Database Management System

CSC 220

Project Submission 02

Train Ticketing System

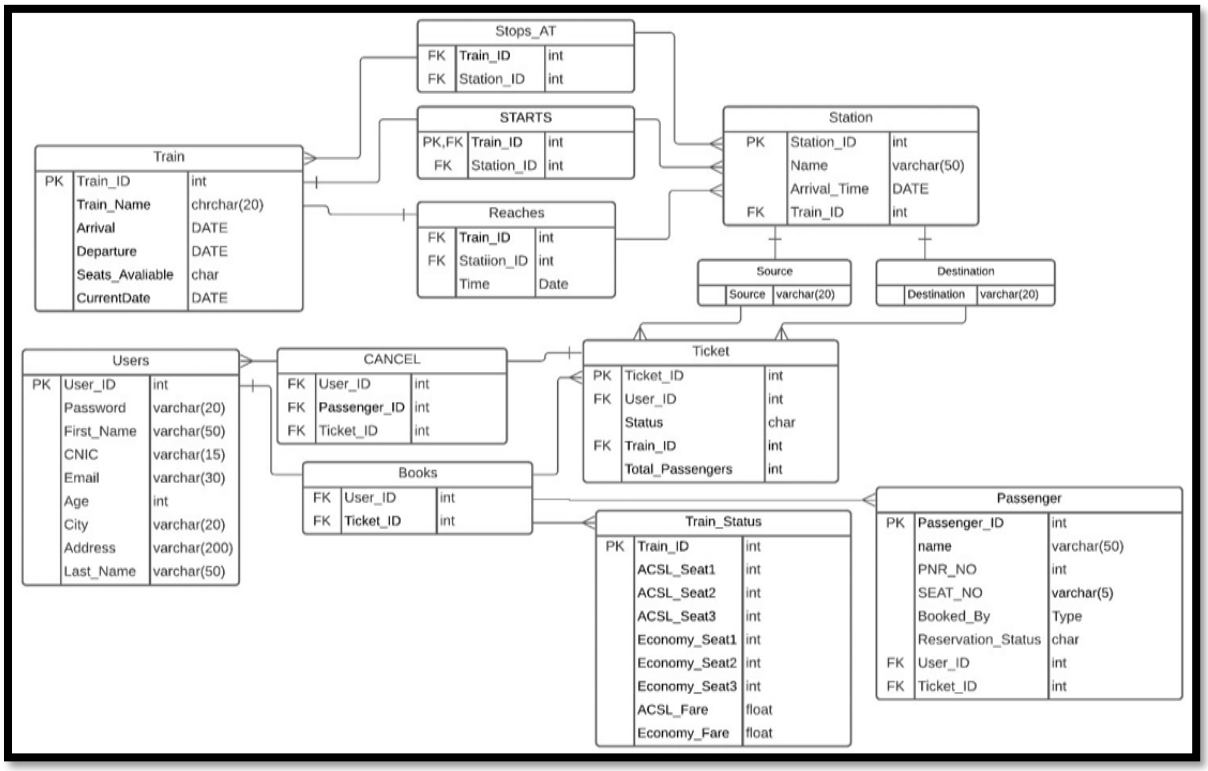
Group Members:

Muhammad Abdullah (049)
Aroob Zuberi (029)
Saiqa Rehmat (081)
BS(CS)-4B

Table of Contents

Project Submission 02

1. Entity Relation Diagram (ERD):



---

2. Relational Schema:

Train (Train_ID, Train_Name, Arrival, Departure, Seats, Available, CurrentDate)

Users (User_ID, Password, First_Name, CNIC, Email, Age, City, Address, Last_Naame)

Stops_AT (Train_ID, Station_ID)

STARTS (Train_ID, Station_ID)

Reaches (Train_ID, Station_ID, Time)

Station (Station_ID, Name, Arrival Time, Train_ID)

Source (Source)

Destination (Destination)

Books (User_ID, Ticket_ID)

CANCEL (User_ID, Passenger_ID, Ticket_ID)

Ticket (Ticket_ID, User_ID, Status, Train_ID, Total_Passengers)

Train_Status (Train_ID, ACSL_Seat1, ACSL_Seat2, ACSL_Seat3, Economy_Seat1, Economy_Seat2, Economy_Seat3, ACSL_Fare, Economy_Fare)

Passenger (Passenger_ID, name, PNR_NO, SEAT_NO, Booked_By, Reservation_Status, User_ID, Ticket_ID)

Project Submission 01

3. Title of the Project

"Train Ticketing System"

4. 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.

5. 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.

6. 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.

7. 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.

8. Data Requirements

User:

- **User_ID**

- Password
- First_Name
- Last_Name
- Gender
- Age
- Email
- CNIC
- Contact_No
- City
- State
- PinCode
- Security_Ques
- Security_Ans

Passenger:

- **Passenger_ID**
- Name
- Gender
- Age
- Seat_No
- Booked_By
- Reservation_Status
- User_ID
- Ticket_ID

Train:

- **Train_ID**
- Train_Name
- Source
- Destination
- Arrival_Time
- Departure_Time
- Availability_Of_Seats
- Current_Date

Station:

- **Station_ID**
- Station_Name
- Train_ID
- Arrival_Time

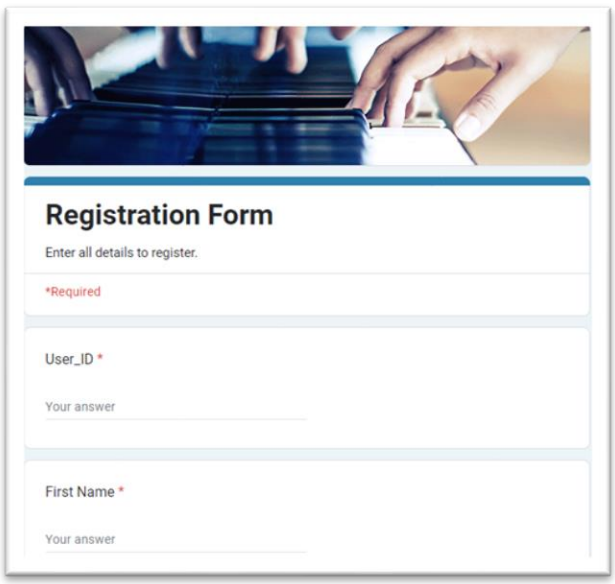
Ticket:

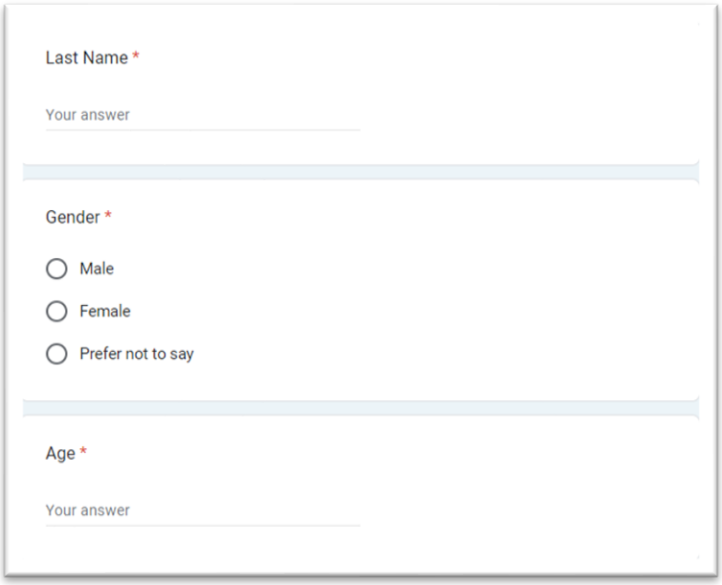
- **Ticket_ID**
- User_ID
- Status
- Train_ID
- Total_Passengers

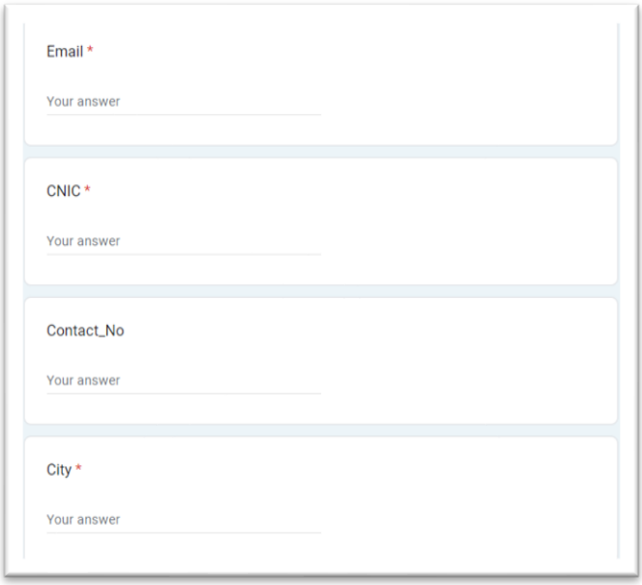
9. Application Design

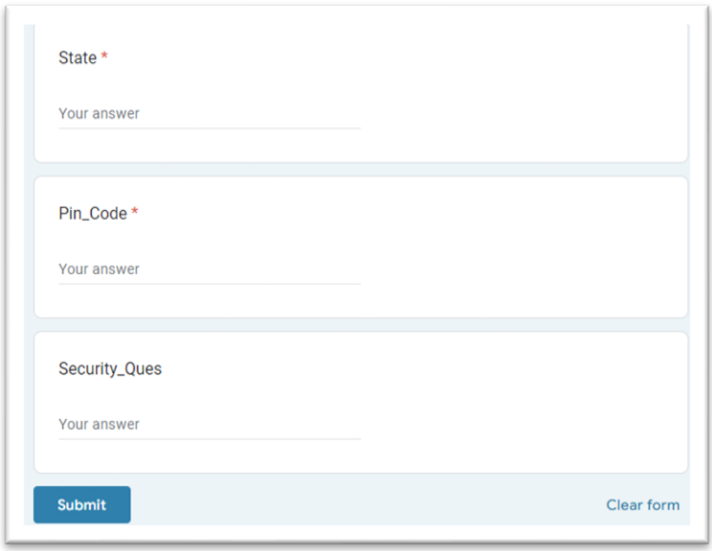
We have used Google forms to sketch the design of interfaces of our forms of the application for an overview.

Registration Form:

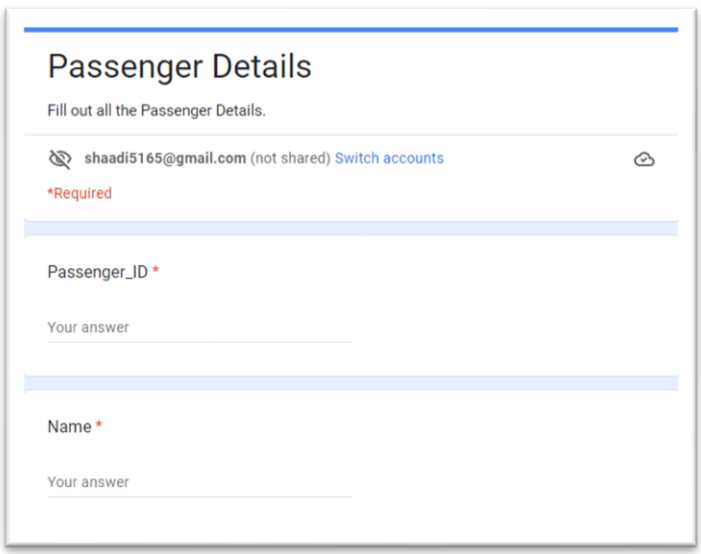


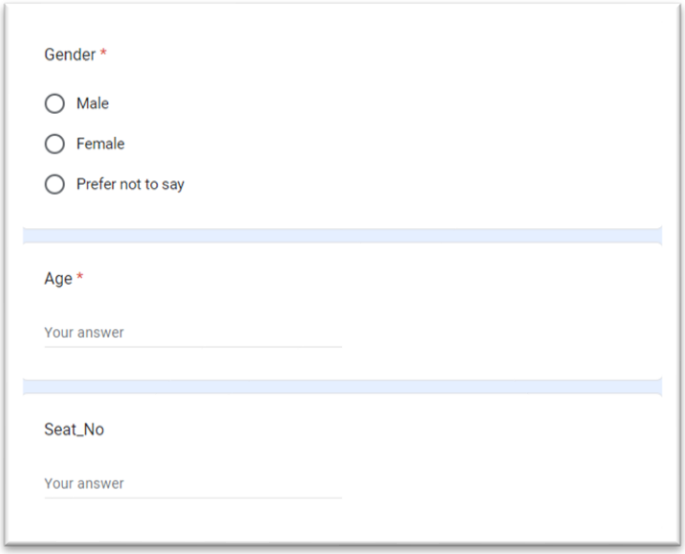


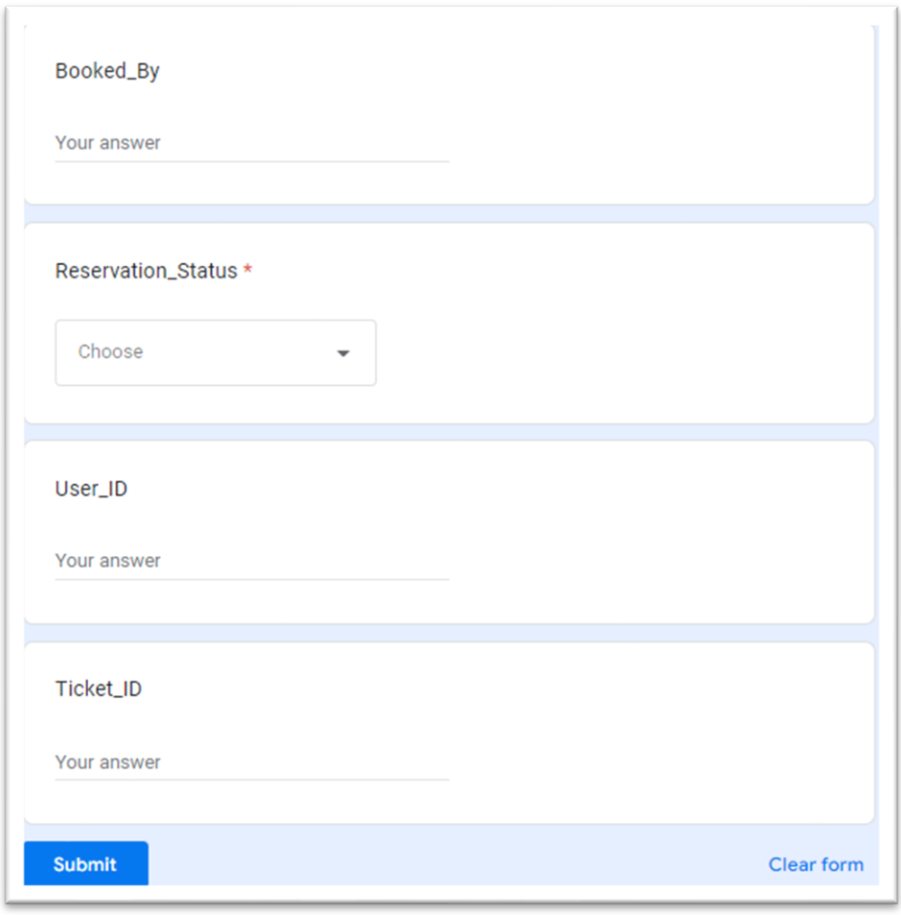




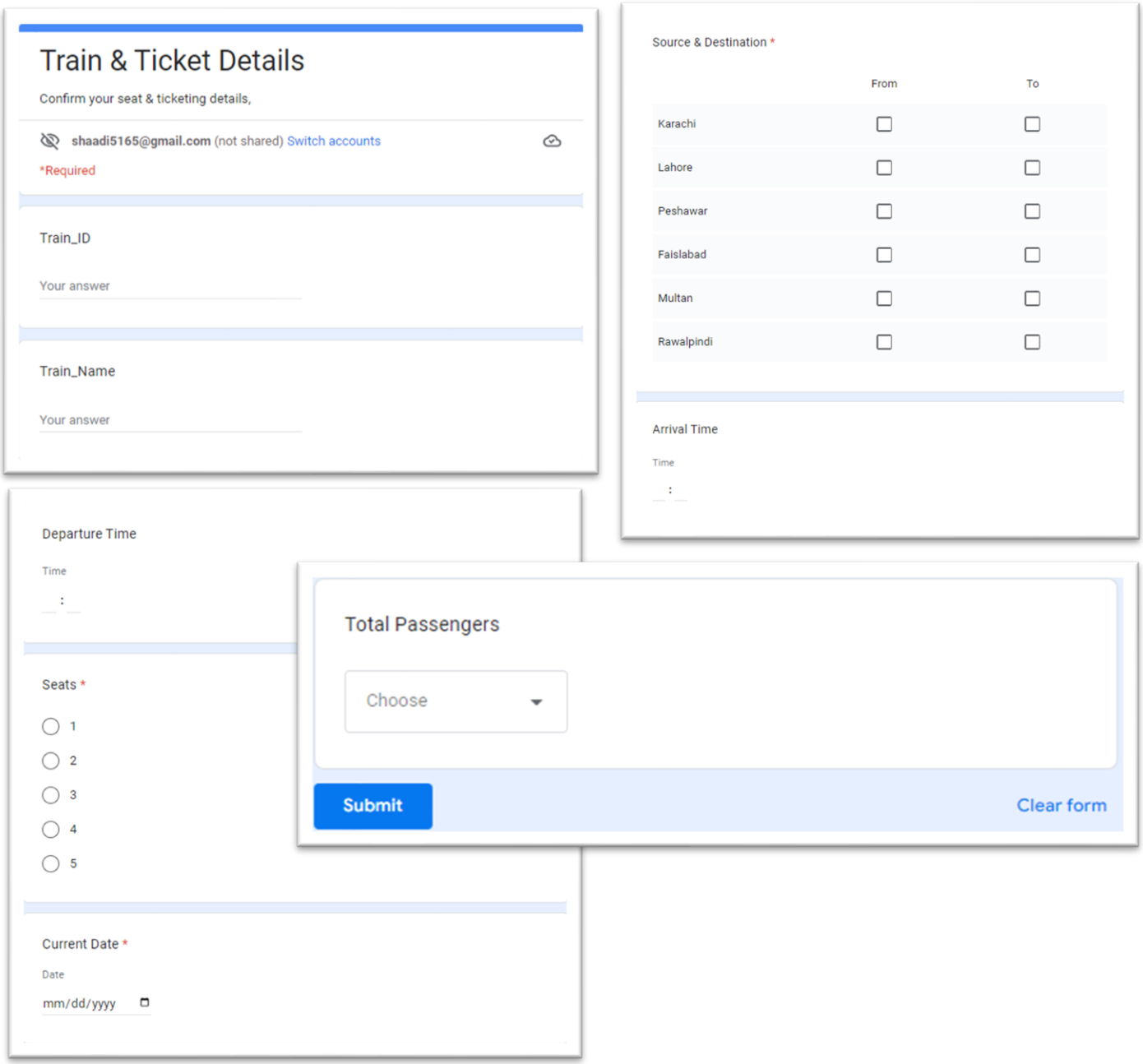
Passenger Details:







Train & Ticketing Details:



10. Architecture

The Three-Level ANSI-SPARC Architecture for Train Ticketing System:

- External Level
- Conceptual Level
- Internal Level

1) Internal Level:

In this level we will describe the internal view of the database and its physical representation on the computer. Thus, internal level has an internal schema which describes the physical storage structure of the database. Here we have entities like Train, Station, Passenger and Tickets and their internal view or scheme is given here:

STORED_STATION record length 74
Number: 4 decimal offset 0 unique
Name: string length 20 offset 4
Location: string length 10 offset 64
No of Trains: 4 decimal
Time of next arrival: string of length 20
Time of next departure: String of length 20

STORED_TRAIN record length 64
Number: 4 decimal offset 0 unique
Station Name: string length 20 offset 4
No of seats: 4 decimal
Passengers : 2 decimal
Time of arrival: string of length 20
Time of departure: String of length 20

STORED_PASSENGER record length 97
CNIC Number: 4 decimal offset 0 unique
First Name : string length 20 offset 4
Last Name: string length 20 offset 4
Gender : string of length 10 offset 2
City: string length 10 offset 64
Address: string length 29 offset 4
Phone no: 4 decimal offset 0 unique
Source: String of length 20
Destination: String of length 20

STORED_TICKET record length 84
Number: 4 decimal offset 0 unique
User Name: string length 20 offset 4
User id: 4 decimal offset 0 unique
User CNIC: 8 decimal offset 0 unique
Booked by: string length 20 offset 4
Destination: string length 10 offset 64
No of Train: 4 decimal offset 0 unique
Seat No: 2 decimal offset 0 unique

2) Conceptual Level:

In this level we will describe the Conceptual view of the database. Describes what data is stored in the database and the relationships among the data. The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level. Here we will design the Conceptual schema of our database.

STATION
Number: Integer[5] key
Name: string[20]
Location: string[10]
No of Trains: Integer[2]
Time of next arrival: String[22]
Time of next departure: String[20]

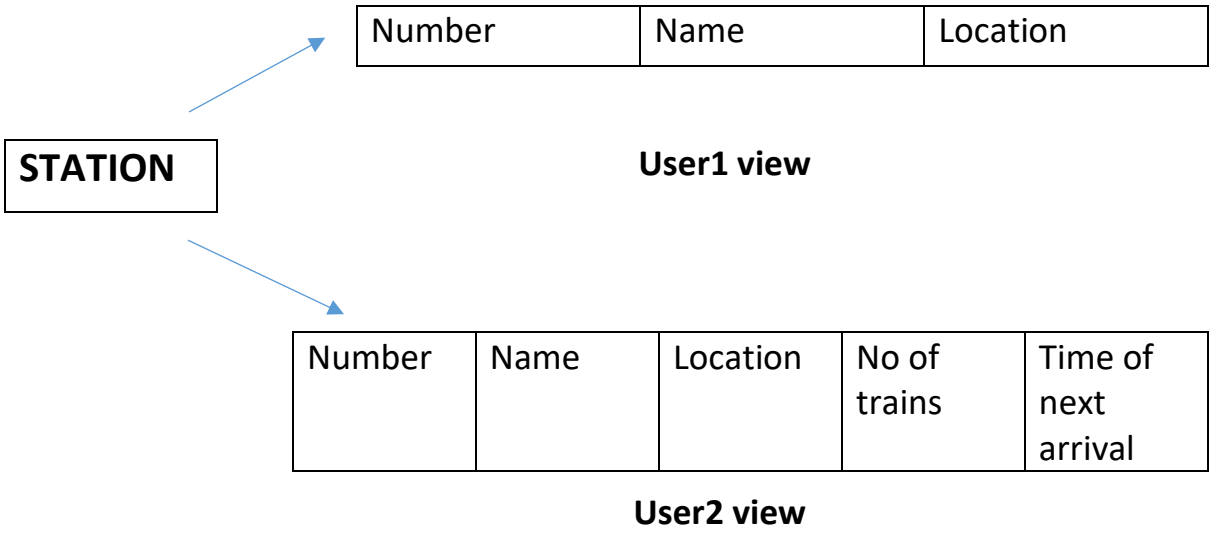
TRAIN
Number: integer[4] key
Station Name: string [20]
No of seats: integer[2]
Passengers : integer[2]
Time of arrival: string[20]
Time of departure: String[20]

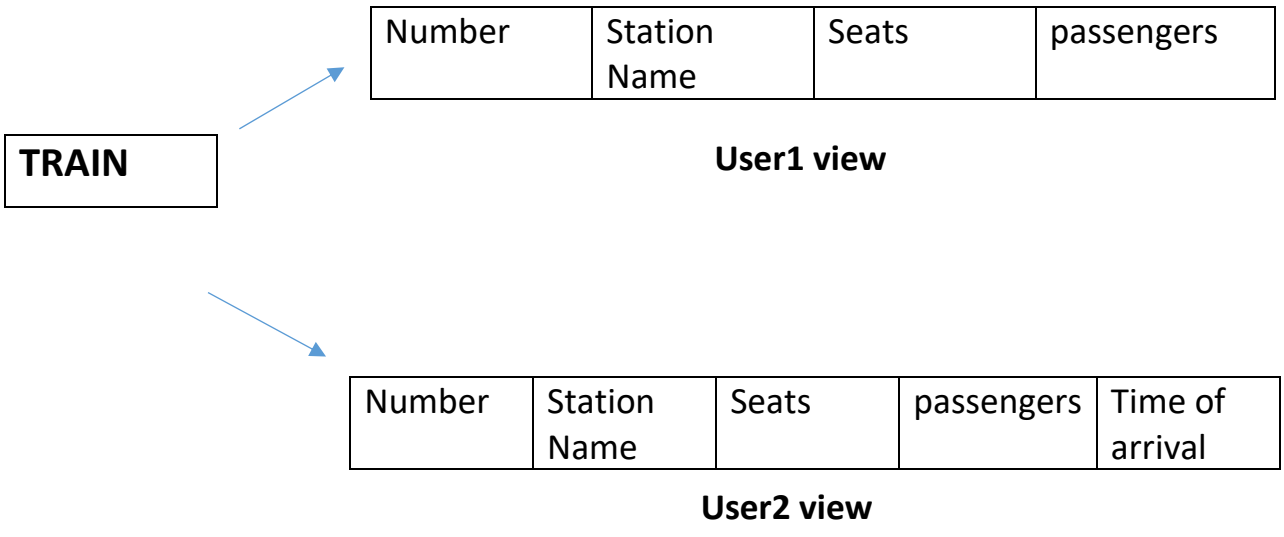
PASSENGER
CNIC Number: integer[4] key
First Name : string [20]
Last Name: string[20]
Gender : string[10]
City: string[10]
Address: string [29]
Phone no: integer[4]
Source: String[20]
Destination: String [20]

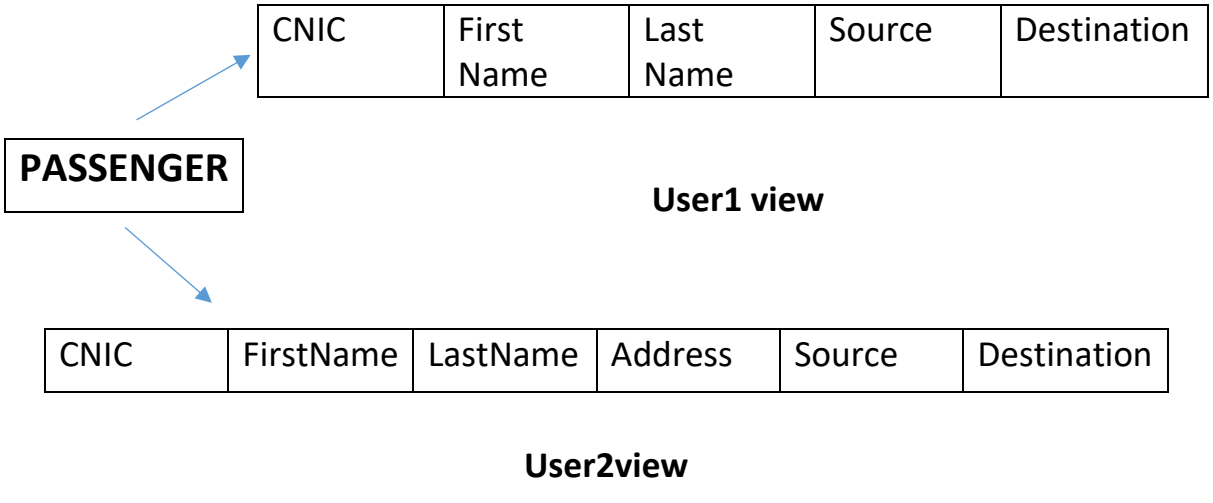
TICKET
Number: integer[4] key
User Name: string [20]
User id: integer[4]
User CNIC: integer[8]
Booked by: string [20]
Destination: string [10]
No of Train: integer[4]
Seat No: integer[2]

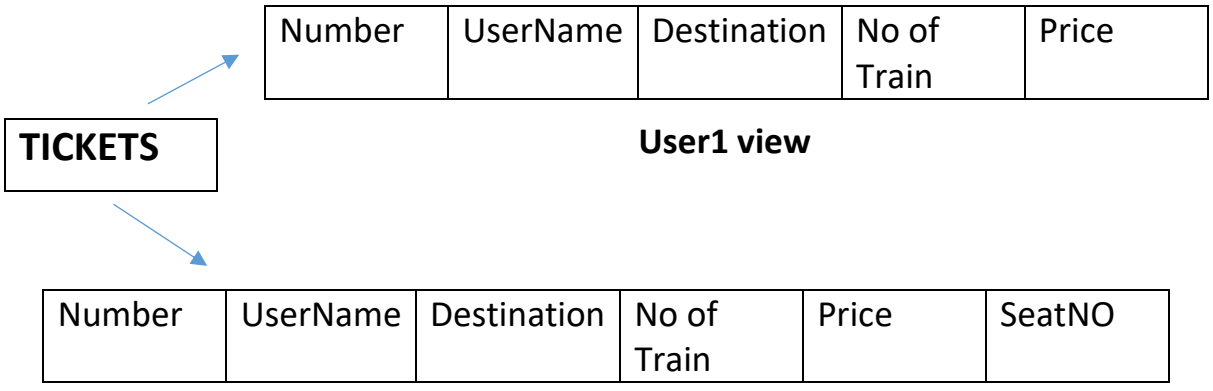
3) External Level:

External level is the users' view of the database. This level describes the part of the database that is relevant to each user. At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database. An external schema is also known as view schema. Here we will design the external schema of Station, Train, Passenger, and Ticket.


Number
Name
Location
STATION
User1 view
Number
Name
Location
No of trains
Time of next arrival
User2 view


Number
Station Name
Seats
passengers
TRAIN
User1 view
Number
Station Name
Seats
passengers
Time of arrival
User2 view


CNIC
First Name
Last Name
Source
Destination
PASSENGER
User1 view
CNIC
FirstName
LastName
Address
Source
Destination
User2view



---

11. 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 cand 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.

-- Thank You 😊 --