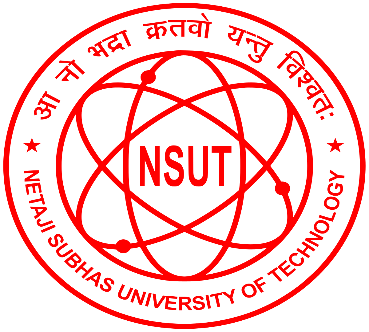
**Netaji Subhas University of Technology**



**Database Management System**

**B. Tech - 4th Semester**

**Project Based Lab**

**ON**

**RAILWAY RESERVATION SYSTEM**

**Submitted by**

Section – 1

Group: 2

|  |  |  |
| --- | --- | --- |
| **Roll Number** | **Student Name** | **Department** |
| 2020UEE4573 | Shiv Kumar | EE |
| 2020UEE4553 | Shubham Gaur | EE |
| 2020UEE4571 | Aryan Nagar | EE |

|  |
| --- |
| TABLE OF CONTENT |
| Introduction |
| Project Description |
| List of Entities & Attributes |
| ER Diagram |
| Schema Diagram |
| Tables and Data |
| Create & Insert SQL Queries |

|  |
| --- |
| 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 programmes, ensuring that data is consistently organized and remains easily accessible. 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.

|  |
| --- |
| Project Description |

* This project is developed by us to solve a major issue in the railways which was the old system.
* Like at initial state ticket were booked manually and one ticket can be booked at a time which eventually decreasing the efficiency but the main need of this railway reservation system needed when the old system affects the growth of our country.
* Think about how less people travel due to lack of information and performance of our old system.  
  Now you all agree that there is a need for a new and better system.
* The railway reservation system facilitates the passengers to enquiry about the train available on the basis of source and destination, enquiry about the status of the booked ticket, whether it is valid or not.  
  This project contains an introduction to the railway reservation system. It is the computerized system of reserving the seats of train seats in advance. It is mainly used for long routes. Online reservation has made the process for the reservation of seats very much easier than ever before.

List of Assumption Since the reservation system is very large in reality, it is not feasible to develop the case study to that extent and prepare documentation at that level. Therefore, a small sample case study has been created to demonstrate the working of the reservation system. To implement this sample case study, some assumptions have made, which are as follows:

* The number of trains has restricted to 5.
* Maximum of 7-8 Queries are performed in this project.
* A total of 6 tables are used in this project-

1. User (Anybody which has travelled in past)
2. Passenger(Anyone who have booked or ever tried to book a ticket recently)
3. Station(Have the data of all stations)
4. Train(Have the data of all trains)
5. Train\_Status(Useful information about each train)
6. Ticket(Have the data of all tickets which are booked or Cancelled)

|  |
| --- |
| List of Entities and Attributes |

|  |
| --- |
| USER |
| User id |
| Name |
| Current Email |
| City |
| State |
| Pin code |
| Gender |

|  |
| --- |
| PASSENGERS |
| Pid |
| Name |
| Gender |
| Age |

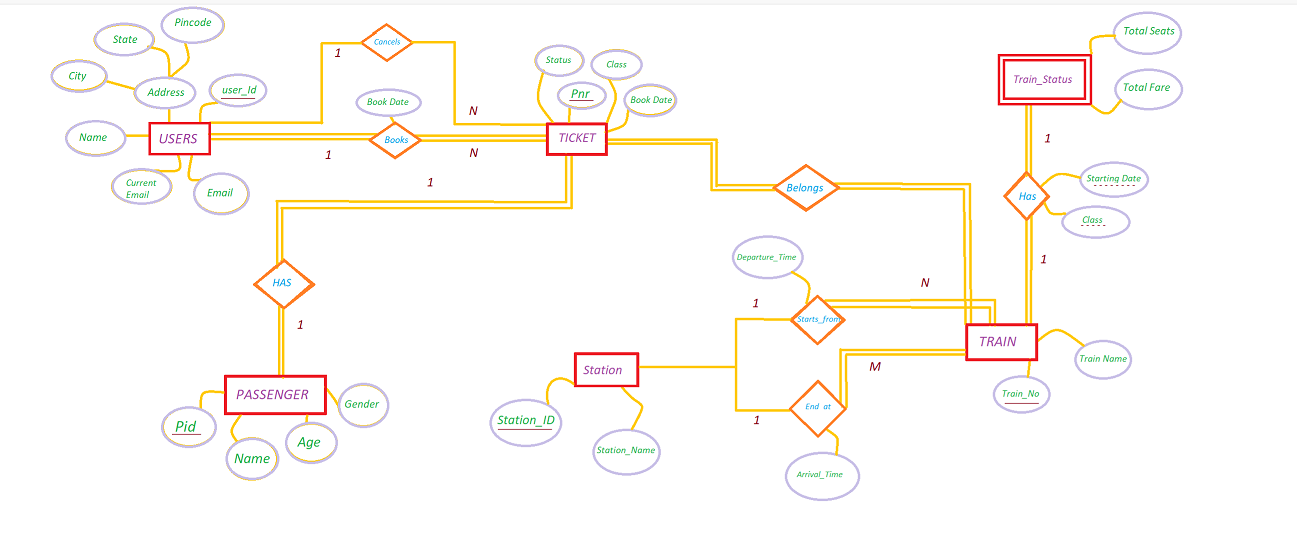
|  |
| --- |
| TICKET |
| Pnr |
| Class |
| Status |
| Book Date |
| Train No |
| User ID |
| Pid |

|  |
| --- |
| TRAIN |
| Train No |
| Name |
| Source Id |
| Destination Id |
| Arriving Time |
| Departure Time |

|  |
| --- |
| Train Status |
| Train No |
| Starting Date |
| Class |
| Total Seat |
| Total Fare |

|  |
| --- |
| STATION |
| Station Id |
| Station Name |

|  |
| --- |
| ER DIAGRAM |



|  |
| --- |
| Schema Diagram |

|  |
| --- |
| TICKET |
| Pnr |
| Class |
| Status |
| Book Date |
| Train No |
| User ID |
| Pid |

|  |
| --- |
| USER |
| User id |
| Name |
| Current Email |
| City |
| State |
| Pin code |
| Gender |

|  |
| --- |
| STATION |
| Station Id |
| Station Name |

|  |
| --- |
| PASSENGERS |
| Pid |
| Name |
| Gender |
| Age |

|  |
| --- |
| Train Status |
| Train No |
| Starting Date |
| Class |
| Total Seat |
| Total Fare |

|  |
| --- |
| TRAIN |
| Train No |
| Name |
| Source Id |
| Destination Id |
| Arriving Time |
| Departure Time |

|  |
| --- |
| TABLE AND DATA |

|  |
| --- |
| *USER* |

CREATE TABLE User (

User\_id INT ,

Name VARCHAR(20) NOT NULL,

Current\_Email VARCHAR(20),

City VARCHAR(20),

State VARCHAR(20),

Pincode INT,

Gender CHAR(1),

PRIMARY KEY (User\_id)

);

INSERT INTO User(User\_id,Name,Current\_Email,City,State,Pincode,Gender)

VALUES

(10,"Shiv","Shiv@gmail.com","Hisar","Haryana",110059,"M"),

(11,"Aryan","Aryan@gmail.com","Noida","UP",110059,"M"),

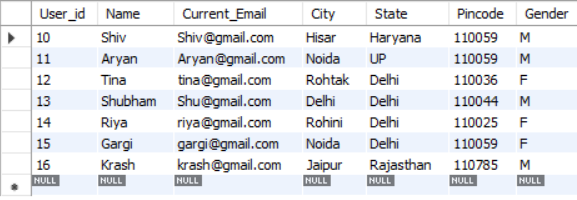
(13,"Shubham","Shu@gmail.com","Delhi","Delhi",110044,"M"),

(12,"Tina","tina@gmail.com","Rohtak","Delhi",110036,"F"),

(14,"Riya","riya@gmail.com","Rohini","Delhi",110025,"F"),

(15,"Gargi","gargi@gmail.com","Noida","Delhi",110059,"F"),

(16,"Krash","krash@gmail.com","Jaipur","Rajasthan",110785,"M");



|  |
| --- |
| *PASSENGER* |

CREATE TABLE Passenger (

pid INT,

Name VARCHAR(20),

Gender CHAR(1),

Age INT,

PRIMARY KEY (pid)

);

INSERT INTO Passenger(pid,Name,Gender,Age)

VALUES

(1000,"Shubham","M",46),

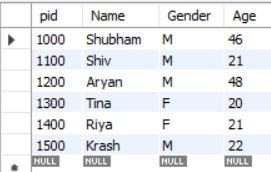
(1100,"Shiv","M",21),

(1200,"Aryan","M",48),

(1300,"Tina","F",20),

(1400,"Riya","F",21),

(1500,"Krash","M",22);

****

|  |
| --- |
| *STATION* |

CREATE TABLE Station (

Station\_id INT,

Station\_name VARCHAR(20),

PRIMARY KEY (Station\_id)

);

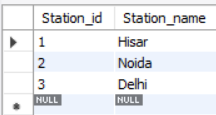
INSERT INTO Station(Station\_id,Station\_name)

VALUES

(1,"Hisar"),

(2,"Noida"),

(3,"Delhi");

****

|  |
| --- |
| *TRAIN* |

CREATE TABLE Train (

Train\_no INT PRIMARY KEY,

Name VARCHAR(20),

Source\_id INT,

Dest\_id INT,

Arr\_time TIME,

Dept\_time TIME

);

INSERT INTO Train (Train\_no,Name,Source\_id,Dest\_id,Arr\_time,Dept\_time)

VALUES

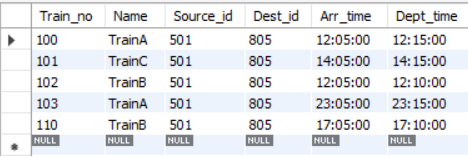
(100,"TrainA",501,805,"12:05:00","12:15:00"),

(102,"TrainB",501,805,"12:05:00","12:10:00"),

(101,"TrainC",501,805,"14:05:00","14:15:00"),

(110,"TrainB",501,805,"17:05:00","17:10:00"),

(103,"TrainA",501,805,"23:05:00","23:15:00");



|  |
| --- |
| *TICKET* |

CREATE TABLE Ticket (

Pnr INT,

Class CHAR(10),

Status varchar(30),

Book\_date DATE,

Train\_no INT,

User\_id INT,

pid INT,

FOREIGN KEY (pid)

REFERENCES Passenger (pid),

FOREIGN KEY (User\_id)

REFERENCES User (User\_id),

FOREIGN KEY (Train\_no)

REFERENCES Train (Train\_no)

);

INSERT INTO Ticket (pnr,Class,Status,Book\_date,Train\_no,User\_id,pid)

VALUES

(1,"AC","Confirmed","2022-04-15",100,11,1200),

(3,"NAC","Cancelled","2022-04-18",102,12,1300),

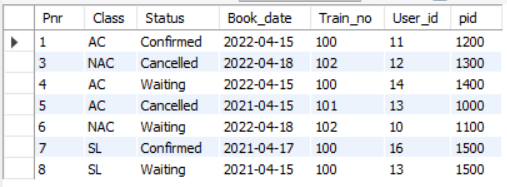
(4,"AC","Waiting","2022-04-15",100,14,1400),

(5,"AC","Cancelled","2021-04-15",101,13,1000),

(6,"NAC","Waiting","2022-04-18",102,10,1100),

(7,"SL","Confirmed","2021-04-17",100,16,1500),

(8,"SL","Waiting","2021-04-15",100,13,1500);



|  |
| --- |
| *TRAIN\_STATUS* |

CREATE TABLE Train\_status (

Train\_no INT,

Starting\_date DATE,

Class CHAR(2),

Total\_seat INT,

Total\_fare DECIMAL(6 , 2 )

);

INSERT INTO Train\_status (Train\_no,Starting\_date,Class,Total\_seat,Total\_fare)

VALUES

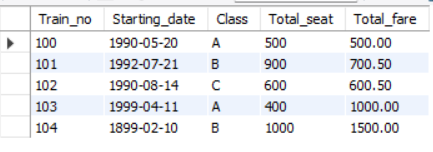
(100,"1990-05-20","A",500,500.00),

(101,"1992-07-21","B",900,700.50),

(102,"1990-08-14","C",600,600.50),

(103,"1999-04-11","A",400,1000.00),

(104,"1899-02-10","B",1000,1500.00);



|  |
| --- |
| CREATE AND INSERT SQL QUERIES |

1. Retrieve all the User\_id and user name that booked the confirm tickets for any class of train.

Ans: -

select u.User\_id,name from User as u join Ticket as t on

t.User\_id=u.User\_id where status='Confirmed' ;

1. Details of the passenger whose ticket has been cancelled.

Ans: -

select Passenger.Name,Passenger.pid,Ticket.pnr,Ticket.status

from Passenger join Ticket on Passenger.pid=Ticket.pid where

status='Cancelled';

1. Details about the passenger that booked a ticket for a particular train ........

Ans: -

select Passenger.\* from Passenger join (select

Pid,Ticket.Train\_no from Ticket join (select Train\_no from Train

where Name ='TrainA') as d on

d.train\_no=Ticket.Train\_no) as p on p.Pid=Passenger.Pid;

1. No. Of booked ticket on the particular date of a particular train.

Ans: -

select Train\_no,Book\_date ,count(pnr) as "total booked" from

Ticket where Book\_date='2022-04-15' group by

Train\_no,Book\_date;

1. Retrieve the details of user who booked more than 1 ticket.

Ans: -

select d.\*,Name from User join (select User\_id,count(pnr)

from Ticket group by User\_id having count(pnr)>=2 ) as d on

d.User\_id=user.User\_id

6. Total booking of a AC and SL class for a train

Ans: -

select Train\_no,class,count(pnr) from Ticket group by

Train\_no,class;

7. Details of passengers whose age is above 45 years (Senior citizens) with name and pnr.

Ans: -

select d.\*,pnr from Ticket join (select Name,pid,age from

Passenger where age>=45) as d on d.pid=Ticket.pid ;

8. Retrieve User\_id, user name and total number of confirm tickets booked by that user for a particular class.

Ans: -

select name ,d.\* from User join (select User\_id,class,

count(pnr) from Ticket where status='Confirmed' group by User\_id ,

class) as d on d.User\_id=User.User\_id;