SQL PROJECT

Railway Database Management System

Abstract

The Railway Database Management System (RDMS) is a comprehensive SQL-based project aimed at efficiently managing railway-related data and operations. The system is designed to provide a robust platform for storing, organizing, and retrieving information related to trains, stations, schedules, bookings, and other essential aspects of railway management.

RUPESH KUMAR

rupeshkumar2002.rk@gmail.com

NIT JAMSHEDPUR B-Tech (Hons) ME

Key Objective

- Data Storage and Organization: The RDMS will establish a structured database to store all relevant information, such as train details, station information, passenger records, bookings, and administrative data. It will ensure efficient data organization, reducing redundancy and improving data integrity.

- Train Management: The system will enable effective train management, including the ability to add new trains, modify existing train information, and track train schedules. It will allow for the storage of train-specific attributes like train number, name, type, capacity, and associated routes.

- The railway management 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. The aim of case study is to design and develop a database maintaining the records of different trains, stations, and passengers. The record of the train includes its number, name, days on which it is available etc.

- Before booking a ticket for a passenger, 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 is available. If yes, the ticket is booked with confirm status and corresponding ticket No is generated which is stored along with other details of the passenger.

- The ticket once booked can be cancelled at any time. For this, the passenger has to provide the ticket ID (the unique key). The ticket ID is searched and the corresponding record is deleted.

Content

Introduction

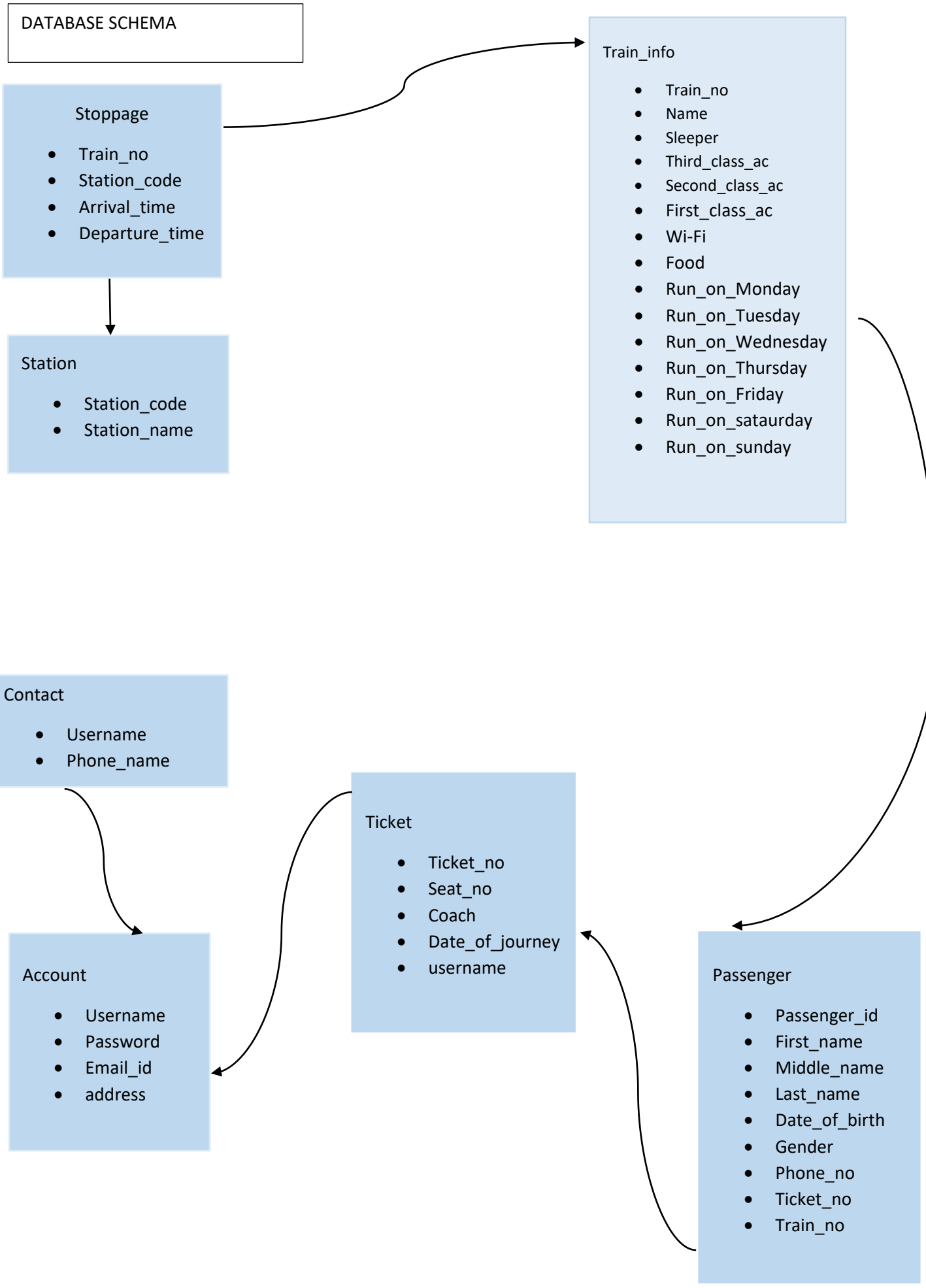
Here are the attributes of our data bases for railway management system.

```
Account
(
Username varchar(15) NOT NULL,
Password varchar(20) NOT NULL,
Email_Id varchar(35) NOT NULL,
Address varchar(50) DEFAULT
NULL,
PRIMARY KEY (Username)
)
Contact
(
Username varchar(15) NOT NULL DEFAULT '',
Phone_No char(10) NOT NULL DEFAULT '',
PRIMARY KEY (Username,Phone_No),
CONSTRAINT Contact_ibfk_1 FOREIGN KEY (Username) REFERENCES
Account (Username) ON DELETE CASCADE
)
Passenger
Passenger_Id int(11) NOT NULL AUTO_INCREMENT,
First_Name varchar(20) NOT NULL,
Last_Name varchar(20) NOT NULL,
Gender char(1) NOT NULL,
Phone_No char(10) DEFAULT NULL,
Ticket_No int(10) NOT NULL,
Age int(11) NOT NULL,
Class varchar(20) NOT NULL,
PRIMARY KEY (Passenger_Id),  KEY ticket_No (Ticket_No),
```

```
CONSTRAINT Passenger_ibfk_1 FOREIGN KEY (Ticket_No) REFERENCES
Ticket (Ticket_No) ON DELETE CASCADE
)
Station
(
Station_Code char(5) NOT NULL DEFAULT '',
Station_Name varchar(25) NOT NULL,
PRIMARY KEY (Station_Code)
)
Stoppage
Train_No int(6) NOT NULL DEFAULT '0',
Station_Code char(5) NOT NULL DEFAULT '',
Arrival_Time time DEFAULT NULL,
Departure_Time time DEFAULT NULL,
PRIMARY KEY (Train_No,station_Code),
KEY Station_Code (Station_Code),
CONSTRAINT Stoppage_ibfk_1 FOREIGN KEY (Train_No) REFERENCES Train_info
(Train_No) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT Stoppage_ibfk_2 FOREIGN KEY (Station_Code) REFERENCES Station
(Station_Code) ON DELETE CASCADE ON UPDATE CASCADE
)
Ticket
{ Ticket_No int(10) NOT NULL AUTO_INCREMENT,
Train_No int(6) NOT NULL,
Date_of_Journey date NOT NULL,
Username varchar(15) NOT NULL,
PRIMARY KEY (Ticket_No),
KEY Username (Username),
KEY Train_No (Train_No),
```

```
CONSTRAINT Ticket_ibfk_1 FOREIGN KEY (Username) REFERENCES
Account (Username) ON DELETE CASCADE,
CONSTRAINT Ticket_ibfk_2 FOREIGN KEY (Train_No) REFERENCES Train_info
(Train_No) ON UPDATE CASCADE
)
Train_info
(
Train_No int(6) NOT NULL DEFAULT
'0',
Name varchar(25) NOT NULL,
Sleeper int(4) NOT NULL,
First_Class_AC int(4) NOT
NULL,
Second_Class_AC int(4) NOT
NULL,
Third_Class_AC int(4) NOT
NULL,
Wifi char(1) NOT NULL,
Food char(1) NOT NULL,
Run_On_Sunday char(1) NOT NULL,
Run_On_Monday char(1) NOT NULL,
Run_On_Tuesday char(1) NOT NULL,
Run_On_Wednesday char(1) NOT NULL,
Run_On_Thursday char(1) NOT NULL,
Run_On_Friday char(1) NOT NULL,
Run_On_Saturday char(1) NOT NULL,
PRIMARY KEY (Train_No)
)
```

DATABASE SCHEMA



Normalization

- 1st NF: According to the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values. The above schema is in 1NF since all the attributes are atomic and not multivalued. Since a passenger could have multiple phone numbers, it would violate the 1NF rules. Hence we have created a separate table called contact to handle this.

- 2nd NF: A table follow the 2NF only if the following conditions hold:

  -Table is in 1NF (First normal form)
  -No non-prime column is dependent on the proper subset of any candidate key of table.

- THIRD NORMAL FORM: A table design is said to be in 3NF if both the following conditions hold:

  -Table must be in 2NF
  -Transitive functional dependency of non-prime attribute on any super key should be removed.

DDL

```
create database Railway_management_project;

use Railway_management_project;

create table Account(

username varchar(15) not null primary key,

password varchar(15) not null,

email_id varchar(15) not null,

address varchar(50) default null
```

```
);
ALTER TABLE account
MODIFY COLUMN password varchar(40);
ALTER TABLE account
MODIFY COLUMN email_id varchar(40);

select * from account;
describe account;

create table contact(
username  varchar(15) not null default '',
phone_no varchar(10)  not null default '',
primary key(username,phone_no),
constraint contact_ibfk_1 foreign key(username) references Account(username) on delete cascade
);
describe contact;

create table train_info(
train_no int(6) primary key not null default '0',
name varchar(25) not null,
sleeper int(4) not null,
first_class_ac int(4) not null,
second_class_ac int(4) not null,
third_class_ac int(4) not null,
food_service char(1) not null,
wifi_service char(1) not null,
run_on_monday char(1) not null,
run_on_tuesday char(1) not null,
run_on_wednesday char(1) not null,
```

```
run_on_thursday char(1) not null,
run_on_friday char(1) not null,
run_on_saturday char(1) not null,
run_on_sunday char(1) not null
);
describe train_info;

drop table ticket;
CREATE TABLE Ticket (
Ticket_No int(10) NOT NULL AUTO_INCREMENT,
Train_No int(6) NOT NULL,
Date_of_Journey date NOT NULL,
Username varchar(15) NOT NULL,
KEY Username (Username),
KEY Train_No (Train_No),
PRIMARY KEY (Ticket_No),
CONSTRAINT Ticket_ibfk_1 FOREIGN KEY (Username) REFERENCES Account
(Username) ON DELETE CASCADE,
CONSTRAINT Ticket_ibfk_2 FOREIGN KEY (Train_No) REFERENCES Train_info
(Train_No) ON UPDATE CASCADE
);
describe ticket;

CREATE TABLE Passenger (
  Passenger_Id int(11) NOT NULL AUTO_INCREMENT,
  First_Name varchar(20) NOT NULL,
  Last_Name varchar(20) NOT NULL,
  Gender char(1) NOT NULL,
  Phone_No char(10) DEFAULT NULL,
```

```
  Ticket_No int(10) NOT NULL,
  Age int(11) NOT NULL,
  Class varchar(20) NOT NULL,
  PRIMARY KEY (Passenger_Id),
  KEY Ticket_No (Ticket_No),
  CONSTRAINT Passenger_ibfk_1 FOREIGN KEY (Ticket_No) REFERENCES Ticket (Ticket_No) ON DELETE CASCADE
);
describe passenger;

 CREATE TABLE Station(
  Station_Code char(5) NOT NULL DEFAULT '',
  Station_Name varchar(25) NOT NULL,
  PRIMARY KEY (Station_Code)
);

describe station;

CREATE TABLE Stoppage(
  Train_No int(6) NOT NULL DEFAULT '0',
  Station_Code char(5) NOT NULL DEFAULT '',
  Arrival_Time time DEFAULT NULL,
  Departure_Time time DEFAULT NULL,
  PRIMARY KEY (Train_No,Station_Code),
  KEY Station_Code (Station_Code),
  CONSTRAINT Stoppage_ibfk_1 FOREIGN KEY (Train_No) REFERENCES Train_info(Train_No) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT Stoppage_ibfk_2 FOREIGN KEY (Station_Code) REFERENCES Station (Station_Code) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
describe stoppage;

alter table Stoppage ADD CHECK (EXTRACT(HOUR FROM Arrival_Time) <24 AND EXTRACT(HOUR FROM
Departure_Time) <24);
```

Trigger define

```
delimiter //

create trigger cancellation

 before delete on ticket

 for each row

BEGIN

  set @trainno=old.train_no;

  set @ticketno=old.ticket_no;

  SET @class = (SELECT p.class

            FROM PASSENGER p

            WHERE p.ticket_no = @ticketno);

  if @class='first class ac' then

    UPDATE Train set first_class_ac = first_class_ac+1 WHERE Train_No = @trainno ;

  elseif @class='sleeper' then

    UPDATE Train set sleeper = sleeper+1 WHERE Train_No = @trainno ;

  elseif @class='second class ac' then

    UPDATE Train set second_class_ac = second_class_ac+1 WHERE Train_No = @trainno ;

  elseif @class='third class ac' then

    UPDATE Train set third_class_ac = third_class_ac+1 WHERE Train_No = @trainno ;

  end if;

END//

delimiter ;
```

Inserting data in to database

```
/* insert data in to the tables */

INSERT INTO `Account` VALUES ('ajitesh','eba094d4d15bc478cdc9','ajitesh@pes.edu','Old airport
road,bangalore'),('anantdadu','proxyman','dadu@cmu.ac.in','New
York'),('atishay','qwerty','Atishay.jain.cse14@gnail.com','Rangmahal Mall,
Panna'),('divyam310','goyal1002','divyam.goyal@gmail.com','Kota,
Rajasthan'),('goku446','dejavu','goku@gmail.com','Kota,
Rajasthan'),('prateek1996','ronaldoisgreat','prateek@gmail.com','New
Delhi'),('user101','eba094d4d15bc478cdc9','atishay.jain.cse14@iitbhu.ac.in','Madhya Pradesh');

INSERT INTO `Contact` VALUES
('anantdadu','8899887766'),('anantdadu','9876543210'),('atishay','7071475390'),('atishay','8009224040'
),('ajitesh','7411452250'),('ajitesh','9650367698'),('ajitesh','9968254144'),('divyam310','8989786765'),('g
oku446','9232453425'),('goku446','9989786756'),('prateek1996','9898342565'),('user101','7071475390')
;

select * from account;

select * from contact;

INSERT INTO Train_info VALUES (12559,'SHIV GANGA
EXP',479,47,96,192,'N','Y','Y','Y','Y','Y','Y','Y','Y'),(12560,'SHIV GANGA
EXP',480,43,96,192,'N','Y','Y','Y','Y','Y','Y','Y','Y'),(12581,'BLR NDLS S F
EX',432,48,80,144,'N','N','Y','Y','Y','Y','Y','Y','Y'),(12582,'BLR NDLS S F
EX',432,48,80,144,'N','N','Y','Y','Y','Y','Y','Y','Y');

select * from train_info;

INSERT INTO Station VALUES ('ALD','ALLAHABAD JUNCTION'),('CNB','KANPUR
CENTRAL'),('GYN','GYANPUR ROAD'),('GZB','GHAZIABAD JUNCTION'),('BLR','BANGALORE'),('NDLS','NEW
DELHI');

select * from station;

INSERT INTO Stoppage VALUES
(12559,'ALD','22:05:00','22:30:00'),(12559,'CNB','01:30:00','01:38:00'),(12559,'BLR','19:20:00','19:30:00')
,(12559,'NDLS','08:10:00',NULL),(12560,'ALD','03:45:00','04:10:00'),(12560,'CNB','01:00:00','01:05:00'),(1
2560,'BLR','07:00:00',NULL),(12560,'NDLS','18:35:00','18:55:00'),(12581,'ALD','01:20:00','01:45:00'),(125
81,'CNB','04:15:00','04:20:00'),(12581,'GYN','23:31:00','23:33:00'),(12581,'GZB','11:30:00','11:32:00'),(12
581,'BLR','22:20:00','22:30:00'),(12581,'NDLS','12:20:00',NULL),(12582,'ALD','07:45:00','08:15:00'),(1258
2,'CNB','04:55:00','05:00:00'),(12582,'GYN','09:21:00','09:23:00'),(12582,'GZB','23:03:00','23:05:00'),(125
82,'BLR','11:20:00',NULL),(12582,'NDLS','22:15:00','22:25:00');

select * from stoppage;
```

```
select * from passenger;

select * from ticket;
```

Booking ticket by inserting account, ticket, and passenger data

```
/* lets book a ticket from account details below */

INSERT INTO Account VALUES ('admin','admin@1234','ajitesh@pes.edu','Old airport road,bangalore');


/* now find a train between to station */

select a.Train_No from Stoppage as a join Stoppage as b on a.Train_No = b.Train_No

where a.Station_Code = "BLR" and b.Station_Code = "NDLS";


/* as we see there are four train available between above destination */

insert into ticket values ('1','12559','2020-04-27','admin');


/* from above query train ticket is booked on a particular date ...below query is about passenger details
*/

INSERT INTO Passenger values ('1','admin','','F','233','1','20','first class ac');


/* Now if above ticket is cancelled, trigger is invoked and it will delete the data from parent as well as
child table and also  add the cancelled ticket to their class */

delete from ticket where ticket_no=1;
```

Conclusion: The above railway management system can successfully give insight information on any train, trains running between two stations, book tickets and cancel tickets. This system could be used for official train booking.However several other features could be added like booking meals on trains etc. Also payment gateways have to be implemented to make sure the transactions happen securely