RAILWAY RESERVATION SYSTEM

MINI PROJECT

NAME – Neha Sharma

UID - 20BCS4576

20IOT1-A

DATABASE MANAGEMENT SYSTEM LAB

# PROJECT DESCRIPTION:

This project is about creating the database about Railway Reservation System.

The railway reservation system facilitates the passengers to enquire about the trains available on the basis of source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

The record of train includes its number, name, source, destination, and days on which it is available, whereas record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked.

Passengers can book their tickets for the train in which seats are available. For this, passenger has to provide the desired train number and the date for which ticket is to be booked. Before booking a ticket for a passenger, the validity of train number and booking date is checked.

Once the train number and booking date are validated, it is checked whether the seat is available. If yes, the ticket is booked with confirm status and corresponding ticket ID is generated which is stored along with other details of the passenger. The ticket once booked can be cancelled at any time. For this, the passenger has to provide the ticket ID (the unique key). The ticket ID is searched and the corresponding record is deleted. With this, the first. ticket with waiting status also gets confirmed.

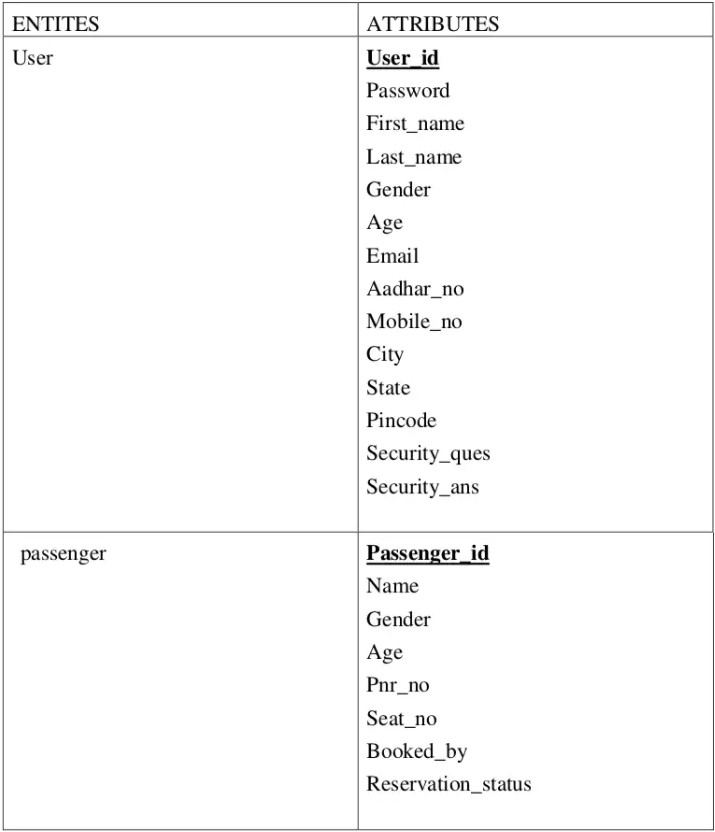
List of Assumption Since the reservation system is very large in reality, it is not feasible to develop the case study to that extent and prepare documentation at the working of the reservation system. To implement this sample case study, some assumptions have been made, which are as follows:

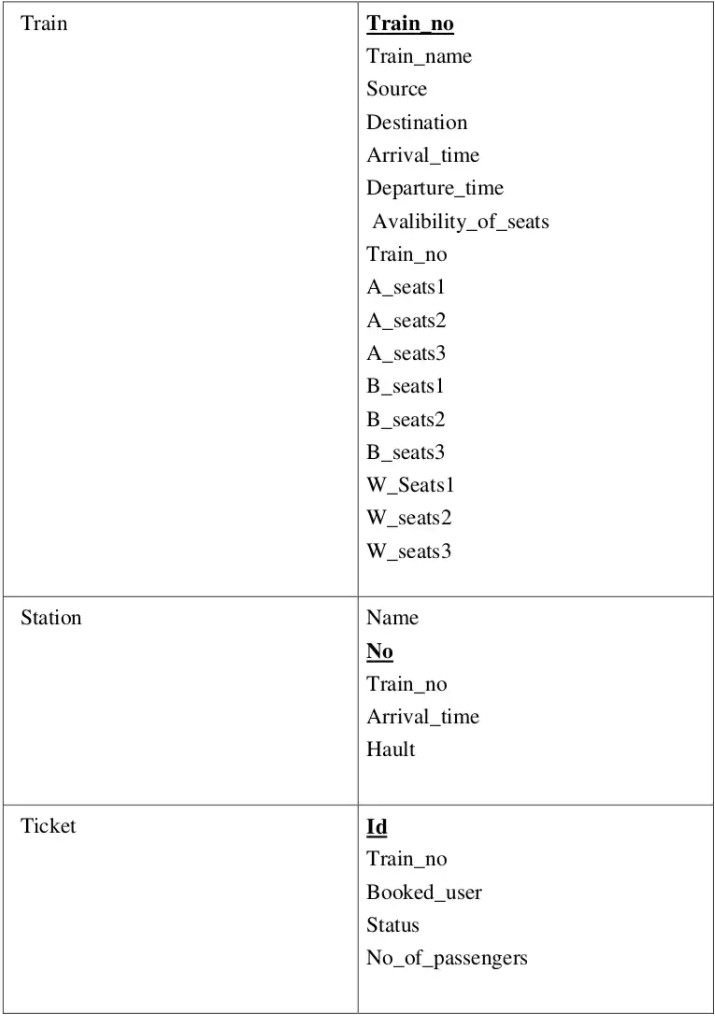
1. The number of trains has been restricted to 5.
2. The booking is open only for next seven days from the current date.
3. Only categories of tickets can be booked, namely, AC and General. 4. The total number of tickets that can be booked in each category (AC and two General) is 10.
4. The total number of tickets that can be given the status of waiting is 2.
5. The in- between stoppage stations and their bookings are not considered.

List of trains has to be maintained. Detailed Passenger information is to be maintained in the booking procedure, the train number, train date, and category are read from the passenger.

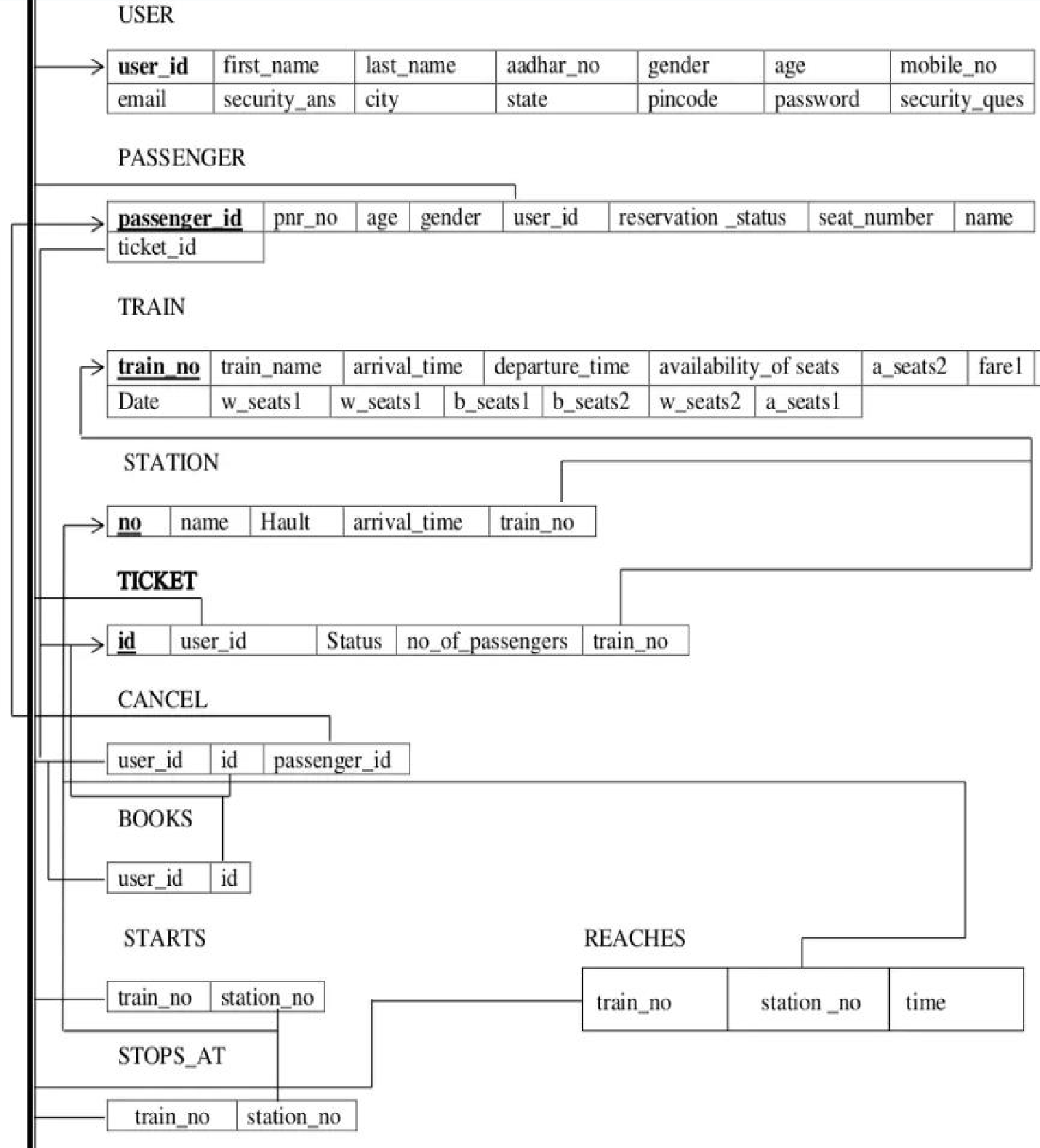
On the basis of the values provided by the passenger, corresponding record is retrieved from the Train Status. If the desired category is AC, then total number of AC seats and number of booked AC seats are compared in order to find whether ticket can be booked or not. Similarly, it can be checked. for the general category. If ticket can be booked, then passenger details are read and stored in the Passenger table. In the cancellation procedure, ticket ID is read from the passenger and corresponding record is searched in the Passenger. If the record exists, it is deleted. After deleting the record (if it is confirmed), first record with waiting status for the same train and same category are searched from the Passenger table and its status is changed to confirm.

# LIST OF ENTITIES & ATTRIBUTES:

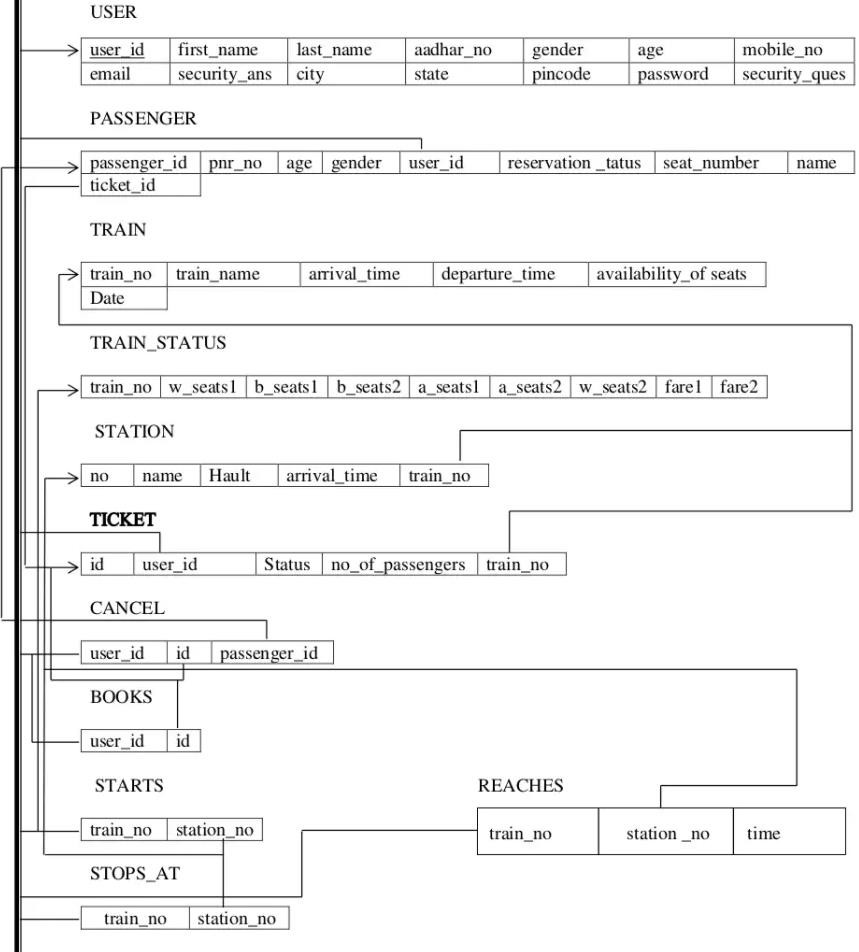




**SCHEMA DIAGRAM:**



# NORMALIZATION & FINAL LIST OF RELATIONS:



## FINAL LIST OF RELATION SHIPS:

* books - Ternary relationship between USER,TRAIN,PASSENGER and TICKET.
* starts -Between TRAIN and STATION
* reaches-Between TRAIN and STATION
* cancel-Between USER and TICKET
* stops at -Between TRAIN and STATION

## CREATE COMMANDS:

create table if not exists USER(user\_id int primary key,first\_name varchar(50),last\_name varchar(50),adhar\_no varchar(20), gender char,age int,mobile\_no varchar(50), email varchar(50),city varchar(50), state varchar(50),pincode varchar(20), password varchar(50), security\_ques varchar(50), security\_ans varchar(50));

reate table if not exsists TRAIN(train\_no int primary key,train\_name varchar(50), arrival\_time time,departure\_time time, availability of seats char,date date);

create table if not exists STATION(no int,name varchar(50),hault int,arrival\_time time,train\_noint,primary key(station\_no,train\_no),constarint foreign key(train\_no) references TRAIN(train\_no));

create table if not exsists TRAIN\_STATUS(train\_no int primary key,b\_seats 1 int,b\_seats2 int,a\_seats1 int,a\_seats2 int,w\_seats1 int,w\_seats2 int,farel float,fare2 float);

create table if not exsists TICKET(id int primary key,user\_id int,status char,no\_of\_passengers int,train\_no int,constraint foreign key (user\_id) references USER(user\_id),constraint foreign key (train\_no) references TRAIN(train\_no)); create table if not exists PASSENGER(passenger\_id int primary key,pnr\_no int,age int,gender char,user\_id int,reservation\_status char,seat\_number varchar(5),name varchar(50),

ticket\_id int,constraint foreign key(user\_id) references USER(user\_id),constraint foreign key(ticket\_id) references TICKET(id));

create table if not exsists STARTS( train\_no int primary key,station\_no int,constraint foreign key(train\_no) references TRAIN(train\_no),constraint foreign key(station\_no) references STATION(no));

create table if not exsists STOPS\_AT( train\_no int station\_no int,constraint foreign key(train\_no) references TRAIN(train\_no),constraint foreign key(station\_no) references STATION(no));

create table if not exsists REACHES (train\_no int,station\_no int,time time,constraint foreign key (train\_no) references TRAIN(train\_no),constraint foreign key(station\_no) references STATION(no));

create table if not exsists BOOKS (user\_id int,id int,constraint foreign key(user\_id) references USER(user\_id),constraint foreign key(id) references TICKET(id));

create table if not exsists CANCEL(user\_id int,id int,passenger\_id int,constraint foreign key(id) references TICKET(id),constraint foreign key (passenger\_id) references PASSENGER (passenger\_id),constraint foreign key(user\_id) references USER(user\_id));

## INSERT COMMANDS:

USER(user\_id,first\_name,last\_name, aadhar\_no,gender,age,mobile\_no,email,city,st ate,pincode, password, security\_ques, security\_ans) values(1701,'vijay','sharma','309887340843','M',34,9887786655',

['vijayl@gmail.co](mailto:%27vijayl@gmail.co) m','vijayawada', 'andhrapradesh', '520001','12345@#','favouritecolour', 'red'), (1702, 'r ohith','kumar', '456709871234','M',45,'9809666555', 'rohith | kumar@gmail.com','gu ntur', 'andhrapradesh', '522004','12 @#345','favourite bike','bmw'), (1703, 'manas vi', 'sree','765843210987','F',20,9995550666','manasvi57@gmail.com','guntu r', 'andhra pradesh','522004','0987hii','favourite flower', 'rose');

insert into TRAIN(train\_no,train\_name,arri val\_time,departure\_time, availability\_of seats,date) values(12711, pinakini exp','113000','114000','A',20170410),(12315, commandel exp','124500',125000', 'NA',20170410);

insert into STATION (no, name,hault,arrival\_time,train\_no) values(111,'vijayawada', 10,'113000', 12711), (222,'tirupathi',5,'1

14500',12315);

insert into TRAIN STATUS(train\_no,w\_seats 1,b\_seats1,b\_seats2,a\_seats 1,a\_seats2.w\_seats 2,fare 1,fare2) values(12711,10,4,0,1,1,0,100,450),(12315,10,5,0,0,2,1,300,600);

insert into TICKET(id,user\_id,status,no\_of\_passengers,train\_no) values(4001,1701, 'C',1,12711),(4002, 1702, 'NC', 1,123 15);

insert into PASSENGERS (passenger\_id,pnr\_no,age,gender,user\_id,reservation

\_status,seat\_number,name,ticket\_id) values(5001,78965,45,'M', 1701, 'C','B6 45','ramesh',4001),(5002,54523,54,F', 1701, W','B3-21',

'surekha',4002); insert into STARTS(train\_no,station\_no) values(12711,111),(12315,222); insert into REACHES(train\_no,station\_no,time) values (12711,222, 040000'),

insert into STOPS\_AT(train\_no,station\_no) values(12711,222),(12315,111);(12315,111,053500');

insert into BOOKS (user\_id,id) values(1701,4001),(1702,4002);

insert into CANCEL(user\_id,id,passenger\_id) values (1701,400 1,5001);

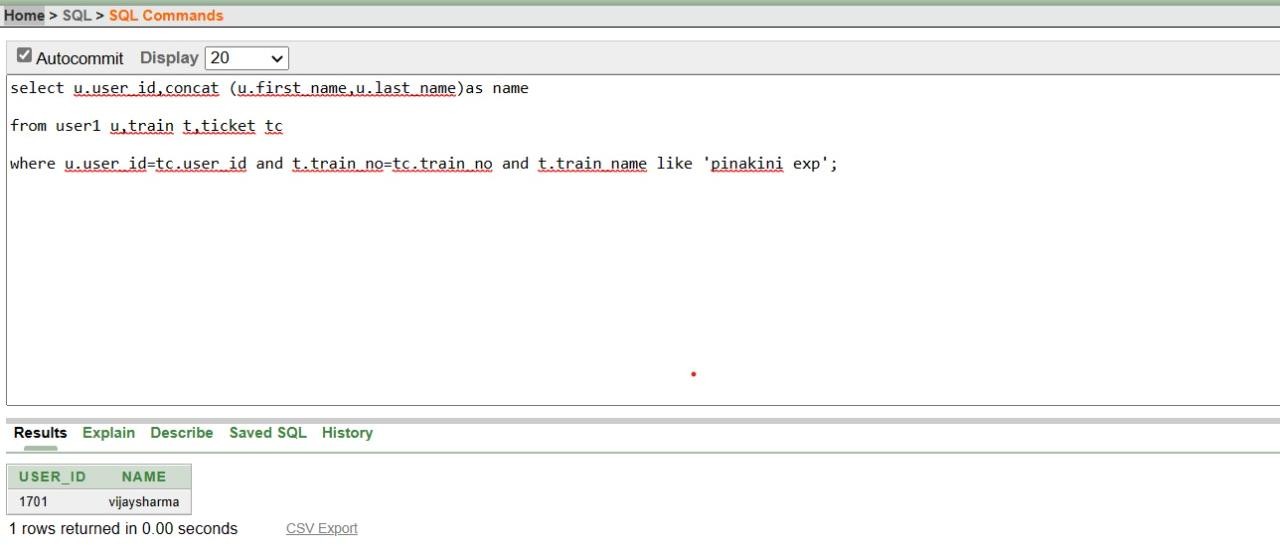
## SQL QUERIES FOR REPORT GENERATION:

1. **print user id and name of all those user who booked ticket for pinakini express**

select u.user\_id,concat (u.first\_name,u.last\_name)as name from user u,train t,ticket tc

where u.user\_id=tc.user\_id and t.train\_no=tc.train\_no and t.train\_name like 'pinakini exp';

**OUTPUT:**



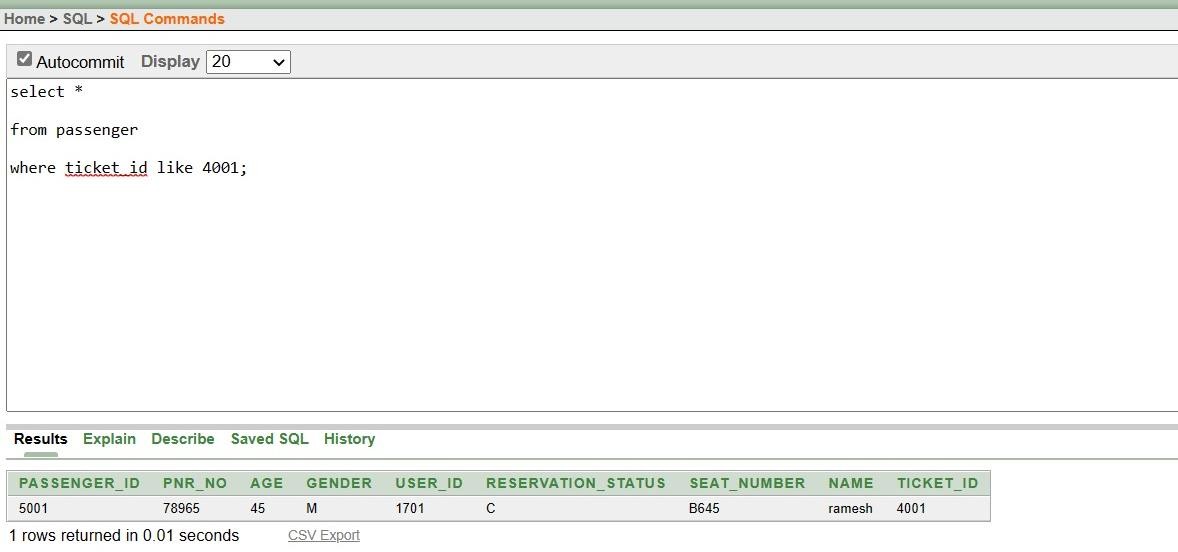
## print details of passengers travelling under ticket no 4001

select \*

from passenger

where ticket\_id like 4001;

**OUTPUT:**



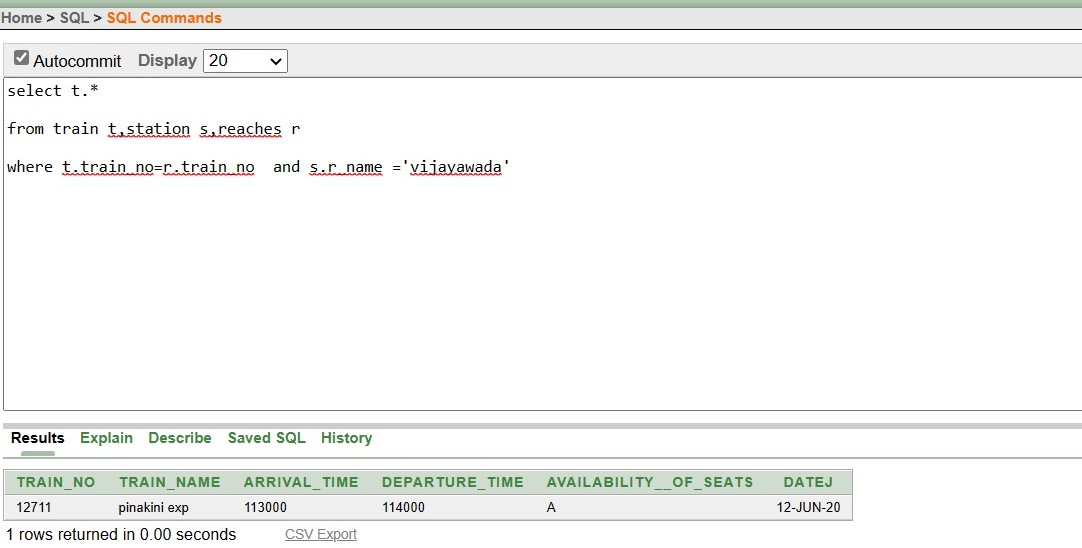
1. **display all those train no's which reach station no**

select t.\*

from train t,station s,reaches r

where t.train\_no=r.train\_no and r.station\_no=s.no and s.name like 'vijayawada';

**OUTPUT:**

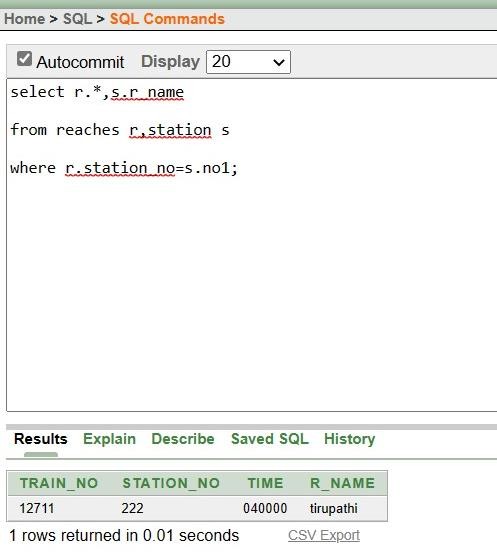


1. **display time at which train no**

select r.\*,s.name

from reaches r,station s where r.station\_no=s.no;

**OUTPUT:**



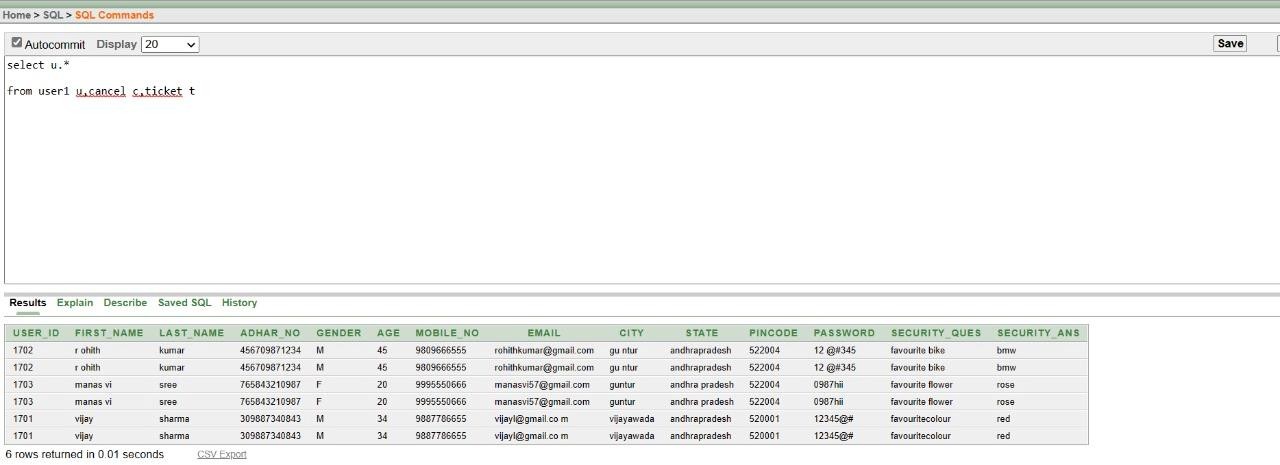
1. **display details of all those users who cancelled tickets for train no**

select u. \*

from user u,cancel c,ticket t

where c.user\_id=u.user\_id and c.id=t.id and t.train\_no like 12711;

**OUTPUT:**

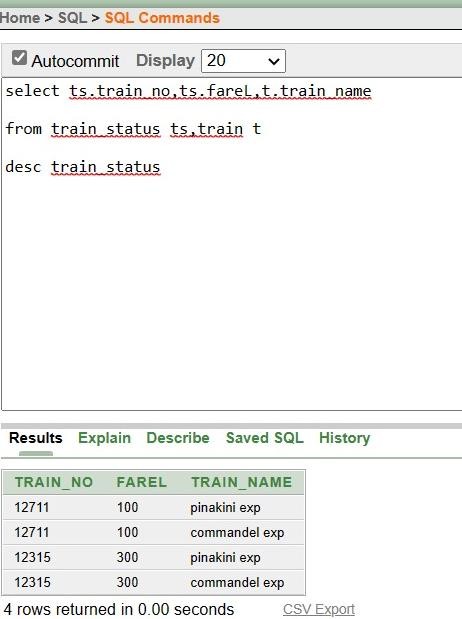


1. **display the train no with increasing order of the fares of class 1**

select ts.train\_no,ts.fare 1,t.train\_name from train status ts,train t

where t.train\_no=ts.train\_no order by fare l asc;

**OUTPUT:**



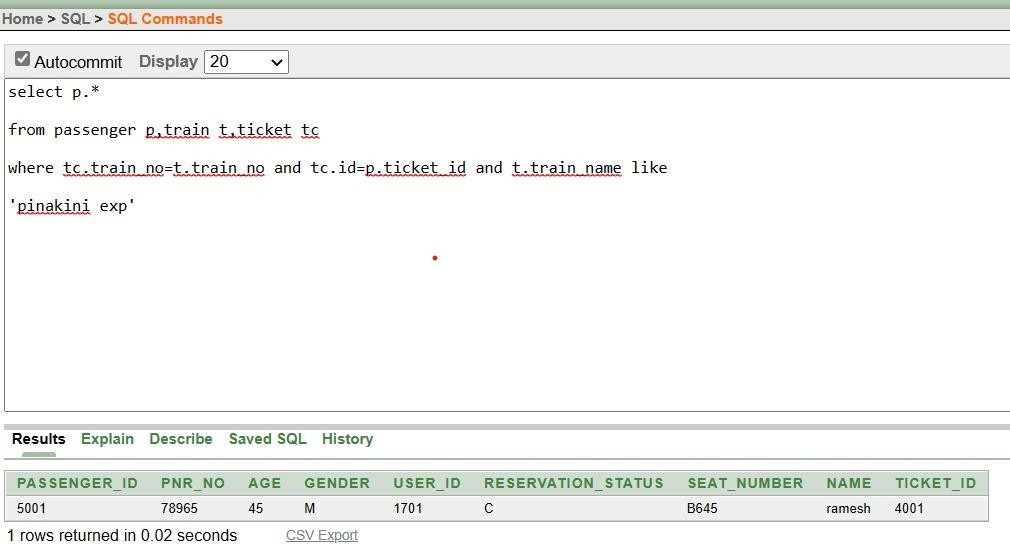
1. **display passenger details for train pinakini.**

select p.\*

from passenger p,train t,ticket tc

where tc.train\_no=t.train\_no and tc.id=p.ticket\_id and t.train\_name like 'pinakini exp';

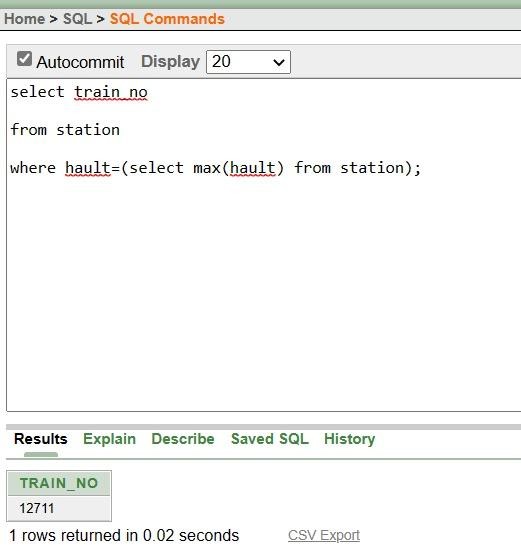
**OUTPUT:**



1. **display the train no which haults for more time in station no**

select train\_no from station having max(hault);

**OUTPUT:**

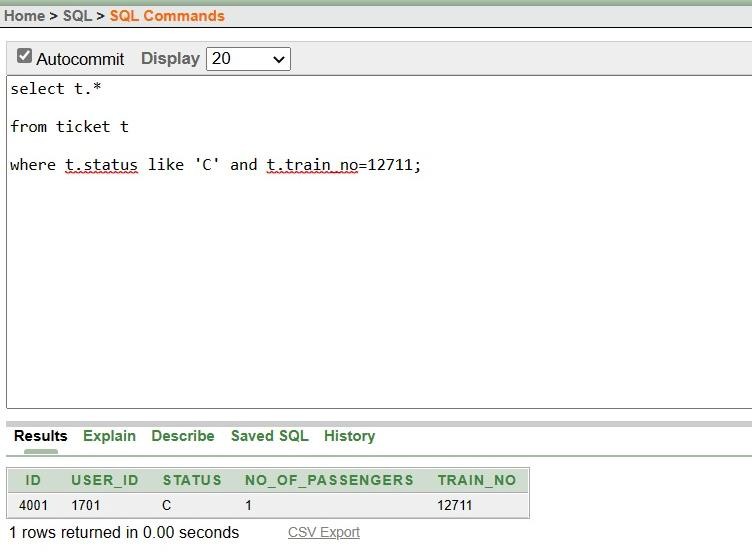


1. **display details of all those passengers whose status is confirmed for train no**

select t.\* from ticket t

where t.status like 'c' and t.train\_no=12711;

**OUTPUT:**



## CONCLUSION:

In our project Railway reservation system, we have stored all the information about the Trains scheduled and the user’s booking tickets and even status of trains, seats etc. This data base is helpful for the applications which facilitate passengers to book the train tickets and check the details of trains and their status from their place itself it avoids inconveniences of going to railway station for each and every query they get. We had considered the most important requirements only, many more features and details cand be added to our project in order to obtain even more user-friendly applications. These applications are already in progress and in future they can be upgraded and may become part of amazing technology.