Project Report
On
“Railway Management System”
By
Priyanshu Mangal

Overview of Project

This project is about creating a database about the Railway Management System. The railway management 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, stations, and passengers. The record of the train includes its number, name, days on which it is available etc. Passengers can book their tickets for the train in which seats are available. For this, passengers have 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 No 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.

Table of Contents

Introduction

Following are the entities along with their attributes.

```
`Account`
          (
              `Username` varchar(15) NOT NULL,
              `Password` varchar(20) NOT NULL,
              `Email_Id` varchar(35) NOT NULL,
              `Address` varchar(50) DEFAULT
             NULL,
              PRIMARY KEY (`Username`)
          )
```

```
'Contact'  (

            `Username` varchar(15) NOT NULL DEFAULT '',
            `Phone_No` char(10) NOT NULL DEFAULT '',
            PRIMARY KEY (`Username`,`Phone_No`),
            CONSTRAINT `Contact_ibfk_1` FOREIGN KEY (`Username`) REFERENCES
           `Account` (`Username`) ON DELETE CASCADE
          )
```

```
'Passenger'
          (
            `Passenger_Id` int(11) NOT NULL AUTO_INCREMENT,
            `First_Name` varchar(20) NOT NULL,
            `Last_Name` varchar(20) NOT NULL,
            `Gender` char(1) NOT NULL,
            `Phone_No` char(10) DEFAULT NULL,
            `Ticket_No` int(10) NOT NULL,
            `Age` int(11) NOT NULL,
            `Class` varchar(20) NOT NULL,
            PRIMARY KEY (`Passenger_Id`),
            KEY `Ticket_No` (`Ticket_No`),
            CONSTRAINT `Passenger_ibfk_1` FOREIGN KEY (`Ticket_No`) REFERENCES
           `Ticket` (`Ticket_No`) ON DELETE CASCADE
          )
```

`Station`

```
(
 `Station_Code` char(5) NOT NULL DEFAULT
 `Station_Name` varchar(25) NOT NULL,
 PRIMARY KEY (`Station_Code`)
 )
```

'Stoppage'

```
(
 `Train_No` int(6) NOT NULL DEFAULT '0',
 `Station_Code` char(5) NOT NULL DEFAULT '',
 `Arrival_Time` time DEFAULT NULL,
 `Departure_Time` time DEFAULT NULL,
 PRIMARY KEY (`Train_No`,`Station_Code`),
 KEY `Station_Code` (`Station_Code`),
 CONSTRAINT `Stoppage_ibfk_1` FOREIGN KEY (`Train_No`) REFERENCES `Train`
(`Train_No`) ON DELETE CASCADE ON UPDATE CASCADE,
 CONSTRAINT `Stoppage_ibfk_2` FOREIGN KEY (`Station_Code`) REFERENCES `Station`
(`Station_Code`) ON DELETE CASCADE ON UPDATE CASCADE
)
```

'Ticket'

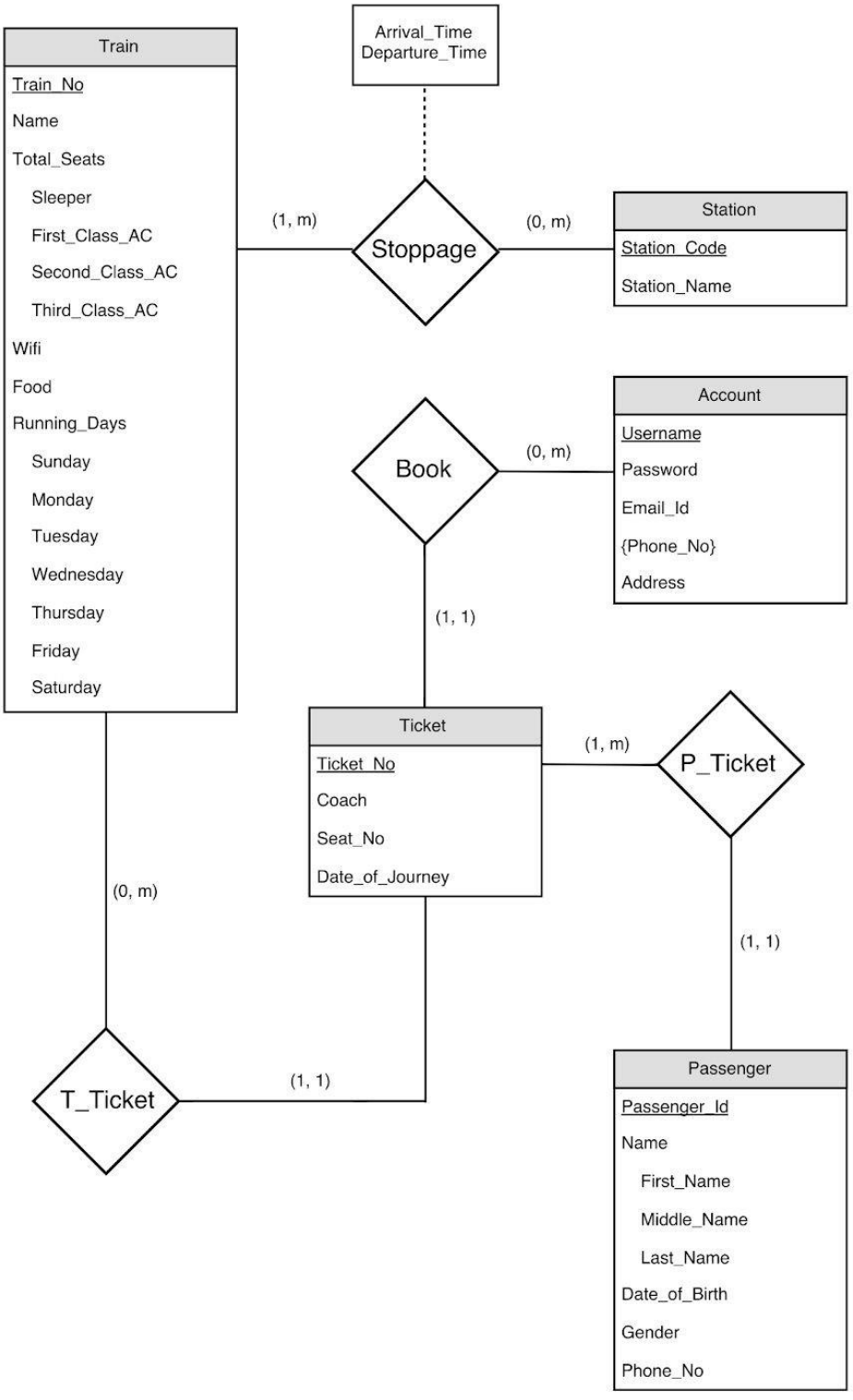
```
(
 `Ticket_No` int(10) NOT NULL AUTO_INCREMENT,
 `Train_No` int(6) NOT NULL,
 `Date_of_Journey` date NOT NULL,
 `Username` varchar(15) NOT NULL,
 PRIMARY KEY (`Ticket_No`),
 KEY `Username` (`Username`),
 KEY `Train_No` (`Train_No`),
 CONSTRAINT `Ticket_ibfk_1` FOREIGN KEY (`Username`) REFERENCES
`Account` (`Username`) ON DELETE CASCADE)
)
```

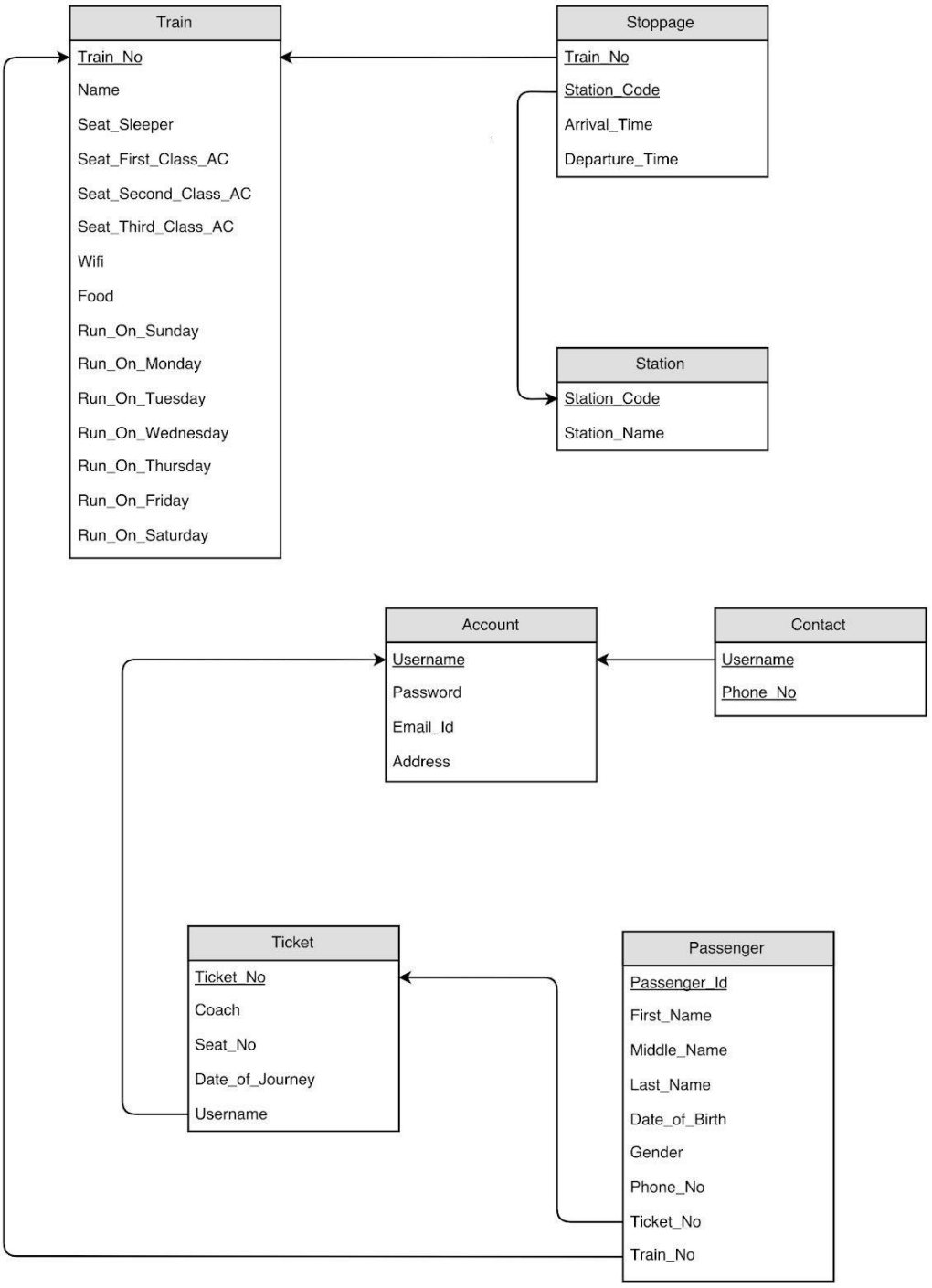
'Train'

```
 (
`Train_No` int(6) NOT NULL DEFAULT '0',
`Name` varchar(25) NOT NULL,
```

```
`Seat_Sleeper` int(4) NOT NULL,

`Seat_First_Class_AC` int(4) NOT NULL,
`Seat_Second_Class_AC` int(4) NOT NULL,
`Seat_Third_Class_AC` int(4) NOT NULL,
`Wifi` char(1) NOT NULL,

`Food` char(1) NOT NULL,

`Run_On_Sunday` char(1) NOT NULL,

`Run_On_Monday` char(1) NOT NULL,

`Run_On_Tuesday` char(1) NOT NULL,

`Run_On_Wednesday` char(1) NOT NULL,

`Run_On_Thursday` char(1) NOT NULL,

`Run_On_Friday` char(1) NOT NULL,

`Run_On_Saturday` char(1) NOT NULL, PRIMARY KEY (`Train_No`)
)
```

ER Model


Train
Train_No
Name
Total_Seats
Sleeper
First_Class_AC
Second_Class_AC
Third_Class_AC
Wifi
Food
Running_Days
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Arrival_Time
Departure_Time
Stoppage
(1, m)
(0, m)
Station
Station_Code
Station_Name
Book
(0, m)
Account
Username
Password
Email_Id
{Phone_No}
Address
(1, 1)
Ticket
Ticket_No
Coach
Seat_No
Date_of_Journey
(1, m)
P_Ticket
(1, 1)
(0, m)
T_Ticket
(1, 1)
Passenger
Passenger_Id
Name
First_Name
Middle_Name
Last_Name
Date_of_Birth
Gender
Phone_No


Train
Train_No
Name
Seat_Sleeper
Seat_First_Class_AC
Seat_Second_Class_AC
Seat_Third_Class_AC
Wifi
Food
Run_On_Sunday
Run_On_Monday
Run_On_Tuesday
Run_On_Wednesday
Run_On_Thursday
Run_On_Friday
Run_On_Saturday
Stoppage
Train_No
Station_Code
Arrival_Time
Departure_Time
Station
Station_Code
Station_Name
Account
Username
Password
Email_Id
Address
Contact
Username
Phone_No
Ticket
Ticket_No
Coach
Seat_No
Date_of_Journey
Username
Passenger
Passenger_Id
First_Name
Middle_Name
Last_Name
Date_of_Birth
Gender
Phone_No
Ticket_No
Train_No

FD and Normalization

Functional dependencies

TRAIN

`Train_No`->(`Name`, `Seat_Sleeper` , `Seat_First_Class_AC`, `Seat_Second_Class_AC`, Seat_Third_Class_AC`,`Wifi`,`Food`,`Run_On_Sunday``Run_On_Monday``Run_On_Tuesday` `Run_On_Wednesday` `Run_On_Thursday` `Run_On_Friday`,`Run_On_Saturday`)

STOPPAGE

(`Train_No`, `Station_Code`) -> (`Arrival_Time` , `Departure_Time`)

TICKET

`Ticket_No`-> (`Train_No` `Date_of_Journey` `Username`)

STATION

`Station_Code` -> `Station_Name`

PASSENGER

`Passenger_Id`-> (`First_Name` `Last_Name` `Gender` `Phone_No` `Ticket_No` `Age` `Class`)

ACCOUNT

`Username`->(`Password` `Email_Id` `Address`)

FIRST NORMAL FORM:

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

The above schema is in 1NF since all the attributes are atomic and not multivalued.

Since a passenger could have multiple phone numbers,it would violate the 1NF rules.Hence we have created a separate table called contact to handle this.

SECOND NORMAL FORM:

A table is said to be in 2NF if both the following conditions hold:

-Table is in 1NF (First normal form)

-No non-prime attribute is dependent on the proper subset of any candidate key of table.

If in Passenger table we consider ticket_no and first_name as the candidate key,then date_of_birth would depend only on the name and it would violate the 2NF.

THIRD NORMAL FORM:

A table design is said to be in 3NF if both the following conditions hold:

-Table must be in 2NF

-Transitive functional dependency of non-prime attribute on any super key should be removed.

Our schema follows the above rules and hence is in 3NF.

DDL

```
CREATE DATABASE railways;
USE railways;
CREATE TABLE account
(
  `Username` VARCHAR(50) PRIMARY KEY ,
  `Password` VARCHAR(10) NOT NULL,
  `Email_Id` VARCHAR(40) NOT NULL,
  `Address`VARCHAR(60) DEFAULT NULL
);
INSERT INTO account
VALUES('Priyanshu01',12345678,'pm01@gmail.com','Old airport road,bangalore'),
      ('Mangal02',34578321,'mp02@gmail.com','Jaipur'),
      ('Som2003',789564321,'som2003@gmail.com','Dholpur'),
      ('Dishu450',543789034,'dishu.dm@gmail.com','Agra'),
      ('Aditya685',789654098,'adi1820@gmail.com','Morena'),
      ('Pawan123',765439087,'pawan45@gmail.com','Bharatpur'),
      ('Ajay34567',890675476,'tyagi390@gmail.com','Gwalior'),
      ('Dhairya237',908765442,'soni678@gmail.com','Indore'),
      ('Shourya896',907654876,'agwl567@gmail.com','Ahmedabad'),
      ('Samyak452',674590875,'kaloo789@gmail.com','Kolkata');

CREATE TABLE contact
(
  Username VARCHAR(50),FOREIGN KEY(username) REFERENCES account(username),
  Phone_No CHAR(10) NOT NULL
);
INSERT INTO contact
VALUES('Priyanshu01',8949967310),
      ('Mangal02',7452885312),
      ('Som2003',9352874214),
```

```
('Dishu450',9530404364),
('Aditya685',7727817094),
('Pawan123',8502979140),
('Ajay34567',9119177656),
('Dhairya237',8955386632),
('Shourya896',8306198068),
('Samyak452',8432056756);
```

CREATE TABLE passenger

```
(
`Passenger_Id` INT NOT NULL AUTO_INCREMENT,
`First_Name` VARCHAR(30) NOT NULL,
`Last_Name` VARCHAR(30) NOT NULL,
`Gender` VARCHAR(10) NOT NULL,
`Phone_No` CHAR(10) DEFAULT NULL,
`Ticket_No` INT NOT NULL,FOREIGN KEY (`Ticket_No`) REFERENCES `Ticket`
(`Ticket_No`) ON DELETE CASCADE,
`Age` INT NOT NULL,
`Class` VARCHAR(20) NOT NULL,
PRIMARY KEY (`Passenger_Id`),
KEY `Ticket_No` (`Ticket_No`)
);
```

CREATE TABLE station

```
(
`Station_Code` CHAR(10) NOT NULL PRIMARY KEY,
`Station_Name` VARCHAR(50) NOT NULL
);
INSERT INTO station
VALUES ('JP','JAIPUR JUNCTION'),
('DHO','DHOLPUR JUNCTION'),
('AGC','AGRA CANTT'),
('GWL','GWALIOR JUNCTION'),
('MTJ','MATHURA JUNCTION'),
('NDLS','NEW DELHI'),
```

```
('NZM','HAZRAT NIZAMUDDIN'),
('CNB','KANPUR CENTRAL'),
('VGLJ','VIRANGANA LAKSHMIBAI JHANSI'),
('BPL','BHOPAL JUNCTION');
```

CREATE TABLE train

```
(
  `Train_No` INT NOT NULL PRIMARY KEY ,
  `Name` VARCHAR(40) NOT NULL,
  `Second_Sitting` VARCHAR(1500) NOT NULL,
  `Seat_Sleeper` VARCHAR(1500) NOT NULL,
  `Seat_First_Class_AC` VARCHAR(1500)  NOT NULL,
  `Seat_Second_Class_AC` VARCHAR(1500)  NOT NULL,
  `Seat_Third_Class_AC` VARCHAR(1500) NOT NULL,
  `Executive_Chair_Car` VARCHAR(1500) NOT NULL,
  `Ac_Chair_Car` VARCHAR(1500)  NOT NULL,
  `Wifi` CHAR(10) NOT NULL,
  `Food` CHAR(10) NOT NULL,
  `Run_On_Sunday` CHAR(10) NOT NULL,
  `Run_On_Monday` CHAR(10) NOT NULL,
  `Run_On_Tuesday` CHAR(10) NOT NULL,
  `Run_On_Wednesday` CHAR(10) NOT NULL,
  `Run_On_Thursday`CHAR(10)NOT NULL,
  `Run_On_Friday` CHAR(10) NOT NULL,
  `Run_On_Saturday` CHAR(10) NOT NULL
 );
 INSERT INTO train
 VALUES (12001,'HBJ SHATABDI EXP','N','N','N','N','N',112,1092,'Y','Y','Y','Y','Y','Y','Y','Y','Y'),
       (12279,'VGLJ NDLS TAJ EXP',1484,'N','N','N','N','N',234,'N','N','Y','Y','Y','Y','Y','Y','Y'),
       (15046,'GKP OKHA EXP','N',560,108,216,216,'N','N','N','Y','Y','Y','Y','N','N','N','N'),
       (16317,'CAPE JAT HIMSAGAR
EXP','N',480,'N',360,216,'N','N','N','Y','Y','Y','N','N','N','Y','Y'),
       (12137,'CSMT FZR PUNJAB
MAIL','N',480,138,216,216,'N','N','N','Y','Y','Y','Y','Y','Y','Y','Y'),
```

(19665,'KURJ UDZ EXP','N',400,130,216,360,'N','N','N','Y','Y','Y','Y','Y','Y','Y','Y');

CREATE TABLE ticket

```
(
  `Ticket_No` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `Train_No` INT NOT NULL,FOREIGN KEY (Train_No) REFERENCES train (Train_No) ON UPDATE CASCADE,
  `Date_of_Journey` DATE NOT NULL,
  `Username` VARCHAR(50) NOT NULL,FOREIGN KEY (Username) REFERENCES account (Username) ON DELETE CASCADE
);
```

CREATE TABLE stoppage

```
(
  `Train_No` INT NOT NULL,FOREIGN KEY (Train_No) REFERENCES train (Train_No) ON DELETE CASCADE ON UPDATE CASCADE,
  `Station_Code` CHAR(10) NOT NULL ,FOREIGN KEY (Station_Code) REFERENCES station (Station_Code) ON DELETE CASCADE ON UPDATE CASCADE,
  `Arrival_Time` TIME DEFAULT NULL,
  `Departure_Time` TIME DEFAULT NULL
);
 -- alter table stoppage ADD CHECK (EXTRACT(HOUR FROM Arrival_Time) <24 AND EXTRACT(HOUR FROM Departure_Time) <24);
 INSERT INTO stoppage
 VALUES(12001,'BPL','15:27:00','15:30:00'),
      (12001,'VGLJ','18:37:00','18:42:00'),
      (12001,'GWL','19:40:00','19:45:00'),
            (12001,'AGC','21:20:00','21:25:00'),
      (12279,'VGLJ',NULL,'15:20:00'),
      (12279,'GWL','16:33:00','16:35:00'),
      (12279,'MTJ','19:00:00','19:02:00'),
      (12279,'NDLS','21:35:00',NULL),
      (12137,'DHO','17:20:00','17:22:00'),
      (12137,'NZM','20:55:00','20:57:00'),
      (15046,'AGC','6:35:00','6:40:00'),
```

```
(15046,'CNB','11:30:00','11:35:00'),
(16317,'BPL','9:05:00','9:10:00'),
(16317,'DHO','16:10:00','16:12:00'),
(16317,'NZM','20:16:00','20:21:00'),
(19665,'VGLJ','13:40:00','13:48:00'),
(19665,'JP','22:30:00','22:40:00');
alter table stoppage ADD CHECK (EXTRACT(HOUR FROM Arrival_Time) <24 AND
EXTRACT(HOUR FROM Departure_Time) <24);
INSERT INTO account
VALUES ('admin','admin@1234','adm19@gmail.com','Old airport road,bangalore');
SELECT a.Train_No
FROM stoppage as a join stoppage as b on a.Train_No = b.Train_No
WHERE a.Station_Code = "VGLJ" and b.Station_Code = "NDLS";
DROP TRIGGER cancellation;
drop table passenger;
INSERT INTO passenger
VALUES ('1','priyanshu','mangal','male','8949967310','1','22','first class ac');
```

Triggers

A trigger has been created which is invoked each time a ticket is cancelled.The trigger helps in increasing the number of seats in a coach after cancellation.

```
delimiter //
create trigger cancellation
 before delete on ticket
 for each row
BEGIN
  set @trainno=old.train_no;
  set @ticketno=old.ticket_no;
  SET @class = (SELECT p.class
              FROM PASSENGER p
              WHERE p.ticket_no = @ticketno);
  if @class='first class ac' then
    UPDATE Train set Seat_First_Class_AC = Seat_First_Class_AC+1 WHERE Train_No =
@trainno ;
  elseif @class='sleeper' then
```

```
    UPDATE Train set Seat_Sleeper = Seat_Sleeper+1 WHERE Train_No = @trainno ;
  elseif @class='second class ac' then
    UPDATE Train set Seat_Second_Class_AC = Seat_Second_Class_AC+1 WHERE
Train_No = @trainno ;
  elseif @class='third class ac' then
    UPDATE Train set Third_Class_AC = Seat_Third_Class_AC+1 WHERE Train_No =
@trainno ;
  end if;
END//
delimiter ;
```

SQL Queries

/* Find total number of first class seats available on any train that reaches bangalore before 7pm on Monday .*/

```
CREATE VIEW  c(Station_code,Train_no,Arrival_Time)AS
SELECT Stoppage.Station_code,Train_no,Arrival_Time
FROM Station INNER JOIN Stoppage ON
station.Station_code=Stoppage.Station_code
WHERE station.Station_name='VIRANGANA LAKSHMIBAI JHANSI';

SELECT *
FROM a;

CREATE VIEW  b(Station_code,Train_no,Arrival_Time)AS
SELECT Station_code,Train_no,Arrival_Time
FROM a
WHERE EXTRACT(HOUR FROM Arrival_Time)<19;

SELECT *
FROM b;

CREATE VIEW
```

```
c(Station_code,Train_no,Arrival_Time,First_Class_seats,Run_on_monday)AS
SELECT
Station_code,train.Train_no,Arrival_Time,Seat_First_Class_AC,Run_on_monday
FROM train INNER JOIN b ON train.Train_No=b.Train_No
WHERE train.Run_On_Monday='Y' AND train.Seat_First_Class_AC >0;

SELECT *
FROM c;

SELECT SUM(First_class_seats)
FROM c;
```

/* Find the time at which last train leaves New delhi station */

```
Create View
f(Departure_time)AS
SELECT Departure_time
FROM Stoppage
WHERE Station_Code IN
(SELECT Station_code
FROM station
WHERE Station_Name='New Delhi') ;
SELECT *
 FROM f;
 SELECT MAX(Departure_time)
 FROM f;
```

/* Find the phone number of the user whose email id is ajitesh@pes.edu */

```
 SELECT phone_no
 FROM contact
 WHERE
 username IN (SELECT username FROM account WHERE email_id='pm01@gmail.com);
```

Conclusion and Future Scope

Our system can successfully give information on any train, find trains running between two stations, book tickets and cancel tickets. This system could be used for official train booking. However, several other features could be added like booking meals on trains etc. Also payment gateways have to be implemented to make sure the transactions happen securely.