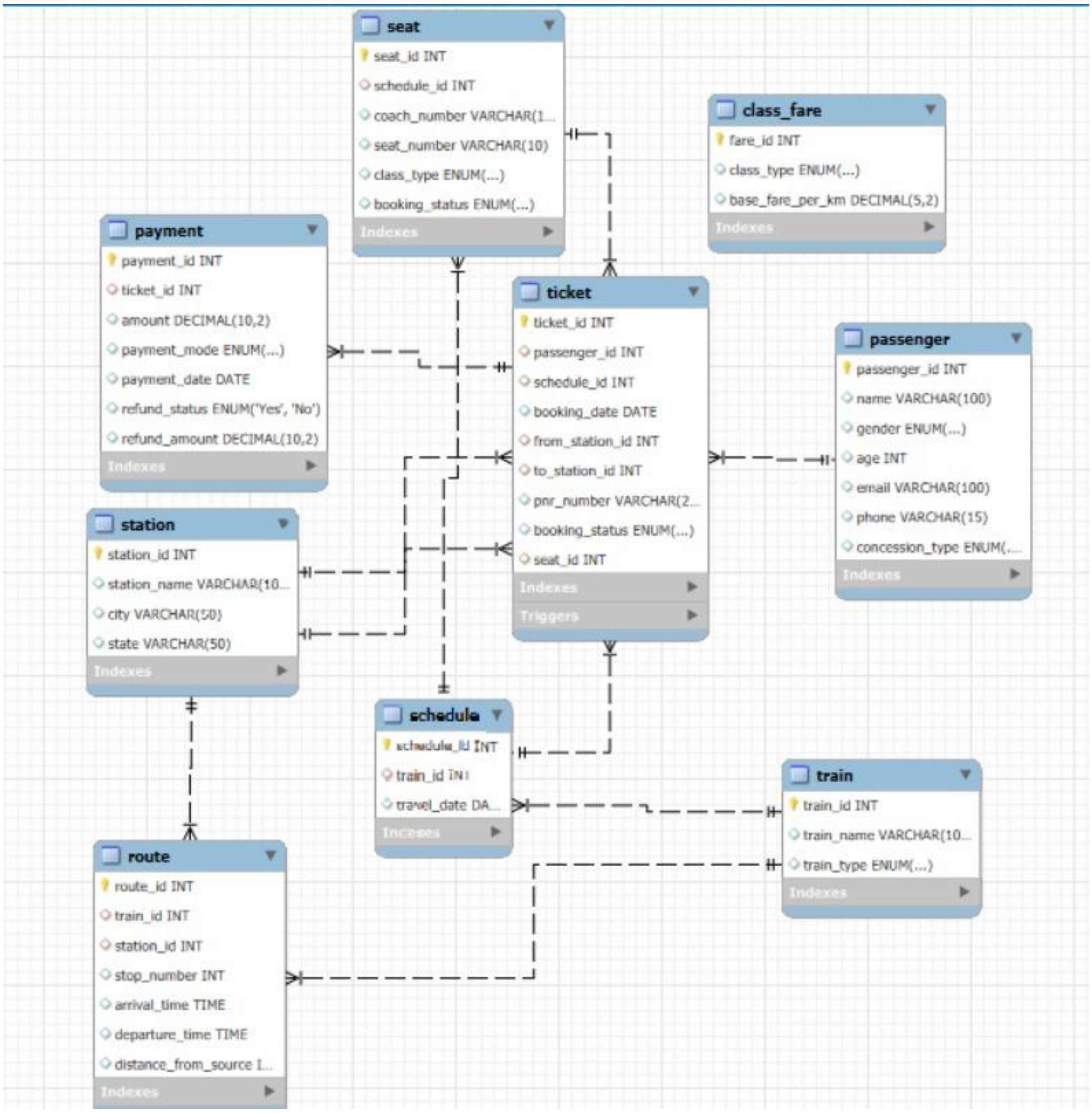
RAILWAY RESERVATION SYSTEM

ER DIAGRAM:



1. passenger Table

This table stores details about each passenger.

passenger_id – Primary Key

name, gender, age, email, phone – Personal details

concession_type – Type of concession (e.g., None, Senior Citizen, etc.)

Connected To: ticket (One passenger can have many tickets)

2. ticket Table

This represents a booking made by a passenger.

ticket_id – Primary Key

passenger_id – Foreign Key → passenger

schedule_id – Foreign Key → schedule

from_station_id / to_station_id – Foreign Keys → station

pnr_number – Unique identifier for the ticket

booking_status – Confirmed, Waitlisted, etc.

seat_id – Foreign Key → seat

booking_date – Date of ticket booking

Connected To: passenger, schedule, station (twice – from and to), seat, payment

3. schedule Table

This tells which train runs on what date.

schedule_id – Primary Key

train_id – Foreign Key → train

travel_date – Actual date of journey

Connected To: train, ticket, seat

4. train Table

Holds train-specific details.

train_id – Primary Key

train_name

train_type – Express, Passenger, Superfast etc.

Connected To: schedule, route

5. station Table

Contains information of stations.

station_id – Primary Key

station_name, city, state

Connected To: ticket (twice), route

6. route Table

Defines the route of each train, listing all the stops.

route_id – Primary Key

train_id – Foreign Key → train

station_id – Foreign Key → station

stop_number – Order of stop

arrival_time, departure_time

distance_from_source – For fare calculation

Connected To: station, train

7. seat Table

Shows available seats for each schedule.

seat_id – Primary Key

schedule_id – Foreign Key → schedule

coach_number, seat_number

class_type – e.g., Sleeper, AC 3-Tier

booking_status – Booked / Available

Connected To: ticket, schedule

8. class_fare Table

Contains fare per kilometer for each class.

fare_id – Primary Key

class_type

base_fare_per_km – Used for fare calculation

9. payment Table

Stores payment information.

payment_id – Primary Key

ticket_id – Foreign Key → ticket

amount, payment_mode, payment_date

refund_status, refund_amount

Connected To: ticket

Relationships Summary:

One Passenger ↔ Many Tickets

One Schedule ↔ Many Tickets / Many Seats

One Train ↔ Many Schedules / Many Routes

One Ticket ↔ One Seat (if booked)

Each Route Entry ↔ One Train & One Station

This ER Diagram Supports:

- Seat allocation
- Ticket booking with PNR
- Fare calculation based on distance & class
- Tracking train schedules
- Payment & refund handling

RELATIONAL SCHEMA:

Passenger table:

Passenger(passenger_id PK, name, gender, age, email, phone, concession_type)

Train Table:

Train(train_id PK, train_name, train_type)

Station Table:

Station(station_id PK, station_name, city, state)

Route Table:

Route(route_id PK, train_id FK → Train(train_id), station_id FK → Station(station_id), stop_number, arrival_time, departure_time, distance_from_source)

Schedule Table:

Schedule(schedule_id PK, train_id FK → Train(train_id), travel_date)

Seat Table:

Seat(seat_id PK, schedule_id FK → Schedule(schedule_id), coach_number, seat_number, class_type, booking_status)

Ticket Table:

Ticket(ticket_id PK, passenger_id FK → Passenger(passenger_id), schedule_id FK → Schedule(schedule_id), booking_date, from_station_id FK → Station(station_id),

to_station_id FK → Station(station_id),

pnr_number UNIQUE, booking_status, seat_id FK → Seat(seat_id))

Payment Table:

Payment(payment_id PK, ticket_id FK → Ticket(ticket_id),

amount, payment_mode, payment_date,

refund_status, refund_amount)

Class Fare Table:

Class_Fare(fare_id PK, class_type, base_fare_per_km)

- All the above tables are in *Third Normal Form(3NF)*

Examples:

Passenger:

No transitive dependencies. All fields depend on passenger_id.

Ticket:

seat_id, from_station_id, and to_station_id are FKs, but not functionally determining anything else in the table. OK.

Payment:

Depends only on ticket_id. No transitive dependency. OK.

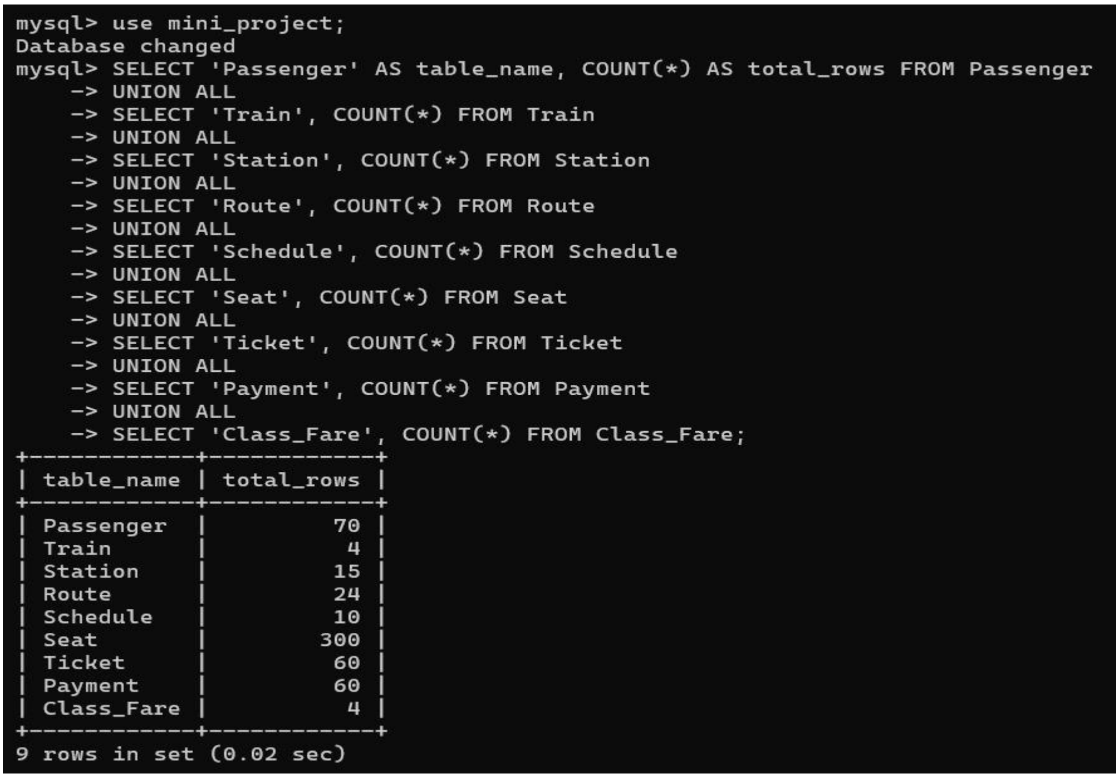
Seat:

Booking status, class, etc., depend on seat_id. All good.

Class_Fare:

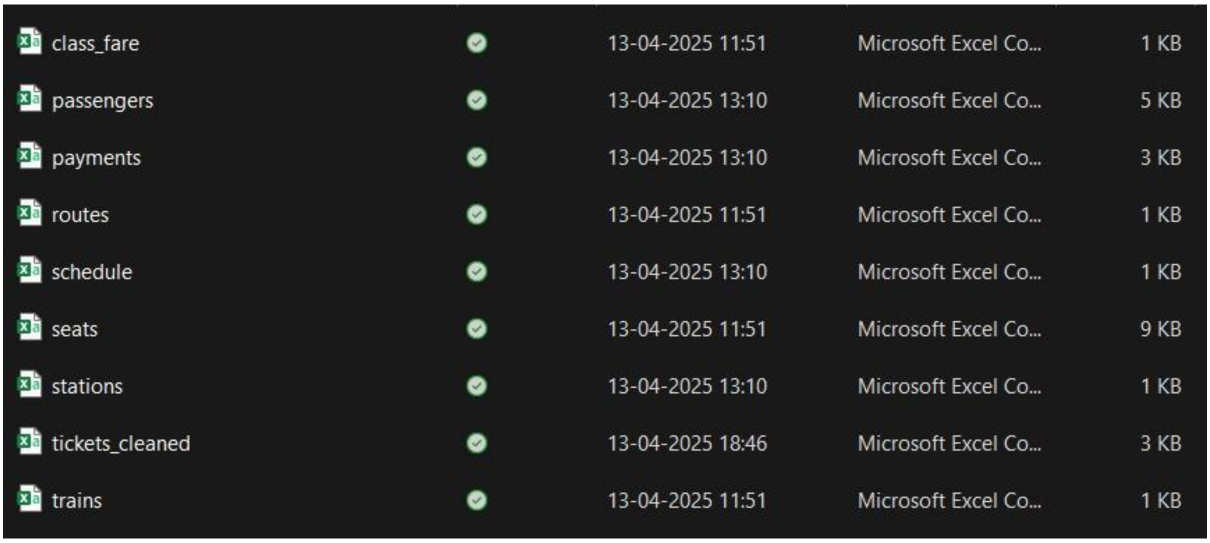
Each class_type maps to one base_fare_per_km. Normalized.

SAMPLE DATA SUMMARY:



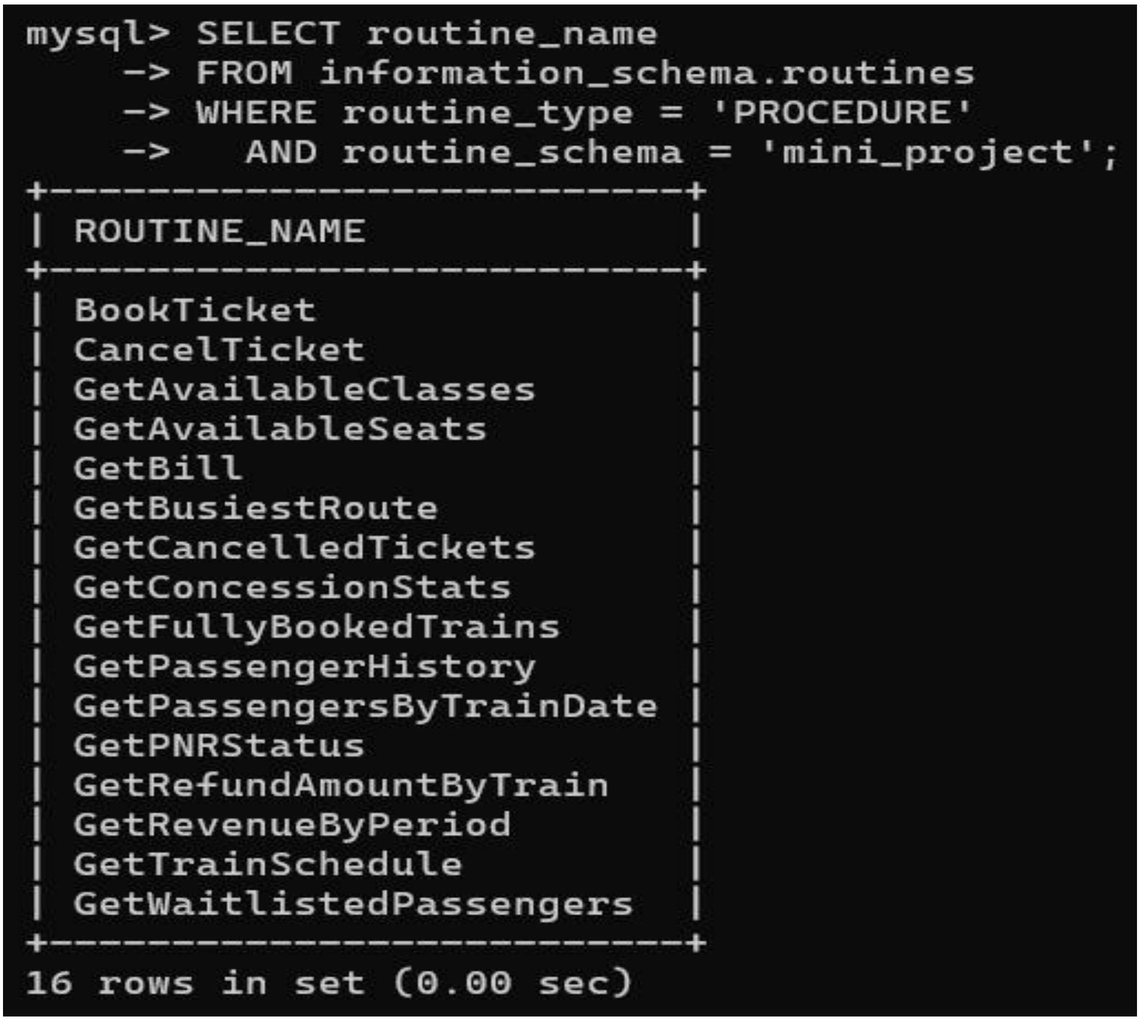
```
mysql> use mini_project;
Database changed
mysql> SELECT 'Passenger' AS table_name, COUNT(*) AS total_rows FROM Passenger
    -> UNION ALL
    -> SELECT 'Train', COUNT(*) FROM Train
    -> UNION ALL
    -> SELECT 'Station', COUNT(*) FROM Station
    -> UNION ALL
    -> SELECT 'Route', COUNT(*) FROM Route
    -> UNION ALL
    -> SELECT 'Schedule', COUNT(*) FROM Schedule
    -> UNION ALL
    -> SELECT 'Seat', COUNT(*) FROM Seat
    -> UNION ALL
    -> SELECT 'Ticket', COUNT(*) FROM Ticket
    -> UNION ALL
    -> SELECT 'Payment', COUNT(*) FROM Payment
    -> UNION ALL
    -> SELECT 'Class_Fare', COUNT(*) FROM Class_Fare;
+------------+------------+
| table_name | total_rows |
+------------+------------+
| Passenger  |         70 |
| Train      |          4 |
| Station    |         15 |
| Route      |         24 |
| Schedule   |         10 |
| Seat       |        300 |
| Ticket     |         60 |
| Payment    |         60 |
| Class_Fare |          4 |
+------------+------------+
9 rows in set (0.02 sec)
```

Data Entered:



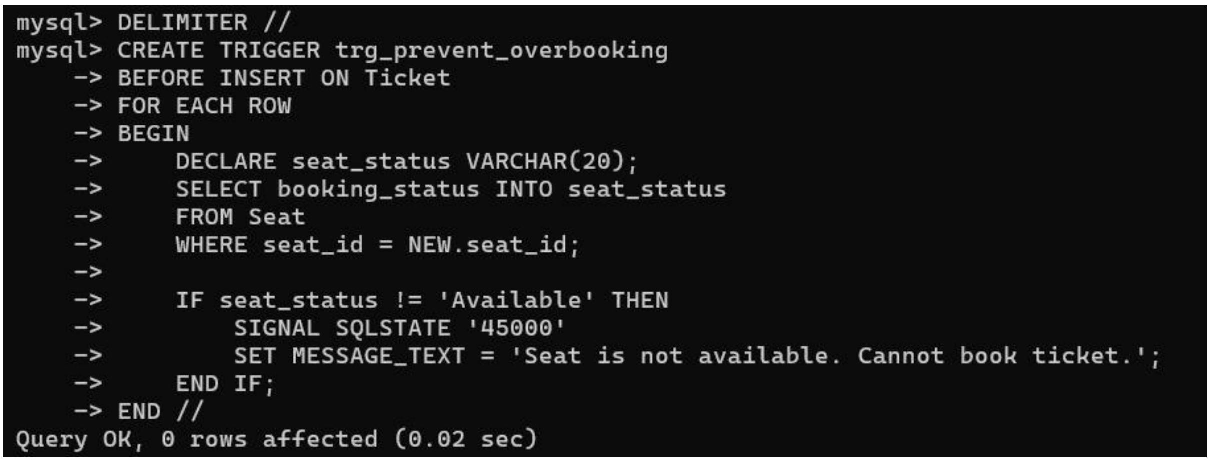
class_fare	✓	13-04-2025 11:51	Microsoft Excel Co...	1 KB
passengers	✓	13-04-2025 13:10	Microsoft Excel Co...	5 KB
payments	✓	13-04-2025 13:10	Microsoft Excel Co...	3 KB
routes	✓	13-04-2025 11:51	Microsoft Excel Co...	1 KB
schedule	✓	13-04-2025 13:10	Microsoft Excel Co...	1 KB
seats	✓	13-04-2025 11:51	Microsoft Excel Co...	9 KB
stations	✓	13-04-2025 13:10	Microsoft Excel Co...	1 KB
tickets_cleaned	✓	13-04-2025 18:46	Microsoft Excel Co...	3 KB
trains	✓	13-04-2025 11:51	Microsoft Excel Co...	1 KB

PROCEDRES INCLUDED [BOTH POTENTIAL AND ADDITIONAL]:

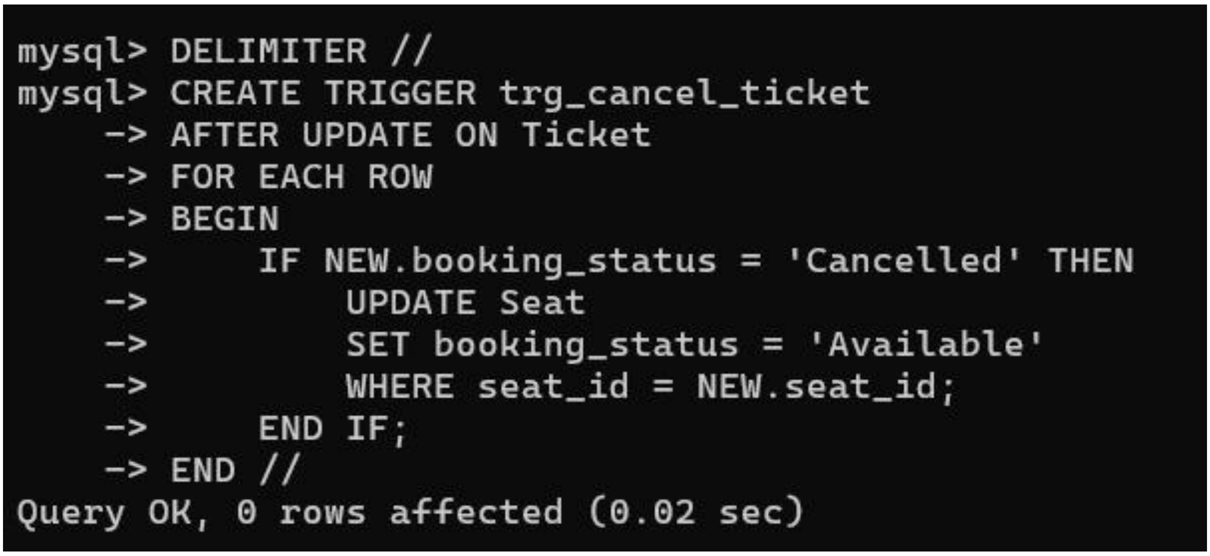


```
mysql> SELECT routine_name
    -> FROM information_schema.routines
    -> WHERE routine_type = 'PROCEDURE'
    ->   AND routine_schema = 'mini_project';
+---------------------------+
| ROUTINE_NAME              |
+---------------------------+
| BookTicket                |
| CancelTicket              |
| GetAvailableClasses       |
| GetAvailableSeats         |
| GetBill                   |
| GetBusiestRoute           |
| GetCancelledTickets       |
| GetConcessionStats        |
| GetFullyBookedTrains      |
| GetPassengerHistory       |
| GetPassengersByTrainDate  |
| GetPNRStatus              |
| GetRefundAmountByTrain    |
| GetRevenueByPeriod        |
| GetTrainSchedule          |
| GetWaitlistedPassengers   |
+---------------------------+
16 rows in set (0.00 sec)
```

TRIGGERS:

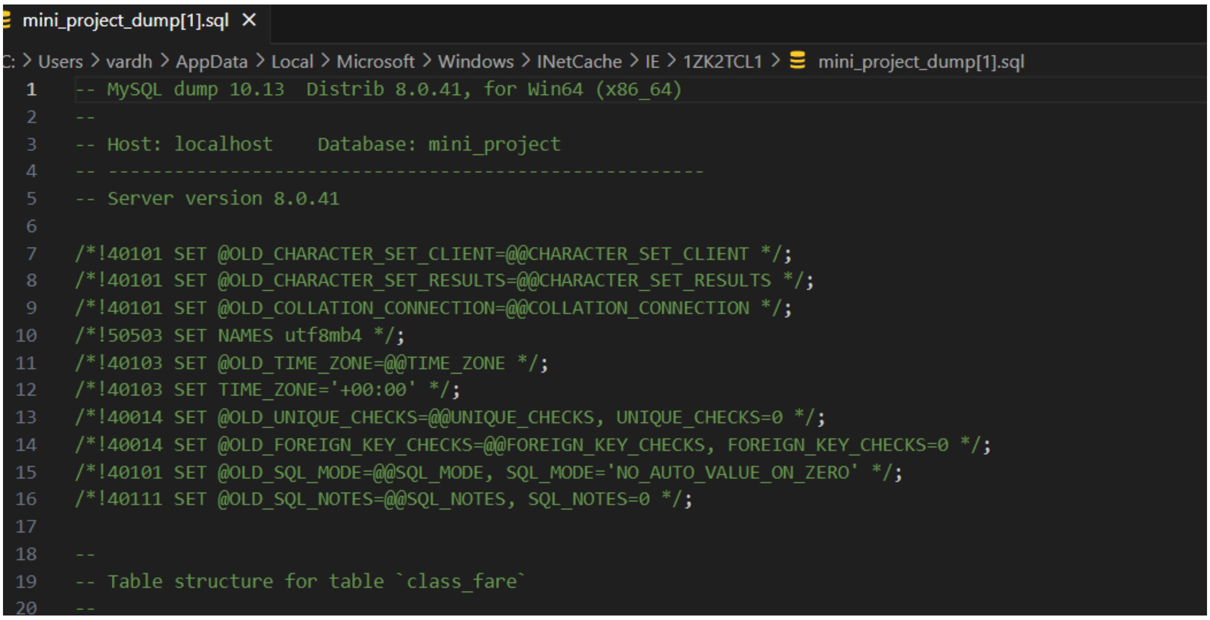


```
mysql> DELIMITER //
mysql> CREATE TRIGGER trg_prevent_overbooking
    -> BEFORE INSERT ON Ticket
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE seat_status VARCHAR(20);
    ->     SELECT booking_status INTO seat_status
    ->     FROM Seat
    ->     WHERE seat_id = NEW.seat_id;
    ->
    ->     IF seat_status != 'Available' THEN
    ->         SIGNAL SQLSTATE '45000'
    ->         SET MESSAGE_TEXT = 'Seat is not available. Cannot book ticket.';
    ->     END IF;
    -> END //
Query OK, 0 rows affected (0.02 sec)
```



```
mysql> DELIMITER //
mysql> CREATE TRIGGER trg_cancel_ticket
    -> AFTER UPDATE ON Ticket
    -> FOR EACH ROW
    -> BEGIN
    ->     IF NEW.booking_status = 'Cancelled' THEN
    ->         UPDATE Seat
    ->         SET booking_status = 'Available'
    ->         WHERE seat_id = NEW.seat_id;
    ->     END IF;
    -> END //
Query OK, 0 rows affected (0.02 sec)
```

SQL FILE DUMP:



mini_project_dump[1].sql ×
C: > Users > vardh > AppData > Local > Microsoft > Windows > INetCache > IE > 1ZK2TCL1 > mini_project_dump[1].sql

```
-- MySQL dump 10.13  Distrib 8.0.41, for Win64 (x86_64)
--
-- Host: localhost    Database: mini_project
-- ------------------------------------------------------
-- Server version 8.0.41

/*!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 */;
/*!50503 SET NAMES utf8mb4 */;
/*!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 `class_fare`
--
```