**Railway Management System (DBMS)**

**Session:- (2022-23)**

Project submitted to Jawaharlal Nehru University

In partial fulfilment of the requirements

For the award of the degree of

**MASTER OF COMPUTER APPLICATION**

Submitted to:

**Prof. T.V Vijay Kumar**

Submitted by:

**Aditya Kumar (22/10/JC/008)**

**Mohd Arsalan (22/10/JC/030)**

**Pooja Ranjan (22/10/JC/026)**

**Umang Kanchan (22/10/JC/010)**



**SCHOOL OF COMPUTER & SYSTEMS SCIENCES**

**JAWAHARLAL NEHRU UNIVERSITY**

**New Delhi- 110067**

**Index**

**1**[**. Introduction**](#_hot4vcqm1dv1) **3**

[**2. Problem Statement**](#_hot4vcqm1dv1) **4**

**3. Entity Relationship Diagram****5**

**3.1 Introduction**

**3.2 ER diagram for Railway Management System**

**4.Mapping ER to Relational Model****7**

**5. Normalisation (till BCNF)****12**

**5.1 Introduction**

**5.2 Normalising Relations for Railway Management System**

**6. Implement Relations as Tables****19**

**7. SQL Script****22**

**8. System Specification****28**

**9. Conclusion****28**

\*\*\*\*

**INTRODUCTION**

The railway management system faces numerous challenges in efficiently managing the operations, reservations, and maintenance of trains, stations, and associated resources. The current manual processes and lack of an integrated system result in inefficiencies, errors, and delays. Therefore, there is a need to develop a comprehensive Railway Management System to streamline the operations and improve the overall management of the railway network.

The primary challenge is to provide a centralised system that enables efficient scheduling, tracking, and management of trains, routes, and stations. The system should allow for the creation and modification of schedules, considering factors such as train availability, station capacity, and passenger demand. Additionally, the system should facilitate real-time tracking of trains, allowing both railway officials and passengers to monitor the train's progress and arrival times accurately.

Another critical aspect is the reservation and ticketing process. The system should enable passengers to easily search for available trains, book tickets, and make secure online payments. It should provide a user-friendly interface for passengers to select seats, specify preferences, and receive e-tickets. Moreover, the system should handle ticket cancellations, refunds, and maintain a reliable database of passenger records.

Maintenance and resource management are crucial for ensuring safe and reliable train operations. The system should track the maintenance schedules, track repairs, and maintenance history of trains.

Furthermore, the system should generate reports and analytics to aid in decision-making and performance evaluation. It should provide insights into key metrics such as passenger traffic, revenue, on-time performance, and resource utilisation.

\*\*\*\*

**PROBLEM STATEMENT**

From a data perspective, the Railway Management System aims to address various data-related challenges in the management of train operations, reservations, and maintenance. The system aims to streamline data processes, ensure data accuracy, and provide timely and relevant information. Here is a breakdown of the data-related aspects:

**1. Data Integration:** The system will integrate data from various sources such as train schedules, station information, passenger details, and maintenance records. It will ensure that data from different departments and systems are consolidated into a centralised database for easy access and analysis.

**2. Data Management:** The system will implement robust data management practices to ensure data accuracy, consistency, and integrity. This includes data validation, verification, and error handling mechanisms to maintain high data quality standards.

**3. Scheduling and Tracking:** The system will handle scheduling data, including train routes, station availability, and timings. It will track the real-time movement of trains, capturing data such as current location, speed, and estimated arrival times.

**4. Reservation and Ticketing:** The system will manage passenger data, including personal details, ticket bookings, and payment information. It will store and retrieve reservation data, handle ticket cancellations, and maintain a secure transactional database.

**5. Maintenance and Resource Management:** The system will maintain data on train maintenance schedules, repairs, and historical maintenance records. It will track resource allocation, including coach assignments, staff deployment, and equipment availability.

By addressing these data-related challenges, the Railway Management System will enhance data visibility, improve decision-making capabilities, and provide a reliable and efficient platform for managing railway operations.

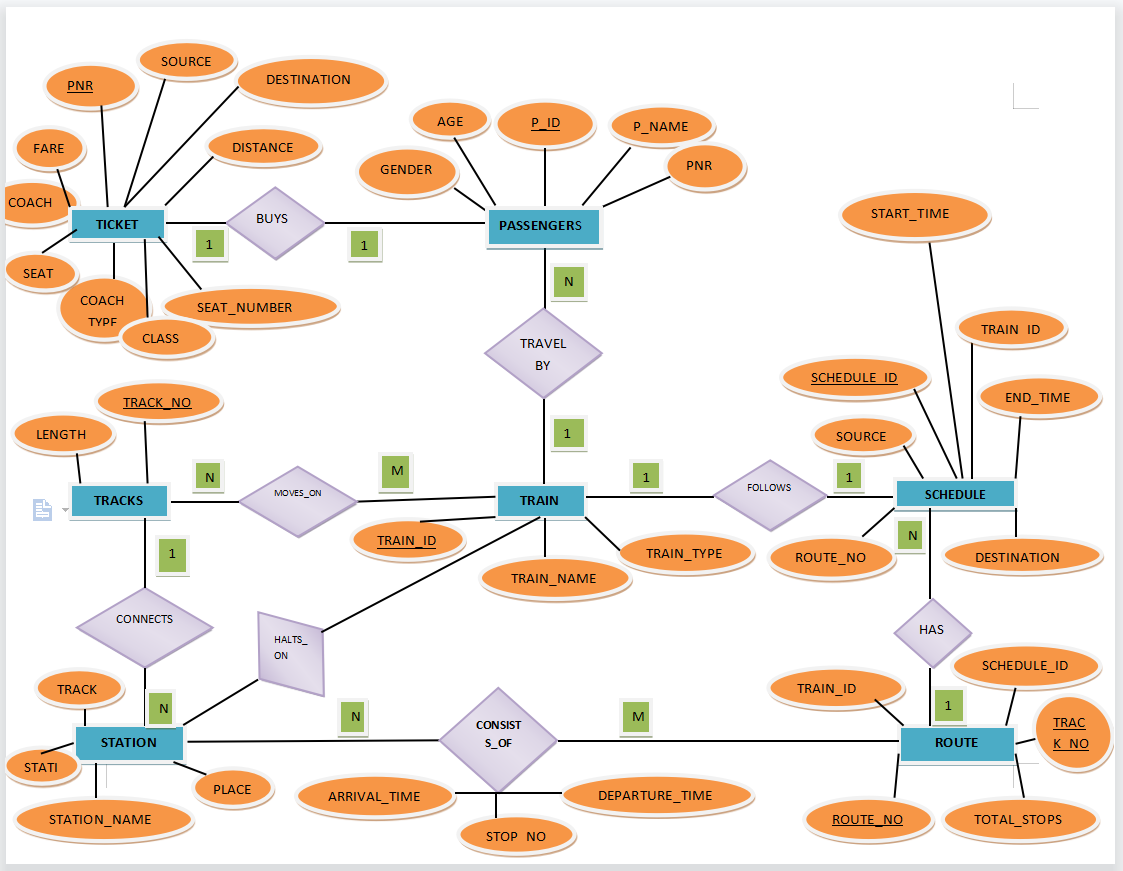
\*\*\*\*

**ER Diagram**

**INTRODUCTION:**

An Entity-Relationship (ER) diagram is a visual representation that depicts the entities, attributes, and relationships within a system or domain. Entities represent real-world objects or concepts, while attributes define the characteristics of these entities. Relationships illustrate the associations between entities and describe how they interact with each other. The ER diagram helps to identify the structure of the database, showcasing the entities as rectangles, attributes as ovals, and relationships as lines connecting the entities. It provides a clear and concise overview of the data model, aiding in database design and understanding the relationships between different entities.

**ER diagram of Railway Management System:**



1. **Entities:**

* Ticket
* Passengers
* Tracks
* Train
* Schedule
* Route
* Station

1. **Relationship Identification:**

* Passenger **BUYS** Ticket
* Passenger **TRAVEL BY** Train
* Train **MOVES\_ON** Tracks
* Tracks **CONNECTS** Station
* Train **FOLLOWS** Schedule
* Route **CONSISTS\_OF** Stations
* Schedule **HAS** Route

1. **Cardinality Identification:**

* **A** ticket is assigned to only **one** passenger
* **Many** passengers travel by **one** train
* **Many** trains moves on **many** track
* **A** train follows **one** schedule
* **A** route has **many** schedule
* **Many** station consists of **many** routes
* **A** track connects **many** station

1. **Identify Attributes:**

| **Entity** | **Primary Key** | **Attribute** |
| --- | --- | --- |
| Ticket | Ticket\_No | Fare, Source, Destination, Distance, PNR, Class, Seat\_No, Date, Time |
| Passengers | P\_ID | P\_Name, Ticket\_No, Seat\_Number, Coach\_type, Train\_ID, Coaches, Gender, Age |
| Tracks | Track\_No | Length |
| Train | Train\_ID | Train\_Name, Train\_Type |
| Schedule | Schedule\_ID | Source, Start\_time, Train\_Id, End\_time, Destination, Route\_no |
| Route | Route\_No | Train\_Id, Schedule\_Id, Total\_stops |
| Station | Station\_Id | Track\_no, place, station\_name |

\*\*\*\*

**MAPPING ER TO RELATIONAL MODEL**

**INTRODUCTION:**

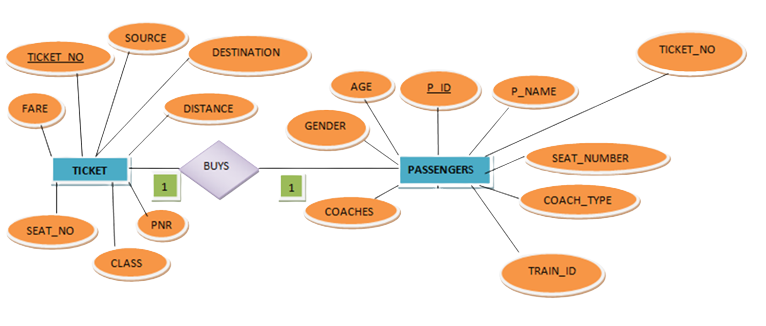
Mapping an ER (Entity-Relationship) diagram to a relational model involves translating the entities, attributes, and relationships into tables, columns, and relationships in a relational database.

In the mapping process, each entity in the ER diagram becomes a table in the relational model, with each attribute becoming a column in that table. The primary key of the entity becomes the primary key column of the corresponding table.

For relationships, there are a few mapping options depending on the cardinality and participation constraints:

**MAPPINGS:**

1. **Passenger buy Ticket (one - to - one)**

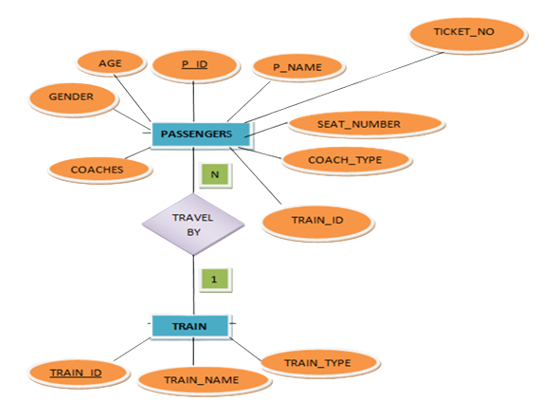
****

Relation:

**PASSENGERS(**P\_ID, P\_NAME, TICKET,\_NO, SEAT\_NUMBER, COACH\_TYPE, TRAIN\_ID, COACHES, GENDER, AGE)

**TICKET(**TICKET\_NO, SOURCE, DESTINATION, FARE, DISTANCE, SEAT\_NO, CLASS, PNR,DATE, TIME)

1. **Passengers Travel\_by Train (many - to - one)**

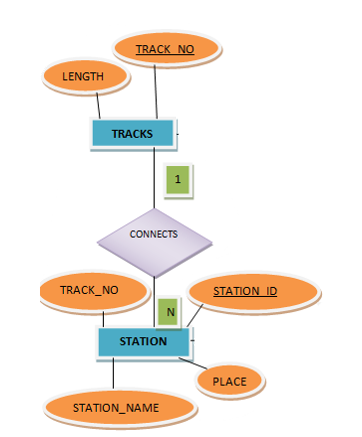
****

Relation:

**PASSENGERS(**P\_ID, P\_NAME, TICKET,\_NO, SEAT\_NUMBER, COACH\_TYPE, TRAIN\_ID, COACHES, GENDER, AGE)

**TRAIN**(TRAIN\_ID,TRAIN\_NAME,TRAIN\_TYPE)

1. **Tracks connects station (one-to-many)**

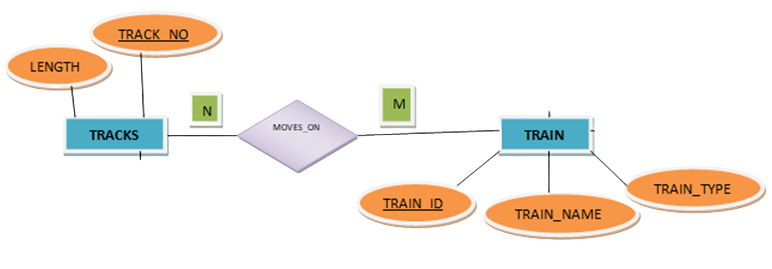
****

Relation:

**TRACKS**(TRACK\_NO,LENGTH)

**STATION**(STATION\_ID,PLACE,STATION\_NAME,TRACK\_NO)

1. **Train moves on tracks (many- to - many)**

****

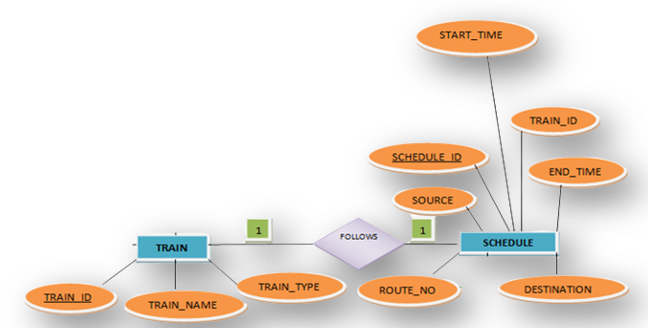
Relations:

**TRAIN**(TRAIN\_ID,TRAIN\_NAME,TRAIN\_TYPE)

**MOVES\_ON**(TRAIN\_ID,TRACK\_NO)

**TRACKS**(TRACK\_NO,LENGTH)

1. **Train follows Schedule (one - to - one)**

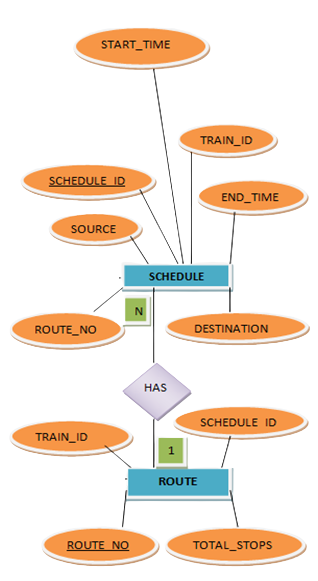


Relation:

**TRAIN**(TRAIN\_ID,TRAIN\_NAME,TRAIN\_TYPE)

**SCHEDULE**(SCHEDULE\_ID,START\_TIME,TRAIN\_ID,END\_TIME, SOURCE,DESTINATION,ROUTE\_NO)

1. **Route has schedule (one - to - many)**

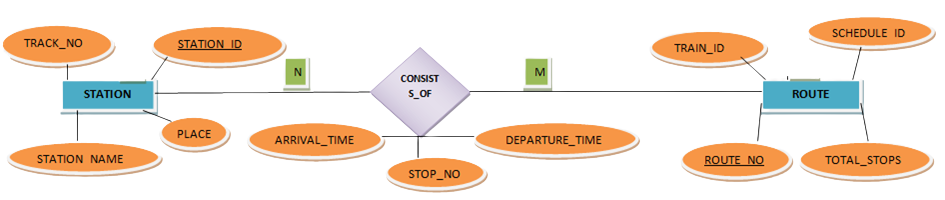


Relations;

**SCHEDULE**(SCHEDULE\_ID,START\_TIME,TRAIN\_ID,END\_TIME, SOURCE,DESTINATION,ROUTE\_NO)

**ROUTE**(ROUT\_NO,TOTAL\_STOPS)

1. **Station consists of route (many - to - many)**

****

Relations:

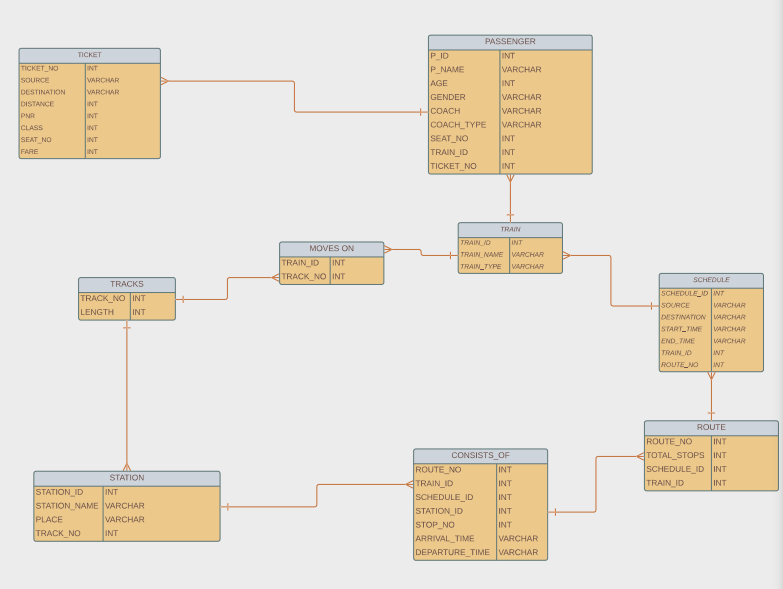
**ROUTE**(ROUTE\_NO,TOTAL\_STOPS,SCHEDULE\_ID,TRAIN\_ID)

**CONSISTS\_OF** (ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID, STOP\_NO, ARRIVAL\_TIME, DEPARTURE\_TIME)

**STATION**(STATION\_ID,PLACE,STATION\_NAME,TRACK\_NO)

**Final Relations in Relational Model:**

* **TRAIN**(TRAIN\_ID,TRAIN\_NAME,TRAIN\_TYPE)
* **TRACKS**(TRACK\_NO,LENGTH)
* **STATION**(STATION\_ID,PLACE,STATION\_NAME,TRACK\_NO)
* **SCHEDULE**(SCHEDULE\_ID,START\_TIME,TRAIN\_ID,END\_TIME, SOURCE,DESTINATION,ROUTE\_NO)
* **ROUTE**(ROUT\_NO,TOTAL\_STOPS)
* **CONSISTS\_OF** (ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID, STOP\_NO, ARRIVAL\_TIME, DEPARTURE\_TIME)
* **TICKET**(TICKET\_NO,SOURCE,DESTINATION,FARE,DISTANCE,SEAT\_NO,CLASS,PNR,DATE, TIME)
* **PASSENGERS**(P\_ID,P\_NAME,TICKET\_NO,SEAT\_NUMBER,COACH\_TYPE,TRAIN\_ID,COACHES,GENDER,AGE)
* **MOVES\_ON**(TRAIN\_ID,TRACK\_NO)



**Fig: Relational Model**

\*\*\*\*

**NORMALISATION**

**INTRODUCTION:**

Normalisation is the process of organising and structuring a relational database to eliminate data redundancy, dependency, and anomalies. There are several normal forms, and one of the most commonly used is the Boyce-Codd Normal Form (BCNF).

The normalisation process typically involves the following steps:

1**. First Normal Form (1NF)**: Ensure that each attribute in a table **contains atomic values,** meaning it cannot be further divided. This eliminates repeating groups and ensures uniqueness.

2. **Second Normal Form (2NF)**: **Remove partial dependencies** by creating separate tables for subsets of attributes that depend on a part of the primary key.

3. **Third Normal Form (3NF**): E**liminate transitive dependencies** by creating separate tables for attributes that depend on non-key attributes.

4. **Boyce-Codd Normal Form (BCNF)**: Further eliminate functional dependencies by ensuring that each determinant (set of attributes that uniquely determines other attributes) is a candidate key.

For BCNF, we need to decompose tables further and create additional tables based on functional dependencies. This process ensures that all non-key attributes in a table are functionally dependent on the entire candidate key.

The goal of normalisation is to minimise data redundancy and ensure data integrity by organising the data into separate tables based on functional dependencies. This allows for efficient storage, reduces data anomalies, and simplifies data maintenance and updates.

**NORMALISING RELATIONS FOR RAILWAY MANAGEMENT SYSTEM:**

* **TRAIN**(TRAIN\_ID,TRAIN\_NAME,TRAIN\_TYPE)

**FUNCTIONAL DEPENDENCY:**

{TRAIN\_ID -> {TRAIN\_ID,TRAIN\_NAME,TRAIN\_TYPE}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF.**

SINCE TRAIN\_ID IS **PRIMARY KEY**, THERE IS **NO PARTIAL DEPENDENCY** AND RELATION IS ALREADY IN **2NF**.

SINCE TRAIN\_ID IS **PRIMARY KEY**, THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF**.

THE RELATION IS IN 1NF,2NF,3NF AND TRAIN\_ID IS **PRIMARY KEY**, THE RELATION IS IN **BCNF**.

SO THE RELATION IS IN **BCNF**

* **TRACKS**(TRACK\_NO,LENGTH)

**FUNCTIONAL DEPENDENCY:**

{TRACK\_NO ->{TRACK\_NO,LENGTH} }

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF**.

SINCE TRACK\_NO IS **PRIMARY KEY**, THERE IS **NO PARTIAL DEPENDENCY** AND RELATION IS ALREADY IN **2NF**.

SINCE TRACK\_NO IS PRIMARY KEY, THERE IS NO **TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF.**

THE RELATION IS IN 1NF,2NF,3NF AND TRACK\_NO IS **PRIMARY KEY**, THE RELATION IS IN **BCNF**.

SO THE RELATION IS IN **BCNF**

* **STATION**(STATION\_ID,PLACE,STATION\_NAME,TRACK\_NO)

**FUNCTIONAL DEPENDENCY:**

{STATION\_ID -> {STATION\_ID,PLACE,STATION\_NAME,TRACK\_NO}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF.**

SINCE STATION\_ID IS **PRIMARY KEY,** THERE IS **NO PARTIAL DEPENDENCY** AND RELATION IS ALREADY IN **2NF.**

SINCE STATION\_ID IS **PRIMARY KEY**, THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF.**

THE RELATION IS IN 1NF,2NF,3NF AND STATION\_ID IS **PRIMARY KEY,** THE RELATION IS IN **BCNF.**

SO THE RELATION IS IN **BCNF**

* **SCHEDULE**(SCHEDULE\_ID,START\_TIME,TRAIN\_ID,END\_TIME, SOURCE,DESTINATION,ROUTE\_NO)

**FUNCTIONAL DEPENDENCY:**

{SCHEDULE\_ID -> {SCHEDULE\_ID,START\_TIME,TRAIN\_ID,END\_TIME, SOURCE,DESTINATION,ROUTE\_NO}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF**.

SINCE SCHEDULE\_ID IS **PRIMARY KEY**, THERE IS **NO PARTIAL DEPENDENCY** AND RELATION IS ALREADY IN **2NF**.

SINCE SCHEDULE\_ID IS **PRIMARY KEY**, THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF.**

THE RELATION IS IN 1NF,2NF,3NF AND SCHEDULE\_ID IS **PRIMARY KEY**, THE RELATION IS IN **BCNF.**

SO THE RELATION IS IN **BCNF**

* **ROUTE**(ROUT\_NO,TOTAL\_STOPS)

**FUNCTIONAL DEPENDENCY:**

{ROUT\_NO -> {ROUT\_NO,TOTAL\_STOPS}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF.**

SINCE ROUT\_NO IS **PRIMARY KEY**, THERE IS **NO PARTIAL DEPENDENCY AND RELATION** IS ALREADY IN **2NF**.

SINCE ROUT\_NO IS **PRIMARY KEY**, THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF**.

THE RELATION IS IN 1NF,2NF,3NF AND ROUT\_NO IS **PRIMARY KEY**, THE RELATION IS IN **BCNF**.

SO THE RELATION IS IN **BCNF**

* **CONSISTS\_OF** (ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID, STOP\_NO, ARRIVAL\_TIME, DEPARTURE\_TIME)

**FUNCTIONAL DEPENDENCY:**

{{ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID} –-> {ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID, STOP\_NO, ARRIVAL\_TIME, DEPARTURE\_TIME}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES,** RELATION IS IN **1NF.**

SINCE {ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID} IS **PRIMARY KEY**, THERE IS **NO PARTIAL DEPENDENCY** AND RELATION IS ALREADY IN **2NF.**

SINCE {ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID} IS **PRIMARY KEY,** THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF**.

THE RELATION IS IN 1NF,2NF,3NF AND {ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID} IS **PRIMARY KEY,** THE RELATION IS **IN BCNF**.

SO THE RELATION IS IN **BCNF**

* **TICKET**(TICKET\_NO,SOURCE,DESTINATION,FARE,DISTANCE,SEAT\_NO,CLASS,PNR,DATE, TIME)

**FUNCTIONAL DEPENDENCY:**

{{TICKET\_NO} -> {TICKET\_NO,SOURCE,DESTINATION,FARE,DISTANCE,SEAT\_NO,CLASS,PNR}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF.**

SINCE TICKET\_NO IS **PRIMARY KEY,** THERE IS **NO PARTIAL DEPENDENCY** AND RELATION IS ALREADY IN **2NF**.

SINCE TICKET\_NO IS **PRIMARY KEY**, THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF.**

THE RELATION IS IN 1NF,2NF,3NF AND TICKET\_NO IS **PRIMARY KEY**, THE RELATION IS IN **BCNF**.

SO THE RELATION IS IN **BCNF**

* **PASSENGERS**(P\_ID,P\_NAME,TICKET\_NO,SEAT\_NUMBER,COACH\_TYPE,TRAIN\_ID,COACHES,GENDER,AGE)

**FUNCTIONAL DEPENDENCY:**

{P\_ID -> {P\_ID,P\_NAME,TICKET\_NO,SEAT\_NUMBER,COACH\_TYPE,TRAIN\_ID,COACHES,GENDER,AGE}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF**.

SINCE P\_ID IS **PRIMARY KEY,** THERE IS **NO PARTIAL DEPENDENCY** AND RELATION IS ALREADY IN **2NF**.

SINCE P\_ID IS **PRIMARY KEY**, THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF.**

THE RELATION IS IN 1NF,2NF,3NF AND P\_ID IS **PRIMARY KEY,** THE RELATION IS IN **BCNF.**

SO THE RELATION IS IN **BCNF**

* **MOVES\_ON**(TRAIN\_ID,TRACK\_NO)

**FUNCTIONAL DEPENDENCY:**

{{TRAIN\_ID,TRACK\_NO} -> {TRAIN\_ID,TRACK\_NO}}

SINCE ALL ATTRIBUTES HAVE **ATOMIC VALUES**, RELATION IS IN **1NF.**

SINCE {TRAIN\_ID,TRACK\_NO} IS **PRIMARY KEY**, THERE IS NO PARTIAL DEPENDENCY AND RELATION IS ALREADY IN **2NF.**

SINCE {TRAIN\_ID,TRACK\_NO} IS **PRIMARY KEY**, THERE IS **NO TRANSITIVE DEPENDENCY** AND RELATION IS ALREADY IN **3NF**.

THE RELATION IS IN 1NF,2NF,3NF AND {TRAIN\_ID,TRACK\_NO} IS **PRIMARY KEY**, THE RELATION IS IN **BCNF.**

SO THE RELATION IS IN **BCNF**

| **Relation** | **Primary Key** | **Attributes** | **Functional dependency** | **Normal Form** |
| --- | --- | --- | --- | --- |
| Ticket | Ticket\_No | Fare, Source, Destination, Distance, PNR, Class, Seat\_No, Date, Time | **TICKET NO -> R( TICKET NO is superkey)** | **BCNF** |
| Passengers | P\_ID | P\_Name, Ticket\_No, Seat\_Number, Coach\_type, Train\_ID, Coaches, Gender, Age | **P\_ID -> R( P\_ID) is a superkey)** | **BCNF** |
| Tracks | Track\_No | Length | **TRACK\_NO -> R(TRACK\_NO is a superkey)** | **BCNF** |
| Train | Train\_ID | Train\_Name, Train\_Type | **T\_ID-> R(T\_ID is a superkey)** | **BCNF** |
| Moves\_On | Train\_ID, Track\_No | Train\_ID, Track\_No | **TRAIN\_ID, TRACK\_NO -> R(TRAIN\_ID, TRACK\_NO IS A SUPER KEY)** | **BCNF** |
| Schedule | Schedule\_ID | Source, Start\_time, Train\_Id, End\_time, Destination, Route\_no | **SCHEDULE\_ID -> R(SCHEDULE\_ID is a superkey)**  **TRAIN\_ID -> R(TRAIN\_ID is a superkey)**  **ROUTE\_NO ->R(ROUTE\_NO is a superkey)** | **BCNF** |
| Route | Route\_No | Train\_Id, Schedule\_Id, Total\_stops | **ROUTE\_NO->R(ROUTE\_NO Is a superkey)**  **SCHEDULE\_ID->R(SCHEDULE\_NO is a superkey)**  **TRAIN\_ID->R(TRAIN\_ID Is a superkey)** | **BCNF** |
| Station | Station\_Id | Track\_no, place, station\_name | **Station\_ID -> R(Station\_ID is a superkey)** | **BCNF** |
| Consists\_Of | Route\_No, Train\_Id, Schedule\_Id, Station\_Id | Stop\_No, Arrival\_Time, Departure\_Time | **ROUTE\_NO->R( ROUTE\_NO is a superkey)**  **TRAIN\_ID->R( TRAIN\_ID is a superkey)**  **SCHEDULE\_ID->R(SCHEDULE\_ID is a superkey)**  **STATION\_ID->R(STATION\_ID-> is a superkey)** | **BCNF** |

Above relations are already in BCNF normal form. So, there is no need to normalise further.

\*\*\*\*

**IMPLEMENT RELATIONS AS TABLES**

**1.Train Table**

CREATE TABLE TRAIN

(

TRAIN\_ID INT NOT NULL PRIMARY KEY,

TRAIN\_NUM VARCHAR(50),

TRAIN\_NAME VARCHAR(50),

TRAIN\_TYPE VARCHAR(50)

);

**2.Tracks Table**

CREATE TABLE TRACKS

(

TRACK\_NO INT NOT NULL PRIMARY KEY,

LENGTH VARCHAR(50)

);

**3.Station Table**

CREATE TABLE STATION

(

STATION\_ID INT NOT NULL PRIMARY KEY,

STATION\_NAME VARCHAR(50),

PLACE VARCHAR(50),

TRACK\_NO INT,

FOREIGN KEY (TRACK\_NO) REFERENCES TRACKS(TRACK\_NO) ON DELETE CASCADE

);

**4.Schedules Table**

CREATE TABLE SCHEDULES

(

SCHEDULES\_ID INT NOT NULL,

SOURCE VARCHAR(50),

DESTINATION VARCHAR(50),

START\_TIME VARCHAR(50),

END\_TIME VARCHAR(50),

TRAIN\_ID INT NOT NULL,

TRAIN\_NUM VARCHAR(50),

ROUTE\_NO INT NOT NULL,

PRIMARY KEY (SCHEDULES\_ID, TRAIN\_ID, ROUTE\_NO),

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE

);

**5.Route Table**

CREATE TABLE ROUTE

(

ROUTE\_NO INT NOT NULL,

TOTAL\_STOPS INT,

SCHEDULE\_ID INT NOT NULL,

TRAIN\_ID INT NOT NULL,

PRIMARY KEY (ROUTE\_NO, SCHEDULE\_ID, TRAIN\_ID),

FOREIGN KEY (SCHEDULE\_ID) REFERENCES SCHEDULES(SCHEDULES\_ID) ON DELETE CASCADE,

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE

);

**6. Consists\_of Table**

CREATE TABLE CONSISTS\_OF

(

ROUTE\_NO INT NOT NULL,

TRAIN\_ID INT NOT NULL,

SCHEDULE\_ID INT NOT NULL,

STATION\_ID INT NOT NULL,

STOP\_NO INT,

ARRIVAL\_TIME VARCHAR(50),

DEPARTURE\_TIME VARCHAR(50),

PRIMARY KEY (ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID),

FOREIGN KEY (ROUTE\_NO, SCHEDULE\_ID, TRAIN\_ID) REFERENCES ROUTE(ROUTE\_NO, SCHEDULE\_ID, TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (STATION\_ID) REFERENCES STATION(STATION\_ID) ON DELETE CASCADE

);

**7. Ticket Table**

CREATE TABLE TICKET

(

TICKET\_NO INT NOT NULL PRIMARY KEY,

FARE VARCHAR(50),

SOURCE VARCHAR(50),

DATE VARCHAR(50),

TIME VARCHAR(50),

DESTINATION VARCHAR(50),

DISTANCE VARCHAR(50),

SEAT\_NO VARCHAR(50),

PNR VARCHAR(50)

);

**8. Passengers Table**

CREATE TABLE PASSENGERS

(

P\_ID INT NOT NULL PRIMARY KEY,

P\_NAME VARCHAR(50),

AGE INT,

GENDER VARCHAR(50),

COACH INT,

COACH\_TYPE VARCHAR(50),

SEAT\_NO VARCHAR(50),

TRAIN\_ID INT,

TICKET\_NO INT,

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (TICKET\_NO) REFERENCES TICKET(TICKET\_NO) ON DELETE CASCADE

);

**9. Moves\_On Table**

CREATE TABLE MOVES\_ON

(

TRAIN\_ID INT NOT NULL,

TRACK\_NO INT NOT NULL,

PRIMARY KEY (TRAIN\_ID, TRACK\_NO),

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (TRACK\_NO) REFERENCES TRACKS(TRACK\_NO) ON DELETE CASCADE

);

\*\*\*\*

**SQL Script**

CREATE DATABASE RAILWAY;

USE RAILWAY;

CREATE TABLE TRAIN

(

TRAIN\_ID INT NOT NULL PRIMARY KEY,

TRAIN\_NUM VARCHAR(50),

TRAIN\_NAME VARCHAR(50),

TRAIN\_TYPE VARCHAR(50)

);

CREATE TABLE TRACKS

(

TRACK\_NO INT NOT NULL PRIMARY KEY,

LENGTH VARCHAR(50)

);

CREATE TABLE STATION

(

STATION\_ID INT NOT NULL PRIMARY KEY,

STATION\_NAME VARCHAR(50),

PLACE VARCHAR(50),

TRACK\_NO INT,

FOREIGN KEY (TRACK\_NO) REFERENCES TRACKS(TRACK\_NO) ON DELETE CASCADE

);

CREATE TABLE SCHEDULES

(

SCHEDULES\_ID INT NOT NULL,

SOURCE VARCHAR(50),

DESTINATION VARCHAR(50),

START\_TIME VARCHAR(50),

END\_TIME VARCHAR(50),

TRAIN\_ID INT NOT NULL,

TRAIN\_NUM VARCHAR(50),

ROUTE\_NO INT NOT NULL,

PRIMARY KEY (SCHEDULES\_ID, TRAIN\_ID, ROUTE\_NO),

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE

);

CREATE TABLE ROUTE

(

ROUTE\_NO INT NOT NULL,

TOTAL\_STOPS INT,

SCHEDULE\_ID INT NOT NULL,

TRAIN\_ID INT NOT NULL,

PRIMARY KEY (ROUTE\_NO, SCHEDULE\_ID, TRAIN\_ID),

FOREIGN KEY (SCHEDULE\_ID) REFERENCES SCHEDULES(SCHEDULES\_ID) ON DELETE CASCADE,

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE

);

CREATE TABLE CONSISTS\_OF

(

ROUTE\_NO INT NOT NULL,

TRAIN\_ID INT NOT NULL,

SCHEDULE\_ID INT NOT NULL,

STATION\_ID INT NOT NULL,

STOP\_NO INT,

ARRIVAL\_TIME VARCHAR(50),

DEPARTURE\_TIME VARCHAR(50),

PRIMARY KEY (ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID),

FOREIGN KEY (ROUTE\_NO, SCHEDULE\_ID, TRAIN\_ID) REFERENCES ROUTE(ROUTE\_NO, SCHEDULE\_ID, TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (STATION\_ID) REFERENCES STATION(STATION\_ID) ON DELETE CASCADE

);

CREATE TABLE TICKET

(

TICKET\_NO INT NOT NULL PRIMARY KEY,

FARE VARCHAR(50),

SOURCE VARCHAR(50),

DATE VARCHAR(50),

TIME VARCHAR(50),

DESTINATION VARCHAR(50),

DISTANCE VARCHAR(50),

SEAT\_NO VARCHAR(50),

PNR VARCHAR(50)

);

CREATE TABLE PASSENGERS

(

P\_ID INT NOT NULL PRIMARY KEY,

P\_NAME VARCHAR(50),

AGE INT,

GENDER VARCHAR(50),

COACH INT,

COACH\_TYPE VARCHAR(50),

SEAT\_NO VARCHAR(50),

TRAIN\_ID INT,

TICKET\_NO INT,

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (TICKET\_NO) REFERENCES TICKET(TICKET\_NO) ON DELETE CASCADE

);

CREATE TABLE MOVES\_ON

(

TRAIN\_ID INT NOT NULL,

TRACK\_NO INT NOT NULL,

PRIMARY KEY (TRAIN\_ID, TRACK\_NO),

FOREIGN KEY (TRAIN\_ID) REFERENCES TRAIN(TRAIN\_ID) ON DELETE CASCADE,

FOREIGN KEY (TRACK\_NO) REFERENCES TRACKS(TRACK\_NO) ON DELETE CASCADE

);

INSERT INTO TRAIN (TRAIN\_ID,TRAIN\_NUM, TRAIN\_NAME, TRAIN\_TYPE)

VALUES (1,'12951/12952', 'Rajdhani EXpress', 'Superfast, air\_conditioner'),

(2, '12013/12014', 'Shatabdi Express', 'Superfast, air\_conditioner'),

(3, '12261/12262', 'Duronto Express', 'air\_conditioner, non stop'),

(4, '12049/12050', 'Gatimaan Express', 'SEMI-high, air\_conditioner'),

(5, '22119/22120', 'Tejas Express', 'High-Speed, air\_conditioner'),

(6, '12571/12572', 'Hamsafar Express', 'High-Speed, fully air\_conditioner'),

(7, '12055/12056', 'Jan Shatabdi Express', 'Superfast, air\_conditioner'),

(8, '12907/12908', 'Sampark Kranti', 'Superfast'),

(9, '12023/12204', 'Garib Rath Express', 'Superfast, air\_conditioner'),

(10,'22417/22418', 'Mahamana Express', 'Premium air\_conditioner\*-');

INSERT INTO TRACKS (TRACK\_NO, LENGTH)

VALUES (12, '100 km'),

(20, '560 km'),

(10, '160 km'),

(34, '1200 km'),

(53, '791 km'),

(61, '911 km'),

(73, '500 km'),

(36, '750 km'),

(67, '250 km'),

(90, '300 km');

INSERT INTO STATION (STATION\_ID, STATION\_NAME, PLACE, TRACK\_NO)

VALUES (1, 'Chhatrapati Shivaji Terminal', 'Mumbai, Maharashtra', 12),

(2, 'Howrah Junction', 'Kolkata, West Bengal', 20),

(3, 'New Delhi Railway Station', 'New Delhi', 10),

(4, 'Chennai Central', 'Chennai, TamilNadu', 34),

(5, 'Bangalore City Junction', 'Bangalore, Karnataka', 53),

(6, 'Patna Junction', 'Patna, Bihar', 61),

(7, 'Ahmedaba Junction', 'Ahmedabad, Gujarat', 73),

(8, 'Secunderabad Junction', 'Hyderabad, Telangana', 36),

(9, 'Kolkata Station', 'Kolkata, West Bengal', 67),

(10, 'Lucknow Junction', 'Lucknow, Uttar Pradesh', 90);

INSERT INTO SCHEDULES (SCHEDULES\_ID, SOURCE, DESTINATION, START\_TIME, END\_TIME,TRAIN\_NUM, TRAIN\_ID, ROUTE\_NO)

VALUES (1, 'New Delhi', 'Mumbai', '08:00', '19:00', '12951/12952',1, 1),

(2, 'New Delhi', 'Jaipur', '09:00', '17:00', '12013/12014',2, 2),

(3, 'Mumbai', 'Kolkata','10:00', '23:00','12261/12262',3, 3),

(4, 'New Delhi', 'Jhansi', '11:00', '13:00', '12049/12050',4, 4),

(5, 'Lucknow', 'New Delhi','12:00', '22:00','22119/22122',5, 5),

(6, 'Patna', 'Ahmedabad','9:00', '15:00','12971/12572',6, 6),

(7, 'Lucknow', 'Varansi', '14:00', '16:00', '12055/12056',7, 7),

(8, 'Patna', 'Delhi', '15:00', '17:00', '12907/12908', 8, 8),

(9, 'Mumbai', 'Kolkata', '16:00', '18:00', '12203/12204',9, 9),

(10,'Varansi', 'New Delhi', '17:00', '19:0' ,'22417/22418',10, 10);

INSERT INTO ROUTE (ROUTE\_NO, TOTAL\_STOPS, SCHEDULE\_ID, TRAIN\_ID)

VALUES (1, 5, 1, 1),

(2, 5, 2, 2),

(3, 5, 3, 3),

(4, 5, 4, 4),

(5, 5, 5, 5),

(6, 5, 6, 6),

(7, 5, 7, 7),

(8, 5, 8, 8),

(9, 5, 9, 9),

(10, 5, 10, 10);

INSERT INTO CONSISTS\_OF (ROUTE\_NO, TRAIN\_ID, SCHEDULE\_ID, STATION\_ID, STOP\_NO, ARRIVAL\_TIME, DEPARTURE\_TIME)

VALUES (1, 1, 1, 1, 1, '08:00', '08:15'),

(2, 2, 2, 2, 2, '09:30', '09:35'),

(3, 3, 3, 4, 4, '11:30', '11:35'),

(3, 3, 3, 5, 5, '12:00', '12:15'),

(4, 4, 4, 5, 5, '13:00', '13:15'),

(5, 5, 5, 1, 1, '12:00', '12:15'),

(6, 6, 6, 4, 4, '14:30', '14:35'),

(6, 6, 6, 5, 5, '15:00', '15:15'),

(7, 7, 7, 1, 1, '14:00', '14:15'),

(9, 9, 9, 1, 1, '16:00', '16:15'),

(9, 9, 9, 2, 2, '16:30', '16:35'),

(10, 10, 10, 5, 5, '19:00', '19:15');

INSERT INTO TICKET (TICKET\_NO, FARE, SOURCE, DESTINATION, DISTANCE, SEAT\_NO, PNR)

VALUES (1, '50', 'New Delhi', 'Mumbai', '100 km', 'A1', 'PNR1'),

(2, '60', 'New Delhi', 'Jaipur', '120 km', 'B1', 'PNR2'),

(3, '55', 'Mumbai', 'Kolkata', '110 km', 'C1', 'PNR3'),

(4, '65', 'New Delhi', 'Jhansi', '130 km', 'D1', 'PNR4'),

(5, '70', 'Lucknow', 'New Delhi', '140 km', 'E1', 'PNR5'),

(6, '75', 'Patna', 'Ahmedabad', '150 km', 'F1', 'PNR6'),

(7, '80', 'Lucknow', 'Varansi', '160 km', 'G1', 'PNR7'),

(8, '85', 'Patna', 'Delhi', '170 km', 'H1', 'PNR8'),

(9, '90', 'Mumbai', 'Kolkata', '180 km', 'I1', 'PNR9'),

(10, '95', 'Varansi', 'New Delhi', '190 km', 'J1', 'PNR10');

INSERT INTO PASSENGERS (P\_ID, P\_NAME, AGE, GENDER, COACH, COACH\_TYPE, SEAT\_NO, TRAIN\_ID, TICKET\_NO)

VALUES (1, 'AMITABH', 25, 'Male', 1, 'AC', 'A1', 1, 1),

(2, 'RASHMI', 30, 'Female', 2, 'NON\_AC', 'B1', 2, 2),

(3, 'DINESH', 35, 'Male', 3, 'AC', 'C1', 3, 3),

(4, 'RITA', 40, 'Female', 4, 'NON\_AC', 'D1', 4, 4),

(5, 'MAHENDRA', 45, 'Male', 5, 'AC', 'E1', 5, 5),

(6, 'ANITA', 50, 'Female', 6, 'AC', 'F1', 6, 6),

(7, 'DHARMENDRA', 55, 'Male', 7, 'NON\_AC', 'G1', 7, 7),

(8, 'ISHITA', 60, 'Female', 8, 'AC', 'H1', 8, 8),

(9, 'RAVINDRA', 65, 'Male', 9, 'NON\_AC', 'I1', 9, 9),

(10,'ALIYA', 70, 'Female', 10, 'AC', 'J1', 10, 10);

INSERT INTO MOVES\_ON (TRAIN\_ID, TRACK\_NO)

VALUES (1, 12),

(2, 10),

(3, 34),

(4, 20),

(5, 61),

(6, 53),

(7, 36),

(8, 73),

(9, 90),

(10, 67);

**System Specification:**

1. Software used:
2. MySQL - version 8.0.30 build 2054668 CE (64 bits) community
3. Notepad

2. System Specification:

a. OS version- 64 bit Operating System

b. Processor - AMD Ryzen 7 4800H

1. RAM- 16 GB

**Conclusion:**

The above project presents a relational database schema for a railway reservation system. It includes tables for trains, passengers, tracks, stations, schedules, routes, and tickets. The schema enables efficient management of train information, passenger details, track data, station associations, schedule planning, route mapping, and ticketing. It serves as a foundation for implementing a functional railway management system that facilitates booking, management, and tracking of train journeys.

E.O.F