Bahria University

Department Of Computer Science

Course Code: CSC 220

Database Management Systems

4th Semester

Academic Year Spring 2023

Project On

Train Ticketing System

Submitted by

Saad Ahmed

01-134211-077

BSCS(4-D)

Table of Contents

Title of the Project

"Train Ticketing System"

Abstract

The **Train Ticketing System** OR **Railway Reservation System** facilitates the passengers to enquire about the trains available based on source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of project is to design and develop a database maintaining the records of different trains, train status, and passengers.

This project contains Introduction to the Train Ticketing System. It is the computerized system of reserving the seats of train seats in advanced. It is mainly used for long route. On-line reservation has made the process for the reservation of seats very much easier than ever before.

In our country, there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Then this project will also contain Entity Relationship model diagram based on Train Ticketing System and introduction to relation model. There is also design of the database of the Train Ticketing System based on relation model.

Introduction

Database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. A **DBMS** makes it possible for end users to create, read, update, and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked, and modified and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

The main purpose of maintaining database for **Train Ticketing System** is to reduce the manual errors involved in the booking and cancelling of tickets and make it convenient for the customers and providers to maintain the data about their customers and about the seats available at them. Due to automation many loopholes that exist in the manual maintenance of the records can be removed. The speed of obtaining and processing the data will be fast. For future expansion the proposed system can be web enabled so that clients can make various enquiries about trains between stations. Due to this, sometimes a lot of problems occur, and they are facing many disputes with customers. To solve the above problem, we will design a data base which includes customer details, availability of seats in trains, no of trains and their details.

Project Description

This project is about creating the database about **Train Ticketing System**. The Train Ticketing System facilitates the passengers to enquire about the trains available based on source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of this project is to design and develop a database maintaining the records of different trains, train status, and passengers. The record of train includes its number, name, source, destination, and days on which it is available, whereas record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked and much more. This was an overview of our project. Detailed description will be in there in the final submission of the project along with the report when the project will be completed.

Users

The users of our Database will be the people travelling on long distance routes such as travelling from one city to another and so on. Our user basically is the passenger or the customer. In short, the user of our database will be **Parametric**

End User who don't have any DBMS knowledge but they still use the database and perform their desired task.

Data Requirements

Users:

- First_Name
- Last_Name
- PhoneNo
- CNIC

Passenger:

- Passenger_ID
- Driver_ID
- Booked_By
- Reservation_Status

- PNR_No
- Seat_No

Train:

- **Train_ID**
- Train_Name
- Source
- Destination
- Availability_Of_Seats
- Driver_ID

Driver:

- **Driver_ID**
- Age
- Email
- Address
- Experience

Station:

- **Station_ID**
- Station_Name
- Available_Trains

TicketForSeats:

- **Ticket_ID**
- Seat Name
- Economy_Seat1
- ACSL_Fare
- Economy_Fare
- ACSL_Seat1
- ACSL_Seat2
- ACSL_Seat3
- Economy_Seat3
- Economy_Seat2
- Category

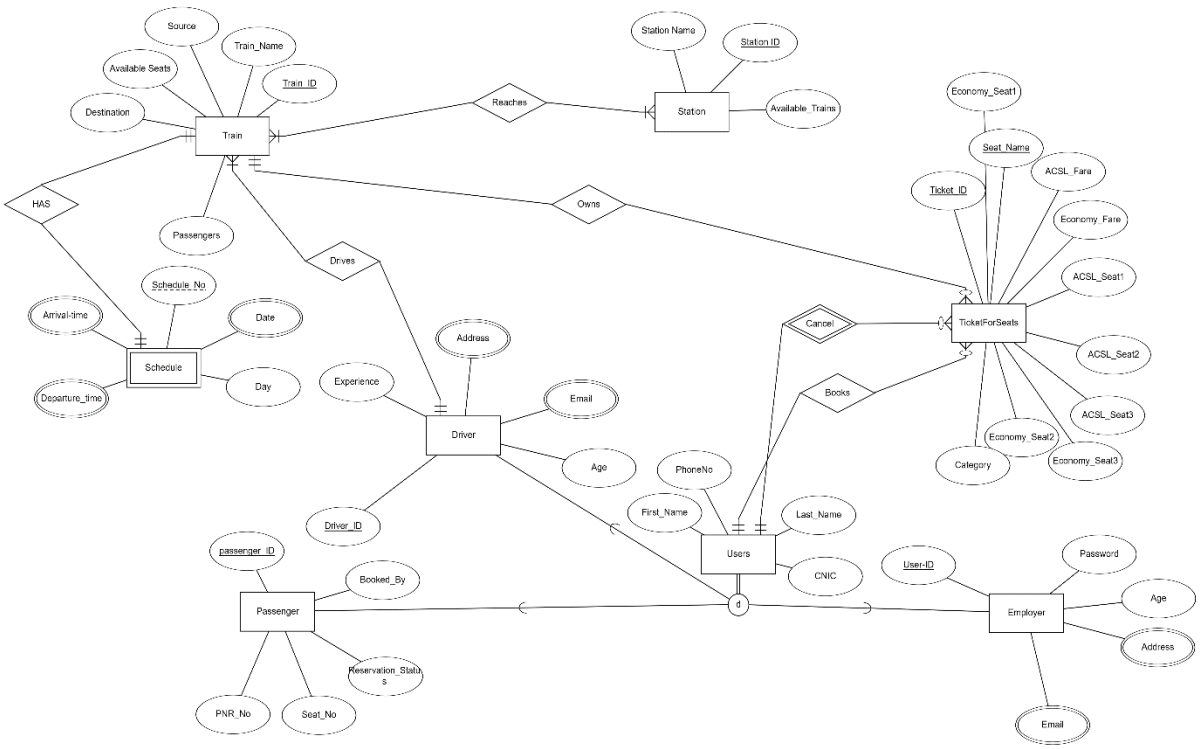
Employer:

- **User_ID**
- Password
- Age
- Address
- Email

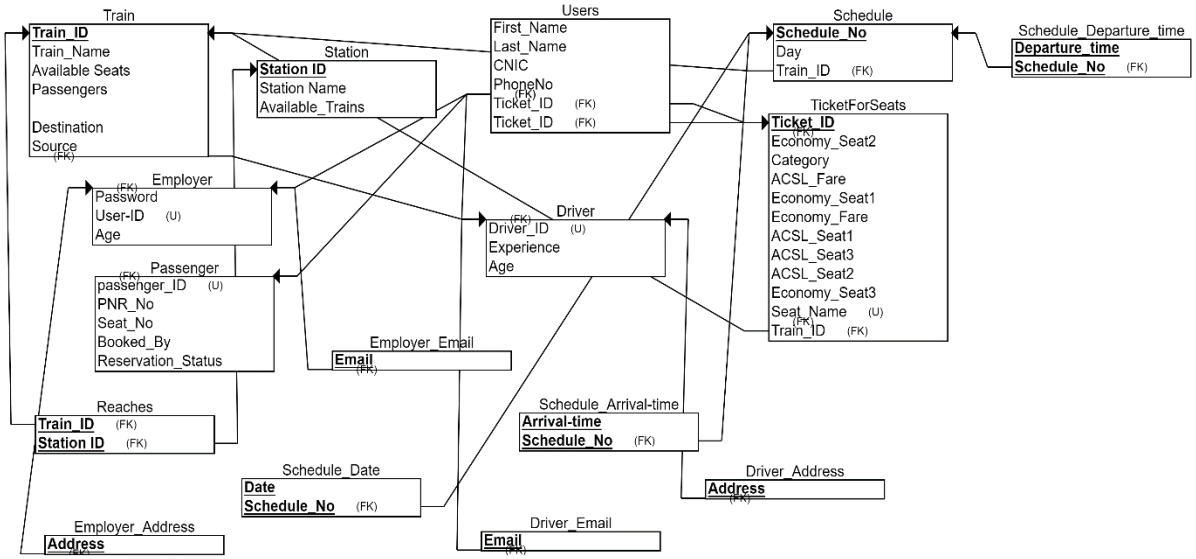
Schedule:

- **Schedule_No**
- Arrival_Time
- Departure_Time
- Date
- Day

Entity Relationship Diagram (ERD)



Relational Schema



Train (Train_ID (PK), Train_Name, Available_seats, Passengers, Driver_ID (FK), Destination, Source)

TicketForSeats (Ticket_ID (PK), Economy_seat2, Category, ACSL_Fare, Economy_seat1, Economy_Fare, ACSL_Seat1, ACSL_Seat3, ACSL_Seat2, Economy_Seat3, Seat_Name (U), Train_ID (FK))

Employer (Employer_ID (PK),Password (U), Age)

Passenger (Passenger_ID (PK), PNR_No, Seat_No, Booked_By, Reservation_Status)

Driver (Driver_ID (PK), Experience, Age)

Station (Station_ID (PK), Station_Name, Available_Trains)

Users (Ticket_ID (FK), First_Name, Last_Name, CNIC, Phone_No)

Schedule (Schedule_No, Day, Train_ID(FK))

Reaches (Train_ID (FK), Station_ID (FK))

Employer_Address (Address)

Employer_Email (Email (FK))

Schedule_Arrival_Time (Arrival_Time, Schedule_No (FK))
Schedule_Departure_Time (Departure_Time, Schedule_No (FK))

Schedule_Date (Date, Schedule_No (FK))

Driver_Email (Email (FK))

Driver_Address (Address) (FK)

Final Set of Relations

1. Train has a binary relation of "**Reaches**" with station. This is a many to many relation such that "A train reaches many stations and A station is reached by many trains" and both are mandatory in this case.

2. Train has a binary relation of "**HAS**" with schedule too. This is a one to one relation such that "A train has only one schedule and one schedule is assigned to one train only" and both are mandatory.

3. Driver has a binary relation of "**Drives**" with Train. This is a one to many relation such that "A driver can drive many trains but one train is driven by one driver only" and both are mandatory.

4. User has a binary relation of "**Books**" with TicketForSeats. This is a one to many relation such that "A user can book many tickets but each ticket is booked by one user only" and here user is mandatory but TicketForSeats is optional.

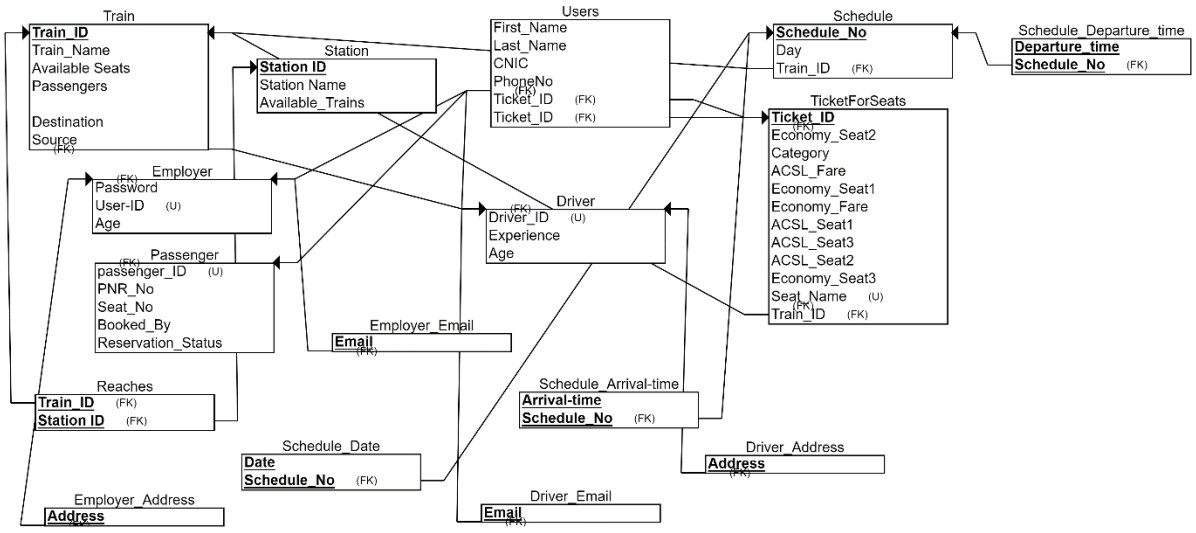
5. User has a binary relation of "**Cancel**" with TicketForSeats too. This is a one to many weak relation such that "A user can cancel many tickets but each ticket is cancelled by one user only" and here User is mandatory but TicketForSeats is optional.

6. Train has a binary relation of "**Owns**" with TicketForSeats. This is a one to many relation such that "A train owns many tickets but each ticket is owned by/belomgs to one train only" and here Train is mandatory but TicketForSeats is optional.

Step by step conversion of relational model till 3NF

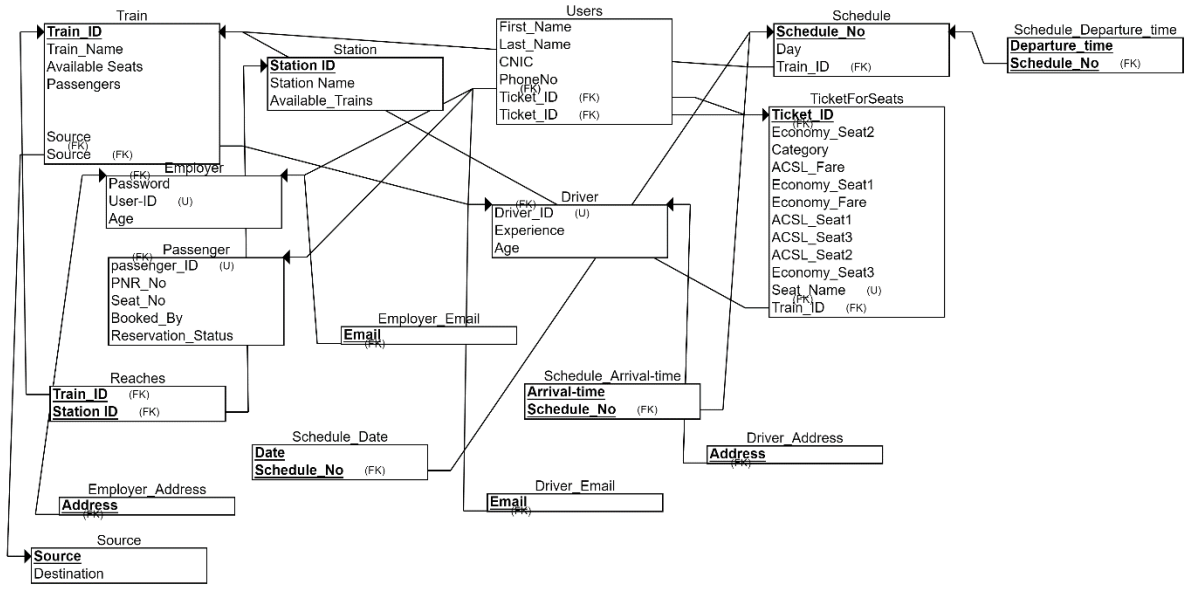
Normalization (2NF)

Our Data base was already in 2NF as:



Normalization (3NF)

3rd normal form



SQL Script

Table Creation

Train Table:

CREATE TABLE Train(
Train_ID num(6) NOT NULL PRIMARY KEY,
Train_Name varchar(255),
Source varchar(255),
AvailableSeats num(4),
Destination varchar(255),
Driver_ID num(6) FOREIGN KEY REFERENCES Driver(Driver_ID)
);

TicketForSeats Table:

CREATE TABLE TicketForSeats(
Ticket_ID num(6) NOR NULL PRIMARY KEY,
Category varchar(255),
Seat_Name varchar(255) NOT NULL UNIQUE,
Train_ID num(6) FOREIGN KEY REFRENCES Train(Train_ID)
);

Employer Table:

```
CREATE TABLE Employer(
Employer_ID NOT NULL PRIMARY KEY,
Password varchar(255) NOT NULL UNIQUE,
Age num(6)
);
```

Passenger Table:

```
CREATE TABLE Passenger(
Passenger_ID num(6) NOT NULL PRIMARY KEY,
PNR_No num(6),
Seat_No num(6),
Booked_By varchar(255),
Reservation Status varchar(20)
);
```

Driver Table:

```
CREATE TABLE Driver (
Driver_ID num(6) NOT NULL PRIMARY KEY,
Experience num(6),
Age num(6)
);
```

Station Table:

```
CREATE TABLE Station (
Station_ID num(6) NOT NULL PRIMARY KEY,
Station_Name varchar(100),
Available_Trains varchar(200)
);
```

Users Table:

```
CREATE TABLE Users (
Ticket_ID num(6) FOREIGN KEY REFRENCES TicketForSeats(Ticket_ID),
First_Name varchar(50),
Last_Name varchar(50),
CNIC varchar(15),
Phone_No varchar(15)
);
```

Schedule Table:

```
CREATE TABLE Schedule (
Schedule_No num(6) NOT NULL PRIMARY KEY,
Day varchar(10),
Train_ID num(6) REFERENCES Train(Train_ID)
```

);

Insertion Queries

Train Table:

```
INSERT INTO Train (Train_ID, Train_Name, Source, AvailableSeats, Destination,
Driver_ID)
VALUES (1, 'Express', 'New York', 100, 'Los Angeles', 23);

INSERT INTO Train (Train_ID, Train_Name, Source, AvailableSeats, Destination,
Driver_ID)
VALUES (2, 'SuperFast', 'London', 50, 'Paris', 24);

INSERT INTO Train (Train_ID, Train_Name, Source, AvailableSeats, Destination,
Driver_ID)
VALUES (3, 'Shatabdi', 'Delhi', 80, 'Mumbai',25);

INSERT INTO Train (Train_ID, Train_Name, Source, AvailableSeats, Destination,
Driver_ID)
VALUES (4, 'Rajdhani', 'Kolkata', 60, 'Chennai', 26);
```

TicketForSeats Table:

```
INSERT INTO TicketForSeats (Ticket_ID, Category, Seat_Name, Train_ID)
VALUES (1, 'Business', 'A1', 5);

INSERT INTO TicketForSeats (Ticket_ID, Category, Seat_Name, Train_ID)
VALUES (2, 'Economy', 'B3', 6);

INSERT INTO TicketForSeats (Ticket_ID, Category, Seat_Name, Train_ID)
VALUES (3, 'Business', 'C5', 7);

INSERT INTO TicketForSeats (Ticket_ID, Category, Seat_Name, Train_ID)
VALUES (4, 'Economy', 'D2', 8);
```

Employer Table:

```
INSERT INTO Employer (Employer_ID, Password, Age)
VALUES ('E1', 'mypassword', 30);

INSERT INTO Employer (Employer_ID, Password, Age)
VALUES ('E2', 'mypassword2', 25);

INSERT INTO Employer (Employer_ID, Password, Age)
```

```
VALUES ('E3', 'mypassword3', 35);

INSERT INTO Employer (Employer_ID, Password, Age)
VALUES ('E4', 'mypassword4', 40);
```

Passenger Table:

```
INSERT INTO Passenger (Passenger_ID, PNR_No, Seat_No, Booked_By,
Reservation_Status)
VALUES (1, 123456, 1, 'John Smith', 'Confirmed');

INSERT INTO Passenger (Passenger_ID, PNR_No, Seat_No, Booked_By,
Reservation_Status)
VALUES (2, 234567, 2, 'Amy Lee', 'Confirmed');

INSERT INTO Passenger (Passenger_ID, PNR_No, Seat_No, Booked_By,
Reservation_Status)
VALUES (3, 345678, 3, 'James Lee', 'Pending');

INSERT INTO Passenger (Passenger_ID, PNR_No, Seat_No, Booked_By,
Reservation_Status)
VALUES (4, 456789, 4, 'Emily Brown', 'Cancelled');
```

Driver Table:

```
INSERT INTO Driver (Driver_ID, Experience, Age)
VALUES (1, 5, 35);

INSERT INTO Driver (Driver_ID, Experience, Age)
VALUES (2, 8, 40);

INSERT INTO Driver (Driver_ID, Experience, Age)
VALUES (3, 12, 45);

INSERT INTO Driver (Driver_ID, Experience, Age)
VALUES (4, 15, 50);
```

Station Table:

```
INSERT INTO Station (Station_ID, Station_Name, Available_Trains)
VALUES (1, 'New York', 'Express, SuperFast, Shatabdi');

INSERT INTO Station (Station_ID, Station_Name, Available_Trains)
VALUES (2, 'London', 'SuperFast, Rajdhani, Shatabdi');

INSERT INTO Station (Station_ID, Station_Name, Available_Trains)
VALUES (3, 'Delhi', 'Rajdhani, Shatabdi, Express');
```

```
INSERT INTO Station (Station_ID, Station_Name, Available_Trains)
VALUES (4, 'Kolkata', 'Shatabdi, SuperFast, Rajdhani');
```

Users Table:

```
INSERT INTO Users (Ticket_ID, First_Name, Last_Name, CNIC, Phone_No)
VALUES (1, 'John', 'Doe', '1234567890123', '+1234567890');

INSERT INTO Users (Ticket_ID, First_Name, Last_Name, CNIC, Phone_No)
VALUES (2, 'Jane', 'Doe', '2345678901234', '+2345678901');


INSERT INTO Users (Ticket_ID, First_Name, Last_Name, CNIC, Phone_No)
VALUES (3, 'Bob', 'Smith', '3456789012345', '+3456789012');


INSERT INTO Users (Ticket_ID, First_Name, Last_Name, CNIC, Phone_No)
VALUES (4, 'Alice', 'Johnson', '4567890123456', '+4567890123');
```

Schedule Table:

```
INSERT INTO Schedule (Schedule_No, Day, Train_ID)
VALUES (1, 'Monday', 5);

INSERT INTO Schedule (Schedule_No, Day, Train_ID)
VALUES (2, 'Tuesday', 6);

INSERT INTO Schedule (Schedule_No, Day, Train_ID)
VALUES(3, 'Wednesday', 7);

INSERT INTO Schedule (Schedule_No, Day, Train_ID)
VALUES (4, 'Thursday', 8);
```

Conclusion

In our project **Train Ticketing System,** we will have to store all the information about the Trains scheduled and the users' booking tickets and even status of trains, seats etc. This data base will be helpful for the applications which facilitate passengers to book the train tickets and check the details of trains and their status from their place itself and avoid inconveniences of going to railway station for each and every query they get. We had considered the most important requirements only, many more features and details can be added to the project in order to obtain even more user-friendly applications. These applications are already in progress and in future they can be upgraded and may become part of amazing technology.