DBMS Project Report

PES University

Database Management Systems

UE18CS252

Submitted By

PES2201800681 AJITESH NAIR

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.

Introduction

Following are the entities in our miniworld 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

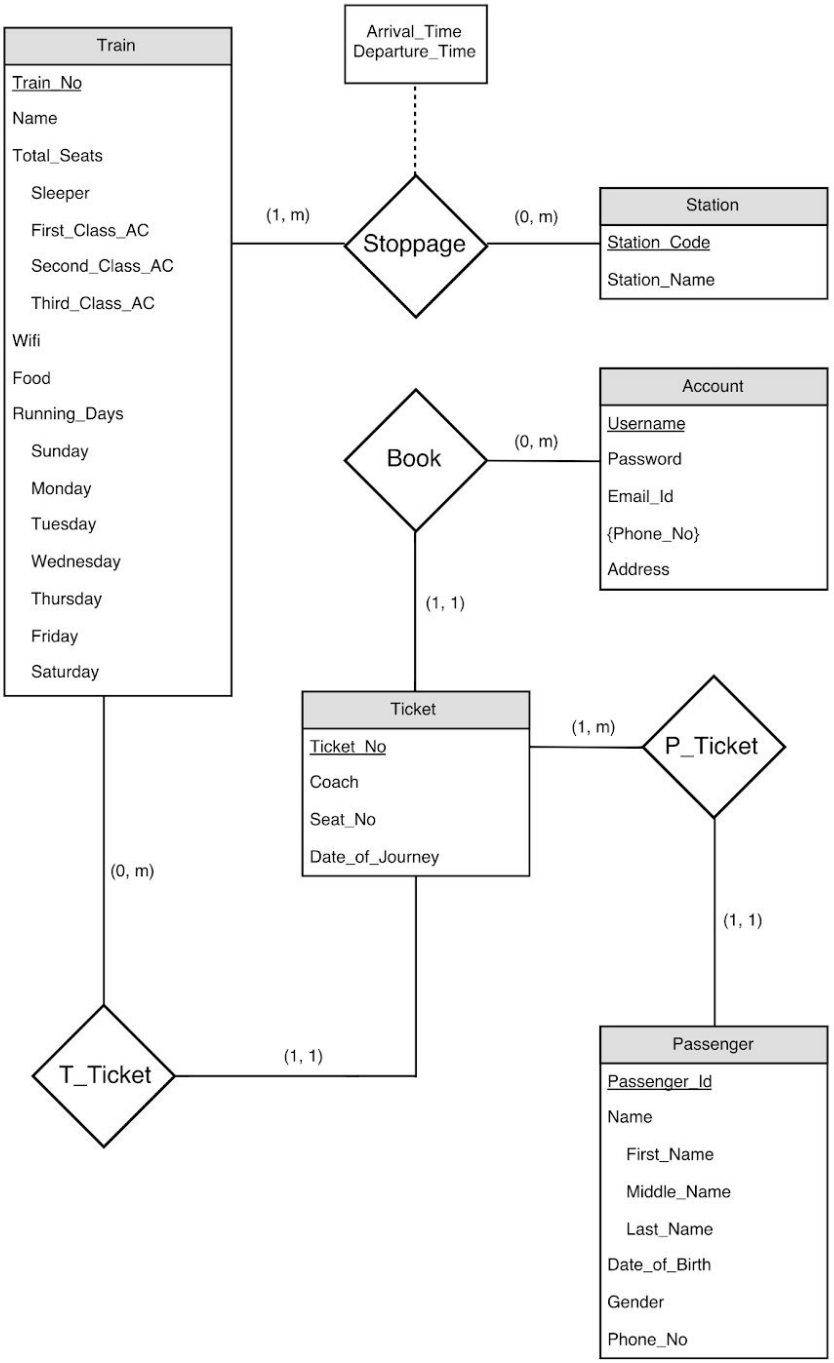
```
`Train_No` int(6) NOT NULL DEFAULT '0',
`Station_Code` char(5) NOT NULL DEFAULT '',
`Arrival_Time` time DEFAULT NULL,
`Departure_Time` time DEFAULT NULL,
PRIMARY KEY (`Train_No`,`Station_Code`),
KEY `Station_Code` (`Station_Code`),
CONSTRAINT `Stoppage_ibfk_1` FOREIGN KEY (`Train_No`) REFERENCES `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
)
```

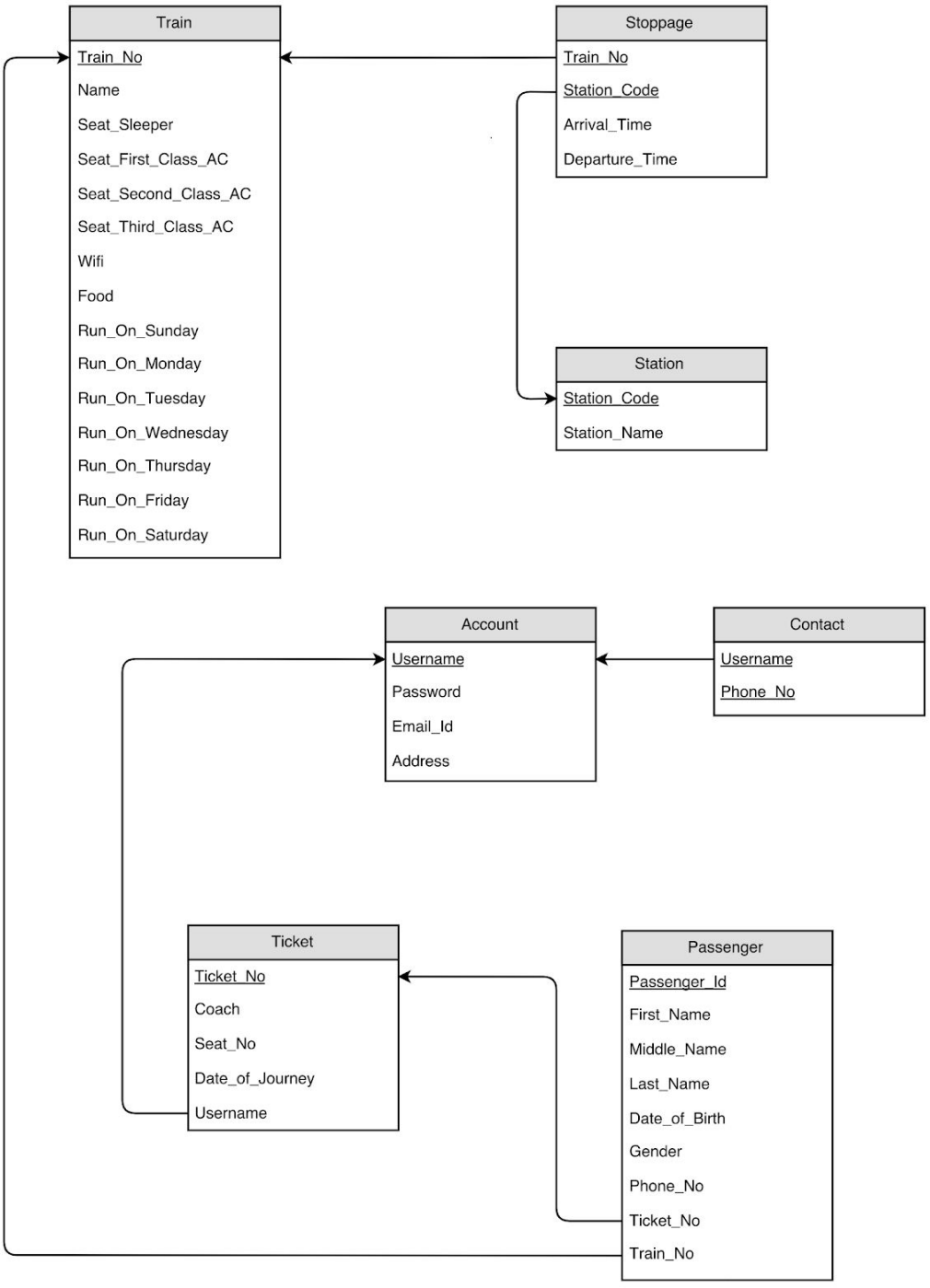
Ticket

```
{ `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,
 CONSTRAINT `Ticket_ibfk_2` FOREIGN KEY (`Train_No`) REFERENCES `Train`
(`Train_No`) ON UPDATE 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`)
           )
```

Data 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



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 PROJECT;
USE PROJECT;

CREATE TABLE `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`)
);
CREATE TABLE `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
) ;
CREATE TABLE `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
);

CREATE TABLE `Station` (
  `Station_Code` char(5) NOT NULL DEFAULT '',
  `Station_Name` varchar(25) NOT NULL,
  PRIMARY KEY (`Station_Code`)
```

```
);
CREATE TABLE `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
);

alter table Stoppage ADD CHECK (EXTRACT(HOUR FROM Arrival_Time) <24 AND
EXTRACT(HOUR FROM Departure_Time) <24);

CREATE TABLE `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,
  CONSTRAINT `Ticket_ibfk_2` FOREIGN KEY (`Train_No`) REFERENCES `Train`
(`Train_No`) ON UPDATE CASCADE
);
CREATE TABLE `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`)
);
```

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 a(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='BANGALORE';
```

```
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='ajitesh@pes.edu');
```

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.