

## Data Ingestion Tasks:

**Task 1.** Create an RDS instance in your AWS account and upload the data to the RDS instance (Note:

Instructions on how to work with RDS can be found

Since the dataset is huge, you need to upload the data from only two files

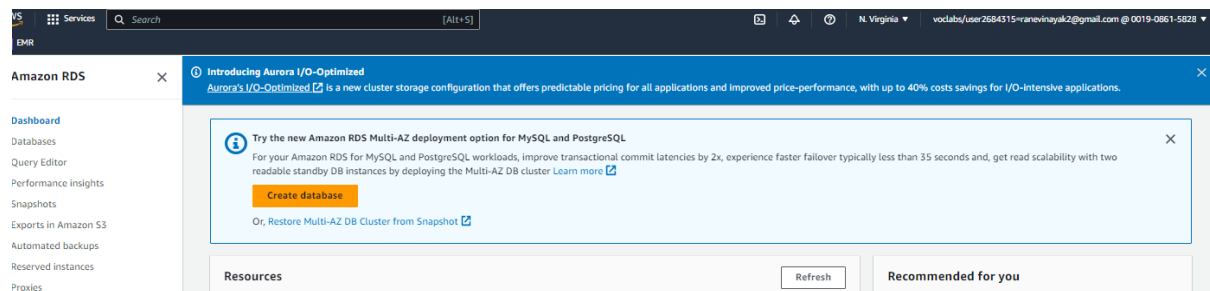
(i.e. `yellow_tripdata_2017-01.csv` & `yellow_tripdata_2017-02.csv`) from the dataset.

### Solution

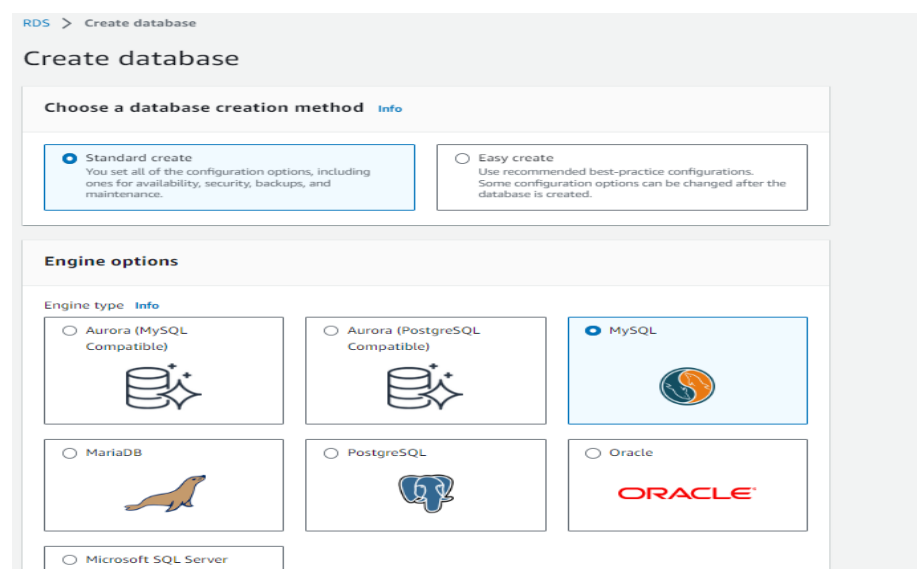
Followed below steps to create RDS instance and to upload data to it.

Steps:

1. Create RDS instance on AWS portal



2. Create MYSQL database with free tier.



3. Create Credential with db. name, user name and password.

Single DB instance (not supported for Multi-AZ DB cluster snapshot)  
Creates a single DB instance with no standby DB instances.

### Settings

**DB instance identifier** [Info](#)  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

demodb

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

**Master username** [Info](#)  
Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

☐ **Manage master credentials in AWS Secrets Manager**  
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

[Learn more](#)

☐ **Auto generate a password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** [Info](#)

\*\*\*\*\*

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @

4. Create EC2 instance and connect it with RDS instance.  
VPC: vpc-02c6bc8cf08bb3771  
EC2 = ec2-3-220-231-186.compute-1.amazonaws.com

## Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags [Info](#)

Name

casestudyec2

[Add additional tags](#)

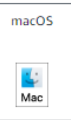
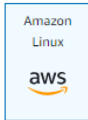
### ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

🔍 Search our full catalog including 1000s of application and OS images

Recents

**Quick Start**



**Browse more AMIs**  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

### ▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2...[read more](#)  
ami-051f7e7f6c2f40dc1

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

**Launch instance**

[Review commands](#)

You are setting up a connection between RDS database [demodb1](#) and EC2 instance [i-01ef3c8a7a1cc616b](#).

To set up a connection between the database and the EC2 instance, VPC security group *rds-ec2-4* is added to the database, and VPC security group *ec2-rds-4* is added to the EC2 instance.



**Bold indicates an addition being made to set up a connection.**

#### Changes to RDS database: demodb1

Attribute	Current value	New value
Security group	taxicasestudy	taxicasestudy, <b>rds-ec2-4</b>

#### Changes to EC2 instance: i-01ef3c8a7a1cc616b

Attribute	Current value	New value
Security group	default	default, <b>ec2-rds-4</b>

5. RDS database has been created which will be accessible through below URL  
Db url = demodb1.cr0zubg47bor.us-east-1.rds.amazonaws.com

The screenshot shows the Amazon RDS console for instance **demodb1**. The left sidebar contains navigation links for Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main content area displays the instance details under the 'Summary' tab. The instance is in the 'Available' state, using the 'db.t3.micro' class in the 'us-east-1f' region. The 'Connectivity & security' tab is selected, showing the endpoint **demodb1.cr0zubg47bor.us-east-1.rds.amazonaws.com** on port **3306**. Networking details include the 'us-east-1f' availability zone, 'vpc-02c6bc8cf08bb3771' VPC, and 'default-vpc-02c6bc8cf08bb3771' subnet group. Security settings show the instance is publicly accessible and associated with the 'taxicasestudy' VPC security group.

Summary			
DB identifier demodb1	CPU 2.52%	Status Available	Class db.t3.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-1f

Connectivity & security		
<b>Endpoint &amp; port</b> Endpoint demodb1.cr0zubg47bor.us-east-1.rds.amazonaws.com Port 3306	<b>Networking</b> Availability Zone us-east-1f VPC vpc-02c6bc8cf08bb3771 Subnet group default-vpc-02c6bc8cf08bb3771 Subnets subnet-01f9d99b6f101fe3f subnet-06201dc4b57d9dc35	<b>Security</b> VPC security groups taxicasestudy (sg-087f27d93a1eceda1) Active rds-ec2-4 (sg-04bb25a6fb7fdec93) Active Publicly accessible Yes Certificate authority rds-ca-2019 Certificate authority date

6. Update Inbound properties of RDS instance to connect EC2 and EMR with RDS

The screenshot shows the 'Edit inbound rules' page for security group **sg-087f27d93a1eceda1**. The page lists existing inbound rules and allows adding new ones. The table below shows the current rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Action
sg-0494f6f0f4d3f3a74	SSH	TCP	22	Custom	0.0.0.0/0	Delete
sg-0920f6902f64cd84a	MySQL/Aurora	TCP	3306	Custom	18.204.7.36/32	Delete
sg-0937ac14394b2a2cf	SSH	TCP	22	Custom	105.10.224.194/32	Delete
sg-0b14de6a8349d2085	All TCP	TCP	0 - 65535	Custom	103.10.224.194/32	Delete
sg-0be262c8061012fd6	MySQL/Aurora	TCP	3306	Custom	0.0.0.0/0	Delete
sg-0724398b961fb725c	All TCP	TCP	0 - 65535	Custom	0.0.0.0/0	Delete
sg-0ac3b4c3aee8990a4	MySQL/Aurora	TCP	3306	Custom	105.10.224.194/32	Delete

7. Launch EMR cluster

Amazon EMR > EMR on EC2: Clusters > Create cluster

## Create cluster [Info](#)

### Name and applications [Info](#)

Name

Amazon EMR release [Info](#)

A release contains a set of applications which can be installed on your cluster.

emr-6.12.0

Application bundle

Spark

Core Hadoop

Flink

HBase

Presto

Trino

Custom

▼ Customise your application bundle

Applications included in bundle

<input type="checkbox"/> Flink 1.17.0	<input type="checkbox"/> Ganglia 3.7.2	<input checked="" type="checkbox"/> HBase 2.4.17
<input type="checkbox"/> HCatalog 3.1.3	<input checked="" type="checkbox"/> Hadoop 3.3.3	<input checked="" type="checkbox"/> Hive 3.1.3
<input checked="" type="checkbox"/> Hue 4.11.0	<input type="checkbox"/> JupyterEnterpriseGateway 2.6.0	<input type="checkbox"/> JupyterHub 1.4.1
<input type="checkbox"/> Livy 0.7.1	<input type="checkbox"/> MXNet 1.9.1	<input type="checkbox"/> Oozie 5.2.1
<input type="checkbox"/> Phoenix 5.1.3	<input type="checkbox"/> Pig 0.17.0	<input type="checkbox"/> Presto 0.281
<input type="checkbox"/> Spark 3.4.0	<input checked="" type="checkbox"/> Sqoop 1.4.7	<input type="checkbox"/> TensorFlow 2.11.0
<input type="checkbox"/> Tez 0.10.2	<input type="checkbox"/> Trino 414	<input type="checkbox"/> Zeppelin 0.10.1
<input type="checkbox"/> ZooKeeper 3.5.10		

AWS Glue Data Catalogue settings

Use the AWS Glue Data Catalog to provide an external metastore for your application.

☐ Use for Hive table metadata

### Summary [Info](#)

#### Name and applications

Name

taxicasesstudy1

Amazon EMR release

emr-6.12.0

Application bundle

Customised (HBase 2.4.17, Hadoop 3.3.3, Hive 3.1.3, Hue 4.11.0, Sqoop 1.4.7)

Amazon Linux release

2.0.20230808.0

#### Cluster configuration

Instance groups

Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)

#### Cluster scaling and provisioning option

Cancel **Create cluster**

8. Login into EMR cluster with root privilege and run below command for RDS connection.

`mysql -h demodb1.cr0zubg47bor.us-east-1.rds.amazonaws.com -P 3306 -u admin -p`

```
[hadoop@ip-172-30-2-67 ~]$ sudo -i
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M M::::::::M R:::::::::R
EE::::::::EEEEEEEEEE E M::::::::M M::::::::M R:::::::::R
E::::E EEEEE M::::::::M M::::::::M RR::::R R::::R
E::::E M::::M:M::M M::M::::M R::R R::R
E::::EEEEEEEEEE M::::M M::M M::M M::::M R::RRRRRR::::R
E::::::::::::E M::::M M::M:M::M M::::M R:::::::::RR
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 EEEEE M::::M MMM M::::M R::R R::R
EE::::::::EEEEEEEE::E M::::M M::::M R::R R::R
E::::::::::::E M::::M M::::M RR::::R R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRR RRRRRR

[root@ip-172-30-2-67 ~]# mysql -h demodb1.cr0zubg47bor.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 19
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)]>
```

## 9. Download yellow\_tripdata csv into EMR cluster

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](https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv)

```
[root@ip-172-31-6-159 ~]# wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-01.csv
--2023-08-28 16:39:45-- 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)... 3.5.29.217, 94.231.192.113, 52.216.170.123, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|3.5.29.217|: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 27.8MB/s in 32s

2023-08-28 16:40:17 (26.9 MB/s) - 'yellow_tripdata_2017-01.csv' saved [914029540/914029540]

[root@ip-172-31-6-159 ~]# wget https://nyc-tlc-upgrad.s3.amazonaws.com/yellow_tripdata_2017-02.csv
--2023-08-28 16:40:23-- 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)... 52.216.166.43, 3.5.29.135, 52.216.109.147, ...
Connecting to nyc-tlc-upgrad.s3.amazonaws.com (nyc-tlc-upgrad.s3.amazonaws.com)|52.216.166.43|: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 25.5MB/s in 33s

2023-08-28 16:40:56 (25.2 MB/s) - 'yellow_tripdata_2017-02.csv' saved [863487050/863487050]

[root@ip-172-31-6-159 ~]#
```

## 10. Move the csv to local mapr\_assignment dir

Command:

Move to local dir:

Mkdir mapr\_assignment

Cd mapr\_assignment

Mkdir input\_dataset

cp /root/yellow\_tripdata\_\* /root/mapr\_assignment/input\_dataset

```
[root@ip-172-31-6-159 ~]# ls -lrt
total 1735860
-rw-r--r-- 1 root root 914029540 Nov 25 2022 yellow_tripdata_2017-01.csv
-rw-r--r-- 1 root root 863487050 Nov 25 2022 yellow_tripdata_2017-02.csv
[root@ip-172-31-6-159 ~]# head -n 3 yellow_tripdata_2017-01.csv
VendorID,tpep_pickup_datetime,tpep_dropoff_datetime,passenger_count,trip_distance,RatecodeID,store_and_fwd_flag,trip_amount,tolls_amount,improvement_surcharge,total_amount,congestion_surcharge,airport_fee
1,2017-01-01 00:32:05,2017-01-01 00:37:48,1,1.2,1,N,140,236,2,6.5,0.5,0.5,0.0,0.0,0.3,7.8,,
1,2017-01-01 00:43:25,2017-01-01 00:47:42,2,0.7,1,N,237,140,2,5.0,0.5,0.5,0.0,0.0,0.3,6.3,,
[root@ip-172-31-6-159 ~]# mkdir mapr_assignment
[root@ip-172-31-6-159 ~]# cd mapr_assignment/
[root@ip-172-31-6-159 mapr_assignment]# mkdir input_dataset
[root@ip-172-31-6-159 mapr_assignment]# pwd
/root/mapr_assignment
```

```
[root@ip-172-31-6-159 ~]# mv yellow_tripdata_2017-01.csv /root/mapr_assignment/input_dataset/yellow_tripdata_2017-01.csv
[root@ip-172-31-6-159 ~]# mv yellow_tripdata_2017-02.csv /root/mapr_assignment/input_dataset/yellow_tripdata_2017-02.csv
[root@ip-172-31-6-159 ~]# ls
mapr_assignment
[root@ip-172-31-6-159 ~]# ls mapr_assignment/input_dataset/
yellow_tripdata_2017-01.csv yellow_tripdata_2017-02.csv
[root@ip-172-31-6-159 ~]#
```

## 11. Move file to hdfs location /user/hadoop/mapr\_assignment/input

```

hadoop fs -mkdir -p /user/hadoop/mapr_assignment/input

hadoop fs -ls /user/hadoop/mapr_assignment/input

cd mapr_assignment/input_dataset/

hadoop fs -put yellow_tripdata_* /user/hadoop/mapr_assignment/input

```

```

[root@ip-172-31-6-159 ~]# ls mapr_assignment/input_dataset/
yellow_tripdata_2017-01.csv yellow_tripdata_2017-02.csv
[root@ip-172-31-6-159 ~]# hadoop fs -mkdir -p /user/hadoop/mapr_assignment/input
[root@ip-172-31-6-159 ~]# hadoop fs -ls /user/hadoop/mapr_assignment/input
[root@ip-172-31-6-159 ~]# cd mapr_assignment/input_dataset/
[root@ip-172-31-6-159 input_dataset]# hadoop fs -put yellow_tripdata_* /user/hadoop/mapr_assignment/input

[root@ip-172-31-6-159 input_dataset]# hadoop fs -ls /user/hadoop/mapr_assignment/input/
Found 2 items
-rw-r--r-- 1 root hdfsadmingroup 914029540 2023-08-28 16:53 /user/hadoop/mapr_assignment/input/yellow_tripdata_2017-01.csv
-rw-r--r-- 1 root hdfsadmingroup 863487050 2023-08-28 16:53 /user/hadoop/mapr_assignment/input/yellow_tripdata_2017-02.csv
[root@ip-172-31-6-159 input_dataset]#

```

12. Connect to RDS database and display database and tables in it

Command:

Show databases;

13. Create database taxidb in rds instance and use taxidb for table creation.

Command:

create database taxidb;

use taxidb;

```

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)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

MySQL [(none)]> create database taxidb;
Query OK, 1 row affected (0.01 sec)

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| taxidb |
+-----+
5 rows in set (0.01 sec)

MySQL [(none)]> use taxidb;
Database changed
MySQL [taxidb]>

```

14. Run below command to create Vendor table schema in RDS taxidb.

Command:

```
create table Vendor
(
  VendorID INT,
  tpep_pickup_datetime DATETIME,
  tpep_dropoff_datetime DATETIME,
  passenger_count INT,
  trip_distance FLOAT,
  RatecodeID INT,
  store_and_fwd_flag VARCHAR(250),
  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,
  congestion_surcharge FLOAT,
  airport_fee FLOAT
);
```

```
MySQL [taxidb]> create table Vendor
-> (
-> VendorID INT,
-> tpep_pickup_datetime DATETIME,
-> tpep_dropoff_datetime DATETIME,
-> passenger_count INT,
-> trip_distance FLOAT,
-> RatecodeID INT,
-> store_and_fwd_flag VARCHAR(250),
-> 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,
-> congestion_surcharge FLOAT,
-> airport_fee FLOAT
-> );
Query OK, 0 rows affected (0.04 sec)
```



15. Load the tripdata csv into RDS instance by using below command.

```
LOAD DATA LOCAL INFILE '/root/mapr_assignment/input_dataset/yellow_tripdata_2017-01.csv'
INTO TABLE Vendor
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

```
MySQL [taxidb]> LOAD DATA LOCAL INFILE '/root/mapr_assignment/input_dataset/yellow_tripdata_2017-01.csv'
-> INTO TABLE Vendor
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;

Query OK, 9710820 rows affected, 65535 warnings (2 min 31.33 sec)
Records: 9710820 Deleted: 0 Skipped: 0 Warnings: 19421640

MySQL [taxidb]>
```

16. Verify the count from Vendor table  
Select count(\*) from Vendor

```
MySQL [taxidb]>
MySQL [taxidb]> select count(*) from Vendor;
+-----+
| count(*) |
+-----+
| 9710820 |
+-----+
1 row in set (20.81 sec)

MySQL [taxidb]> █
```

17. Verify count with CSV file  
wc -l yellow\_tripdata\_2017-01.csv  
9710821

file contain column as well so data count = 9710821

```
[root@ip-172-31-6-159 input_dataset]# wc -l yellow_tripdata_2017-01.csv
9710821 yellow_tripdata_2017-01.csv
[root@ip-172-31-6-159 input_dataset]# wc -l yellow_tripdata_2017-02.csv
9169776 yellow_tripdata_2017-02.csv
[root@ip-172-31-6-159 input_dataset]# █
```

18. Load the 2<sup>nd</sup> tripdata csv into RDS instance by using below command.

```
LOAD DATA LOCAL INFILE '/root/mapr_assignment/input_dataset/yellow_tripdata_2017-02.csv'
INTO TABLE Vendor
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

```

MySQL [taxidb]> LOAD DATA LOCAL INFILE '/root/mapr_assignment/input_dataset/yellow_tripdata_2017-02.csv' INTO TABLE Vendor FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES;

```

19. Display top 5 records from the table  
Select \* from vendor limit 5;

```

MySQL [taxidb]> select * from Vendor limit 5;

```

VendorID	trip_pickup_datetime	trip_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	FULocationID	DOLocationID	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	congestion_surcharge	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	0.3													
1	2017-01-01 00:43:25	2017-01-01 00:47:42	2	0.7	1	N		237	140	2								
5	0.5	0.5	0	0	0.3													
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	0.3													
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	0.3													
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	0.3													

```

rows in set (0.01 sec)

```

```

2023-08-29 18:50:32,127 INFO orm.CompilationManager: Waiting jar file: /tmp/sqoop-root/compile/42771f3286140c5b0c9fe9f459428cf0/Vendor.jar
2023-08-29 18:50:32,146 WARN manager.MySQLManager: It looks like you are importing from mysql.
2023-08-29 18:50:32,146 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
2023-08-29 18:50:32,146 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
2023-08-29 18:50:32,147 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
2023-08-29 18:50:32,171 INFO mapreduce.ImportJobBase: Beginning import of Vendor
2023-08-29 18:50:32,308 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
2023-08-29 18:50:32,327 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
2023-08-29 18:50:33,937 WARN mapreduce.TableMapReduceUtil: The addDependencyJars(Configuration, Class<?>...) method has been deprecated since it is easy to use incorrectly. M
users should rely on addDependencyJars(Job) instead. See HBASE-8386 for more details.
2023-08-29 18:50:34,162 INFO client.DefaultNoHARMFollowerProxyProvider: Connecting to ResourceManager at ip-172-30-2-198.ec2.internal/172.30.2.198:8032
2023-08-29 18:50:34,325 INFO client.AHSProxy: Connecting to Application History server at ip-172-30-2-198.ec2.internal/172.30.2.198:10200
2023-08-29 18:50:34,937 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1693332422648_0003
2023-08-29 18:50:40,405 INFO db.DBInputFormat: Using read committed transaction isolation
2023-08-29 18:50:40,406 INFO db.DataDrivenDBInputFormat: BoundingValueQuery: SELECT MIN('trip_pickup_datetime'), MAX('trip_pickup_datetime') FROM 'Vendor'
2023-08-29 18:51:20,547 INFO db.IntegerSplitter: Split size: 1274399750; Num splits: 4 from: 1483228800000 to: 148326399000
2023-08-29 18:51:20,634 INFO mapreduce.JobSubmitter: number of splits:4
2023-08-29 18:51:20,809 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2023-08-29 18:51:20,964 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1693332422648_0003
2023-08-29 18:51:20,965 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-08-29 18:51:21,168 INFO conf.Configuration: resource-types.xml not found
2023-08-29 18:51:21,169 INFO resource.ResourceUtil: Unable to find 'resource-types.xml'.
2023-08-29 18:51:21,708 INFO Impl.YarnClientImpl: Submitted application application_1693332422648_0003
2023-08-29 18:51:21,740 INFO mapreduce.Job: The url to track the job: http://ip-172-30-2-198.ec2.internal:20888/proxy/application_1693332422648_0003/
2023-08-29 18:51:21,740 INFO mapreduce.Job: Running job: job_1693332422648_0003

```