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

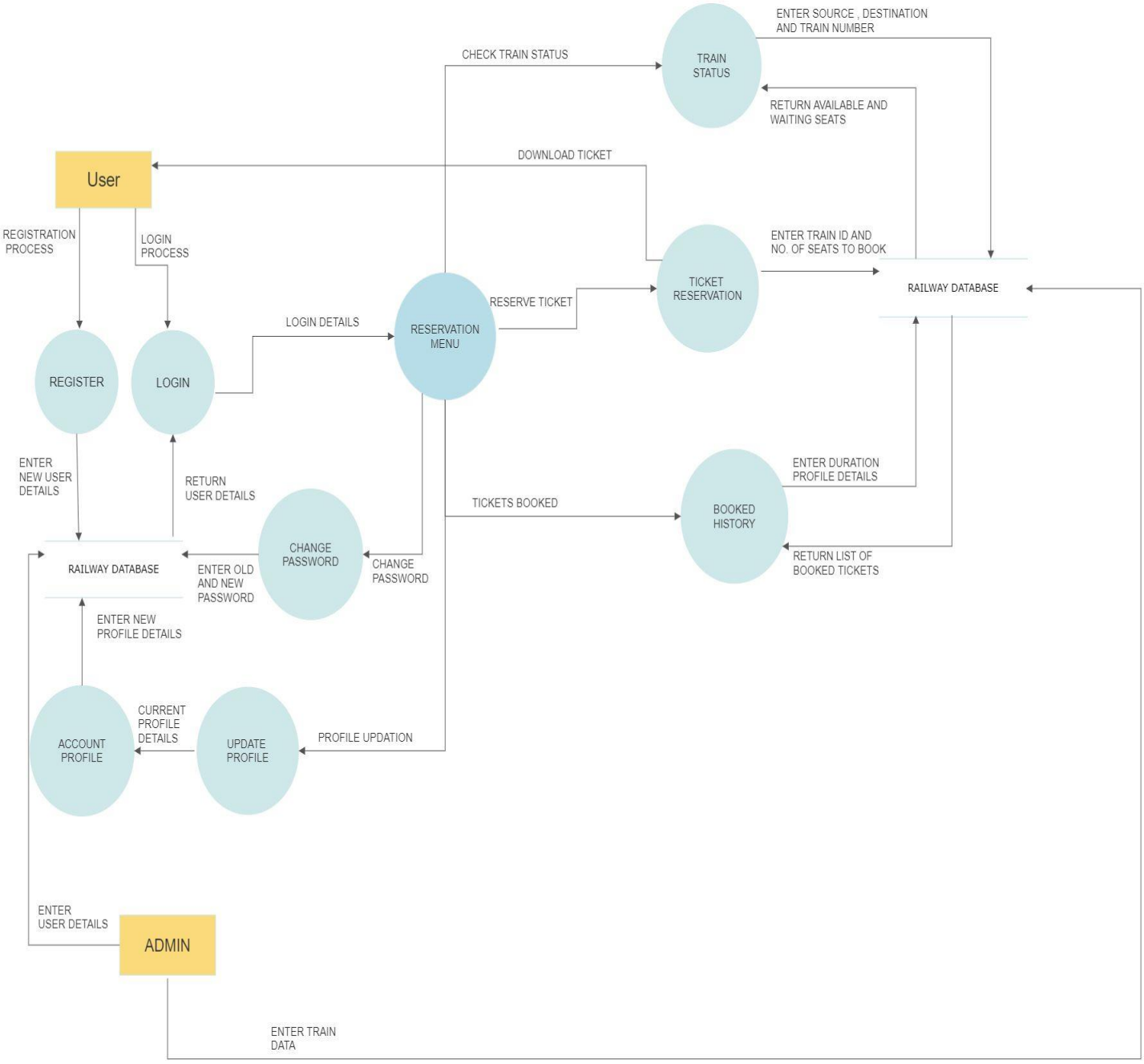
To prepare a database for **Online Railway Booking System** using MYSQL and execute queries and triggers on it.

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.

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).

8. CLASS

   Attributes: Class ID, Sleeper, AC.

RELATIONSHIP WITH ATTRIBUTES:

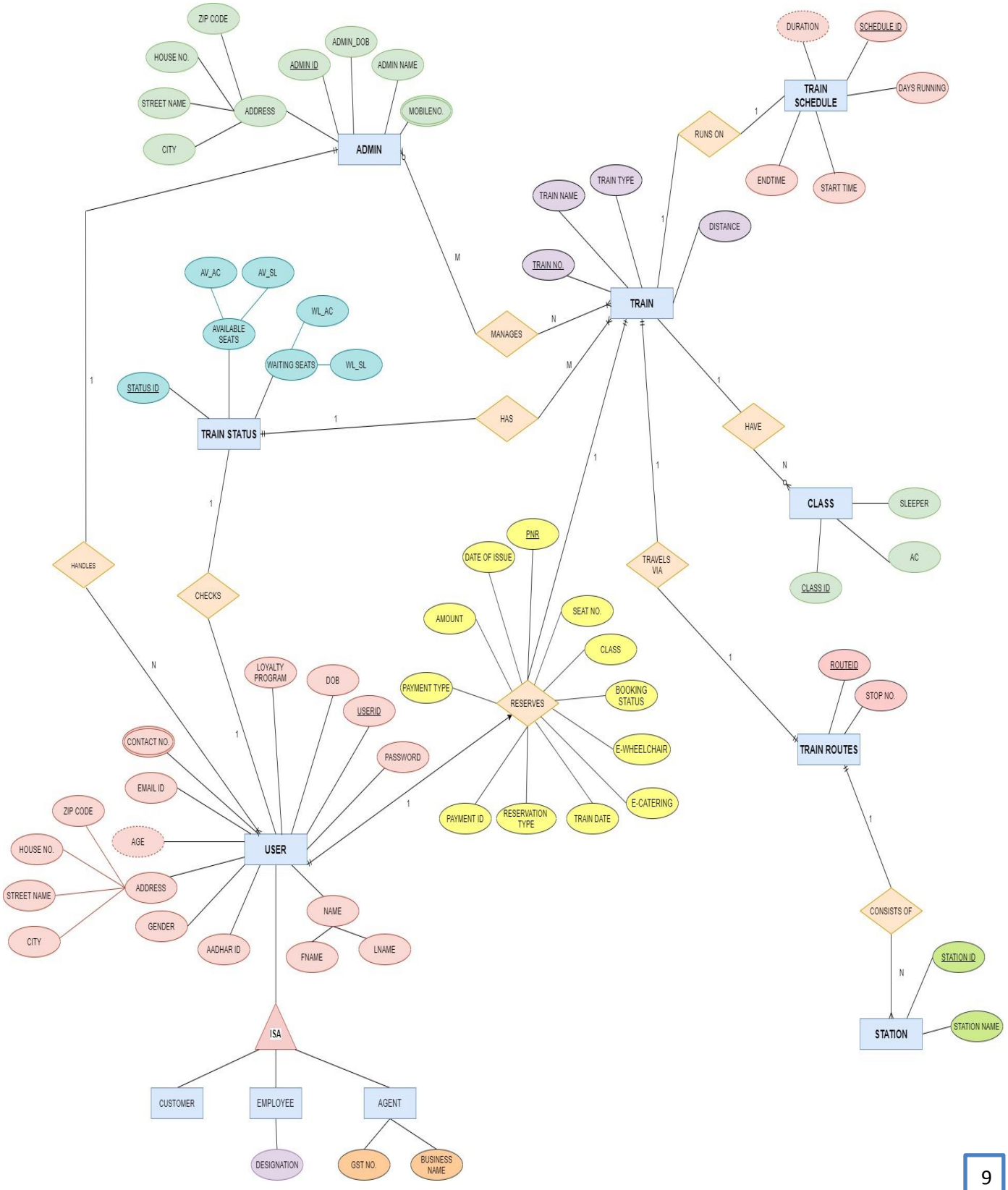
1. Reserves

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

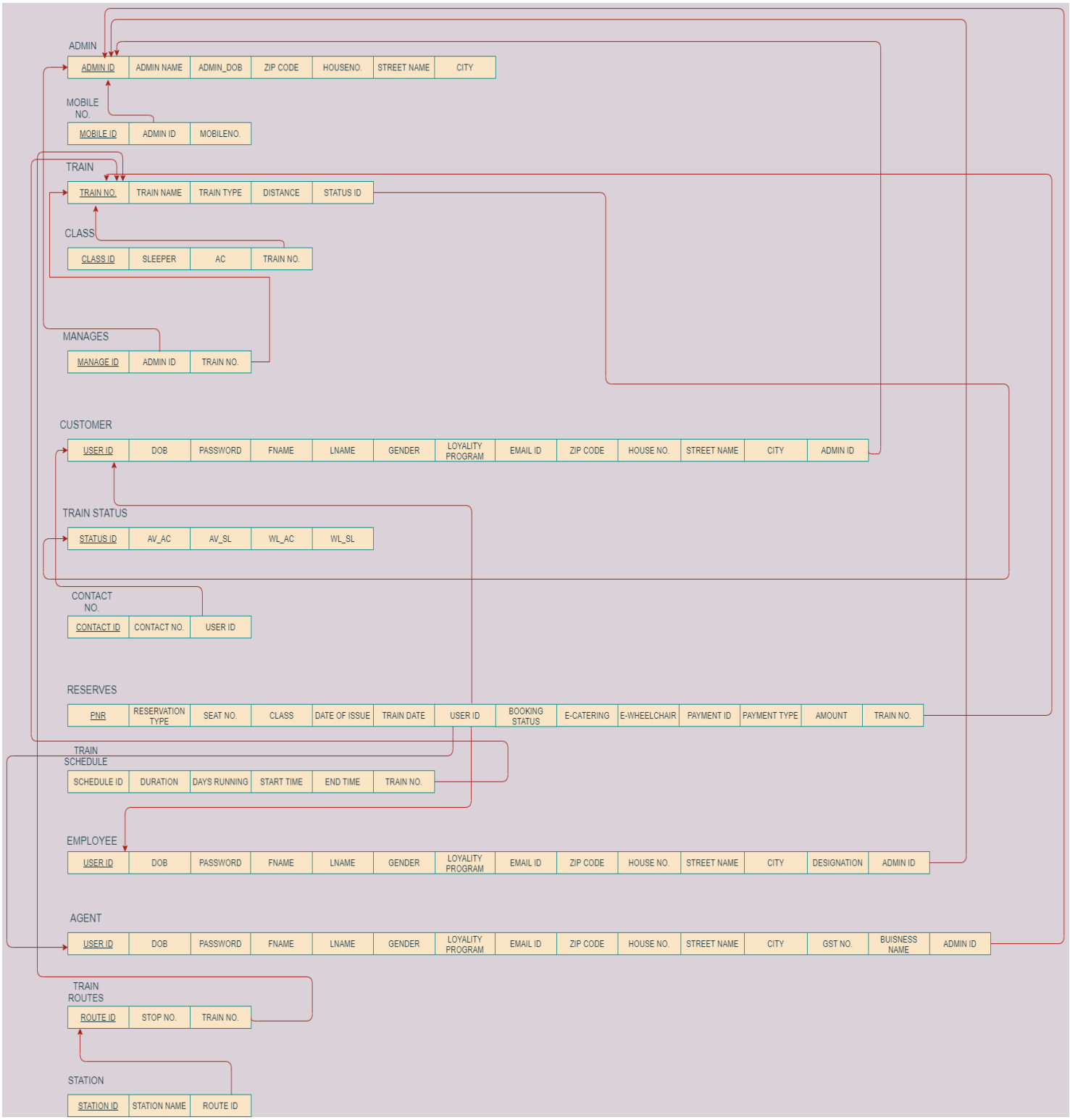
RELATIONSHIPS:

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
CANDIDATE KEY	Status ID
FOREIGN KEY	Train no.

4.**CLASS**:

PRIMARY KEY	Class ID
CANDIDATE KEY	Class ID
FOREIGN KEY	Train no.

5.**USER(CUSTOMER/AGENT/EMPLOYEE)**:

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

6.**TRAIN ROUTES**:

PRIMARY KEY	ROUTEID
CANDIDATE KEY	ROUTEID
FOREIGN KEY	Train no.

7.**STATION**:

PRIMARY KEY	StationID
CANDIDATE KEY	StationID
FOREIGN KEY	RouteID

8.**TRAIN SCHEDULE:**

PRIMARY KEY	ScheduleID
CANDIDATE 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.