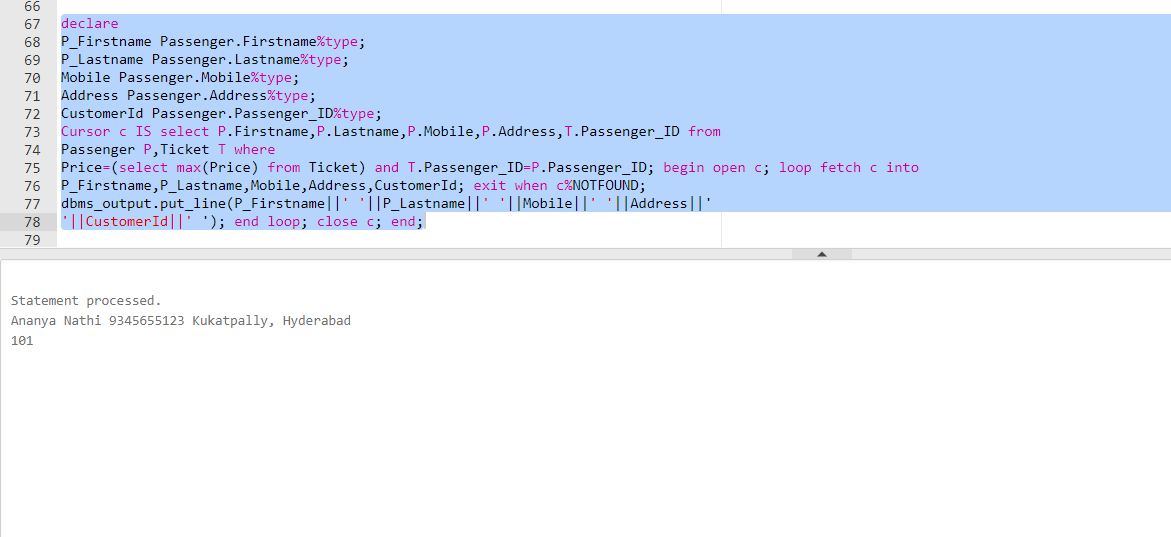
Passenger P,Ticket T where

Price=(select max(Price) from Ticket) and T.Passenger\_ID=P.Passenger\_ID; begin open c; loop fetch c into

P\_Firstname,P\_Lastname,Mobile,Address,CustomerId; exit when c%NOTFOUND;

dbms\_output.put\_line(P\_Firstname||' '||P\_Lastname||' '||Mobile||' '||Address||'

'||CustomerId||' '); end loop; close c; end;



## 3.To display the details of passenger whose train is in evening declare

declare

P\_Firstname Passenger.Firstname%type;

P\_Lastname Passenger.Lastname%type;

Mobile Passenger.Mobile%type;

Address Passenger.Address%type;

Train\_No Passenger.Train\_No%type;

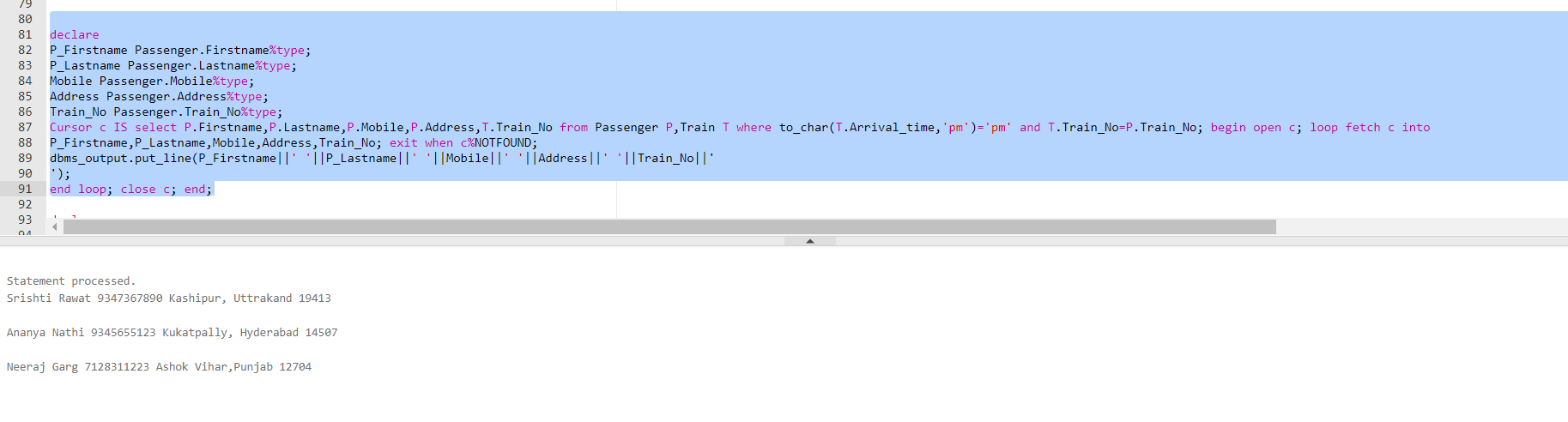
Cursor c IS select P.Firstname,P.Lastname,P.Mobile,P.Address,T.Train\_No from Passenger P,Train T where to\_char(T.Arrival\_time,'pm')='pm' and T.Train\_No=P.Train\_No; begin open c; loop fetch c into

P\_Firstname,P\_Lastname,Mobile,Address,Train\_No; exit when c%NOTFOUND;

dbms\_output.put\_line(P\_Firstname||' '||P\_Lastname||' '||Mobile||' '||Address||' '||Train\_No||'

');

end loop; close c; end;



## 4. Exception handling

declare

P\_Email Passenger.Email%type; P\_Passenger\_ID Passenger.Passenger\_ID%type; e\_invalid\_ID EXCEPTION; begin update Passenger set Email = 'random@gmail.com' where P\_Passenger\_ID = 2; if SQL%NOTFOUND then

raise e\_invalid\_ID; End if; commit;

EXCEPTION

When no\_data\_found then dbms\_output.put\_line('No such record'); When too\_many\_rown then dbms\_output.put\_line('More than one record found'); When e\_invalid\_ID then dbms\_output.put\_line('No such passenger exists'); When others then dbms\_output.put\_line('Some error found'); end;

