

1.

Find the average distance between subsequent stations for every train.

```
MariaDB [pes1ug20cs494]> select train_no, avg(distance) as AVG_DIST from route_info group by train_no;
+-----+-----+
| train_no | AVG_DIST |
+-----+-----+
| 25260    | 277.1667 |
| 25261    | 277.1667 |
| 58450    | 280.3333 |
| 58451    | 279.8333 |
| 62620    | 184.4000 |
| 62621    | 185.0000 |
+-----+-----+
6 rows in set (0.009 sec)
```

2.

Find the average distance between subsequent stations for every train and display them in descending order of distance.

```
MariaDB [pes1ug20cs494]> select train_no, avg(distance) as AVG_DIST from route_info group by train_no order by AVG_DIST desc;
+-----+-----+
| train_no | AVG_DIST |
+-----+-----+
| 58450    | 280.3333 |
| 58451    | 279.8333 |
| 25260    | 277.1667 |
| 25261    | 277.1667 |
| 62621    | 185.0000 |
| 62620    | 184.4000 |
+-----+-----+
6 rows in set (0.006 sec)
```

3.

Display the list of train numbers and the total distance traveled by each in descending order of the distance traveled.

```
MariaDB [pes1ug20cs494]> select train_no, sum(distance) as total_distance from route_info group by train_no order by total_distance desc;
+-----+-----+
| train_no | total_distance |
+-----+-----+
| 62621    | 1850           |
| 62620    | 1844           |
| 58450    | 1682           |
| 58451    | 1679           |
| 25260    | 1663           |
| 25261    | 1663           |
+-----+-----+
6 rows in set (0.001 sec)
```

4.

List those trains that have maximum and minimum number compartments and also display number of compartments they have. (2 queries one to find max and other to find min)

```
MariaDB [pes1ug20cs494]> select train_number, count(compartment_no) as MAX_COMPARTMENT from compartment group by train_number order by
MAX_COMPARTMENT desc limit 1;
+-----+-----+
| train_number | MAX_COMPARTMENT |
+-----+-----+
| 62621 | 5 |
+-----+-----+
1 row in set (0.006 sec)
```

```
MariaDB [pes1ug20cs494]> select train_number, count(compartment_no) as MIN_COMPARTMENT from compartment group by train_number order by
MIN_COMPARTMENT asc limit 1;
+-----+-----+
| train_number | MIN_COMPARTMENT |
+-----+-----+
| 58451 | 2 |
+-----+-----+
1 row in set (0.001 sec)
```

5.

Display the number of phone numbers corresponding to the user_id(s) ADM_001, USR_006, USR_10.

```
MariaDB [pes1ug20cs494]> select user_id, count(phone_no) as NO_PHONE from user_phone where user_id in ("ADM_001", "USR_006", "USR_010"
group by user_id;
+-----+-----+
| user_id | NO_PHONE |
+-----+-----+
| ADM_001 | 2 |
| USR_006 | 2 |
| USR_010 | 2 |
+-----+-----+
3 rows in set (0.023 sec)
```

6.

Find the average fare per km for each train type specified and display the train type and corresponding average fare per km as 'Avg_Fare' in decreasing order of Avg_Fare.

```
MariaDB [pes1ug20cs494]> select Train_Type, avg(fare_per_km) as AVG_FARE_PER_KM from fare group by Train_Type order by AVG_FARE_PER_KM
desc;
+-----+-----+
| Train_Type | AVG_FARE_PER_KM |
+-----+-----+
| Fast | 5.0000 |
| Express | 4.0000 |
| Superfast | 2.0000 |
| Mail | 1.2000 |
+-----+-----+
4 rows in set (0.004 sec)
```

7.

Retrieve all details of the oldest passenger.

```
MariaDB [pes1ug20cs494]> select * from ticket_passenger where age = (select max(age) from ticket_passenger);
+-----+-----+-----+-----+
| seat_no | name   | age  | pnr    |
+-----+-----+-----+-----+
| F01-13  | Ramya R | 45   | PNR012 |
+-----+-----+-----+-----+
1 row in set (0.007 sec)
```

8.

Count the number of passengers whose name consists of 'Ulla'. (Hint: Use the LIKE operator)

```
MariaDB [pes1ug20cs494]> select count(*) as NO_OF_PEOPLE from ticket_passenger where name like "%Ulla%";
+-----+
| NO_OF_PEOPLE |
+-----+
|              4 |
+-----+
1 row in set (0.003 sec)
```