**RAILWAY RESERVATION SYSTEM**

**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.

**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 Entities and Attributes**

|  |  |
| --- | --- |
| **ENTITIES** | **ATTRIBUTES** |
| **USERS** | **user\_id**  password  first\_name  last\_name  gender  age  email  adhar\_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  availability\_of\_seats  train\_no  A\_seats1  A\_seats2  A\_seats3  B\_seats1  B\_seats2  B\_seats3  W\_seats1  W\_seats2  W\_seats3 |
| **STATION** | **Station\_no**  Station\_name  train\_no  arrival\_time  hault |

|  |  |
| --- | --- |
| **TICKET** | **Ticket\_id**  train\_no  booked\_user  Status  no\_of\_passengers |

**Data Flow Diagram(DFD’S)**

**ZERO LEVEL DFD:-**

**Zero Level DFD-Railway Reservation System**

**System User Management**

**Login Management**

**Booking Management**

**Ticket Management**

**Route Management**

**Train Management**

**Railway Reservation System**

**First Level DFD:-**

**Railway Reservation System**

**Train Management**

**Booking Management**

**Ticket Management**

**Route Management**

**Login Management**

**System User Management**

**Generate Trains Report**

**Generate Booking Report**

**Generate Ticket Report**

**Generate Route Report**

**Check User Login Details**

**Generate System User Report**

**First Level DFD-Railway Reservation System**

**Railway Reservation System**

**Generate Trains Report**

**Train Management**

**Booking Management**

**Ticket Management**

**Route Management**

**Login Management**

**System User Management**

**Second Level DFD:-**

**Manage Report**

**Manage Cancelled Train Details**

**Check Roles Of Access**

**Manage Modules**

**Check Credentials**

**Login to System**

**Manage User Permission**

**Manage Roles of User**

**Manage System Admins**

**Second Level DFD-Railway Reservation System**

**Manage Train Schedule Details**

**Manage Payment Details**

**Manage Ticket Details**

**Manage Booking Details**

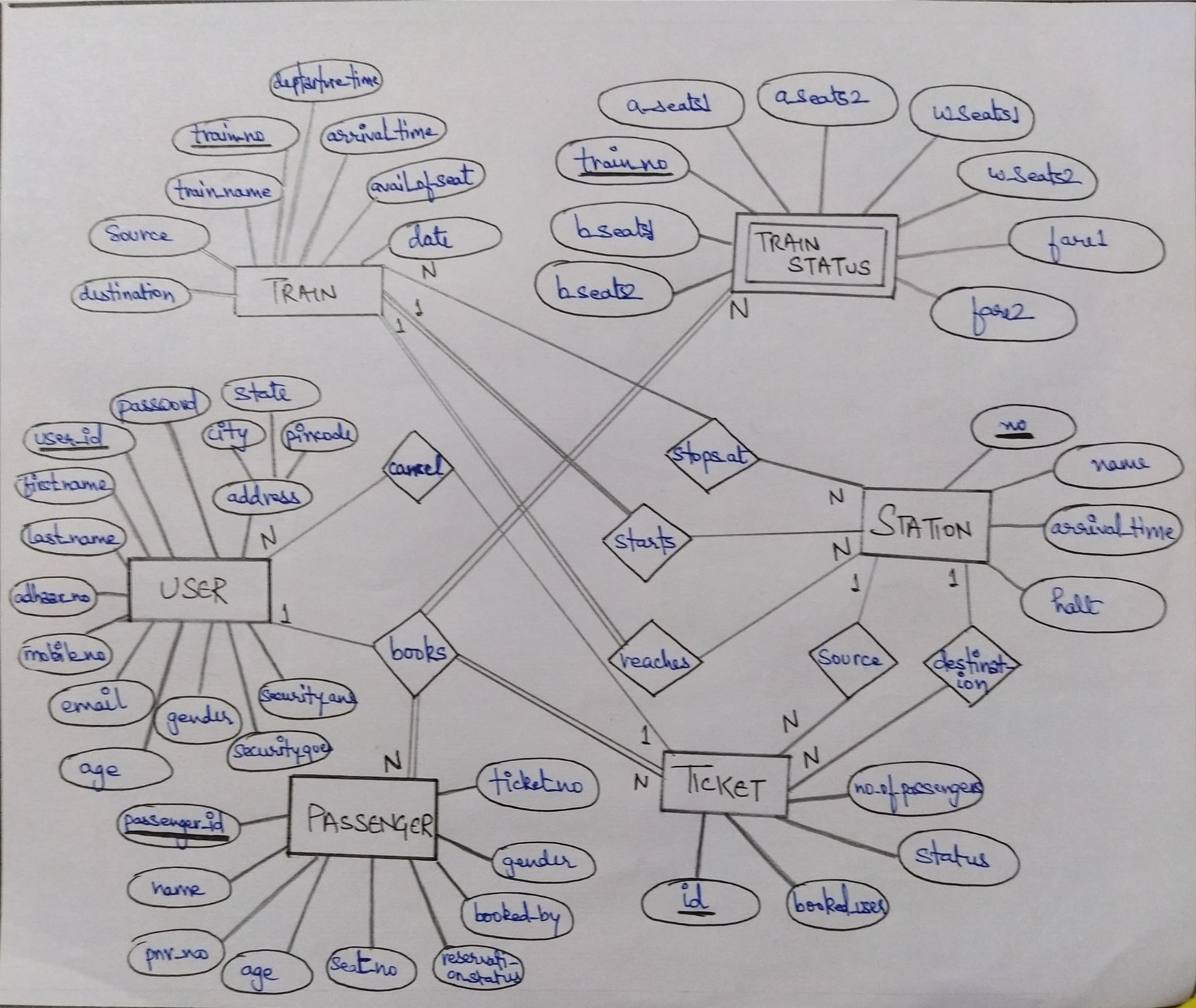
**Manage Train Details**

**Send Email to User**

**Forgot Password**

**Admin**

**ER-Diagram**

****

**CREATION QUERIES FOR TABLES:-**

**TABLE-1**

create table USERS(user\_id int primary key,first\_name varchar2(50),last\_name varchar2(50),

adhar\_no varchar2(20),gender char,age int,mobile\_no varchar2(50),email varchar2(50),city varchar2(50),

state varchar2(50),pincode varchar2(20),password varchar2(50),security\_ques varchar2(50),security\_ans varchar2(50));

**TABLE-2**

create table TRAIN(train\_no int primary key,train\_name varchar2(50),arrival\_time VARCHAR2(10),departure\_time VARCHAR2(10),

availability\_of\_seats char,DATE\_train Date);

**TABLE-3**

create table STATION(Station\_no int primary key, Station\_Name varchar2(50),hault int,arrival\_time VARCHAR2(10),train\_no int,

FOREIGN KEY(train\_no) REFERENCES TRAIN(train\_no));

**TABLE:-4**

create table TRAIN\_STATUS(train\_no int primary key,b\_seats1 int,b\_seats2 int,a\_seats int,a\_seats2 int,w\_seats int,w\_seats2 int,farel float,fare2 float);

**TABLE-5**

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 USERS(user\_id),

FOREIGN KEY(train\_no) REFERENCES TRAIN(train\_no));

**TABLE-6**

create table 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,

FOREIGN KEY(user\_id) REFERENCES USERS(user\_id),

FOREIGN KEY(ticket\_id) REFERENCES TICKET(Ticket\_id));

**TABLE-7**

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));

**TABLE-8**

create table STOPS\_AT( train\_no int,station\_no int,

FOREIGN KEY(train\_no) REFERENCES TRAIN(train\_no),

FOREIGN KEY(station\_no) REFERENCES STATION(Station\_no));

**TABLE-9**

create table REACHES(train\_no int,station\_no int, time VARCHAR2(10),

FOREIGN KEY(train\_no) REFERENCES TRAIN(train\_no),

FOREIGN KEY(station\_no) REFERENCES STATION(Station\_no));

**TABLE-10**

create table BOOKS( user\_id int,ticket\_id int,FOREIGN KEY(user\_id)

REFERENCES USERS(user\_id),FOREIGN KEY(ticket\_id) REFERENCES TICKET(Ticket\_id));

**TABLE-11**

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 USERS(user\_id));

**INSERT QUERIES:**

INSERT INTO USERS(user\_id,first\_name,last\_name,adhar\_no,gender,age,mobile\_no,email,city,state,pincode, password,security\_ques,security\_ans)

VALUES(1701, 'vijay','sharma',309887340843,'M',34,9887786655, 'vijay@gmail.com','vijayawada', 'andhrapradesh', 520001,'12345@#','favouritecolour', 'red');

INSERT INTO USERS(user\_id,first\_name,last\_name,adhar\_no,gender,age,mobile\_no,email,city,state,pincode, password,security\_ques,security\_ans)

VALUES (1702,'rohit', 'kumar', 456709871234,'M',45,9809666555,'rohitkumar@gmail.com','guntur', 'andhrapradesh', 522004,'12@#345','favouritebike','bmw');

INSERT INTO USERS(user\_id,first\_name,last\_name,adhar\_no,gender,age,mobile\_no,email,city,state,pincode, password,security\_ques,security\_ans)

VALUES(1703, 'manasvi','sree',765843210987, 'F',20,9995550666,'manasvi57@gmail.com','guntur','andhrapradesh', 522004,'0987hii','favourite flower','rose');

INSERT INTO

TRAIN(train\_no,train\_name,arrival\_time,departure\_time,availability\_ofseats,date) values(12711,'pinakiniexp','113000','114000','A',20170410),(12315, cormandelexp','124500,125000', 'NA',20170410);

INSERT INTO

STATION(no,name,hault,arrival\_time,train\_no)values(111, vijayawada', 10, 113000', 12711),(222, 'tirupathi',S,'114500,12315);

INSERT INTO

TRAIN STATUS(train\_no,w\_seats 1,b\_seats 1,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

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

STOPS\_AT(train\_no,station\_no) values(12711,222),(12315,111);

INSERT INTO

REACHES(train\_no,station\_no,time) values(12711,222,040000'),(12315,111,053500');

INSERT INTO

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

INSERT INTO

CANCEL(user\_id,id passenger\_id) values(1701,4001,5001);

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

from USERS u,train t,ticket te

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

|  |
| --- |
|  |

**2. Print details of passengers travelling under ticket no 4001**

select \*from PASSENGER where ticket\_id like 4001;

|  |
| --- |
|  |

**3. 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';

|  |
| --- |
|  |

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

select r.\*,s.name

from REACHES r,station s

where r.station\_no=s.no;

|  |
| --- |
|  |

**5. Display details of all those users who cancled tickets for train no------**

select u.\* from USERS u,cancel c,ticket t

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

|  |
| --- |
|  |

**6.Display passenger details for train pinakini.**

select p.\*

from PASSENGER p,train t,ticket te

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

|  |
| --- |
|  |

**7. Display immediate train from tirupathi to Vijayawada**

select distinct t.\* from TRAIN t,station s,starts st,stops at sa

where st.station\_no=(select no from station where name like 'tirupathi')

and sa.station\_no=(select no from station where name like 'vijayawada')

order by date;

|  |
| --- |
|  |

**8. Display the train no which haults for more time in station no--------**

select train\_no from STATION

having max(hault);

|  |
| --- |
|  |

**9. Display details of all those passengers whose status is confirmed for train no----**

select t.\* from TICKET t

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

|  |
| --- |
|  |

**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 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 can 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.