```
hadoop@0e8a607bbfab:/$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/hadoop/hive/lib/hive-common-2.3.9.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using H
ive 1.X releases.
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.hive.common.StringInternUtils (file:/home/hadoop/hive/lib/hive-common-2.3.9.jar) to field java.net.U
RI.string
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.hive.common.StringInternUtils
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
hive> SHOW DATABASES;
OK
default
tripdata
Time taken: 0.383 seconds, Fetched: 2 row(s)
hive> CREATE DATABASE IF NOT EXISTS tripdata_andres_leal_db;
OK
Time taken: 0.057 seconds
hive> USE tripdata_andres_leal_db;
OK
Time taken: 0.024 seconds
hive>
```

```
hadoop@0e8a607bbfab: /   hadoop@0e8a607bbfab: ~/lar
hive> CREATE TABLE IF NOT EXISTS tripdata_andres_leal_db.payments(vendorID INT, tpep_pickup_datetime TIMESTAMP, payment_type STRING, total_amount DOUBLE)
    > COMMENT 'PAYMENTS TABLE'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.098 seconds
hive> CREATE TABLE IF NOT EXISTS tripdata_andres_leal_db.passengers(tpep_pickup_datetime TIMESTAMP, passenger_count INT, total_amount DOUBLE)
    > COMMENT 'PASSENGERS TABLE'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.056 seconds
hive> CREATE TABLE IF NOT EXISTS tripdata_andres_leal_db.tolls(tpep_pickup_datetime TIMESTAMP, passenger_count INT, tolls_amount DOUBLE, total_amount DOUBLE)
    > COMMENT 'TOLLS TABLE'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.062 seconds
hive> CREATE TABLE IF NOT EXISTS tripdata_andres_leal_db.congestion(tpep_pickup_datetime TIMESTAMP, passenger_count INT, congestion_surcharge DOUBLE, total_amount DOUBLE)
    > COMMENT 'CONGESTION TABLE'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.056 seconds
hive> CREATE TABLE IF NOT EXISTS tripdata_andres_leal_db.distance(tpep_pickup_datetime TIMESTAMP, passenger_count INT, trip_distance DOUBLE, total_amount DOUBLE)
    > COMMENT 'DISTANCE TABLE'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.053 seconds
hive>
```

```
hive> DESCRIBE FORMATTED passengers;
OK
# col_name              data_type               comment

tpep_pickup_datetetime  timestamp
passenger_count         int
total_amount            double

# Detailed Table Information
Database:               tripdata_andres_leal_db
Owner:                  hadoop
CreateTime:             Sun Dec 24 16:10:30 ART 2023
LastAccessTime:         UNKNOWN
Retention:              0
Location:               hdfs://172.17.0.2:9000/user/hive/warehouse/tripdata_andres_leal_db.db/passengers
Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   {\"BASIC_STATS\":\"true\"}
        comment                 PASSENGERS TABLE
        numFiles                0
        numRows                 0
        rawDataSize             0
        totalSize               0
        transient_lastDdlTime   1703445030

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.073 seconds, Fetched: 34 row(s)
hive>
```

```
hive> DESCRIBE FORMATTED distance;
OK
# col_name              data_type               comment

tpep_pickup_datetetime  timestamp
passenger_count         int
trip_distance           double
total_amount            double

# Detailed Table Information
Database:               tripdata_andres_leal_db
Owner:                  hadoop
CreateTime:             Sun Dec 24 16:10:50 ART 2023
LastAccessTime:         UNKNOWN
Retention:              0
Location:               hdfs://172.17.0.2:9000/user/hive/warehouse/tripdata_andres_leal_db.db/distance
Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   {\"BASIC_STATS\":\"true\"}
        comment                 DISTANCE TABLE
        numFiles                0
        numRows                 0
        rawDataSize             0
        totalSize               0
        transient_lastDdlTime   1703445050

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.047 seconds, Fetched: 35 row(s)
hive>
```

```
hadoop@0e8a607bbfab:~/landing$ hdfs dfs -ls /user/hadoop
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2023-12-24 20:32 /user/hadoop/.sparkStaging
drwxr-xr-x   - hadoop supergroup          0 2022-01-22 23:56 /user/hadoop/inputs
hadoop@0e8a607bbfab:~/landing$ hdfs dfs -put yellow_tripdata_2021-01.csv /user/hadoop
hadoop@0e8a607bbfab:~/landing$ hdfs dfs -ls /user/hadoop
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2023-12-24 20:32 /user/hadoop/.sparkStaging
drwxr-xr-x   - hadoop supergroup          0 2022-01-22 23:56 /user/hadoop/inputs
-rw-r--r--   1 hadoop supergroup  125981363 2023-12-24 20:33 /user/hadoop/yellow_tripdata_2021-01.csv
hadoop@0e8a607bbfab:~/landing$
```

```
>>> df = spark.read.options(delimiter=";", header=True).csv("yellow_tripdata_2021-01.csv")
>>> df.printSchema()
root
 |-- 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_am
ount,tolls_amount,improvement_surcharge,total_amount,congestion_surcharge: string (nullable = true)

>>>
```

```
>>> df = spark.read.options(header=True).csv("yellow_tripdata_2021-01.csv")
>>> df.printSchema()
root
 |-- VendorID: string (nullable = true)
 |-- tpep_pickup_datetime: string (nullable = true)
 |-- tpep_dropoff_datetime: string (nullable = true)
 |-- passenger_count: string (nullable = true)
 |-- trip_distance: string (nullable = true)
 |-- RatecodeID: string (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- PULocationID: string (nullable = true)
 |-- DOLocationID: string (nullable = true)
 |-- payment_type: string (nullable = true)
 |-- fare_amount: string (nullable = true)
 |-- extra: string (nullable = true)
 |-- mta_tax: string (nullable = true)
 |-- tip_amount: string (nullable = true)
 |-- tolls_amount: string (nullable = true)
 |-- improvement_surcharge: string (nullable = true)
 |-- total_amount: string (nullable = true)
 |-- congestion_surcharge: string (nullable = true)

>>> df2 = spark.read.options(header=True, inferSchema=True).csv("yellow_tripdata_2021-01.csv")
>>> df2.printSchema()
root
 |-- VendorID: integer (nullable = true)
 |-- tpep_pickup_datetime: string (nullable = true)
 |-- tpep_dropoff_datetime: string (nullable = true)
 |-- passenger_count: integer (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- RatecodeID: integer (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- PULocationID: integer (nullable = true)
 |-- DOLocationID: integer (nullable = true)
 |-- payment_type: integer (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- extra: double (nullable = true)
 |-- mta_tax: double (nullable = true)
 |-- tip_amount: double (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- improvement_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- congestion_surcharge: double (nullable = true)
```

```
>>> payments_df_credit_card = payments_df.filter(col("payment_type")==1)
>>> payments_df_credit_card.write.insertInto("tripdata_andres_leal_db.payments")
2023-12-24 21:44:22,198 WARN conf.HiveConf: HiveConf of name hive.metastore.local does not exist
2023-12-24 21:44:22,656 WARN session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
>>> payments_df_credit_card.show(10)
+--------+--------------------+------------+------------+
|VendorID|tpep_pickup_datetime|payment_type|total_amount|
+--------+--------------------+------------+------------+
|       1| 2021-01-01 00:43:30|           1|       51.95|
|       1| 2021-01-01 00:15:48|           1|       36.35|
|       2| 2021-01-01 00:31:49|           1|       24.36|
|       1| 2021-01-01 00:16:29|           1|       14.15|
|       1| 2021-01-01 00:26:12|           1|       18.95|
|       2| 2021-01-01 00:15:52|           1|        24.3|
|       2| 2021-01-01 00:46:36|           1|       10.79|
|       2| 2021-01-01 00:31:06|           1|       14.16|
|       2| 2021-01-01 00:17:48|           1|        10.3|
|       2| 2021-01-01 00:33:38|           1|       12.09|
+--------+--------------------+------------+------------+
only showing top 10 rows

>>>
```

```
hive> SELECT * FROM payments LIMIT 10;
OK
1       2021-01-01 00:43:30     1       51.95
1       2021-01-01 00:15:48     1       36.35
2       2021-01-01 00:31:49     1       24.36
1       2021-01-01 00:16:29     1       14.15
1       2021-01-01 00:26:12     1       18.95
2       2021-01-01 00:15:52     1       24.3
2       2021-01-01 00:46:36     1       10.79
2       2021-01-01 00:31:06     1       14.16
2       2021-01-01 00:17:48     1       10.3
2       2021-01-01 00:33:38     1       12.09
Time taken: 0.151 seconds, Fetched: 10 row(s)
hive>
```

```
>>> passengers_df_condition = passengers_df.filter( ( col("passenger_count") > 2 ) & ( col("total_amount") > 8 ) )
>>> passengers_df_condition.write.insertInto("tripdata_andres_leal_db.passengers")
>>> passengers_df_condition.show(10)
+--------------------+---------------+------------+
|tpep_pickup_datetime|passenger_count|total_amount|
+--------------------+---------------+------------+
| 2021-01-01 00:15:52|              3|        24.3|
| 2021-01-01 00:31:06|              5|       14.16|
| 2021-01-01 00:42:11|              5|         8.3|
| 2021-01-01 00:43:41|              3|         9.3|
| 2021-01-01 00:34:37|              4|        18.3|
| 2021-01-01 00:06:08|              4|        13.3|
| 2021-01-01 00:19:57|              3|        40.3|
| 2021-01-01 00:28:07|              5|        14.8|
| 2021-01-01 00:08:04|              3|       18.59|
| 2021-01-01 00:22:02|              3|       13.56|
+--------------------+---------------+------------+
only showing top 10 rows

>>>
```

hadoop@0e8a607bbfab: / | hadoop@0e8a607bbfab: ~/lar

```
hive> SELECT * FROM passengers LIMIT 10;
OK
2021-01-01 00:15:52     3       24.3
2021-01-01 00:31:06     5       14.16
2021-01-01 00:42:11     5       8.3
2021-01-01 00:43:41     3       9.3
2021-01-01 00:34:37     4       18.3
2021-01-01 00:06:08     4       13.3
2021-01-01 00:19:57     3       40.3
2021-01-01 00:28:07     5       14.8
2021-01-01 00:08:04     3       18.59
2021-01-01 00:22:02     3       13.56
Time taken: 0.174 seconds, Fetched: 10 row(s)
hive>
```

hadoop@0e8a607bbfab: / | hadoop@0e8a607bbfab: ~/la

```
>>> tolls_df_condition = tolls_df.filter( (col("tolls_amount") > 0.1) & (col("passenger_count") > 1) )
>>> tolls_df_condition.show(10)
+--------------------+---------------+------------+------------+
|tpep_pickup_datetime|passenger_count|tolls_amount|total_amount|
+--------------------+---------------+------------+------------+
| 2021-01-01 00:10:46|              2|        6.12|       33.92|
| 2021-01-01 00:37:40|              2|        6.12|       59.42|
| 2021-01-01 00:07:26|              2|        6.12|       35.92|
| 2021-01-01 00:16:22|              6|        6.12|        40.1|
| 2021-01-01 00:18:47|              3|        6.12|        54.0|
| 2021-01-01 00:14:05|              2|         2.8|        34.1|
| 2021-01-01 01:30:07|              4|        6.12|       61.42|
| 2021-01-01 01:04:32|              4|        6.12|       51.42|
| 2021-01-01 01:42:43|              2|       11.75|       12.05|
| 2021-01-01 01:22:03|              6|        6.12|       71.42|
+--------------------+---------------+------------+------------+
only showing top 10 rows

>>> tolls_df_condition.write.insertInto("tripdata_andres_leal_db.tolls")
>>>
```

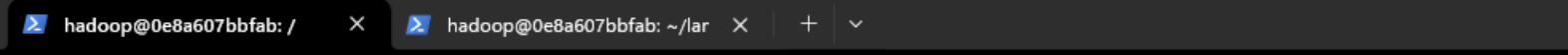
```
hive> SELECT * FROM tolls LIMIT 10;
OK
2021-01-01 00:10:46     2       6.12    33.92
2021-01-01 00:37:40     2       6.12    59.42
2021-01-01 00:07:26     2       6.12    35.92
2021-01-01 00:16:22     6       6.12    40.1
2021-01-01 00:18:47     3       6.12    54.0
2021-01-01 00:14:05     2       2.8     34.1
2021-01-01 01:30:07     4       6.12    61.42
2021-01-01 01:04:32     4       6.12    51.42
2021-01-01 01:42:43     2       11.75   12.05
2021-01-01 01:22:03     6       6.12    71.42
Time taken: 0.138 seconds, Fetched: 10 row(s)
hive>
```

```
>>> congestion_df_condition = congestion_df.filter( (col("tpep_pickup_datetime") > '2021-01-01 00:00') & (col("tpep_pickup_datetime") < '2021-01-02 00:00') )
>>> congestion_df_condition.show(10)
+-------------------+---------------+--------------------+------------+
|tpep_pickup_datetime|passenger_count|congestion_surcharge|total_amount|
+-------------------+---------------+--------------------+------------+
| 2021-01-01 00:30:10|              1|                 2.5|        11.8|
| 2021-01-01 00:51:20|              1|                 0.0|         4.3|
| 2021-01-01 00:43:30|              1|                 0.0|       51.95|
| 2021-01-01 00:15:48|              0|                 0.0|       36.35|
| 2021-01-01 00:31:49|              1|                 2.5|       24.36|
| 2021-01-01 00:16:29|              1|                 2.5|       14.15|
| 2021-01-01 00:00:28|              1|                 0.0|        17.3|
| 2021-01-01 00:12:29|              1|                 2.5|        21.8|
| 2021-01-01 00:39:16|              1|                 0.0|        28.8|
| 2021-01-01 00:26:12|              2|                 2.5|       18.95|
+-------------------+---------------+--------------------+------------+
only showing top 10 rows

>>> congestion_df_condition.write.insertInto("tripdata_andres_leal_db.congestion")
>>>
```

hadoop@0e8a607bbfab: / | hadoop@0e8a607bbfab: ~/lar

```
hive> SELECT * FROM congestion LIMIT 10;
OK
2021-01-01 00:30:10     1       2.5     11.8
2021-01-01 00:51:20     1       0.0     4.3
2021-01-01 00:43:30     1       0.0     51.95
2021-01-01 00:15:48     0       0.0     36.35
2021-01-01 00:31:49     1       2.5     24.36
2021-01-01 00:16:29     1       2.5     14.15
2021-01-01 00:00:28     1       0.0     17.3
2021-01-01 00:12:29     1       2.5     21.8
2021-01-01 00:39:16     1       0.0     28.8
2021-01-01 00:26:12     2       2.5     18.95
Time taken: 0.138 seconds, Fetched: 10 row(s)
hive>
```

```
hadoop@0e8a607bbfab: /    hadoop@0e8a607bbfab: ~/la
>>> distance_df = df.select( col("tpep_pickup_datetime").cast("timestamp"), col("passenger_count").cast("int"), col("trip_distance").cast("double"), col("total_amount").cast("double") )
>>> distance_df_condition = distance_df.filter( (col("tpep_pickup_datetime") > '2020-12-31 00:00') & (col("tpep_pickup_datetime") < '2021-01-01 00:00') & (col("passenger_count")==1) & (col
("trip_distance")>15) )
>>> distance_df_condition.show(10)
+-------------------+---------------+-------------+------------+
|tpep_pickup_datetime|passenger_count|trip_distance|total_amount|
+-------------------+---------------+-------------+------------+
| 2020-12-31 21:40:20|              1|        17.96|        53.3|
+-------------------+---------------+-------------+------------+

>>> distance_df_condition.write.insertInto("tripdata_andres_leal_db.distance")
>>>
```


hadoop@0e8a607bbfab: /
hadoop@0e8a607bbfab: ~/lar
hive> SELECT * FROM distance;
OK
2020-12-31 21:40:20 1 17.96 53.3
Time taken: 0.144 seconds, Fetched: 1 row(s)
hive>