

DBMS

LAB 6

Aggregate Functions

Hita Juneja
Section K
Roll No. 27
11th October, 2022.

1. Find the average distance between subsequent stations for every train.

```
MariaDB [peslug20cs645_railway]> SELECT AVG(distance), train_no FROM route_info GROUP BY train_no;
```

AVG(distance)	train_no
277.1667	25260
277.1667	25261
280.3333	58450
279.8333	58451
184.4000	62620
185.0000	62621

6 rows in set (0.011 sec)

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

```
MariaDB [peslug20cs645_railway]> SELECT AVG(distance), train_no FROM route_info GROUP BY train_no ORDER BY AVG(distance) DESC;
```

AVG(distance)	train_no
280.3333	58450
279.8333	58451
277.1667	25260
277.1667	25261
185.0000	62621
184.4000	62620

6 rows in set (0.011 sec)

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

```
MariaDB [pes1ug20cs645_railway]> SELECT train.train_no, distance FROM train, route_info WHERE
-> from_station_name = source AND to_station_name = destination ORDER BY distance DESC;
+-----+-----+
| train_no | distance |
+-----+-----+
| 58450 | 504 |
| 58451 | 503 |
| 25260 | 481 |
| 25261 | 481 |
| 62621 | 362 |
| 62620 | 361 |
+-----+-----+
6 rows in set (0.006 sec)
```

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

```
MariaDB [pes1ug20cs645_railway]> WITH X AS (SELECT train_number, COUNT(*) as compartment_number FROM compar
tment GROUP BY train_number) SELECT train_number, compartment_number FROM X ORDER BY compartment_number ASC
LIMIT 1;
+-----+-----+
| train_number | compartment_number |
+-----+-----+
| 58451 | 2 |
+-----+-----+
1 row in set (0.004 sec)
```

```
MariaDB [pes1ug20cs645_railway]> WITH X AS (SELECT train_number, COUNT(*) as compartment_number FROM compar
tment GROUP BY train_number) SELECT train_number, compartment_number FROM X ORDER BY compartment_number DES
C LIMIT 1;
+-----+-----+
| train_number | compartment_number |
+-----+-----+
| 62621 | 5 |
+-----+-----+
1 row in set (0.003 sec)
```

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

```
MariaDB [pes1ug20cs645_railway]> SELECT user_id, phone_no FROM user_phone WHERE user_id IN ('ADM_001', 'USR
_006', 'USR_010');
+-----+-----+
| user_id | phone_no |
+-----+-----+
| ADM_001 | 9845012345 |
| ADM_001 | 9900123456 |
| USR_006 | 9845012345 |
| USR_006 | 9900123456 |
| USR_010 | 9845012345 |
| USR_010 | 9900123456 |
+-----+-----+
6 rows in set (0.009 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 [peslug20cs645_railway]> SELECT train_type, AVG(fare_per_km) FROM fare GROUP BY train_type ORDER BY
AVG(fare_per_km) DESC;
+-----+-----+
| train_type | AVG(fare_per_km) |
+-----+-----+
| Fast      | 2.0000           |
| Superfast | 2.0000           |
| Mail      | 1.3333           |
+-----+-----+
3 rows in set (0.004 sec)

```

- Retrieve all details of the oldest passenger.

```

MariaDB [peslug20cs645_railway]> select * from ticket_passenger where Age = (select max(Age) as max_age from
ticket_passenger);
+-----+-----+-----+-----+
| seat_no | name   | age | pnr   |
+-----+-----+-----+-----+
| F01-13 | Ramya R | 45 | PNR012 |
+-----+-----+-----+-----+
1 row in set (0.008 sec)

```

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

```

MariaDB [peslug20cs645_railway]> SELECT COUNT(*) FROM ticket_passenger WHERE name LIKE '%Ullal%';
+-----+
| COUNT(*) |
+-----+
| 4 |
+-----+
1 row in set (0.007 sec)

```