Vehicle Monitoring and Tracking System (DBA Assignment Group # 62)

Version Number	Date	Author/Owner	Description of Change
0.1	10/10/2021	Kishor Mane / Uday	Draft of Problem Statement & Requirement Definition
0.2	28/10/2021	Padmanabhan, Uday, Aswathy, Hariprasad, Upendra	Updated document with ER Diagram,
1.0	5/11/2021	Kishor Mane	Draft before Release
1.1	9/11/2021	Kishor Mane	After review updated Section 3, Section 4 and Section 8
1.2	10/11/2021	Upendra	After review updated Section4

Table of Contents

1 Team Members and their Responsibilities 4

2 Problem Statement & Requirements Definition 5

2.1 Current Business scope: 5

2.2 Future Business scope 5

2.3 Challenges 5

2.4 Requirements 6

3 Entity Relationship Model 7

4 Relational Database Schema 8

5 Table Definitions 9

6 Stored Procedures/ Triggers 14

1. Procedures 14

2. Triggers 14

7 User Interface and Outcome 15

8 Appendix 22

Tools and Technology Used 22

1 Team Members and their Responsibilities

SL No	Name	BITS ID	Responsibility
1	Upendra Singh Rathore	2021mt12206@wilp.bits-pilani.ac.in	Group Leader, Physical Design / Implementation
2	Hariprasad P	2021mt12015@wilp.bits-pilani.ac.in	Logical Design, Physical design/ implementation
3	Narendra Nath Mahato	2020mt93258@wilp.bits-pilani.ac.in	Physical Design / Implementation
4	Kishor Vijaykumar Mane	2020HT66041@wilp.bits-pilani.ac.in	Problem Statement / Requirement , Documentation, Facilitation
5	Aswathy N A	2021mt12360@wilp.bits-pilani.ac.in	Logical Design
6	Saurabh Rohit Narvekar	2021mt12019@wilp.bits-pilani.ac.in	Conceptual Design
7	Padmanabham Bodda	2021MT12271@wilp.bits-pilani.ac.in	Logical Design, Physical Design / Implementation
8	Uday Kiran Pichika	2021MT12335@wilp.bits-pilani.ac.in	Problem Statement / Requirement , Physical Design / Implementation
9	Nakul Krishnan	2021MT12381@wilp.bits-pilani.ac.in	Conceptual Design
10	Karthikeyan V	2021MT12458@wilp.bits-pilani.ac.in	Review Document

2 Problem Statement & Requirements Definition

Vehicle Monitoring & Tracking System (VMTS App):

We decided to create a Rental Vehicle Monitoring & Tracking System that can help organize Rental Vehicles tracking and business monitoring for a business owner who can track the usage of Vehicles by their usage, expenses etc.

VMTS application will be designed to assist Travel Business through which real aspects of business can be monitored and tracked for a given period of time. It will help the business owner to take data driven decisions either to rectify certain plan or actions or add a check to bring back business on track. Further futuristic aspect of this is to predict certain aspects which may be hidden but can be explored via data analytics.

2.1 Current Business scope:

Business owner can have one or more taxis which he/she can rent on monthly basis (e.g. certain government projects like Metro construction) for which he gets paid on monthly frequency on the agreed terms between the parties.

2.2 Future Business scope

Business owner has a plan to expand his travel business which he will lease his taxi either to run in within the city or Outstation

2.3 Challenges

Main challenge is to figure out investments and returns on monthly basis to reconcile and assess the profitability of business

Investments / Expenses

- Office Expenses: Monthly Rent, Electricity , Phone,
- Employee / Staff Expenses : Salary, Bonus
- Vehicle Operations Cost: Fuel Expenses on daily basis by the different vehicle types (CNG, petrol, diesel), Challan / Penalties cost & unforeseen expenses. (miscellaneous expenses)
- Vehicle Maintenance cost (regular service cost in 2 month once) , expenses occur while vehicle on road e.g. punchers repairs etc)
- Monthly Diver cost (as per contract, for a given 8 hrs of the day).

Returns

- Rental charges are based on per kilometre usage of vehicle, customer will pay based on fixed usage with fixed values and add on usage with additional rates e.g. Monthly 18000 INR for first 1500 kilometres and then after 10 INR per kilometre.
- Accounts Receivable by customer

2.4 Requirements

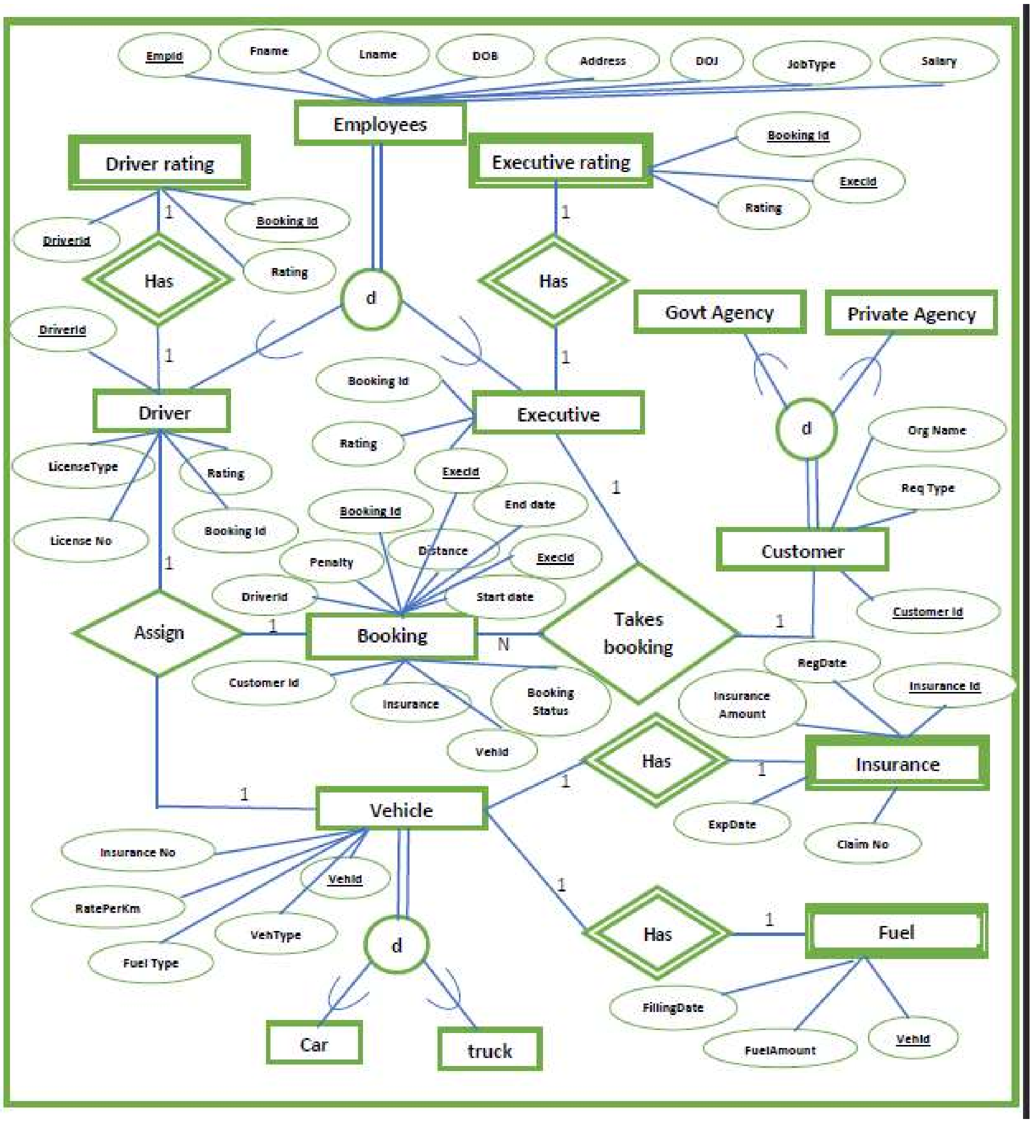
Create a database application for business owners to maintain & track the Daily / Monthly expenses, Account Receivables, Payments etc and analyse the data for making better business decisions for profitability.

Reports

- Monthly Requests per customer
- Daily Usage of vehicles in kilometres by customer.
- Monthly billing per customer with Travel Request details and authorization.
- Monthly earnings / payments by customer
- Daily Fuel expenses by vehicle
- Daily miscellaneous expenses by vehicle.
- Mileage cost per vehicle per month.
- Monthly Employee Expenses
- Monthly Driver Expenses
- Monthly Total Expenses
- Monthly Total Earnings

3 Entity Relationship Model

This section details the ER Model for VMTS application



4 Relational Database Schema

Below detail for RDS

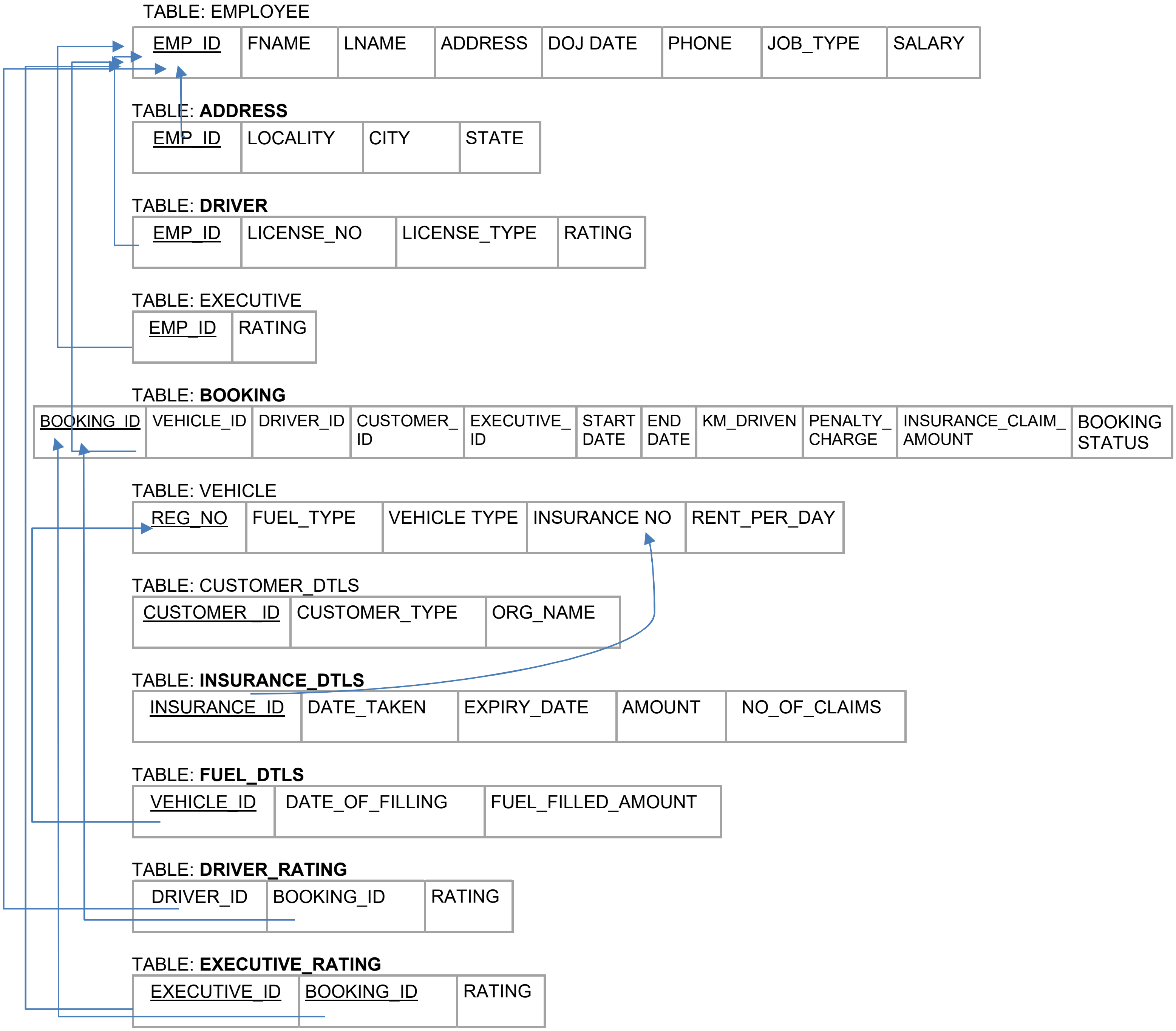


TABLE: EMPLOYEE

EMP_ID	FNAME	LNAME	ADDRESS	DOJ DATE	PHONE	JOB_TYPE	SALARY

TABLE: **ADDRESS**

EMP_ID	LOCALITY	CITY	STATE

TABLE: **DRIVER**

EMP_ID	LICENSE_NO	LICENSE_TYPE	RATING

TABLE: EXECUTIVE

EMP_ID	RATING

TABLE: **BOOKING**

BOOKING_ID	VEHICLE_ID	DRIVER_ID	CUSTOMER_ ID	EXECUTIVE_ ID	START DATE	END DATE	KM_DRIVEN	PENALTY_ CHARGE	INSURANCE_CLAIM_ AMOUNT	BOOKING STATUS

TABLE: VEHICLE

REG_NO	FUEL_TYPE	VEHICLE TYPE	INSURANCE NO	RENT_PER_DAY

TABLE: CUSTOMER_DTLS

CUSTOMER_ ID	CUSTOMER_TYPE	ORG_NAME

TABLE: **INSURANCE_DTLS**

INSURANCE_ID	DATE_TAKEN	EXPIRY_DATE	AMOUNT	NO_OF_CLAIMS

TABLE: **FUEL_DTLS**

VEHICLE_ID	DATE_OF_FILLING	FUEL_FILLED_AMOUNT

TABLE: **DRIVER_RATING**

DRIVER_ID	BOOKING_ID	RATING

TABLE: **EXECUTIVE_RATING**

EXECUTIVE_ID	BOOKING_ID	RATING

5 Table Definitions

Table Definitions and Data Contents

```
CREATE TABLE VMTS.EMPLOYEE (

  EMP_ID INTEGER NOT NULL AUTO_INCREMENT,

  FNAME VARCHAR(150) ,

  LNAME VARCHAR(150) ,

  DOJ DATE,

  PHONE INTEGER,

  JOB_TYPE INTEGER DEFAULT NULL,

  SALARY INTEGER DEFAULT NULL,

  PRIMARY KEY (EMP_ID)

);


CREATE TABLE VMTS.ADDRESS (

  EMP_ID INTEGER NOT NULL,

  LOCALITY VARCHAR(150) DEFAULT NULL,

  CITY VARCHAR(150) DEFAULT NULL,

  STATE VARCHAR(150) DEFAULT NULL,

  PRIMARY KEY (EMP_ID),

  FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE (EMP_ID)

)


CREATE TABLE VMTS.DRIVER(

 EMP_ID INTEGER NOT NULL,
 LICENSE_NO VARCHAR(32),
 LICENSE_TYPE enum('lmv','hmv'),
 RATING DOUBLE,
 PRIMARY KEY (EMP_ID),
 FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID)
);
CREATE TABLE VMTS.EXECUTIVE(
 EMP_ID INTEGER NOT NULL,
 RATING DOUBLE,
 PRIMARY KEY (EMP_ID),
 FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID)
);

CREATE TABLE  VMTS.VEHICLE (
  REG_NO VARCHAR(10) NOT NULL,
  FUEL_TYPE VARCHAR(20) DEFAULT NULL,
  INSURANCE_NO VARCHAR(150) DEFAULT NULL,
  VEHICLE_TYPE enum('LMV','HMV') DEFAULT NULL,
  RENT_PER_DAY DOUBLE DEFAULT NULL,
  PRIMARY KEY (REG_NO),
  KEY INSURANCE_NO (INSURANCE_NO),
  FOREIGN KEY (INSURANCE_NO) REFERENCES INSURANCE_DTLS (INSURANCE_ID)
)

CREATE TABLE VMTS.INSURANCE_DTLS (
  INSURANCE_ID VARCHAR (20) NOT NULL,
  DATE_TAKEN DATE ,
  EXPIRY_DATE DATE ,
  AMOUNT  DOUBLE,
  NO_OF_CLAIMS INTEGER,
  PRIMARY KEY (INSURANCE_ID),
  FOREIGN KEY (INSURANCE_ID) REFERENCES VEHICLE(INSURANCE_NO)
);
CREATE TABLE VMTS.FUEL_DTLS(
 VEHICLE_ID VARCHAR(10) NOT NULL,
 FUEL_FILLED_AMOUNT DOUBLE,
 DATE_OF_FILLING DATE,
 FUEL_FILLED DOUBLE,
 PRIMARY KEY (VEHICLE_ID),
 FOREIGN KEY (VEHICLE_ID) REFERENCES VEHICLE(REG_NO)
);
CREATE TABLE VMTS.CUSTOMER_DTLS (
  CUSTOMER_ID INTEGER NOT NULL AUTO_INCREMENT,
  CUSTOMER_TYPE enum('govt','private') ,
  ORG_NAME VARCHAR (150),
  PRIMARY KEY (CUSTOMER_ID)
);
CREATE TABLE VMTS.BOOKING (
  BOOKING_ID INTEGER NOT NULL AUTO_INCREMENT,
  VEHICLE_ID VARCHAR(20) ,
  CUSTOM_ID VARCHAR (150),

  DIVER_ID INTEGER,

  EXECUTIVE_ID INTEGER,

  START_DATE DATE,

  END_DATE DATE,

  PENALTY_CHARGE DOUBLE,

  KM_DRIVEN DOUBLE,

  INSURANCE_CLAIM_AMOUNT DOUBLE,

  BOOKING_STATUS enum('booked','confirmed','cancelled','completed','paid'),

  PRIMARY KEY (BOOKING_ID)

);

CREATE TABLE VMTS.DRIVER_RATING(

    DRIVER_ID INTEGER NOT NULL,

    BOOKING_ID INTEGER NOT NULL,

    RATING DOUBLE,

    PRIMARY KEY (DRIVER_ID, BOOKING_ID),

    FOREIGN KEY (DRIVER_ID) REFERENCES EMPLOYEE(EMP_ID),

    FOREIGN KEY (BOOKING_ID) REFERENCES BOOKING(BOOKING_ID)

);
```

```
CREATE TABLE VMTS.EXECUTIVE_RATING(

   EXECUTIVE_ID INTEGER NOT NULL,

   BOOKING_ID INTEGER NOT NULL,

   RATING DOUBLE,

   PRIMARY KEY (EXECUTIVE_ID, BOOKING_ID),

   FOREIGN KEY (EXECUTIVE_ID) REFERENCES EMPLOYEE(EMP_ID),

   FOREIGN KEY (BOOKING_ID) REFERENCES BOOKING(BOOKING_ID)

);
```

6 Stored Procedures/ Triggers

1. Procedures

1. Procedure to update driver rating in Driver table based on the average values of corresponding records in Driver_rating table.

```
DELIMITER //
 CREATE PROCEDURE vmts.update_driver_rating ()
 BEGIN
  update vmts.driver d set d.rating =  round((select avg(r.rating)
  from vmts.driver_rating r where r.driver_id=d.emp_id group by r.driver_id ),2) ;
 END
```

2. Procedure to update executive rating in Executive table based on the average ratings from executive_rating table.

```
DELIMITER //
 CREATE PROCEDURE vmts.update_executive_rating ()
 BEGIN
  update vmts.executive e set e.rating =  round((select avg(r.rating)
  from vmts.executive_rating r where r.executive_id=e.emp_id group by r.executive_id ),2) ;
 END
```

2. Triggers

1. Trigger to call the update_driver_rating procedure for each row insertion in driver_rating table.

```
DELIMITER //
CREATE TRIGGER vmts.update_driver
   AFTER INSERT
   ON vmts.driver_rating FOR EACH ROW
BEGIN
   call vmts.update_driver_rating();
END
```

2. Trigger to execute update_execute_rating procedure for each record insertion in executive_rating table.

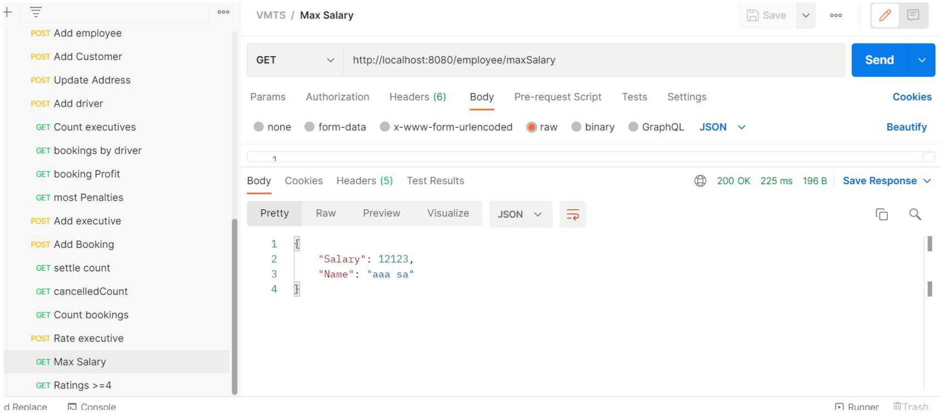
```
DELIMITER //
CREATE TRIGGER vmts.update_executive
   AFTER INSERT
   ON vmts.executive_rating FOR EACH ROW
BEGIN
   call vmts.update_executive_rating();
END
```

7 User Interface and Outcome

Below section details the UI Screens and SQL Statements including outcome of the application

1. Name of emp with the highest salary

Select * from EMPLOYEE where SALARY =(select MAX(SALARY) from EMPLOYEE);



VMTS / Max Salary

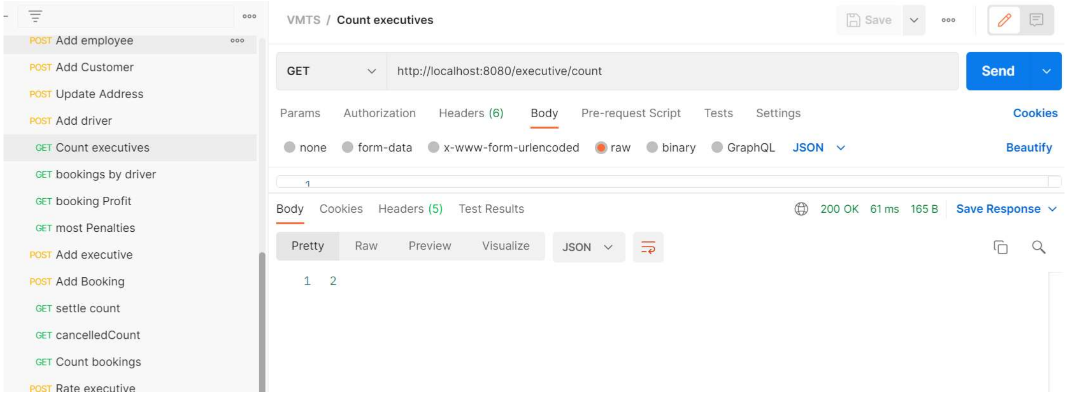
GET http://localhost:8080/employee/maxSalary

200 OK 225 ms 196 B

```
{
    "Salary": 12123,
    "Name": "aaa sa"
}
```

2. Number of executives in the company

SELECT COUNT(*) FROM VMTS.EXECUTIVE;



VMTS / Count executives

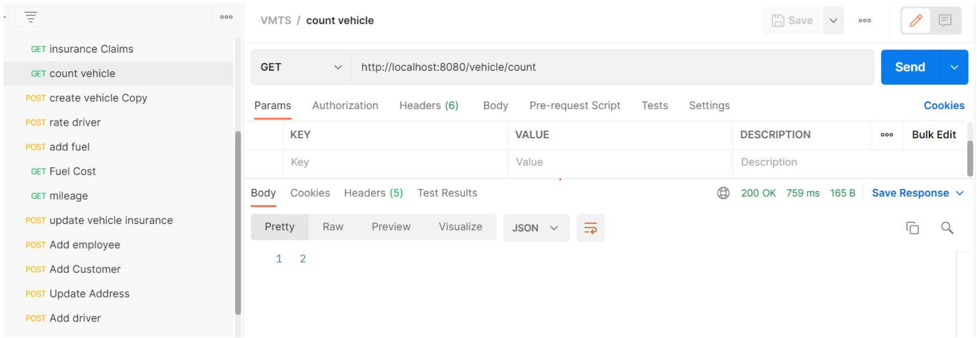
GET http://localhost:8080/executive/count

200 OK 61 ms 165 B

```
2
```

3. Number of vehicles in the company

SELECT COUNT(*) FROM VMTS.VEHICLE;



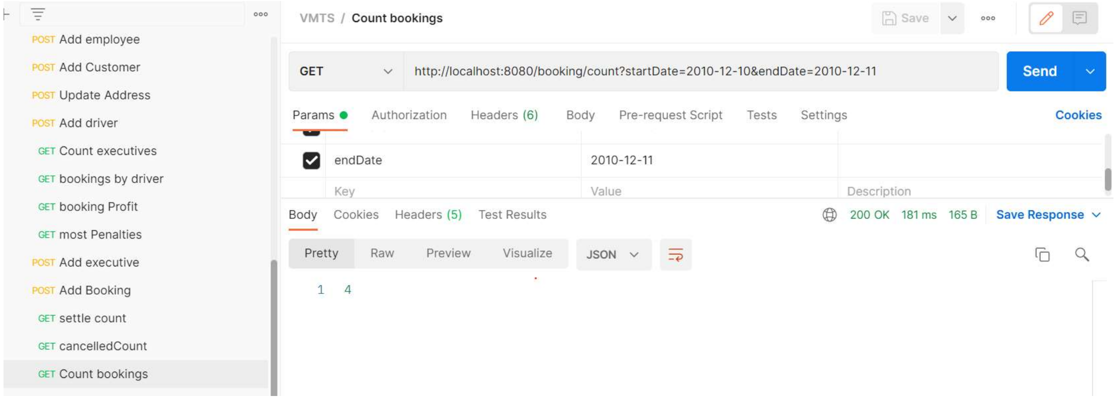
GET http://localhost:8080/vehicle/count

200 OK 759 ms 165 B

```
2
```

4. <u>Number of total bookings made between specific dates</u>

SELECT COUNT(*) FROM VMTS.BOOKING WHERE START_DATE BETWEEN <from_date> AND <to_date>;



GET http://localhost:8080/booking/count?startDate=2010-12-10&endDate=2010-12-11

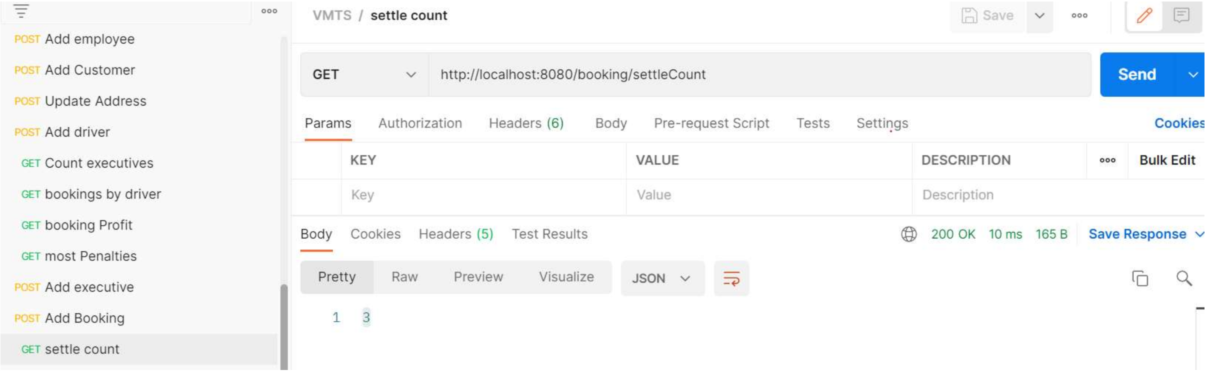
☑ endDate 2010-12-11

200 OK 181 ms 165 B

```
4
```

5. <u>Number of bookings that need to be settled</u>

SELECT COUNT(*) FROM VMTS.BOOKING WHERE BOOKING_STATUS = 'completed';



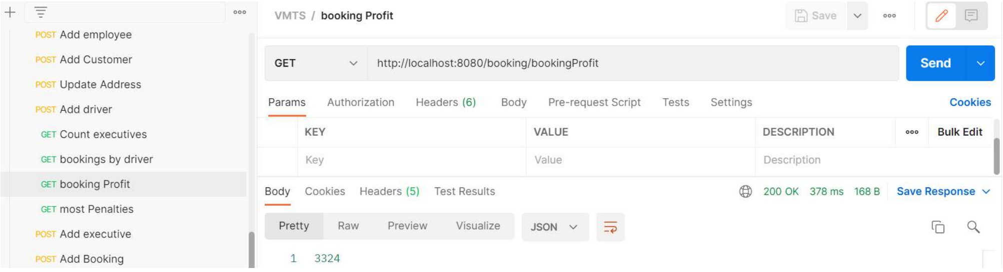
GET http://localhost:8080/booking/settleCount

```
1 3
```

6. Profit from bookings

select (sum(V.RENT_PER_DAY * ((B.END_DATE - B.START_DATE) + 1)) - sum(B.penalty_charge) - sum(F.fuel_filled_amount)) AS 'Total PROFIT'

from VMTS.VEHICLE V, VMTS.Booking B, VMTS.fuel_dtls F \where V.reg_no = B.vehicle_id AND B.booking_status = 'PAID';



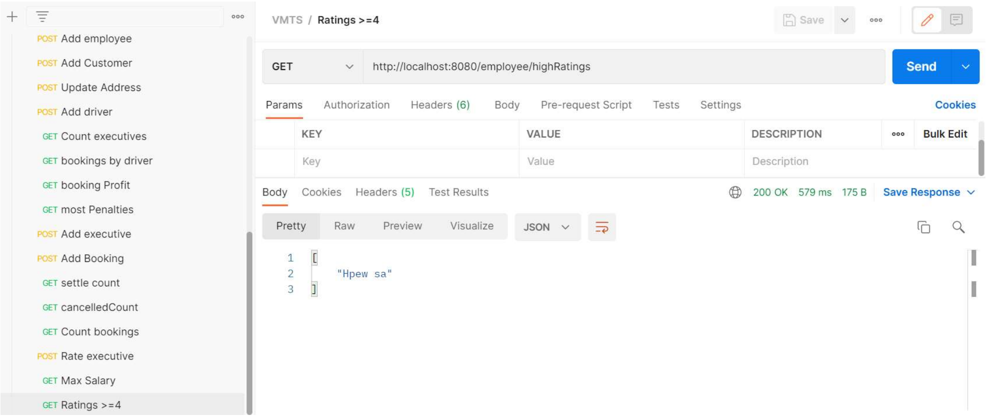
GET http://localhost:8080/booking/bookingProfit

```
1 3324
```

7. Get driver name/executive name with rating '4'

SELECT distinct CONCAT(A.FNAME,' ', A.LNAME) AS Highest_rated_employee FROM VMTS.EMPLOYEE A, VMTS.DRIVER B, VMTS.EXECUTIVE E

WHERE (A.EMP_ID = E.EMP_ID AND E.RATING >= '4') OR (A.EMP_ID = B.EMP_ID AND B.RATING >= '4');



VMTS / Ratings >=4

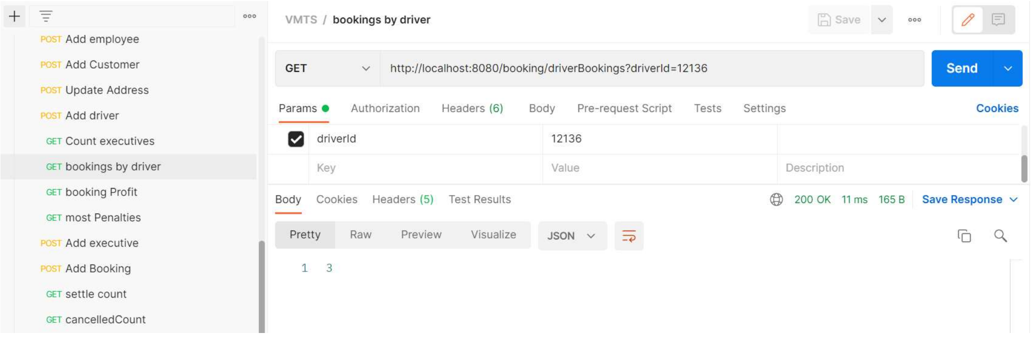
GET http://localhost:8080/employee/highRatings

Body — 200 OK 579 ms 175 B

```
[
    "Hpew sa"
]
```

8. <u>No.of bookings made by specific driver</u>

SELECT COUNT(BOOKING_ID) FROM VMTS.BOOKING WHERE DRIVER_ID = '<ID >'



VMTS / bookings by driver

GET http://localhost:8080/booking/driverBookings?driverId=12136

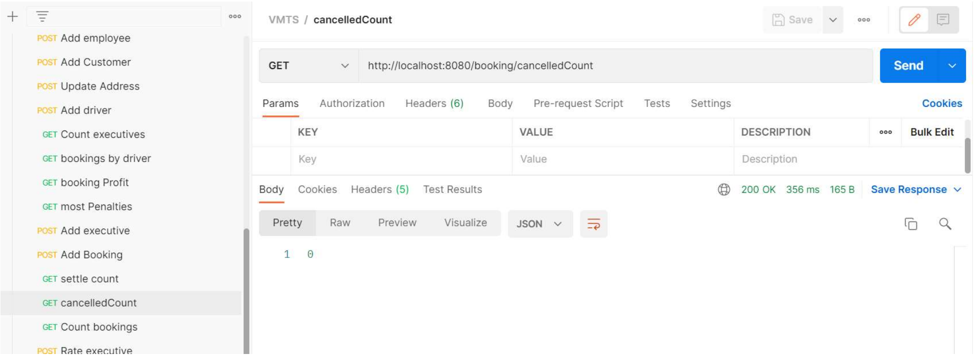
☑ driverId 12136

Body — 200 OK 11 ms 165 B

```
3
```

9. <u>Number of cancelled bookings</u>

SELECT COUNT(*) FROM VMTS.BOOKING WHERE BOOKING_STATUS = 'cancelled';



VMTS / cancelledCount

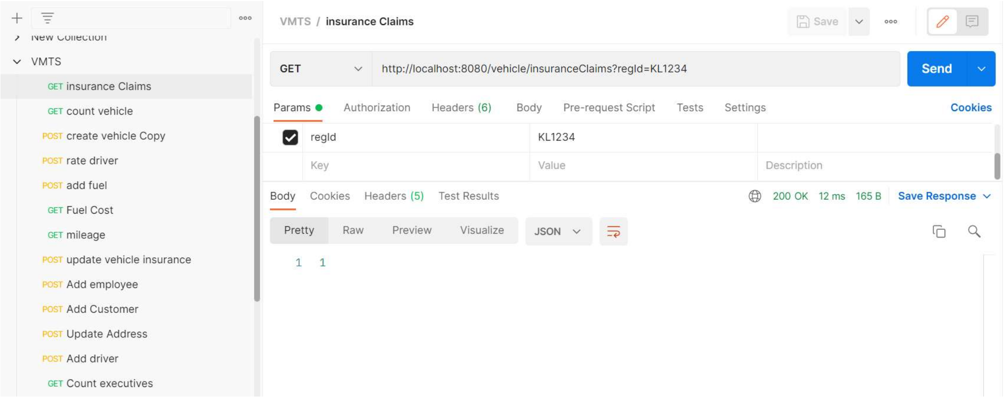
GET http://localhost:8080/booking/cancelledCount

200 OK 356 ms 165 B

```
0
```

10. <u>No.of insurance claims made by vehicle with Reg.No</u>

SELECT A.NO_OF_CLAIMS FROM VMTS.INSURANCE_DTLS A,VMTS.VEHICLE B WHERE B.INSURANCE_NO = A.INSURANCE_ID AND B.REG_NO = <regNo>;



VMTS / insurance Claims

GET http://localhost:8080/vehicle/insuranceClaims?regId=KL1234

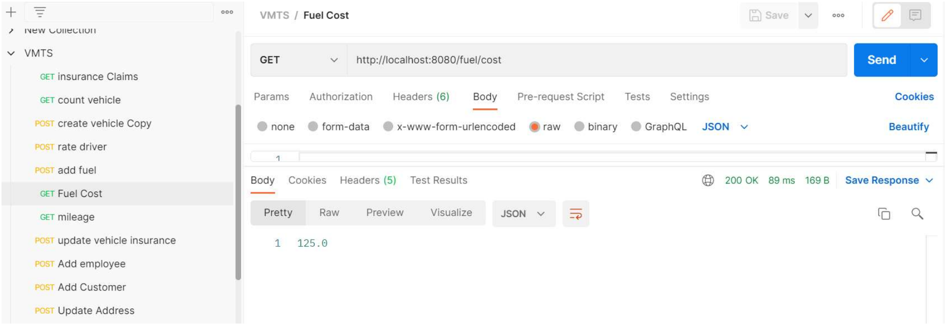
☑ regId KL1234

200 OK 12 ms 165 B

```
1
```

11. <u>Total fuel cost in last month</u>

SELECT SUM(FUEL_FILLED_AMOUNT) FROM VMTS.FUEL_DTLS WHERE DATE_OF_FILLING < current_date() AND DATE_OF_FILLING > current_date()-30;



VMTS / Fuel Cost

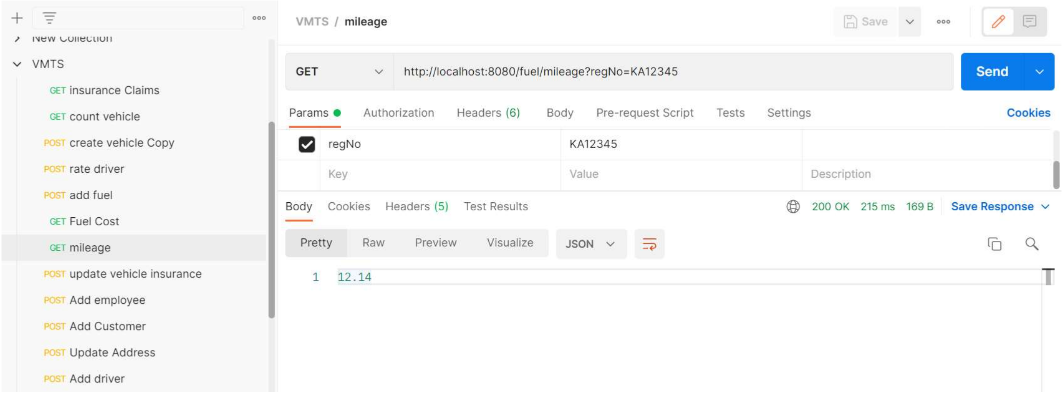
GET http://localhost:8080/fuel/cost

200 OK 89 ms 169 B

```
125.0
```

12. Mileage for a vehicle with reg no (new column added fuel filled)

SELECT round((SUM(b.km_driven)/SUM(f.fuel_filled)),2) AS mileage from vmts.fuel_dtls f, vmts.booking b where f.vehicle_id=b.vehicle_id AND f.vehicle_id = <vehicleId>;



VMTS / mileage

GET http://localhost:8080/fuel/mileage?regNo=KA12345

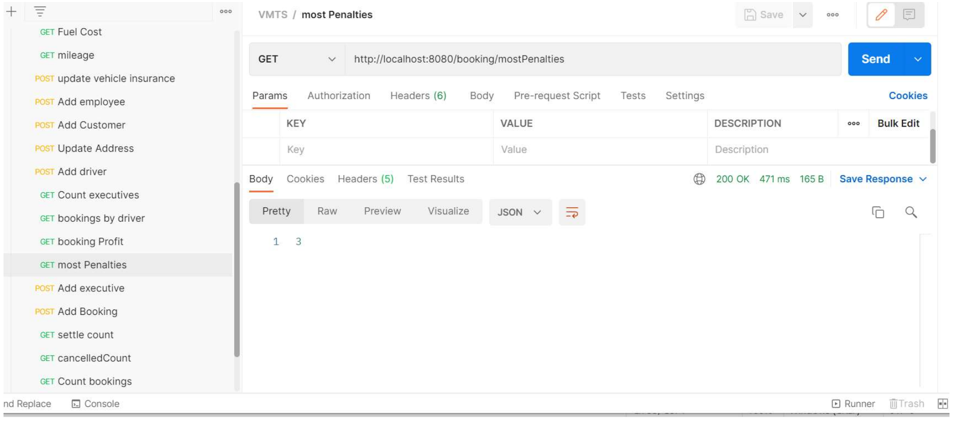
☑ regNo KA12345

200 OK 215 ms 169 B

```
12.14
```

13. Driver involved with more number of penalties

select MAX(count) from (select driver_id, count(penalty_charge) count from vmts.booking where penalty_charge <> 0 group by driver_id) AS Penalty_Count;



VMTS / most Penalties

GET http://localhost:8080/booking/mostPenalties

200 OK 471 ms 165 B

```
1 3
```

8 Appendix

Tools and Technology Used

Below is the list of Tools & Technology Used for creating the VMTS Application

1. MySQL

2. Java

3. Spring Boot

3. Spring data JPA

4. STS

6. Oracle MySQL workbench