Memberships

How many different kinds of memberships are there?

Query 1

```
SELECT DISTINCT membership_name FROM _________;
```

Are there any rentals in the system missing a membership?

Query 2

```
SELECT COUNT(*) FROM __________ WHERE _______________ IS NULL;
```

How many rentals were there per membership?

Query 3a

```
SELECT COUNT(___) FROM _____________ GROUP BY membership_name;
```

Query 3b

```
_____ COUNT(___), membership_name FROM _____________ GROUP __ membership_name;
```

Memberships are unique per person. So one person with a monthly membership could have been responsible for all of the monthly membership rentals.

How many distinct user memberships are there?

Query 4

```
_____ COUNT(DISTINCT user_id, membership_name) as total_memberships FROM ___;
```

Ok, so how many distinct user memberships are there per membership (group them)?

Query 5

```
_____ COUNT(DISTINCT _____, ________), membership_name
FROM _____ GROUP BY _____________;
```

Explain the difference between query 3b and query 5. What do they represent? How are the results different? Why?

Ride Dates (Based on Start Time)

What was the earliest date of a rental?

Query 1

```
SELECT MIN(_________) FROM __________;
```

What was the latest date of a rental?

Query 2

```
SELECT ___(_________) FROM __________;
```

Which rentals occurred in 2016?

Query 3

```
______ * _____ rides WHERE start_time < '2017';
```

How many rentals occurred in 2016?

Query 4

```
_______  _______(*) FROM _______ WHERE __________ < '2017';
```

How many rentals occurred per year?

Query 5

```
_______ _______(*), YEAR(start_time) as year ______ _____ GROUP BY year;
```

What is strange about the results of that query? Why do you see NULL?

How many records are missing a start_time?

Query 6

```
______ COUNT(*) ______ _____ WHERE ______ IS NULL;
```

Is there anything else we notice about the rows where start_time is missing?

Query 7

```
______ * FROM _____ WHERE _______ __ _____;
```

How many rentals were there per month?

Query 8

```
_____ COUNT(*), YEAR(start_time) as year, MONTH(start_time) as month
____ ____ GROUP BY year, month;
```

Challenge: Write a query to display the number of rentals per day. Then, by hour.

Ride Dates (Based on Start Time)

How long (in minutes) was each ride, from longest to shortest?

Query 1

```
_____ start_time, end_time,
      TIMESTAMPDIFF(MINUTE, start_time, end_time) as ride_length

_____ _____
WHERE start_time IS NOT NULL AND end_time IS NOT NULL
ORDER BY ride_length DESC;
```

How long (in minutes) was each ride, from shortest to longest?

Query 2

(Same as **Query 1**, but change one thing.)

When viewing the shortest rides, you should notice something interesting. What do you notice about a large number of the shortest rides? (How long are they?) Explain why you believe many of the rides have the length of time that they do. (Notice the start_time and end_time values.)

How many times was a bike ridden for particular lengths of time?

Query 3

```
_____ COUNT(*), TIMESTAMPDIFF(MINUTE, start_time, end_time) as ride_length
_____ ____
_____ start_time IS NOT NULL ___ end_time IS NOT ___
_____ BY ride_length ORDER __ ride_length DESC;
```

The *frequency* of a ride length is the number of times a bike was rented for a particular length of time. Try the query again, but order it in a manner that helps you see the frequency of each length of time.

How many times was a bike ridden for particular lengths of time, in order of frequency?

Query 4

```
______ COUNT(*) as frequency,
       TIMESTAMPDIFF(MINUTE, start_time, end_time) as ride_length
____ rides
WHERE start_time IS NOT NULL AND end_time IS NOT NULL
GROUP BY ride_length ORDER BY _______ DESC;
```

Think about some of the very long rides. Do you have an explanation?
We call such examples that are "far outside the norm" potential *anomalies* or *outliers*.
More approachable information on outliers/anomalies here, here and here.