

SQL RAILWAY DATASET ANALYSIS



1 • `create schema railway;`

2 • `select * from railway.railway;`

3

4 #1. Retrieve all transactions where the ticket

5 # price is below 50.

6 • `select * from railway.railway`

7 `where Price < 50;`

8

9 #2. List all unique departure stations.

10 • `select Distinct(`Departure Station`)`

11 `from railway.railway;`

12

13 #3. Count the total number of transactions.

14 • `select count(*) from railway.railway;`

15


```
16 #4. Find all transactions made using contactless payment.
17 • select * from railway.railway
18 where `Payment Method` = 'Contactless';
19
20 #5. Display the transactions where the railcard type is 'Adult'.
21 • select * from railway.railway
22 where Railcard = 'Adult';
23
24 #6. Retrieve all transactions for journeys between
25 # 'London Paddington' and 'Reading'.
26 • select * from railway.railway
27 where `Departure Station` = 'London Paddington' and
28       `Arrival Destination` = 'Reading';
```

```
30 #7. Calculate the total revenue generated from ticket sales.
31 • select sum(Price) as Total_Revenue
32 from railway.railway;
33
34 #8. Find the average ticket price for each ticket type.
35 • select `Ticket Type` , avg(Price) as Avg_Price
36 from railway.railway
37 group by `Ticket Type`;
38
39 #9. Get the count of transactions by payment method.
40 • select `Payment Method`, count(*) as Total_Transactions
41 from railway.railway
42 group by `Payment Method`;
43
```

```
44 #10. Identify journeys delayed due to 'Signal Failure'.
45 • select * from railway.railway
46 where `Reason for Delay` = 'Signal Failure';
47
48 #11. Find the station pair with the highest number of journeys.
49 • select `Departure Station`,`Arrival Destination` ,
50 count(*) as journey_count from railway.railway
51 group by `Departure Station` , `Arrival Destination`
52 order by journey_count desc
53 limit 1;
```



```
55  #12. Determine the percentage of journeys that were delayed.
56 • select `Journey Status` , (count(*) * 100.0 / (select count(*)
57   from railway.railway)) as Delayed_Percentage
58   from railway.railway
59   where `Journey Status` = 'Delayed';
60
61  #13. Find the top 5 stations with the most departures.
62 • select `Departure Station` , count(*) as Counts
63   from railway.railway
64   group by `Departure Station`
65   order by Counts Desc
66   limit 5;
67
```

68 #14. List transactions with a refund request for delayed journeys.

```
69 • select *from railway.railway  
70 where `Journey Status` = 'Delayed'  
71 and `Refund Request` = 'Yes';
```

72

73 #15. Find the ticket class with the highest revenue.

```
74 • select `Ticket Class` , Sum(Price) as Total_Revenue  
75 from railway.railway  
76 group by `Ticket Class`  
77 order by Total_Revenue desc  
78 limit 1;
```

79

```
80  #16. Identify the day with the most ticket purchases.
81 • select `Date of Purchase`, COUNT(*) as Purchase_Count
82 from railway.railway
83 group by `Date of Purchase`
84 order by Purchase_Count desc
85 LIMIT 1;
86
87 #17. Find the railcard type contributing the least to revenue.
88 • select Railcard, sum(Price) as Total_Revenue
89 from railway.railway
90 group by Railcard
91 order by Total_Revenue asc
92 limit 1;
```