DATABASE MANAGEMENT

TEAM MEMBERS:

E. M. MILIN DHITSHITHAA – 3122 21 5001 052

R. DARSHAN - 3122 21 5001 021

RAILWAY MANAGEMENT SYSTEM

PROBLEM OBJECTIVE:

To design the database schema for an Online booking and service system.

1. The instructor introduces the general aspects of Online booking and service system with constraints involved. Further, the faculty provides pointers to several variants of this system mentioned above.

2. The Students are grouped in teams and each team is asked to formulate a variant of Online booking and service system and identify the constraints and scope of their problem.

3. This problem can be solved in two phases.

PHASE-1:

IDENTIFICATION OF CONSTRAINTS AND FUNCTIONAL DEPENDENCIES (FD) AMONG SET OF ATTRIBUTES

PHASE-2:

NORMALIZING THE RELATIONS INTO 3NF OR BCNF

*In this Documentation

The Specifications and Assumptions have been made for the Railway database system.

The needed Entities and Relationships has been identified.

The basic ER Diagram has been established.

The FDs for the relations have been identified and the closure set has been found.

The candidate keys for each relation/entity have been identified.

Checking normal state of relation and normalizing until BCNF/3NF.The refined/final ER Diagram has been established.

Mapping of ER to Relational Schema to produce schema diagram is made.

This Railway Database Management System is a SQL plus database model enhanced with Java GUI application designed to automate the operations of managing a railway database system. It contains various aspects such as ticket booking, train scheduling, passenger information management, fare calculation and more. This system aims to provide an efficient and user-friendly platform for both railway administrators and passengers.

Key Components of the Railway Database Management System:

Ticket Booking: The system allows passengers to search for trains, check seat availability, and book tickets online. It provides a convenient way for passengers to select their preferred train, class, and seat.

Train Scheduling: The system manages the scheduling of trains, including their departure and arrival times, source and destination stations, and intermediate stops. It ensures proper coordination and optimization of train routes to minimize delays and maximize efficiency.

Passenger Information Management: The system maintains a database of passenger details, including their names, ages, genders, and contact information. It facilitates the management of passenger records, ticket cancellations, and modifications.

Fare Calculation: The system calculates the fare for different train routes based on class of travel. It ensures accurate fare calculation and provides transparent pricing information to passengers.

Train and Class Management: The system manages the information related to trains and their coaches. It keeps track of the train's capacity, coach types (e.g., sleeper, AC, general), and the number of available seats in each coach. This allows efficient allocation of seats during the booking process.

Automated Interface: The system includes a Java-based Graphical User Interface (GUI) that provides an easy-to-use platform for interacting with the database. The GUI presents a layout with organized menus, buttons, and forms, making it user-friendly for users.

ASSUMPTIONS MADE

Database System:

This developed system assumes the use of a relational database management system (RDBMS) to store and manage railway-related data.

It assumes that the database system provides data integrity, transaction management, and query processing capabilities.

The system assumes the availability of sufficient storage capacity to accommodate the expected amount of data.

Data Consistency and Concurrency:

This system assumes that the database maintains data consistency and integrity through the use of appropriate constraints and validation rules.

It assumes the implementation of concurrency control mechanisms to handle multiple users accessing and modifying data simultaneously.

Scalability and Performance:

The system assumes the need for scalability to accommodate a growing number of users, trains, and transactions over time.

It assumes the implementation of automated query processing to make the process easier.

Different Roles:

The system assumes the presence of different user roles, such as administrators, railway staff, and passengers.

The interface assumes the implementation of user authentication and access control mechanisms to ensure secure access to the system.

The system assumes that appropriate security measures are in place to protect user information and prevent unauthorized access.

ENTITIES WITH ATTRIBUTES

1. PASSSENGER
   - P_id
   - F_Name
   - L_Name
   - Gender
   - Phone
   - Age

2. TRAIN
   - Train_no
   - Name
   - Source
   - Dest
   - Seat_CC
   - Seat_SL
   - Seat_1A
   - Seat_2A
   - Seat_3A
   - Sunday
   - Monday
   - Tuesday
   - Wednesday
   - Thursday
   - Friday
   - Saturday

3. TICKET
   - T_no
   - Train_no
   - P_id
   - No_of_passengers
   - Date
   - Seat No
   - Class
   - Status
   - Price

4. AVAILABILITY
    - Train_no
    - Rem_CC
    - Rem_SL
    - Rem_1A
    - Rem_2A
    - Rem_3A

5. SCHEDULE
    - Train_no
    - Station_id
    - Station_Name
    - A_time
    - D_time

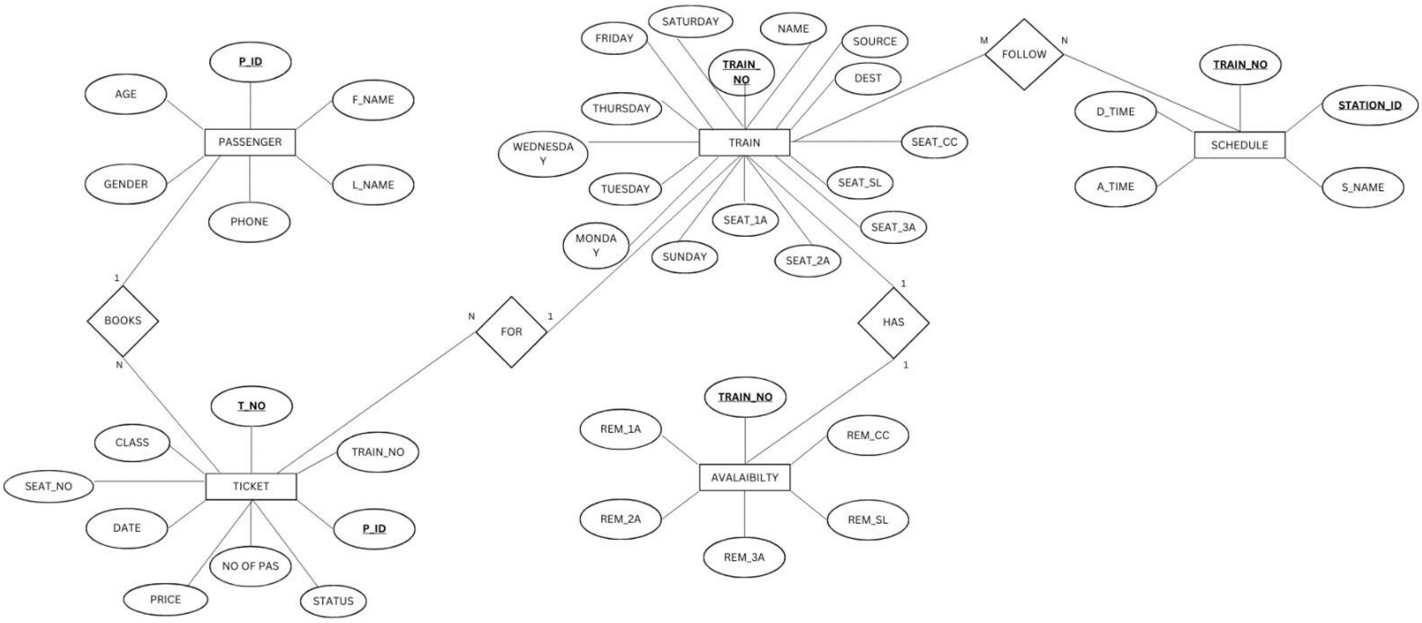
TABLE DEFINITIONS:

- The Trains table stores information about different trains, including their names, source station, destination station, departure time, and arrival time.
- The Stations table lists all the available stations.
- The Ticket Bookings table tracks the bookings made by users, including the associated user ID, train ID, booking date, seat number, and status of the booking.
- The Passengers table stores details of passengers associated with each booking, including their names, age, and gender.
- The Ticket Fare table stores the fare for different train routes between source and destination stations.
- The Schedule Table stores the details of arrival time, departure time and Station Id with its corresponding Train Number.
- The availability Table stores the details of remaining seats in each class for every available trains.

ENTITY-RELATIONSHIP DIAGRAM

Identified Relationships

1. Passenger – BOOKS – TICKET (1:N)
2. Ticket – FOR – Train (N:1)
3. Train – HAS – Availability (1:1)
4. Train – FOLLOW – Schedule (M:N)



FUNCTIONAL DEPENDENCIES

Functional dependencies are an important concept in database design. They can be used toimprove the efficiency of queries and to prevent data redundancy.

PASSENGER

The following are the functional dependencies for the attributes in the table Passenger:

Passenger_id --> First_Name, Last_Name, Gender, Phone, Age

TRAIN

The following are the functional dependencies for the TRAIN relation:

* Train_no → Name,Source, Dest, Seat_CC, Seat_SL, Seat_1A, Seat_2A, Seat_3A, Sunday ,Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

TICKET

The following are the functional dependencies for the TICKET relation:

* T_no → Train_no, P_id, No_of_passengers, Date, Seat No, Class, Status, Price

* Class → Price

AVAILABILITY

The following are the functional dependencies for the AVAILABILITY relation:

* Train_no → Rem_CC, Rem_SL, Rem_1A, Rem_2A, Rem_3A

SCHEDULE

The following are the functional dependencies for the SCHEDULE relation:

* Train_no,Station_id → Station_Name, A_time, D_time

* Station_id → Station_Name

PASSENGER

CLOSURE

$\{P\_id\}^{+}$ → {P_id, First_Name, Last_Name, Gender, Phone, Age}

$\{Phone\}^{+}$ → {P_id, First_Name, Last_Name, Gender, Phone, Age}

POSSIBLE SUPER KEY

There is no subset of P_id

SUPER KEY: {P_id}, {Phone}, {P_id, Phone}, … etc(One

of the attributes must be P_id or Phone)

CANDIDATE KEY: {P_id}, {Phone}

TRAIN

CLOSURE

$\{Train\_no\}^{+}$ → {Name,Source, Dest, Seat_CC, Seat_SL, Seat_1A, Seat_2A, Seat_3A,Sunday ,Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}

POSSIBLE SUPER KEY

There is no subset of Train_no

SUPER KEY: {Train_no}, {Train_no, Name}, … etc(One

of the attributes must be Train_no)

CANDIDATE KEY: {Train_no}

TICKET

CLOSURE

$\{T\_no\}^{+}$ → {T_no,Train_no, P_id, No_of_passengers, Date, Seat No, Class, Status, Price}

POSSIBLE SUPER KEY

There is no subset of T_no

SUPER KEY: {T_no}, {T_no, Train_no}, {T_no , P_id} , ... etc(One

of the attributes must be T_no)

CANDIDATE KEY: {T_no}

AVAILABILTY

CLOSURE

$\{Train\_no\}^+ \rightarrow \{Rem\_CC, Rem\_SL, Rem\_1A, Rem\_2A, Rem\_3A\}$

POSSIBLE SUPER KEY

There is no subset of Train_no

SUPER KEY: {Train_no}

(One of the attributes must be Train_no)

CANDIDATE KEY: {Train_no}

SCHEDULE

CLOSURE

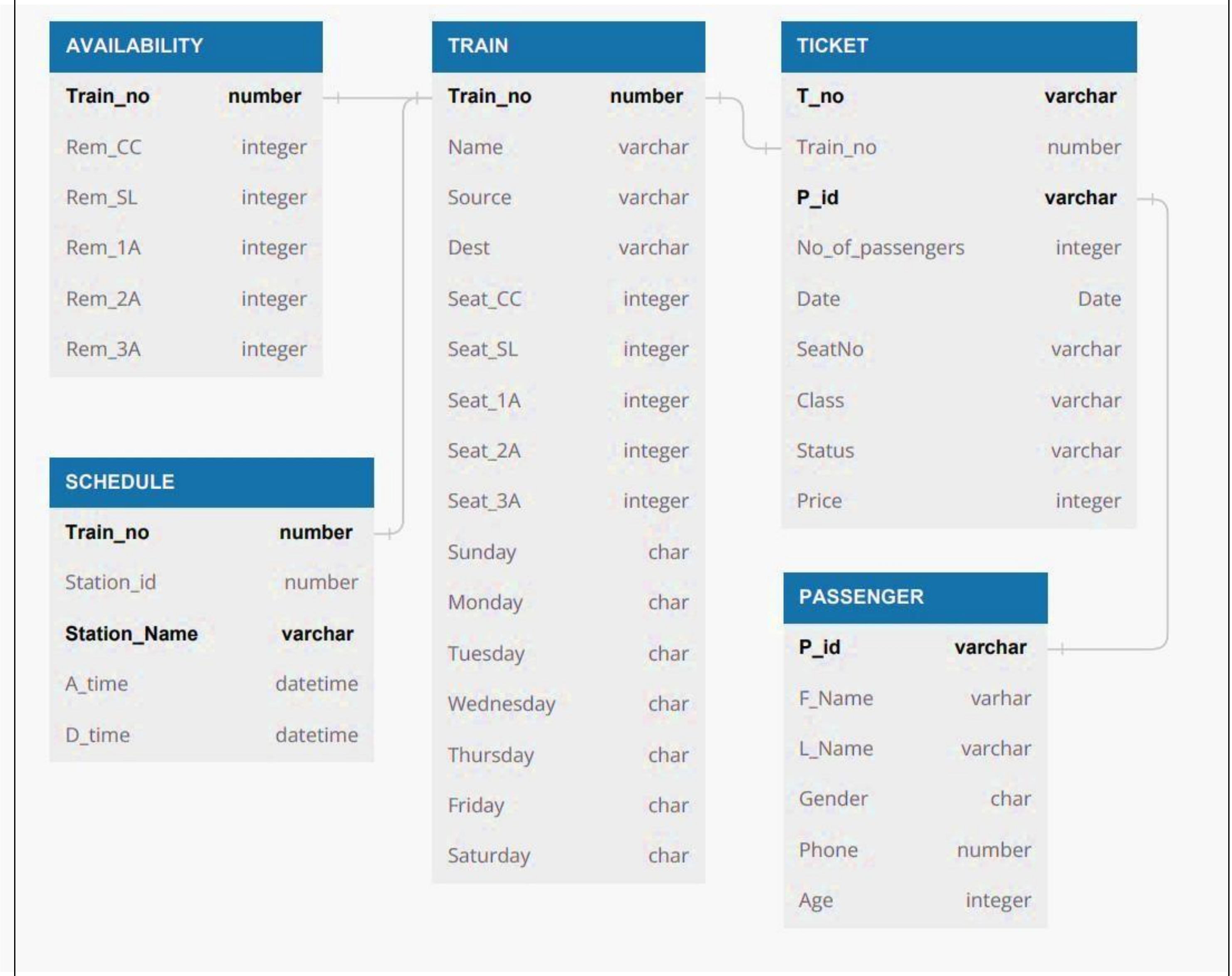
$\{Train_no,Station_id\}^+ \rightarrow \{Station_Name, A_time, D_time\}$POSSIBLE

SUPER KEY

SUPER KEY: {Train_no, Station_id}, {Train_no, Station_id, Station_name } … etc(One of the attributes must be Train_no and Station_id)

CANDIDATE KEY: {Train_no, Station_id}

INITIAL SCHEMA DIAGRAM



AVAILABILITY	
Train_no	**number**
Rem_CC	integer
Rem_SL	integer
Rem_1A	integer
Rem_2A	integer
Rem_3A	integer

TRAIN	
Train_no	**number**
Name	varchar
Source	varchar
Dest	varchar
Seat_CC	integer
Seat_SL	integer
Seat_1A	integer
Seat_2A	integer
Seat_3A	integer
Sunday	char
Monday	char
Tuesday	char
Wednesday	char
Thursday	char
Friday	char
Saturday	char

TICKET	
T_no	**varchar**
Train_no	number
P_id	**varchar**
No_of_passengers	integer
Date	Date
SeatNo	varchar
Class	varchar
Status	varchar
Price	integer

SCHEDULE	
Train_no	**number**
Station_id	number
Station_Name	**varchar**
A_time	datetime
D_time	datetime

PASSENGER	
P_id	**varchar**
F_Name	varhar
L_Name	varchar
Gender	char
Phone	number
Age	integer

NORMALIZATION

PASSENGER

1NF

Domain of an attribute must include only atomic (simple, indivisible) values and that thevalue of any attribute in a tuple must be a single value.

Disallows composite attributes, multivalued attributes, and nested relations; attributes whosevalues for an individual tuple are non-atomic.

2NF

The candidate keys are {P_id}, {Phone}, The

set of key attributes are: {P_id,Phone}

for each non-trivial FD, checking whether the LHS is a proper subset of some candidate keyor the RHS are not all key attributes.

Checking FD: P_id --> FirstName,Last_Name,Gender,Phone,Age

Checking FD: Phone --> P_id

3NF

The candidate keys are {P_id}, {Phone},

The set of key attributes are: {P_id,Phone }

for each FD, checking whether the LHS is super key or the RHS are all key attributes.

checking FD: P_id --> FirstName,Last_Name,Gender,Phone,Age

checking FD: Phone --> P_id

BCNF

A table is in BCNF if and only if for every non-trivial FD, the LHS is a super key.

✓ Doesn't contain any multi-valued attributes – Satisfies 1NF

✓ Doesn't contain non-prime key attribute - Satisfies 2NF

✓ No transitive Dependency – Satisfies 3NF

✓ LHS is a Super key – Satisfies BCNF

Therefore, the table is already in BCNF form.

TRAIN

✓ Doesn’t contain any multi-valued attributes – Satisfies 1NF

✓ Doesn’t contain non-prime key attribute - Satisfies 2NF

✓ No transitive Dependency – Satisfies 3NF

✓ LHS is a Super key – Satisfies BCNF

Therefore, the table is already in BCNF form.

TICKET

1NF

Domain of an attribute must include only atomic (simple, indivisible) values and that thevalue of any attribute in a tuple must be a single value.

Disallows composite attributes, multivalued attributes, and nested relations; attributes whosevalues for an individual tuple are non-atomic.

2NF

The candidate keys are {T_no}, The set of key attributes are: {T_no }

for each non-trivial FD, check whether the LHS is a proper subset of some candidate key orthe RHS are not all key attributes.

Checking FD: T_no --> Train_no,P_id,No_of_passengers,Date,Seat_No,Class,Status,PriceChecking

FD: Class --> Price

3NF

The candidate keys are {T_no},

The set of key attributes are: {T_no}

for each FD, check whether the LHS is superkey or the RHS are all key attributes. checking FD:

T_no --> Train_no,P_id,No_of_passengers,Date,Seat_No,Class,Status,Pricechecking FD: Class --

> Price

The above FD violates definition of 3NF:

It is non-trivial, LHS is not super key, RHS contains a non-key attribute.

BCNF

A table is in BCNF if and only if for every non-trivial FD, the LHS is a super key.

The FD: Class --> Price is non-trivial and its LHS is not a super key. It violates BCNF.

X Doesn’t contain any multi-valued attributes – Does not Satisfy 1NF

✓ Doesn’t contain non-prime key attribute - Satisfies 2NF but Does not Satisfy 1NF

X No transitive Dependency – Does not Satisfy 3NF

X LHS is a Super key - Does not Satisfy BCNF

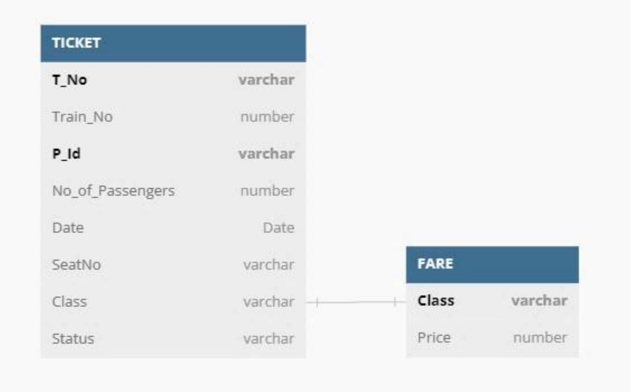
NORMALIZED TABLE

1NF:

T_No	Train_No	P_Id	No_of_Pass	Date	Seat_No	Class	Status	Price
T1921	16209	P98	3	2023-06-20	SU21, SU22, SU23	SL	CNF	250
T1922	12235	P04	1	2023-07-11	CL02	CC	CNF	350

T_No	Train_No	P_Id	No_of_Pass	Date	Seat_No	Class	Status	Price
T1921	16209	P98	3	2023-06-20	SU21	SL	CNF	250
T1921	16209	P98	3	2023-06-20	SU22	SL	CNF	250
T1921	16209	P98	3	2023-06-20	SU23	SL	CNF	250
T1922	12235	P04	1	2023-07-11	CL02	CC	CNF	350

3NF:



TICKET	
T_No	**varchar**
Train_No	number
P_Id	**varchar**
No_of_Passengers	number
Date	Date
SeatNo	varchar
Class	varchar
Status	varchar

FARE	
Class	**varchar**
Price	number

Now checking the conditions:

✓ Doesn’t contain any multi-valued attributes – Satisfies 1NF

✓ Doesn’t contain non-prime key attribute - Satisfies 2NF

✓ No transitive Dependency – Satisfies 3NF

✓ LHS is a Super key – Satisfies BCNF

Therefore, the table is in BCNF form.

AVAILABILITY

✓ Doesn’t contain any multi-valued attributes – Satisfies 1NF

✓ Doesn’t contain non-prime key attribute - Satisfies 2NF

✓ No transitive Dependency – Satisfies 3NF

✓ LHS is a Super key – Satisfies BCNF

Therefore, the table is already in BCNF form.

SCHEDULE

1NF

Domain of an attribute must include only atomic (simple, indivisible) values and that thevalue of any attribute in a tuple must be a single value.

Disallows composite attributes, multivalued attributes, and nested relations; attributes whosevalues for an individual tuple are non-atomic.

2NF

The candidate keys are {Train_no,Station_id}, The set of key attributes are: { Train_no,Station_id }

for each non-trivial FD, check whether the LHS is a proper subset of some candidate key orthe RHS are not all key attributes.

Checking FD: Train_no,Station_id --> Station_Name,A_time,D_time

Checking FD: Station_id --> Station_Name

The FD: violates definition of 2NF -- LHS is a proper subset of some CK.

3NF

The candidate keys are {Train_no,Station_id}, The set of key attributes are: { Train_no,Station_id }

for each FD, check whether the LHS is superkey or the RHS are all key attributes.

Checking FD: Train_no,Station_id --> Station_Name,A_time,D_time

Checking FD: Station_id --> Station_Name

The above FD violates definition of 3NF: it is non-trivial, LHS is not super key, RHScontains a non-key attribute.

BCNF

A table is in BCNF if and only if for every non-trivial FD, the LHS is a superkey. The FD Station_id --> Station_Name is non-trivial and its LHS is not a superkey. It violates BCNF.

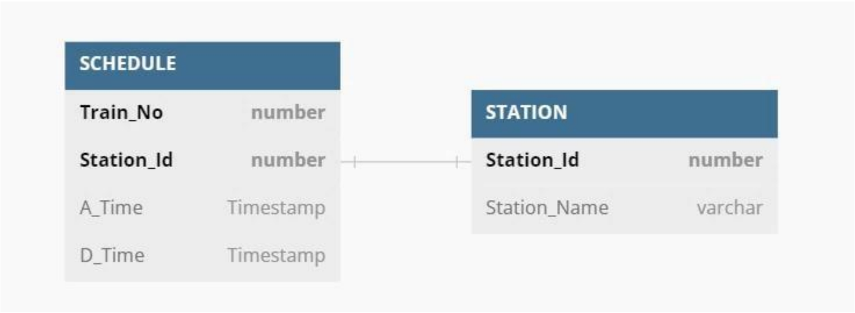
✓ Doesn’t contain any multi-valued attributes – Satisfies 1NF

X Doesn’t contain non-prime key attribute - Does not Satisfy 2NFX No transitive Dependency – Does not Satisfy 3NF

X LHS is a Super key - Does not Satisfy BCNF

NORMALIZED TABLE

2NF:



SCHEDULE	
Train_No	number
Station_Id	number
A_Time	Timestamp
D_Time	Timestamp

STATION	
Station_Id	number
Station_Name	varchar

Now checking the conditions:

✓ Doesn’t contain any multi-valued attributes – Satisfies 1NF

✓ Doesn’t contain non-prime key attribute - Satisfies 2NF

✓ No transitive Dependency – Satisfies 3NF

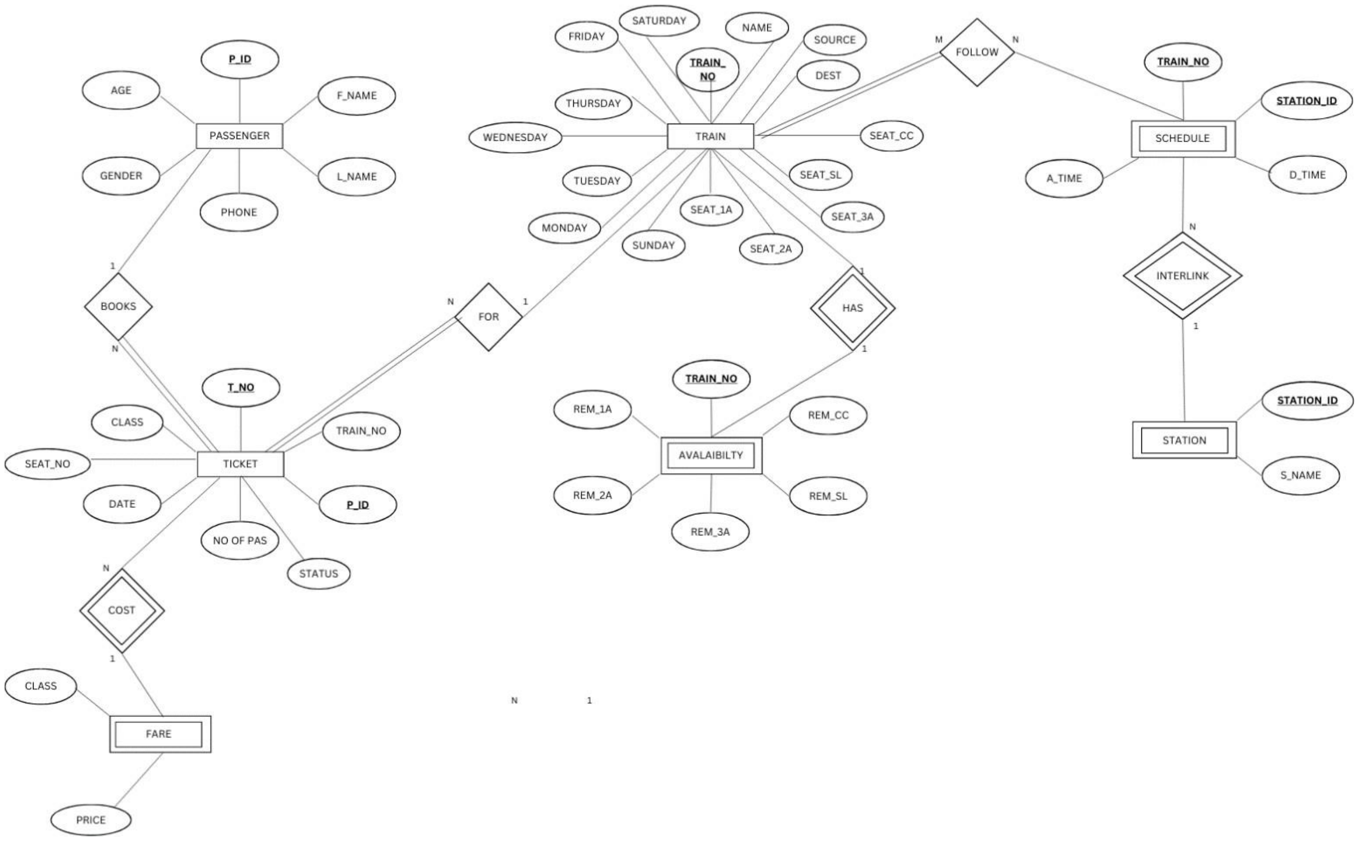
✓ LHS is a Super key – Satisfies BCNF

Therefore, the table is in BCNF form.

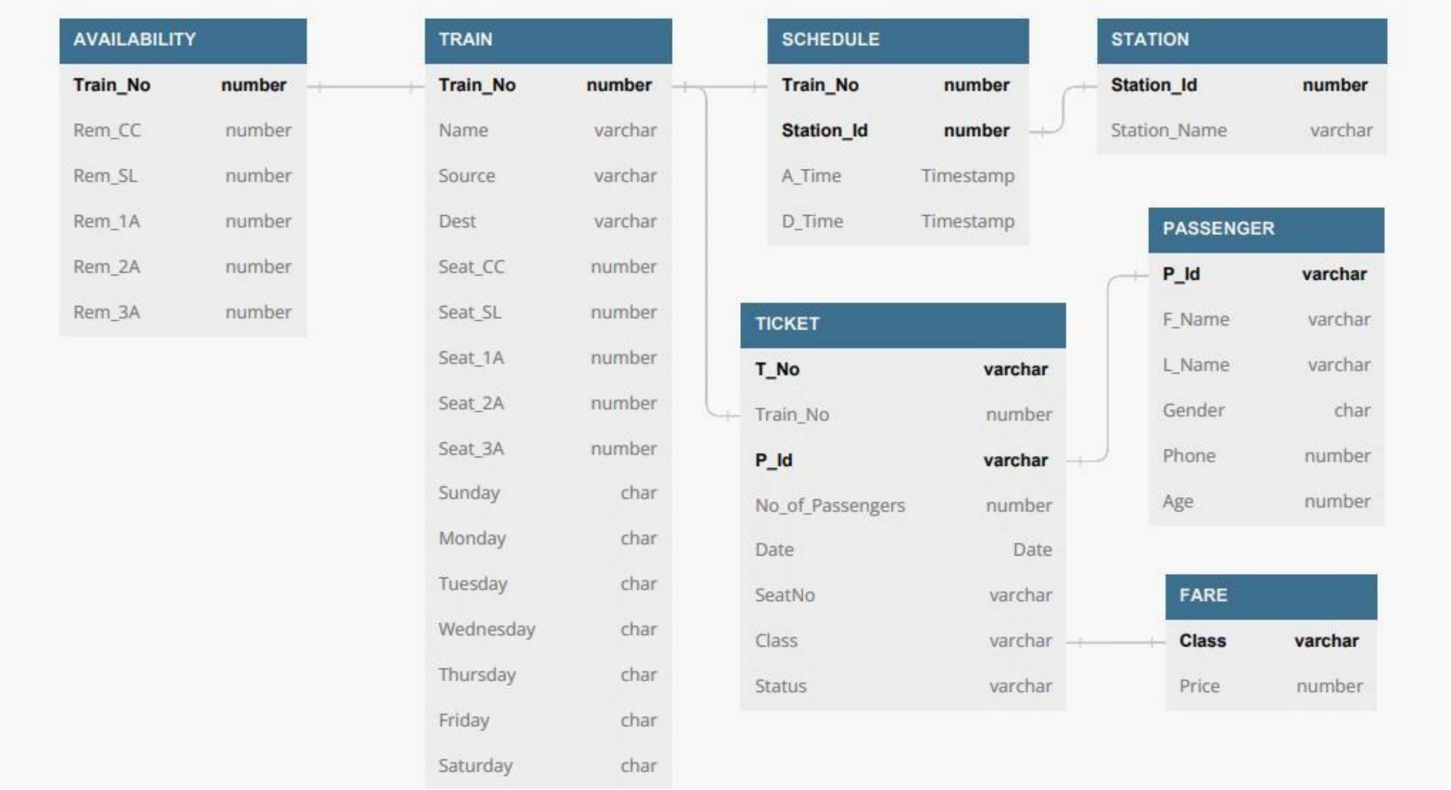
REFINED ENTITY RELATIONSHIP DIAGRAM

Newly Identified Relationships

1. Fare – COST – Ticket (1:N)
2. Station – INTERLINK – Schedule (1:N)



FINAL SCHEMA DIAGRAM



AVAILABILITY	
Train_No	**number**
Rem_CC	number
Rem_SL	number
Rem_1A	number
Rem_2A	number
Rem_3A	number

TRAIN	
Train_No	**number**
Name	varchar
Source	varchar
Dest	varchar
Seat_CC	number
Seat_SL	number
Seat_1A	number
Seat_2A	number
Seat_3A	number
Sunday	char
Monday	char
Tuesday	char
Wednesday	char
Thursday	char
Friday	char
Saturday	char

SCHEDULE	
Train_No	**number**
Station_Id	**number**
A_Time	Timestamp
D_Time	Timestamp

STATION	
Station_Id	**number**
Station_Name	varchar

TICKET	
T_No	**varchar**
Train_No	number
P_Id	**varchar**
No_of_Passengers	number
Date	Date
SeatNo	varchar
Class	varchar
Status	varchar

PASSENGER	
P_Id	**varchar**
F_Name	varchar
L_Name	varchar
Gender	char
Phone	number
Age	number

FARE	
Class	**varchar**
Price	number