**Database Management System**

**INFO – 6210**

**FINAL PROJECT**

**ONLINE RAILWAY RESERVATION SYSTEM**

Sudhanshu Garg

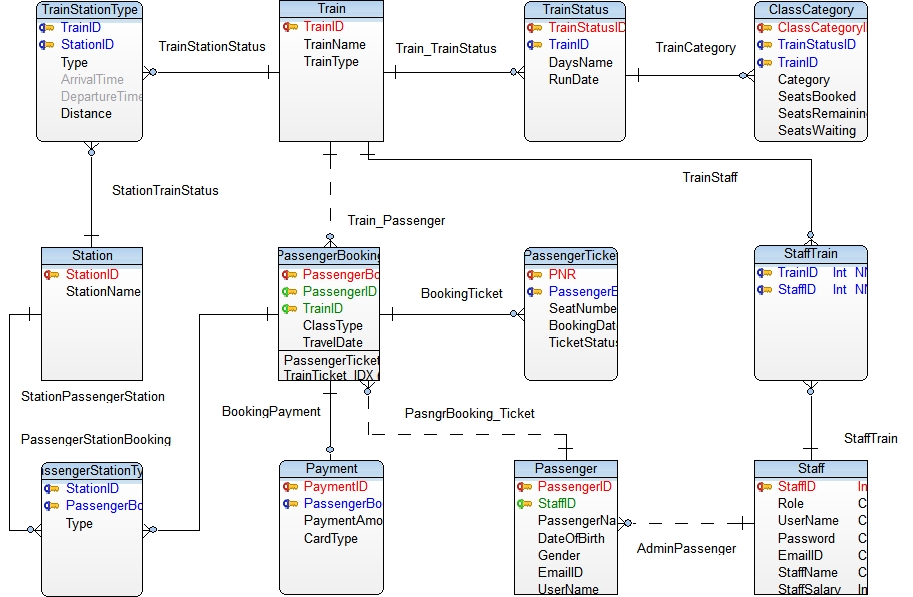
NEU ID: 001267033

PROBLEM STATEMENT:

MySQL Database

This system is basically concerned with reservation of railway tickets to the passenger. It has to be developed for automating the manual railway reservation system. Any person who wants to book a railway ticket can be benefited by this system as it is more efficient way of booking a ticket online along with keeping the check on seat availability in trains. A passenger should be able to reserve a seat in the train specified by him only if available. For this he has to fill a reservation details about his journey.  If the seat is available, then a ticket is generated with an allotted seat to that passenger and his ticket status. The passenger is then asked to pay the ticket price. All the users should be able to see the information about the reservation status & train schedule, train name, train number etc. There are different railway staffs, TT who are allotted to different trains, and all the details of the staff member could be fetched. A passenger would have a user account controlled by admin which has all the passenger details.

ER DIAGRAMS:



Definitions and source codes of Stored Procedures, Triggers:

**Stored Procedures:**

1.Back Up

Definition:

This will create a backup of my online railway reservation database every week using an event.

Code:

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_full\_bkp`()

block1: BEGIN

DECLARE tab\_name char(50);

DECLARE q varchar(1500);

DECLARE done INTEGER DEFAULT 0;

DECLARE cursorBackupTable CURSOR FOR SELECT TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES WHERE table\_schema='railway' and table\_type='BASE TABLE';

DROP DATABASE IF EXISTS railway\_backup;

CREATE DATABASE railway\_backup;

open cursorBackupTable;

block2: begin

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

cur\_loop:Loop

FETCH cursorBackupTable into tab\_name;

IF done= 1 THEN LEAVE cur\_loop;

END IF;

SET @q=CONCAT('DROP TABLE IF EXISTS railway\_backup.',tab\_name);

PREPARE stmt FROM @q;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

SET @q=CONCAT('CREATE TABLE railway\_backup.',tab\_name,' AS SELECT \* FROM railway.',tab\_name,' WHERE 1=1');

PREPARE stmt FROM @q;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

END LOOP cur\_loop;

END block2;

close cursorBackupTable;

END block1

2. Train Details

Definition:

This will give us all the details of a given Train.

Code:

CREATE DEFINER=`root`@`localhost` PROCEDURE `TrainDetail`(

IN TrainID INTEGER

)

block1:BEGIN

select t.TrainName, t.TrainType, s.StationName, tst.Type, tst.ArrivalTime, tst.DepartureTime

from train as t

inner join trainstationtype as tst

on t.TrainID = tst.TrainID

inner join station as s

on t.TrainID = tst.TrainID and tst.StationID = s.StationID;

END block1

3. Seat Details for a Train

Definition:

This will give us all the details regarding the Seat Availability of a Category in a Train running on a specified Date.

Code:

CREATE DEFINER=`root`@`localhost` PROCEDURE `SeatDetailsForATrain`(TrainId Int, RunDate Date , Category Char(30))

BEGIN

select t.TrainName,cc.Category,ts.RunDate,cc.SeatsRemaining, cc.SeatsWaiting

from train as t

inner join TrainStatus as ts

on t.TrainID = ts.TrainID

inner join ClassCategory cc

on ts.TrainStatusID = cc.TrainStatusID

where ts.TrainID = TrainID and ts.RunDate = RunDate and cc.Category= Category;

END

4. Trains passing through a Station

Definition:

This will give us all the trains passing through a Station.

Code:

CREATE DEFINER=`root`@`localhost` PROCEDURE `TrainStation`(

IN StationId INTEGER

)

block1:BEGIN

select s.StationName, t.TrainName, t.TrainType, tst.Type, tst.ArrivalTime, tst.DepartureTime

from Station as s

inner join trainstationtype as tst

on s.StationID = tst.StationID

inner join train as t

on tst.TrainID = t.TrainID

where tst.StationID = StationId;

END block1

**Functions**:

1.Ticket Price

Definition:

For a particular ticket, it calculates the amount a customer needs to pay.

Code:

CREATE DEFINER=`root`@`localhost` FUNCTION `Ticket\_Price`(PNR INT) RETURNS decimal(10,2)

DETERMINISTIC

BEGIN

DECLARE PassengerBookingId Int(11);

DECLARE TrainID Int(11);

Declare SourceStationId Int(11);

Declare DestinationStationId Int(11);

Declare SourceDistance Int(11);

Declare DestinationDistance Int(11);

Declare TicketPrice decimal(10,2);

Select pt.PassengerBookingId into PassengerBookingId

from PassengerTicket as pt

where pt.PNR = PNR;

Select pb.TrainId into TrainID

from PassengerBooking as pb

where pb.PassengerBookingId = PassengerBookingId;

Select pst.StationId into SourceStationId

from PassengerStationType as pst

where pst.PassengerBookingId = PassengerBookingId and pst.type = 'Source';

Select pst.StationId into DestinationStationId

from PassengerStationType as pst

where pst.PassengerBookingId = PassengerBookingId and pst.type = 'Destination';

Select tst.Distance into SourceDistance

from TrainStationType as tst

where tst.TrainId = TrainID and tst.StationId = SourceStationId;

Select tst.Distance into DestinationDistance

from TrainStationType as tst

where tst.TrainId = TrainID and tst.StationId = DestinationStationId;

Set TicketPrice = (DestinationDistance - SourceDistance) \* 1.2;

return TicketPrice;

END

2. Ticket Status

Definition:

For a PNR, this will provide us the current ticket status.

Code:

CREATE DEFINER=`root`@`localhost` FUNCTION `Ticket\_Status`(PNR INT) RETURNS varchar(30) CHARSET utf8

DETERMINISTIC

BEGIN

DECLARE A varchar(30);

SELECT

t.ticketstatus

INTO A FROM

passengerticket AS t

WHERE

t.PNR = PNR;

RETURN A;

END

**Triggers:**

1.Ticket\_Insert\_Check

Definition:

Before generating a ticket, it will check the availability.

Code:

CREATE DEFINER=`root`@`localhost` TRIGGER ticket\_insert\_check

Before INSERT ON passengerbooking

FOR EACH ROW

Begin

Declare varTrainID Int;

Declare varTravelDate Date;

Declare varTrainStatusID Int;

Declare varClassCategoryID Int;

Declare varCategory char(20);

Declare varCountRemainingSeats Int;

Declare varCountSeatsWaiting Int;

Set varTrainID = NEW.TrainID;

Set varTravelDate = NEW.TravelDate;

Set varCategory = NEW.ClassType;

Select TrainStatusID into varTrainStatusID

from TrainStatus as ts

where ts.TrainID = varTrainID and ts.RunDate = varTravelDate;

select ClassCategoryID into varClassCategoryID

from ClassCategory as cc1

where cc1.TrainID = varTrainID and cc1.TrainStatusID = varTrainStatusID and cc1.Category = varCategory;

select cc2.SeatsRemaining into varCountRemainingSeats

from ClassCategory as cc2

where cc2.ClassCategoryID = varClassCategoryID;

select cc3.SeatsWaiting into varCountSeatsWaiting

from ClassCategory as cc3

where cc3.ClassCategoryID = varClassCategoryID;

if(varCountRemainingSeats > 0) then

update ClassCategory as cc4

set cc4.SeatsRemaining = cc4.SeatsRemaining - 1, cc4.SeatsBooked = cc4.SeatsBooked + 1

where cc4.ClassCategoryID = varClassCategoryID;

end if;

if(varCountRemainingSeats = 0) then

if(varCountSeatsWaiting < 10) then

update ClassCategory as cc5

set cc5.SeatsWaiting = cc5.SeatsWaiting + 1

where cc5.ClassCategoryID = varClassCategoryID;

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Waiting Full!!!!';

end if;

end if;

End

2. Ticket\_Generate\_Check

Definition:

After inserting the ticket, it will assign a seat number and ticket status in another table.

Code:

CREATE DEFINER=`root`@`localhost` TRIGGER ticket\_generate\_check

After INSERT ON passengerbooking

FOR EACH ROW

Begin

Declare varPassengerBookingID Int;

Declare varTrainID Int;

Declare varTravelDate Date;

Declare varTrainStatusID Int;

Declare varClassCategoryID Int;

Declare varCategory char(20);

Declare varCountRemainingSeats Int;

Declare varCountSeatsWaiting Int;

Declare varSeatNumber Int;

Declare varCountPassengerTicket Int;

Set varPassengerBookingID = NEW.PassengerBookingID;

Set varTrainID = NEW.TrainID;

Set varTravelDate = NEW.TravelDate;

Set varCategory = NEW.ClassType;

Select TrainStatusID into varTrainStatusID

from TrainStatus as ts

where ts.TrainID = varTrainID and ts.RunDate = varTravelDate;

select ClassCategoryID into varClassCategoryID

from ClassCategory as cc1

where cc1.TrainID = varTrainID and cc1.TrainStatusID = varTrainStatusID and cc1.Category = varCategory;

select cc2.SeatsRemaining into varCountRemainingSeats

from ClassCategory as cc2

where cc2.ClassCategoryID = varClassCategoryID;

select cc3.SeatsWaiting into varCountSeatsWaiting

from ClassCategory as cc3

where cc3.ClassCategoryID = varClassCategoryID;

Select Max(SeatNumber) into varSeatNumber

from PassengerTicket;

Select count(\*) into varCountPassengerTicket

from PassengerTicket;

if(varCountRemainingSeats > 0) then

if(varCountPassengerTicket > 0) then

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, varSeatNumber+1, sysdate(), 'Confirmed');

else

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, 1, sysdate(), 'Confirmed');

end if;

end if;

if(varCountRemainingSeats = 0) then

if(varCountSeatsWaiting < 10) then

if(varCountPassengerTicket > 0) then

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, 0, sysdate(), 'Waiting');

else

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, 0, sysdate(), 'Waiting');

end if;

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Waiting Full!!!!';

end if;

end if;

End

3. Station\_Insert\_Check

Definition:

This will check before inserting the passenger’s source and destination station whether it is a station for a particular train.

Code:

Delimiter $$

CREATE TRIGGER station\_insert\_check

Before INSERT ON PassengerStationType

FOR EACH ROW

block1: BEGIN

DECLARE finished1 INTEGER default 0;

DECLARE counter INTEGER default 0;

DECLARE tempPasngrStationNumber INTEGER;

DECLARE tempTrainStationNumber INTEGER;

Set tempPasngrStationNumber = (SELECT NEW.StationID from PassengerStationType where PassengerBookingID = NEW.PassengerBookingID);

If(!(tempPasngrStationNumber IN (SELECT StationID FROM TrainStationType where TrainId =

(Select TrainId from PassengerBooking where PassengerBookingId = NEW.PassengerBookingId)))) then

SIGNAL SQLSTATE '45000';

end if;

END block1;

$$

4. Staff\_Insert\_Check

Definition:

This will insure that only TT’s in the staff are allotted a train.

Code:

Delimiter $$

CREATE TRIGGER staff\_insert\_check

Before INSERT ON StaffTrain

FOR EACH ROW

block1: BEGIN

DECLARE varStaffID INTEGER;

DECLARE varRole Varchar(20);

DECLARE MESSAGE\_TEXT Varchar(50);

Set varStaffID = NEW.StaffID;

Select Role into varRole from Staff where StaffId = varStaffID;

If varRole != 'TT' then

SIGNAL SQLSTATE '45000';

SET MESSAGE\_TEXT = 'You are not authorized!!!!';

end if;

END block1;

$$

**Views:**

1.Staff Details

Definition:

This creates a view to fetch all the employee Details.

Code:

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `staffdetail` AS

SELECT

`st`.`StaffID` AS `StaffID`,

`st`.`Role` AS `Role`,

`st`.`UserName` AS `UserName`,

`st`.`Password` AS `Password`,

`st`.`EmailID` AS `EmailID`,

`st`.`StaffName` AS `StaffName`,

`st`.`StaffSalary` AS `StaffSalary`

FROM

`staff` `st`

WHERE

(`st`.`Role` <> 'Admin')

2. Waiting Passengers

Definition:

This will create a view of all the passengers which have a waiting ticket status.

Code:

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `waitingpassengers` AS

SELECT

`pt`.`PNR` AS `PNR`,

`pt`.`PassengerBookingID` AS `PassengerBookingID`,

`pt`.`SeatNumber` AS `SeatNumber`,

`pt`.`BookingDate` AS `BookingDate`,

`pt`.`TicketStatus` AS `TicketStatus`

FROM

`passengerticket` `pt`

WHERE

(`pt`.`TicketStatus` = 'Waiting')

**Question’s Answered:**

Question1. Display a Train Details given a Train Id?

Answer1. CALL `railway`.`TrainDetail`(<{IN TrainID INTEGER}>);

Question2. If a user enters a PNR, show the user’s ticket status.

Answer2. Select Ticket\_Status(<{PNR INT}>) as TicketStatus;

Question3. Display Seat Details for a Category for a given Train Id running on a specified Date.

Answer3. CALL `railway`.`SeatDetailsForATrain`(<{TrainId Int}>, <{RunDate Date}>, <{Category Char(30)}>);

Question4. Calculate a Payment amount of a Passenger?

Answer4. Select Ticket\_Price(<{PNR INT}>) as TicketPrice;

Question5. Show all the Staff’s except Admin ?

Answer5. Select \* from staffdetail; (view)

Question6. Display all the trains that passes through a give Station ?

Answer6. CALL `railway`.`TrainStation`(<{IN StationId INTEGER}>);

Question7. Show all the passengers with waiting status?

Answer7. Select \* from waitingpassengers; (view)

Question8. Display details of a passenger ?

Anwer8. Select p.PassengerName, pt.PNR, pt.SeatNumber, pt.TicketStatus, pb.TravelDate, pst.Type

from passenger as p

inner join passengerbooking as pb

on p.PassengerID = pb.PassengerID

inner join passengerticket as pt

on pb.PassengerBookingID = pt.PassengerBookingID

inner join passengerstationtype as pst

on pb.PassengerBookingID = pst.PassengerBookingID;

**APPENDIX**

CREATE DATABASE IF NOT EXISTS `railway` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `railway`;

-- MySQL dump 10.13 Distrib 5.7.12, for Win64 (x86\_64)

--

-- Host: localhost Database: railway

-- ------------------------------------------------------

-- Server version 5.7.15-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `classcategory`

--

DROP TABLE IF EXISTS `classcategory`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `classcategory` (

`ClassCategoryID` int(11) NOT NULL AUTO\_INCREMENT,

`TrainStatusID` int(11) NOT NULL,

`TrainID` int(11) NOT NULL,

`Category` char(30) NOT NULL,

`SeatsBooked` int(11) NOT NULL,

`SeatsRemaining` int(11) NOT NULL,

`SeatsWaiting` int(11) NOT NULL,

PRIMARY KEY (`ClassCategoryID`,`TrainStatusID`,`TrainID`),

KEY `TrainCategory` (`TrainStatusID`,`TrainID`),

CONSTRAINT `TrainCategory` FOREIGN KEY (`TrainStatusID`, `TrainID`) REFERENCES `trainstatus` (`TrainStatusID`, `TrainID`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `classcategory`

--

LOCK TABLES `classcategory` WRITE;

/\*!40000 ALTER TABLE `classcategory` DISABLE KEYS \*/;

INSERT INTO `classcategory` VALUES (1,1,1,'AC',24,17,0),(2,1,1,'Sleeper',40,0,4),(3,3,2,'AC',46,8,0),(4,4,3,'AC',32,0,3),(5,5,4,'AC',27,0,10),(6,6,5,'AC',18,3,0),(7,2,1,'AC',45,1,0);

/\*!40000 ALTER TABLE `classcategory` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `passenger`

--

DROP TABLE IF EXISTS `passenger`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `passenger` (

`PassengerID` int(11) NOT NULL AUTO\_INCREMENT,

`StaffID` int(11) NOT NULL,

`PassengerName` char(50) NOT NULL,

`DateOfBirth` date NOT NULL,

`Gender` char(1) NOT NULL,

`EmailID` char(50) NOT NULL,

`UserName` char(30) NOT NULL,

`Password` char(50) NOT NULL,

PRIMARY KEY (`PassengerID`),

KEY `AdminPassenger\_IDX` (`StaffID`),

CONSTRAINT `AdminPassenger` FOREIGN KEY (`StaffID`) REFERENCES `staff` (`StaffID`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `passenger`

--

LOCK TABLES `passenger` WRITE;

/\*!40000 ALTER TABLE `passenger` DISABLE KEYS \*/;

INSERT INTO `passenger` VALUES (1,1,'p1','1991-01-30','M','p1@gmail.com','p1','p1'),(2,1,'p2','1991-01-03','F','p2@gmail.com','p2','p2'),(3,1,'p3','1991-02-02','M','p3@gmail.com','p3','p3'),(4,1,'p4','1991-01-30','F','p4@gmail.com','p4','p4'),(5,1,'p5','1991-05-30','M','p5@gmail.com','p5','p5'),(6,1,'p6','1991-10-30','F','p6@gmail.com','p6','p6'),(7,1,'p7','1991-01-30','M','p7@gmail.com','p7','p7'),(8,1,'p8','1991-01-30','F','p8@gmail.com','p8','p8'),(9,1,'p9','1991-07-30','M','p9@gmail.com','p9','p9'),(10,1,'p10','1991-06-30','F','p10@gmail.com','p10','p10');

/\*!40000 ALTER TABLE `passenger` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `passengerbooking`

--

DROP TABLE IF EXISTS `passengerbooking`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `passengerbooking` (

`PassengerBookingID` int(11) NOT NULL AUTO\_INCREMENT,

`PassengerID` int(11) NOT NULL,

`TrainID` int(11) NOT NULL,

`ClassType` char(20) NOT NULL,

`TravelDate` date NOT NULL,

PRIMARY KEY (`PassengerBookingID`),

KEY `PassengerTicket\_IDX` (`PassengerID`),

KEY `TrainTicket\_IDX` (`TrainID`),

CONSTRAINT `PasngrBooking\_Ticket` FOREIGN KEY (`PassengerID`) REFERENCES `passenger` (`PassengerID`),

CONSTRAINT `Train\_Passenger` FOREIGN KEY (`TrainID`) REFERENCES `train` (`TrainID`)

) ENGINE=InnoDB AUTO\_INCREMENT=13 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `passengerbooking`

--

LOCK TABLES `passengerbooking` WRITE;

/\*!40000 ALTER TABLE `passengerbooking` DISABLE KEYS \*/;

INSERT INTO `passengerbooking` VALUES (1,1,1,'AC','2016-12-12'),(2,2,1,'Sleeper','2016-12-13'),(3,1,2,'AC','2016-12-14'),(4,3,3,'AC','2016-12-15'),(5,4,5,'AC','2016-12-16'),(6,5,5,'AC','2016-12-17'),(7,6,1,'AC','2016-12-12'),(8,7,2,'AC','2016-12-14'),(9,8,3,'AC','2016-12-15'),(10,9,3,'AC','2016-12-16'),(11,10,5,'AC','2016-12-17');

/\*!40000 ALTER TABLE `passengerbooking` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER ticket\_insert\_check

Before INSERT ON passengerbooking

FOR EACH ROW

Begin

Declare varTrainID Int;

Declare varTravelDate Date;

Declare varTrainStatusID Int;

Declare varClassCategoryID Int;

Declare varCategory char(20);

Declare varCountRemainingSeats Int;

Declare varCountSeatsWaiting Int;

Set varTrainID = NEW.TrainID;

Set varTravelDate = NEW.TravelDate;

Set varCategory = NEW.ClassType;

Select TrainStatusID into varTrainStatusID

from TrainStatus as ts

where ts.TrainID = varTrainID and ts.RunDate = varTravelDate;

select ClassCategoryID into varClassCategoryID

from ClassCategory as cc1

where cc1.TrainID = varTrainID and cc1.TrainStatusID = varTrainStatusID and cc1.Category = varCategory;

select cc2.SeatsRemaining into varCountRemainingSeats

from ClassCategory as cc2

where cc2.ClassCategoryID = varClassCategoryID;

select cc3.SeatsWaiting into varCountSeatsWaiting

from ClassCategory as cc3

where cc3.ClassCategoryID = varClassCategoryID;

if(varCountRemainingSeats > 0) then

update ClassCategory as cc4

set cc4.SeatsRemaining = cc4.SeatsRemaining - 1, cc4.SeatsBooked = cc4.SeatsBooked + 1

where cc4.ClassCategoryID = varClassCategoryID;

end if;

if(varCountRemainingSeats = 0) then

if(varCountSeatsWaiting < 10) then

update ClassCategory as cc5

set cc5.SeatsWaiting = cc5.SeatsWaiting + 1

where cc5.ClassCategoryID = varClassCategoryID;

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Waiting Full!!!!';

end if;

end if;

End \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER ticket\_generate\_check

After INSERT ON passengerbooking

FOR EACH ROW

Begin

Declare varPassengerBookingID Int;

Declare varTrainID Int;

Declare varTravelDate Date;

Declare varTrainStatusID Int;

Declare varClassCategoryID Int;

Declare varCategory char(20);

Declare varCountRemainingSeats Int;

Declare varCountSeatsWaiting Int;

Declare varSeatNumber Int;

Declare varCountPassengerTicket Int;

Set varPassengerBookingID = NEW.PassengerBookingID;

Set varTrainID = NEW.TrainID;

Set varTravelDate = NEW.TravelDate;

Set varCategory = NEW.ClassType;

Select TrainStatusID into varTrainStatusID

from TrainStatus as ts

where ts.TrainID = varTrainID and ts.RunDate = varTravelDate;

select ClassCategoryID into varClassCategoryID

from ClassCategory as cc1

where cc1.TrainID = varTrainID and cc1.TrainStatusID = varTrainStatusID and cc1.Category = varCategory;

select cc2.SeatsRemaining into varCountRemainingSeats

from ClassCategory as cc2

where cc2.ClassCategoryID = varClassCategoryID;

select cc3.SeatsWaiting into varCountSeatsWaiting

from ClassCategory as cc3

where cc3.ClassCategoryID = varClassCategoryID;

Select Max(SeatNumber) into varSeatNumber

from PassengerTicket;

Select count(\*) into varCountPassengerTicket

from PassengerTicket;

if(varCountRemainingSeats > 0) then

if(varCountPassengerTicket > 0) then

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, varSeatNumber+1, sysdate(), 'Confirmed');

else

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, 1, sysdate(), 'Confirmed');

end if;

end if;

if(varCountRemainingSeats = 0) then

if(varCountSeatsWaiting < 10) then

if(varCountPassengerTicket > 0) then

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, 0, sysdate(), 'Waiting');

else

insert into PassengerTicket(PassengerBookingID, SeatNumber, BookingDate, TicketStatus)

values (varPassengerBookingID, 0, sysdate(), 'Waiting');

end if;

else

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Waiting Full!!!!';

end if;

end if;

End \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Table structure for table `passengerstationtype`

--

DROP TABLE IF EXISTS `passengerstationtype`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `passengerstationtype` (

`StationID` int(11) NOT NULL,

`PassengerBookingID` int(11) NOT NULL,

`Type` char(50) NOT NULL,

PRIMARY KEY (`StationID`,`PassengerBookingID`),

KEY `PassengerStationBooking` (`PassengerBookingID`),

CONSTRAINT `PassengerStationBooking` FOREIGN KEY (`PassengerBookingID`) REFERENCES `passengerbooking` (`PassengerBookingID`),

CONSTRAINT `StationPassengerStation` FOREIGN KEY (`StationID`) REFERENCES `station` (`StationID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `passengerstationtype`

--

LOCK TABLES `passengerstationtype` WRITE;

/\*!40000 ALTER TABLE `passengerstationtype` DISABLE KEYS \*/;

INSERT INTO `passengerstationtype` VALUES (1,1,'Source'),(2,2,'Destination'),(2,7,'Source'),(3,7,'Destination'),(4,3,'Source'),(4,8,'Source'),(5,3,'Destination'),(5,5,'Source'),(5,6,'Source'),(5,8,'Destination'),(5,11,'Source'),(6,4,'Source'),(6,9,'Source'),(6,10,'Source'),(7,4,'Destination'),(7,9,'Destination'),(7,10,'Destination'),(10,5,'Destination'),(10,6,'Destination'),(10,11,'Destination');

/\*!40000 ALTER TABLE `passengerstationtype` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER station\_insert\_check

Before INSERT ON PassengerStationType

FOR EACH ROW

block1: BEGIN

DECLARE finished1 INTEGER default 0;

DECLARE counter INTEGER default 0;

DECLARE tempPasngrStationNumber INTEGER;

DECLARE tempTrainStationNumber INTEGER;

Set tempPasngrStationNumber = (SELECT NEW.StationID from PassengerStationType where PassengerBookingID = NEW.PassengerBookingID);

If(!(tempPasngrStationNumber IN (SELECT StationID FROM TrainStationType where TrainId =

(Select TrainId from PassengerBooking where PassengerBookingId = NEW.PassengerBookingId)))) then

SIGNAL SQLSTATE '45000';

end if;

END block1 \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Table structure for table `passengerticket`

--

DROP TABLE IF EXISTS `passengerticket`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `passengerticket` (

`PNR` int(11) NOT NULL AUTO\_INCREMENT,

`PassengerBookingID` int(11) NOT NULL,

`SeatNumber` int(10) NOT NULL,

`BookingDate` date NOT NULL,

`TicketStatus` char(20) NOT NULL,

PRIMARY KEY (`PNR`,`PassengerBookingID`),

KEY `BookingTicket` (`PassengerBookingID`),

CONSTRAINT `BookingTicket` FOREIGN KEY (`PassengerBookingID`) REFERENCES `passengerbooking` (`PassengerBookingID`)

) ENGINE=InnoDB AUTO\_INCREMENT=10028 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `passengerticket`

--

LOCK TABLES `passengerticket` WRITE;

/\*!40000 ALTER TABLE `passengerticket` DISABLE KEYS \*/;

INSERT INTO `passengerticket` VALUES (10020,1,1,'2016-12-12','Confirmed'),(10021,3,2,'2016-12-12','Confirmed'),(10022,4,0,'2016-12-12','Waiting'),(10023,6,3,'2016-12-12','Confirmed'),(10024,7,4,'2016-12-12','Confirmed'),(10025,8,5,'2016-12-12','Confirmed'),(10026,9,0,'2016-12-12','Waiting'),(10027,11,6,'2016-12-12','Confirmed');

/\*!40000 ALTER TABLE `passengerticket` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `payment`

--

DROP TABLE IF EXISTS `payment`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `payment` (

`PaymentID` int(11) NOT NULL AUTO\_INCREMENT,

`PassengerBookingID` int(11) NOT NULL,

`PaymentAmount` decimal(10,2) NOT NULL,

`CardType` char(20) NOT NULL,

PRIMARY KEY (`PaymentID`,`PassengerBookingID`),

KEY `BookingPayment` (`PassengerBookingID`),

CONSTRAINT `BookingPayment` FOREIGN KEY (`PassengerBookingID`) REFERENCES `passengerbooking` (`PassengerBookingID`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `payment`

--

LOCK TABLES `payment` WRITE;

/\*!40000 ALTER TABLE `payment` DISABLE KEYS \*/;

INSERT INTO `payment` VALUES (1,3,720.00,'credit'),(2,4,600.00,'credit'),(3,5,182.40,'credit'),(4,6,60.00,'credit'),(5,7,720.00,'credit'),(6,8,600.00,'credit'),(7,9,182.40,'credit');

/\*!40000 ALTER TABLE `payment` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `staff`

--

DROP TABLE IF EXISTS `staff`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `staff` (

`StaffID` int(11) NOT NULL AUTO\_INCREMENT,

`Role` char(20) NOT NULL,

`UserName` char(30) NOT NULL,

`Password` char(40) NOT NULL,

`EmailID` char(60) NOT NULL,

`StaffName` char(50) NOT NULL,

`StaffSalary` int(11) NOT NULL,

PRIMARY KEY (`StaffID`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `staff`

--

LOCK TABLES `staff` WRITE;

/\*!40000 ALTER TABLE `staff` DISABLE KEYS \*/;

INSERT INTO `staff` VALUES (1,'Admin','admin','admin','admin@railway.com','Ramesh',45000),(2,'TT','tt1','tt1','tt1@railway.com','Dinesh',54000),(3,'TT','tt2','tt2','tt2@railway.com','Mahesh',78000),(4,'TT','tt3','tt3','tt3@railway.com','Suresh',32145),(5,'TT','tt4','tt4','tt4@railway.com','Jitesh',2000),(6,'Caterer','c1','c1','c1@railway.com','Jignesh',3000),(7,'TT','tt5','tt5','tt5@railway.com','Sarvesh',5600),(8,'Driver','d1','d1','d1@railway.com','Bhavesh',100000);

/\*!40000 ALTER TABLE `staff` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Temporary view structure for view `staffdetail`

--

DROP TABLE IF EXISTS `staffdetail`;

/\*!50001 DROP VIEW IF EXISTS `staffdetail`\*/;

SET @saved\_cs\_client = @@character\_set\_client;

SET character\_set\_client = utf8;

/\*!50001 CREATE VIEW `staffdetail` AS SELECT

1 AS `StaffID`,

1 AS `Role`,

1 AS `UserName`,

1 AS `Password`,

1 AS `EmailID`,

1 AS `StaffName`,

1 AS `StaffSalary`\*/;

SET character\_set\_client = @saved\_cs\_client;

--

-- Table structure for table `stafftrain`

--

DROP TABLE IF EXISTS `stafftrain`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `stafftrain` (

`TrainID` int(11) NOT NULL,

`StaffID` int(11) NOT NULL,

PRIMARY KEY (`TrainID`,`StaffID`),

KEY `StaffTrain` (`StaffID`),

CONSTRAINT `StaffTrain` FOREIGN KEY (`StaffID`) REFERENCES `staff` (`StaffID`),

CONSTRAINT `TrainStaff` FOREIGN KEY (`TrainID`) REFERENCES `train` (`TrainID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `stafftrain`

--

LOCK TABLES `stafftrain` WRITE;

/\*!40000 ALTER TABLE `stafftrain` DISABLE KEYS \*/;

INSERT INTO `stafftrain` VALUES (1,2),(1,3),(2,4),(3,5),(4,5),(5,7);

/\*!40000 ALTER TABLE `stafftrain` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

/\*!50003 CREATE\*/ /\*!50017 DEFINER=`root`@`localhost`\*/ /\*!50003 TRIGGER staff\_insert\_check

Before INSERT ON StaffTrain

FOR EACH ROW

block1: BEGIN

DECLARE varStaffID INTEGER;

DECLARE varRole Varchar(20);

DECLARE MESSAGE\_TEXT Varchar(50);

Set varStaffID = NEW.StaffID;

Select Role into varRole from Staff where StaffId = varStaffID;

If varRole != 'TT' then

SIGNAL SQLSTATE '45000';

SET MESSAGE\_TEXT = 'You are not authorized!!!!';

end if;

END block1 \*/;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Table structure for table `station`

--

DROP TABLE IF EXISTS `station`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `station` (

`StationID` int(11) NOT NULL AUTO\_INCREMENT,

`StationName` char(50) NOT NULL,

PRIMARY KEY (`StationID`)

) ENGINE=InnoDB AUTO\_INCREMENT=17 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `station`

--

LOCK TABLES `station` WRITE;

/\*!40000 ALTER TABLE `station` DISABLE KEYS \*/;

INSERT INTO `station` VALUES (1,'New Delhi'),(2,'Chennai'),(3,'Mumbai'),(4,'Bangalore'),(5,'Kolkata'),(6,'Raipur'),(7,'Varanasi'),(8,'Hyderabad'),(9,'Goa'),(10,'Jaipur');

/\*!40000 ALTER TABLE `station` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `train`

--

DROP TABLE IF EXISTS `train`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `train` (

`TrainID` int(11) NOT NULL AUTO\_INCREMENT,

`TrainName` char(50) NOT NULL,

`TrainType` char(20) NOT NULL,

PRIMARY KEY (`TrainID`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `train`

--

LOCK TABLES `train` WRITE;

/\*!40000 ALTER TABLE `train` DISABLE KEYS \*/;

INSERT INTO `train` VALUES (1,'Gomti','Express'),(2,'GangaCavery','Superfast'),(3,'Duronto','Express'),(4,'Shatabdi','Express'),(5,'Rajdhani','Express');

/\*!40000 ALTER TABLE `train` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `trainstationtype`

--

DROP TABLE IF EXISTS `trainstationtype`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `trainstationtype` (

`TrainID` int(11) NOT NULL,

`StationID` int(11) NOT NULL,

`Type` char(50) NOT NULL,

`ArrivalTime` time DEFAULT NULL,

`DepartureTime` time DEFAULT NULL,

`Distance` int(11) DEFAULT NULL,

PRIMARY KEY (`TrainID`,`StationID`),

KEY `StationTrainStatus` (`StationID`),

CONSTRAINT `StationTrainStatus` FOREIGN KEY (`StationID`) REFERENCES `station` (`StationID`),

CONSTRAINT `TrainStationStatus` FOREIGN KEY (`TrainID`) REFERENCES `train` (`TrainID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `trainstationtype`

--

LOCK TABLES `trainstationtype` WRITE;

/\*!40000 ALTER TABLE `trainstationtype` DISABLE KEYS \*/;

INSERT INTO `trainstationtype` VALUES (1,1,'Source','00:00:00','03:22:56',0),(1,2,'Intermediate','04:22:56','04:26:56',50),(1,3,'Destination','05:22:56',NULL,100),(2,4,'Source',NULL,'05:22:56',0),(2,5,'Destination','05:22:56','00:00:00',600),(3,6,'Source',NULL,'05:22:56',0),(3,7,'Destination','05:22:56',NULL,500),(4,8,'Source','00:00:00','05:22:56',0),(4,9,'Destination','05:22:56',NULL,488),(5,5,'Destination','05:22:56',NULL,152),(5,10,'Source',NULL,'05:22:56',0);

/\*!40000 ALTER TABLE `trainstationtype` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `trainstatus`

--

DROP TABLE IF EXISTS `trainstatus`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `trainstatus` (

`TrainStatusID` int(11) NOT NULL AUTO\_INCREMENT,

`TrainID` int(11) NOT NULL,

`DaysName` char(20) NOT NULL,

`RunDate` date NOT NULL,

PRIMARY KEY (`TrainStatusID`,`TrainID`),

KEY `Train\_TrainStatus` (`TrainID`),

CONSTRAINT `Train\_TrainStatus` FOREIGN KEY (`TrainID`) REFERENCES `train` (`TrainID`)

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `trainstatus`

--

LOCK TABLES `trainstatus` WRITE;

/\*!40000 ALTER TABLE `trainstatus` DISABLE KEYS \*/;

INSERT INTO `trainstatus` VALUES (1,1,'Mon','2016-12-12'),(2,1,'Tue','2016-12-13'),(3,2,'Wed','2016-12-14'),(4,3,'Thu','2016-12-15'),(5,4,'Fri','2016-12-16'),(6,5,'Sat','2016-12-17');

/\*!40000 ALTER TABLE `trainstatus` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Temporary view structure for view `waitingpassengers`

--

DROP TABLE IF EXISTS `waitingpassengers`;

/\*!50001 DROP VIEW IF EXISTS `waitingpassengers`\*/;

SET @saved\_cs\_client = @@character\_set\_client;

SET character\_set\_client = utf8;

/\*!50001 CREATE VIEW `waitingpassengers` AS SELECT

1 AS `PNR`,

1 AS `PassengerBookingID`,

1 AS `SeatNumber`,

1 AS `BookingDate`,

1 AS `TicketStatus`\*/;

SET character\_set\_client = @saved\_cs\_client;

--

-- Dumping events for database 'railway'

--

/\*!50106 SET @save\_time\_zone= @@TIME\_ZONE \*/ ;

/\*!50106 DROP EVENT IF EXISTS `FULL\_BKP` \*/;

DELIMITER ;;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;;

/\*!50003 SET character\_set\_client = utf8 \*/ ;;

/\*!50003 SET character\_set\_results = utf8 \*/ ;;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;;

/\*!50003 SET @saved\_time\_zone = @@time\_zone \*/ ;;

/\*!50003 SET time\_zone = 'SYSTEM' \*/ ;;

/\*!50106 CREATE\*/ /\*!50117 DEFINER=`root`@`localhost`\*/ /\*!50106 EVENT `FULL\_BKP` ON SCHEDULE EVERY 1 WEEK STARTS '2016-12-12 00:00:00' ON COMPLETION NOT PRESERVE ENABLE DO CALL railway.sp\_full\_backup() \*/ ;;

/\*!50003 SET time\_zone = @saved\_time\_zone \*/ ;;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;;

DELIMITER ;

/\*!50106 SET TIME\_ZONE= @save\_time\_zone \*/ ;

--

-- Dumping routines for database 'railway'

--

/\*!50003 DROP FUNCTION IF EXISTS `Ticket\_Price` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` FUNCTION `Ticket\_Price`(PNR INT(11)) RETURNS decimal(10,2)

DETERMINISTIC

BEGIN

DECLARE PassengerBookingId Int(11);

DECLARE TrainID Int(11);

Declare SourceStationId Int(11);

Declare DestinationStationId Int(11);

Declare SourceDistance Int(11);

Declare DestinationDistance Int(11);

Declare TicketPrice decimal(10,2);

Select pt.PassengerBookingId into PassengerBookingId

from PassengerTicket as pt

where pt.PNR = PNR;

Select pb.TrainId into TrainID

from PassengerBooking as pb

where pb.PassengerBookingId = PassengerBookingId;

Select pst.StationId into SourceStationId

from PassengerStationType as pst

where pst.PassengerBookingId = PassengerBookingId and pst.type = 'Source';

Select pst.StationId into DestinationStationId

from PassengerStationType as pst

where pst.PassengerBookingId = PassengerBookingId and pst.type = 'Destination';

Select tst.Distance into SourceDistance

from TrainStationType as tst

where tst.TrainId = TrainID and tst.StationId = SourceStationId;

Select tst.Distance into DestinationDistance

from TrainStationType as tst

where tst.TrainId = TrainID and tst.StationId = DestinationStationId;

Set TicketPrice = abs((DestinationDistance - SourceDistance) \* 1.2);

return TicketPrice;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP FUNCTION IF EXISTS `Ticket\_Status` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` FUNCTION `Ticket\_Status`(PNR INT) RETURNS varchar(30) CHARSET utf8

DETERMINISTIC

BEGIN

DECLARE A varchar(30);

SELECT

t.ticketstatus

INTO A FROM

passengerticket AS t

WHERE

t.PNR = PNR;

RETURN A;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `SeatDetailsForATrain` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `SeatDetailsForATrain`(TrainId Int, RunDate Date , Category Char(30))

BEGIN

select t.TrainName,cc.Category,ts.RunDate,cc.SeatsRemaining, cc.SeatsWaiting

from train as t

inner join TrainStatus as ts

on t.TrainID = ts.TrainID

inner join ClassCategory cc

on ts.TrainStatusID = cc.TrainStatusID

where ts.TrainID = TrainID and ts.RunDate = RunDate and cc.Category= Category;

END ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `sp\_full\_bkp` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_full\_bkp`()

block1: BEGIN

DECLARE tab\_name char(50);

DECLARE q varchar(1500);

DECLARE done INTEGER DEFAULT 0;

DECLARE cursorBackupTable CURSOR FOR SELECT TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES WHERE table\_schema='railway' and table\_type='BASE TABLE';

DROP DATABASE IF EXISTS railway\_backup;

CREATE DATABASE railway\_backup;

open cursorBackupTable;

block2: begin

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

cur\_loop:Loop

FETCH cursorBackupTable into tab\_name;

IF done= 1 THEN LEAVE cur\_loop;

END IF;

SET @q=CONCAT('DROP TABLE IF EXISTS railway\_backup.',tab\_name);

PREPARE stmt FROM @q;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

SET @q=CONCAT('CREATE TABLE railway\_backup.',tab\_name,' AS SELECT \* FROM railway.',tab\_name,' WHERE 1=1');

PREPARE stmt FROM @q;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

END LOOP cur\_loop;

END block2;

close cursorBackupTable;

END block1 ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `TrainDetail` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `TrainDetail`(

IN TrainID INTEGER

)

block1:BEGIN

select t.TrainName, t.TrainType/\*, ts.DaysName, ts.rundate \*/, s.StationName, tst.Type, tst.ArrivalTime, tst.DepartureTime

from train as t

/\*inner join trainstatus as ts

on t.TrainID = ts.TrainID\*/

inner join trainstationtype as tst

on t.TrainID = tst.TrainID

inner join station as s

on t.TrainID = tst.TrainID and tst.StationID = s.StationID;

END block1 ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

/\*!50003 DROP PROCEDURE IF EXISTS `TrainStation` \*/;

/\*!50003 SET @saved\_cs\_client = @@character\_set\_client \*/ ;

/\*!50003 SET @saved\_cs\_results = @@character\_set\_results \*/ ;

/\*!50003 SET @saved\_col\_connection = @@collation\_connection \*/ ;

/\*!50003 SET character\_set\_client = utf8 \*/ ;

/\*!50003 SET character\_set\_results = utf8 \*/ ;

/\*!50003 SET collation\_connection = utf8\_general\_ci \*/ ;

/\*!50003 SET @saved\_sql\_mode = @@sql\_mode \*/ ;

/\*!50003 SET sql\_mode = 'STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION' \*/ ;

DELIMITER ;;

CREATE DEFINER=`root`@`localhost` PROCEDURE `TrainStation`(

IN StationId INTEGER

)

block1:BEGIN

select s.StationName, t.TrainName, t.TrainType, tst.Type, tst.ArrivalTime, tst.DepartureTime

from Station as s

inner join trainstationtype as tst

on s.StationID = tst.StationID

inner join train as t

on tst.TrainID = t.TrainID

where tst.StationID = StationId;

END block1 ;;

DELIMITER ;

/\*!50003 SET sql\_mode = @saved\_sql\_mode \*/ ;

/\*!50003 SET character\_set\_client = @saved\_cs\_client \*/ ;

/\*!50003 SET character\_set\_results = @saved\_cs\_results \*/ ;

/\*!50003 SET collation\_connection = @saved\_col\_connection \*/ ;

--

-- Final view structure for view `staffdetail`

--

/\*!50001 DROP VIEW IF EXISTS `staffdetail`\*/;

/\*!50001 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50001 SET @saved\_cs\_results = @@character\_set\_results \*/;

/\*!50001 SET @saved\_col\_connection = @@collation\_connection \*/;

/\*!50001 SET character\_set\_client = utf8 \*/;

/\*!50001 SET character\_set\_results = utf8 \*/;

/\*!50001 SET collation\_connection = utf8\_general\_ci \*/;

/\*!50001 CREATE ALGORITHM=UNDEFINED \*/

/\*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER \*/

/\*!50001 VIEW `staffdetail` AS select `st`.`StaffID` AS `StaffID`,`st`.`Role` AS `Role`,`st`.`UserName` AS `UserName`,`st`.`Password` AS `Password`,`st`.`EmailID` AS `EmailID`,`st`.`StaffName` AS `StaffName`,`st`.`StaffSalary` AS `StaffSalary` from `staff` `st` where (`st`.`Role` <> 'Admin') \*/;

/\*!50001 SET character\_set\_client = @saved\_cs\_client \*/;

/\*!50001 SET character\_set\_results = @saved\_cs\_results \*/;

/\*!50001 SET collation\_connection = @saved\_col\_connection \*/;

--

-- Final view structure for view `waitingpassengers`

--

/\*!50001 DROP VIEW IF EXISTS `waitingpassengers`\*/;

/\*!50001 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50001 SET @saved\_cs\_results = @@character\_set\_results \*/;

/\*!50001 SET @saved\_col\_connection = @@collation\_connection \*/;

/\*!50001 SET character\_set\_client = utf8 \*/;

/\*!50001 SET character\_set\_results = utf8 \*/;

/\*!50001 SET collation\_connection = utf8\_general\_ci \*/;

/\*!50001 CREATE ALGORITHM=UNDEFINED \*/

/\*!50013 DEFINER=`root`@`localhost` SQL SECURITY DEFINER \*/

/\*!50001 VIEW `waitingpassengers` AS select `pt`.`PNR` AS `PNR`,`pt`.`PassengerBookingID` AS `PassengerBookingID`,`pt`.`SeatNumber` AS `SeatNumber`,`pt`.`BookingDate` AS `BookingDate`,`pt`.`TicketStatus` AS `TicketStatus` from `passengerticket` `pt` where (`pt`.`TicketStatus` = 'Waiting') \*/;

/\*!50001 SET character\_set\_client = @saved\_cs\_client \*/;

/\*!50001 SET character\_set\_results = @saved\_cs\_results \*/;

/\*!50001 SET collation\_connection = @saved\_col\_connection \*/;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2016-12-12 11:10:44