Symbiosis Institute of Technology

A DBMS Project Report on

RAILWAY RESERVATION SYSTEM DATABASE (IRCTC)

Submitted by

Adhiraj Dev Goswami(18070124004)

Aditya Singh(18070124005)

Anvita Gupta(18070124013)

Under the Guidance of

Dr.Shruti Patil
(Assistant Professor,CS&IT Dept.)

Department of Computer Science

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Index

Introduction

Rail transport is one of the most important transport systems in India. It has played a very important role in our country's economy. With rapid development, the railway lines and passengers have been increasing every year in the country. With such a huge customer base, buying train tickets in a fast and efficient way is a very prominent problem. Taking the problem of offline ticketing to online on the internet has shown a huge increase in sales and also keeps it more organized. It is not only a technological innovation but also improves railway services. Online reservation has made the process very much easier. To keep the data of trains and customers organized we need a database to store all the information. Database also helps a lot in reducing manual errors involved.

Being more specific this, online reservation system can perform the basic functions like reservation .The users are required to register on the server for getting access to the database enquiry result retrieval upon registration completion each user has an account which is essentially referred to as the view level of the customer .The account contains comprehensive information of the user enter during the registration and allows the user to enquire about train schedule ,seat availability and make new reservations.

The objectives of the system are:

- To reduce paperwork
- Reduce operational time
- Increase accuracy and reliability

- Fast process
- Increase operational efficiency
- Data security

Problem statement

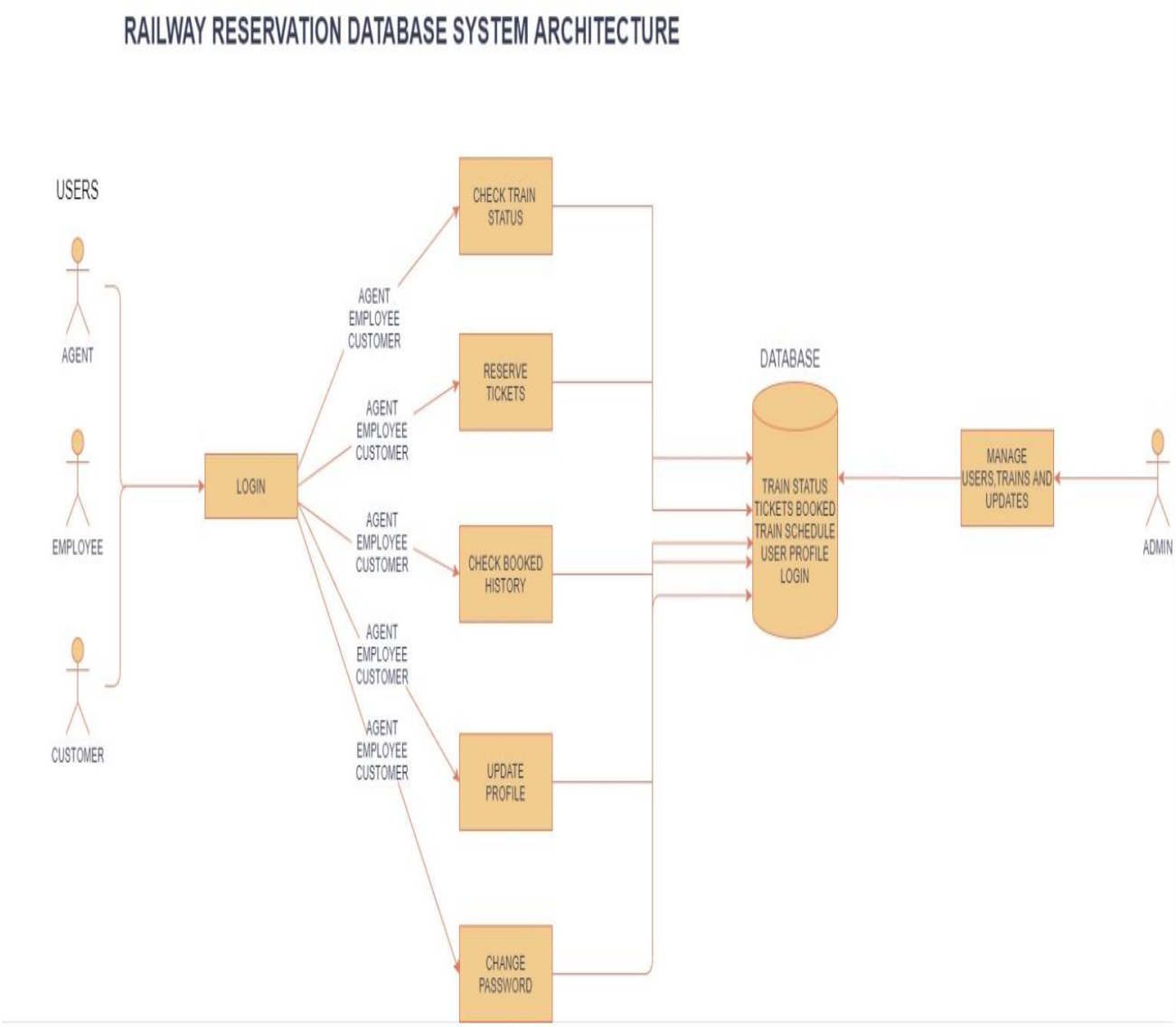
To prepare a database for **Online Railway Booking System** using MYSQL and execute queries and triggers on it.

Solution:

This database system is basically concerned with the reservation and handling of railway tickets. The need of the system arouse because it is the known fact that India is the largest railway network in the whole world and it is not possible to handle such a large system manually.By computerizing it ,it became possible to overcome the limitations and make the system operations more efficient. The complexity in handling data and records of such a vast system got reduced and became easier by computerizing the system.

The system facilitates the user to enquire about the trains available between the given source and destinations, booking of tickets, enquire about the status of trains and booked tickets etc. The aim is to design and develop a database maintaining the records of different trains, tickets, train status and passengers.

System Architecture:



System **architecture** is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. In this system, an admin manages train data and details. Admin also handles the user which can be a

customer, employee or an agent. Train runs on a particular schedule which contains the days running and departure and arrival times. Train runs on a route which consists of many stops and starting and ending stations. The user can register and login to enquire about train status which contains the available seats and waiting list. The user can book tickets after doing the payment. The user can download and print the ticket.

Modules:

The database will contain the following modules:

- Admin Module:

  This module is used to manage all the trains and their status

- Train Module:

  It is used to manage trains.

- Class Module:

  It is used to manage train classes.

- Train Status Module:

  It is used to manage availability of tickets.

- User Module:

  This module helps the customer or agent to log in and check or book tickets.

- Train Schedule Module:

  This module helps to know the schedule of the train.

- Station Module:

  This module helps to check via which station the train goes.

- Train Routes Module:

  This module helps the customer to check for the routes.

Functional Requirements:

- Admin manages User and Train details.
- Train has different classes.
- Train has a train status which consists of waiting and available seats.
- Train runs on a particular route and has many stops.
- Train runs on a schedule and on particular days.
- User can register for an account with username and password.
- User can enquire about trains available on a particular route, train schedule and ticket availability.
- Customer can reserve trains after doing the payment.
- Customer can print tickets and enquire about booking status if confirmed or waitlisted.
- Loyalty program for customers.
- Users can book meals via E-catering.
- E-wheelchair option will be provided to users.

ENTITIES ,ATTRIBUTES AND RELATIONSHIPS

ENTITIES AND ATTRIBUTES:

1. ADMIN

   Attributes: Admin ID, Admin name,Mobileno.,

   Admin_DOB,Address(Houseno,Streetname,City,ZIP Code).

2. USER

   Attributes: User ID, Password, DOB, Contactno.,

   Emailid,Age,Address(Houseno,Streetname,City,ZIP Code),Loyalty

   Program,Name(Fname,Lname),Gender,Aadhar ID.

   →EMPLOYEE

   Attributes: Designation

   →AGENT

   Attributes: GST No., Business Name

   →CUSTOMER

3. TRAIN SCHEDULE

   Attributes: Schedule ID, Start Time, End time, Duration, Days Running.

4. TRAIN ROUTES

   Attributes: RouteID , StopNo.,Arrival time,Departuretime,DayNo.

5. STATION

   Attributes: Station ID, Station Name.

6. TRAIN

   Attributes: Train no., Train name, Train type, ,Distance.

7. TRAIN STATUS

   Attributes: Status ID, Available seats (AV_AC,AV_SL), Waiting seats(WL_AC,WL_SL),Source ,Destination.

8. CLASS

   Attributes: Class ID, Sleeper, AC.

RELATIONSHIP WITH ATTRIBUTES:

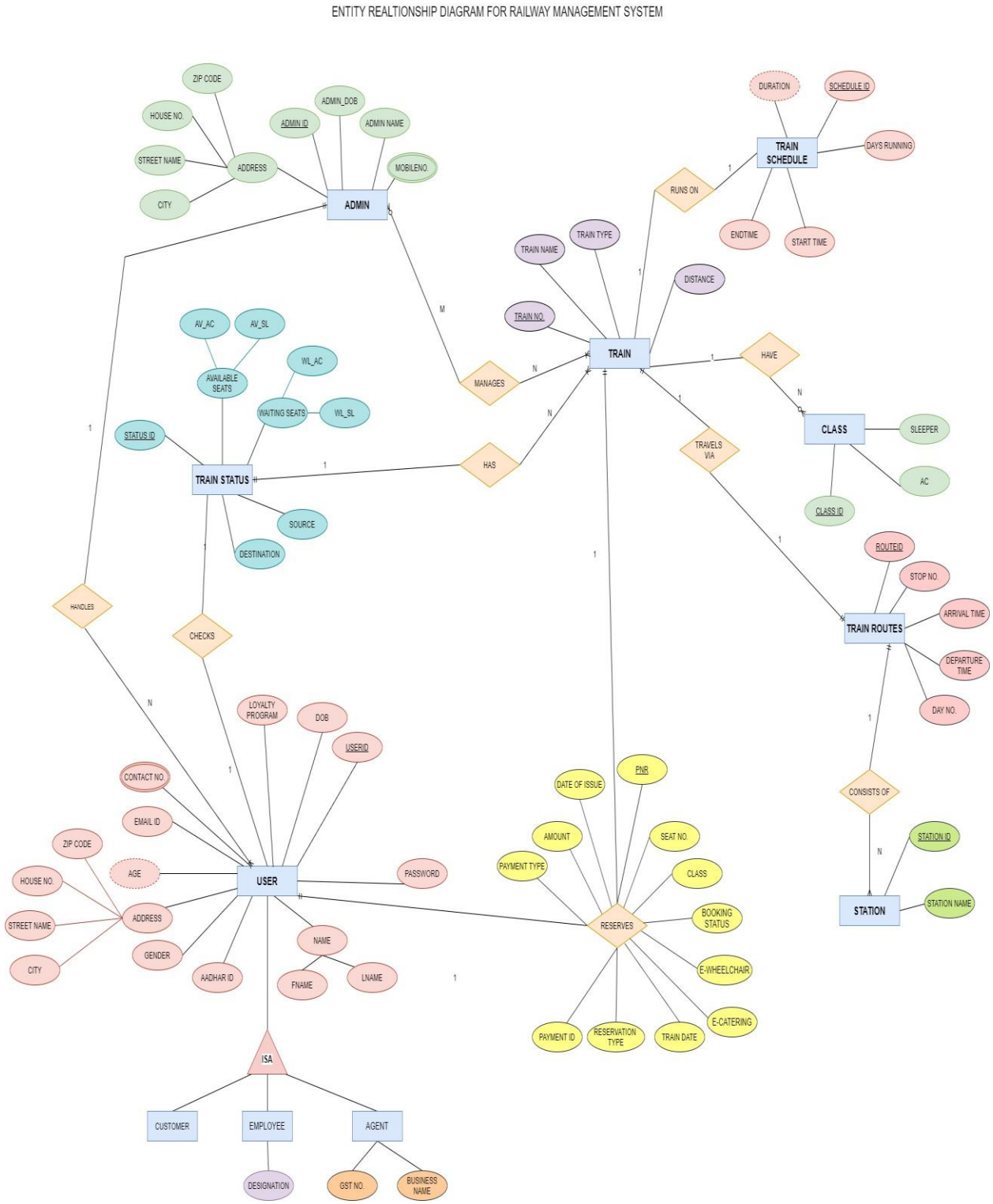
1. Reserves

Attributes: PNR, Seatno.,Class,BookingStatus,E-wheelchair,E catering,TrainDate,ReservationType,PaymentID,PaymentType,Amount, DateofIssue.

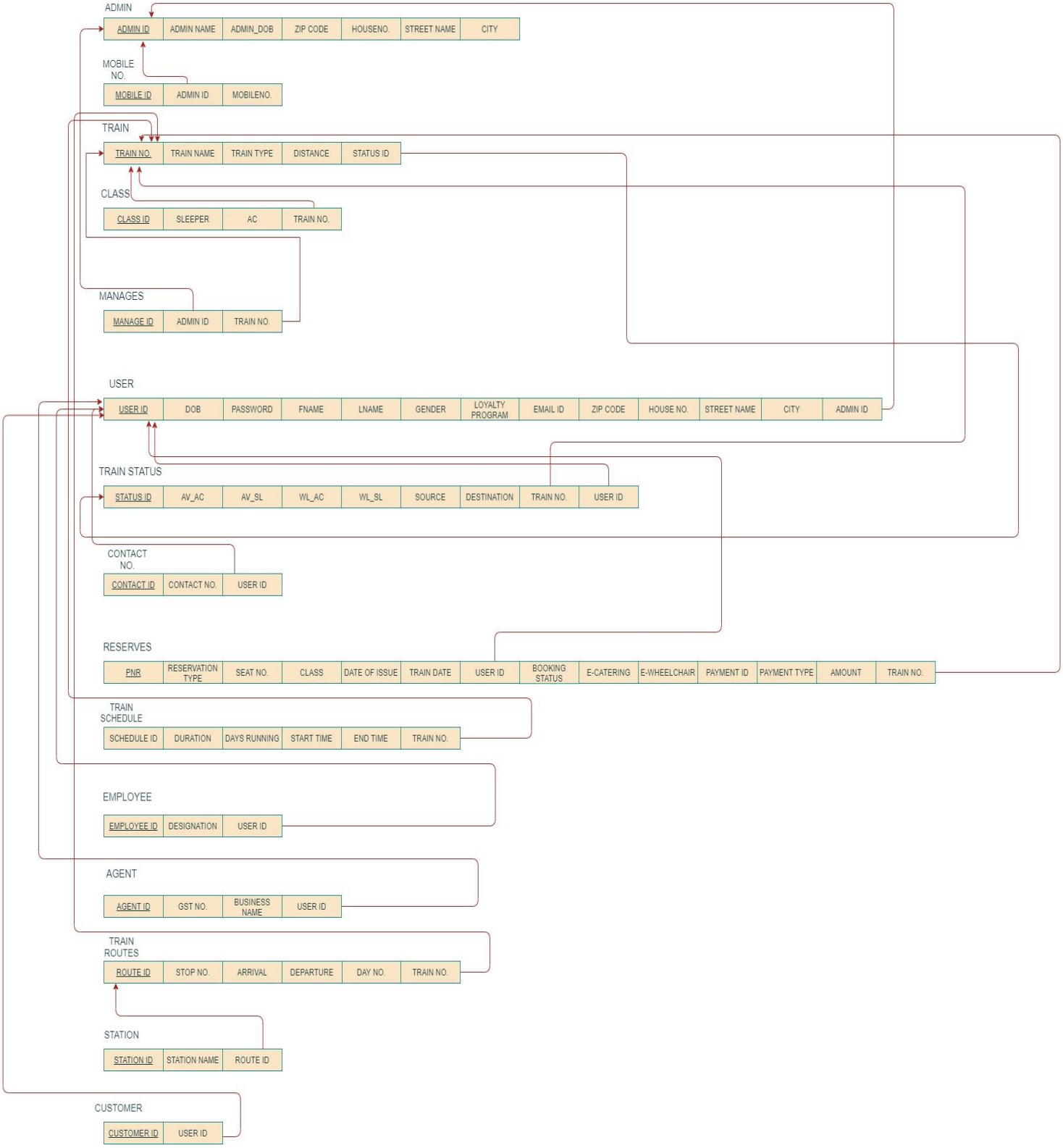
RELATIONSHIPS:

Relation	Cardinality
(Admin)**manages**(Train)	Many : Many
(Admin)**handles**(User)	1 : Many
(Train)**have**(Class)	1 : Many
(Train)**has**(Train status)	Many : 1
(Train)**travelsVia**(TrainRoutes)	1 : 1
(Train)**RunsOn**(TrainSchedule)	1 : 1
(User)**checks**(Train status)	1 : 1
(User)**reserves**(Train)	1 : 1
(User)**isa**(Agent)or(Customer)or(Employee)	Many : Many
(TrainRoutes)**consistsof**(Station)	1 : Many

ENTITY-RELATIONSHIP DIAGRAM



RELATIONAL SCHEMA



KEYS

1. **TRAIN**:

PRIMARY KEY	Train no.
CANDIDATE KEY	Train no., Train name
FOREIGN KEY	ADMINID
ALTERNATE KEY	Train name

2. **ADMIN**:

PRIMARY KEY	Admin ID
CANDIDATE KEY	Admin ID, AdminName+Address
ALTERNATE KEY	AdminName+Address

3. **TRAIN STATUS**:

PRIMARY KEY	Status ID
FOREIGN KEY	Train no.

4.**CLASS**:

PRIMARY KEY	Class ID
FOREIGN KEY	Train no.

5. **CUSTOMER/ EMPLOYEE**:

PRIMARY KEY	User ID
CANDIDATE KEY	User ID,AADHARID,Name+Address,EmailID

FOREIGN KEY	ADMINID
ALTERNATE KEY	AADHARID, Name+Address,EmailID

6. **AGENT:**

PRIMARY KEY	User ID
CANDIDATE KEY	User ID,AADHARID,Name+Address,EmailID,GSTNO.
FOREIGN KEY	ADMINID
ALTERNATE KEY	AADHARID, Name+Address,EmailID,GSTNO.

7.**TRAIN ROUTES**:

PRIMARY KEY	ROUTEID
FOREIGN KEY	Train no.

8.**STATION**:

PRIMARY KEY	StationID
CANDIDATE KEY	StationID,Station Name
FOREIGN KEY	RouteID,Train no.
ALTERNATE KEY	Station Name

9.**TRAIN SCHEDULE:**

PRIMARY KEY	ScheduleID
FOREIGN KEY	TrainNo.

CODD'S RULE

Rule 1: Information Rule

This rule requires all data in relational database management system (RDBMS) should be stored as values in tables at logical level. Some DBMS use Key-Value to store data, 'Redis' for example, which contradict the Information Rule, so these DBMS will not be regarded as relational DBMS. This rule is satisfied by all the databases.

This project will be implemented using MySQL. MySQL does store all data in the form of tables with values in columns of rows. Users can only access to values that are stored in tables. So, MySQL meets the requirement of rule 1.

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data. This rule refers to the primary key. It states that any data/column/attribute in the table should be able logically accessed by using the table in which it is stored, the primary key column of the table and the column which we want to access. When combination of these 3 is used, it should give the correct result. Any column/ cell value should not be directly accessed without specifying the table and primary key.So, our project fulfils the requirement of Rule2.

Rule 3: Systematic Treatment of NULL Values

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following − data is missing, data is not known, or data is not applicable. This rule states about handling the NULLs in the database. As database consists of various types of data, each cell will have different datatypes. If any of the cell value is unknown, or not applicable or missing, it cannot be represented as zero or empty. It will be always represented as NULL. This NULL should be acting irrespective of the data type used for the cell. When used in logical or arithmetical operation, it should result the value correctly.

This project fulfills this requirement by supporting NULL value and treats it in a systematic way. In MySQL, 'NULL' is supported and is regarded as missing data following ANSI/ODBC SQL standard. MySQL implements ternary logic. Users cannot compare values with NULL, even NULL with NULL by using '=', because NULL is missing data. The results of those compares are 'unknown'. MySQL provides 'IS NULL' and 'IS NOT NULL' statement in order to treat the compares with value 'NULL'.

Rule 5: Comprehensive Data Sublanguage Rule

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

MySQL follows the ANSI/ODBC SQL standard, yet there are several differences between them in several cases. The difference can be seen in documents of MySQL. All these differences are just about statement syntax. All database use in MySQL can be implemented by using SQL regardless of whether the syntax is different from standard SQL. So, MySQL fulfills Rule 5.

Rule 7: High-Level Insert, Update, and Delete Rule

A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records. This rule states that every query language used by the database should support INSERT, DELETE and UPDATE on the records.

It should also support set operations like UNION, UNION ALL, MINUS, INTERSECT and INTERSECT ALL. All these operation should not be restricted to single table or row at a time. It should be able to handle multiple tables and rows in its operation. This project is made in MYSQL. MYSQL supports insertion, updation and deletion.

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

MySQL can export one database by creating a 'backup' file. This file can be restored by MySQL on another computer. The physical underlying of this database has changed while the table structure will not be changed and users can access this restored one without any adjustment on

their queries. Therefore this rule is also satisfied.

Rule 10: Integrity Independence

Integrity constraints specific to a particular relational database must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.

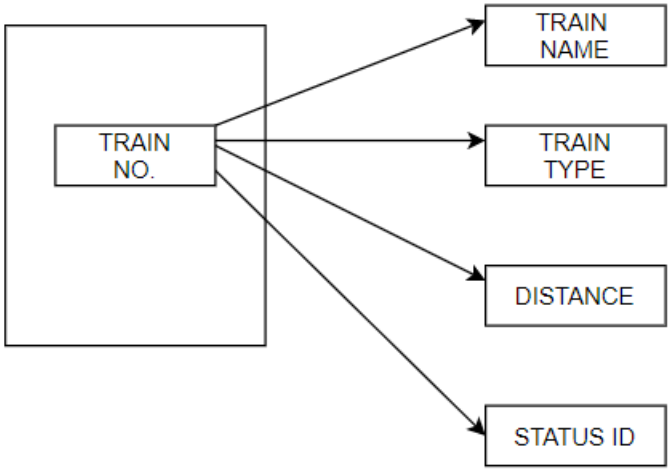
A minimum of the following two integrity constraints must be supported:

1. Entity integrity: No component of a primary key is allowed to have a null value. That is, no records can have NULL values in its Primary Key attribute.

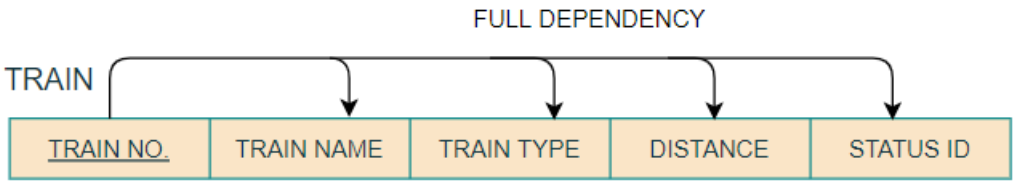
2. Relational integrity: For each distinct non-null foreign key value in a relational database, there must exist a matching primary key value from the same domain. In other words, if a foreign key cannot have null values as its component then it must refer a matching primary key value with the same set of permitted values to accept any new records.This project will have all non-null primary keys and it also follows relational integrity.

Functional Dependency and Anomalies

1. Train



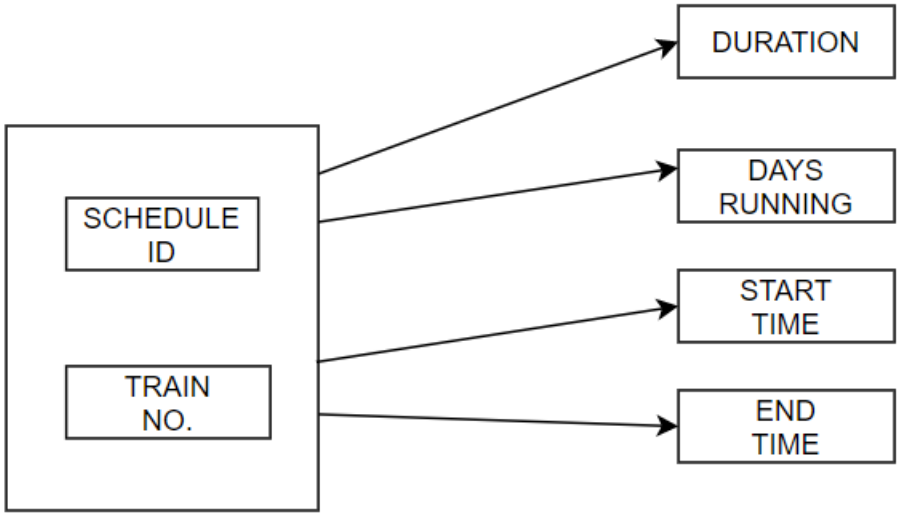
Functional Dependency Chart:



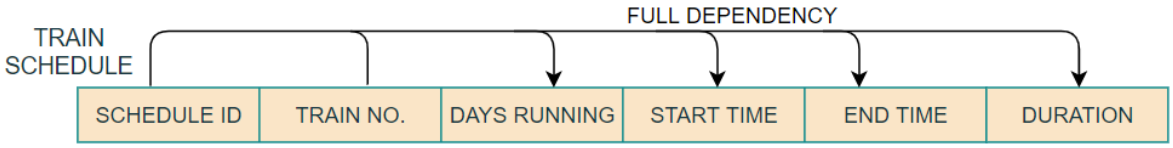
Anomalies:

Insertion	None
Deletion	None
Updation	None

2. Train Schedule



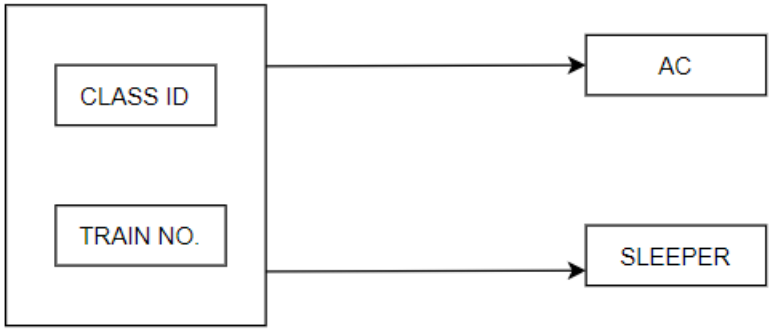
Functional Dependency Chart:



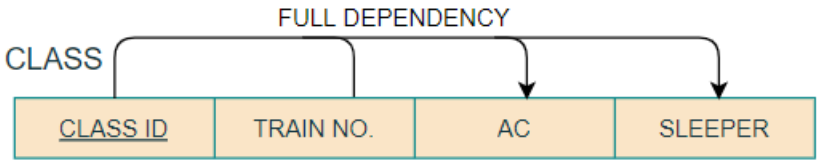
Anomalies:

Insertion	None
Deletion	None
Updation	None

3. Class



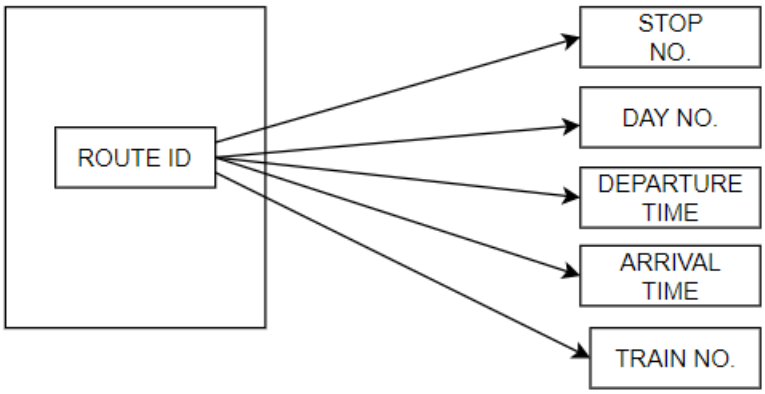
Functional Dependency Chart:



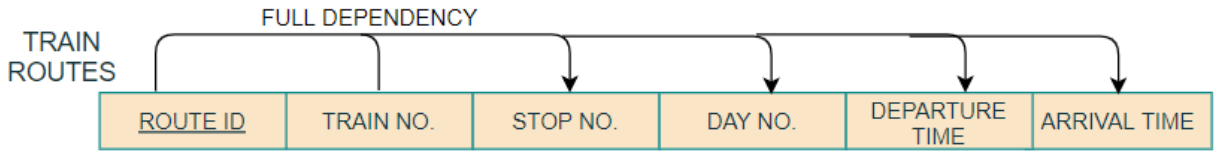
Anomalies:

Insertion	None
Deletion	None
Updation	None

4. Train Routes



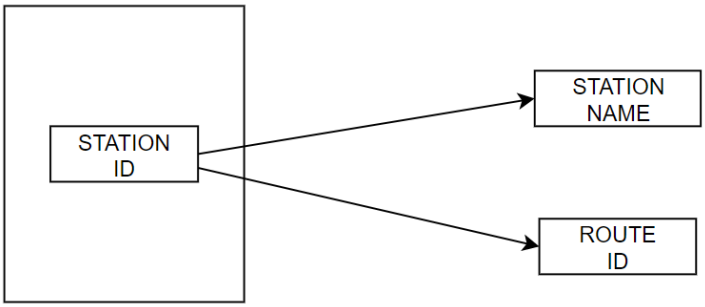
Functional Dependency Chart:



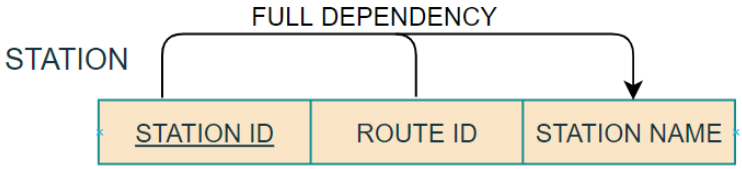
Anomalies:

Insertion	None
Deletion	None
Updation	None

5. Station



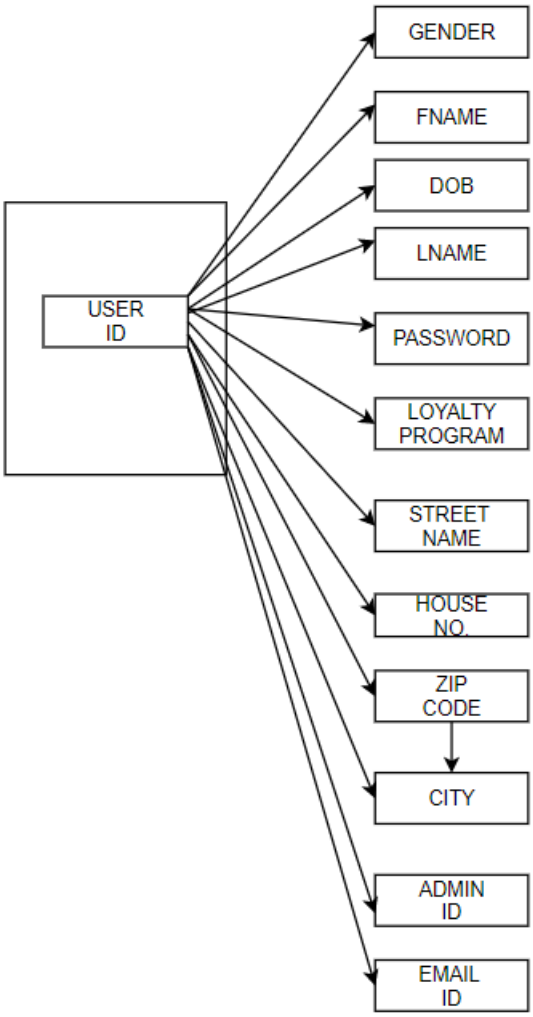
Functional Dependency Chart:

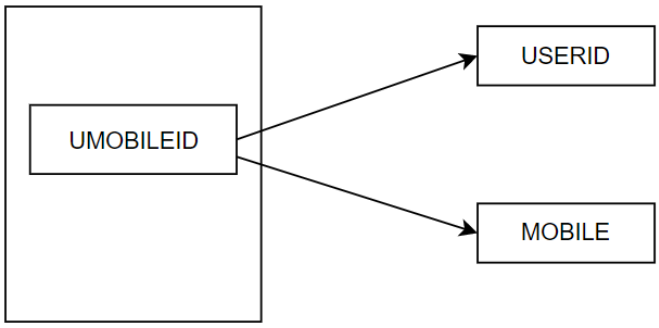


Anomalies:

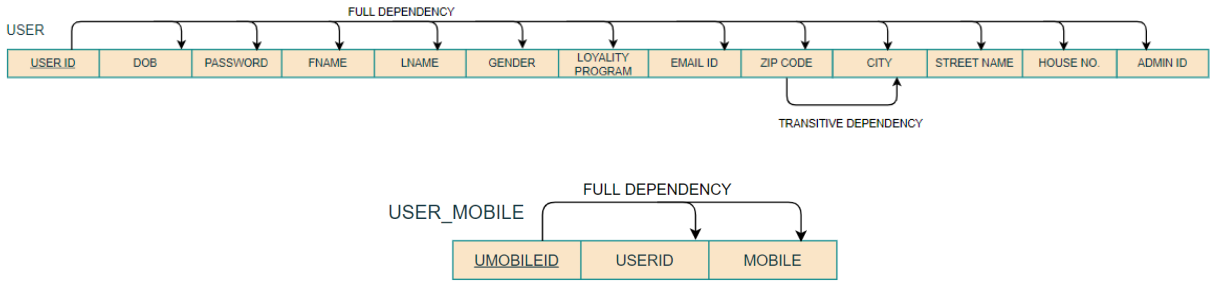
Insertion	None
Deletion	None
Updation	None

6. User


USER ID
GENDER
FNAME
DOB
LNAME
PASSWORD
LOYALTY PROGRAM
STREET NAME
HOUSE NO.
ZIP CODE
CITY
ADMIN ID
EMAIL ID


UMOBILEID
USERID
MOBILE

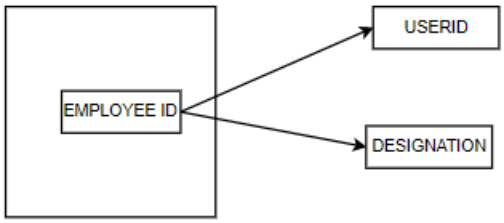
Functional Dependency Chart:



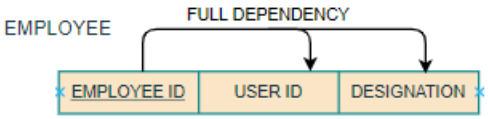
Anomalies:

Insertion	None
Deletion	None
Updation	After updating city one must update zipcode

7.Employee



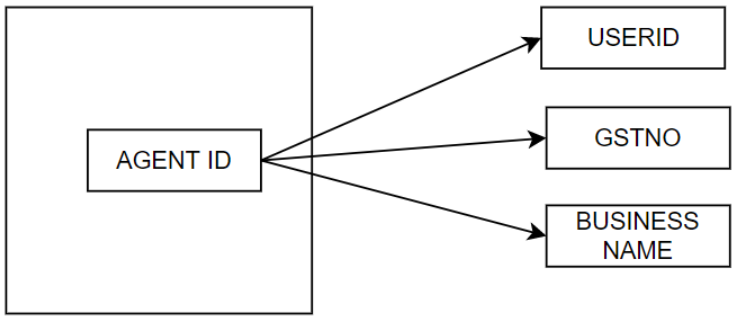
Functional Dependency Chart:



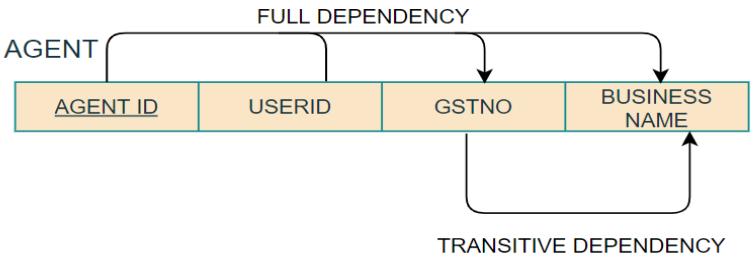
Anomalies:

Insertion	None
Deletion	None
Updation	None

8. Agent



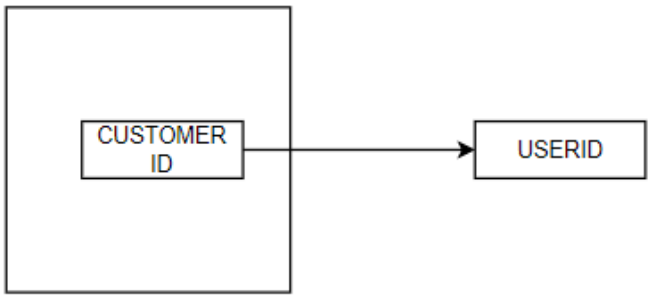
Functional Dependency Chart:



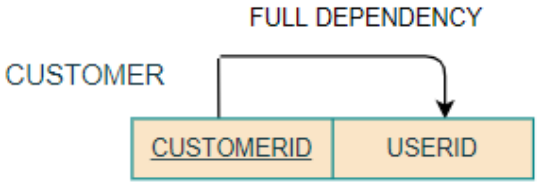
Anomalies:

Insertion	None
Deletion	None
Updation	After updating GST no or business name one must update the other

9.Customer



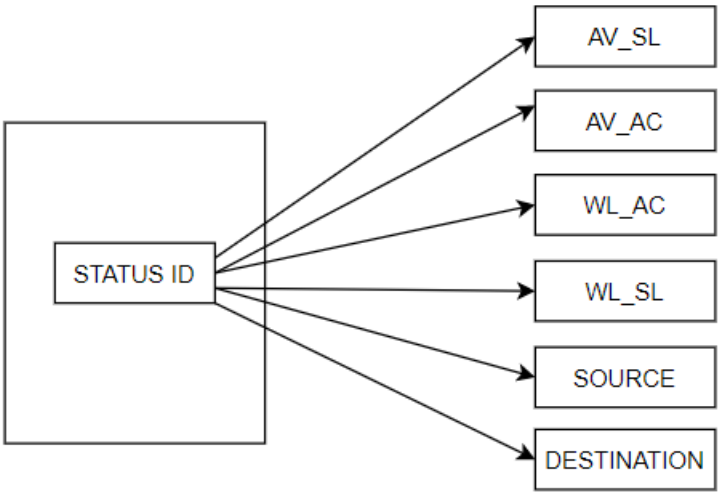
Functional Dependency Chart:



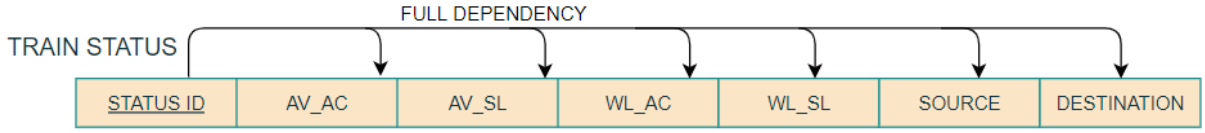
Anomalies:

Insertion	None
Deletion	None
Updation	None

10. Train Status



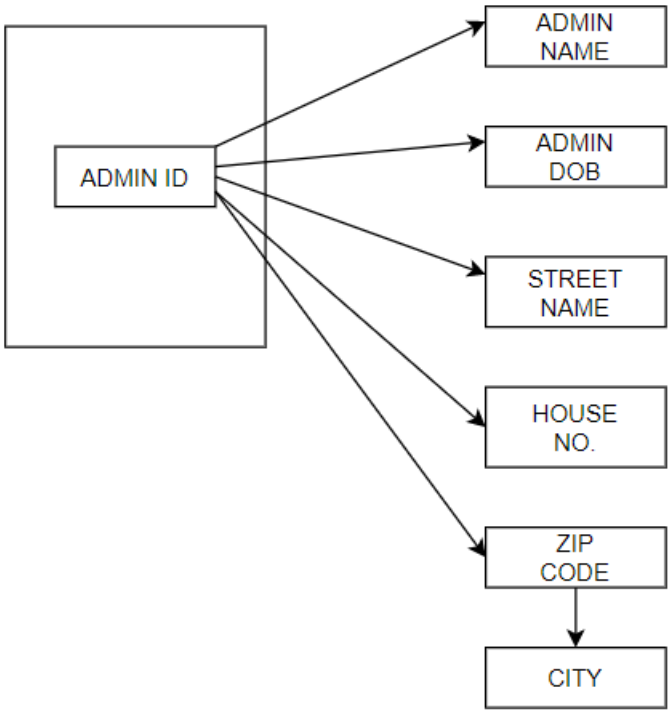
Functional Dependency Chart:

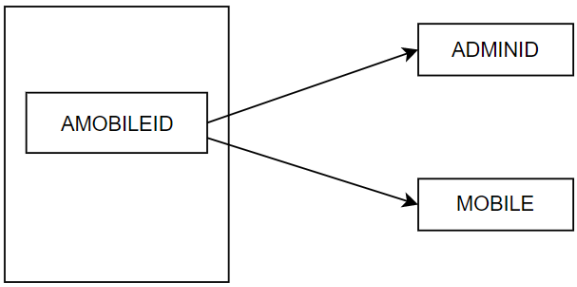


Anomalies:

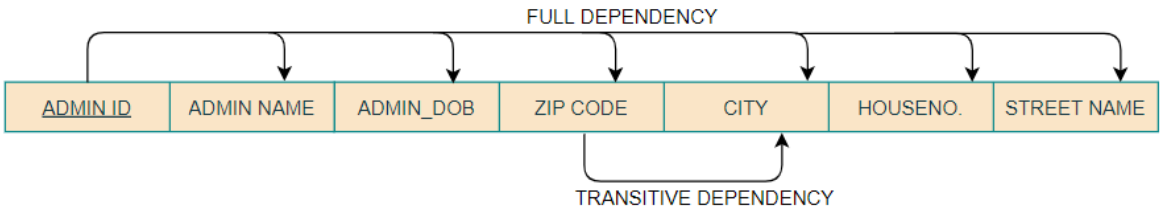
Insertion	None
Deletion	None
Updation	None

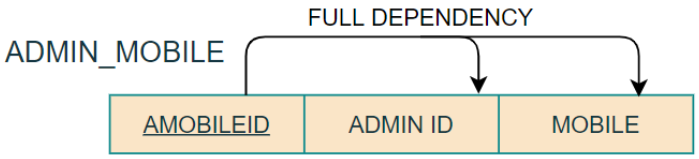
11.Admin





Functional Dependency Chart:

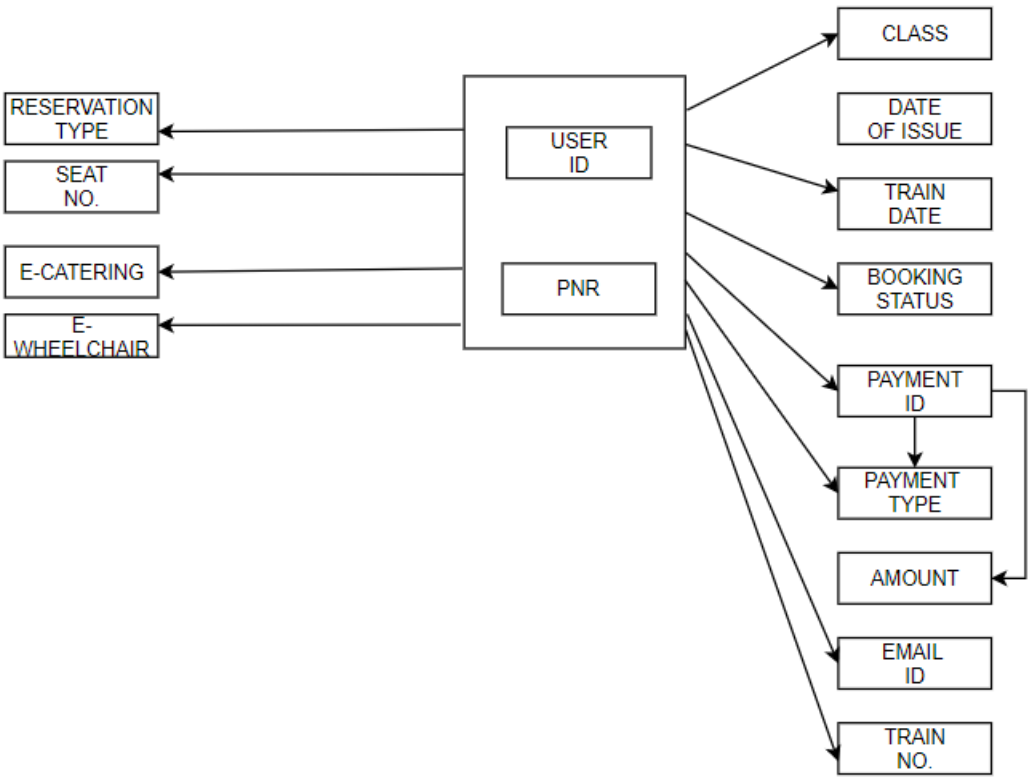




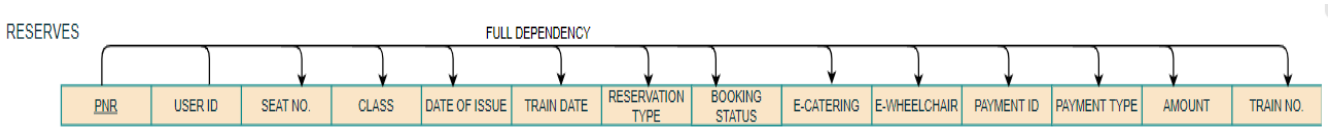
Anomalies:

Insertion	None
Deletion	None
Updation	None

12. Reserves



Functional Dependency Chart:



Anomalies:

Insertion	None
Deletion	None
Updation	After updating one must update the train status

Normalization

All the tables in our database are in 3NF form except the user tables (Customer, Employee and Agent) and Admin Table. So to convert it from 2NF form to 3NF form, we will have to remove transitive dependency as follows:

1.User tables (Customer, Employee and Agent):

UserID	Fname	Lname	Gender	DOB	Password	Loyalty No.	Email ID	Streetname	Houseno.	City	ZipCode

User:

UserID	Fname	Lname	Gender	DOB	Password	Loyalty No.	EmailID	Street name	Houseno	Zipcode

+

City:

CityName	Zipcode

2.Admin table:

AdminID	Adminname	AdminDOB	Streetname	Houseno.	City	ZipCode

Admin:

AdminID	Adminname	AdminDOB	Streetname	Houseno.	Zipcode

+

City:

CityName	Zipcode

3. Agent Table:

Agent:

UserID	Fname	Lname	Gender	DOB	Password	Loyalty No.	EmailID	Street name	Houseno	Zipcode	GSTNo.	Busines Name

+

City:

CityName	Zipcode

Agent:

UserID	Fname	Lname	Gender	DOB	Password	Loyalty No.	EmailID	Street name	Houseno	Zipcode	GSTNo.

+

City:

CityName	Zipcode

+

GST:

Business Name	GST No.

IMPLEMENTATION

```
mysql> show tables;
+------------------------------+
| Tables_in_railway_management |
+------------------------------+
| admin                        |
| admin_mobile                 |
| agent                        |
| class                        |
| customer                     |
| employee                     |
| gst                          |
| reserves                     |
| station                      |
| train                        |
| trainroutes                  |
| trainschedule                |
| trainstatus                  |
| user                         |
| user_mobile                  |
| zip                          |
+------------------------------+
16 rows in set (0.39 sec)
```

create database railway_management;
use railway_management;
create table zip(zipcode varchar(10) primary key,city varchar(20));
create table gst(gstno varchar(20) primary key,businessname varchar(50));
create table admin(adminid int auto_increment primary key,adminname varchar(50),admin_dob varchar(20),houseno varchar(10),streetname varchar(20),zipcode varchar(10),foreign key(zipcode) references zip(zipcode));
create table admin_mobile(amobile int primary key,adminid int,foreign key(adminid) references admin(adminid),mobile varchar(20));
create table train(trainno varchar(20) primary key,trainname varchar(50) unique,traintype varchar(30) not null,distance int,adminid int,foreign key(adminid) references admin(adminid));
create table station(stationid int auto_increment primary key,stationname varchar(20) not null);
create table trainroutes(routeid int primary key,stopno int,arrivaltime varchar(10),departuretime varchar(10),dayno int not null,trainno varchar(20),foreign key(trainno) references train(trainno),stationid int,foreign key(stationid) references station(stationid));
create table trainschedule(scheduleid int primary key,daysrunning varchar(10),duration varchar(20),starttime varchar(10),endtime varchar(10),trainno varchar(20),foreign key(trainno) references train(trainno));

create table class(classid int primary key,ac varchar(10),sleeper varchar(10),trainno varchar(20),foreign key(trainno) references train(trainno));
create table user(userid int primary key,dob varchar(15) not null,password varchar(20) not null,fname varchar(20),lname varchar(20),gender varchar(10),loyaltyprogram varchar(20),emailid varchar(50),zipcode varchar(10),foreign key(zipcode) references zip(zipcode),houseno varchar(10),streetname varchar(30),adminid int,foreign key(adminid) references admin(adminid));
create table agent(agentid int primary key,gstno varchar(20),foreign key(gstno) references gst(gstno),userid int,foreign key(userid) references user(userid));
create table customer(customerid int primary key,userid int,foreign key(userid) references user(userid));
create table employee(employeeid int primary key,userid int,foreign key(userid) references user(userid),designation varchar(30) not null);
create table user_mobile(umobile int primary key,userid int,foreign key(userid) references user(userid),mobile varchar(20));
create table reserves(pnr varchar(20) primary key,dateofissue varchar(15) not null,seatno varchar(5),bookingstatus varchar(20),reservationtype varchar(20),class varchar(20),e_wheelchair varchar(10),e_catering varchar(10),traindate varchar(15) not null,paymenttype varchar(20),paymentid varchar(10),amount int,trainno varchar(20),foreign key(trainno) references train(trainno),userid int,foreign key(userid) references user(userid));
create table trainstatus(statusid int auto_increment primary key,available_sleeper int not null,available_ac int not null,waiting_sleeper int not null,waiting_ac int not null,source varchar(20) not null,destination varchar(20) not null,trainno varchar(20),foreign key(trainno) references train(trainno), userid int,foreign key(userid) references user(userid);

SCREENSHOTS AFTER INSERTION OF VALUES:

```
mysql> select * from admin;
+---------+-------------+-------------+---------+--------------+---------+
| adminid | adminname   | admin_dob   | houseno | streetname   | zipcode |
+---------+-------------+-------------+---------+--------------+---------+
|       1 | Ram Singh   | 12-Mar-1970 | 12-A    | Saket Road   | 110001  |
|       2 | Surya Kumar | 25-Nov-1986 | 62-C    | Nehru place  | 600001  |
|       3 | Naresh Yadav| 6-Jun-1975  | 23      | JM road      | 700001  |
|       4 | Aman Gupta  | 21-Apr-1990 | 59      | FC road      | 411002  |
|       5 | Shyam Dev   | 30-Oct-1983 | 123/34  | Lajpat nagar | 320008  |
+---------+-------------+-------------+---------+--------------+---------+
5 rows in set (0.07 sec)

mysql> select * from admin_mobile;
+-----------+---------+------------+
| amobileid | adminid | mobile     |
+-----------+---------+------------+
|         1 |       1 | 9875641235 |
|         2 |       1 | 8564245974 |
|         3 |       2 | 7456125936 |
|         4 |       3 | 9756321568 |
|         5 |       4 | 8563241256 |
|         6 |       4 | 9854632156 |
|         7 |       5 | 7051145623 |
+-----------+---------+------------+
7 rows in set (0.00 sec)

mysql> select * from train;
+---------+-----------------------------------------+--------------+----------+---------+
| trainno | trainname                               | traintype    | distance | adminid |
+---------+-----------------------------------------+--------------+----------+---------+
| 11077   | Pune Jammu Tawi Jhelum Express          | Mail Express |     2176 |       5 |
| 12019   | Howrah- Ranchi Shatabdi Express         | Shatabdi     |      426 |       5 |
| 12267   | Mumbai Central- Ahmedabad AC Duronto EXP| Duronto      |      790 |       1 |
| 12423   | Dibrugarh Town Rajdhani Express         | Passenger    |     2434 |       1 |
| 12426   | Jammu Tawi- New Delhi Rajdhani Express  | Rajdhani     |      577 |       2 |
| 12430   | New Delhi- Lucknow AC SF Express        | Superfast    |      492 |       3 |
| 12437   | Secunderbad- Hazrat Nizamuddin Rajdhani Exp | Rajdhani | 1661 |       4 |
| 22201   | Kolkata Seldah- Puri Duronto Express    | Duronto      |      541 |       2 |
| 22204   | Secunderabad- Visakhapatnam AC Duronto Exp | Duronto   |      698 |       2 |
| 22206   | Madurai- Chennai Central AC Duronto Exp | Duronto      |      556 |       3 |
+---------+-----------------------------------------+--------------+----------+---------+
10 rows in set (0.03 sec)
```

```
mysql> select * from trainstatus;
+----------+-------------------+--------------+-----------------+------------+--------------+---------------+---------+--------+
| statusid | available_sleeper | available_ac | waiting_sleeper | waiting_ac | source       | destination   | trainno | userid |
+----------+-------------------+--------------+-----------------+------------+--------------+---------------+---------+--------+
|        1 |                 0 |          864 |               0 |          0 | Dibrugarh    | New Delhi     |   12423 |      1 |
|        2 |               238 |          578 |              12 |         52 | Ahmedabad    | Mumbai        |   12267 |     13 |
|        3 |               310 |          420 |               0 |          0 | Jammu        | New Delhi     |   12426 |     11 |
|        4 |               197 |          209 |               0 |        125 | Kolkata      | Puri          |   22201 |      5 |
|        5 |                 0 |          913 |               0 |         25 | Secunderabad | Visakhapatnam |   22204 |      8 |
|        6 |               428 |          542 |               5 |          0 | Madurai      | Chennai       |   22206 |      9 |
|        7 |               432 |          711 |               1 |          0 | New Delhi    | Lucknow       |   12430 |     11 |
|        8 |               521 |          670 |               0 |          0 | Secunderabd  | New Delhi     |   12437 |     15 |
|        9 |               389 |          564 |               0 |          3 | Howrah       | Ranchi        |   12019 |      3 |
|       10 |               782 |          983 |               5 |          6 | Pune         | Jammu         |   11077 |      7 |
+----------+-------------------+--------------+-----------------+------------+--------------+---------------+---------+--------+
10 rows in set (0.04 sec)

mysql> select * from trainschedule;
+------------+-------------+---------------+-----------+----------+---------+
| scheduleid | daysrunning | duration      | starttime | endtime  | trainno |
+------------+-------------+---------------+-----------+----------+---------+
|          1 | M           | 37hours40mins | 20:35     | 10:15+2n | 12423   |
|          2 | M,T,W,F,S   | 6hours35mins  | 11:25     | 5:55+1n  | 12267   |
|          3 | M,T,Th,S,S  | 9hours15mins  | 19:45     | 5:00+1n  | 12426   |
|          4 | M,W,F       | 8hours35mins  | 20:00     | 4:35+1n  | 22201   |
|          5 | M,W,S       | 9hours35mins  | 20:15     | 6:30+1n  | 22204   |
|          6 | T,Th        | 8hours35mins  | 22:45     | 7:20+1n  | 22206   |
|          7 | M,T,F,S     | 8hours5mins   | 23:25     | 7:30+1n  | 12430   |
|          8 | W           | 21hours35mins | 12:45     | 10:40+1n | 12437   |
|          9 | W,Th,F,S    | 7hours5mins   | 6:05      | 13:15    | 12019   |
|         10 | M,T,S       | 51hours10mins | 17:20     | 10:15+2n | 11077   |
+------------+-------------+---------------+-----------+----------+---------+
10 rows in set (0.04 sec)

mysql> select * from trainroutes;
+---------+--------+-------------+---------------+-------+---------+-----------+
| routeid | stopno | arrivaltime | departuretime | dayno | trainno | stationid |
+---------+--------+-------------+---------------+-------+---------+-----------+
|       1 |      1 | Start       | 20:35         |     1 | 12423   |         1 |
|       2 |      2 | 21:40       | 2:13          |     2 | 12423   |         2 |
|       3 |      3 | 6:38        | 6:58          |     2 | 12423   |         3 |
|       4 |      4 | 19:05       | 19:15         |     2 | 12423   |         4 |
|       5 |      5 | 4:57        | 5:02          |     3 | 12423   |         5 |
|       6 |      6 | 10:15       | End           |     3 | 12423   |         6 |
|       7 |      1 | Start       | 19:05         |     1 | 12267   |         9 |
|       8 |      2 | 20:37       | 20:47         |     1 | 12267   |         7 |
|       9 |      3 | 23:15       | 23:35         |     1 | 12267   |         8 |
|      10 |      4 | 6:00        | End           |     1 | 12267   |        10 |
|      11 |      1 | Start       | 19:45         |     1 | 12426   |        11 |
|      12 |      2 | 20:45       | 20:47         |     1 | 12426   |        12 |
|      13 |      3 | 21:22       | 21:27         |     1 | 12426   |        13 |
|      14 |      4 | 0:05        | 0:15          |     2 | 12426   |        14 |
|      15 |      5 | 5:00        | End           |     2 | 12426   |         6 |
|      16 |      1 | Start       | 20:00         |     1 | 22201   |        16 |
|      17 |      2 | 22:15       | 22:20         |     1 | 22201   |        17 |
|      18 |      3 | 4:35        | End           |     2 | 22201   |        18 |
|      19 |      1 | Start       | 20:15         |     1 | 22204   |        19 |
|      20 |      2 | 1:05        | 1:15          |     2 | 22204   |        20 |
|      21 |      3 | 6:30        | End           |     2 | 22204   |        21 |
|      22 |      1 | Start       | 22:45         |     1 | 22206   |        22 |
|      23 |      2 | 2:25        | 2:30          |     2 | 22206   |        23 |
|      24 |      3 | 7:20        | End           |     2 | 22206   |        24 |
|      25 |      1 | Start       | 23:25         |     1 | 12430   |         6 |
|      26 |      2 | 0:05        | 0:07          |     2 | 12430   |        26 |
|      27 |      3 | 2:13        | 2:15          |     2 | 12430   |        27 |
|      28 |      4 | 3:37        | 3:39          |     2 | 12430   |        28 |
|      29 |      5 | 4:43        | 4:45          |     2 | 12430   |        29 |
|      30 |      6 | 7:30        | End           |     2 | 12430   |        30 |
|      31 |      1 | Start       | 12:45         |     1 | 12437   |        19 |
|      32 |      2 | 14:28       | 14:30         |     1 | 12437   |        32 |
|      33 |      3 | 17:45       | 17:50         |     1 | 12437   |        33 |
|      34 |      4 | 20:40       | 20:45         |     1 | 12437   |        34 |
|      35 |      5 | 2:00        | 2:10          |     2 | 12437   |        35 |
|      36 |      6 | 5:45        | 5:50          |     2 | 12437   |        36 |
|      37 |      7 | 10:40       | End           |     2 | 12437   |        37 |
|      38 |      1 | Start       | 6:05          |     1 | 12019   |        38 |
|      39 |      2 | 7:48        | 7:50          |     1 | 12019   |        39 |
|      40 |      3 | 8:06        | 8:07          |     1 | 12019   |        40 |
|      41 |      4 | 8:22        | 8:24          |     1 | 12019   |        41 |
|      42 |      5 | 9:21        | 9:26          |     1 | 12019   |        42 |
|      43 |      6 | 10:26       | 10:28         |     1 | 12019   |        43 |
|      44 |      7 | 10:55       | 11:00         |     1 | 12019   |        44 |
|      45 |      8 | 12:00       | 12:02         |     1 | 12019   |        45 |
|      46 |      9 | 13:15       | End           |     1 | 12019   |        46 |
|      47 |      1 | Start       | 17:20         |     1 | 11077   |        47 |
|      48 |      2 | 20:22       | 20:25         |     1 | 11077   |        48 |
|      49 |      3 | 1:55        | 2:00          |     2 | 11077   |        49 |
|      50 |      4 | 16:24       | 16:26         |     2 | 11077   |        50 |
|      51 |      5 | 17:55       | 17:59         |     2 | 11077   |        51 |
|      52 |      6 | 20:26       | 20:28         |     2 | 11077   |        52 |
|      53 |      7 | 21:00       | 21:15         |     2 | 11077   |        53 |
|      54 |      8 | 21:49       | 21:50         |     2 | 11077   |         6 |
|      55 |      9 | 22:32       | 22:34         |     2 | 11077   |        54 |
|      56 |     10 | 23:20       | 23:25         |     2 | 11077   |        55 |
|      57 |     11 | 23:32       | 23:34         |     2 | 11077   |        56 |
|      58 |     12 | 1:35        | 1:45          |     3 | 11077   |        57 |
|      59 |     13 | 1:56        | 1:58          |     3 | 11077   |        58 |
|      60 |     14 | 2:13        | 2:15          |     3 | 11077   |        59 |
|      61 |     15 | 4:20        | 4:22          |     3 | 11077   |        14 |
```

```
|       62 |      16 | 5:25         | 5:27           |       3 | 11077   |          60 |
|       63 |      17 | 8:13         | 8:15           |       3 | 11077   |          12 |
|       64 |      18 | 9:32         | 9:34           |       3 | 11077   |          61 |
|       65 |      19 | 10:15        | End            |       3 | 11077   |          11 |
+----------+---------+--------------+----------------+---------+---------+-------------+
65 rows in set (0.04 sec)

mysql> select * from station;
+------------+--------------------+
| stationid  | stationname        |
+------------+--------------------+
|          1 | Dibrugarh Town     |
|          2 | NEW TINSUKIA JN    |
|          3 | Dimapur            |
|          4 | Guwahati           |
|          5 | Barauni            |
|          6 | NewDelhi           |
|          7 | Rajkot             |
|          8 | Surendranagar      |
|          9 | Ahmedabad          |
|         10 | Mumbai Central     |
|         11 | Jammu Tawi         |
|         12 | Kathua             |
|         13 | Pathankot Cantt    |
|         14 | Ludhiana           |
|         16 | Kolkata Seldah     |
|         17 | Kharagpur          |
|         18 | Puri               |
|         19 | Secunderadbad      |
|         20 | Vijaywada          |
|         21 | Visakhapatnam      |
|         22 | Madurai            |
|         23 | Salem JN           |
|         24 | Chennai Central    |
|         26 | Ghaziabad          |
|         27 | Moradabad          |
|         28 | Bareilly           |
|         29 | Shahjehanpur       |
|         30 | Lucknow            |
|         32 | Kazipet JN         |
|         33 | Balharshah         |
|         34 | Nagpur             |
|         35 | Bhopal JN          |
|         36 | Jhansi JN          |
|         37 | Delhi              |
|         38 | Howrah JN          |
|         39 | Durgapur           |
|         40 | Raniganj           |
|         41 | Asansol JN         |
|         42 | Dhanbad JN         |
|         43 | Chandrapura        |
|         44 | Bokaro Steel City  |
|         45 | Muri               |
|         46 | Ranchi             |
|         47 | Pune JN            |
|         48 | Ahmednagar         |
|         49 | Jalgaon JN         |
|         50 | Gwalior JN         |
|         51 | Agra Cantt         |
|         52 | Mathura JN         |
|         53 | Faridabad          |
|         54 | Sonipat            |
|         55 | Panipat JN         |
|         56 | Karnal             |
|         57 | Kurukshetra JN     |
|         58 | Ambala cantt JN    |
|         59 | Ambala City        |
|         60 | Jalandhar Cantt    |
|         61 | Vijaypur Jammu     |
+------------+--------------------+
58 rows in set (0.04 sec)

mysql> select * from user;
+--------+------------+-------------+----------+----------+--------+---------------+---------------------+---------+---------+--------------------+-------
--+
| userid | dob        | password    | fname    | lname    | gender | loyaltyprogram | emailid             | zipcode | houseno | streetname         | admini
d |
+--------+------------+-------------+----------+----------+--------+---------------+---------------------+---------+---------+--------------------+-------
--+
|      1 | 12-Mar-1970 | ani@1234    | Anirudh  | Batra    | Male   | No            | anirudh12@yahoo.com | 238727  | 97B     | Ankita Nagar       |
1 |
|      2 | 25-Nov-1986 | javedbest789 | Javed    | Malik    | Male   | Yes           | javedmick@gmail.com | 110001  | 103     | Yamini Chowk       |
2 |
|      3 | 6-Jun-1975  | randhawa09  | Mohan    | Randhawa | Male   | No            | mohanswa@gmail.com  | 222601  | 99      | Neela Villas       |
2 |
|      4 | 21-Apr-1990 | up34678     | Upasana  | Singh    | Female | No            | upasaad@gmail.com   | 102255  | 66      | Fakaruddin Heights |
3 |
|      5 | 30-Oct-1983 | nancoon12   | Nancy    | Khurana  | Female | Yes           | nancoon@gmail.com   | 700001  | 156/C   | Tulsi Nagar        |
4 |
|      6 | 5-Feb-1995  | sfafsf23g   | Isha     | Sodhi    | Female | No            | ishasd4@yahoo.com   | 136536  | 25      | Aundh              |
3 |
|      7 | 12-Sep-1989 | rg5st8ey    | Samir    | Ram      | Male   | No            | samiara@gmail.com   | 442414  | 58/K    | Sameedha Society   |
4 |
|      8 | 15-Mar-1975 | stgerh85h   | Sameedha | Char     | Female | Yes           | samehar@gmail.com   | 281953  | 456     | Borivali           |
1 |
|      9 | 19-Aug-1993 | hyfju12f    | Owais    | Dash     | Male   | No            | owais23@rediff.com  | 548519  | 75      | VimalaGunj         |
4 |
|     10 | 16-Apr-1978 | set4styy    | Manoj    | Nayar    | Male   | Yes           | manoj213@outlook.com | 600001  | 334     | Yeru road          |
```

```
|      7 | 12-Sep-1989 | rg5st8ey     | Samir    | Ram      | Male   | No             | samiara@gmail.com    | 442414  | 58/K    | Sameedha Society   |
4 |
|      8 | 15-Mar-1975 | stgerh85h    | Sameedha | Char     | Female | Yes            | samehar@gmail.com    | 281953  | 456     | Borivali           |
1 |
|      9 | 19-Aug-1993 | hyfju12f     | Owais    | Dash     | Male   | No             | owais23@rediff.com   | 548519  | 75      | VimalaGunj         |
4 |
|     10 | 16-Apr-1978 | set4styy     | Manoj    | Nayar    | Male   | Yes            | manoj213@outlook.com | 600001  | 334     | Yeru road          |
5 |
|     11 | 9-Dec-1988  | yryssr85     | Jatin    | Chander  | Male   | No             | jatider@gmail.com    | 339142  | 40      | Harmada            |
2 |
|     12 | 17-Aug-1965 | ramesh@567   | Ramesh   | Pratap   | Male   | Yes            | rameshpt@gmail.com   | 102255  | 55D     | Chinchwad          |
5 |
|     13 | 1-Jan-1992  | ysrhsr98s    | Nupur    | Patil    | Female | Yes            | nupurp@outlook.com   | 547656  | 41      | Anandpur           |
5 |
|     14 | 25-Aug-1978 | aehhej7j6    | Nitin    | Dev      | Male   | Yes            | nitinlla@gmail.com   | 495954  | 33      | Kanika Heights     |
1 |
|     15 | 6-May-1989  | jdkel65ds    | Vishnu   | Chand    | Male   | Yes            | vishnuc098@gmail.com | 411002  | 486     | Yerwada            |
5 |
+--------+-------------+--------------+----------+----------+--------+----------------+----------------------+---------+---------+--------------------+------
--+
15 rows in set (0.03 sec)

mysql> select * from agent;
+---------+--------------+--------+
| agentid | gstno        | userid |
+---------+--------------+--------+
|       1 | 09AHLPG56980 |      7 |
|       2 | 09AHPERG4981 |      8 |
|       3 | 08APYTH84522 |      9 |
|       4 | 07WERFF45689 |     10 |
|       5 | 08ASDGH34527 |     11 |
+---------+--------------+--------+
5 rows in set (0.05 sec)

mysql> select * from customer;
+------------+--------+
| customerid | userid |
+------------+--------+
|          1 |      1 |
|          2 |      2 |
|          3 |      3 |
|          4 |      4 |
|          5 |      5 |
|          6 |      6 |
+------------+--------+
6 rows in set (0.03 sec)

mysql> select * from employee;
+------------+--------+-------------------------+
| employeeid | userid | designation             |
+------------+--------+-------------------------+
|          1 |     12 | Senior Systems Engineer |
|          2 |     13 | Nodal officer           |
|          3 |     14 | TC                      |
|          4 |     15 | OperationalManager      |
+------------+--------+-------------------------+
4 rows in set (0.04 sec)

mysql> select * from reserves;
+----------+-------------+--------+---------------+-----------------+-------+-------------+------------+-------------+-------------+-----------+---------+-
-------+---------+
| pnr      | dateofissue | seatno | bookingstatus | reservationtype | class | e_wheelchair | e_catering | traindate   | paymenttype | paymentid | trainno |
userid | amount |
+----------+-------------+--------+---------------+-----------------+-------+-------------+------------+-------------+-------------+-----------+---------+-
-------+---------+
| AGH10001 | 12-Sep-2018 | 12A    | Confirmed     | Regular         | AC    | Yes          | Yes        | 13-Oct-2018 | CreditCard  | 1256974521 | 22204   |
     2 |   1200 |
| HDS10004 | 19-Aug-2020 | 76B    | Confirmed     | Tatkal          | AC    | No           | Yes        | 20-Aug-2020 | UPI         | 3215462102 | 22201   |
    12 |   1200 |
| JSA10002 | 15-Mar-2020 | 25B    | Waiting       | Regular         | SL    | No           | No         | 19-Aug-2020 | UPI         | 5632145896 | 12426   |
     3 |    750 |
| LPH10003 | 9-Dec-2018  | 34A    | Confirmed     | Regular         | AC    | No           | No         | 9-Jan-2019  | CreditCard  | 7854123695 | 11077   |
    10 |   1500 |
| PLK10008 | 15-Apr-2020 | 56A    | Confirmed     | Regular         | AC    | Yes          | Yes        | 1-Sep-2020  | UPI         | 8035912067 | 12267   |
     1 |   1200 |
| TMK10005 | 9-June-2020 | 52A    | Waiting       | Regular         | SL    | No           | No         | 29-July-2020| CreditCard  | 1023540256 | 12019   |
     8 |    750 |
| UGC10006 | 15-Oct-2019 | 65A    | Confirmed     | Tatkal          | AC    | No           | No         | 16-Oct-2019 | DebitCard   | 1602354896 | 11077   |
     7 |   1500 |
| WER10007 | 18-Aug-2020 | 89B    | Waiting       | Regular         | AC    | No           | Yes        | 26-Sep-2020 | DebitCard   | 2013542065 | 22201   |
     7 |   1200 |
+----------+-------------+--------+---------------+-----------------+-------+-------------+------------+-------------+-------------+-----------+---------+-
-------+---------+
8 rows in set (0.02 sec)

mysql> select * from user_mobile;
+-----------+--------+------------+
| umobileid | userid | mobile     |
+-----------+--------+------------+
|         1 |      1 | 9875134565 |
|         2 |      2 | 7894153246 |
|         3 |      3 | 8945214569 |
|         4 |      4 | 9845763210 |
|         5 |      5 | 9402365187 |
|         6 |      6 | 8945203617 |
```

```
|       7 | 7520361245 |
|       8 | 8895478210 |
|       9 | 8445436974 |
|      10 | 9675665012 |
|      11 | 7842365109 |
|      12 | 9927054896 |
|      13 | 8997745612 |
|      14 | 9654785211 |
|      15 | 8556321459 |
|       5 | 9852364456 |
|       6 | 7855699512 |
|       8 | 8897455638 |
|       9 | 9545333469 |
|       1 | 7789874520 |
|      12 | 9837125640 |
+---------+------------+
21 rows in set (0.04 sec)

mysql> select * from zip;
+---------+-------------+
| zipcode | city        |
+---------+-------------+
| 102255  | Darjeeling  |
| 110001  | NewDelhi    |
| 136536  | Panaji      |
| 222601  | Gangtok     |
| 238727  | Mysore      |
| 281953  | Patna       |
| 320008  | Ahmedabad   |
| 339142  | Gandhinagar |
| 411002  | Pune        |
| 442414  | DehraDun    |
| 495954  | Kota        |
| 547656  | Gurugram    |
| 548519  | Indore      |
| 600001  | Chennai     |
| 700001  | Kolkata     |
+---------+-------------+
15 rows in set (0.00 sec)

mysql> select * from gst;
+---------------+----------------+
| gstno         | businessname   |
+---------------+----------------+
| 07WERFF45689  | RailWheels     |
| 08APYTH84522  | RailCentre     |
| 08ASDGH34527  | Chander Rails  |
| 09AHLPG56980  | Ram tours      |
| 09AHPERG4981  | SC travels     |
+---------------+----------------+
```

QUERIES

1.Check how many reservations are made in sleeper class

select count(class) from reserves where class='SL';

After the execution of this query, the following result is obtained:

```
mysql> use railway_management;
Database changed
mysql> select count(class) from reserves where class='SL';
+--------------+
| count(class) |
+--------------+
|            2 |
+--------------+
1 row in set (0.17 sec)
```

2 Calculate the average of amount in reservations

select avg(amount) from reserves;

After the execution of this query, the following result is obtained:

```
mysql> select avg(amount) from reserves;
+--------------+
| avg(amount) |
+--------------+
|   1162.5000 |
+--------------+
1 row in set (0.14 sec)
```

3 Check how many tickets are sold

select count(pnr) from reserves;

After the execution of this query, the following result is obtained:

```
mysql> select count(pnr) from reserves;
+------------+
| count(pnr) |
+------------+
|          8 |
+------------+
1 row in set (0.28 sec)
```

4 Print the final sum of amount

select sum(amount) from reserves;

After the execution of this query, the following result is obtained:

```
mysql> select sum(amount) from reserves;
+--------------+
| sum(amount) |
+--------------+
|        9300 |
+--------------+
1 row in set (0.00 sec)
```

5 Give class wise user details

select r.class as 'class acquired',count(r.class) as 'Number of users' from reserves r group by class;

After the execution of this query, the following result is obtained:

MySQL 8.0 Command Line Client - Unicode

```
mysql> select r.class as 'class acquired',count(r.class) as 'Number of users' from reserves r group by class;
+----------------+-----------------+
| class acquired | Number of users |
+----------------+-----------------+
| AC             |               6 |
| SL             |               2 |
+----------------+-----------------+
2 rows in set (0.42 sec)
```

6 Give fname ,lname and amount according to amount in ascending order

select u.fname,u.lname,r.amount from user u,reserves r where u.userid=r.userid order by r.amount asc;

After the execution of this query, the following result is obtained:

```
mysql> select u.fname,u.lname,r.amount from user u,reserves r where u.userid=r.userid  order by r.amount asc;
+----------+----------+--------+
| fname    | lname    | amount |
+----------+----------+--------+
| Mohan    | Randhawa |    750 |
| Sameedha | Char     |    750 |
| Javed    | Malik    |   1200 |
| Ramesh   | Pratap   |   1200 |
| Anirudh  | Batra    |   1200 |
| Samir    | Ram      |   1200 |
| Manoj    | Nayar    |   1500 |
| Samir    | Ram      |   1500 |
+----------+----------+--------+
8 rows in set (0.10 sec)
```

7 Give the user details that have checked train status

select * from user where userid in(select userid from trainstatus);

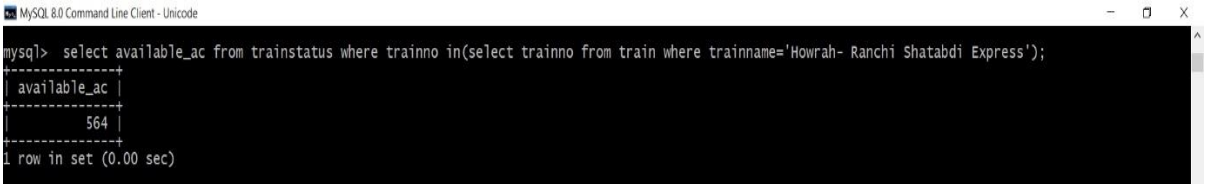
After the execution of this query, the following result is obtained:

```
+----------+----------+--------+
 rows in set (0.10 sec)

ysql> select * from user where userid in(select userid from trainstatus);
+--------+------------+----------+----------+----------+--------+----------------+---------------------+---------+---------+---------------+----------+
| userid | dob        | password | fname    | lname    | gender | loyaltyprogram | emailid             | zipcode | houseno | streetname    | adminid  |
+--------+------------+----------+----------+----------+--------+----------------+---------------------+---------+---------+---------------+----------+
|      1 | 12-Mar-1970 | ani@1234 | Anirudh  | Batra    | Male   | No             | anirudh12@yahoo.com | 238727  | 97B     | Ankita Nagar  |        1 |
|      3 | 6-Jun-1975 | randhawa09 | Mohan  | Randhawa | Male   | No             | mohanswa@gmail.com  | 222601  | 99      | Neela Villas  |        2 |
|      5 | 30-Oct-1983 | nancoon12 | Nancy   | Khurana  | Female | Yes            | nancoon@gmail.com   | 700001  | 156/C   | Tulsi Nagar   |        4 |
|      7 | 12-Sep-1989 | rg5st8ey | Samir    | Ram      | Male   | No             | samiara@gmail.com   | 442414  | 58/K    | Sameedha Society |        4 |
|      8 | 15-Mar-1975 | stgerh85h | Sameedha | Char     | Female | Yes            | samehar@gmail.com   | 281953  | 456     | Borivali      |        1 |
|      9 | 19-Aug-1993 | hyfju12f | Owais    | Dash     | Male   | No             | owais23@rediff.com  | 548519  | 75      | VimalaGunj    |        4 |
|     11 | 9-Dec-1988 | yryssr85 | Jatin    | Chander  | Male   | No             | jatider@gmail.com   | 339142  | 40      | Harmada       |        2 |
|     13 | 1-Jan-1992 | ysrhsr98s | Nupur   | Patil    | Female | Yes            | nupurp@outlook.com  | 547656  | 41      | Anandpur      |        5 |
|     15 | 6-May-1989 | jdkel65ds | Vishnu   | Chand    | Male   | Yes            | vishnuc098@gmail.com | 411002 | 486     | Yerwada       |        5 |
+--------+------------+----------+----------+----------+--------+----------------+---------------------+---------+---------+---------------+----------+
 rows in set (0.01 sec)

ysql> select count(available_ac) from trainstatus where trainno in(select trainno where trainname='Howrah- Ranchi Shatabdi Express');
```

8 Give ac available seats for train Howrah-Ranchi Shatabdi Express

select available_ac from trainstatus where trainno in(select trainno from train where trainname='Howrah- Ranchi Shatabdi Express');

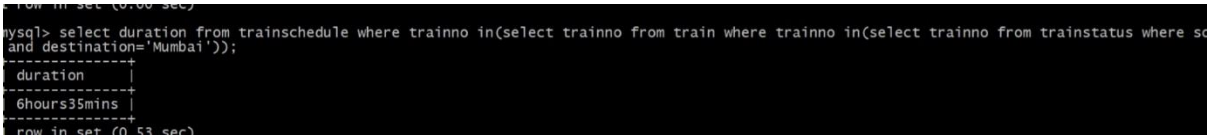
After the execution of this query, the following result is obtained:



9 Give the duration of train from Ahmedabad to Mumbai

select duration from trainschedule where trainno in(select trainno from train where trainno in(select trainno from trainstatus where source='Ahmedabad' and destination='Mumbai'));

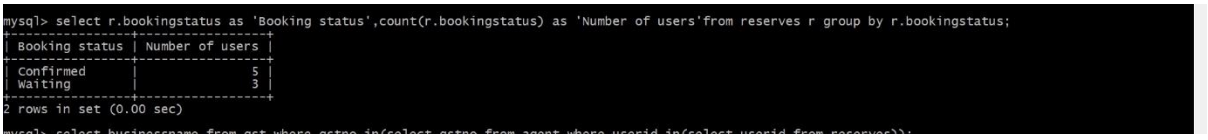
After the execution of this query, the following result is obtained:



10 Give the booking status wise user details

select r.bookingstatus as 'Booking status',count(r.bookingstatus) as 'Number of users'from reserves r group by r.bookingstatus;

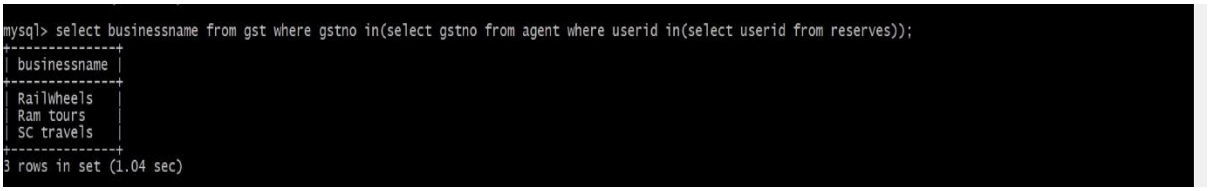
After the execution of this query, the following result is obtained:



11 Give business name of agents who have reserved train tickets

select businessname from gst where gstno in(select gstno from agent where userid in(select userid from reserves));

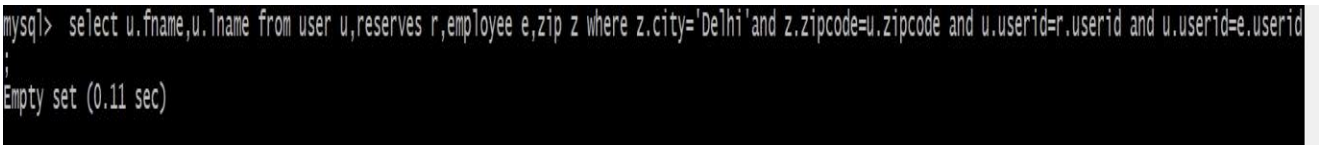
After the execution of this query, the following result is obtained:



12 Give employees first name and last name who are from Delhi and has reserved the train tickets

select u.fname,u.lname from user u,reserves r,employee e,zip z where z.city='Delhi'and z.zipcode=u.zipcode and u.userid=r.userid and u.userid=e.userid;

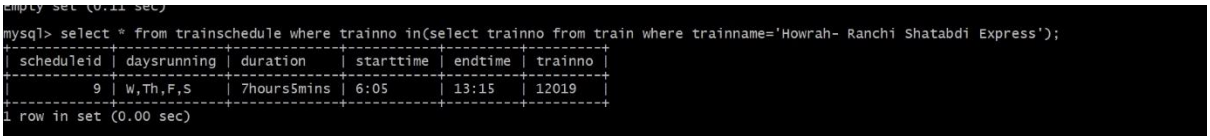
After the execution of this query, the following result is obtained:



13 Give the train schedule of train Howrah- Ranchi Shatabdi Express

select * from trainschedule where trainno in(select trainno from train where trainname='Howrah- Ranchi Shatabdi Express');

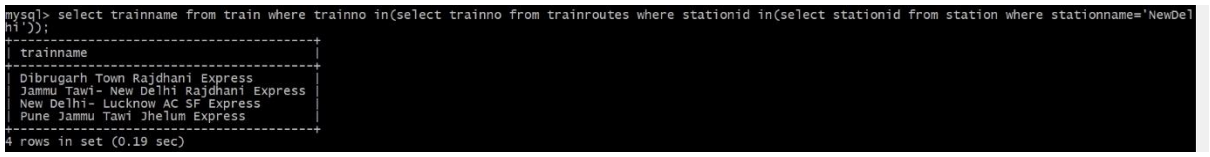
After the execution of this query, the following result is obtained:



14 Give the train names which goes through New Delhi station

select trainname from train where trainno in(select trainno from trainroutes where stationid in(select stationid from station where stationname='NewDelhi'));

After the execution of this query, the following result is obtained:



15 Give the list of customers whose name starts with 'A' and reserved a train seat

select u.fname,u.lname from user u,reserves r,customer c where fname like 'A%' and u.userid=c.userid and u.userid=r.userid;

After the execution of this query, the following result is obtained:

16 Give the details of users who has chosen the reservation type Tatkal

select * from user where userid in(select userid from reserves where reservationtype='Tatkal');

After the execution of this query, the following result is obtained:

```
mysql> select * from user where userid in(select userid from reserves where reservationtype='Tatkal');
+--------+-------------+----------+-------+-------+--------+---------------+-------------------+---------+---------+-----------------+---------+
| userid | dob         | password | fname | lname | gender | loyaltyprogram | emailid           | zipcode | houseno | streetname      | adminid |
+--------+-------------+----------+-------+-------+--------+---------------+-------------------+---------+---------+-----------------+---------+
|     12 | 17-Aug-1965 | ramesh@567 | Ramesh | Pratap | Male   | Yes            | rameshpt@gmail.com | 102255  | 55D     | Chinchwad       |       5 |
|      7 | 12-Sep-1989 | rg5st8ey | Samir | Ram   | Male   | No             | samiara@gmail.com | 442414  | 58/K    | Sameedha Society |       4 |
+--------+-------------+----------+-------+-------+--------+---------------+-------------------+---------+---------+-----------------+---------+
2 rows in set (0.00 sec)

mysql> select trainname from train where trainno in(select trainno from trainstatus where available_sleeper<500 and available_sleeper>200);
```

17 .Give the name of the train where sleeper class is available between 200 and 500

select trainname from train where trainno in(select trainno from trainstatus where available_sleeper<500 and available_sleeper>200);

After the execution of this query, the following result is obtained:

```
mysql> select trainname from train where trainno in(select trainno from trainstatus where available_sleeper<500 and available_sleeper>200);
+-----------------------------------------+
| trainname                               |
+-----------------------------------------+
| Mumbai Central- Ahmedabad AC Duronto EXP |
| Jammu Tawi- New Delhi Rajdhani Express  |
| Madurai- Chennai Central AC Duronto Exp |
| New Delhi- Lucknow AC SF Express        |
| Howrah- Ranchi Shatabdi Express         |
+-----------------------------------------+
5 rows in set (0.00 sec)
```

18 Give the name and mobile number of admin who manages train Mumbai Central- Ahmedabad AC Duronto EXP

select a.adminname,m.mobile from admin a,admin_mobile m,train t where t.trainname='Mumbai Central- Ahmedabad AC Duronto EXP' and t.adminid=a.adminid and a.adminid=m.adminid;

After the execution of this query, the following result is obtained:

```
mysql> select a.adminname,m.mobile from admin a,admin_mobile m,train t where t.trainname='Mumbai Central- Ahmedabad AC Duronto EXP' and t.adminid=a.adminid an
d a.adminid=m.adminid;
+-----------+------------+
| adminname | mobile     |
+-----------+------------+
| Ram Singh | 9875641235 |
| Ram Singh | 8564245974 |
+-----------+------------+
2 rows in set (0.00 sec)

mysql>
```

19 Give the details of all admins whose name doesn't end with 'ta'

select * from admin where adminname not like '%ta';

After the execution of this query, the following result is obtained:

```
mysql> select * from admin where adminname not like '%ta';
+---------+-------------+-------------+---------+-------------+---------+
| adminid | adminname   | admin_dob   | houseno | streetname  | zipcode |
+---------+-------------+-------------+---------+-------------+---------+
|       1 | Ram Singh   | 12-Mar-1970 | 12-A    | Saket Road  | 110001  |
|       2 | Surya Kumar | 25-Nov-1986 | 62-C    | Nehru place | 600001  |
|       3 | Naresh Yadav | 6-Jun-1975 | 23      | JM road     | 700001  |
|       5 | Shyam Dev   | 30-Oct-1983 | 123/34  | Lajpat nagar | 320008 |
+---------+-------------+-------------+---------+-------------+---------+
4 rows in set (0.00 sec)
```

20 Give the train details managed by Ram Singh

select * from train where adminid in(select adminid from admin where adminname='Ram Singh');

After the execution of this query, the following result is obtained:

```
mysql> select * from train where adminid in(select adminid from admin where adminname='Ram Singh');
+---------+-----------------------------------------+-----------+----------+---------+
| trainno | trainname                               | traintype | distance | adminid |
+---------+-----------------------------------------+-----------+----------+---------+
| 12267   | Mumbai Central- Ahmedabad AC Duronto EXP | Duronto   |      790 |       1 |
| 12423   | Dibrugarh Town Rajdhani Express          | Passenger |     2434 |       1 |
+---------+-----------------------------------------+-----------+----------+---------+
2 rows in set (0.00 sec)
```

21 Print the user name in capital whose booking status is confirmed

select upper(fname),upper(lname) from user where userid in(select userid from reserves where bookingstatus='confirmed');

After the execution of this query, the following result is obtained:

```
mysql> select upper(fname),upper(lname) from user where userid in(select userid from reserves where bookingstatus='confirmed');
+---------------+---------------+
| upper(fname)  | upper(lname)  |
+---------------+---------------+
| JAVED         | MALIK         |
| RAMESH        | PRATAP        |
| MANOJ         | NAYAR         |
| ANIRUDH       | BATRA         |
| SAMIR         | RAM           |
+---------------+---------------+
5 rows in set (0.47 sec)
```

22 Give the payment type wise user details

select r.paymenttype as 'Paid Through',count(r.paymenttype) as 'Number of users' from reserves r group by r.paymenttype;

After the execution of this query, the following result is obtained:

```
mysql>     select r.paymenttype as 'Paid Through',count(r.paymenttype) as 'Number of users' from reserves r group by r.paymenttype;
+--------------+-----------------+
| Paid Through | Number of users |
+--------------+-----------------+
| CreditCard   |               3 |
| UPI          |               3 |
| DebitCard    |               2 |
+--------------+-----------------+
3 rows in set (0.10 sec)
```

23 Give the agent names who has paid by credit card

select u.fname,u.lname from user u,agent a,reserves r where r.paymenttype='CreditCard' and r.userid=u.userid and r.userid=a.userid;

After the execution of this query, the following result is obtained:

```
mysql> select u.fname,u.lname from user u,agent a,reserves r where r.paymenttype='CreditCard' and r.userid=u.userid and r.userid=a.userid;
+----------+-------+
| fname    | lname |
+----------+-------+
| Manoj    | Nayar |
| Sameedha | Char  |
+----------+-------+
2 rows in set (0.00 sec)

mysql>
```

24 Give the name of users who have reserved their seats in any train

select fname,lname from user where userid in(select userid from reserves);

After the execution of this query, the following result is obtained:

```
mysql>   select fname,lname from user where userid in(select userid from reserves);
+----------+----------+
| fname    | lname    |
+----------+----------+
| Anirudh  | Batra    |
| Javed    | Malik    |
| Mohan    | Randhawa |
| Samir    | Ram      |
| Sameedha | Char     |
| Manoj    | Nayar    |
| Ramesh   | Pratap   |
+----------+----------+
7 rows in set (0.00 sec)
```

25 Give the names of user who have taken reservation in Kolkata Seldah- Puri Duronto Express train

select fname,lname from user where userid in(select userid from reserves where trainno in(select trainno from train where trainname='Kolkata Seldah- Puri Duronto Express'));

After the execution of this query, the following result is obtained:

```
mysql> select fname,lname from user where userid in(select userid from reserves where trainno in(select trainno from train where trainname='Kolkata Seldah- Pu
ri Duronto Express'));
+--------+--------+
| fname  | lname  |
+--------+--------+
| Ramesh | Pratap |
| Samir  | Ram    |
+--------+--------+
2 rows in set (0.03 sec)
```

FUNCTIONS

1) Create a function to return the total reservations done.

```
mysql> delimiter @@
mysql> create function tot1()
    -> returns int
    -> DETERMINISTIC
    -> begin
    -> declare total int;
    -> select count(pnr) into total from reserves;
    -> return total;
    -> end @@
Query OK, 0 rows affected (0.20 sec)

mysql> select tot1()\g
+--------+
| tot1() |
+--------+
|      8 |
+--------+
1 row in set (0.25 sec)
```

2) Create a function to return the amount paid by accepting first and last name of user

```
mysql> delimiter @@
mysql> create function amt(x varchar(20),y varchar(20))
    -> returns int
    -> DETERMINISTIC
    -> begin
    -> declare amt1 int;
    -> select amount into amt1 from reserves where userid in(select userid from user where fname=x and lname=y);
    -> return amt1;
    -> end @@
Query OK, 0 rows affected (0.46 sec)

mysql> select amt('Javed','Malik')\g
+----------------------+
| amt('Javed','Malik') |
+----------------------+
|                 1200 |
+----------------------+
1 row in set (0.14 sec)
```

3) Create a function to return the average of the total amount paid on reservations

```
mysql|> ^C
mysql> delimiter @@
mysql> create function avg1()
    -> returns decimal(5,1)
    -> DETERMINISTIC
    -> begin
    -> declare total decimal(5,1);
    -> select avg(amount) into total from reserves;
    -> return total;
    -> end @@
Query OK, 0 rows affected (1.17 sec)

mysql> select avg1()\g
+---------+
| avg1() |
+---------+
| 1162.5 |
+---------+
1 row in set (0.03 sec)
```

4) Create a function to return the maximum amount paid on reservation

```
mysql> delimiter @@
mysql> create function max1()
    -> returns int
    -> DETERMINISTIC
    -> begin
    -> declare total int;
    -> select max(amount) into total from reserves;
    -> return total;
    -> end @@
Query OK, 0 rows affected (0.22 sec)

mysql> select max()\g
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ')'
at line 1
mysql> select max1()\g
+---------+
| max1() |
+---------+
|   1500 |
+---------+
1 row in set (0.00 sec)
```

5) Create a function to return the seat number by accepting the pnr

```
mysql> delimiter @@
mysql> create function seat(pnr1 varchar(30))
    -> returns varchar(30)
    -> DETERMINISTIC
    -> begin
    -> declare tot varchar(30);
    -> select seatno into tot from reserves where pnr=pnr1;
    -> return tot;
    -> end @@
Query OK, 0 rows affected (1.13 sec)

mysql>
mysql>
mysql> select seat('AGH10001')\g
+-------------------+
| seat('AGH10001') |
+-------------------+
| 12A               |
+-------------------+
1 row in set (0.00 sec)
```

PROCEDURES

1.Print the total amount of money in reservations

```
mysql> delimiter $$
mysql> create procedure get_amount()
    -> BEGIN
    -> select sum(amount) from reserves;
    -> END $$
Query OK, 0 rows affected (0.21 sec)

mysql> call get_amount();\G
*************************** 1. row ***************************
sum(amount): 9300
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.04 sec)
```

2.Print all the train numbers

```
mysql> delimiter $$
mysql> create procedure get_trainno()
    -> BEGIN
    -> SELECT trainno FROM train;
    -> END $$
Query OK, 0 rows affected (0.13 sec)

mysql> call get_trainno(); \G
*************************** 1. row ***************************
trainno: 12267
*************************** 2. row ***************************
trainno: 12423
*************************** 3. row ***************************
trainno: 12426
*************************** 4. row ***************************
trainno: 22201
*************************** 5. row ***************************
trainno: 22204
*************************** 6. row ***************************
trainno: 12430
*************************** 7. row ***************************
trainno: 22206
*************************** 8. row ***************************
trainno: 12437
*************************** 9. row ***************************
trainno: 11077
*************************** 10. row ***************************
trainno: 12019
10 rows in set (0.04 sec)

Query OK, 0 rows affected (0.14 sec)
```

3.Update the user contact details using procedure

```
mysql> delimiter //
mysql> create procedure update_contact(IN temp_userid int,IN new_contact varchar(20))
    -> begin
    -> update user_mobile set mobile=new_contact where userid=temp_userid;
    -> end; //
Query OK, 0 rows affected (0.16 sec)

mysql> call update_contact('15','7589620311'); //
Query OK, 1 row affected (0.30 sec)
```

```
mysql> delimiter ;
mysql> select * from user_mobile;
+--------+------------+
| userid | mobile     |
+--------+------------+
|      1 | 9875134565 |
|      2 | 7894153246 |
|      3 | 8945214569 |
|      4 | 9845763210 |
|      7 | 7520361245 |
|      8 | 8895478210 |
|      9 | 8445436974 |
|     10 | 9675665012 |
|     11 | 7842365109 |
|     12 | 9927054896 |
|     13 | 8997745612 |
|     14 | 9654785211 |
|     15 | 7589620311 |
|      8 | 8897455638 |
|      9 | 9545333469 |
|      1 | 7789874520 |
|     12 | 9837125640 |
+--------+------------+
17 rows in set (0.00 sec)
```

4.Display gender of users

```
mysql> delimiter //
mysql> create procedure disp_gender(INOUT mfgender int,IN user_gender varchar(10))
    -> begin
    -> select count(gender) into mfgender from user where gender=user_gender;
    -> end ; //
Query OK, 0 rows affected (0.22 sec)

mysql> delimiter ;
```

```
mysql> call disp_gender(@M,"Male");
Query OK, 1 row affected (0.00 sec)

mysql> select @M;
+------+
| @M   |
+------+
|   10 |
+------+
1 row in set (0.00 sec)

mysql> call disp_gender(@F,"Female");
Query OK, 1 row affected (0.00 sec)

mysql> select @F;
+------+
| @F   |
+------+
|    6 |
+------+
1 row in set (0.00 sec)
```

5. Print out the maximum amount in train reservations

```
mysql> delimiter //
mysql> create procedure disp_max(OUT highestprice int)
    -> begin
    -> select max(amount) into highestprice from reserves;
    -> end; //
Query OK, 0 rows affected (0.15 sec)

mysql> delimiter ;
mysql> call disp_max(@A);
Query OK, 1 row affected (0.02 sec)

mysql> select @A;
+------+
| @A   |
+------+
| 1500 |
+------+
1 row in set (0.00 sec)
```

TRIGGERS

1. Before updating the users name storing the old values in a different table using trigger

```
mysql> delimiter //
mysql> create trigger update_user
    -> before update on user
    -> for each row
    -> begin
    -> insert into user_update values(old.userid,old.fname,old.lname);
    -> end;//
Query OK, 0 rows affected (0.14 sec)

mysql> delimiter ;
```

```
mysql> update user set fname='Mamta' where userid='13';
Query OK, 1 row affected (0.85 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from user_update;
+--------+-------+-------+
| userid | fname | lname |
+--------+-------+-------+
|     13 | Nupur | Patil |
+--------+-------+-------+
1 row in set (0.00 sec)
```

2. After updating the users name storing the new values in a different table using trigger

```
mysql> delimiter //
mysql> create trigger afterupdate
    -> after update on user
    -> for each row
    -> begin
    -> insert into after_update_user values (new.userid,new.fname,new.lname);
    -> end; //
Query OK, 0 rows affected (0.47 sec)
```

```
mysql> delimiter ;
mysql> update user set fname='Suresh' where userid='1';
Query OK, 1 row affected (0.50 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from after_update_user;
+--------+--------+-------+
| userid | fname  | lname |
+--------+--------+-------+
|      1 | Suresh | Batra |
+--------+--------+-------+
1 row in set (0.00 sec)
```

3. After inserting users mobile number storing it in different table

```
mysql> create table insertcontacts(userid int,mobile varchar(20),foreign key(userid) references user(userid));
Query OK, 0 rows affected (1.30 sec)

mysql> delimiter //
mysql> create trigger contacts_after_insert
    -> after insert
    -> on user_mobile for each row
    -> begin
    -> insert into insertcontacts values(new.userid,new.mobile);
    -> end; //
Query OK, 0 rows affected (0.23 sec)

mysql> insert into user_mobile values ('6','8874912360');
    -> //
Query OK, 1 row affected (0.33 sec)

mysql> delimiter ;
mysql> select * from insertcontacts;
+--------+------------+
| userid | mobile     |
+--------+------------+
|      6 | 8874912360 |
+--------+------------+
1 row in set (0.00 sec)
```

4. Before deleting users mobile number storing it in different table

```
mysql>  create table delcontacts(userid int,mobile varchar(20),foreign key(userid) references user(userid));
Query OK, 0 rows affected (2.42 sec)

mysql> delimiter //
mysql> create trigger contacts_before_delete
    -> before delete
    -> on user_mobile for each row
    -> begin
    -> insert into delcontacts values(old.userid,old.mobile);
    -> end ; //
Query OK, 0 rows affected (0.29 sec)

mysql> delimiter ;
mysql> delete from user_mobile where userid='5';
Query OK, 3 rows affected (0.54 sec)

mysql> select * from delcontacts;
+--------+------------+
| userid | mobile     |
+--------+------------+
|      5 | 9402365187 |
|      5 | 9852364456 |
|      5 | 9874512360 |
+--------+------------+
3 rows in set (0.00 sec)
```

5.After deleting users mobile number storing it in different table for future use

```
mysql> create table afterdelcontacts(userid int,mobile varchar(20),foreign key(userid) references user(userid));
Query OK, 0 rows affected (2.61 sec)

mysql> delimiter //
mysql> create trigger contacts_after_delete
    -> after delete
    -> on user_mobile for each row
    -> begin
    -> insert into afterdelcontacts values(old.userid,old.mobile);
    -> end; //
Query OK, 0 rows affected (0.40 sec)

mysql> delete from user_mobile where userid='6';
    -> //
Query OK, 3 rows affected (0.39 sec)

mysql> delimiter ;
mysql> select * from afterdelcontacts;
+--------+------------+
| userid | mobile     |
+--------+------------+
|      6 | 8945203617 |
|      6 | 7855699512 |
|      6 | 8874912360 |
+--------+------------+
3 rows in set (0.00 sec)
```

CONCLUSION

In this project we implemented the railway management system database system. The system is structured into customer registration, ticket inquiries, online booking, train schedule and their management. System process design and database design is the focus of this system which are clearly and effectively designed by the system architecture and database ER diagram. The efficiency of booking is improved, manual booking errors are reduced, the management of railway passenger transport and customer booking is facilitated. During our database management course we have learned about the basics of database design. This project gave us the opportunity to try our new skills in practice. While doing this project we also gained deeper understanding on database design and how it can be implemented in real life situations. We believe we can use our database designing skills also in other future projects.

REFERENCES

1.Database System Concepts ,***Seventh Edition*** by:Avi Silberschatz,Henry F. Korth,S. Sudarshan

2. Database Systems: A Practical Approach to Design, Implementation, and Management Book by Carolyn Begg and Thomas M. Connolly

3. Database Systems: Design, Implementation, & Management:Carlos Coronel

4. Database Management Systems:Book by Johannes Gehrke and Raghu Ramakrishnan

5. Database Management Systems:Book by Elmasri-Navathe

6.For understanding of railway management structure-https://www.irctc.co.in/nget/

7.https://www.researchgate.net/publication/271604100_Railway_Online_Booking_System_Design_and_Implementation

8.https://www.researchgate.net/publication/268153640_Smart_Computing_Applications_in_Railway_Systems_-_A_case_study_in_Indian_Railways_Passenger_Reservation_System