**Chipi techonology**



<http://chipitechnologies.com/>

Bachelor of Computer Applications

Data Base Management System

Project Based

On

Railway reservation system

Submitted by vishal kumar

DECLARATION

We hereby declare that the project titled “Railway Reservation System” submitted to [ Chipi Techonology ], is a record of an original work done by me under the guidance of [ Krishna Kumar Sinha ]. This project is submitted in partial fulfillment of the requirement for the certificate of the course [ DBMS] in [ Chipi Techonology].

GUIDE/PROFESSOR

[ Krishna Kumar Sinha ]

Date:

[22/06/24]

ACKNOWLEDGMENTS……………………………………………………54

**TABLE OF CONTENTS**

Abstract………………………………………………………………………54

Introduction………………………………………………………………… 54

PROJECT DESCRIPTION…………………………………………………54

List of Entities & Attributes………………………………………………….54

ER Diagram (Conceptual Model)…………………………………………..54

Create & Insert SQL Queries……………………………………………….54

Conclusion……………………………………………………………………54

**ACKNOWLEDGMENTS**

It is a great pleasure for me to express my gratitude to our honorable president Krishna Kumar Sinha Sir for giving me opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our CEO of CHIPI TECHOMOLOGY for his administration towards our academic growth.

Finally , it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

The Railway Reservation System facilitates the passengers to enquire

**ABSTRACT**

about the trains available on the basis of source and destination , Booking

and Cancellation of tickets , enquire about the status of the booked tickets ,

etc. The aim of case study is to design and develop a database maintaining

the records of different trains , trains status, and passengers.

This project contains Introduction to the Railway reservation system.

It is computerized system of reserving the seats of train seats in advanced.

It is mainly used for long route. Online reservation has made the process

for the reservation of seats very much easier than ever before.

In our country India , there are number of counters for the reservation of the

seats and one can easily make reservation and get tickets. Then this

project contains entity relationship model diagram based on railway

reservation system and introduction to relation model . There is also

design of the database of the railway reservations system based on the

relation model. There is also design of the database of the railway

reservation system based on the relation model. Example of some SQL

queries to retrieves data from rail management database.

Database is a organized collection of data . This data is typically organized

**INTRODUCTION**

to model aspects of reality in a way that supports processes requiring

information. A DBMS essentially serves as an interface between the

database and end users or application programs , ensuring the data is

consistently organized and remains easily accessible. The DBMS manages

three important things : the data , the database engine that allow 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 railway

Reservation System is to reduce the manual errors involved in the booking

and cancelling of tickets and make it convenient for the customer and

providers to maintain the data about their customer and also about the

seats available at them. Due to automation many loopholes that exist in

the manual maintenance of the record can be removed . The speed of

obtaining and processing the data will be fast . for further expansion the

proposed system can be web enabled so that clients make various

enquiries about trains between stations . Due to this , sometimes a lot of

problem occur and they are facing many disputes with customers . To

solve the above problem , we design a database which includes customers

details , availability of seats in trains , no of trains and their details.

This project is about creating the database about Railway Reservations

**PROJECT DESCRIPTION**

System.

The railway reservations system facilitates the passenger to enquire about the train available on the basis of train number , source and destination , to check the ticket is available or not . , enquire about the status of the booked tickets ,

etc. The aim of case study is to design and develop a database maintaining

the records of different trains , trains status, and passengers. The record of

train includes its number , name , source destination, and day 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.

Passengers can be book their tickets for the train in which seats are

available . For this, passengers has to be provide the desired train number

and the date for which tickets is to be booked. Before booking a tickets for

a passengers , the validity of train number and booking date is checked .

Once the train number and booking date are validated , it is checked

whether the seat available . If yes, the tickets is booked with confirm status

and corresponding tickets ID is generated which is stored along with other

details of the passengers . The tickets once booked can be cancelled at

any time. For this , the passenger has to provide the tickets ID ( the unique

key ). The tickets ID is searched and the corresponding record is deleted .

With this , the first tickets with waiting status also gets confirmed .

|  |  |
| --- | --- |
| Entities  **LIST OF ENTITIES & ATTRIBUTES** | Attributes |
| Users | User\_id-  First\_name  Last\_name  Gender  Age  Email  Aadhar\_no  Mobile\_no  City  State  Pincode |
| Passenger | Passenger\_ID  Name  Gender  Age  Pnr\_no  Seat\_no  Booked\_by  Reservation \_ status |
| Train | Train\_no  Train\_name  Source  Destination  Arrival\_time  Departure\_time  Availabilty\_of\_seats  Train\_no  A\_seats1  A\_seats2  A\_seats3  B\_seats1  B\_seats2  B\_seats3  W\_seats1  W\_seats2  W\_seats3 |
| Station | Name  Station\_no  Train\_no  Arrival time  Hault |
| Tickets | Tickets\_ID  Train \_no  Booked\_user  Status  No\_of\_passenger |

**ER DIAGRAM (CONCEPTUAL MODEL)**

Cancel

reaches

starts

Stop\_at

BOOKED

destination

source

Passengers

Tickets

Train

Station

Users

Relation between two entities

1. In above Er diagram “Destination” is the relation between station and

tickets , “Source” is relation between station and tickets.

1. “Reaches” is the relation between Train and Station , “Starts at” is the

relation between train and station and “Stop at” is the relation

between train and station.

1. “Cancel” is the relation between in user and tickets and “Booked” is

The relation between user and tickets.

Cardinality Ratio

1. User and tickets

Cardinality ratio:- (One to Many)

1. Train and Station

Cardinality ratio:- ( One to Many)

1. Passenger and train

Cardinality ratio:- (Many to Many)

1. Train and tickets

Cardinality ratio:- (One to Many)

Create table for:-

**CREATE & INSERT QUERIES**

1. Passeneger
2. Tickets
3. Train
4. Users
5. station

* For users

CREATE TABLE Users (

user\_id INT PRIMARY KEY,

first\_name VARCHAR(255),

last\_name VARCHAR(255),

gender VARCHAR(10),

age INT,

email VARCHAR(255),

aadhar\_no VARCHAR(12),

mobile\_no VARCHAR(10),

city VARCHAR(255),

state VARCHAR(255),

pincode VARCHAR(6),

);

* For passengers

CREATE TABLE Passenger (

passenger\_id INT PRIMARY KEY,

name VARCHAR(255),

gender VARCHAR(10),

age INT,

pnr\_no VARCHAR(10),

seat\_no VARCHAR(10),

booked\_by INT,

reservation\_status VARCHAR(20),

FOREIGN KEY (booked\_by) REFERENCES Users(user\_id)

);

* For train

CREATE TABLE Train (

train\_no INT PRIMARY KEY,

train\_name VARCHAR(255),

source VARCHAR(255),

destination VARCHAR(255),

arrival\_time TIME,

departure\_time TIME,

availability\_of\_seats INT,

a\_seats1 INT,

a\_seats2 INT,

a\_seats3 INT,

b\_seats1 INT,

b\_seats2 INT,

b\_seats3 INT,

w\_seats1 INT,

w\_seats2 INT,

w\_seats3 INT

);

* For station

CREATE TABLE Station (

station\_no INT PRIMARY KEY,

name VARCHAR(255),

train\_no INT,

arrival\_time TIME,

hault VARCHAR(20),

FOREIGN KEY (train\_no) REFERENCES Train(train\_no)

);

* For tickets

CREATE TABLE Tickets (

tickets\_id INT PRIMARY KEY,

train\_no INT,

booked\_user INT,

status VARCHAR(20),

no\_of\_passenger INT,

FOREIGN KEY (train\_no) REFERENCES Train(train\_no),

FOREIGN KEY (booked\_user) REFERENCES Users(user\_id)

);

\*Queries related to above table\*

Insert query for users , train and passengers

--For users—

Insert into Users

(

user\_id , first\_name , last\_name , aadhar\_no , gender , age , mobile\_no , email , city , state , pincode

)

values

(

1701 , 'aman' , 'kumar' , '930495047273' , 'male' , 19 , '9113326518' , 'vishalkr16feb@gmail.com' , 'danapur' ,

'patna' , '801503'

),

(

1702 , 'rohit' , 'kumar' , '799244596465' , 'male' , 45 , '9304110809' , 'bittukr07may@gmail.com' , 'dalupura' ,

'delhi' , '801604'

),

(

1703 , 'vanshika' , 'soni' , '620714761213' , 'female' , 20 , '7050111203' , 'vanshika09@gmail.com' , 'saguna' ,

'supaul' , '801707'

)

--for train--

INSERT INTO Train

(

train\_no, train\_name, arrival\_time, departure\_time, availability\_of\_seats,

a\_seats1, a\_seats2, a\_seats3, b\_seats1, b\_seats2, b\_seats3, w\_seats1, w\_seats2, w\_seats3

)

VALUES

(

12711, 'rajdhani\_exp', '06:00:00', '08:00:00', 'yes' , 10, 4, 0, 1, 2, 0, 6, 7, 3

),

(

12712, 'cormandal exp', '09:00:00', '10:00:00', 'no' , 15, 12, 13, 16, 18, 17, 11, 14, 19

);

--for passengers--

INSERT INTO Passenger (passenger\_id, name, gender, age, pnr\_no, seat\_no)

VALUES

(5001, 'vanshika', 'Female', 22, 'PNR123456', 'A3'),

(5002, 'rohit', 'male', 23, 'PNR654321', 'A1'),

(5003, 'aman', 'male', 25, 'PNR124567', 'A2');

--This is function to see all the table--

select\*from Users

select\*from Passenger

select\*from Train

select\*from Station

select\*from Tickets

QUERIES OPERATION IN PASSENGERS

--PRINT DETAIL OF PASSENGER USING PASSENGER ID--

SELECT \* FROM Passenger WHERE passenger\_id = 5001;

--GET ALL FEMALE PASSENEGER--

SELECT \* FROM Passenger WHERE gender = 'Female';

--Get all passengers with a specific PNR number--

SELECT \* FROM Passenger WHERE pnr\_no = 'PNR123456';

--Get all passengers with a specific seat number--

SELECT \* FROM Passenger WHERE seat\_no = 'A3';

-- count of male passengers--

SELECT COUNT(\*) FROM Passenger WHERE gender = 'Male';

--AVERAGE OF PASSENEGR--

SELECT AVG(age) FROM Passenger;

--get the passenger details who booked the ticket with PNR number 'PNR654321'--

SELECT \* FROM Passenger WHERE pnr\_no = 'PNR654321';

--count of passengers who are 25 years old--

SELECT COUNT(\*) FROM Passenger WHERE age = 25;

--Get the passenger details who are sitting in seat 'A1'--

SELECT \* FROM Passenger WHERE seat\_no = 'A1';

QUERIES OPERATION IN USERS

--Get all users--

SELECT \* FROM Users

--Get a specific user by user\_id--

SELECT \* FROM Users WHERE user\_id = 1701

--Get all male users--

SELECT \* FROM Users WHERE gender = 'male'

--Get all users from a specific city--

SELECT \* FROM Users WHERE city = 'danapur'

-- Get all users from a specific state--

SELECT \* FROM Users WHERE state = 'patna'

-- Get the count of female users--

SELECT COUNT(\*) FROM Users WHERE gender = 'female'

--Get the average age of all users--

SELECT AVG(age) FROM Users

--Get the user details who has aadhar number '930495047273'--

SELECT \* FROM Users WHERE aadhar\_no = '930495047273'

--Get the count of users who are 20 years old--

SELECT COUNT(\*) FROM Users WHERE age = 20

QUERIES OPERATION IN TRAIN

SELECT \* FROM Train

--Get train details by train number--

SELECT \* FROM Train WHERE train\_no = 12711

--Get train names that start with 'r'--

SELECT train\_name FROM Train WHERE train\_name LIKE 'r%'

--Get trains with available seats--

SELECT \* FROM Train WHERE availability\_of\_seats = 'yes'

--Get total number of seats in each class for a specific train--

SELECT

train\_no,

train\_name,

(a\_seats1 + a\_seats2 + a\_seats3) AS total\_a\_seats,

(b\_seats1 + b\_seats2 + b\_seats3) AS total\_b\_seats,

(w\_seats1 + w\_seats2 + w\_seats3) AS total\_w\_seats

FROM Train

WHERE train\_no = 12711;

-- Get trains with departure time between 08:00:00 and 10:00:00--

SELECT \* FROM Train WHERE departure\_time BETWEEN '08:00:00' AND '10:00:00'

--Update availability of seats for a specific train--

UPDATE Train SET availability\_of\_seats = 'no' WHERE train\_no = 12711;

--Delete a train record--

DELETE FROM Train WHERE train\_no = 12712