A PROJECT REPORT ON
"RAILWAY MANAGEMENT SYSTEM"

COURSE:
IT 214 : DATABASE MANAGEMENT SYSTEMS

UNDER THE GUIDANCE OF
PROF. P.M. JAT

Team Details

Name	ID
Ujaval Bhatt	201501403
Abhin Kakkad	201501419
Manthan Mehta	201501420
Smit Thakkar	201501440

1 Scope of Database

- This database consists of all the major functionalities required for having a working railway management portal.
- It includes all the necessary thing from railway transactions like reservation, cancellation, scheduling and ticketing, routing, employee information and customer details.

2 Description

The current database implements the basic functionalities of the railway portal of India i.e. IRCTC. There are around 12,000 trains in the Indian railway network and over 8.000 railway stations. It employs over a million people and carries 23 million passengers on a daily basis. We have tried to emulate a small part of this vast network in the project.

2.1 Entities

The current railway management system has the following entities:

- Trains
- Routes
- Stations
- Reservation
- User
- Passenger

2.2 Functionalities

The functionalities that are provided are as follows:

- Register as a user of the portal after providing certain details.
- Reserve a seat/s as a passenger after providing certain details.
- Do cancellation of a reserved seat.
- See trains between a pair of stations, check seat availability and get fare details.
- As a “Master” user the administrator is allowed to add trains and stations. Also he is allowed to view employee details of any employee.

3 Entity Relationship Diagram

3.1 List of Entities

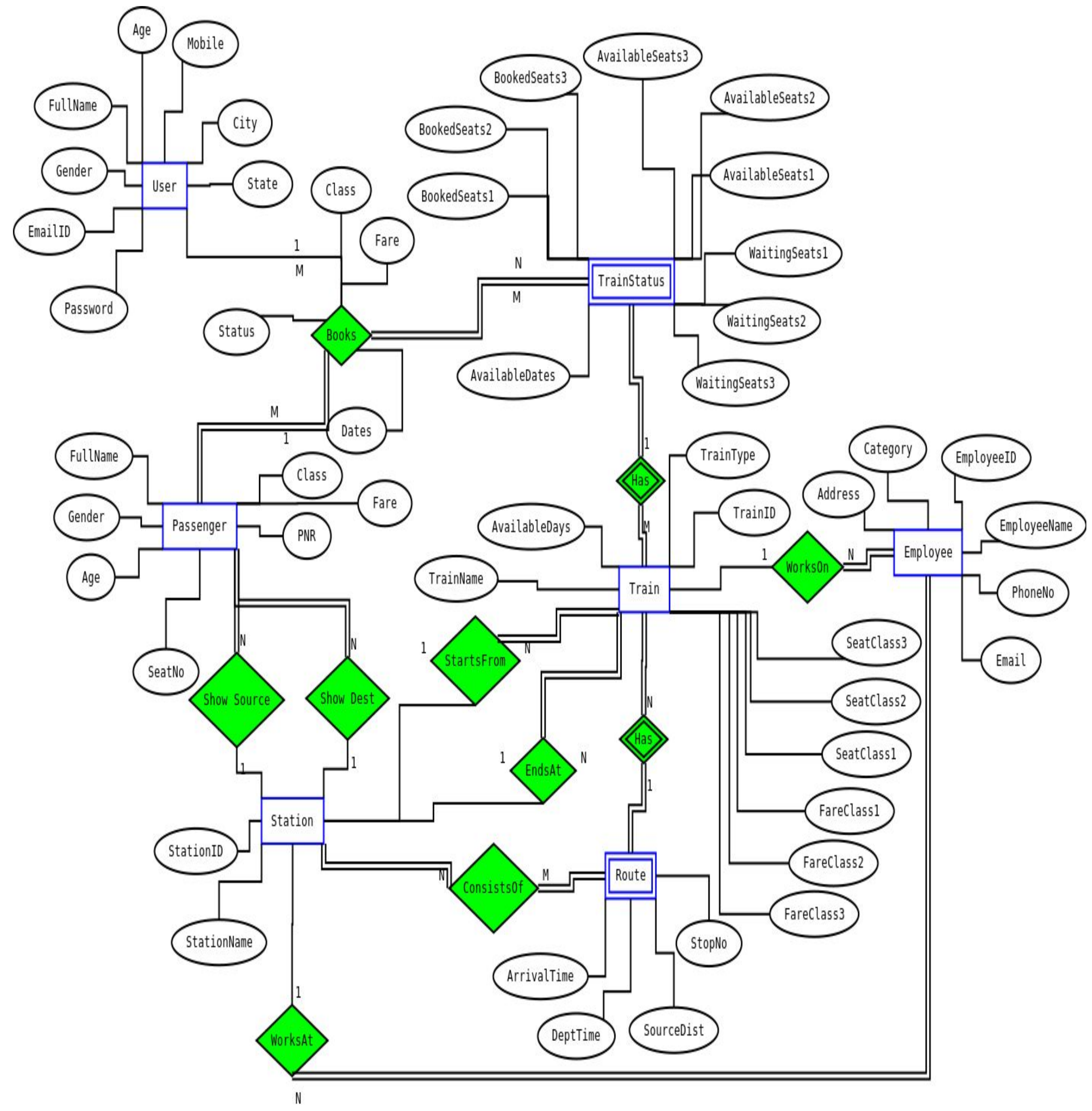
1.	Trains	a) TrainID b) TrainNumber c) Train Type d) SeatsClass1 e) SeatsClass2 f) SeatsClass3 g) FareClass1 h) FareClass2 i) FareClass3 j) SourceID k) DestinationID
2.	Route	a) TrainID b) StopNumber c) StationID d) Source Distance e) Arrival Time f) Departure Time
3.	Station	a) StationID b) Station Name
4.	Train Status	a) TrainID b) Available Dates c) BookedSeats1 d) BookedSeats2 e) BookedSeats3 f) AvailableSeats1 g) AvailableSeats2

		h) AvailableSeats3 i) WaitingSeats1 j) WaitingSeats2 k) WaitingSeats3
5.	User	a) Email ID b) Full Name c) Password d) Age e) Gender f) Mobile No g) City h) State
6.	Passenger	a) PNR b) PName c) Gender d) Age e) SeatNo f) Class g) Fare
7.	Employee	a) EID b) EName c) Category d) Train ID e) Station ID f) Mobile g) Email h) Address

3.2 List of Relations

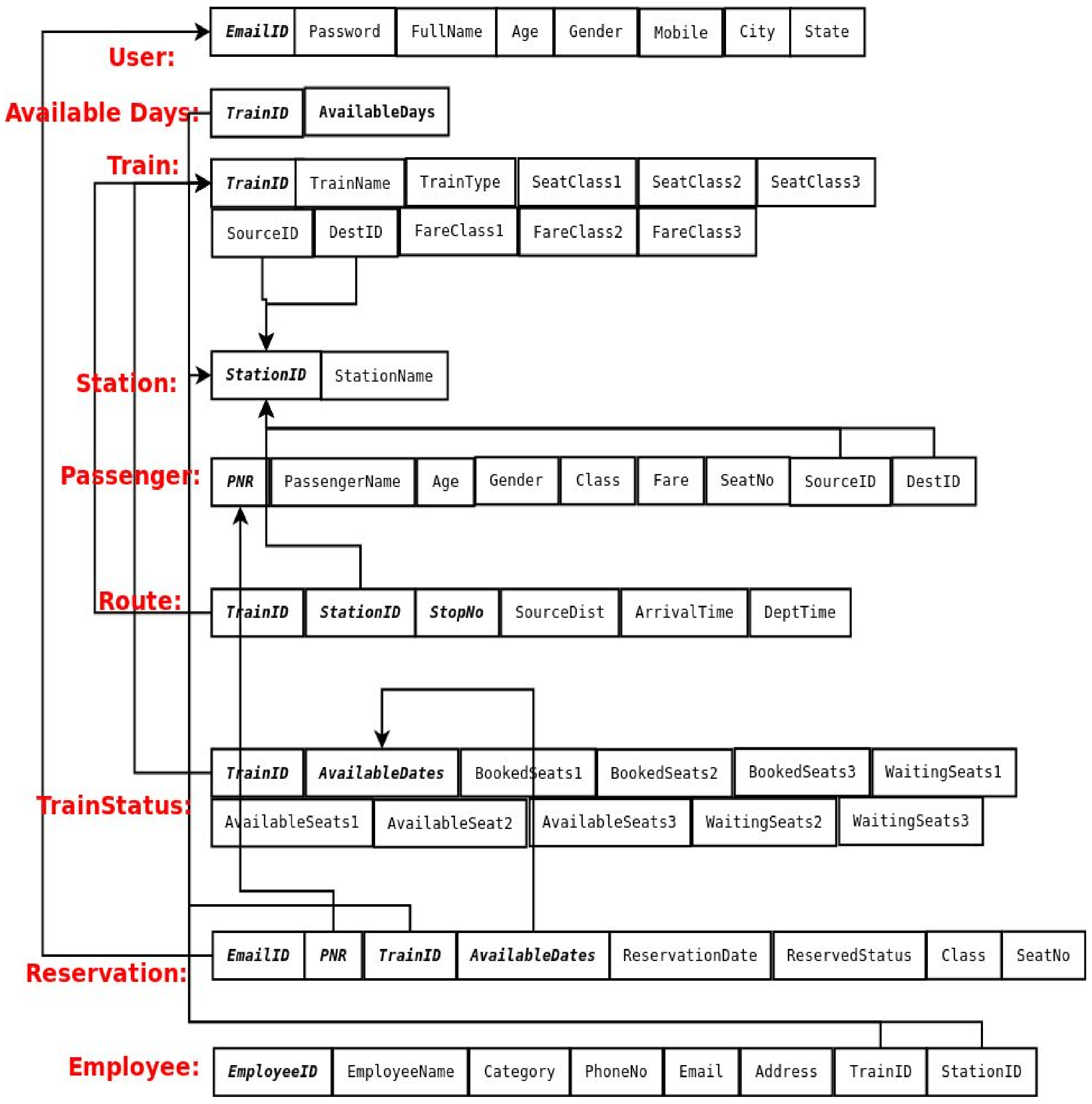
Number	Relationship Type	Entities Involved
1.	Books	User, Passenger, Train Status
2.	Starts From	Train, Station
3.	Ends At	Train, Station
4.	Show Source	Passenger, Station
5.	Show Destination	Passenger, Station
6.	Works On	Employee, Train
7.	Works At	Employee, Station
8.	Consists Of	Route, Station

3.3 ER Diagram



4 Relational Schema

4.1 Schema Diagram



*Entities in **bold** are keys for that particular table.

4.2 Functional Dependencies

1) User:

Email id → Full name
Email id → Password
Email id → Age
Email id → Gender
Email id → Mobile
Email id → City
Email id → State

2) Train:

Train id → Train Name
Train id → Train Type
Train id → SeatsClass1
Train id → SeatsClass2
Train id → SeatsClass3
Train id → FareClass1
Train id → FareClass2
Train id → FareClass3
Train id → AvailableDays
Train id → SorceID
Train id → DestinationID

3) Employee:

EID → EName
EID → Category
EID → TrainID
EID → StationID
EID → Phoneno
EID → EmailID
EID → Address

4) Train Status:

{TrainID,AvailableDays} → Bookedseats1
{TrainID,AvailableDays} → Bookedseats2
{TrainID,AvailableDays} → Bookedseats3
{TrainID,AvailableDays} → Waitingseats1
{TrainID,AvailableDays} → Waitingseats2
{TrainID,AvailableDays} → Waitingseats3
{TrainID,AvailableDays} → Availableseats1
{TrainID,AvailableDays} → Availableseats2
{TrainID,AvailableDays} → Availableseats3

5) Passenger:

PNR → PName
PNR → Gender
PNR → Age
PNR → SeatNo
PNR → Class
PNR → Fare
PNR → SourceID
PNR → DestinationID

6) Route:

{TrainId,Stopnumber} → StationID
{TrainId,Stopnumber} → SorceDestination
{TrainId,Stopnumber} → ArrivalTime
{TrainId,Stopnumber} → DepartureTime

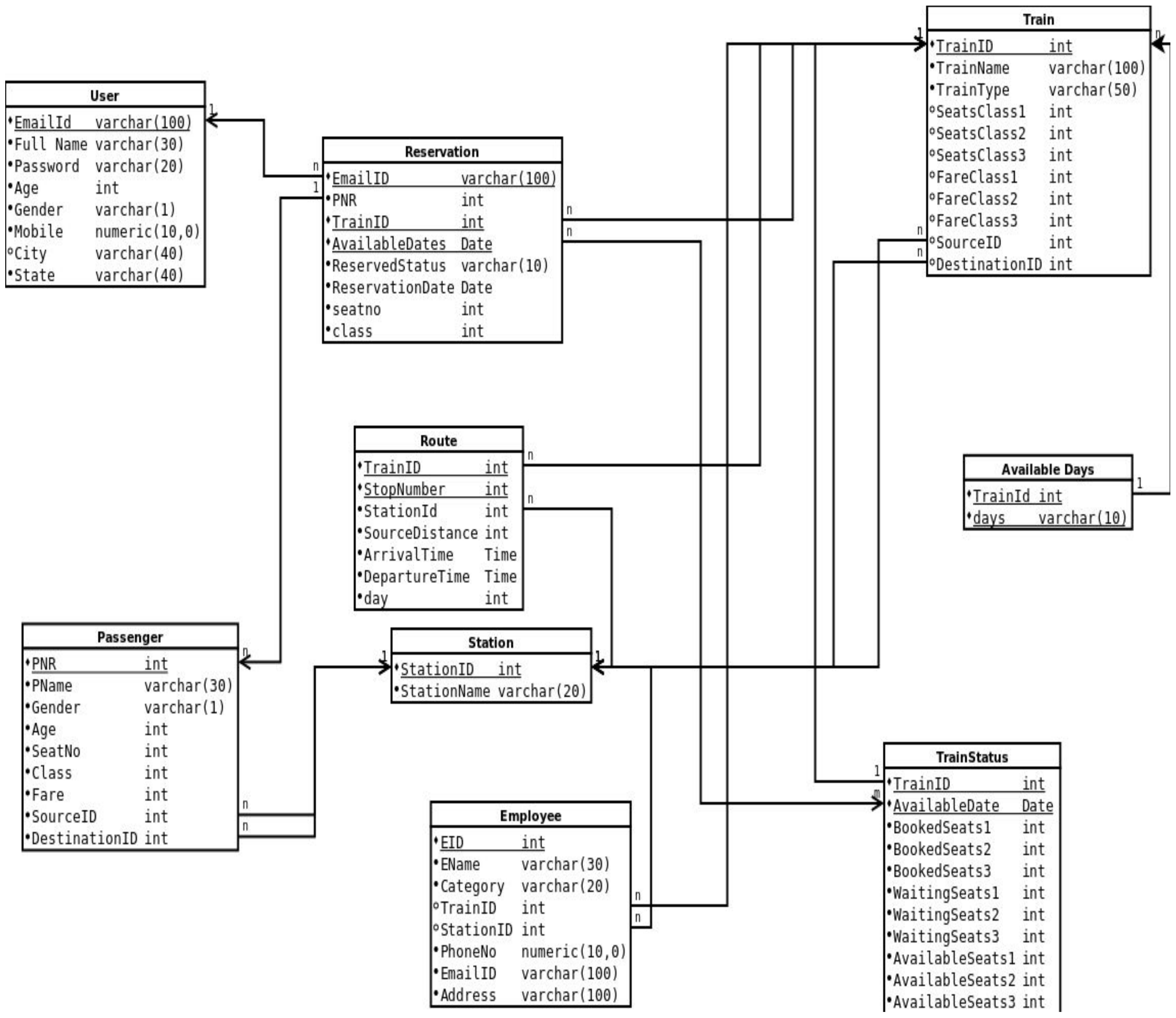
7) Station:

StationID → StationName

8) Reservation:

{EmailID,TrainID,Availableseats} → PNR
{EmailID,TrainID,Availableseats} → Reservedstatus
{EmailID,TrainID,Availableseats} → ReservationDate

4.3 RDBMS


User
EmailId varchar(100)
Full Name varchar(30)
Password varchar(20)
Age int
Gender varchar(1)
Mobile numeric(10,0)
City varchar(40)
State varchar(40)
Reservation
EmailID varchar(100)
PNR int
TrainID int
AvailableDates Date
ReservedStatus varchar(10)
ReservationDate Date
seatno int
class int
Train
TrainID int
TrainName varchar(100)
TrainType varchar(50)
SeatsClass1 int
SeatsClass2 int
SeatsClass3 int
FareClass1 int
FareClass2 int
FareClass3 int
SourceID int
DestinationID int
Route
TrainID int
StopNumber int
StationId int
SourceDistance int
ArrivalTime Time
DepartureTime Time
day int
Available Days
TrainId int
days varchar(10)
Passenger
PNR int
PName varchar(30)
Gender varchar(1)
Age int
SeatNo int
Class int
Fare int
SourceID int
DestinationID int
Station
StationID int
StationName varchar(20)
Employee
EID int
EName varchar(30)
Category varchar(20)
TrainID int
StationID int
PhoneNo numeric(10,0)
EmailID varchar(100)
Address varchar(100)
TrainStatus
TrainID int
AvailableDate Date
BookedSeats1 int
BookedSeats2 int
BookedSeats3 int
WaitingSeats1 int
WaitingSeats2 int
WaitingSeats3 int
AvailableSeats1 int
AvailableSeats2 int
AvailableSeats3 int
1
n
1
n
n
n
1
n
n
n
1
n
n
1
1
n
n
1
m
n
n

5 Queries

5.1 DDL Statements

DDL Statements:

CREATE SCHEMA railways;

SET SEARCH_PATH TO railways;

Station

```
create Table Station
(
StationID int not null,
StationName varchar(50) not null,
primary key (StationID)
)
```

Train

```
create Table Train
(
TrainID int not null,
TrainName varchar(100) not null,
TrainType varchar(50) not null,
SeatsClass1 int,
SeatsClass2 int,
SeatsClass3 int,
FareClass1 int,
FareClass2 int,
FareClass3 int,
SourceID int null,
DestID int null,
primary key (TrainID),
foreign key (SourceID) references Station(StationID) on update cascade on
delete cascade,
foreign key (DestID) references Station(StationID) on update no action on
delete no action
)
```

AvailableDays

```
create Table AvailableDays
(
TrainID int not null,
Days varchar(10) not null,
primary key (TrainID, Days)
)
```

TrainStatus

```
create Table TrainStatus
(
TrainID int not null,
AvailableDate date not null,
BookedSeats1 int,
BookedSeats2 int,
BookedSeats3 int,
WaitingSeats1 int,
WaitingSeats2 int,
WaitingSeats3 int,
AvailableSeats1 int,
AvailableSeats2 int,
AvailableSeats3 int,
primary key (TrainID, AvailableDate),
foreign key (TrainID) references Train(TrainID) on update cascade on delete
cascade )
```

Route

```
create Table Route
(
TrainID int not null,
StopNumber int not null,
StationID int not null,
SourceDist int not null,
ArrTime time,
DeptTime time,
Day int not null,
primary key(TrainID,StopNumber,StationID),
foreign key (TrainID) references Train(TrainID) on update cascade on delete
cascade,
foreign key (StationID) references Station(StationID) on update cascade on
delete cascade
)
```

User

```
create Table "User"
(
EmailID varchar(100) not null,
UName varchar(30) not null,
Password varchar(20) not null,
Age int not null,
Gender varchar(1) not null,
MobileNo numeric(10,0) not null,
City varchar(50) not null,
State varchar(50) not null,
primary key (EmailID)
)
```

Passenger

```
create Table Passenger
(
PNR SERIAL not null,
PName varchar(30) not null,
Gender varchar(1) not null,
Age int not null,
SeatNo int not null,
Class int not null,
Fare int not null,
SourceID int not null,
DestID int not null,
primary key (PNR),
foreign key (SourceID) references Station(StationID) on update cascade on
delete cascade,
foreign key (DestID) references Station(StationID) on update no action on
delete no action
)
```

Reservation

```
create Table Reservation
(
EmailID varchar(100) not null,
PNR int not null,
TrainID int not null,
AvailableDate date not null,
ReservedStatus varchar(10) null,
ReservationDate date not null,
SeatNo int not null,
Class int not null,
primary key (EmailID, PNR, TrainID, AvailableDate),
foreign key (TrainID, AvailableDate) references TrainStatus(TrainID,
AvailableDate) on update cascade on delete cascade,
foreign key (EmailID) references "User"(EmailID) on update cascade on
delete cascade,
foreign key (PNR) references Passenger(PNR) on update cascade on delete
cascade
)
```

Employee

```
create Table Employee
(
EmpID int not null,
EmpName varchar(30) not null,
Category varchar(20) not null,
TrainID int null,
StationID int null,
PhoneNo numeric(10,0),
EmailID varchar(100),
Address varchar(100),
primary key(EmpID),
foreign key (TrainID) references Train(TrainID) on update cascade on delete
cascade,
foreign key (StationID) references Station(StationID) on update cascade on
delete cascade
)
```

5.2 Data Insertion Statements

Station

```
INSERT INTO railways.station(
            stationid, stationname)
    VALUES (?, ?);
```

Train

```
INSERT INTO railways.train(
            trainid, trainname, traintype, seatsclass1, seatsclass2,
seatsclass3,
            fareclass1, fareclass2, fareclass3, sourceid, destid)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```

AvailableDays

```
INSERT INTO railways.availabledays(
            trainid, days)
    VALUES (?, ?);
```

TrainStatus

```
INSERT INTO railways.trainstatus(
            trainid, availabledate, bookedseats1, bookedseats2,
bookedseats3,
            waitingseats1, waitingseats2, waitingseats3, availableseats1,
            availableseats2, availableseats3)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```

Route

```
INSERT INTO railways.route(
            trainid, stopnumber, stationid, sourcedist, arrtime, depttime,
            day)
    VALUES (?, ?, ?, ?, ?, ?, ?);
```

User

```
INSERT INTO railways."User"(
            emailid, uname, password, age, gender, mobileno, city, state)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?);
```

Passenger

```
INSERT INTO railways.passenger(
            pnr, pname, gender, age, seatno, class, fare, sourceid, destid)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);
```

Reservation

```
INSERT INTO railways.reservation(
            emailid, pnr, trainid, availabledate, reservedstatus,
reservationdate,
            seatno, class)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?);
```

Employee

```
INSERT INTO railways.employee(
            empid, empname, category, trainid, stationid, phoneno, emailid,
            address)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?);
```

*Individual data entries attached as SQL file.

5.3 Data Manipulation Statements

*All queries and stored procedure attached as text file.

5.4 Console Application

*Console application source code attached as text file.