Question 1: Write Hive and SparkSQL queries to create a table called used_cars from data and let it infer the schema in SparkSQL and for Hive define the schema yourself.

For Hive:

Firstly, In Hadoop local directory, Cars.zip file is downloaded from the given link on assignment webpage. Then, data is stored in directory of Hadoop HDFS. After that, as per observation of data, schema is created for table in Hive. After that from HDFS, data is transferred in the schema.

```
Question 1: Part 1/2: Creating Schema and table
wget -q https://github.com/tofighi/BigData/blob/main/datasets/cars/cars.zip?raw=true -O cars.zip
ls -lah cars.zip
 -rw-r--r-- 1 zeppelin zeppelin 89M Feb 20 06:55 cars.zip
 Took 3 sec. Last updated by anonymous at February 20 2022, 1:55:06 AM. (outdated)
  unzip cars.zip && rm cars.zip
 hadoop fs -mkdir /user/assignmentlhive
hadoop fs -put cars.csv /user/assignmentlhive
hadoop fs -ls -h /user/assignmentlhive
replace carsArch.icvsev:? [cya]ress.,z i[pn ]o, [A]ll, [N]one, [r]ename: NULL (EOF or read error, treating as "[N]one" ...) mkdir: '/user/assignmentlhive': File exists
 -rw-r--r-- 2 zeppelin hadoop 400.0 M 2022-02-20 06:55 /user/assignment1hive/cars.csv
 -rw-r--r 2 zeppelin hadoop 63.0 K 2022-02-19 15:42 /user/assignment1hive/carstrial.csv
Took 10 sec. Last undated by anonymous at February 20 2022, 1:55:20 AM
 %Hive create database if not exists cars_db_hive;
 use cars_db_hive;

create table if not exists used_cars (maker string, model string, mileage int, manufacture_year int, engine_displacement int, engine_power int, body_type string, color_slug string, stk_year int, transmission string, door_count int, seat_count int, fuel_type string, date_created string, date_last_seen string, price_eur float)

row format delimited fields terminated by '.' lines terminated by '\n' stored as TEXTFILE tblproperties("skip.header.line.count"="1");
Query executed successfully. Affected rows : -1
        Question 1 (Part 2) loading data in Schema
        load data inpath '/user/assignment1hive/cars.csv' into table cars db hive.used cars;
        Query executed successfully. Affected rows : -1
         Took 1 sec. Last updated by anonymous at February 20 2022, 1:55:38 AM
```

For Spark:

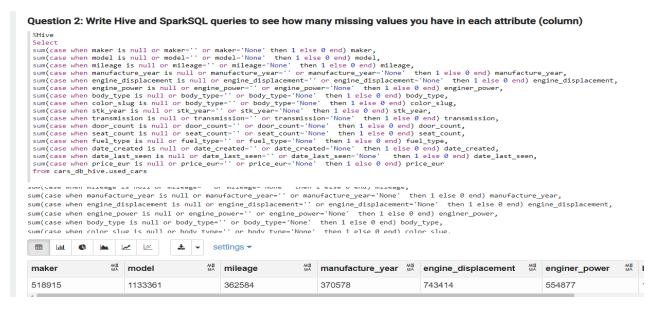
Cars.csv is downloaded from given link. Firstly, data is stored in directory of Hadoop HDFS. For Spark, we don't need to make schema as it is automatically making table as per data schema.

SPARK Question 1 (Part 1 & Part 2) %sh wget -q https://github.com/tofighi/BigData/blob/main/datasets/cars.zip?raw=true -O cars.zip unzip cars.zip && rm cars.zip hadoop fs -mkdir /user/assignment1spark hadoop fs -put cars.csv /user/assignment1spark hadoop fs -ls -h /user/assignment1spark Archive: cars.zip replace cars.csv? [y]es, [n]o, [A]11, [N]one, [r]ename: NULL (EOF or read error, treating as "[N]one" ...) mkdir: `/user/assignment1spark': File exists put: `/user/assignment1spark/cars.csv': File exists Took 8 sec. Last updated by anonymous at February 20 2022, 3:38:48 AM. (outdated) %spark.sql use cars_db_spark; Create table if not exists used_cars USING CSV OPTIONS (path "/user/assignment1spark/cars.csv", header "true" , inferSchema "true") Took 0 sec. Last updated by anonymous at February 20 2022, 3:42:10 AM.

Question 2: Write Hive and SparkSQL queries to see how many missing values you have in each attribute (column)

For Hive:

Here I considered 'null', blank and 'None' as missing values. In column name: 'stk_year', almost values are 'None'. Results are as per below.



For Spark:

SPARK Question 2



Final answer from both Hive and Spark:



Question 3: Write Hive and SparkSQL queries to create a new table called clean_used_cars from used_cars with the following conditions:

- o Drop the columns with more than 50% missing values (NULL or empty) (you do not need to automate this step, you can find them, then exclude them while creating the clean dataset)
- o The manufacture year between 2000 and 2017 including 2000 and 2017
- o Both maker and model should exist in the row
- o The price range is from 3000 to 2000,000 (3000 \leq price \leq 2000,000)

For Hive:

Question 3: Part 1/2 Drop the columns with more than 50% missing values (NULL or empty)

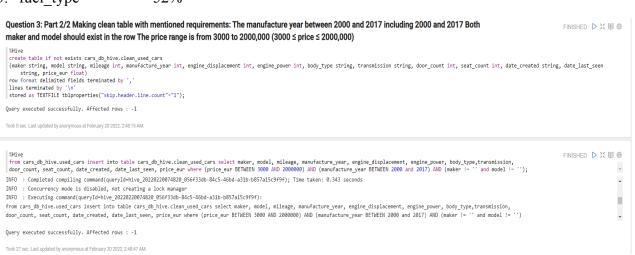


Took 26 sec. Last updated by anonymous at February 20 2022, 2:12:54 AM. (outdated)

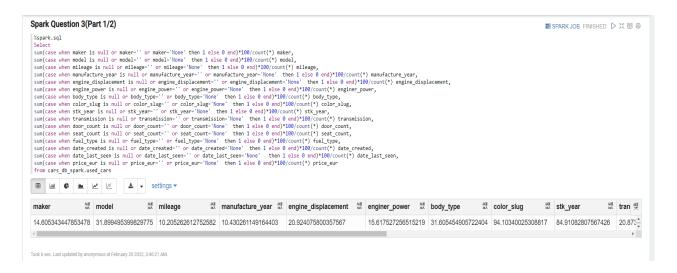
As per above result, there are total 3 columns which have more than 50% null/none values. Therefore, in part 2, schema table is created without those columns. Then, data is transferred to clean used cars table from used cars table.

Name of the columns which have more than 50% null/none value:

1: "color_slug" =94.10% 2: "stk_year =84.91% 3: "fuel_type" = 52%"



For Spark:



As per above result, there are total 3 columns which have more than 50% null/none values. Therefore, in part 2, clean_used_cars table is created by removing those columns from used_cars table.

Name of the columns which have more than 50% null/none value:

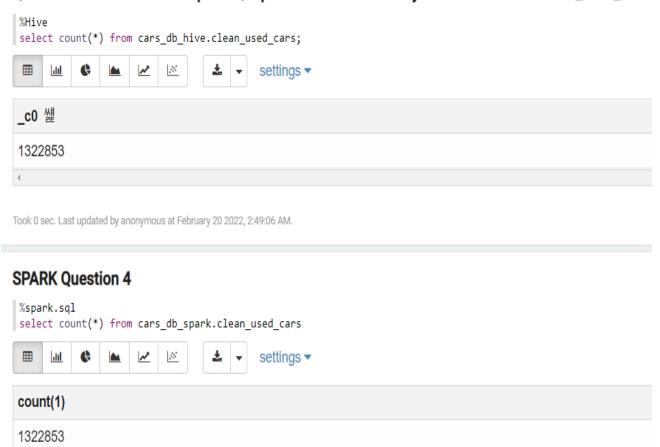
```
1: "color_slug" =94.10%
2: "stk_year =84.91%
3: "fuel type" = 52%"
```



Question 4: Write Hive and SparkSQL queries to find how many records remained clean_used_cars

For Hive and Spark, result is same as per below. There is total 1322853 records in clean_used_cars table.

Question 4: Write Hive and SparkSQL queries to find how many records remained clean_used_cars

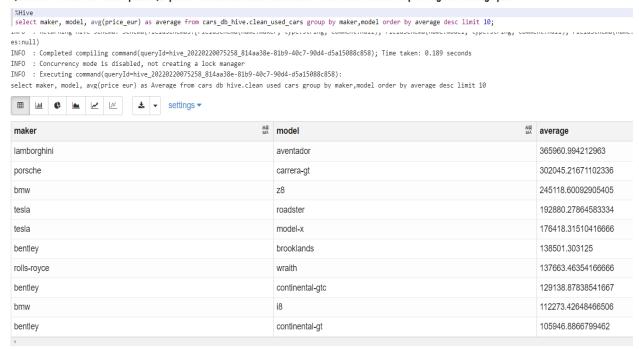


Question 5: Write Hive and SparkSQL queries to find the make and model for the cars with the top 10 highest average price:

Here as per below screenshots, results are same for both Hive and Spark. Avg. of 'price_eur', groupby and orderby (in descending order) functions are used for the result. The range of result (average in EUR) is from 365960.99 to 105946.89.

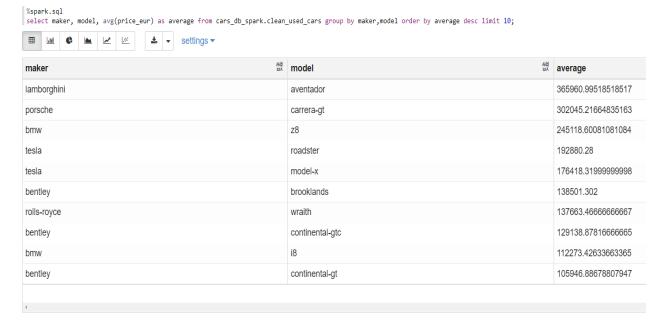
For Hive:

Question 5: Write Hive and SparkSQL queries to find the make and model for the cars with the top 10 highest average price



For Spark:

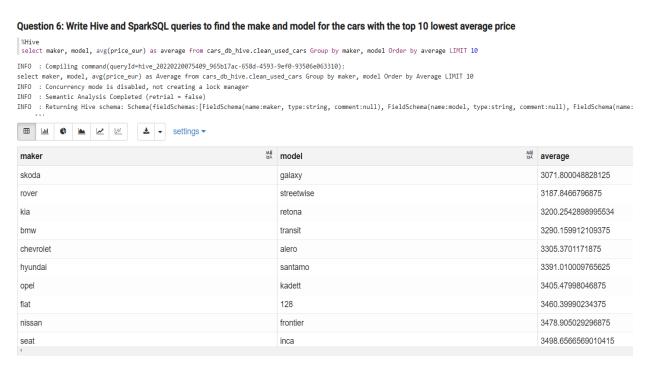
SPARK Question 5



Question 6: Write Hive and SparkSQL queries to find the make and model for the cars with the top 10 lowest average price

Here as per below screenshots, results are same for both Hive and Spark. Avg. of 'price_eur', groupby and orderby (in ascending order) functions are used for the result. The range of result (average in EUR) is from 3071.80 to 3498.66.

For Hive:



Took 20 sec. Last updated by anonymous at February 20 2022, 2:54:29 AM. (outdated)

For Spark:

SPARK Question 6 %spark.sql

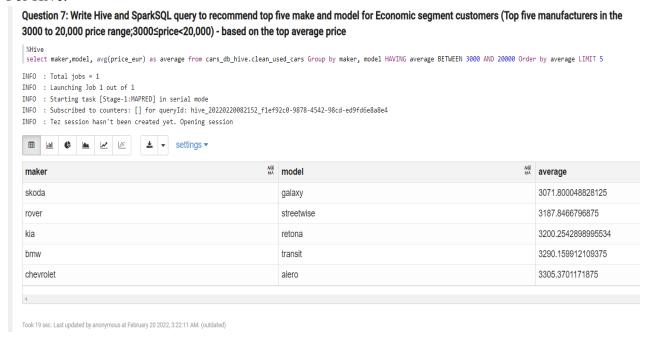


Took 2 sec. Last updated by anonymous at February 20 2022, 3:47:51 AM. (outdated

Question 7: Write Hive and SparkSQL query to recommend top five make and model for Economic segment customers (Top five manufacturers in the 3000 to 20,000 price range;3000≤price<20,000) - based on the top average price

Here, Hive and Spark queries are designed to get cars recommendation list for economic segment customers. The recommended results are based on grouping the maker and model in range of 3000 to 20000 price category. The top 5 range of the result (avg. price in EUR) is from 3071.8 to 3305.37 for economic segment customers. Results from both Hive and Spark queries are same.

For Hive:



For Spark:

SPARK Question 7

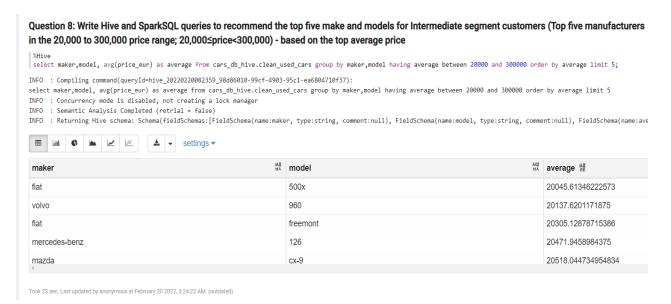


Took 2 sec. Last updated by anonymous at February 20 2022, 3:48:12 AM.

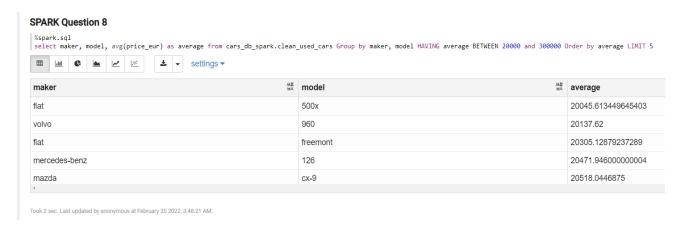
Question 8: Write Hive and SparkSQL queries to recommend the top five make and models for Intermediate segment customers (Top five manufacturers in the 20,000 to 300,000 price range; 20,000≤price<300,000) - based on the top average price

Here, Hive and Spark queries are designed to get cars recommendation list for intermediate segment customers. The recommended results are based on grouping the maker and model in range of 20,000 to 300,000 price category. The top 5 range of the result (avg. price in EUR) is from 20,045.61 to 20,518.04 for Intermediate segment customers. Results from both Hive and Spark queries are same.

For Hive:



For Spark:



Question 9: Write Hive and SparkSQL queries to recommend the top five make and models for the Luxury segment customers (Top five manufacturers in the 300,000 to 2000,000 price range; 300,000≤price<2000,000) - based on the top average price

For Hive and Spark:

Here, Hive and Spark queries are designed to get cars recommendation list for Luxury segment customers. The recommended results are based on grouping the maker and model in range of 300,000 TO 2000,000 price category. We are getting only two cars for the recommendation for luxury segment customers. Results from both Hive and Spark queries are the same as per below.

Question 9: Write Hive and SparkSQL queries to recommend the top five make and models for the Luxury segment customers (Top five manufacturers in the 300,000 to 2000,000 price range; 300,000≤price<2000,000) - based on the top average price



Took 1 sec. Last updated by anonymous at February 20 2022, 3:48:43 AM. (outdated)

Final Answers:

- 1. Number of records in used_cars table 3552912
- 2. Number of records in clean_used_cars table 1322853
- 3. Make and model for the cars with the top 10 highest average prices and their average price

maker	model	average

lamborghini	aventador	365960.995
porsche	carrera-gt	302045.22
bmw	z8	245118.60
tesla	roadster	192880.28
tesla	model-x	176418.32
bentley	brooklands	138501.30
rolls-royce	wraith	137663.47
bentley	continental-gtc	129138.88
bmw	i8	112273.43
bentley	continental-gt	105946.89

4. Make and model for the cars with the top 10 lowest average prices and their average price

maker	model	average
skoda	galaxy	3071.8
rover	streetwise	3187.847
kia	retona	3200.254
bmw	transit	3290.16
chevrolet	alero	3305.37
hyundai	santamo	3391.01
opel	kadett	3405.48
fiat	128	3460.4
nissan	frontier	3478.905
seat	inca	3498.657

5. Write the name of the top five make and models for Economic segment customers

maker	model
skoda	galaxy

rover	streetwise
kia	retona
bmw	transit
chevrolet	alero

6. Write the name of the top five make and models for Intermediate segment customers

maker	model
fiat	500x
volvo	960
fiat	freemont
mercedes-benz	126
mazda	cx-9

7. Write the name of the top five make and models for Luxury segment customers

maker	model
porsche	carrera-gt
lamborghini	aventador