

Task 1: Setting up the AWS environment and loading data

Creating an RDS instance in my AWS account and uploading the data to the RDS instance

Since the dataset is huge, I uploaded the data from only two files (i.e. yellow_tripdata_2017-01.csv & yellow_tripdata_2017-02.csv) from the dataset.

Note: I created an appropriate schema for the data sets to upload them to RDS.

1. RDS instance creation in AWS

The screenshot displays the AWS Management Console interface for an Amazon RDS instance. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databaseid=dbinstance;js-cluster=false`. The console header includes the AWS logo, a search bar, and navigation links for Services, CloudShell, Feedback, and Language. The main content area is titled 'dbinstance' and includes a 'Modify' button and an 'Actions' dropdown menu.

Summary

DB identifier dbinstance	CPU 7.67%	Status Available	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-1d

Connectivity & security

Endpoint & port Endpoint dbinstance.ca1depqqnwki.us-east-1.rds.amazonaws.com	Networking Availability Zone us-east-1d	Security VPC security groups default (sg-029a450e6c38d99c7) Active
---	--	--

The bottom of the screenshot shows the Windows taskbar with the date and time as 3:28 PM on 2023-09-05, and the system temperature as 20°C.

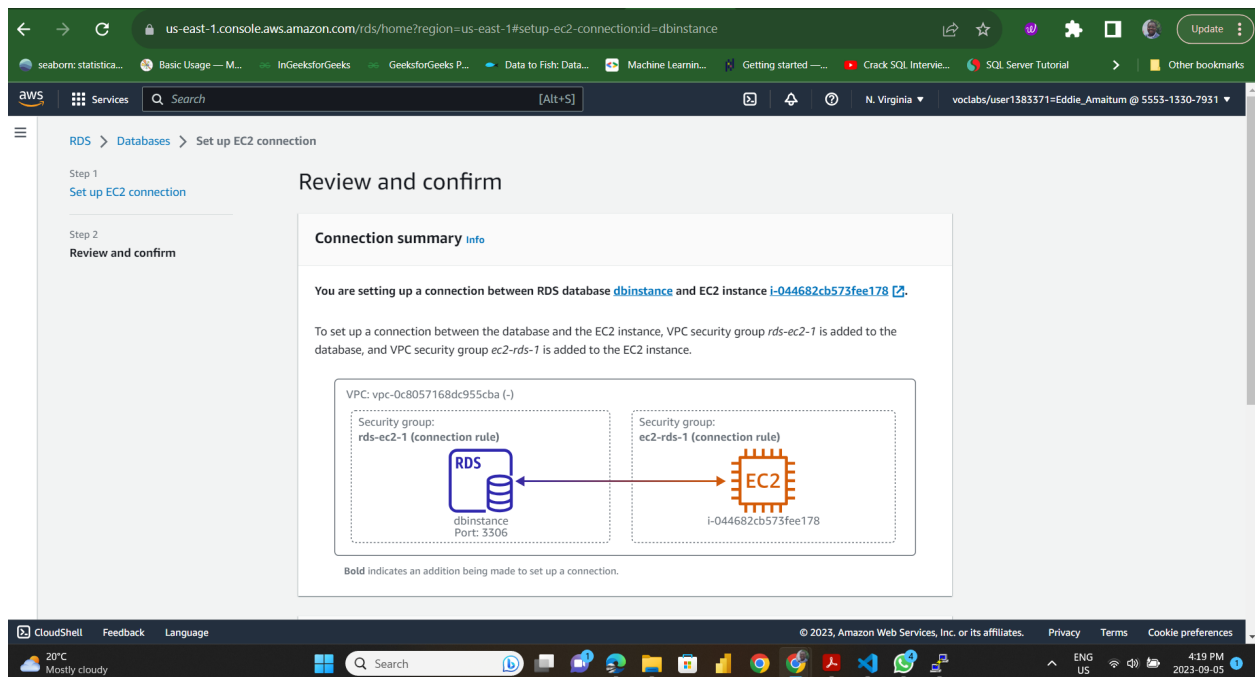
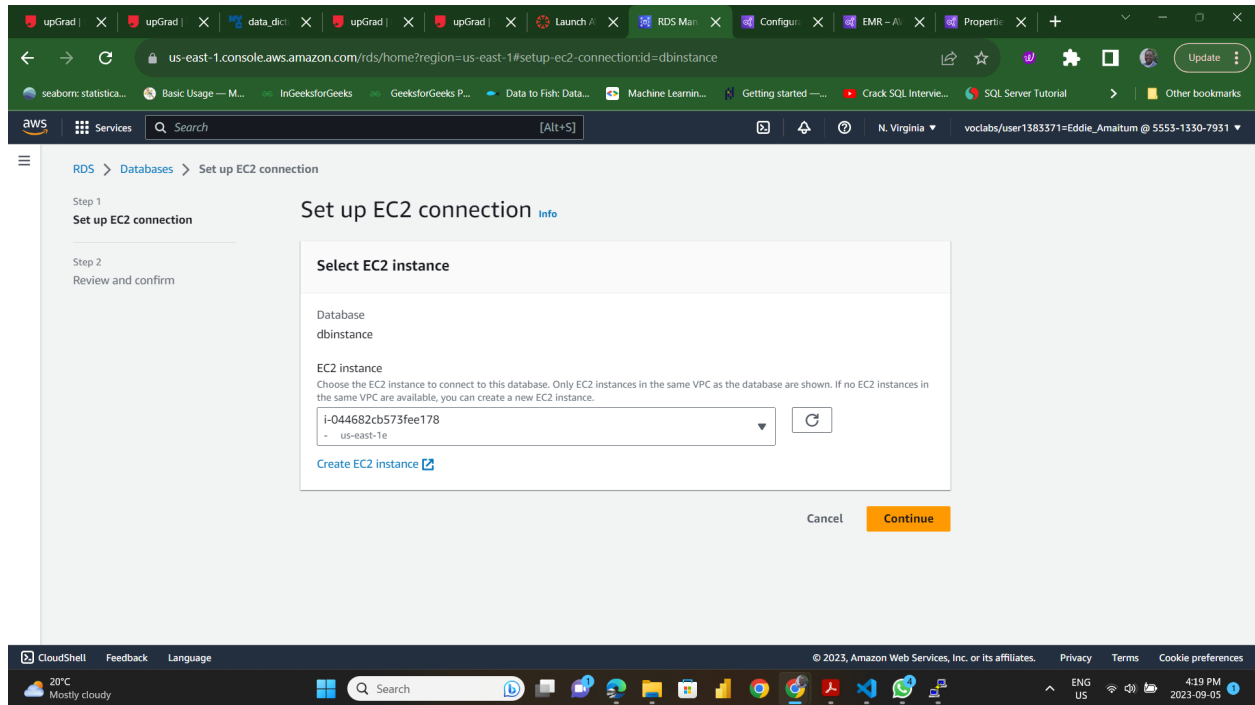
2. EMR creation, some applications selected include:
Apache Sqoop, Apache Hbase, Hadoop

The screenshot shows the Amazon EMR console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and a user profile. Below the navigation bar, a sidebar on the left lists various services like EMR Studio, EMR Serverless, and Clusters. The main content area displays the details for a cluster named 'emrcluster', which is in a 'Waiting' state. The cluster details are organized into sections: Summary, Configuration details, Application user interfaces, and Network and hardware. The Summary section includes the cluster ID (j-MZ7DLJD0KEXI), creation date (2023-09-05 15:45 UTC-7), and elapsed time (17 minutes). The Configuration details section lists the release label (emr-5.30.1), Hadoop distribution (Amazon 2.8.5), and applications (Hive 2.3.6, Pig 0.17.0, Hue 4.6.0, Sqoop 1.4.7, HBase 1.4.13). The Application user interfaces section shows persistent user interfaces (YARN timeline server, Tez UI) and on-cluster user interfaces (Not Enabled). The Network and hardware section indicates the availability zone (us-east-1e) and subnet (subnet-02ad872cee53f7682).

3. Connecting RDS with the EMR instance:
 - We configure security group by editing inbound rules

The screenshot shows the Amazon EC2 console interface, specifically the 'Edit inbound rules' page for a security group. The page title is 'Edit inbound rules' with a sub-header 'Inbound rules control the incoming traffic that's allowed to reach the instance.' Below the title, there's a table of inbound rules. The table has columns for Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There are two rules listed: one for 'All traffic' and another for 'MySQL/Aurora'. The 'MySQL/Aurora' rule is selected, and its details are shown in a modal window. The modal window displays the rule ID (sgr-0bc104d6c32ed76c4), type (MySQL/Aurora), protocol (TCP), port range (3306), and source (Anywh...). There are also buttons for 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

- Then we click on 'Action' button on RDS menu and then 'Set up EC2 connection'



4. We then log into RDS through EMR instance using the command:

We enter password upon request to complete login

```

[~] hadoop@ip-172-31-49-39-:
[~] login as: hadoop
[~] Authenticating with public key "imported-openssh-key"
Last login: Tue Sep 5 23:05:28 2023

  ____      _
 / ___|  __| | | |
 \___ \  | | | | | |
  ___) | | | | | | |
 |_____|_|_|_|_|_|_|

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
94 package(s) needed for security, out of 162 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR
E::::::::::::::::::::: M ::::: M M ::::: M R ::::: ::::: R
EB:::::::::::::::::::: M ::::: M M ::::: M R ::::: RRRRRR :::: R
E :::: E EEEEE M ::::: M M ::::: M RR :::: R R :::: R
E :::: E M ::::: M M ::::: M R :::: R R :::: R
E :::: EEEEE M :::: M :::: M M :::: M R :::: RRRRRR :::: R
E :::: E M :::: M M :::: M M :::: M R :::: RRRRRR :::: R
E :::: E M :::: M M :::: M R :::: R R :::: R
E :::: E EEEEE M :::: M M :::: M R :::: R R :::: R
EB:::::::::::::::::::: M :::: M M :::: M R :::: R R :::: R
E :::: E M :::: M M :::: M R :::: RR :::: R R :::: R
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRR RRRRRR

[hadoop@ip-172-31-49-39 ~]$ mysql -h dbinstance.caldepqgwki.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>

```

5. We then create database using code below:

```
> create database yellow_taxi;
```

```
[~] hadoop@ip-172-31-49-39:~  
┌───┴──────────┐ Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
64 package(s) needed for security, out of 162 available  
Run "sudo yum update" to apply all updates.  
  
EEEEEEEEEEEEEEEEEE MMMMMMMM MRRRRRRRRRRRRR  
E::::::::::::::::::E M:::MM M:::MM R:::MM  
EE::::::::::::::::::E M:::MM M:::MM R:::RRRRRR:::R  
 E:::E EEEE M:::MM M:::MM RR:::R R:::R  
 E:::E M:::MM:M:M M:::MM R:::R R:::R  
 E:::EEEEEEEEEE M:::M M:::M M:::M R:::RRRRRR:::R  
 E::::::::::::::::::E M:::M M:::M:M M:::M R:::MM  
 E:::EEEEEEEEEE M:::M M:::M M:::M R:::RRRRRR:::R  
 E:::E M:::M M:::M M:::M R:::R R:::R  
 E:::E EEEE M:::M MM M:::M R:::R R:::R  
 EE::::::::::::::::::E M:::M M:::M R:::R R:::R  
 E::::::::::::::::::E M:::M M:::MM RR:::R R:::R  
 EEEEEEEEEEEEEEEEE MMMMMMMM MRRRRRR RRRRRR  
  
[hadoop@ip-172-31-49-39 ~]$ mysql -h dbinstance.caldepqgwki.us-east-1.rds.amazonaws.com -P 3306 -u admin -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MySQL connection id is 26  
Server version: 8.0.33 Source distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MySQL [(none)]> create database yellow_taxi  
->  
Query OK, 1 row affected (0.02 sec)  
  
MySQL [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| yellow_taxi |  
+-----+  
5 rows in set (0.00 sec)
```

We then create table using code below;

> use yellow_taxi;

> create table trip_records (VendorID INT, tpep_pickup_datetime VARCHAR(255), tpep_dropoff_datetime VARCHAR(255), Passenger_count INT, Trip_distance FLOAT, RatecodeID INT, store_and_fwd_flag VARCHAR(50), PULocationID INT, DOLocationID INT, payment_type INT, fare_amount FLOAT, extra FLOAT, mta_tax FLOAT, tip_amount FLOAT, tolls_amount FLOAT, improvement_surcharge FLOAT, total_amount FLOAT, Airport_fee FLOAT);

```
hadoop@ip-172-31-49-39:~$ mysql -h dbinstance.caldepqgwkl.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> create database yellow_taxi
-> ;
Query OK, 1 row affected (0.02 sec)

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| yellow_taxi |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> use yellow_taxi;
Database changed
MySQL [yellow_taxi]> create table trip_records (VendorID INT,tpep_pickup_datetime VARCHAR(255),tpep_dropoff_datetime VARCHAR(255),passenger count INT,trip distance FLOAT,RatecodeID INT,stor
e_and_fwd_flag VARCHAR(50),PULocationID INT,DOLocationID INT,payment_type INT,fare_amount FLOAT,extra FLOAT,mta_tax FLOAT,tip_amount FLOAT,tolls_amount FLOAT,improvement_surcharge FLOAT,tot
al_amount FLOAT,Airport_fee FLOAT);
Query OK, 0 rows affected (0.05 sec)

MySQL [yellow_taxi]> show tables;
+-----+
| Tables_in_yellow_taxi |
+-----+
| trip_records |
+-----+
1 row in set (0.00 sec)

MySQL [yellow_taxi]>
```

6. To download the required csv files, we use the following commands:

wget "https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv"

wget "https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv"

```
hadoop@ip-172-31-49-39:~$
1 yellow_taxi
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> use yellow_taxi;
Database changed
MySQL [yellow_taxi]> create table trip_records (VendorID INT,tpep_pickup_datetime VARCHAR(255),tpep_dropoff_datetime VARCHAR(255),passenger_count INT,trip_distance FLOAT,RatecodeID INT,stor
e and fwd flag VARCHAR(50),PULocationID INT,DOLocationID INT,payment_type INT,fare_amount FLOAT,extra FLOAT,mta_tax FLOAT,tip_amount FLOAT,Tolls_amount FLOAT,improvement_surcharge FLOAT,tot
al_amount FLOAT,Airport_fee FLOAT);
Query OK, 0 rows affected (0.05 sec)

MySQL [yellow_taxi]> show tables;
+-----+
1 Tables in yellow_taxi |
+-----+
1 trip_records
+-----+
1 row in set (0.00 sec)

MySQL [yellow_taxi]> exit()
>
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'exit()' at line 1
MySQL [yellow_taxi]> exit:
Bye
[hadoop@ip-172-31-49-39 ~]$ wget "https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv"
--2023-09-06 00:09:38-- https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
Resolving nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)... 3.5.10.150, 3.5.25.16, 3.5.28.244, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.10.150|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 914029540 (872M) [text/csv]
Saving to: 'yellow_tripdata_2017-01.csv'

100%[=====] 914,029,540 24.7MB/s in 33s

2023-09-06 00:10:11 (26.8 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[hadoop@ip-172-31-49-39 ~]$ wget "https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv"
--2023-09-06 00:10:47-- https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
Resolving nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)... 3.5.17.120, 3.5.27.160, 52.216.52.17, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.17.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 863487050 (823M) [text/csv]
Saving to: 'yellow_tripdata_2017-02.csv'

100%[=====] 863,487,050 23.4MB/s in 33s

2023-09-06 00:11:20 (25.0 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]

[hadoop@ip-172-31-49-39 ~]$
```

7. To load data in MySQL table, we login and run SQL commands below:

- > LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
 - > INTO TABLE trip_records
 - > FIELDS TERMINATED BY ','
 - > LINES TERMINATED BY '\n'
 - > IGNORE 1 LINES;
-
- > LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'
 - > INTO TABLE trip_records
 - > FIELDS TERMINATED BY ','
 - > LINES TERMINATED BY '\n'
 - > IGNORE 1 LINES;

```
hadoop@ip-172-31-49-39:~$
+-----+
| 18880595 |
+-----+
1 row in set (1 min 0.46 sec)

MySQL [yellow_taxi]> DROP TABLE trip_records;
Query OK, 0 rows affected (0.69 sec)

MySQL [yellow_taxi]> show tables;
Empty set (0.02 sec)

MySQL [yellow_taxi]> create table trip_records (VendorID INT,tpep_pickup_datetime VARCHAR(255),tpep_dropoff_datetime VARCHAR(255),passenger_count INT,trip_distance FLOAT,RatecodeID INT,store_and_fwd_flag VARCHAR(50),PULocationID INT,DOLocationID INT,payment_type INT,fare_amount FLOAT,extra FLOAT,mta_tax FLOAT,tip_amount FLOAT,tolls_amount FLOAT,improvement_surcharge FLOAT,total_amount FLOAT,Airport_fee FLOAT);
Query OK, 0 rows affected (0.15 sec)

MySQL [yellow_taxi]> show tables;
+-----+
| Tables_in_yellow_taxi |
+-----+
| trip_records           |
+-----+
1 row in set (0.01 sec)

MySQL [yellow_taxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
-> INTO TABLE trip_records
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 9710820 rows affected, 65535 warnings (3 min 8.00 sec)
Records: 9710820 Deleted: 0 Skipped: 0 Warnings: 9710820

MySQL [yellow_taxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'
-> INTO TABLE trip_records
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 9169775 rows affected, 65535 warnings (2 min 44.64 sec)
Records: 9169775 Deleted: 0 Skipped: 0 Warnings: 9169775

MySQL [yellow_taxi]> select count(*) from trip_records;
+-----+
| count(*) |
+-----+
| 18880595 |
+-----+
1 row in set (50.74 sec)

MySQL [yellow_taxi]>
```

8. Confirming that data is loaded: to do this, we run simple SQL queries:

> select count (*) from trip_records;

> select * from trip_records limit 5;

```
hadoop@ip-172-31-49-39:~$
+-----+
| 18880595 |
+-----+
1 row in set (0.01 sec)

MySQL [yellow_taxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-01.csv'
-> INTO TABLE trip_records
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 9710820 rows affected, 65535 warnings (3 min 8.00 sec)
Records: 9710820 Deleted: 0 Skipped: 0 Warnings: 9710820

MySQL [yellow_taxi]> LOAD DATA LOCAL INFILE '/home/hadoop/yellow_tripdata_2017-02.csv'
-> INTO TABLE trip_records
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 9169775 rows affected, 65535 warnings (2 min 44.64 sec)
Records: 9169775 Deleted: 0 Skipped: 0 Warnings: 9169775

MySQL [yellow_taxi]> select count(*) from trip_records;
+-----+
| count(*) |
+-----+
| 18880595 |
+-----+
1 row in set (50.74 sec)

MySQL [yellow_taxi]> select * from trip_records limit 5;
+-----+
| VendorID | tpep_pickup_datetime | tpep_dropoff_datetime | passenger_count | trip_distance | RatecodeID | store_and_fwd_flag | PULocationID | DOLocationID | payment_type | fare_amount | extra | mta_tax | tip_amount | tolls_amount | improvement_surcharge | total_amount | Airport_fee |
+-----+
| 1 | 2017-01-01 00:32:05 | 2017-01-01 00:37:48 | 1 | 1.2 | 1 | N | 140 | 236 | 2 | 6.5 | 0.5 | 0.5 | 0 | 0 | 0 | 7.5 | 0 |
| 1 | 2017-01-01 00:43:25 | 2017-01-01 00:47:42 | 1 | 0.7 | 1 | N | 237 | 140 | 2 | 5 | 0.5 | 0.5 | 0 | 0 | 0 | 6.3 | 0 |
| 1 | 2017-01-01 00:49:10 | 2017-01-01 00:53:53 | 2 | 0.8 | 1 | N | 140 | 237 | 2 | 5.5 | 0.5 | 0.5 | 0 | 0 | 0 | 6.8 | 0 |
| 1 | 2017-01-01 00:36:42 | 2017-01-01 00:41:09 | 1 | 1.1 | 1 | N | 41 | 42 | 2 | 6 | 0.5 | 0.5 | 0 | 0 | 0 | 7.3 | 0 |
| 1 | 2017-01-01 00:07:41 | 2017-01-01 00:18:16 | 1 | 3 | 1 | N | 48 | 263 | 2 | 11 | 0.5 | 0.5 | 0 | 0 | 0 | 12.3 | 0 |
+-----+
5 rows in set (0.03 sec)

MySQL [yellow_taxi]>
```