**Online Railway Ticket Booking system**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

## Bachelor of Technology

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

Poonam

AP20110010365

Under the Guidance of

## Prof. Dr Rajiv Senapati

Department of Computer science Engineering

## SRM University–AP

## Neerukonda,Mangalagiri,Guntur

**Andhra Pradesh – 522 240**

**[December, 2022]**

# **Certificate**

Date: 12-DEC-22

This is to certify that the work present in this Project entitled “**Online Railway Ticket Booking system**” has been carried out by **Poonam** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

**Supervisor**

(Signature)

Dr. Rajiv Senapati

Associative Professor of Computer Science Department

# **Acknowledgements**

The successful presentation of the DBMS MINI PROJECT would be incomplete without the mention of the people who made it possible and whose constant guidance crowned my effort with success. It is great pleasure for me to express my gratitude towards **Dr. Rajiv Senapati sir** for giving the opportunity and platform with facilities in accomplishing the project based laboratory report. Indeed it was a great learning opportunity. This subject has gave me a broader view to explore this field.

# **Table of Contents**

Certificate 1

[Acknowledgements **3**](#_TOC_250011)

[Table](#_TOC_250010) of Contents **4**

Abstract 5

Introduction 6

1. Methodology 7
   1. Entity List 7
      1. Entities With Attributes 7

2.2 ER Model 9

* 1. ER Model to Relational Model 10
  2. Normalization 11
  3. SQL Code 14

1. SQL Queries 20

4. Conclusion 21

5. References 22

**ABSTRACT**

The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket etc.

In the given project we will be developing only backend which will help users to find train details, book and cancel tickets and the exact rates of their tickets to the desired destination.

**INTRODUCTION**

There are many different ways to store and retrieve the data. Older versions like using files for data organizing have many issues and take more manpower and time. Data Redundancy, Data Inconsistency, Data Isolation, Atomicity, Concurrent access, Security are the few tasks which becomes an anomaly by using files.

Database Management System (DBMS) helps to hold all above issues.

Online Railway Booking System is one of the applications where maintaining a database is required. In this report, the simple requirements are considered. This database system stores the data about the users, availability of trains and the information about the reserved tickets. Users can check for the train details and can reserve a ticket.

**2.METHODOLOGY**

**2.1 Entity List:**

1.USER

2.TRAIN

3.CLASS

4.TICKET

5.PAYMENT

**2.1.1 ENTITIES WITH THEIR ATTRIBUTES:**

**USER**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **DATA TYPE** | **CONSTRAINT** |
| USER\_ID | VARCHAR (10) | PRIMARY KEY |
| NAME | VARCHAR (20) | NOT NULL |
| AGE | INT |  |
| DOB | DATE | NOT NULL |
| ADDRESS | VARCHAR (30) | NOT NULL |
| GENDER | CHAR (6) | NOT NULL |
| MOBILE\_NO | VARCHAR (10) | NOT NULL |

**TRAIN**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **DATA TYPE** | **CONSTRAINT** |
| TRAIN\_NO | VARCHAR (10) | PRIMARY KEY |
| TRAIN\_NAME | VARCHAR (20) | NOT NULL |
| ARRIVAL\_TIME | TIME | NOT NULL |
| DEPT\_TIME | TIME | NOT NULL |
| SOURCE | VARCHAR (30) | NOT NULL |
| DESTINATION | VARCHAR (30) | NOT NULL |
| SEAT\_AVAILABLE | INT | NOT NULL |

**TICKET**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **DATA TYPE** | **CONSTRAINT** |
| PNR\_NO | VARCHAR (10) | PRIMARY KEY |
| TRAIN\_NO | VARCHAR (20) | NOT NULL |
| SOURCE | VARCHAR (30) | NOT NULL |
| DESTINATION | VARCHAR (30) | NOT NULL |
| SEAT\_NO | VARCHAR (30) | NOT NULL |
| PASSENGER\_NAME | VARCHAR (30) | NOT NULL |
| CLASS | VARCHAR (10) | NOT NULL |
| BOOKING\_DATE | DATE | NOT NULL |

**PAYMENT**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **DATA TYPE** | **CONSTRAINT** |
| TRANSACTION\_ID | VARCHAR (10) | PRIMARY KEY |
| AMOUNT | VARCHAR (20) | NOT NULL |
| ACCOUNT\_ID | VARCHAR (30) | NOT NULL |
| DATE\_TIME | VARCHAR (30) | NOT NULL |

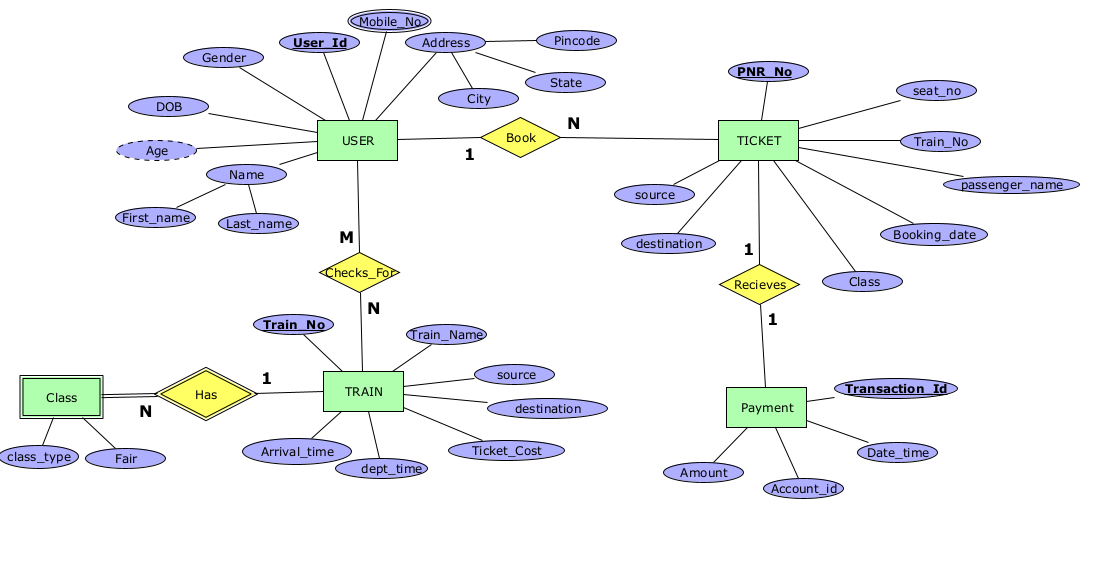
**CLASS**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE** | **DATA TYPE** | **CONSTRAINT** |
| TRAIN\_NO | VARCHAR (30) | NOT NULL |
| CLASS\_TYPE | VARCHAR (10) | NOT NULL |
| FARE | REAL | NOT NULL |

**2.2 CARDINALITY RELATIONSHIP &ER DIAGRAM:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ENTITY-1** | **ENTITY-2** | **RELATION** | **CARDINALITY** |
| USER | TRAIN | CHECKS\_FOR | M: N |
| USER | TICKET | BOOK | 1: N |
| TRAIN | CLASS | HAS | 1: N |
| TICKET | PAYMENT | RECIEVES | 1:1 |

**ER DIAGRAM:**



**RELATIONAL MODEL:** A relational database is a collection of tables. Every entity is represented as a table.

**ER TO RELATIONAL MODEL**

**Rules:**

1. An entity in the ER model is represented by a relational table in relational model.

2. All the attributes of ER model is represented in different columns as an attribute.

3. Primary key attribute of ER model is represented as primary key in relational model.

4. Composite key of ER model is split in different columns in relational model.

5.Dervied attribute must be dropped in relational model.

6. Multi valued attribute needs to be kept separate in a new table.

7. If the relationship between two entities is 1:1 then primary key of one relation becomes foreign key in another relation.

8. If the relationship between entities is 1:M or M:1 then the primary key attribute of one-sided relation becomes a foreign key in many sided relations.

9. If the relationship between entities is M:M then a new table needs to be created to represent that relation, where the new relation will have primary key of both relations as the foreign keys.

**Schema Diagram:** A database schema along with primary key and foreign key dependencies can be expressed pictorially by schema diagram

**USER**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| USER\_ID (P.K) | FNAME | LNAME | CITY | STATE | PINCODE | GENDER | DOB |

**USER\_CONTACT**

|  |  |
| --- | --- |
| USER\_ID (F.K) | MOB\_NO |

**TRAIN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TRAIN\_NO (P.K) | TRAIN\_NAME | ARRIVAL\_TIME | DEPT\_TIME | SOURCE | DESTINATION | TICKET\_COST |

**USER\_TRAIN**

|  |  |
| --- | --- |
| USER\_ID(FK) | TRAIN\_NO(FK) |

**CLASS**

|  |  |  |
| --- | --- | --- |
| TRAIN\_NO (F.K) | CLASS\_TYPE | FARE |

**TICKET**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PNR\_NO(P.K) | USER\_ID (F.K) | TRAIN\_NO | CLASS | SEAT\_NO | PASSENGER\_NAME | DATE\_TIME | SOURCE | DEST |

**PAYMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAYMENT\_ID(P.K) | DATE\_TIME | AMOUNT | ACCOUNT\_ID | PNR\_NO(F.K) |

**Normalization**

Normalization is a process of analyzing and decomposing the complex relation which satisfy some constraint to form a simple relation.

There are 5 Normal forms:

1. First Normal Form (1NF)

2. Second Normal Form (2NF)

3. Third Normal Form (3NF) or Boyes Cord Normal Form (BCNF)

4. Forth Normal Form (4NF)

5. Fifth Normal Form (5F) or Project Join Normal Form (PJNF)

**FIRST NORMAL FORM(1NF)**

* A relation is said to be in 1NF, if it has got no non-atomic attributes i.e., which cannot be sub divided.
  + In this relational model all attributes in every relation are atomic so it’s in 1NF.

**SECOND NORMAL FORM (2NF)**

* A relation is in 1NF is said to be in 2NF, if it satisfies any one of the following conditions. They are,

1. The primary key consists of only one attribute.
2. There exists no non key attribute.
3. Every non key attribute present in relation should functionally depends upon full set of primary keys.

* All the relations in our model have only one attribute in primary key except USER\_CONTACT, USER\_TRAIN and CLASS.
* USER\_CONTACT and USER\_TRAIN has no non key attribute.
* CLASS non key attribute functionally depends on its set of primary keys.
* Each relation satisfies at least one of the conditions. So, model is in 2NF.

**THIRD NORMAL FORM (3NF)**

* A relation which is in 2NF is said to be in 3NF, if there exist no transitive functional dependency of any non-key attribute on the set of primary keys.
* Transitivity says that, if X->Y and Y->Z then X->Z.
* In USER relation, USER\_ID -> PINCODE and PINCODE -> STATE, CITY

So, the relation should be decomposed.

**USER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| USER\_ID(PK) | FNAME | LNAME | DOB | GENDER | PINCODE |

**USER\_PINCODE**

|  |  |  |
| --- | --- | --- |
| PINCODE(PK) | STATE | CITY |

* In TICKET relation, PNR\_NO -> TRAIN\_NO and

TRAIN\_NO -> SOURCE, DESTINATION, DATE\_TIME.

**TICKET**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PNR\_NO(PK) | USER\_ID(FK) | TRAIN\_NO | CLASS | PASSENGER\_NAME | SEAT\_NO |

**TICKET\_TRAIN\_NO**

|  |  |  |  |
| --- | --- | --- | --- |
| TRAIN\_NO (PK) | SOURCE | DESTINATION | DATE\_TIME |

**All Relations after normalization**

**USER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| USER\_ID(PK) | FNAME | LNAME | DOB | GENDER | PINCODE |

**USER\_PINCODE**

|  |  |  |
| --- | --- | --- |
| PINCODE(PK) | STATE | CITY |

**USER\_CONTACT**

|  |  |
| --- | --- |
| USER\_ID(FK) | MOBILE\_NO |

**TICKET**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PNR\_NO(PK) | USER\_ID(FK) | TRAIN\_NO | CLASS | PASSENGER\_NAME | SEAT\_NO |

**TICKET\_TRAIN\_NO**

|  |  |  |  |
| --- | --- | --- | --- |
| PNR\_NO (PK) | SOURCE | DESTINATION | DATE\_TIME |

**PAYMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| TRANSCATION\_ID(PK) | PNR\_NO(FK) | ACCOUNT\_ID | AMOUNT |

**TRAIN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TRAIN\_NO(P.K) | TRAIN\_NAME | ARRIVAL\_TIME | DEPT\_TIME | SOURCE | DESTINATION | SEAT\_AVAILABLE |

**USER\_TRAIN**

|  |  |
| --- | --- |
| USER\_ID(FK) | TRAIN\_NO(FK) |

**CLASS**

|  |  |  |
| --- | --- | --- |
| TRAIN\_NO(FK) | CLASS\_TYPE | FARE |

**SCHEMA DIAGRAM AFTER NORMALISATION**

**USER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| USER\_ID(PK) | FNAME | LNAME | DOB | GENDER | PINCODE |

**USER\_PINCODE**

|  |  |  |
| --- | --- | --- |
| PINCODE(PK) | STATE | CITY |

**USER\_CONTACT**

|  |  |
| --- | --- |
| USER\_ID(FK) | MOBILE\_NO |

**TICKET**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PNR\_NO(PK) | USER\_ID(FK) | TRAIN\_NO | CLASS | PASSENGER\_NAME | SEAT\_NO |

**PAYMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| TRANSCATION\_ID(PK) | PNR\_NO(FK) | ACCOUNT\_ID | AMOUNT |

**TICKET\_TRAIN\_NO**

|  |  |  |  |
| --- | --- | --- | --- |
| PNR\_NO (PK) | SOURCE | DESTINATION | DATE\_TIME |

**TRAIN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TRAIN\_NO(P.K) | TRAIN\_NAME | ARRIVAL\_TIME | DEPT\_TIME | SOURCE | DESTINATION | SEAT\_AVAILABLE |

**USER\_TRAIN**

|  |  |
| --- | --- |
| USER\_ID(FK) | TRAIN\_NO(FK) |

**CLASS**

|  |  |  |
| --- | --- | --- |
| TRAIN\_NO(FK) | CLASS\_TYPE | FARE |

**2.5 SQL Code**

**/\* creating database \*/**

create database project;

**/\* using created database \*/**

use project;

**/\*creating table for USER \*/**

create table USER (

user\_id varchar (10) primary key,

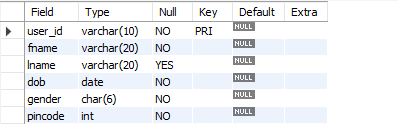
fname varchar(20) not null,

lname varchar (20),

dob date not null,

gender char(6) not null,

pincode int not null);



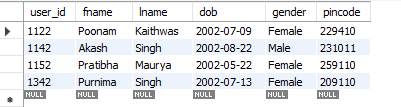
**/\* inserting data into user \*/**

insert into user values(1122,"Poonam","Kaithwas","2002-07-09","Female",229410);

insert into user values(1342,"Purnima","Singh","2002-07-13","Female",209110);

insert into user values(1152,"Pratibha","Maurya","2002-05-22","Female",259110);

insert into user values(1142,"Akash","Singh","2002-08-22","Male",231011);



**/\* Creating USER\_PINCODE table\*/**

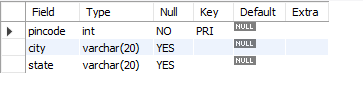
create table user\_pincode(

pincode int primary key,

city varchar(20),

state varchar (20)

);

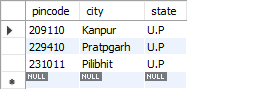


**/\* inserting values into user\_pincode table \*/**

insert into user\_pincode values(229410,"Pratpgarh","U.P");

insert into user\_pincode values(209110,"Kanpur","U.P");

insert into user\_pincode values(231011,"Pilibhit","U.P");



**/\*creating table for user\_contact \*/**

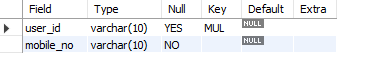
create table user\_contact (

user\_id varchar (10),

mobile\_no varchar (10) not null,

foreign key(user\_id) references user(user\_id)

);

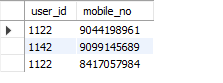


**/\* inserting values into user\_contact table \*/**

insert into user\_contact values(1122,9044198961);

insert into user\_contact values (1122,8417057984);

insert into user\_contact values (1142,9099145689);



**/\* Creating ticket table \*/**

create table ticket (

PNR\_NO varchar (20) primary key,

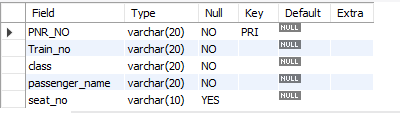
Train\_no varchar (20) not null,

class varchar (20) not null,

passenger\_name varchar (20) not null,

seat\_no varchar (10)

);

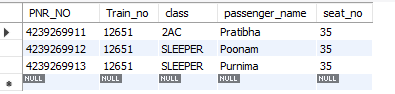


**/\*Inserting data into ticket table \*/**

insert into ticket values(4239269912,12651,"SLEEPER","Poonam",35);

insert into ticket values(4239269913,12651,"SLEEPER","Purnima",35);

insert into ticket values(4239269911,12651,"2AC","Pratibha",35);



**/\* creating table ticket\_train \*/**

create table ticket\_train (

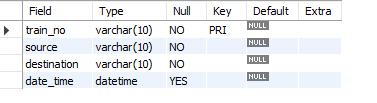
train\_no varchar(10) primary key,

source varchar(10) not null,

destination varchar (10) not null,

date\_time datetime

);

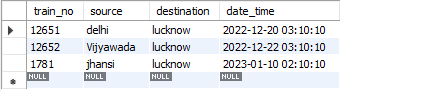


**/\*Inserting data into table \*/**

insert into ticket\_train values (12651,"delhi","lucknow","2022-12-20 03:10:10");

insert into ticket\_train values (12652,"Vijyawada","lucknow","2022-12-22 03:10:10");

insert into ticket\_train values (1781,"jhansi","lucknow","2023-01-10 02:10:10");



**/\* creating table payment \*/**

create table payment (

transcation\_id varchar (20),

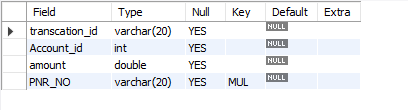
Account\_id int,

amount real,

PNR\_NO varchar (20),

foreign key (PNR\_NO) references ticket (PNR\_NO)

);

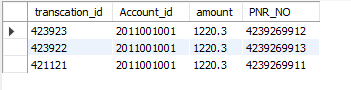


**/\*Inserting data into table \*/**

insert into payment values (423923,2011001001,1220.3,4239269912);

insert into payment values (423922,2011001001,1220.3,4239269913);

insert into payment values (421121,2011001001,1220.3,4239269911);



**/\* creating Train table \*/**

create table train (

train\_no varchar (20) primary key,

train\_name varchar (20) not null,

arrival\_time time,

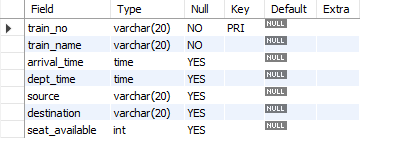
dept\_time time,

source varchar(20),

destination varchar(20),

seat\_available int

);

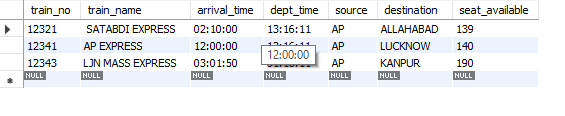


/\***Inserting values into train table \*/**

insert into train values (12341,"AP EXPRESS","12:00:00","12:16:11","AP","LUCKNOW",140);

insert into train values (12321," SATABDI EXPRESS","2:10:00","13:16:11","AP","ALLAHABAD",139);

insert into train values (12343,"LJN MASS EXPRESS","3:01:50","1:16:11","AP","KANPUR",190);

****

**/\*creating table user\_train \*/**

create table user\_train (

user\_id varchar (10),

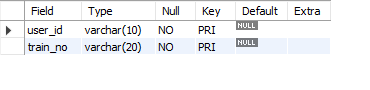
train\_no varchar (20),

foreign key(user\_id) references USER (user\_id),

foreign key(train\_no) references train(train\_no),

primary key (user\_id, train\_no)

);

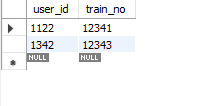


**/\* inserting values into tables \*/**

insert into user\_train values (1122,12341);

insert into user\_train values (1122,123421);

insert into user\_train values (1342,12343);



**/\* creating class table \*/**

create table class (

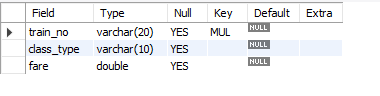
train\_no varchar (20),

class\_type varchar (10),

fare real,

foreign key(train\_no) references train(train\_no)

);

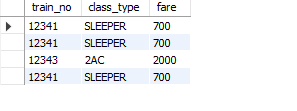


**/\*Inserting values into table class \*/**

insert into class values(12341,"SLEEPER",700);

insert into class values(12343,"2AC",2000);

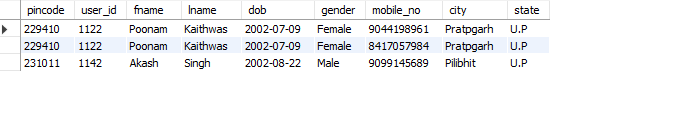
insert into class values(12341,"SLEEPER",700);



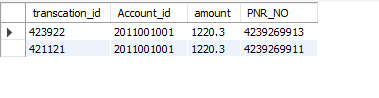
**SQL Queries**

**SQL Query:**

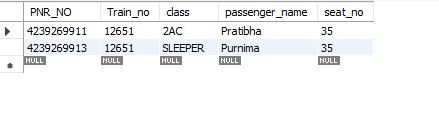
1.select \*from user natural join user\_contact natural join user\_pincode;



2.delete from payment where PNR\_NO="4239269912";



3. delete from ticket where PNR\_NO="4239269912";

. 

**Conclusion**

I haven’t found any anomalies while inserting, updating and deleting data. It passed all test cases.

**References**

1. [**https://www.slideshare.net/shashankkarnati/railway-management-system-database-mini-project**](https://www.slideshare.net/shashankkarnati/railway-management-system-database-mini-project)
2. [**https://www.youtube.com/watch?v=ma8V3QPnQ0E**](https://www.youtube.com/watch?v=ma8V3QPnQ0E)