

MPA-1 Big data for Machine Learning

Group-15

Tania Khan

V S Shabarish

Vinod A

Submitted on

01-12-2023

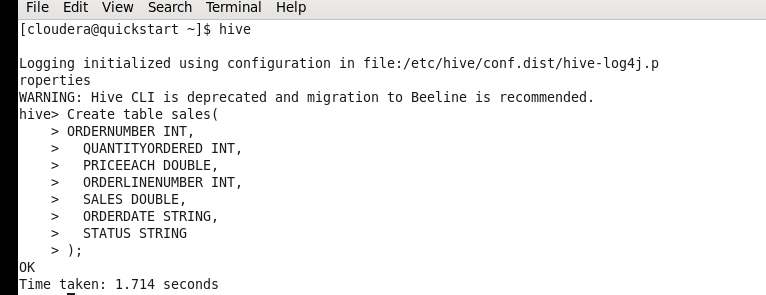
# Write shell commands for the following.

## 1.Create a partitioned table sales partitioned by year and month. (2 marks)

## 

To Create a partitioned table, we first need to initialize a shell window with hive. we type the command “hive” to create the hive environment. We then need to create a table with required

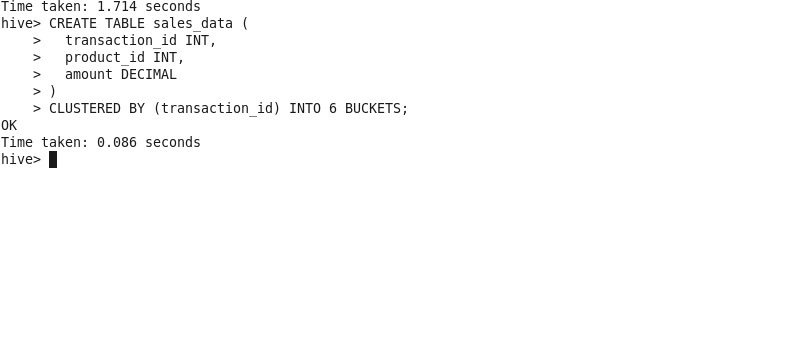
The command will be written in shell as shown in the below image.



***Output:*** This code creates a partitioned table named "sales" in Hive. The table has the following columns: ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, ORDERDATE, and STATUS. The table is partitioned by the YEAR and MONTH columns.

The code also inserts 10 sample values into the table using the INSERT INTO statement. Each value represents a row in the table and includes the corresponding values for the columns. The YEAR and MONTH values are specified as part of the PARTITION clause.

## 2.a) Write a HiveQL query to create a table named sales\_data with columns transaction\_id, product\_id, and amount clustered into 6 buckets based on transaction\_id. (2 marks)

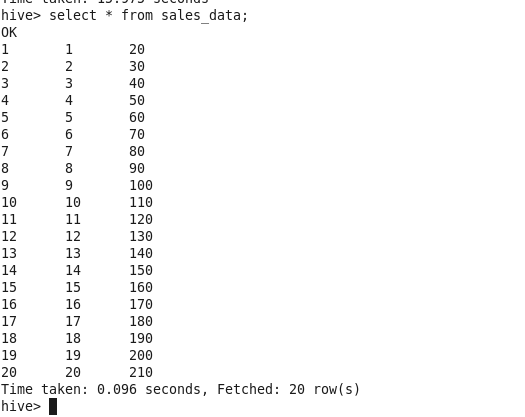


Output: This HiveQL query creates a table named sales\_data with three columns: transaction\_id, product\_id, and amount. The transaction\_id column is used as the clustering column, and the table is clustered into 6 buckets based on the values in the transaction\_id column.

## 2. b) Insert data into the sales\_data table you created, inserting records for transactions with IDs 1 to 100, products ranging from 1 to 20, and random transaction amounts. (3 marks)

To insert data into the sales\_data table we need to use the insert query and pass appropriate values for each column.





Output: This HiveQL query inserts 20 data into sales\_data with three columns: transaction\_id, product\_id, and amount.

## 3. Write a Sqoop command to import data from a MySQL database table named "employees" into the Hadoop Distributed File System (HDFS). The data should be saved in a directory named "employee\_data" in HDFS. (3 marks)

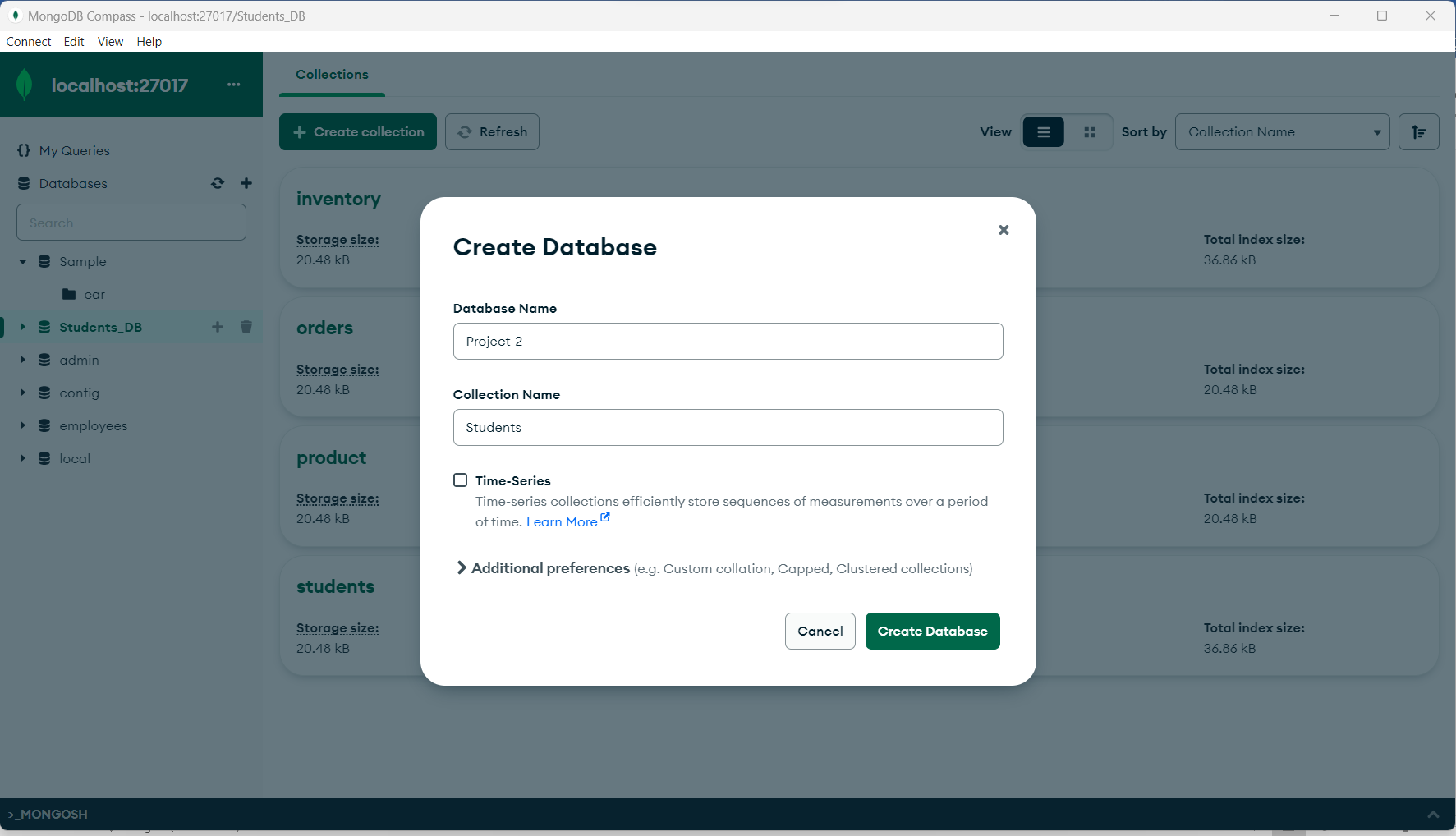
To import data from mysql to HDFS we use the below sqoop command.

## 

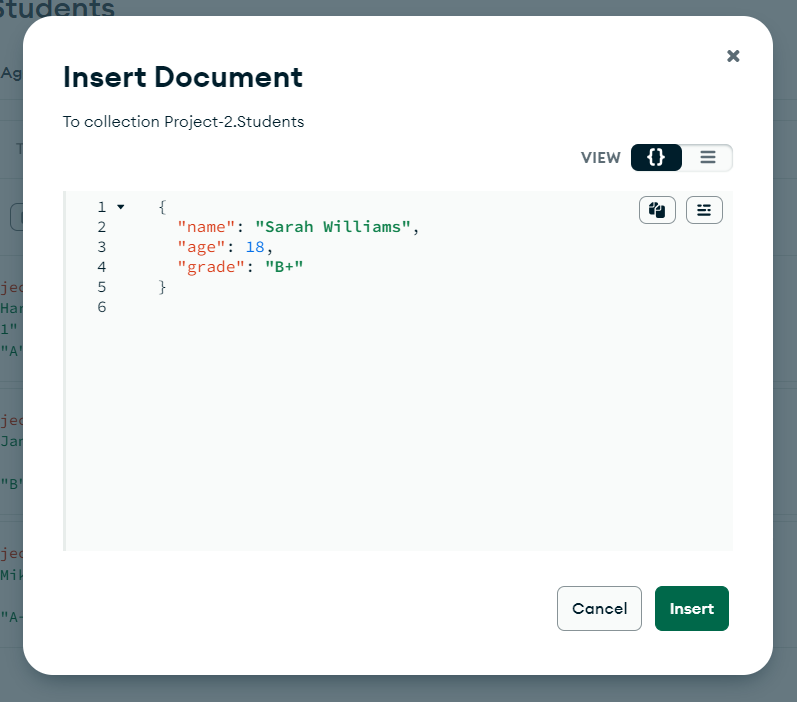
Output: data is imported from a MySQL database table named "employees" into the Hadoop Distributed File System (HDFS). The data should be saved in a directory named "employee\_data" in HDFS

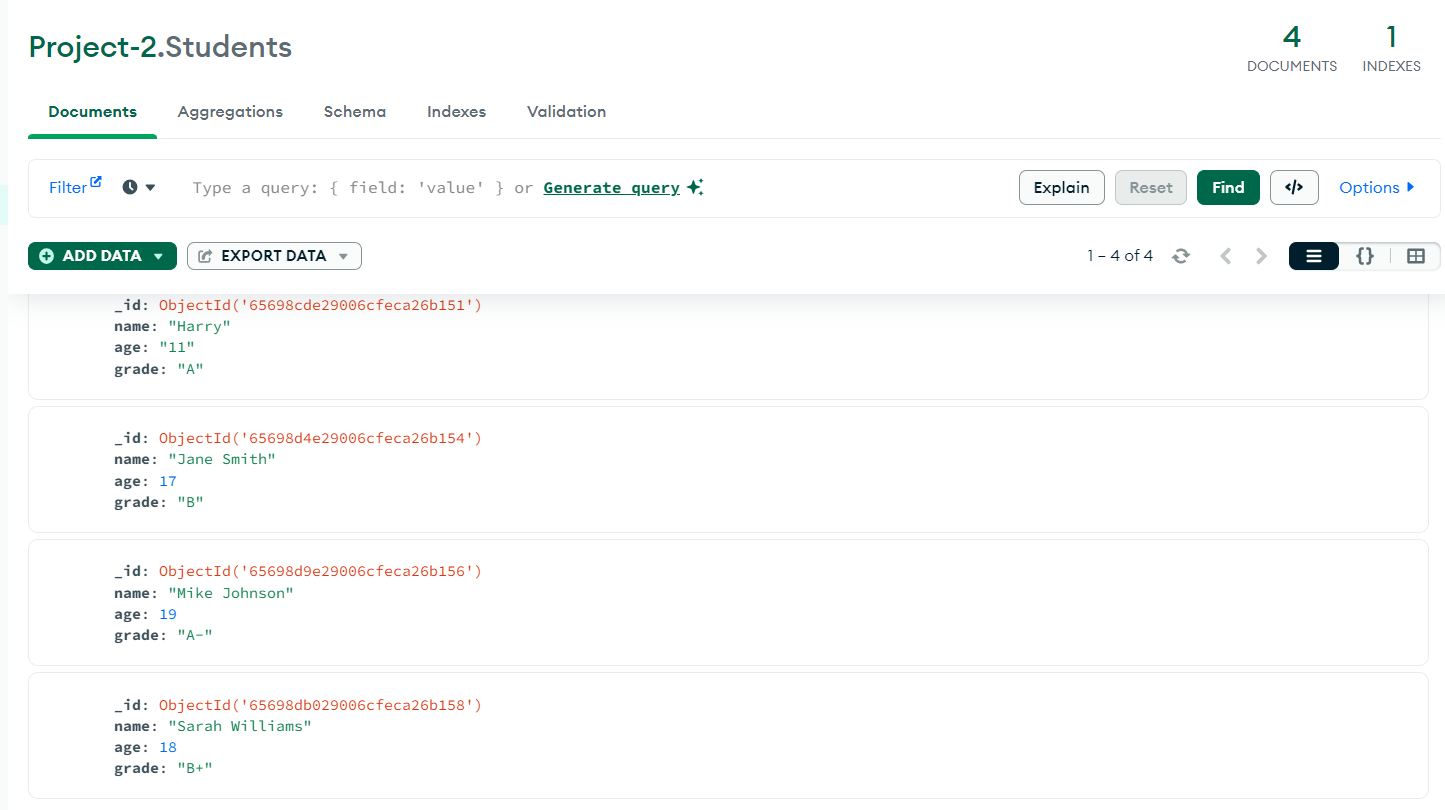
## 4. Using MongoDB Compass, create a new document in a collection named "students" with the following fields: "name," "age," and "grade". (2 marks)

To create a new collection we first need to create a database or add a collection to existing database as shown in the image below.



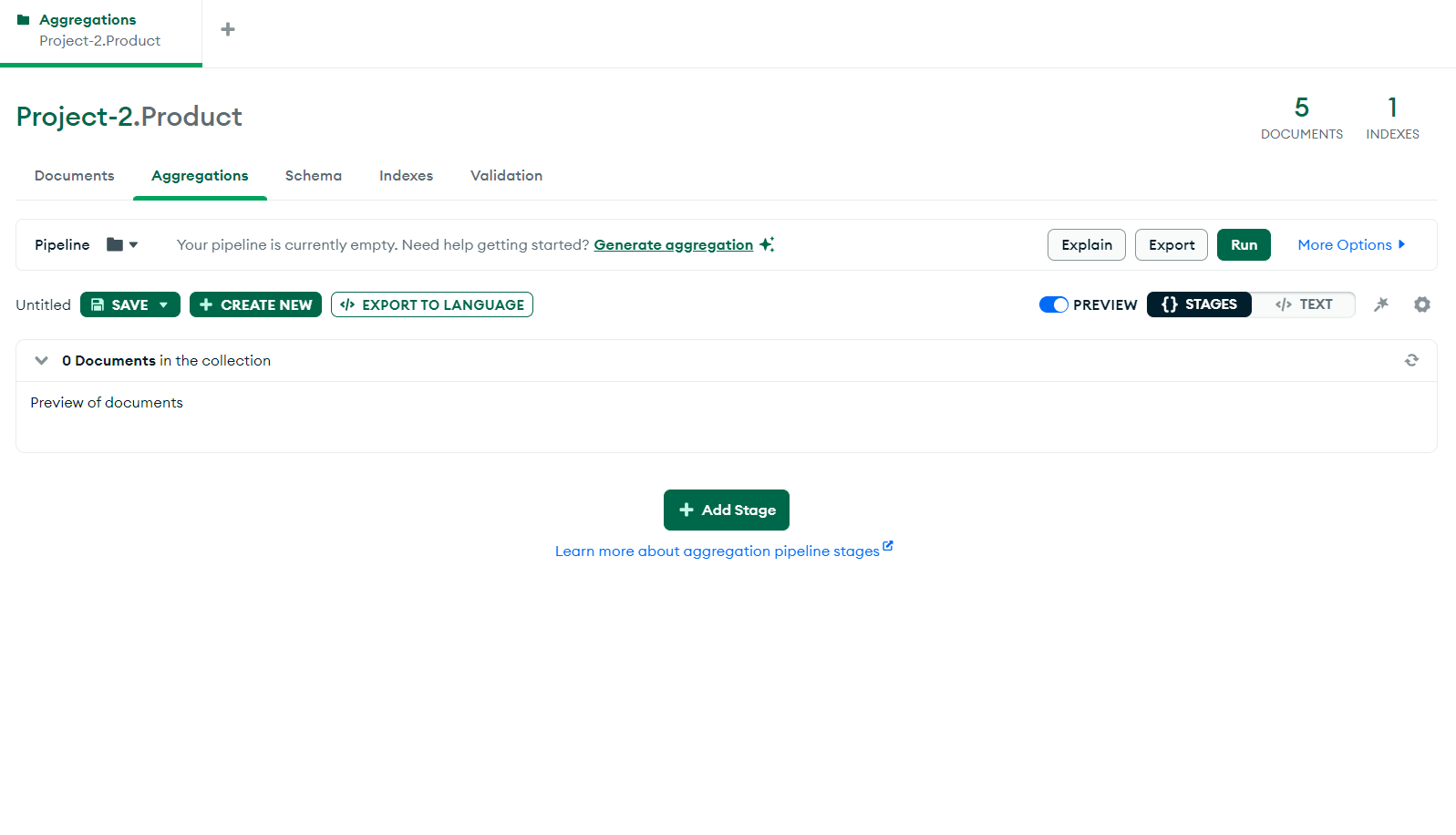
Once the collection is added to the database we can now insert documents into the collection.





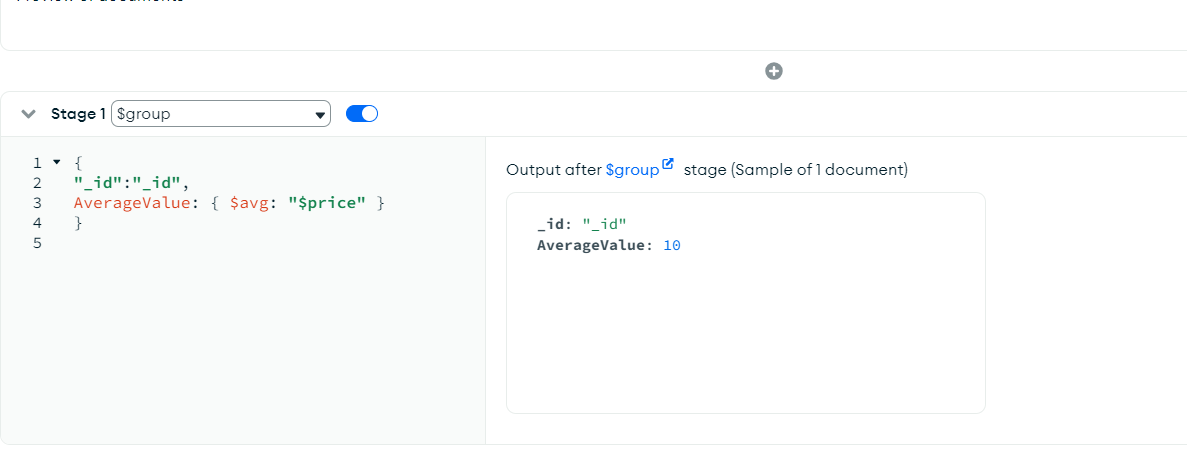
## 4. Use MongoDB Compass to perform an aggregation operation that calculates the average "price" of products in the "products" collection. (3 marks)

Step 1: Click on the aggregation window in the top of the compass pane below the selected collection. And click on the add stage button below pipeline window



Step 2: Select the group option in the stage window and enter the below query.

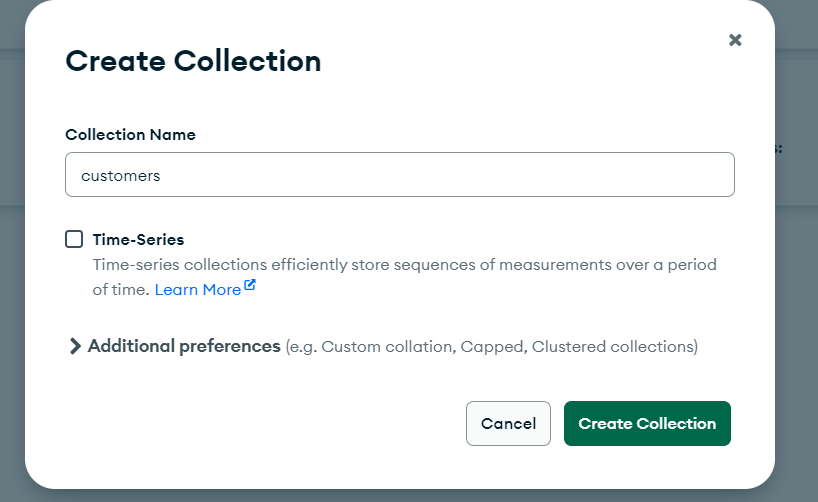
{“\_id”: “\_id”, AverageValue: {$avg: “$price”}}



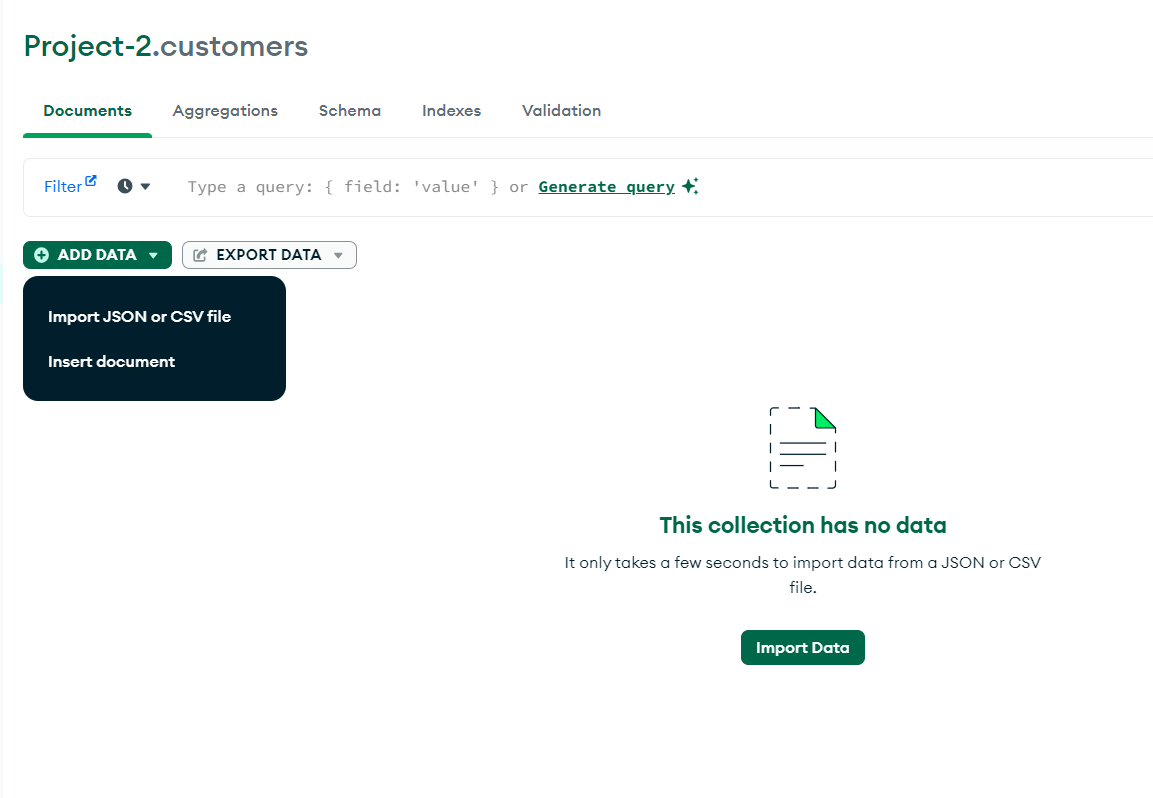
Output: This Mongo Db compass query creates an aggregate or average price of all products present in the products collection.

## 5. Import a JSON file containing customer data into a new collection named "customers" in your MongoDB database using MongoDB Compass. (3 marks)

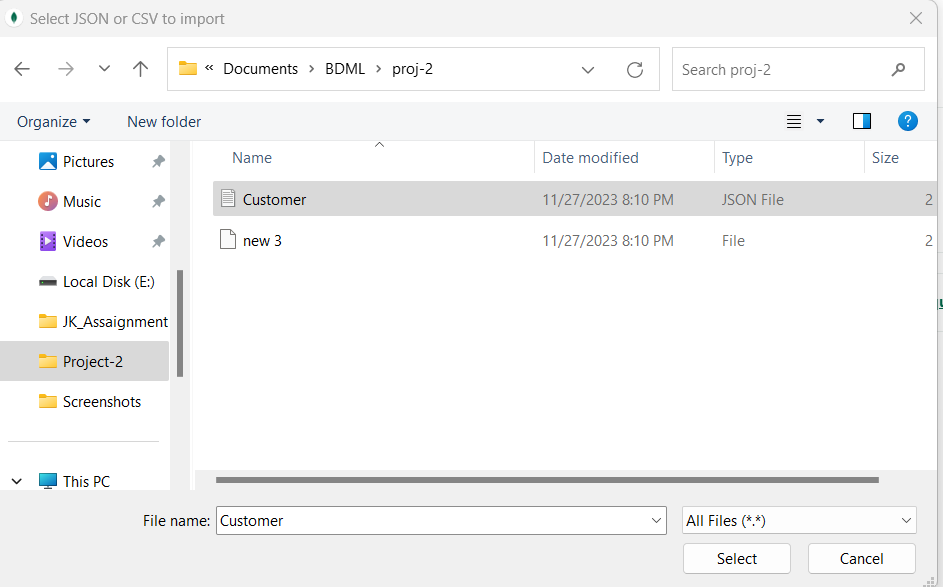
Step 1: Click on Create collection and enter the name customers



Step 2: click on add data under the collection on documents pane.



Step 3: Select the appropriate JSON file to be imported and click import button.



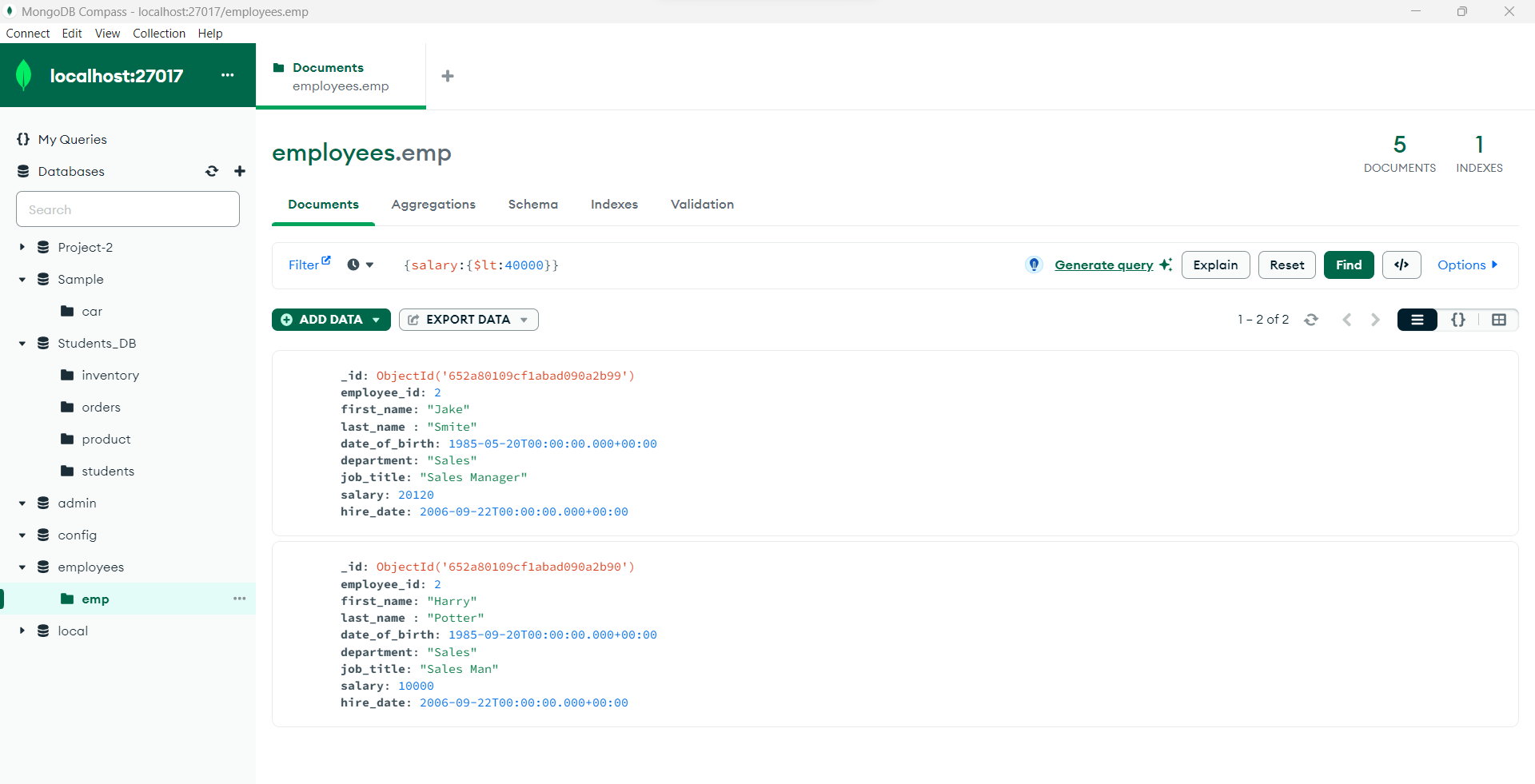
Step 4: We can see the imported JSON file in the document pane.



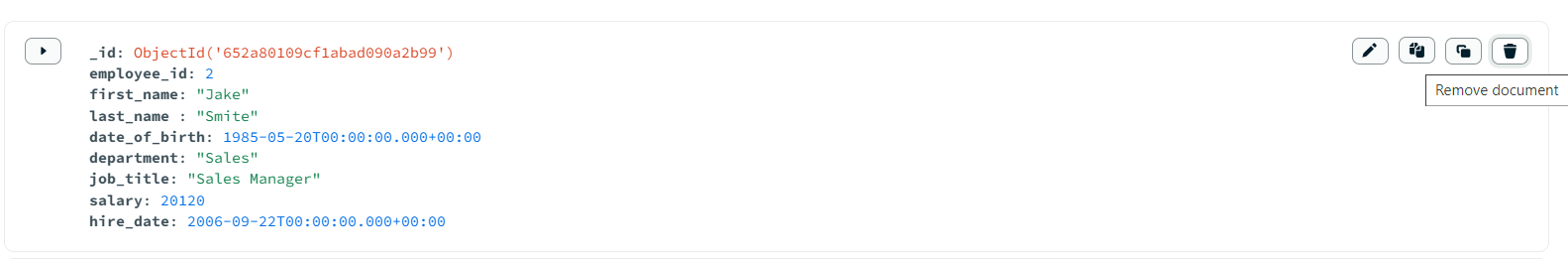
Output: The above steps have created a collection and inserted 10 record in the customers collection.

## 6. Delete a document in the "employees" collection where the "salary" field is less than $40,000 using MongoDB Compass. (2 marks)

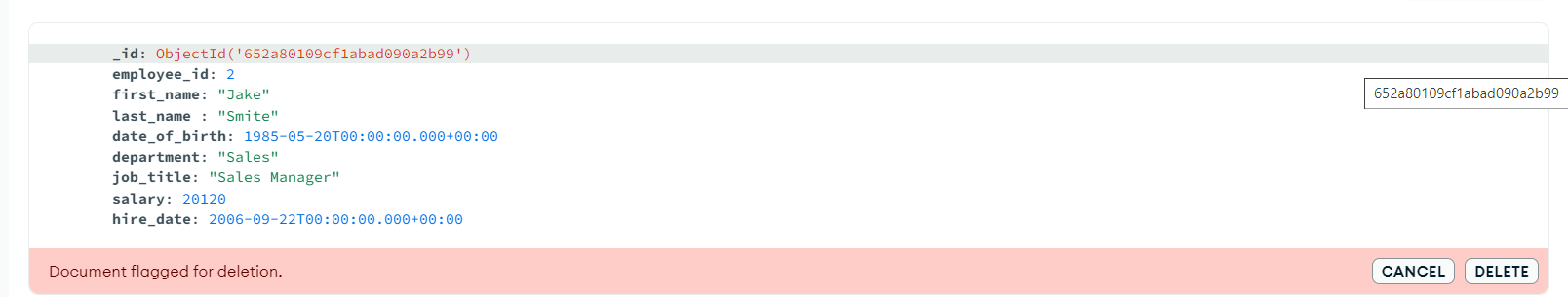
Step 1: Open The employees collection and enter the filter {salary:{$lt:40000}}.



Step 2 : The documents that pass the filter condition will be displayed in the document pane. Click on delete/remove document.



Step 3: a Pop up shows flagged for delete click on delete in the below pop up.



Output: The document with salary below 40000 is deleted.