

# Uber Data Project

You're given 2 datasets: **uber\_driver\_trips** and **uber\_driver\_info**. There are 2 ways to access this data.

1. Go to any question on stratascratch.com and type `select \* from uber\_driver\_trips` and `select \* from uber\_driver\_info` in the code editor.
2. Upload the files into a database. Instructions can be found on [ICA#5](#) and the dataset files can be found in the Files section on Canvas.

Answer the questions below to complete the data project. Provide the code in the blank space provided and copy and paste the output below the code, either as an image or table.

## Week 1: Introduction to Databases and MySQL Basics

### Basic SQL Syntax

1. Write a basic SQL query to retrieve all data from one of the datasets

```
select
*
from uber_driver_trips
```

name	trips_completed	accept_rate	supply_hours	rating
Abdul	1	1	3	4.8
Abraham	12	0.83	5	4.7
Adelina	1	1	2	4.7
Akilah	1	1	2	4.9
Alec	21	0.76	11	5
Alesha	7	1	4	4.8
Alvaro	17	0.88	11	4.6
Andra	16	0.94	11	4.6
Augusta	19	0.84	11	4.7
Aurora	10	0.9	4	4.6
Buffy	13	0.54	6	5

2. Write a basic SQL query to retrieve only the `name` and `rating` columns from the dataset.

```
select
  name,
  rating
from uber_driver_trips
```

name	rating
Abdul	4.8
Abraham	4.7
Adelina	4.7
Akilah	4.9
Alec	5
Alesha	4.8
Alvaro	4.6
Andra	4.6
Augusta	4.7
Aurora	4.6
Buffy	5
Byron	4.9

3. Retrieve all drivers who completed more than 10 trips.

```
select
  name,
  trips_completed
from uber_driver_trips
where trips_completed > 10
```

name	trips_completed
Abraham	12
Alec	21
Alvaro	17
Andra	16
Augusta	19
Buffy	13
Byron	15
Carlota	14
Chu	14
Corine	16
Dannette	14
Deane	22

- Retrieve all drivers who have an acceptance rate of less than 90%.

```
select
  name,
  accept_rate
from uber_driver_trips
where accept_rate < 0.90
```

name	accept_rate
Abraham	0.83
Alec	0.76
Alvaro	0.88
Augusta	0.84
Buffy	0.54
Chu	0.71
Corine	0.63
Cris	0.71
Dalila	0.57
Dannielle	0.67
Deane	0.77

## Advanced SELECT Statements

5. Retrieve the **name** and **rating** for all drivers with a rating greater than 4.5

```
select
  name,
  rating
from uber_driver_trips
where rating > 4.5
```

name	rating
Abdul	4.8
Abraham	4.7
Adelina	4.7
Akilah	4.9
Alec	5
Alesha	4.8
Alvaro	4.6
Andra	4.6
Augusta	4.7
Aurora	4.6
Buffy	5
Byron	4.9

6. Retrieve the **name**, **Trips Completed**, and **Supply Hours** for all drivers.

```
select
  name,
  trips_completed,
  supply_hours
from uber_driver_trips
```

name	trips_completed	supply_hours
Abdul	1	3
Abraham	12	5
Adelina	1	2
Akilah	1	2
Alec	21	11
Alesha	7	4
Alvaro	17	11
Andra	16	11
Augusta	19	11
Aurora	10	4
Buffy	13	6
Byron	15	10

## Using LIKE, IN, and BETWEEN

7. Retrieve all drivers with an acceptance rate between 80% and 90%

```
select
  name,
  accept_rate
from uber_driver_trips
where accept_rate between 0.80 and 0.90
```

name	accept_rate
Abraham	0.83
Alvaro	0.88
Augusta	0.84
Aurora	0.9
Domenica	0.89
Dustin	0.85
Fernande	0.86
Hee	0.89
Lamonica	0.88
Lianne	0.82
Lilla	0.89
Loree	0.89

8. Retrieve all drivers whose **name** contains the letter 'A' (use the LIKE operator).

```
select
  name
from uber_driver_trips
where name like '%a%'
```

name
Abdul
Abraham
Adelina
Akilah
Alec
Alesha
Alvaro
Andra
Augusta
Aurora
Carlota
Celesta

## Sorting and Limiting Results

9. Retrieve the top 5 drivers with the highest number of completed trips.

```
select
  name,
  trips_completed
from uber_driver_trips
order by trips_completed desc
limit 5
```

name	trips_completed
Michelle	31
Deneen	26
Lamonica	24
Elton	23
Man	23

10. Retrieve the top 3 drivers with the highest rating.

```
select
  name,
  rating
from uber_driver_trips
order by rating desc
limit 3
```

name	rating
Dannielle	5
Marica	5
Cris	5

## Week 2: Data Types and Functions

## String Functions

1. Find the length of the `name` for each driver.

```
select
  name, length(name) as name_length
from uber_driver_trips
```

name	name_length
Abdul	5
Abraham	7
Adelina	7
Akilah	6
Alec	4
Alesha	6
Alvaro	6
Andra	5
Augusta	7
Aurora	6
Buffy	5
Byron	5

2. Concatenate the `name` with the string ' - Uber' and display it as a new column named `uber_driver_info`.

```
select
  concat (name, ' - Uber') as uber_driver_info
from uber_driver_trips
```



uber_driver_info
Abdul - Uber
Abraham - Uber
Adelina - Uber
Akilah - Uber
Alec - Uber
Alesha - Uber
Alvaro - Uber
Andra - Uber
Augusta - Uber
Aurora - Uber
Buffy - Uber
Byron - Uber

## Week 3: Aggregate Functions, Grouping Data, Logical Statements

### Aggregate Functions

1. Calculate the total number of trips completed by all drivers.

```
select
  sum(trips_completed) as total_trips_completed
from uber_driver_trips
```

total\_trips\_completed

1421

2. Calculate the average rating of all drivers.

```
select
  avg(rating) as _avg_rating
from uber_driver_trips
```

\_avg\_rating

4.699

## GROUP BY Clause

3. Calculate the total number of trips completed by drivers with the same rating.

```
select
  rating,
  count(trips_completed) as total_trips_completed
from uber_driver_trips
group by rating
order by rating desc
```

rating	total_trips_completed
5	20
4.9	27
4.8	17
4.7	18
4.6	9
4.55	1
4.5	18
4.38	1
4.3	5
3.2	1
3	1
1.6	1

4. Calculate the average acceptance rate for drivers with the same rating.

```
select
  rating,
  avg(accept_rate) as avg_accept_rate
from uber_driver_trips
group by rating
order by rating desc
```

rating	avg_accept_rate
5	0.85
4.9	0.89
4.8	0.73
4.7	0.88
4.6	0.86
4.55	0.36
4.5	0.89
4.38	0.8
4.3	0.84
3.2	1
3	0.86
1.6	1

## HAVING Clause

5. Find the ratings that have more than 5 total trips completed.

```
select
  rating,
  count(trips_completed) as total_trips_completed
from uber_driver_trips
group by rating
```

```
having count(trips_completed) > 5
order by rating desc
```

rating	total_trips_completed
5	20
4.9	27
4.8	17
4.7	18
4.6	9
4.5	18

6. Find the ratings with an average acceptance rate greater than 80%.

```
select
  rating,
  avg(accept_rate) as avg_accept_rate
from uber_driver_trips
group by rating
having count(accept_rate) > 0.80
order by rating desc
```

rating	avg_accept_rate
5	0.85
4.9	0.89
4.8	0.73
4.7	0.88
4.6	0.86
4.55	0.36
4.5	0.89
4.38	0.8
4.3	0.84
3.2	1
3	0.86
1.6	1

## Case Statements

7. Categorize drivers as 'High', 'Medium', or 'Low' based on their rating.

- A rating of 4.7 or greater should be labeled as “high”
- A rating of 4.0 or greater should be labeled as “medium”
- Else, the rating should be labeled as “low”

```
select
  name,
  rating,
  case when rating >= 4.7 then 'High'
        when rating >= 4.0 then 'Medium'
        else 'Low'
  end as rating_category
from uber_driver_trips
```

name	rating	rating_category
Abdul	4.8	High
Abraham	4.7	High
Adelina	4.7	High
Akilah	4.9	High
Alec	5	High
Alesha	4.8	High
Alvaro	4.6	Medium
Andra	4.6	Medium

8. Categorize drivers based on the number of trips completed into 'Active', 'Moderately Active', and 'Inactive'.
- If the number of trips completed is 10 or greater then label “active”
  - If the number of trips completed is 5 or greater then label as “moderately active”
  - Else, label as “inactive”

```
select
  name,
  trips_completed,
  case when trips_completed >= 10 then 'active'
        when trips_completed >= 5 then 'moderately active'
        else 'inactive'
  end as rating_category
from uber_driver_trips
```

name	trips_completed	rating_category
Abdul	1	inactive
Abraham	12	active
Adelina	1	inactive
Akilah	1	inactive
Alec	21	active
Alesha	7	moderately active
Alvaro	17	active
Andra	16	active

## Week 4: Joins and Subqueries

### JOIN Operations

1. Find the drivers with the highest rating and compare their **Accept Rate** with the average **Accept Rate** of all other drivers.

```
select
  a.name,
  a.rating,
  a.accept_rate as highest_rated_accept_rate,
  avg(b.accept_rate) as avg_accept_rate
from uber_driver_trips a
join uber_driver_trips b on a.rating = b.rating
where a.rating = (select max(rating) from uber_driver_trips)
group by
  a.name,
  a.rating,
  a.accept_rate
```

name	rating	highest_rated_accept_rate	avg_accept_rate_others
Yessenia	5	0.88	0.85
Tenisha	5	0.88	0.85
Stephania	5	0.8	0.85
Son	5	0.67	0.85
Samella	5	0.44	0.85

2. Retrieve a list of all drivers and their **Accept Rate** from `uber_driver_trips`, and for each driver, retrieve their name and age.

```
select
  a.name,
  a.accept_rate,
  driver_age
from uber_driver_trips a
join uber_driver_info b on a.name = b.driver_name
```

name	accept_rate	driver_age
Abdul	1	30
Abraham	0.83	31
Adelina	1	32
Akilah	1	33
Alec	0.76	34

Subqueries



3. Retrieve drivers who have a rating higher than the average rating of all drivers.

```
select
  name,
  rating,
  (select avg(rating) from uber_driver_trips) as avg_rating
from uber_driver_trips
where rating > (select avg(rating) from uber_driver_trips)
```

name	rating	avg_rating
Abdul	4.8	4.7
Abraham	4.7	4.7
Adelina	4.7	4.7
Akilah	4.9	4.7
Alec	5	4.7
Alesha	4.8	4.7

4. Retrieve drivers who have completed more trips than the average number of trips completed.

```
select
  name,
  trips_completed,
  (select avg(trips_completed) from uber_driver_trips) as avg_trips_completed
from uber_driver_trips
where trips_completed > (select avg(trips_completed) from uber_driver_trips)
```

name	trips_completed	avg_trips_completed
Abraham	12	11.94
Alec	21	11.94
Alvaro	17	11.94
Andra	16	11.94
Augusta	19	11.94
Buffy	13	11.94

5. Retrieve the names and ages of drivers from the `uber_driver_info` table who have a rating higher than the average rating of all drivers.

```
select
  i.driver_name,
  i.driver_age,
  t.rating,
  (select avg(t.rating)
   from uber_driver_trips t
   where t.name = t.name) AS avg_rating_others
from uber_driver_info i
join uber_driver_trips t on i.driver_name = t.name
where t.rating > (select avg(rating) from uber_driver_trips)
```

driver_name	driver_age	rating	avg_rating_others
Abdul	30	4.8	4.7
Abraham	31	4.7	4.7
Adelina	32	4.7	4.7
Akilah	33	4.9	4.7
Alec	34	5	4.7

## CTEs

6. Create a CTE to find the total number of hours worked by drivers with a rating above 4.7.

```
with driver_rating_above_4_7 as
(select
  name,
  supply_hours
from uber_driver_trips
where rating > 4.7)
select
  sum(supply_hours) as total_hours_worked
from driver_rating_above_4_7
```

total\_hours\_worked

489

7. Create a CTE to find the average acceptance rate of drivers who completed more than 10 trips.

```
with driver_avg_accept_rate_over_10_trips as
(select
  name,
  accept_rate
from uber_driver_trips
where trips_completed > 10)
select
  avg(accept_rate) as avg_accept_rate
from driver_avg_accept_rate_over_10_trips
```

avg\_accept\_rate

0.82

8. Create a CTE to find the average age of drivers (from `uber_driver_info`) who have completed more than 15 trips.

```

with avg_age_over_15_trips as
(select
  driver_name,
  driver_age
from uber_driver_info
join uber_driver_trips
where trips_completed > 15)
select
  avg(driver_age) as avg_driver_age
from avg_age_over_15_trips

```

avg\_driver\_age

34.46

## Week 5:

Using the Uber tables you created from ICA #5, answer the questions below

### Free Response Assignment

Scenario: It is going to be a huge Saturday and there will need to be many more cars on the road than last week. In order to get drivers to go online, we're assessing the following two bonus options in terms of cost:

- Option 1: \$50 for each driver that is online at least 8 hours, accepts 90% of requests, completes 10 trips, and has a rating of 4.7 or better during the time frame;
- Option 2: \$4/trip for all drivers who complete 12 trips, and have a 4.7 or better rating.

Using the dataset provided and given the scenario, provide answers to the questions below:

- How much would the total bonus payout be with Option 1?
- How much would the total bonus payout be with Option 2?
- How many drivers would qualify for a bonus under Option 1 but not under Option 2?
- What percentages of drivers online completed less than 10 trips, had an acceptance rate of less than 90%, and had a rating of 4.7 or higher?

#### 1. How much would the total bonus payout be with Option 1?

**Criteria for Option 1:**

- Online at least 8 hours
- Accepts 90% of requests
- Completes 10 trips
- Has a rating of 4.7 or better

```
select
  count(*) * 50 as total_bonus_payout_option_1
from uber_driver_trips
where supply_hours >= 8
  and accept_rate >= 0.90
  and trips_completed >= 10
  and rating >= 4.7
```

total\_bonus\_payout\_option\_1

1050

This query calculates the total number of drivers who meet the criteria for Option 1 and multiplies that count by \$50 to get the total bonus payout.

## 2. How much would the total bonus payout be with Option 2?

### Criteria for Option 2:

- Completes 12 trips
- Has a rating of 4.7 or better

```
select
  count(*) * 4 as total_bonus_payout_option_2
from uber_driver_trips
where trips_completed >= 12
  and rating >= 4.7
```

total\_bonus\_payout\_option\_2

184

This query calculates the total number of trips completed by drivers who meet the criteria for Option 2 and multiplies that by \$4 to get the total bonus payout.

## 3. How many drivers would qualify for a bonus under Option 1 but not under Option 2?

```

select
  count(*) as drivers_qualify_option_1_not_option_2
from uber_driver_trips
where
  supply_hours >= 8
  and accept_rate >= 0.90
  and rating >= 4.7
  and name not in
    (select
      name
    from uber_driver_trips
    where trips_completed >= 12
      and rating >= 4.7)

```

drivers\_qualify\_option\_1\_not\_option\_2

2

This query counts the number of drivers who meet the criteria for Option 1 but do not meet the criteria for Option 2.

**4. What percentages of drivers online completed less than 10 trips, had an acceptance rate of less than 90%, and had a rating of 4.7 or higher?**

```

select
  (count(*) * 100.0 /
   (select count(*)
    from uber_driver_trips)) as
  percentage_less_10_trips_acceptance_less_90_rating_4_7_higher
from uber_driver_trips
where trips_completed < 10
  and accept_rate < 0.90
  and rating >= 4.7

```

percentage\_less\_10\_trips\_acceptance\_less\_90\_rating\_4\_7\_higher

10.92

This query calculates the percentage of drivers who completed less than 10 trips, had an acceptance rate of less than 90%, and had a rating of 4.7 or higher among all drivers who were online.

