

**DAYANANDA SAGAR UNIVERSITY**

**DEPT. OF COMPUTER SCIENCE AND ENGINEERING**

**SOFTWARE REQUIREMENT SPECIFICATION**

**TOPIC: RAILWAY MANAGEMENT SYSTEM**

**5TH SEMESTER -3RD YEAR**

**DBMS PROJECT**

**TEAM MEMBERS:**

***MUSKAAN GOEL(TEAM LEADER)***

***LYSETTI LAKSHMI POOJITHA***

***MEGHANA SHREE M***

***MUSKAAN SINHA***

**GUIDE: PROF. BHARGAVI MOKASHI (DEPT. OF COMPUTER SCIENCE AND ENGINEERING)**

**TABLE OF CONTENTS ACKNOWLEDGMENTS**

Abstract...................................................................................................................

Introduction...........................................................................................................

Project Description..............................................................................................

List of Entities & Attributes...............................................................................

ER Diagram (Conceptual Model) ....................................................................

Normalization Schema Diagram.....................................................................

Final List of Relations ......................................................................................

Create & Insert SQL Queries.............................................................................

SQL Queries related to Report Generation...................................................

Conclusion .........................................................................................................

**ABSTRACT:**

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.

This project contains Introduction to the Railways reservation system .It is the computerized system of reserving the seats of train seats in advanced. It is mainly used for long route. On-line reservation has made the process for the reservation of seats very much easier than ever before.

In our country India, there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Then this project contains entity relationship model diagram based on railway reservation system and introduction to relation model .There is also design of the database of the railway reservation system based on relation model. Example of some SQL queries to retrieves data from rail management database.

**INTRODUCTION:**

Database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

The main purpose of maintaining database for Railway Reservation System is to reduce the manual errors involved in the booking and cancellation of tickets and make it convenient for the customers and providers to maintain the data about their customers and also about the seats available at them.Due to automation many loopholes that exist in the manual maintenance of the records can be removed. The speed of obtaining and processing the data will be fast. For future expansion the proposed system can be web enabled so that clients can make various enquiries about trains between stations. Due to this, sometimes a lot of problems occur and they are facing many disputes with customers. To solve the above problem, we design a database which includes customer details, availability of seats in trains, no of trains and their details.

**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 that level. Therefore, a small sample case study has been created to demonstrate.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 the next seven days from the current date.

3. Only two 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 General) is 10.

5. The total number of tickets that can be given the status of waiting is 2.

6. The in‐ between stoppage stations and their bookings are not considered.

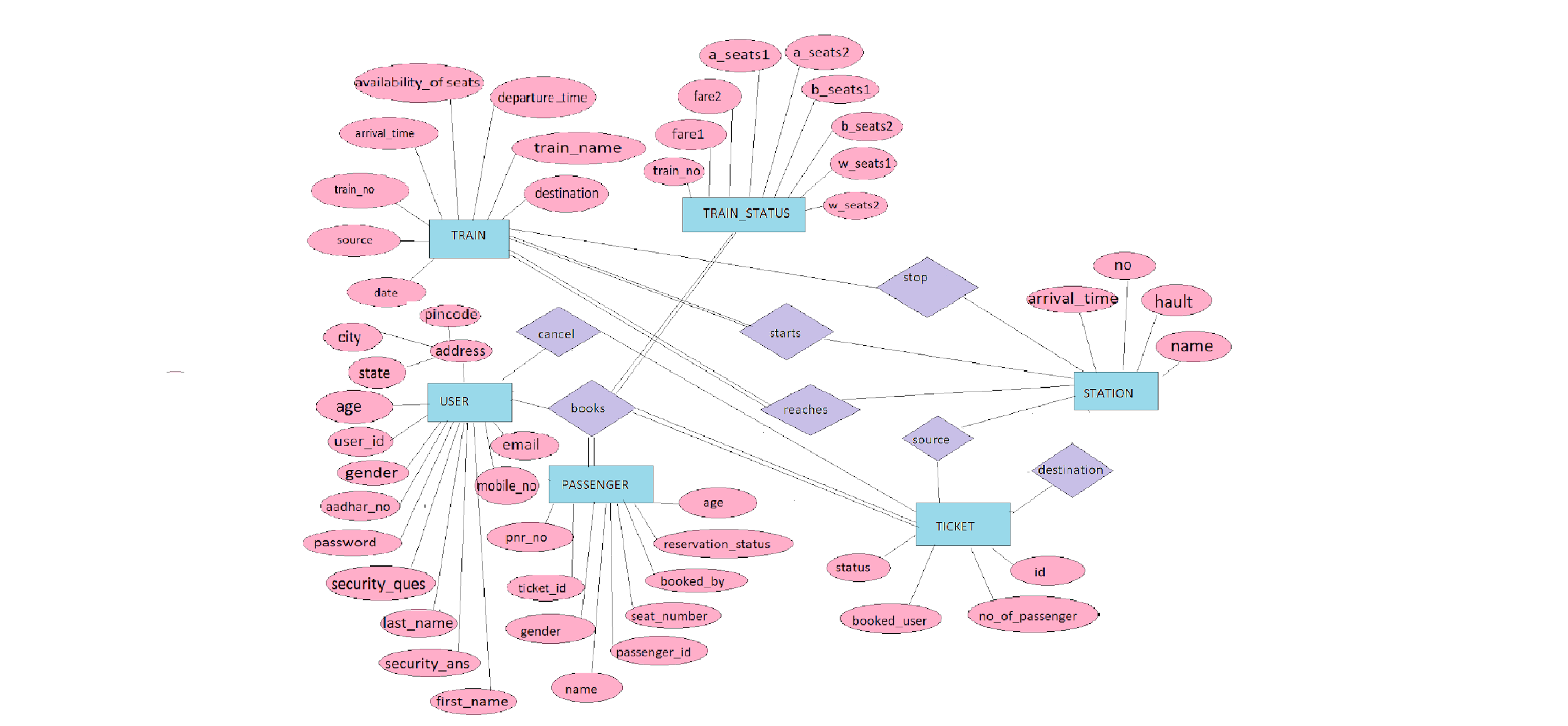
List of trains has to be maintained. Detailed Passenger information is to be maintained In the booking procedure, the train number, train date, and category are read from the passenger. On the basis of the values provided by the passenger, a corresponding record is retrieved from the Train\_Status. If the desired category is AC, then the total number of AC seats and number of booked AC seats are compared in order to find whether tickets can be booked or not. Similarly, it can be checked for the general category. If a 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), the 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:**

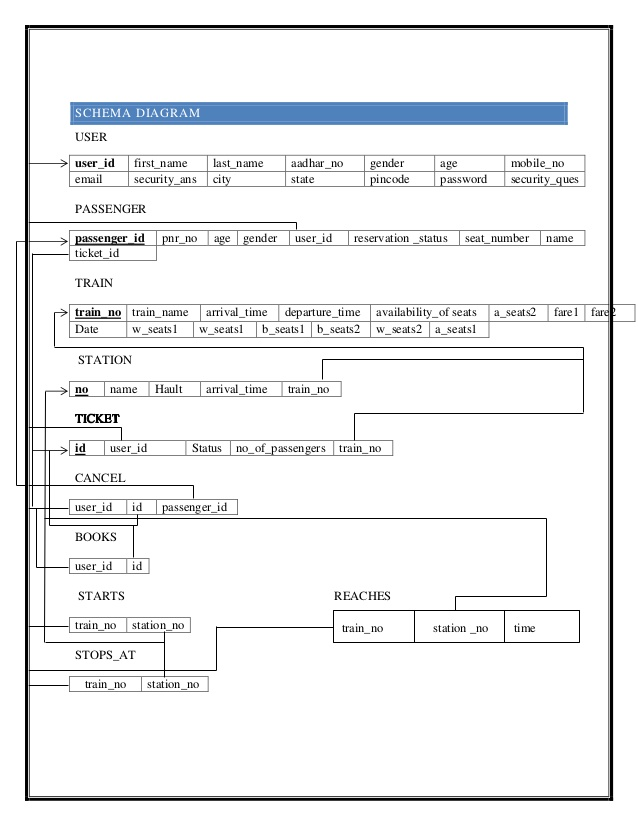
|  |  |
| --- | --- |
| **ENTITIES** | **ATTRIBUTES** |
| User | User\_id  Password  First\_name  Last\_name  Gender  Age  Email  Aadhar\_no  Mobile\_no  City  State  Pincode  Security\_ques  Security\_ans |
| Passenger | Passenger\_id  Name  Gender  Age  Pnr\_no  Seat\_no  Booked\_by  Reservation\_status |

|  |  |
| --- | --- |
| Train | Train\_no  Train\_name  Source  Destination  Arrival\_time  Departure\_time  Avalibility\_of\_seats  Train\_no  A\_seats1  A\_seats2  A\_seats3  B\_seats1  B\_seats2  B\_seats3  W\_Seats1  W\_seats2  W\_seats3 |
| Station | Station\_Name  Station\_No  Train\_no  Arrival\_time  Hault |
| Ticket | Ticket\_id  User\_id  Train\_no  Booked\_user  Status  No\_of\_passengers |

**ER DIAGRAM (CONCEPTUAL MODEL):**



**Normalization Schema Diagram:**

****

**FINAL LIST OF RELATIONSHIPS:**

1. books -Ternary relationship between USER,TRAIN,PASSENGER and TICKET.
2. starts –Between TRAIN and STATION
3. reaches –Between TRAIN and STATION
4. cancel –Between USER and TICKET
5. stops\_at –Between TRAIN and STATION

**CREATE & INSERT SQL QUERIES CREATE COMMANDS:**

**CREATE QUERIES:**

CREATE TABLE USER(USER\_ID INT PRIMARY KEY,\_PASSWORD VARCHAR(50),SECURITY\_QUESTION VARCHAR(50),SECURITY\_ANSWER VARCHAR(50));

CREATE TABLE TRAIN(TRAIN\_NO INT PRIMARY KEY,TRAIN\_NAME VARCHAR(50),ARRIVA\_TIME TIME,DEPARTTURE\_TIME TIME,AVAILABILITY\_OF\_SEATS CHAR,DATE DATE);

CREATE TABLE STATION(STATION\_NO INT,STATION\_NAME VARCHAR(50),HAULT INT,ARRIVAL\_TIME TIME,TRAIN\_NO INT,PRIMARY KEY(STATION\_NO,TRAIN\_NO),FOREIGN KEY(TRAIN\_NO) REFERENCES TRAIN(TRAIN\_NO));

CREATE TABLE TRAIN\_STATUS(TRAIN\_NO INT PRIMARY KEY,B\_SEATS1 INT,B\_SEATS2 INT,A\_SEATS1 INT,A\_SEATS2 INT,W\_SEATS1 INT,W\_SEATS2 INT,FARE1 FLOAT,FARE2 FLOAT);

CREATE TABLE TICKET(TICKET\_ID INT PRIMARY KEY,USER\_ID INT,STATUS CHAR,NO\_OF\_PASSENGERS INT,TRAIN\_NO INT,FOREIGN KEY(USER\_ID) REFERENCES USER(USER\_ID),FOREIGN KEY(TRAIN\_NO) REFERENCES TRAIN(TRAIN\_NO));

CREATE TABLE PASSENGER(PASSENGER\_ID INT PRIMARY KEY,PNR\_NO INT,AGE INT,GENDER CHAR,USER\_ID INT,RESERVATION\_STATUS CHAR,SEAT\_NO VARCHAR(50),NAME VARCHAR(50),TICKET\_ID INT, FOREIGN KEY(USER\_ID) REFERENCES USER(USER\_ID),FOREIGN KEY(TICKET\_ID) REFERENCES TICKET(TICKET\_ID));

CREATE TABLE STARTS(TRAIN\_NO INT PRIMARY KEY,STATION\_NO INT,FOREIGN KEY(TRAIN\_NO) REFERENCES TRAIN(TRAIN\_NO),FOREIGN KEY(STATION\_NO) REFERENCES STATION(STATION\_NO));

CREATE TABLE STOPS(TRAIN\_NO INT,STATION\_NO INT,FOREIGN KEY(TRAIN\_NO) REFERENCES TRAIN(TRAIN\_NO),FOREIGN KEY(STATION\_NO) REFERENCES STATION(STATION\_NO));

CREATE TABLE REACHES(TRAIN\_NO INT,STATION\_NO INT,TIME TIME,FOREIGN KEY(TRAIN\_NO) REFERENCES TRAIN(TRAIN\_NO),FOREIGN KEY(STATION\_NO) REFERENCES STATION (STATION\_NO));

CREATE TABLE CANCEL(USER\_ID INT,TICKET\_ID INT,PASSENGER\_ID INT,FOREIGN KEY(TICKET\_ID) REFERENCES TICKET(TICKET\_ID),FOREIGN KEY(PASSENGER\_ID) REFERENCES PASSENGER(PASSENGER\_ID),FOREIGN KEY(USER\_ID) REFERENCES USER(USER\_ID));

**INSERT QUERIES:**

INSERT INTO USER VALUES(150,"HELLO","AGE",20),

(165,"HELLOO","AGE",20),(177,"HELLOOO","AGE",20),(178,"HELLOOOO","AGE",20);

INSERT INTO TRAIN VALUES(1001,"RAJDHANI",11,1,"P",20201011),

(1002,"INDIAN\_RAILWAYS",12,2,"M",20201011),(1003,"KARNATAKA\_EXPRESS",13,3,"G",20201011),(1004,"MUMBAI\_EXPRESS",14,4,"S",20201011);

INSERT INTO STATIONVALUES(100,"RAJASTHAN",10,112030,1001),

(101,"DELHI",11,112131,1002),(102,"BANGALORE",12,112032,1003),

(103,"MUMBAI",13,112132,1004);

INSERT INTO TRAIN\_STATUS VALUES(1001,90,91,80,81,70,71,7000,6000),

(1002,20,21,30,31,40,41,9000,8000),(1003,10,11,20,21,60,68,7500,5400),(1004,33,34,56,57,87,89,8700,4567);

INSERT INTO TICKET VALUES(9880,150,"A",5,1001),(9881,165,"N",4,1002),

(9882,177,"A",3,1003),(9883,178,"N",2,1004);

INSERT INTO PASSENGER VALUES(1500,123,20,”F”,150,”A”,21,"POOJITHA",9880) ,(1650,234,20,"F",165,"A",22,"MEGHANA",9881),(1770,345,20,"F",177,"A",23,"MUSKAANG",9882),(1780,456,20,"F",178,"A",24,"MUSKAANS",9883);

INSERT INTO STARTS VALUES(1001,100),(1002,101),(1003,102),(1004,103);

INSERT INTO STOPS VALUES(1004,100),(1003,101),(1002,102),(1001,103);

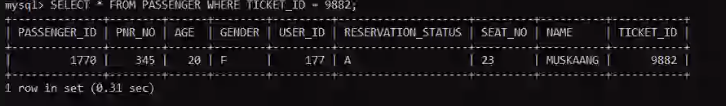
INSERT INTO REACHES VALUES(1001,103,112233),(1002,102,122334),

(1003,101,132435),(1004,100,142536);

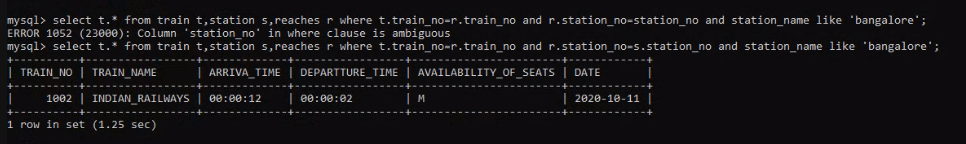
INSERT INTO CANCEL VALUES(150,9880,1500);

**SQL QUERIES RELATED TO REPORT GENERATION**

1. **print user id and name of all those user who booked ticket for** KARNATAKA\_EXPRESS
2. **print details of passengers travelling under ticket no 9882**



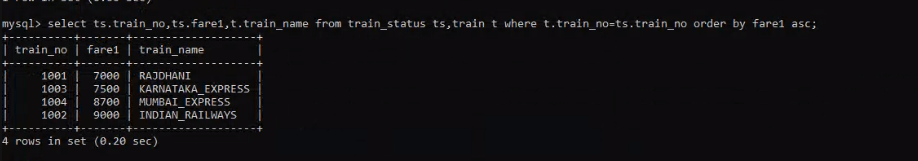
**3.display all those train no's which reach station no**

****

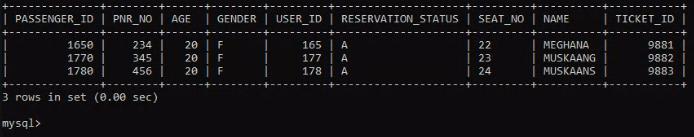
**4. display time at which train no----- reaches station no ------**

**5.display details of all those users who canceled tickets for train no--**

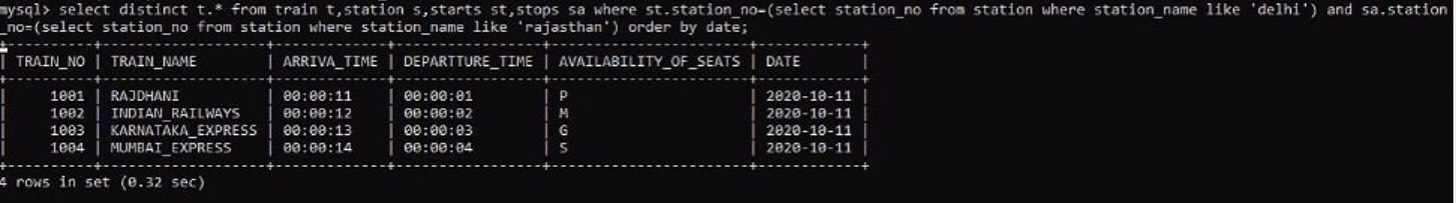
**6. display the train no with increasing order of the fares of class 1**

****

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

****

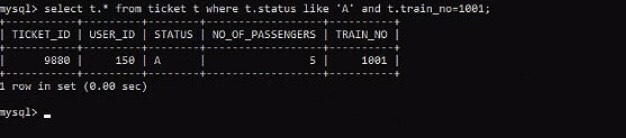
**8 display immediate train from tirupathi to Vijayawada**

****

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

****

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

****

**select p.\* from passenger p,train t,ticket tc where tc.train\_no=t.train\_no and tc.ticket\_id=p.ticket\_id and t.train\_name like "rajdhani";**

**select distinct t.\* from train t,station s,starts st,stops sa where st.station\_no=(select station\_no from station where station\_name like 'delhi') and sa.station\_no=(select station\_no from station where station\_name like 'rajasthan') order by date;**

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

**select t.\* from ticket t where t.status like 'A' and t.train\_no=1001;**

**CONCLUSION:**

In our project Railway reservation system we have stored all the information about the Trains scheduled and the users booking tickets and even status of trains, seats etc. This database 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 the railway station for each and every query they get. We had considered the most important requirements only, many more features and details can be added to our project inorder 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.