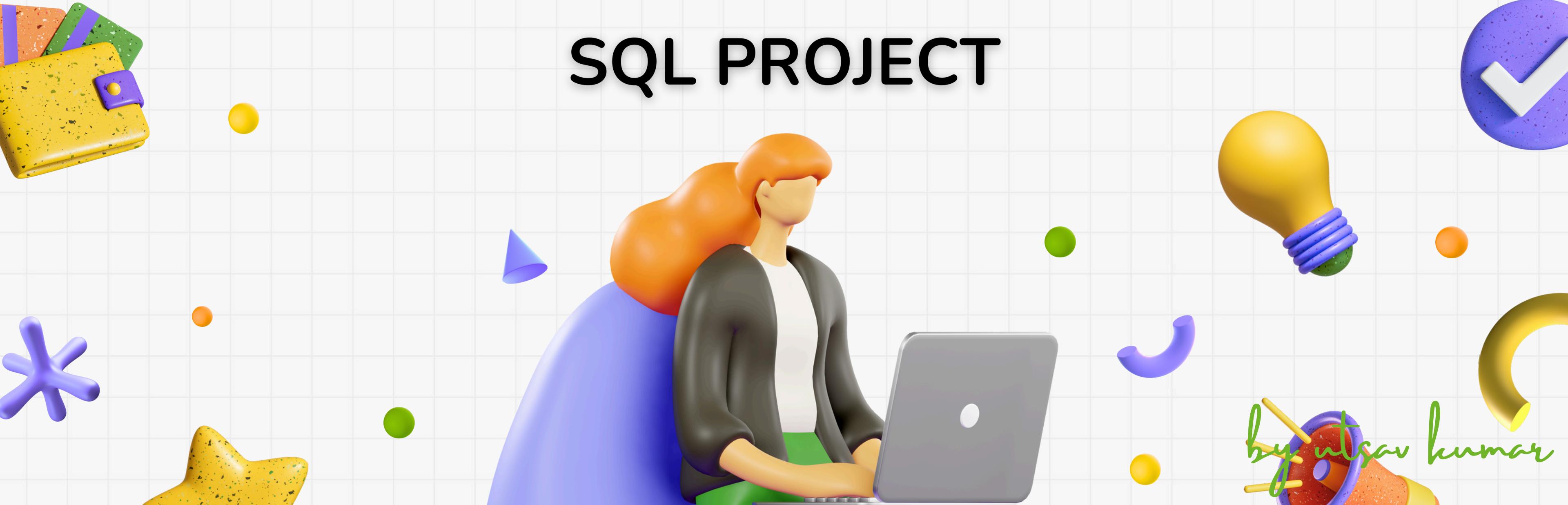


Namma Yatri

Ride-Sharing Data Analysis

SQL PROJECT



Introduction

Hello, My name is Utsav Kumar and
in this project, I have utilized SQL
queries to solve questions related
● to Ride Sharing of Namma Yatri.

total trips

```
.86
.87 -- total trips
.88
.89 • select count(distinct tripid) from trips_details;
.90
.91
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	id	method
▶	1	cash
	2	upi
	3	debit card
	4	credit card

total earnings

The screenshot shows a database interface with the following elements:

- Toolbar:** Includes icons for file, database, lightning bolt, refresh, search, and other database operations.
- Text Area:** Displays numbered steps:
 - 01
 - 02
 - 03 • **select sum(fare) fare from trips;**
 - 04
 - 05
 - 06
- Result Grid:** A table with one row showing the total fare.

	fare
▶	751343
- Buttons:** Result Grid, Filter Rows (with a search bar), Export, and Wrap Cell Content.

total completed trips

```
07 -- total completed trips
08
09
10 • select count(distinct tripid) trips from trips;
11
12
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	trips
▶	983

total searches

```
13
14 -- total searches
15
16 • select sum(searches) searches from trips_details4;
17
18
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

trips	983
-------	-----

total searches which got estimate

```
1° Open a script file in this editor
19 -- total searches which got estimate
20
21 • select sum(searches_got_estimate) searches from trips_details4;
22
23
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

searches
1758

total searches for quotes

```
23
24 -- total searches for quotes
25
26 • select sum(searches_for_quotes) searches from trips_details4;
27
28
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

searches
1455

total driver cancelled

```
32
33
34
35 • select count(*) - sum(driver_not_cancelled) cancelled from trips_details4;
36
37
38
```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

cancelled
1021

total otp entered

```
41 -- total otp entered
42
43 • select sum(otp_entered) otp from trips_details4;
44
45
46
47
```

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content: |

	otp
▶	983

total end ride

```
46
47
48    -- total end ride
49
50 • select sum(end_ride) from trips_details4;
51
52
```

Result Grid			Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:	

	sum(end_ride)
▶	983

average distance per trip

```
55
56 -- average distance per trip
57
58 • select avg(distance) from trips;
59
60
61
```

| Result Grid | Filter Rows: | Export: Wrap Cell Content:

	avg(distance)
▶	14.3927

average fare per trip



```
62  
63 -- average fare per trip  
64
```

```
65 • select avg(fare) from trips;
```

```
66  
67  
68
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

avg(fare)
764.3367

distance travelled

```
69
70
71 -- distance travelled
72
73 • select sum(distance) from trips;
74
75
```

Result Grid | Filter Rows: _____ | Export: Wrap Cell Content:

	sum(distance)
▶	14148

which is the most used payment method

The screenshot shows a MySQL query editor interface with a toolbar at the top and a code editor below. The code editor displays a SQL query with line numbers from 3376 to 3389. The query uses a subquery to count distinct trip IDs for each fare method and then joins it back to the payment table to find the maximum count.

```
3376 -- which is the most used payment method
3377
3378 • SELECT
3379     a.method
3380 FROM
3381     payment a
3382         INNER JOIN
3383     (SELECT
3384         faremethod, COUNT(DISTINCT tripid) cnt
3385     FROM
3386         trips
3387     GROUP BY faremethod
3388     ORDER BY COUNT(DISTINCT tripid) DESC
3389     LIMIT 1) b ON a.id = b.faremethod;
```

At the bottom, there are navigation buttons for 'Result Grid' (highlighted), 'Filter Rows:', 'Export:', and 'Wrap Cell Content:'. A small preview table shows the results:

	method
▶	credit card

the highest payment was made through which instrument

The screenshot shows a MySQL Workbench interface. The query editor window contains the following SQL code:

```
3391    -- the highest payment was made through which instrument
3392 •  SELECT
3393      a.method
3394  FROM
3395    payment a
3396      INNER JOIN
3397    (SELECT
3398      *
3399    FROM
3400      trips
3401    ORDER BY fare DESC LIMIT 1) b ON a.id = b.faremethod
3402 ;
```

The results grid at the bottom shows the following data:

method
credit card

which two locations had the most trips

The screenshot shows a database query editor interface with the following details:

Toolbar: Includes icons for file operations, search, and other common functions, along with a "Limit to 5000 rows" button.

Text Area: Displays the following SQL query:

```
3413
3414 -- which two locations had the most trips
3415
3416 • select * from
3417   (select *,dense_rank() over(order by trip desc) rnk
3418     from
3419     ( select loc_from, loc_to, count(distinct tripid) trip from trips
3420       group by loc_from, loc_to
3421     )a)b
3422   where rnk=1;
3423
3424
```

Result Grid: Shows the results of the query in a grid format:

	loc_from	loc_to	trip	rnk
▶	16	21	5	1
	35	5	5	1

top 5 earning drivers

The screenshot shows a database query editor interface with the following details:

- Toolbar:** Includes icons for file, copy, cut, paste, search, refresh, and other common operations.
- Text Area:** Displays the following SQL query:

```
3425
3426 -- top 5 earning drivers
3427
3428 • select * from
3429   (select *,dense_rank() over(order by fare desc) rnk
3430   from
3431   (select driverid, sum(fare) fare from trips
3432   group by driverid)b)c
3433 where rnk<6;
3434
3435
3436
```
- Result Grid:** A table showing the results of the query:

	driverid	fare	rnk
▶	12	36787	1
	8	30101	2
	21	29787	3
	24	28870	4
	30	28853	5

- Bottom Bar:** Includes buttons for "Result Grid", "Filter Rows:", "Export:", and "Wrap Cell Content:".

which duration had more trips

```
136
137 -- which duration had more trips
138
139 • select * from
140   (select *,rank() over(order by cnt desc) rnk from
141     (select duration, count(distinct tripid) cnt from trips
142      group by duration)b)c
143   where rnk=1;
144
145
```

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

	duration	cnt	rnk
▶	1	53	1

which driver , customer pair had more orders

```
143
144 -- which driver , customer pair had more orders
145
146 • select * from
147   (select *, rank() over(order by cnt desc) rnk from
148     (select driverid, custid, count(distinct tripid) cnt from trips
149      group by driverid, custid)c)d
150   where rnk=1 ;
151
```

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

	driverid	custid	cnt	rnk
▶	17	96	4	1
	28	15	4	1

search to estimate rate

```
151
152
153 -- search to estimate rate
154
155
156 • select sum(searches_got_estimate)*100/sum(searches) Rate
157   from trips_details4;
158
159
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Rate
▶	81.3512

which area got the highest trips in which duration

```
| 170  
| 171    -- which area got highest trips in which duration  
| 172  
| 173 • select * from |  
| 174     (select *, rank() over(partition by duration order by cnt desc ) rnk from |  
| 175       (select duration, loc_from, count(distinct tripid) cnt  
| 176         from trips group by duration, loc_from) a ) c  
| 177   where rnk=1 ;  
| 178  
| 179
```

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content: |

	duration	loc_from	cnt	rnk
▶	1	20	4	1
	2	9	5	1
	3	13	3	1
	3	22	3	1
	3	34	3	1
	4	4	3	1

Result 16 ×

which area got the highest fares

The screenshot shows a database query editor interface with the following details:

- Toolbar:** Includes icons for file, database, refresh, search, and other common operations.
- Text Area:** Displays the following SQL query:

```
492
493
494 • Ⓜ select * from (select *,rank() over(order by fare desc) rnk
495   from
496     ( select loc_from, sum(fare) fare from trips
497      group by loc_from) b )c
498   where rnk=1;
499
500
501
502
```
- Result Grid:** Shows the output of the query in a grid format.

	loc_from	fare	rnk
▶	6	30295	1
- Bottom Bar:** Includes buttons for "Result Grid", "Filter Rows:", "Export:", and "Wrap Cell Content:".
- Status Bar:** Shows a "Res Grid" icon.

which area got the highest cancellations by driver

The screenshot shows a database query editor interface with the following details:

- Toolbar:** Includes icons for file, database, search, and other common operations.
- Text Area:** Displays the following SQL query:

```
501
502
503 • select * from (select *, rank() over(order by can desc) rnk
504   from
505   (select loc_from, count(*) - sum(driver_not_cancelled) can
506     from trips_details4
507    group by loc_from)b)c
508  where rnk =1;
509
510
511
```
- Result Grid:** A table showing the results of the query:

	loc_from	can	rnk
▶	1	43	1
- Bottom Bar:** Includes buttons for "Result Grid", "Filter Rows:", "Export:", and "Wrap Cell Content:".
- Right Sidebar:** A blue sidebar labeled "Result Grid".

which area got the highest cancellations by customer

The screenshot shows a database query editor interface with the following details:

- Toolbar:** Includes icons for file, copy, cut, paste, search, refresh, and other common database operations.
- Text Area:** Displays the following SQL query:

```
510
511
512 • Ⓜ select * from (select *, rank() over(order by can desc) rnk
513   from
514     (select loc_from, count(*) - sum(customer_not_cancelled) can
515      from trips_details4
516      group by loc_from)b)c
517   where rnk =1;
518
519
520
```
- Result Grid:** A table showing the results of the query:

	loc_from	can	rnk
▶	4	40	1
- Bottom Bar:** Includes buttons for "Result Grid" (highlighted in blue), "Filter Rows:", "Export:", and "Wrap Cell Content:".

which duration got the highest fares

The screenshot shows a database query editor interface with the following details:

- Toolbar:** Includes icons for file, copy, cut, paste, search, refresh, and other common database operations.
- Text Area:** Displays the following SQL query:

```
519
520
521
522 • select * from (select *,rank() over(order by fare desc) rnk
523   from
524   ( select duration, sum(fare) fare from trips
525     group by duration) b )c
526   where rnk=1;
527
528
529
```
- Result Grid:** A table showing the results of the query:

	duration	fare	rnk
▶	1	45019	1
- Bottom Bar:** Includes buttons for "Result Grid" (highlighted), "Filter Rows", "Export", and "Wrap Cell Content".
- Right Sidebar:** Shows a large blue arrow pointing right with the text "Result Grid" below it.

which duration got the highest trips

The screenshot shows a database query editor interface with the following details:

Toolbar: Includes icons for file, copy, cut, paste, search, refresh, and other common database operations. A "Limit to 5000 rows" button is also present.

Text Area: Displays the following SQL query:

```
527
528 • select * from (select *,rank() over(order by trip desc) rnk
529   from
530   ( select duration, count(distinct tripid) trip from trips
531     group by duration) b )c
532   where rnk=1;
533
534
```

Result Grid: Shows the results of the query in a tabular format:

	duration	trip	rnk
▶	1	53	1

Bottom Bar: Includes buttons for "Result Grid" (selected), "Filter Rows:", "Export:", and "Wrap Cell Content:". A vertical toolbar on the right side contains icons for "Result Grid" and "Form Editor".

thank
you!

