Seoul Bike Sharing Demand

PREPARED BY
DITI V JAIN
KANISKA S

Index

DEMAND PREDICTION

Hadoop Commands - load store and manage
MapReduce functions - Python
Hive Querying - Group by and statistics
Pig - Generating reports

What can be analyzed?

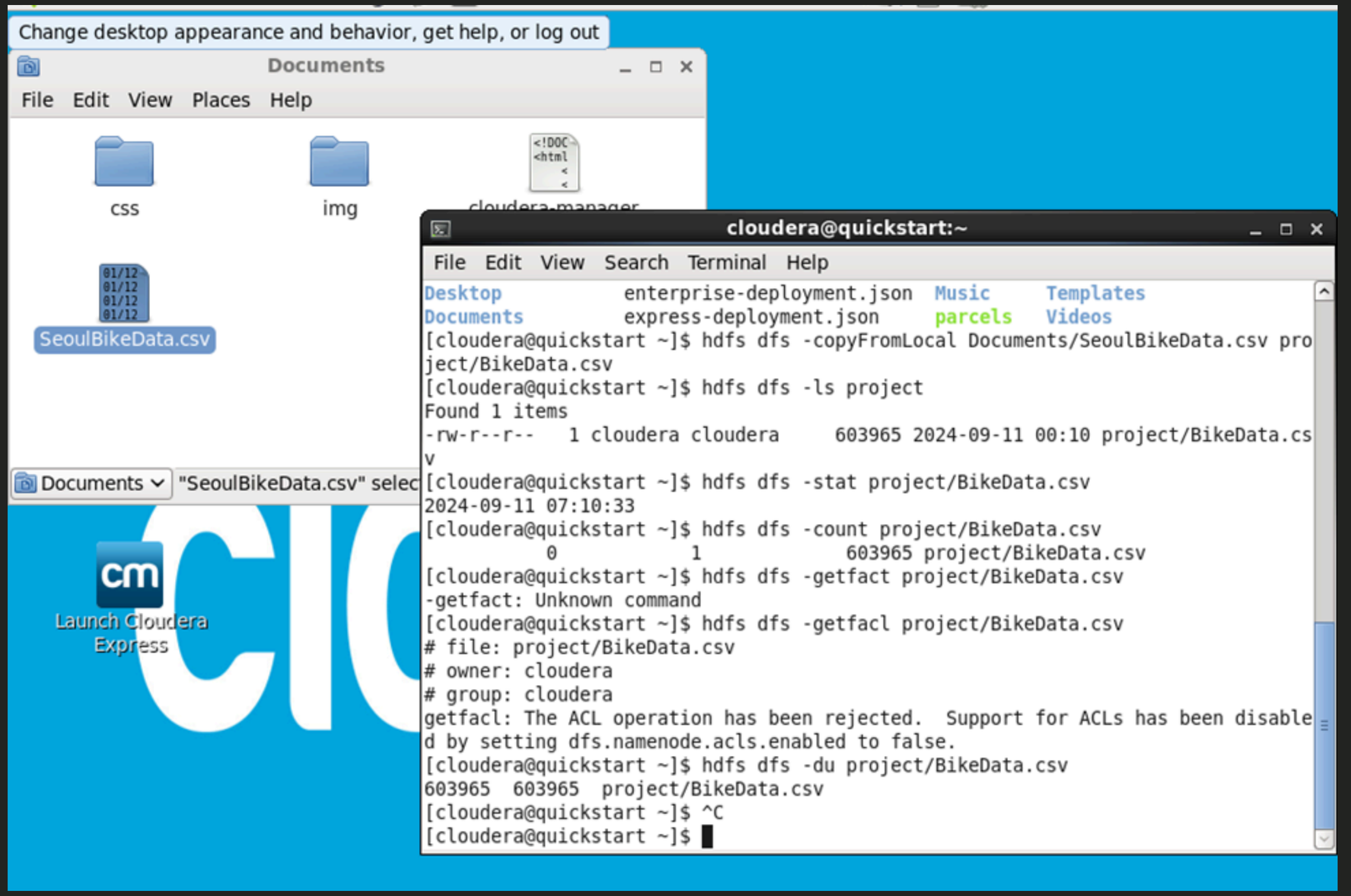
The project aims to forecast bike demand accurately to optimize fleet zent and availability.

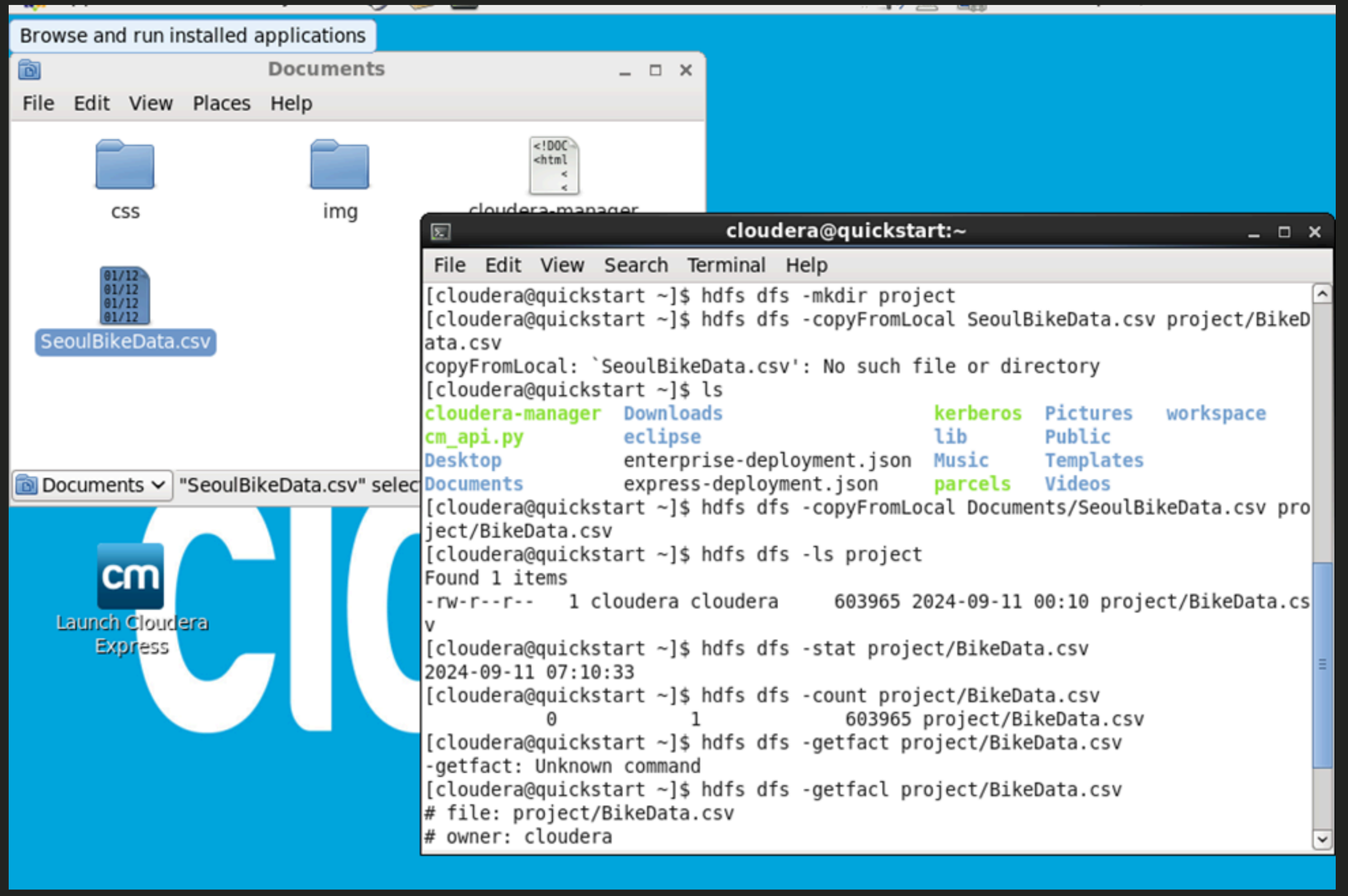
Dataset Information

1. Weather data: Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall.
2. Temporal data: Date, Time.
3. Target variable: Number of bikes rented per hour.

Hadoop commands

- Copying files from local to HDFS
- Making a directory
- Listing files
- Checking the statistics
- Checking the count
- Access control of a file


Change desktop appearance and behavior, get help, or log out
Documents
File Edit View Places Help
css
img
SeoulBikeData.csv
Documents
"SeoulBikeData.csv" selec
Launch Cloudera Express
cloudera@quickstart:~
File Edit View Search Terminal Help
Desktop enterprise-deployment.json Music Templates
Documents express-deployment.json parcels Videos
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal Documents/SeoulBikeData.csv project/BikeData.csv
[cloudera@quickstart ~]$ hdfs dfs -ls project
Found 1 items
-rw-r--r-- 1 cloudera cloudera 603965 2024-09-11 00:10 project/BikeData.csv
[cloudera@quickstart ~]$ hdfs dfs -stat project/BikeData.csv
2024-09-11 07:10:33
[cloudera@quickstart ~]$ hdfs dfs -count project/BikeData.csv
0 1 603965 project/BikeData.csv
[cloudera@quickstart ~]$ hdfs dfs -getfact project/BikeData.csv
-getfact: Unknown command
[cloudera@quickstart ~]$ hdfs dfs -getfacl project/BikeData.csv
file: project/BikeData.csv
owner: cloudera
group: cloudera
getfacl: The ACL operation has been rejected. Support for ACLs has been disabled by setting dfs.namenode.acls.enabled to false.
[cloudera@quickstart ~]$ hdfs dfs -du project/BikeData.csv
603965 603965 project/BikeData.csv
[cloudera@quickstart ~]$ ^C
[cloudera@quickstart ~]$


Browse and run installed applications
Documents
File Edit View Places Help
css
img
cloudera-manager
SeoulBikeData.csv
Documents
"SeoulBikeData.csv" selec
Launch Cloudera Express
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hdfs dfs -mkdir project
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal SeoulBikeData.csv project/BikeData.csv
copyFromLocal: `SeoulBikeData.csv': No such file or directory
[cloudera@quickstart ~]$ ls
cloudera-manager Downloads kerberos Pictures workspace
cm_api.py eclipse lib Public
Desktop enterprise-deployment.json Music Templates
Documents express-deployment.json parcels Videos
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal Documents/SeoulBikeData.csv project/BikeData.csv
[cloudera@quickstart ~]$ hdfs dfs -ls project
Found 1 items
-rw-r--r-- 1 cloudera cloudera 603965 2024-09-11 00:10 project/BikeData.csv
[cloudera@quickstart ~]$ hdfs dfs -stat project/BikeData.csv
2024-09-11 07:10:33
[cloudera@quickstart ~]$ hdfs dfs -count project/BikeData.csv
0 1 603965 project/BikeData.csv
[cloudera@quickstart ~]$ hdfs dfs -getfact project/BikeData.csv
-getfact: Unknown command
[cloudera@quickstart ~]$ hdfs dfs -getfacl project/BikeData.csv
file: project/BikeData.csv
owner: cloudera

MapReduce functions

1. Total Count Season Wise
2. Functioning Day Vs Non Fucntioning Day Bike Count
3. Average no of bikes rented for each temperature category

Total Count Season Wise

Tells us about seasonal trends in Bike Rentals - Autumn, Spring, Summer, Winter

Mapper

mapper.py

```
import sys
for line in sys.stdin:
        fields = line.strip().split(',')
        rented_count = fields[1]
        season = fields[11]
        print(('%s\t%s')%(season,rented_count))
```

Reducer

reducer.py

```
import sys
current_season = None
total_count = 0
for line in sys.stdin:
        season, count = line.strip().split('\t')
        count = int(count)
        if current_season == season:
                total_count = total_count + count
        else:
                if current_season:
                        print(current_season + "\t" + str(total_count))
                current_season = season
                total_count = count
if current_season:
        print(current_season + "\t" + str(total_count))
```

Output

```
[cloudera@quickstart Documents]$ cat SeoulBikeData.csv | python map
Autumn  1790002
Spring  1611909
Summer  2283234
Winter  487169
```

Bike rentals drop in winter and peak in summer, so more bikes should be available in summer and fewer in winter.

Functioning Day Vs Non Functioning Day Count

Tells us about Holiday vs non holiday demand for the bike count

Mapper

reducer_bike.py | mapper_bike.py | SeoulBikeData.csv

```
import sys
for line in sys.stdin:
    line = line.strip()
    columns = line.split(',')
    functioning_day = columns[12]
    rented_bike_count = columns[1]
    print(("%s\t%s")%(functioning_day,rented_bike_count))
```

Reducer

reducer_bike.py | mapper_bike.py | SeoulBikeData.csv

```
import sys
current_day = None
current_count = 0
for line in sys.stdin:
    line = line.strip()
    functioning_day, bike_count = line.split('\t')
    bike_count = int(bike_count)
    if current_day == functioning_day:
        current_count += bike_count
    else:
        if current_day:
            print(("%s\t%s")%(current_day,current_count))
        current_day = functioning_day
        current_count = bike_count
if current_day == functioning_day:
    print(("%s\t%s")%(current_day,current_count))
```

Output

```
[cloudera@quickstart ~]$ cat SeoulBikeData.csv | python mapper_bike.py | sort | python reducer_bike.py
Holiday 215895
No Holiday      5956419
```

Bike rentals are significantly higher on non-holidays (5,956,419) compared to holidays (215,895), suggesting more frequent rentals for commuting on regular workdays.

Output

```
[cloudera@quickstart ~]$ hdfs dfs -put SeoulBikeData.csv /user/cloudera
```

```
[cloudera@quickstart ~]$ hdfs dfs -put mapper_bike.py  /user/cloudera
```

```
[cloudera@quickstart ~]$ hdfs dfs -put reducer bike.py  /user/cloudera
```

```
[cloudera@quickstart ~]$ hdfs dfs -ls
Found 4 items
-rw-r--r--   1 cloudera cloudera     603965 2024-09-21 20:19 SeoulBikeData.csv
-rw-r--r--   1 cloudera cloudera        215 2024-09-21 23:17 mapper_bike.py
drwxr-xr-x   - cloudera cloudera          0 2024-09-20 00:22 project
-rw-r--r--   1 cloudera cloudera        469 2024-09-21 23:18 reducer_bike.py
```

Output

```
[cloudera@quickstart ~]$ hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-cdh5.13.0.jar -input SeoulBik
eData.csv -output output -mapper "python mapper bike.py" -reducer "python reducer bike.py"  -file mapper bike.py  -file reducer bike.py
```

```
[cloudera@quickstart ~]$ hdfs dfs -ls output
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2024-09-22 01:33 output/_SUCCESS
-rw-r--r--   1 cloudera cloudera         34 2024-09-22 01:33 output/part-00000
[cloudera@quickstart ~]$ hdfs dfs -cat output/part-00000
Holiday 215895
No Holiday      5956419
```

Average no.of bikes rented for each temperature category

Tells us about Holiday vs non holiday demand for the bike count

Mapper

mapper_temp.py

```
import sys
for line in sys.stdin:
    line = line.strip()
    columns = line.split(',')
    temperature = float(columns[3])
    rented_bike_count = columns[1]
    if temperature < 10:
         temp_category = "cold"
    elif 10 <= temperature <= 25:
        temp_category = "normal"
    else:
        temp_category = "hot"
    print(("%s\t%s")%(temp_category,rented_bike_count))
```

Reducer

reducer_bike.py | mapper_bike.py | SeoulBikeData.csv | mapper_temp.py | reducer_temp.py

```python
import sys
current_category = None
total_count = 0
total_bikes = 0

for line in sys.stdin:
    line = line.strip()
    temp_category, bike_count = line.split('\t')
    bike_count = int(bike_count)


    if current_category == temp_category:
        total_bikes += bike_count
        total_count += 1
    else:
        if current_category:
            avg_bikes = total_bikes / total_count
            print(current_category + "\t" + str(avg_bikes))
        current_category = temp_category
        total_bikes = bike_count
        total_count = 1
if current_category == temp_category:
    avg_bikes = total_bikes / total_count
    print(current_category + "\t" + str(avg_bikes))
```

Output

```
[cloudera@quickstart ~]$ hdfs dfs -put mapper_temp.py /user/cloudera
[cloudera@quickstart ~]$ hdfs dfs -put reducer_temp.py /user/cloudera
[cloudera@quickstart ~]$ hdfs dfs -ls
Found 7 items
-rw-r--r--   1 cloudera cloudera     603965 2024-09-21 20:19 SeoulBikeData.csv
-rw-r--r--   1 cloudera cloudera        215 2024-09-21 23:17 mapper_bike.py
-rw-r--r--   1 cloudera cloudera        356 2024-09-22 04:42 mapper_temp.py
drwxr-xr-x   - cloudera cloudera          0 2024-09-22 01:33 output
drwxr-xr-x   - cloudera cloudera          0 2024-09-20 00:22 project
-rw-r--r--   1 cloudera cloudera        469 2024-09-21 23:18 reducer_bike.py
-rw-r--r--   1 cloudera cloudera        633 2024-09-22 04:42 reducer_temp.py
```

```
[cloudera@quickstart ~]$ hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-cdh5.
13.0.jar -input SeoulBikeData.csv -output tempoutput -mapper "python mapper_temp.py" -reducer "python reducer_temp.py
" -file mapper temp.py -file reducer temp.py
```

Output

```
[cloudera@quickstart ~]$ hdfs dfs -ls tempout
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2024-10-08 05:19 tempout/_SUCCESS
-rw-r--r--   1 cloudera cloudera         29 2024-10-08 05:19 tempout/part-00000
[cloudera@quickstart ~]$ hdfs dfs -cat tempout/part-00000
cold    331
hot     1181
normal  871
```

Bike rentals are highest in hot weather, followed by normal conditions, and lowest in cold weather, indicating a preference for biking in warmer temperatures.

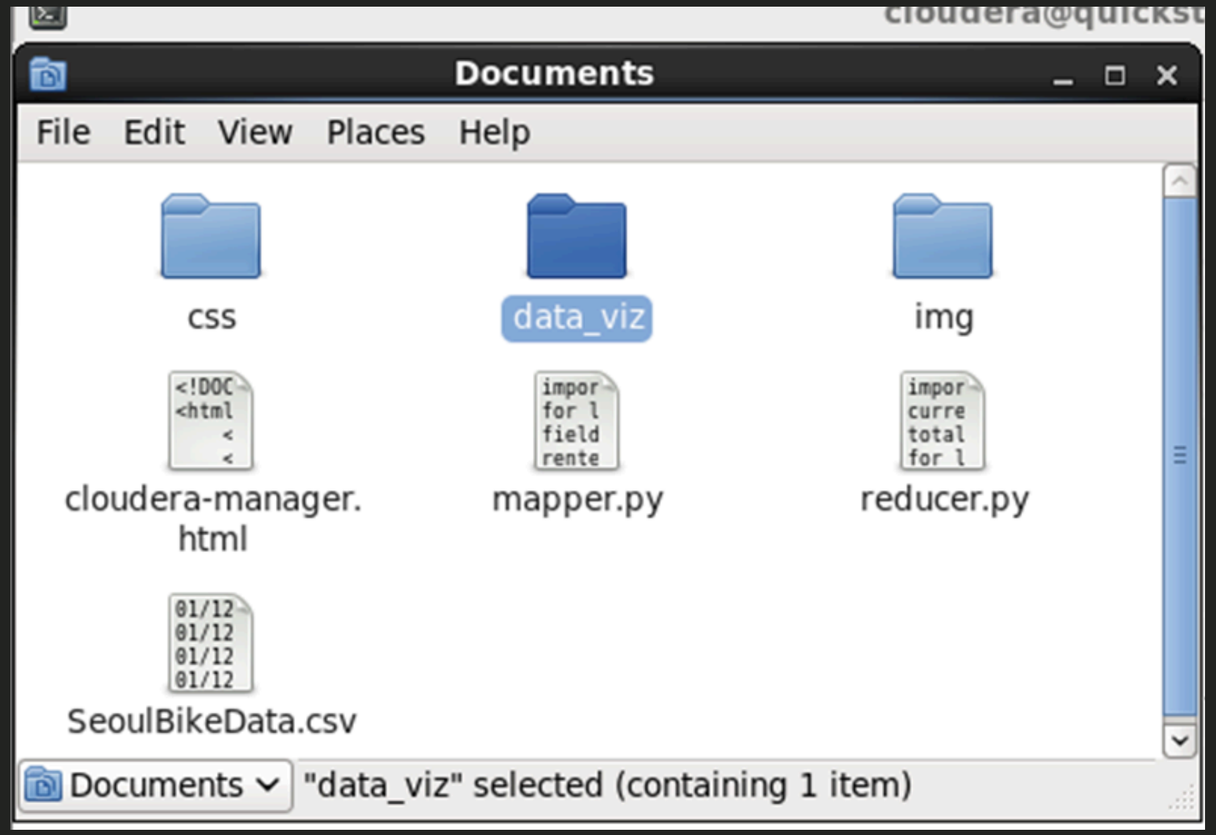
Data Visualization

Saving the hive query output in the local

```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/home/cloudera/Documents/da
ta_viz'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > SELECT seasons, AVG(rented_bike_count) AS AvgBikeCount
    > FROM bike_rentals
    > GROUP BY seasons;
Query ID = cloudera_20241017082323_b4427ca4-009b-436a-a2e2-a4b4f610
1c17
Total jobs = 1
```

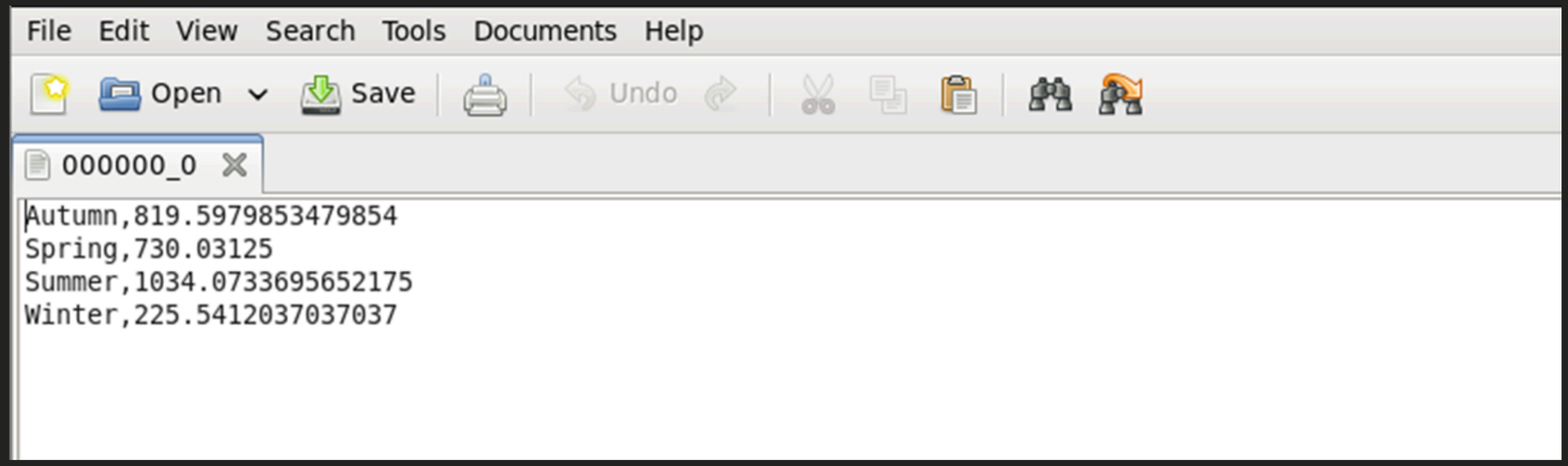
Data Visualization

Locating the folder



Data Visualization

Seeing the output csv

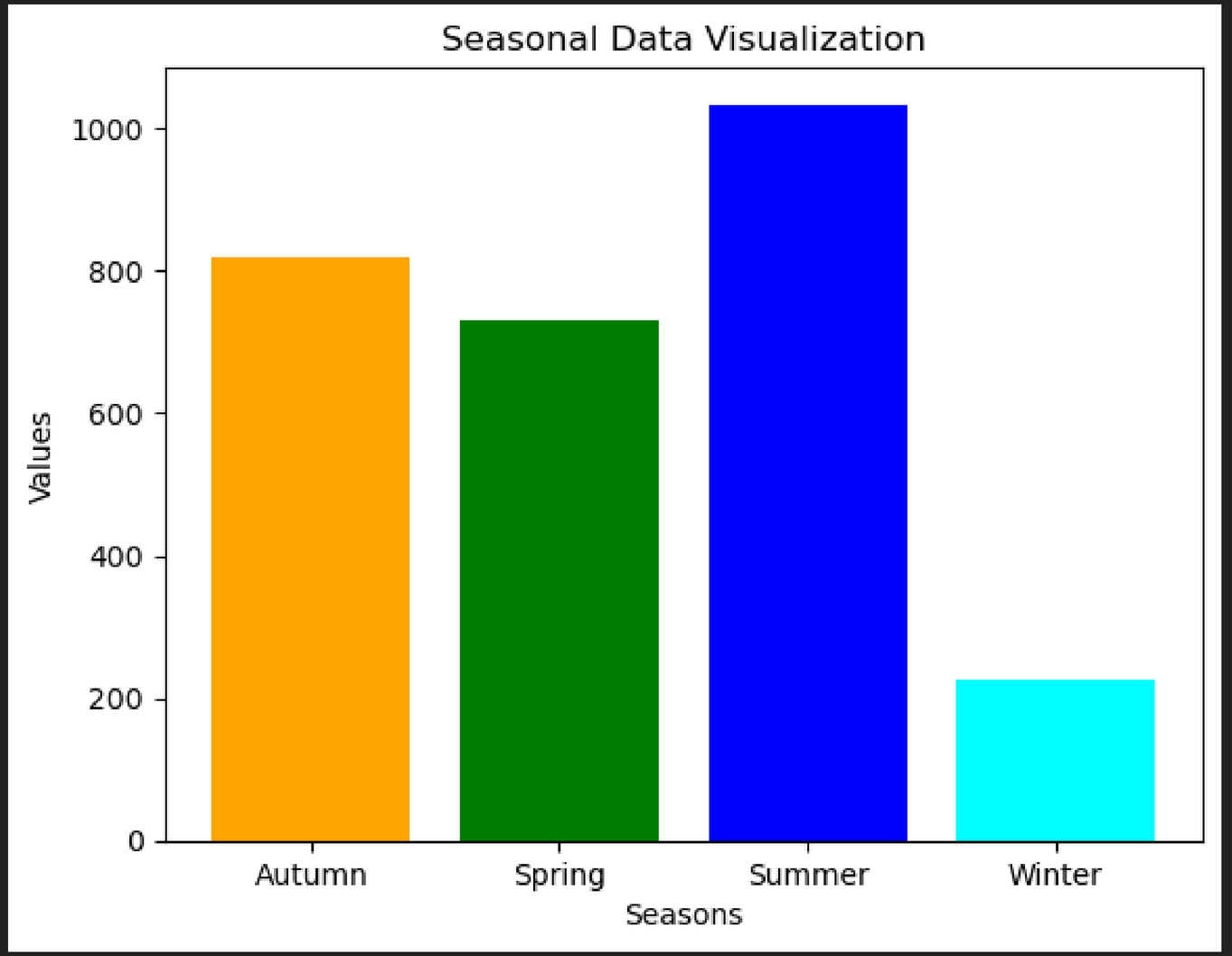


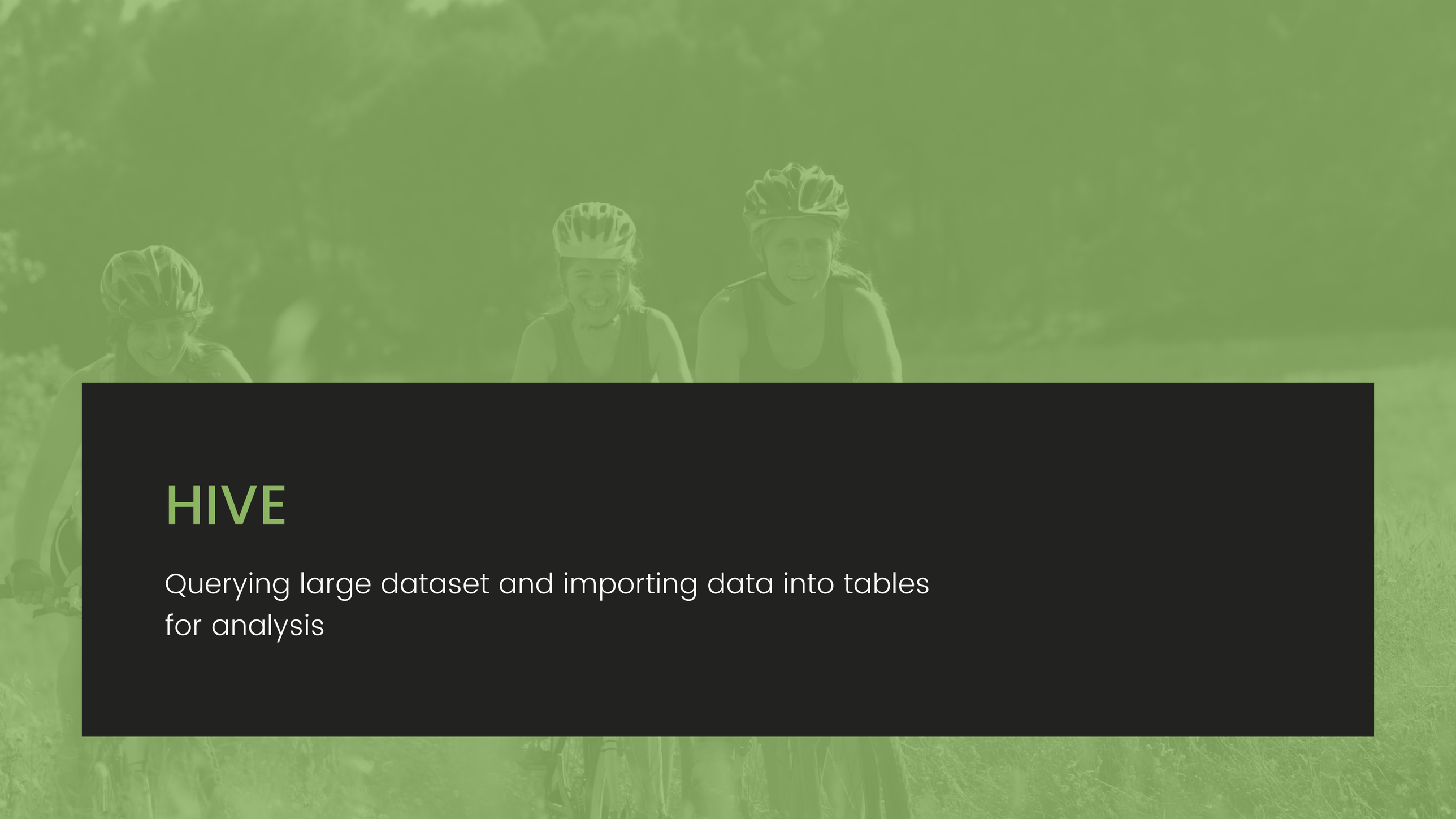
Data Visualization

Program for visualizing the data

```
import matplotlib.pyplot as plt
import csv

seasons = []
values = []

with open('seasons_data.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        seasons.append(row[0])
        values.append(float(row[1]))

plt.bar(seasons, values, color=['orange', 'green', 'blue', 'cyan'])

plt.xlabel('Seasons')
plt.ylabel('Values')
plt.title('Seasonal Data Visualization')

plt.show()
```

Seasonal Bike Rental Count





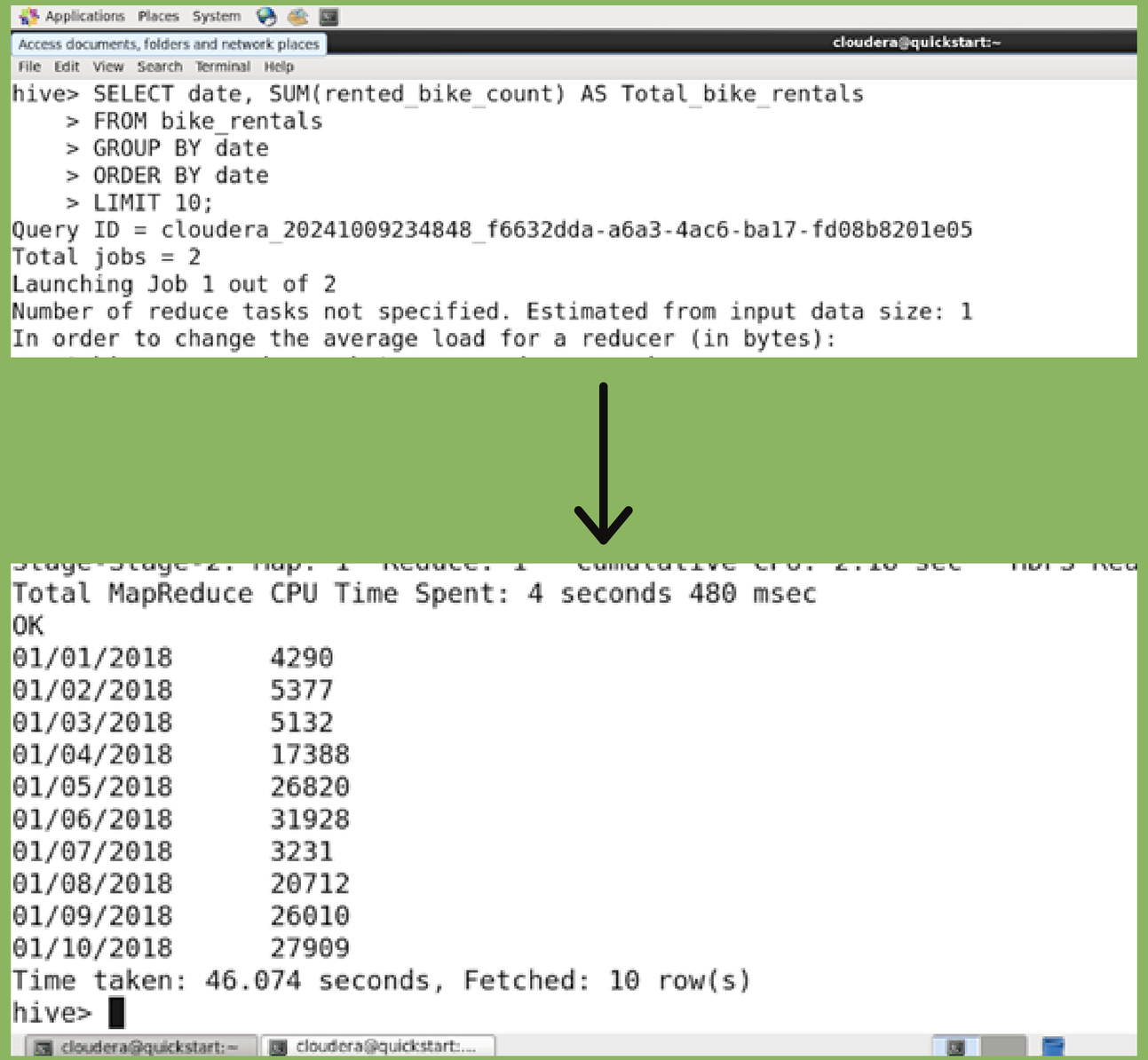
HIVE

Querying large dataset and importing data into tables for analysis

```
hive> create table bike_rentals(date string, rented_bike_count int,hour int,temperature double,humidity int, wind_speed double,visibility int,
 dew_point_temperature double, solar_radiation double,rainfall int,snowfall int,seasons string,holiday string, functioning_day string)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.114 seconds
hive> LOAD DATA LOCAL INPATH 'SeoulBikeData.csv' OVERWRITE INTO TABLE bike_rentals;
Loading data to table default.bike_rentals
Table default.bike_rentals stats: [numFiles=1, numRows=0, totalSize=603965, rawDataSize=0]
OK
Time taken: 0.344 seconds
```

Creating a table, loading dataset

Total Bike Rentals per Day



Total Bike Rentals per Hour

```
FAILED: SemanticException [Error 10025]: Line 1:7 Expression not in GROUP BY key
hive> SELECT hour, SUM(rented_bike_count) AS Total_bike_rentals
    > FROM bike_rentals
    > GROUP BY hour
    > ORDER BY hour
    > LIMIT 10;
Query ID = cloudera_20241009234444_8348f49c-548e-4427-bd89-f06449787460
Total jobs = 2
Launching Job 1 out of 2
```

```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 2.42 sec
Total MapReduce CPU Time Spent: 5 seconds 0 msec
OK
0       197633
1       155557
2       110095
3       74216
4       48396
5       50765
6       104961
7       221192
8       370731
9       235784
Time taken: 50.521 seconds, Fetched: 10 row(s)
hive>
```

Impact of Weather on Bike Rentals

```
hive> SELECT Temperature, Humidity, Wind_speed, Rainfall, Snowfall,
    > SUM(Rented_Bike_Count) AS Total_Bike_Rentals
    > FROM bike_rentals
    > GROUP BY Temperature, Humidity, Wind_speed, Rainfall, Snowfall
    > LIMIT 10;
Query ID = cloudera_20241011012525_64e50a3e-5c27-4049-a1c5-14525e07a16c
Total jobs = 1
Launching Job 1 out of 1
```

```
OK
temperature     humidity        wind_speed      rainfall        snowfall        total_bike_ren
-17.8   40      2.5     0       0       322
-17.5   37      3.8     0       0       149
-17.5   38      3.4     0       0       142
-17.4   40      1.6     0       0       64
-16.9   38      2.3     0       0       36
-16.5   33      3.4     0       0       96
-16.4   32      3.3     0       0       349
-16.4   36      1.5     0       0       24
-16.2   30      1.6     0       0       364
-16.2   33      2.2     0       0       165
Time taken: 34.148 seconds, Fetched: 10 row(s)
```

Seasonal Trends

```
hive> SELECT seasons, SUM(rented_bike_count) AS total_bike_rentals
    > FROM bike_rentals
    > GROUP BY seasons
    > LIMIT 10;
Query ID = cloudera_20241011024545_11118d30-1202-40fd-9a9b-e9d0c1cfb1f0
Total jobs = 1
Launching Job 1 out of 1
```

```
OK
seasons total_bike_rentals
Autumn  1790002
Spring  1611909
Summer  2283234
Winter  487169
Time taken: 23.979 seconds, Fetched: 4 row(s)
```

Functioning vs non functioning

```
hive> SELECT functioning_day, SUM(rented_bike_count) AS total_bike_rentals
    > FROM bike_rentals
    > GROUP BY functioning_day
    > LIMIT 10;
Query ID = cloudera_20241011034848_117f6f78-895c-406e-ace0-362107a97e56
Total jobs = 1
Launching Job 1 out of 1
```

```
OK
functioning_day total_bike_rentals
No      0
Yes     6172314
Time taken: 23.838 seconds, Fetched: 2 row(s)
```

Pig – Filter & Group

```
grunt> bike = LOAD 'SeoulBikeData.csv' USING PigStorage(',')  AS (Date:chararray,Rented_Bike_Count:int,Hour:int,Temperature:
float,Humidity:int,Wind_Speed:float,Visibility:int,Dew_Point_Temperature:float,Solar_Radiation:float,Rainfall:float,Snowfall
:float,Seasons:chararray,Holiday:chararray,Functioning_Day:chararray);
grunt> bike_data = LIMIT bike 10;
grunt> DUMP bike_data;
```

```
(Date,,,,,,,,,,Seasons,Holiday,Functioning Day)
(01/12/2017,78,4,-6.0,36,2.3,2000,-18.6,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,100,5,-6.4,37,1.5,2000,-18.7,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,107,3,-6.2,40,0.9,2000,-17.6,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,173,2,-6.0,39,1.0,2000,-17.7,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,181,6,-6.6,35,1.3,2000,-19.5,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,204,1,-5.5,38,0.8,2000,-17.6,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,254,0,-5.2,37,2.2,2000,-17.6,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,460,7,-7.4,38,0.9,2000,-19.3,0.0,0.0,0.0,Winter,No Holiday,Yes)
(01/12/2017,930,8,-7.6,37,1.1,2000,-19.8,0.01,0.0,0.0,Winter,No Holiday,Yes)
grunt>
```

Pig – Filter

```
grunt> functioning_days = FILTER bike BY Functioning_Day == 'Yes';
grunt> DUMP functioning_days;
```

```
(30/11/2018,740,13,7.1,24,2.8,1838,-12.1,1.83,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,761,14,7.8,20,2.2,2000,-13.8,1.67,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,768,15,7.0,20,3.3,1994,-14.4,1.21,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,837,16,7.2,23,1.5,1945,-12.6,0.72,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,1047,17,6.0,29,2.1,1877,-10.7,0.23,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,1384,18,4.7,34,1.9,1661,-9.8,0.0,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,1003,19,4.2,34,2.6,1894,-10.3,0.0,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,764,20,3.4,37,2.3,2000,-9.9,0.0,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,694,21,2.6,39,0.3,1968,-9.9,0.0,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,712,22,2.1,41,1.0,1859,-9.8,0.0,0.0,0.0,Autumn,No Holiday,Yes)
(30/11/2018,584,23,1.9,43,1.3,1909,-9.3,0.0,0.0,0.0,Autumn,No Holiday,Yes)
```

Pig – Group

```
grunt> season_gp = GROUP bike BY Seasons;
grunt> avg_rented_bikes = FOREACH season_gp GENERATE group AS season, AVG(bike.Rented_Bike_Count) AS avg_rented_bikes;
grunt> DUMP avg_rented_bikes;
```

```
(Autumn,819.5979853479854)
(Spring,730.03125)
(Summer,1034.0733695652175)
(Winter,225.5412037037037)
(Seasons,)
```