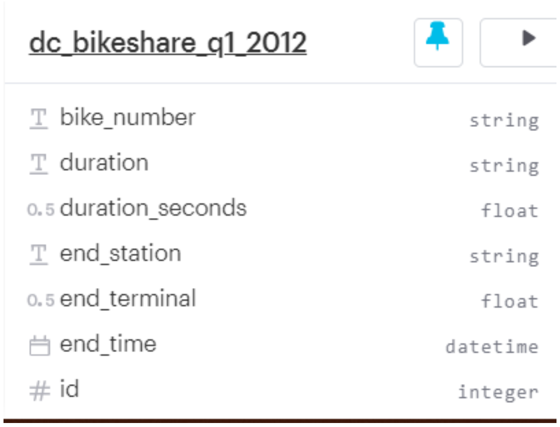
Project on “dc_bikeshare_q1_2012” Dataset on mode

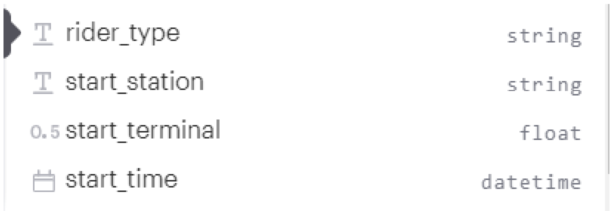
1. Short summary of the data

a. Fields and their respective data types



dc_bikeshare_q1_2012

bike_number	string
duration	string
duration_seconds	float
end_station	string
end_terminal	float
end_time	datetime
id	integer



rider_type	string
start_station	string
start_terminal	float
start_time	datetime

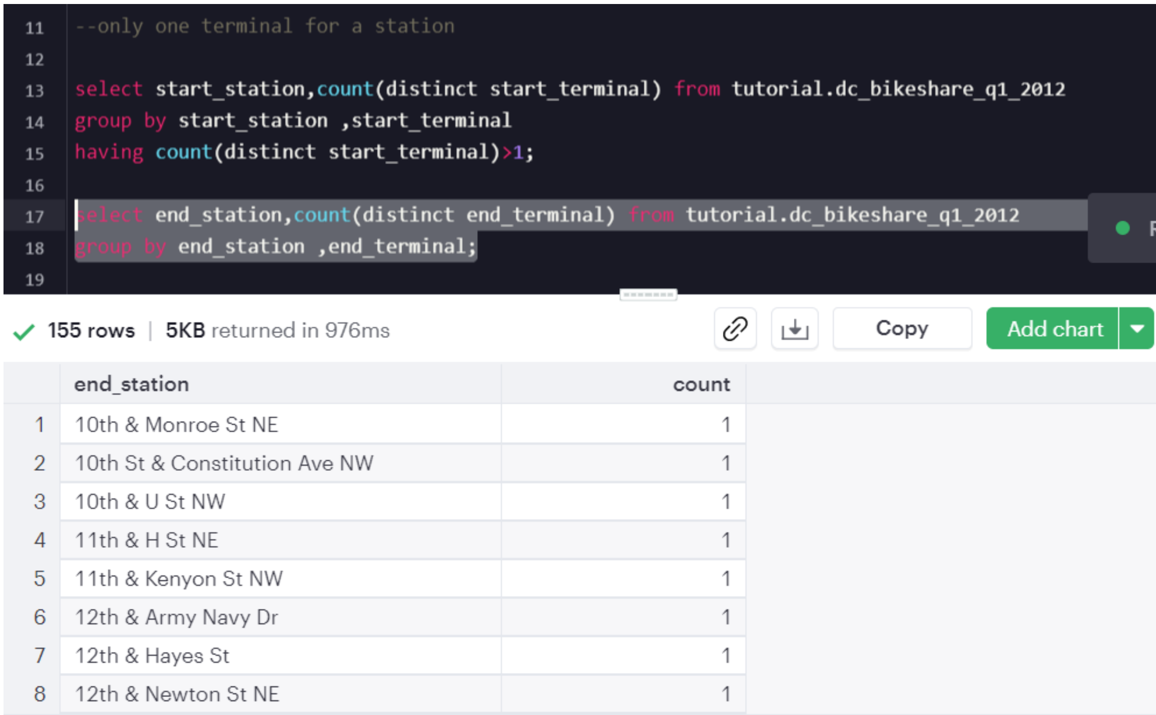
b. What do they represent

- ID: Primary Key (Surrogate Key) which is unique for each row → ID is not null for any field
- The data set represents data for a bike sharing service, whose customer base is divided into two types:
  - Registered Riders
  - Casual Riders
- Duration of a ride in hh:mm:ss format
- Duration of a ride in seconds
- Start Time: Date time, time stamp of when the ride is started
- Start Station: Name of the Station from where the ride is started
- Start Terminal Number

- End Time: Date time, time stamp of when the ride ends
- End Station: Name of the Station from where the ride ends
- End Terminal Number

c. What can the data be used for

- To find out which type of riders use the service more
- To find out which type of riders take longer rides because longer ride = more revenue
- Which bike number has done more ride duration → meaning that bike may need service?
- To find out which stations have most riders
- To find out what are the busiest times where the service is most used
- Which terminal of a station is more used?
    - Checked → there is only one terminal per station



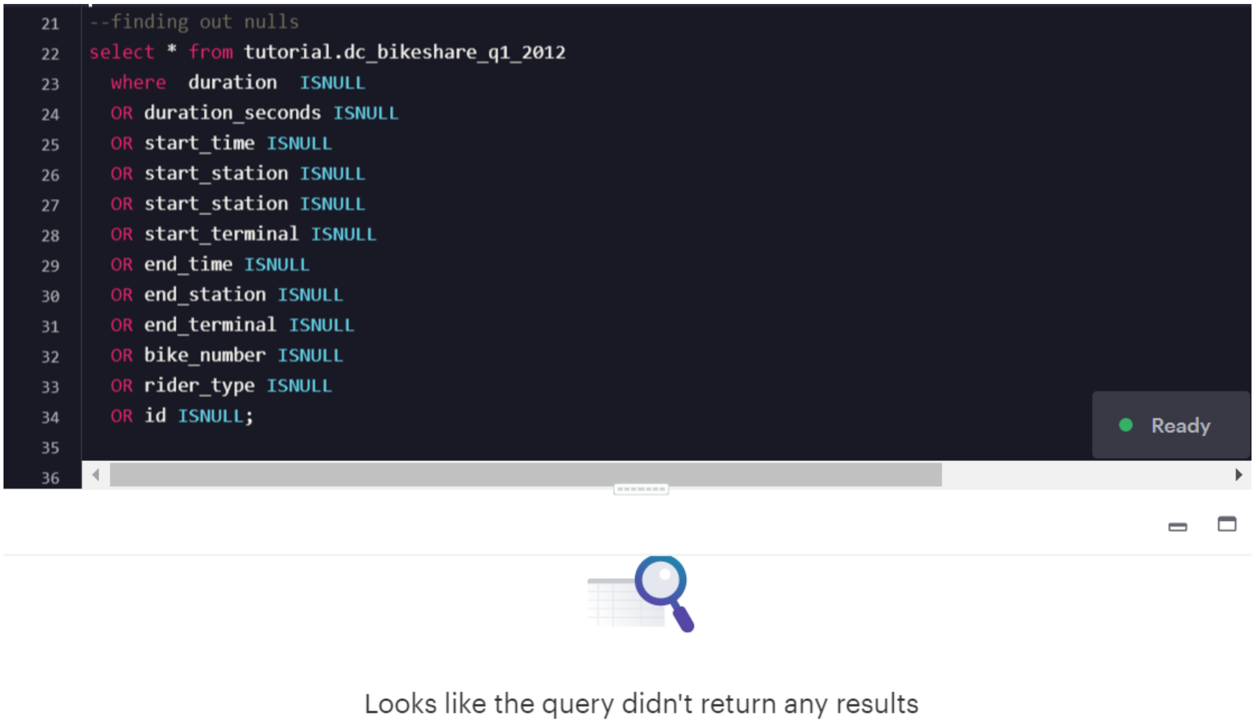
```sql
--only one terminal for a station

select start_station,count(distinct start_terminal) from tutorial.dc_bikeshare_q1_2012
group by start_station ,start_terminal
having count(distinct start_terminal)>1;

select end_station,count(distinct end_terminal) from tutorial.dc_bikeshare_q1_2012
group by end_station ,end_terminal;
```

✓ 155 rows | 5KB returned in 976ms

	end_station	count
1	10th & Monroe St NE	1
2	10th St & Constitution Ave NW	1
3	10th & U St NW	1
4	11th & H St NE	1
5	11th & Kenyon St NW	1
6	12th & Army Navy Dr	1
7	12th & Hayes St	1
8	12th & Newton St NE	1

d. Checking for null values

There are no nulls.



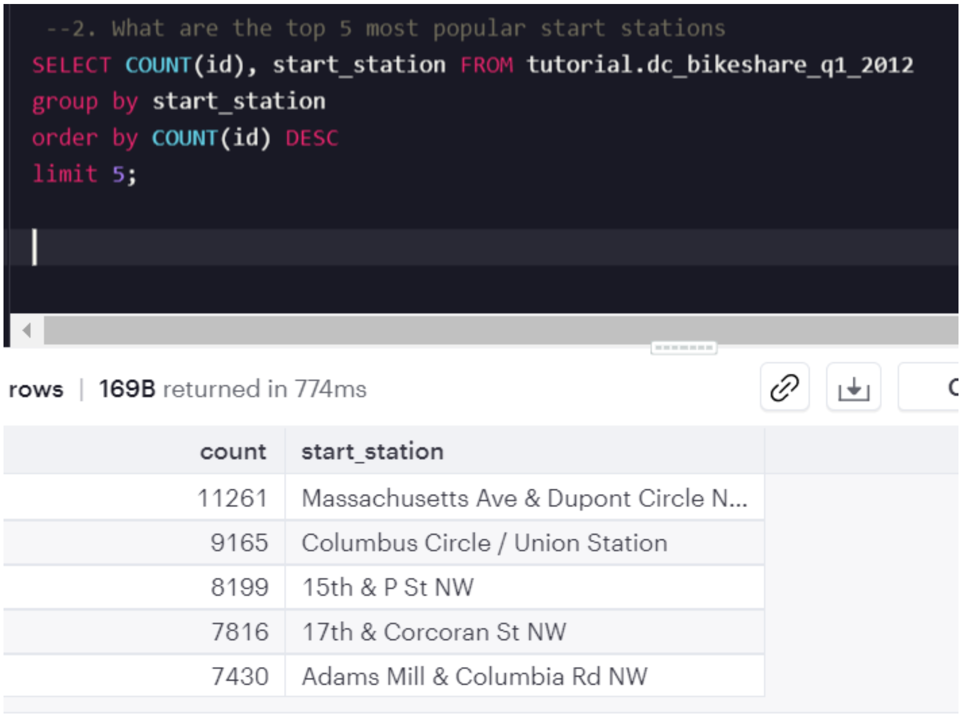
```
--finding out nulls
select * from tutorial.dc_bikeshare_q1_2012
  where  duration  ISNULL
  OR duration_seconds ISNULL
  OR start_time ISNULL
  OR start_station ISNULL
  OR start_station ISNULL
  OR start_terminal ISNULL
  OR end_time ISNULL
  OR end_station ISNULL
  OR end_terminal ISNULL
  OR bike_number ISNULL
  OR rider_type ISNULL
  OR id ISNULL;
```

Looks like the query didn't return any results

i. Solution if null values existed?

- We could've fixed the nulls, let's say one of duration/duration_seconds is missing we could calculate one from the other
- Same goes for Starttime/endtime if one of these fields was missing then we could've calculated those from the duration fields
- If one from startstation/startterminal was missing, then we could've replaced the value for that row because we know that only one terminal exists for each station in our dataset
- Same goes for endStation/EndTerminal
- If a bike number was missing, then we could either delete the whole row if our analysis has importance towards bike numbers otherwise we could let it be a null
- If a ridertype was missing, then we can do nothing but to delete the row because we do not have a riderid against each row.
  - If we had riderid then we could've looked for the rider type against that riderid

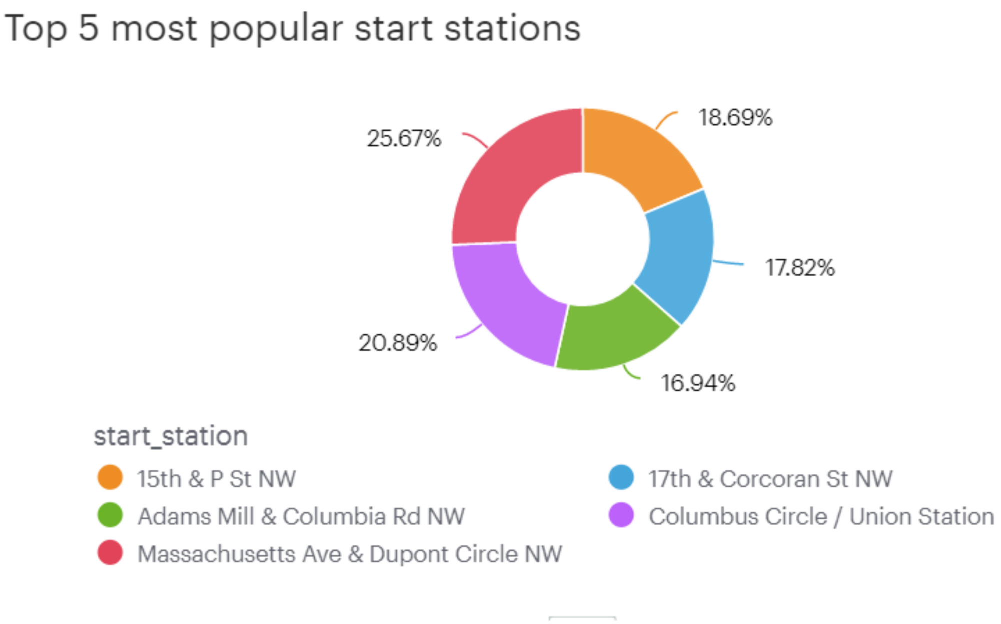
2. What are the top 5 most popular start stations



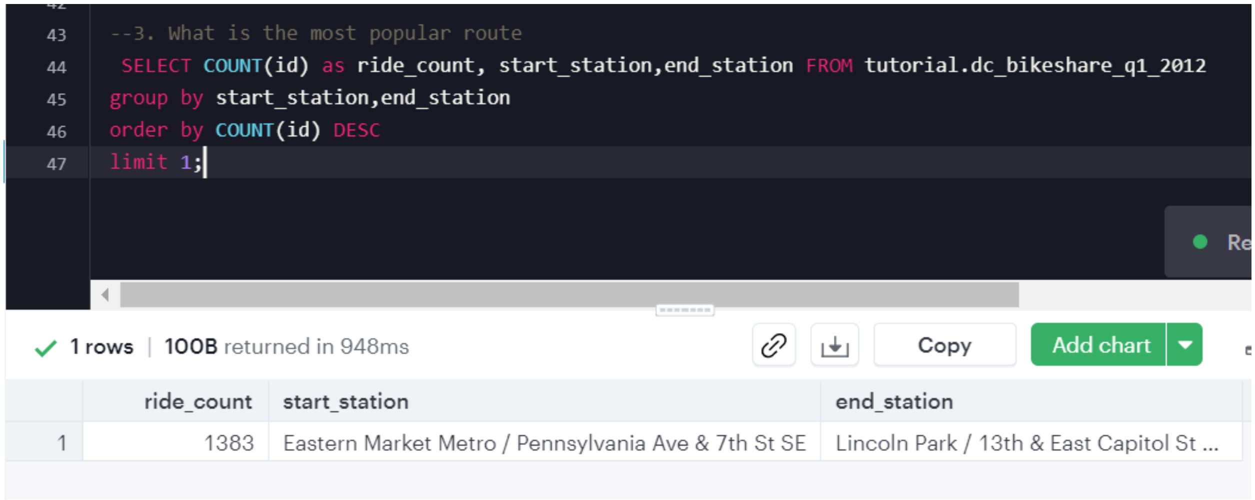
```sql
 --2. What are the top 5 most popular start stations
SELECT COUNT(id), start_station FROM tutorial.dc_bikeshare_q1_2012
group by start_station
order by COUNT(id) DESC
limit 5;
```

rows | 169B returned in 774ms

count	start_station
11261	Massachusetts Ave & Dupont Circle N...
9165	Columbus Circle / Union Station
8199	15th & P St NW
7816	17th & Corcoran St NW
7430	Adams Mill & Columbia Rd NW



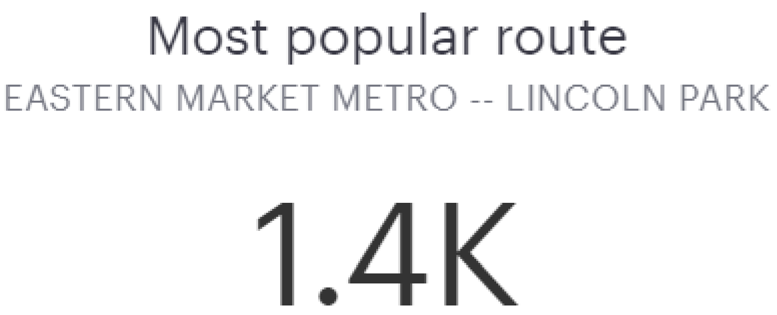
3. What is the most popular route



```
--3. What is the most popular route
 SELECT COUNT(id) as ride_count, start_station,end_station FROM tutorial.dc_bikeshare_q1_2012
group by start_station,end_station
order by COUNT(id) DESC
limit 1;
```

1 rows | 100B returned in 948ms

	ride_count	start_station	end_station
1	1383	Eastern Market Metro / Pennsylvania Ave & 7th St SE	Lincoln Park / 13th & East Capitol St ...



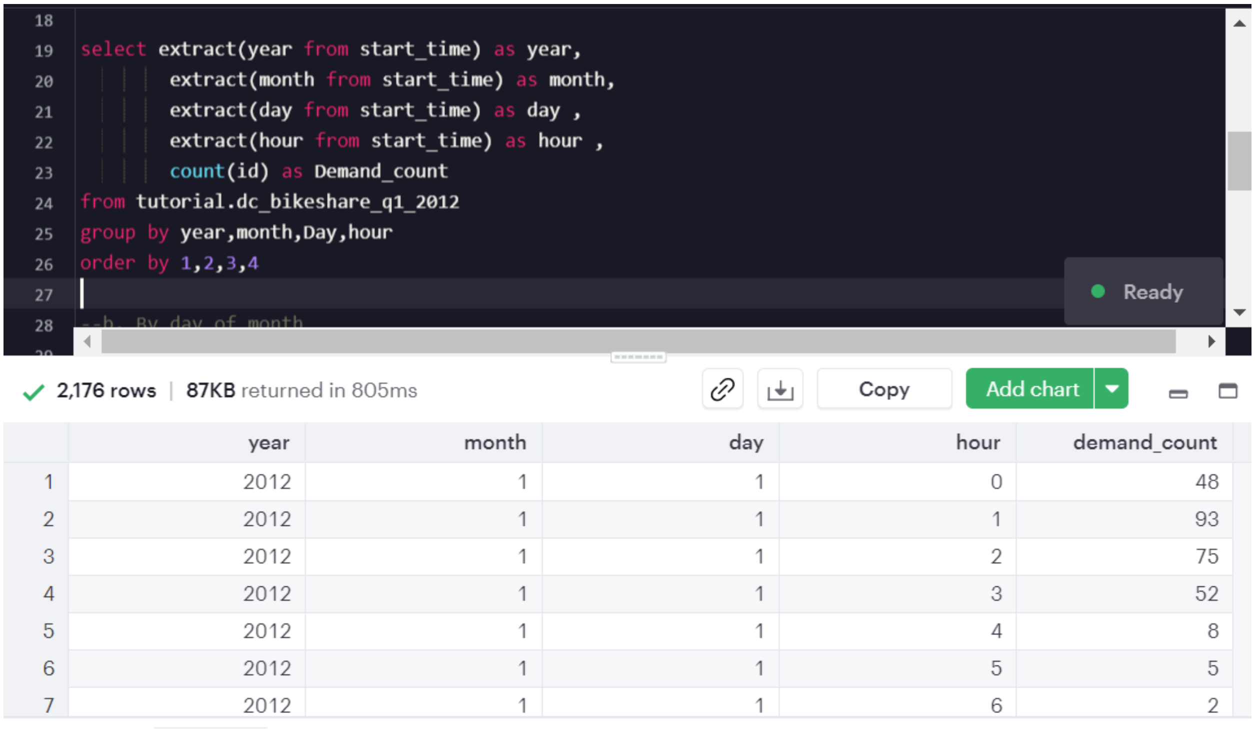
Most popular route

EASTERN MARKET METRO -- LINCOLN PARK

1.4K

4. When is bikeshare demand high and low

a. By hour of day

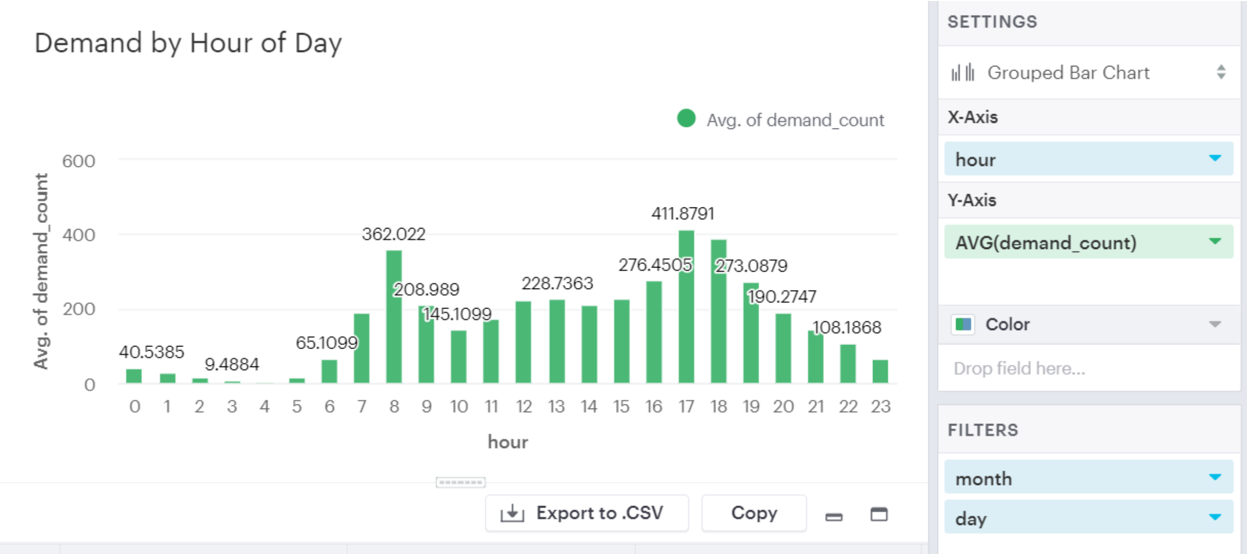


```
select extract(year from start_time) as year,
       extract(month from start_time) as month,
       extract(day from start_time) as day ,
       extract(hour from start_time) as hour ,
       count(id) as Demand_count
from tutorial.dc_bikeshare_q1_2012
group by year,month,Day,hour
order by 1,2,3,4

--b. By day of month
```

✓ 2,176 rows | 87KB returned in 805ms

	year	month	day	hour	demand_count
1	2012	1	1	0	48
2	2012	1	1	1	93
3	2012	1	1	2	75
4	2012	1	1	3	52
5	2012	1	1	4	8
6	2012	1	1	5	5
7	2012	1	1	6	2

- The following graph averages the demand count on all hours of all days, we can also change the month and day filters to represent the specific days of specific months to view more trends and patterns.



SETTINGS

Grouped Bar Chart

X-Axis

hour

Y-Axis

AVG(demand_count)

Color

Drop field here...

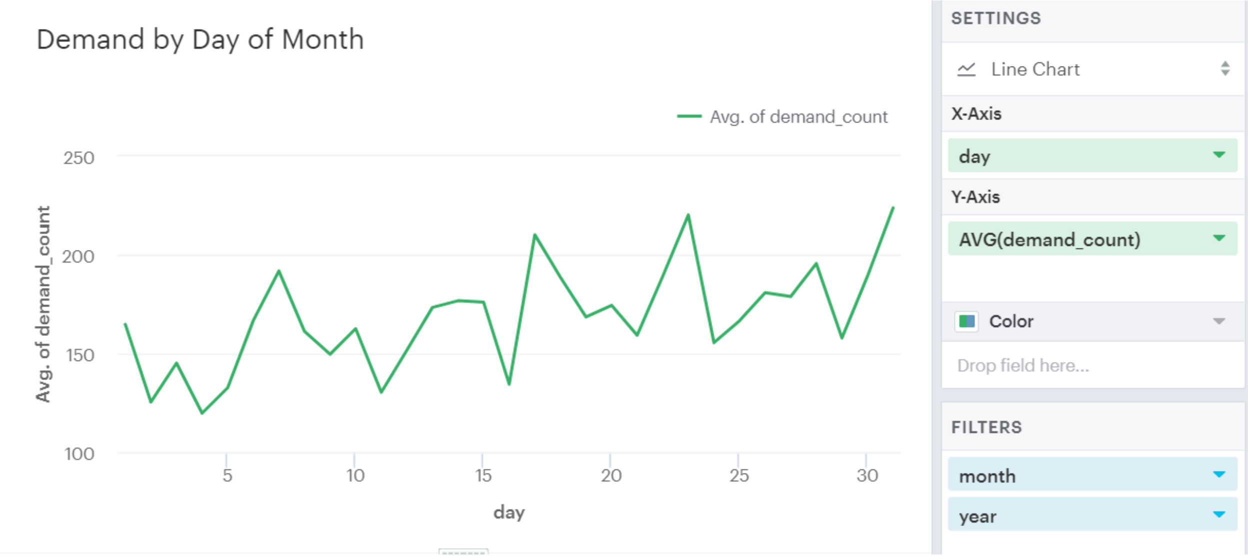
FILTERS

month

day

b. By day of month

- The following graph averages the demand count on all days of all months, we can also change the month and day filters to represent the specific days of specific months to view more trends and patterns.



SETTINGS

Line Chart

X-Axis

day

Y-Axis

AVG(demand_count)

Color

Drop field here...

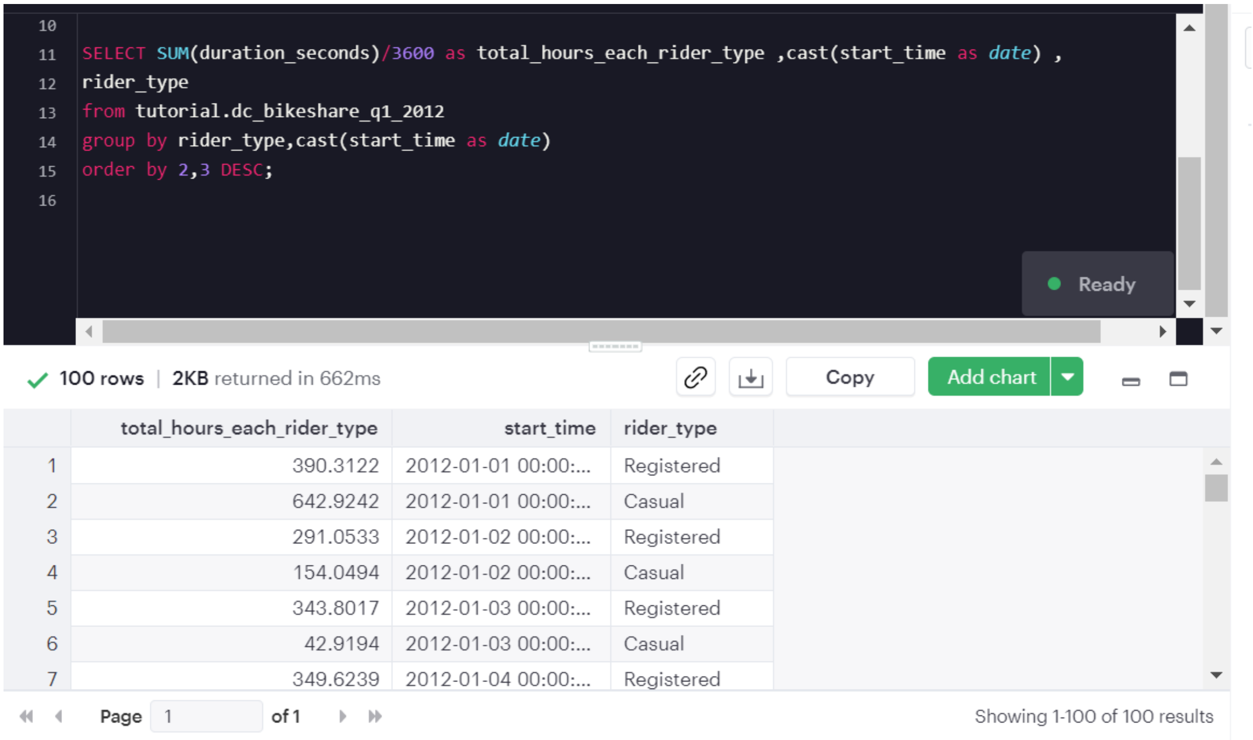
FILTERS

month

year

5. Comment on trip duration and how it varies by rider type:

It is observed that generally registered riders are the ones using the service most often, these may include people relying on the bike sharing service for their daily commute, exercise rides or simply regular joy rides around the city. Hence, they make up for 76% of all rides. Therefore, it is important for the bike sharing service to try to bring more people on board as registered riders by introducing perks and price effective packages for the registered riders.



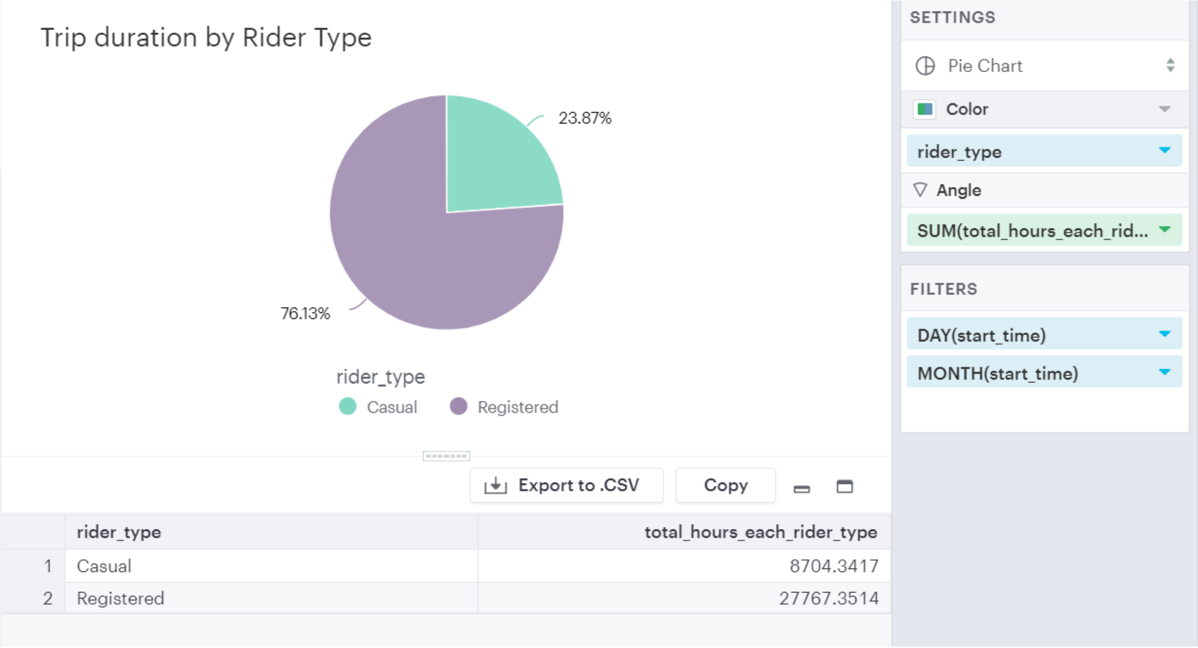
```
SELECT SUM(duration_seconds)/3600 as total_hours_each_rider_type ,cast(start_time as date) ,
rider_type
from tutorial.dc_bikeshare_q1_2012
group by rider_type,cast(start_time as date)
order by 2,3 DESC;
```

✓ 100 rows | 2KB returned in 662ms

	total_hours_each_rider_type	start_time	rider_type
1	390.3122	2012-01-01 00:00:...	Registered
2	642.9242	2012-01-01 00:00:...	Casual
3	291.0533	2012-01-02 00:00:...	Registered
4	154.0494	2012-01-02 00:00:...	Casual
5	343.8017	2012-01-03 00:00:...	Registered
6	42.9194	2012-01-03 00:00:...	Casual
7	349.6239	2012-01-04 00:00:...	Registered

Page 1 of 1

Showing 1-100 of 100 results



Trip duration by Rider Type

SETTINGS

Pie Chart

Color

rider_type

Angle

SUM(total_hours_each_rid...

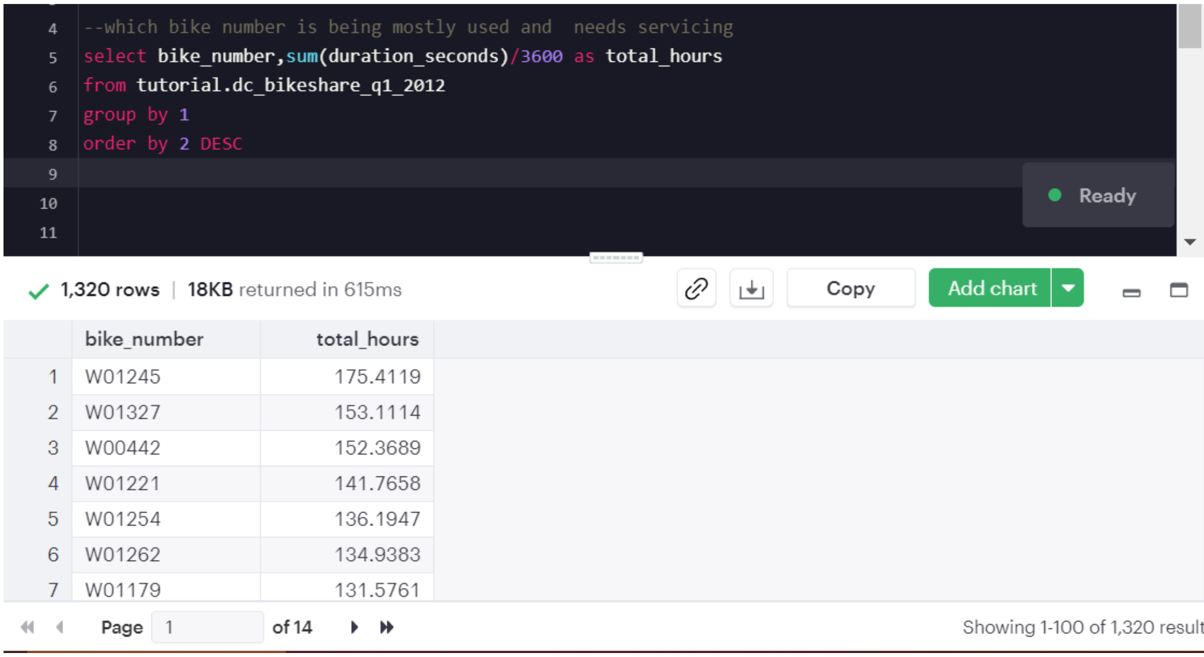
FILTERS

DAY(start_time)

MONTH(start_time)

	rider_type	total_hours_each_rider_type
1	Casual	8704.3417
2	Registered	27767.3514

6. Which bike numbers are being most used? hence requiring service.



```
--which bike number is being mostly used and  needs servicing
select bike_number,sum(duration_seconds)/3600 as total_hours
from tutorial.dc_bikeshare_q1_2012
group by 1
order by 2 DESC
```

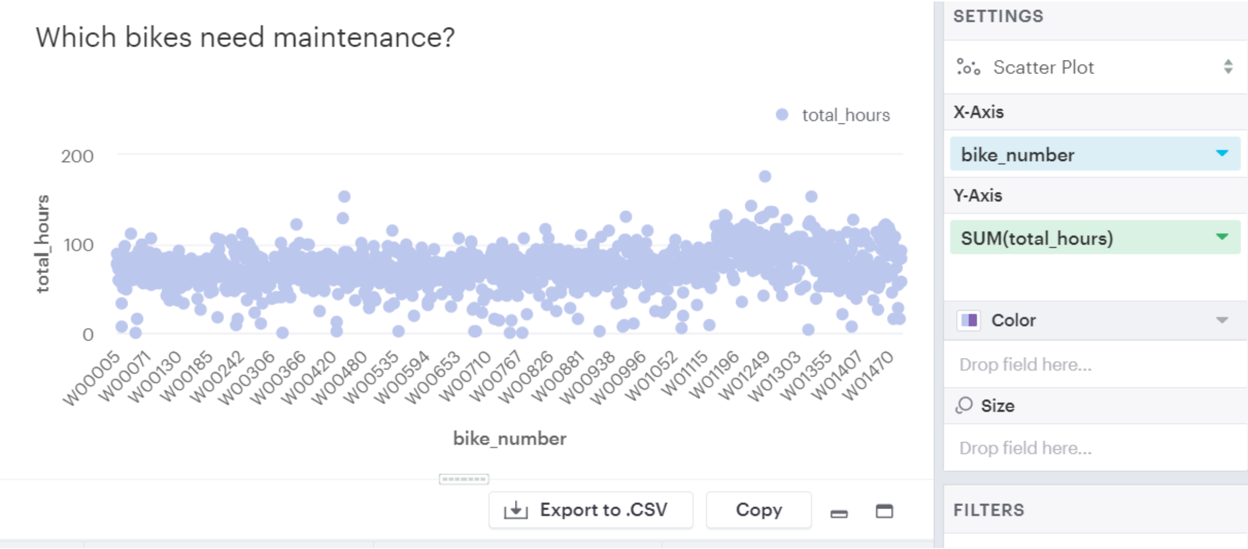
● Ready

✓ 1,320 rows | 18KB returned in 615ms

	bike_number	total_hours
1	W01245	175.4119
2	W01327	153.1114
3	W00442	152.3689
4	W01221	141.7658
5	W01254	136.1947
6	W01262	134.9383
7	W01179	131.5761

Page 1 of 14

Showing 1-100 of 1,320 result



Which bikes need maintenance?

SETTINGS

Scatter Plot

X-Axis

bike_number

Y-Axis

SUM(total_hours)

Color

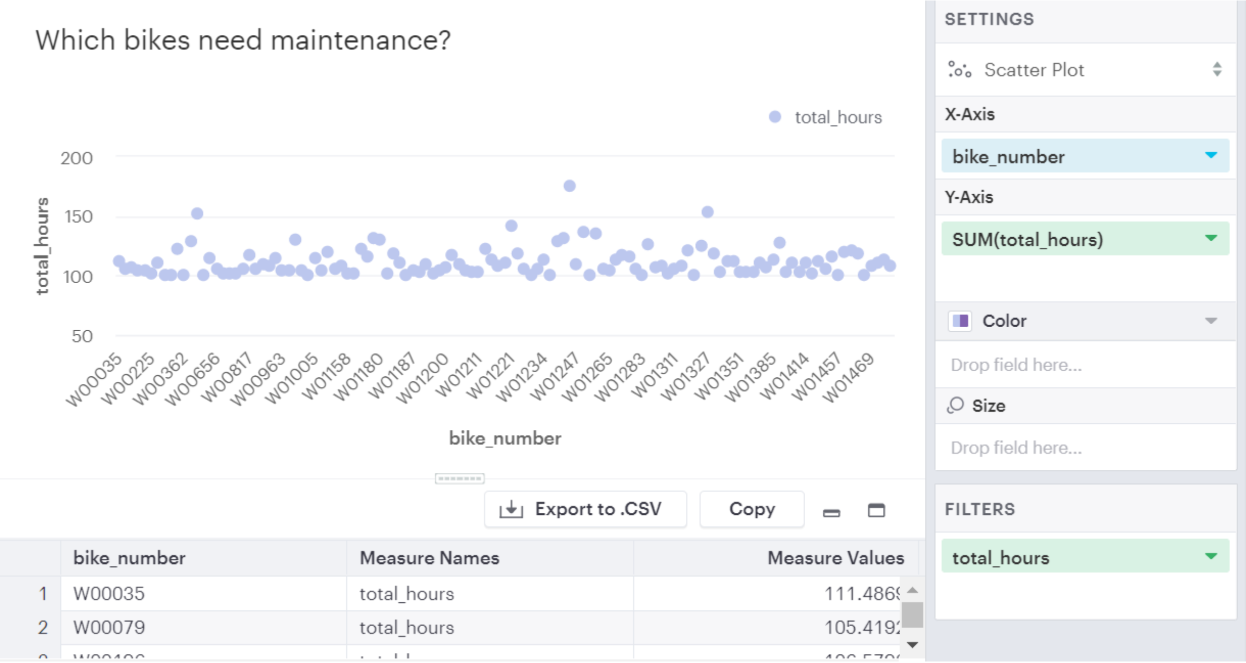
Drop field here...

Size

Drop field here...

FILTERS

- Filter on total hours 100-175 to further drill down on most used bikes



	bike_number	Measure Names	Measure Values
1	W00035	total_hours	111.486
2	W00079	total_hours	105.419

SETTINGS

Scatter Plot

X-Axis

bike_number

Y-Axis

SUM(total_hours)

Color

Drop field here...

Size

Drop field here...

FILTERS

total_hours