COMP1702 - BIG DATA

Richard Raja

001370307

MSc Data Science

Table of Contents

Connecting to Hive via Virtual Machine 4

Task A: Hive Data Warehouse Design 4

Task B: MapReduce Programming 15

Task C: Big Data Project Analysis 17

Task C.1 17

Task C.2 17

Task C.3 18

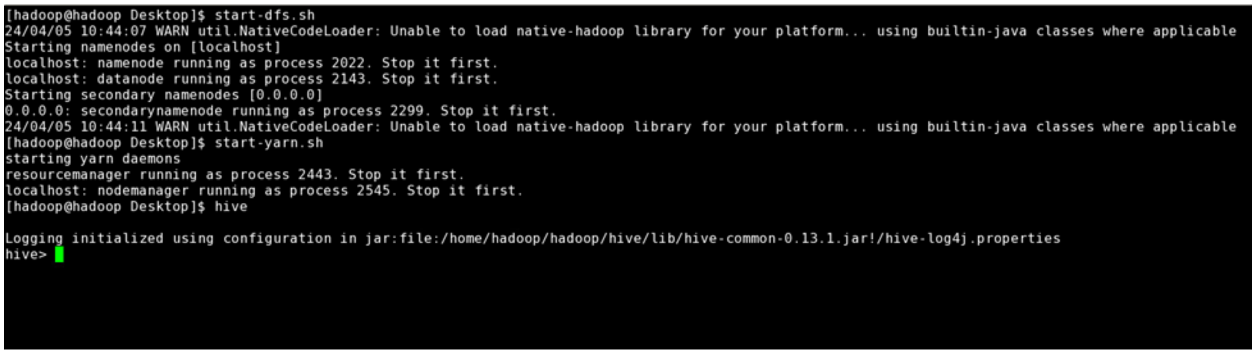
Reference 20

List of Figures

Figure 1: Virtual Machine (VM) named COMP1702-161 4
Figure 2: Creating “employee_db” database 5
Figure 3: Connecting the database 6
Figure 4: List of employees 6
Figure 5: List of departments available on the database 7
Figure 6: List of salaries from the database 8
Figure 7: Retrieval of all employees and their departments with salaries 9
Figure 8: Finding the average salary in the company 10
Figure 9: Listing the 5 departments with the highest total salary expense 10
Figure 10: Counting the number of employees in each department 11
Figure 11: Find the employees with the highest salary in each department 12
Figure 12: Listing all employees in the 'Research and Development' department 12
Figure 13: 5 youngest employees in the company 13
Figure 14: Calculating the total salary expense for each department 13
Figure 15: Counting the number of employees with salaries above 90000 14
Figure 16: Finding the department with the highest average employee age 14

Connecting to Hive via Virtual Machine

This screenshot shows the virtual machine (VM) named COMP1702-161 provided by the University for exploring Big Data



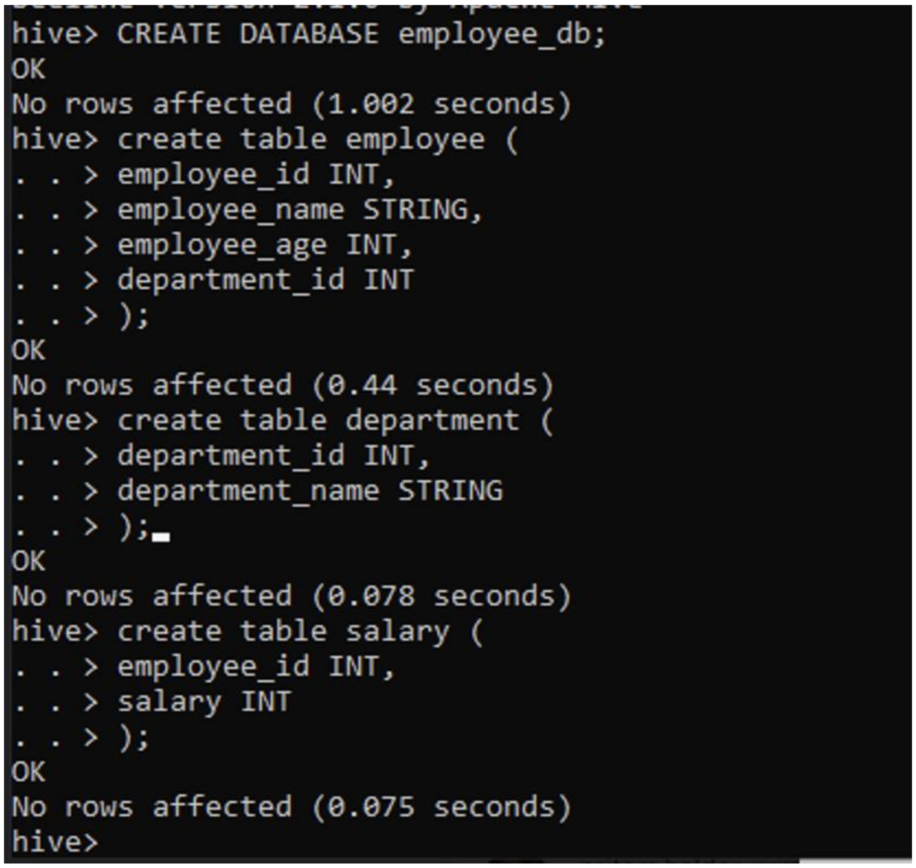
```
[hadoop@hadoop Desktop]$ start-dfs.sh
24/04/05 10:44:07 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: namenode running as process 2022. Stop it first.
localhost: datanode running as process 2143. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 2299. Stop it first.
24/04/05 10:44:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[hadoop@hadoop Desktop]$ start-yarn.sh
starting yarn daemons
resourcemanager running as process 2443. Stop it first.
localhost: nodemanager running as process 2545. Stop it first.
[hadoop@hadoop Desktop]$ hive

Logging initialized using configuration in jar:file:/home/hadoop/hadoop/hive/lib/hive-common-0.13.1.jar!/hive-log4j.properties
hive>
```

Figure 1: Virtual Machine (VM) named COMP1702-161

Task A: Hive Data Warehouse Design

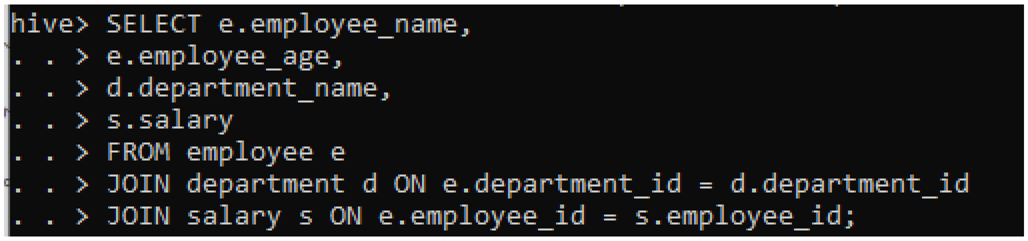
A data warehouse delivers a main method to store a large amount of data that can be efficiently explored to make instructed, data-driven decisions (Khajonmote *et al.* 2022). Apache Hive is a warehouse system that provides distributed, fault-tolerant data that enables analytics at a massive scale and permits users to read, write, and handle Petabytes of information utilizing SQL (Zhang *et al.* 2021). Hadoop is the framework that provides Hive and methods to properly implement it and allows users to store, sort, and analyze data on a massive scale.



```
hive> CREATE DATABASE employee_db;
OK
No rows affected (1.002 seconds)
hive> create table employee (
. . > employee_id INT,
. . > employee_name STRING,
. . > employee_age INT,
. . > department_id INT
. . > );
OK
No rows affected (0.44 seconds)
hive> create table department (
. . > department_id INT,
. . > department_name STRING
. . > );
OK
No rows affected (0.078 seconds)
hive> create table salary (
. . > employee_id INT,
. . > salary INT
. . > );
OK
No rows affected (0.075 seconds)
hive>
```

Figure 2: Creating “employee_db” database

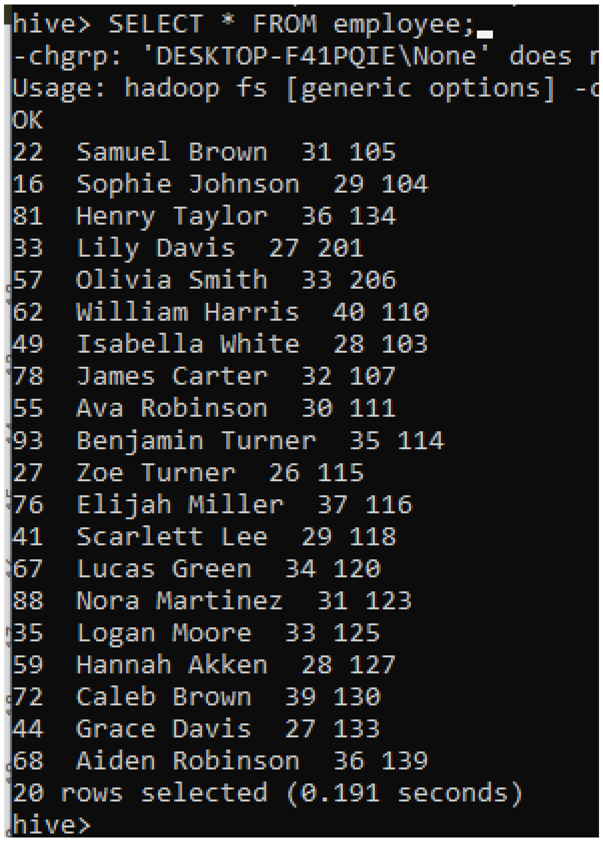
This is the creation of the “employee_db” database which is used to store various employee-related tables and data. It can be seen from the picture that Apache Hive has been utilized to create this database. The “employee_db” database contains three tables the “employee”, the “department”, and the “salary”. The “employee” table is used to store employee information and has the attributes “employee_id” which refers to the Identifier number allocated to each employee, “employee_name” which refers to the name of an individual employee, “employee_age” which is to stores the age of individual employees, and “department_id” which refers to where each and every employee belongs to.



```
hive> SELECT e.employee_name,
. . > e.employee_age,
. . > d.department_name,
. . > s.salary
. . > FROM employee e
. . > JOIN department d ON e.department_id = d.department_id
. . > JOIN salary s ON e.employee_id = s.employee_id;
```

Figure 3: Connecting the database

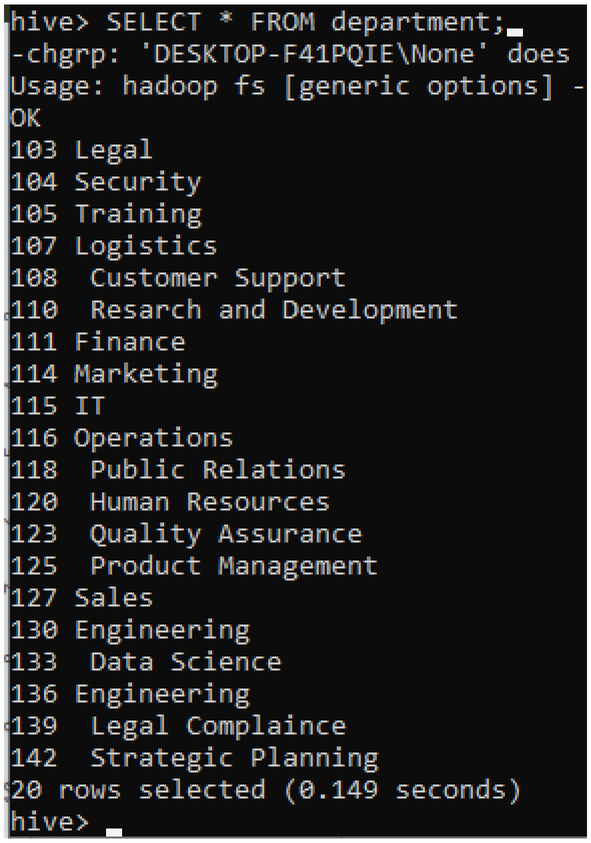
The above images show how the various attributes available on the database are connected to each other.



```
hive> SELECT * FROM employee;
-chgrp: 'DESKTOP-F41PQIE\None' does r
Usage: hadoop fs [generic options] -c
OK
22   Samuel Brown   31 105
16   Sophie Johnson   29 104
81   Henry Taylor   36 134
33   Lily Davis   27 201
57   Olivia Smith   33 206
62   William Harris   40 110
49   Isabella White   28 103
78   James Carter   32 107
55   Ava Robinson   30 111
93   Benjamin Turner   35 114
27   Zoe Turner   26 115
76   Elijah Miller   37 116
41   Scarlett Lee   29 118
67   Lucas Green   34 120
88   Nora Martinez   31 123
35   Logan Moore   33 125
59   Hannah Akken   28 127
72   Caleb Brown   39 130
44   Grace Davis   27 133
68   Aiden Robinson   36 139
20 rows selected (0.191 seconds)
hive>
```

Figure 4: List of employees

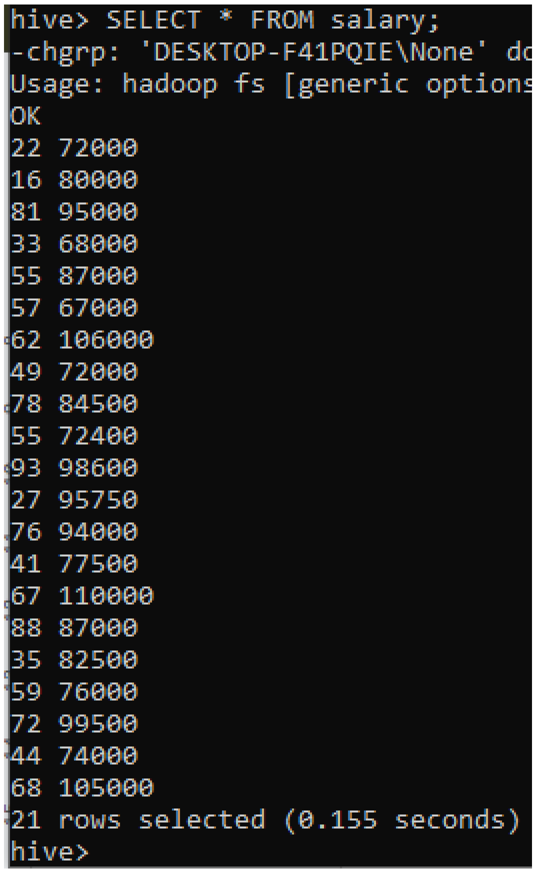
This is the list of total employees available from the “employee_db” database it can be seen that the total number of employees is 20. From the above picture, it is visible that the output contains the name of the employee and the relevant data related to that employee. The “employee_id” 22 belongs to the person with “employee_name” Samuel Brown who has the “employee_age” of 31 and belongs to the “department_id” 105.



```
hive> SELECT * FROM department;
-chgrp: 'DESKTOP-F41PQIE\None' does
Usage: hadoop fs [generic options] -
OK
103 Legal
104 Security
105 Training
107 Logistics
108  Customer Support
110  Resarch and Development
111 Finance
114 Marketing
115 IT
116 Operations
118  Public Relations
120  Human Resources
123  Quality Assurance
125  Product Management
127 Sales
130 Engineering
133  Data Science
136 Engineering
139  Legal Complaince
142  Strategic Planning
20 rows selected (0.149 seconds)
hive>
```

Figure 5: List of departments available on the database

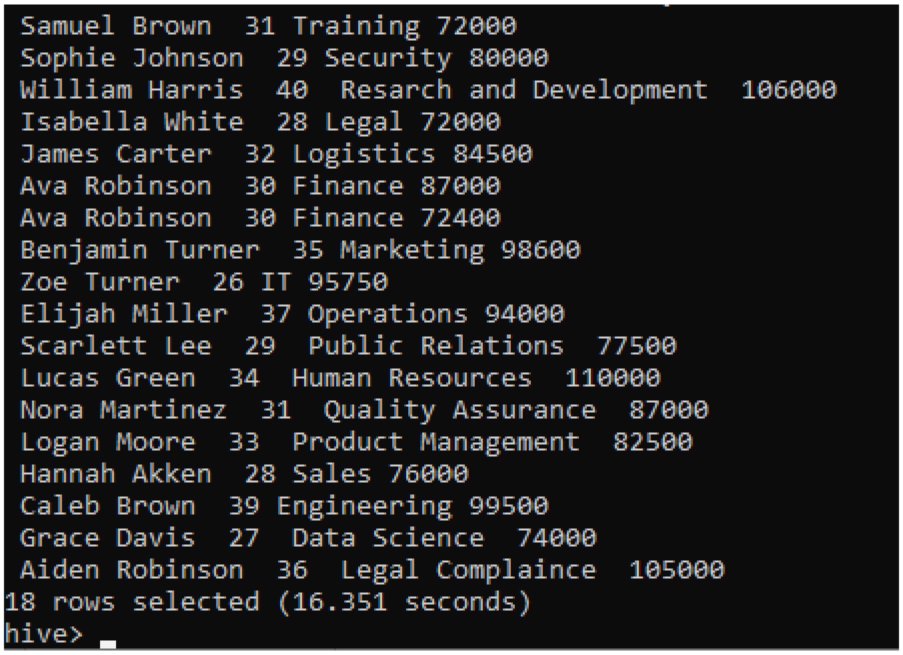
The above figure shows the list of departments available on the “department” table from the “employee_db” database, it also shows the total number of employees, and from the figure, it can be seen that the total number of employees is 20. It can be seen that the “department_id” 103 belongs to the department with “department_name” Legal.



```
hive> SELECT * FROM salary;
-chgrp: 'DESKTOP-F41PQIE\None' do
Usage: hadoop fs [generic options
OK
22 72000
16 80000
81 95000
33 68000
55 87000
57 67000
62 106000
49 72000
78 84500
55 72400
93 98600
27 95750
76 94000
41 77500
67 110000
88 87000
35 82500
59 76000
72 99500
44 74000
68 105000
21 rows selected (0.155 seconds)
hive>
```

Figure 6: List of salaries from the database

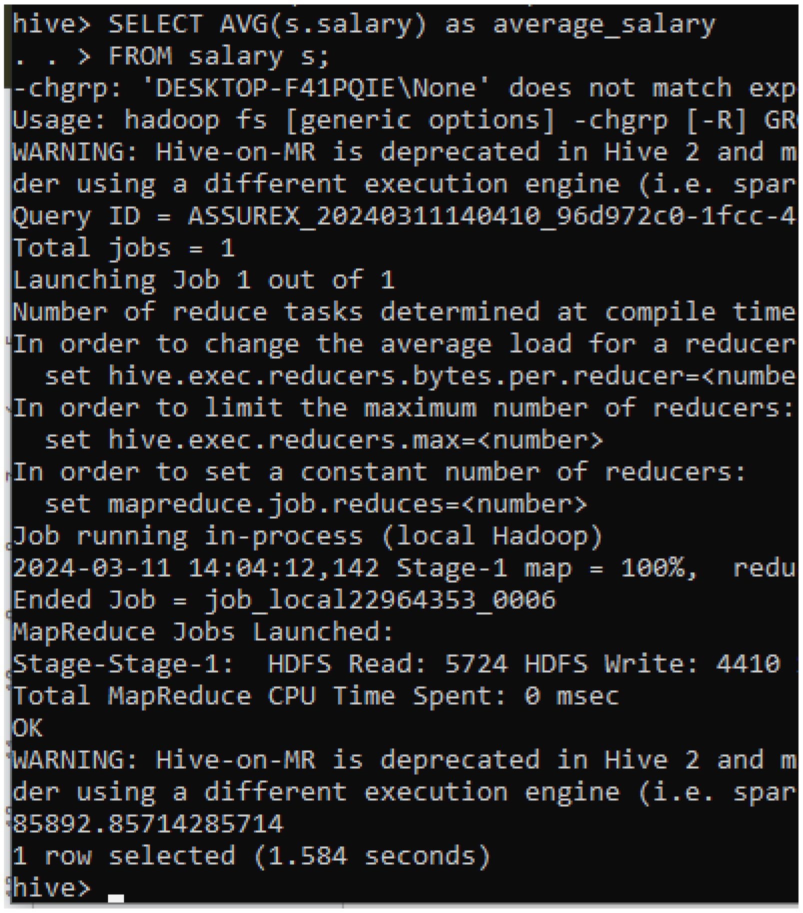
This image represents the list of salaries each employees get after a regular interval which is stored in the “salary” table from the “employee_db” and by analyzing the picture it can be concluded that the total number of items stored on the “salary” table is 21.



```
 Samuel Brown  31 Training 72000
 Sophie Johnson  29 Security 80000
 William Harris  40  Resarch and Development  106000
 Isabella White  28 Legal 72000
 James Carter  32 Logistics 84500
 Ava Robinson  30 Finance 87000
 Ava Robinson  30 Finance 72400
 Benjamin Turner  35 Marketing 98600
 Zoe Turner  26 IT 95750
 Elijah Miller  37 Operations 94000
 Scarlett Lee  29  Public Relations  77500
 Lucas Green  34  Human Resources  110000
 Nora Martinez  31  Quality Assurance  87000
 Logan Moore  33  Product Management  82500
 Hannah Akken  28 Sales 76000
 Caleb Brown  39 Engineering 99500
 Grace Davis  27  Data Science  74000
 Aiden Robinson  36  Legal Complaince  105000
18 rows selected (16.351 seconds)
hive>
```

Figure 7: Retrieval of all employees and their departments with salaries

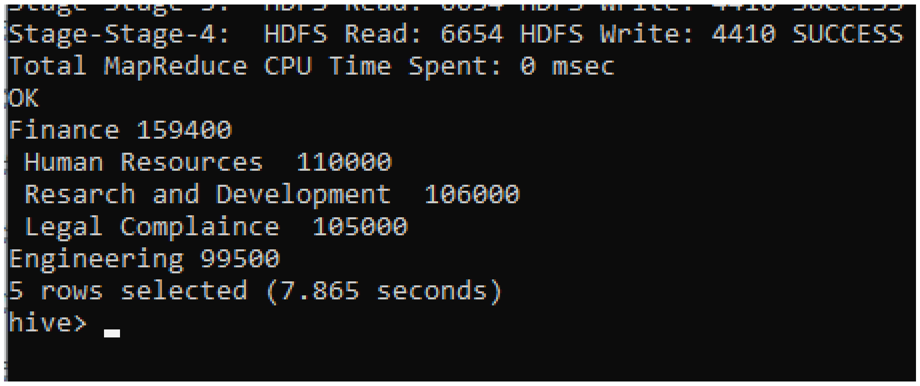
This query represents the retrieval of all employees with their respective salaries, it also shows other pieces of information such as their age and which department they belong to. The analysis of the above picture shows that “Samuel Brown” who is 31 of age belongs to the “Training” department and his salary is 72000.



```
hive> SELECT AVG(s.salary) as average_salary
. . > FROM salary s;
-chgrp: 'DESKTOP-F41PQIE\None' does not match exp
Usage: hadoop fs [generic options] -chgrp [-R] GR
WARNING: Hive-on-MR is deprecated in Hive 2 and m
der using a different execution engine (i.e. spar
Query ID = ASSUREX_20240311140410_96d972c0-1fcc-4
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time
In order to change the average load for a reducer
  set hive.exec.reducers.bytes.per.reducer=<numbe
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-03-11 14:04:12,142 Stage-1 map = 100%,  redu
Ended Job = job_local22964353_0006
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 5724 HDFS Write: 4410
Total MapReduce CPU Time Spent: 0 msec
OK
WARNING: Hive-on-MR is deprecated in Hive 2 and m
der using a different execution engine (i.e. spar
85892.85714285714
1 row selected (1.584 seconds)
hive>
```

Figure 8: Finding the average salary in the company

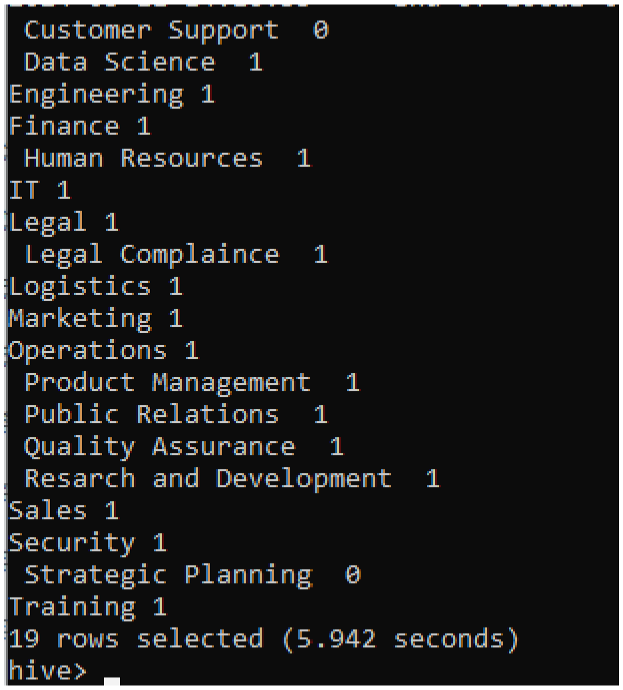
It is the representation of the average salary of a person in the company available to the database, analyzing it reveals that the average salary of a person working in this company is 85892.85.



```
Stage-Stage-4:  HDFS Read: 6654 HDFS Write: 4410 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Finance 159400
 Human Resources  110000
 Resarch and Development  106000
 Legal Complaince  105000
Engineering 99500
5 rows selected (7.865 seconds)
hive>
```

Figure 9: Listing the 5 departments with the highest total salary expense

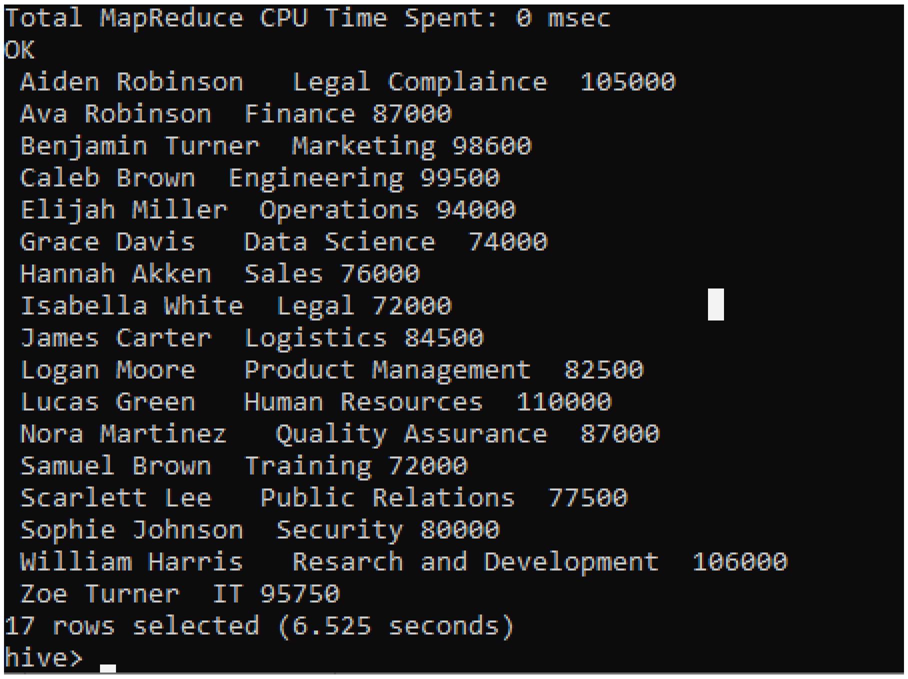
The above images represent the query search for the 5 highest paid departments from the database. It can be seen that the finance department is the highest paid department available from the database with a total salary count of 159400. It is followed by Human Resources with an 110000 salary, the Research and Development department with a salary count of 106000, the Legal Compliance department with 105000, and the Engineering department with a salary count of 99500.



```
 Customer Support  0
 Data Science  1
Engineering 1
Finance 1
 Human Resources  1
IT 1
Legal 1
 Legal Complaince  1
Logistics 1
Marketing 1
Operations 1
 Product Management  1
 Public Relations  1
 Quality Assurance  1
 Resarch and Development  1
Sales 1
Security 1
 Strategic Planning  0
Training 1
19 rows selected (5.942 seconds)
hive>
```

Figure 10: Counting the number of employees in each department

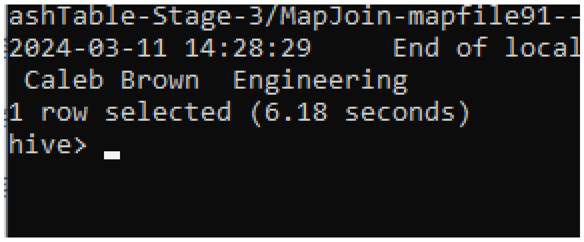
The above picture represents the total number of employees available in each department from the database. It can be seen from the query output that each departments have one employee each for every department respectively.



```
Total MapReduce CPU Time Spent: 0 msec
OK
 Aiden Robinson   Legal Complaince  105000
 Ava Robinson  Finance 87000
 Benjamin Turner  Marketing 98600
 Caleb Brown  Engineering 99500
 Elijah Miller  Operations 94000
 Grace Davis   Data Science  74000
 Hannah Akken  Sales 76000
 Isabella White  Legal 72000
 James Carter  Logistics 84500
 Logan Moore   Product Management  82500
 Lucas Green   Human Resources  110000
 Nora Martinez   Quality Assurance  87000
 Samuel Brown  Training 72000
 Scarlett Lee   Public Relations  77500
 Sophie Johnson  Security 80000
 William Harris   Resarch and Development  106000
 Zoe Turner  IT 95750
17 rows selected (6.525 seconds)
hive> 
```

Figure 11: Find the employees with the highest salary in each department

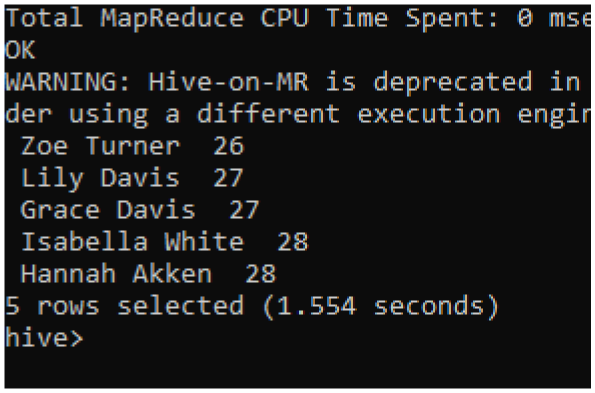
This is the representation of the highest-paid employee from each department. It can be seen from the picture that the highest-paid employee is from the Human Resouces department with a salary count of 110000 followed by William Harris from the Research and Development Department with a salary of 106000 and Aiden Robinson from the Legal Compliance department with a salary count of 105000. There are also employees from other departments such as Ava Robinson from the finance department with a salary of 87000, Celeb Brown from the Engineering department with a salary of 99500, and Operations department Elijah Miller with a salary count of 94000.



```
ashTable-Stage-3/MapJoin-mapfile91--
2024-03-11 14:28:29     End of local
 Caleb Brown  Engineering
1 row selected (6.18 seconds)
hive> 
```

Figure 12: Listing all employees in the 'Research and Development' department

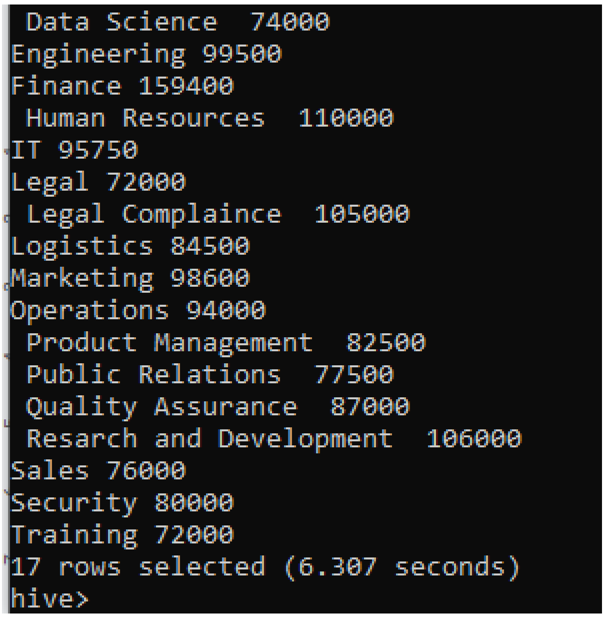
This query search represents the list of employees working in the Engineering department, and from the query output, it can be concluded that there's one employee working for the Engineering department with the name Caleb Brown.



```
Total MapReduce CPU Time Spent: 0 mse
OK
WARNING: Hive-on-MR is deprecated in
der using a different execution engin
 Zoe Turner  26
 Lily Davis  27
 Grace Davis  27
 Isabella White  28
 Hannah Akken  28
5 rows selected (1.554 seconds)
hive>
```

Figure 13: 5 youngest employees in the company

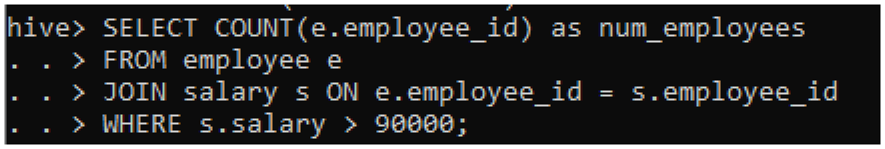
This query Search represents the five youngest employees working in the company from the database. Analyzing the query output reveals that it shows the names of the five youngest people and their ages. The youngest person working in the company is Zoe Turner the age 26 followed by 27 year old Lily Davis and Grace Davis. The oldest of the youngest is Isabella White and Hannah Akken at the age of 28.



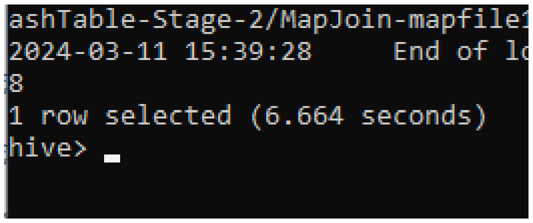
```
 Data Science  74000
Engineering 99500
Finance 159400
 Human Resources  110000
IT 95750
Legal 72000
 Legal Complaince  105000
Logistics 84500
Marketing 98600
Operations 94000
 Product Management  82500
 Public Relations  77500
 Quality Assurance  87000
 Resarch and Development  106000
Sales 76000
Security 80000
Training 72000
17 rows selected (6.307 seconds)
hive>
```

Figure 14: Calculating the total salary expense for each department

The above picture shows the salary expanse of each department that needs to be paid by the company, and it can be seen that the Finance department needs to be paid the most.



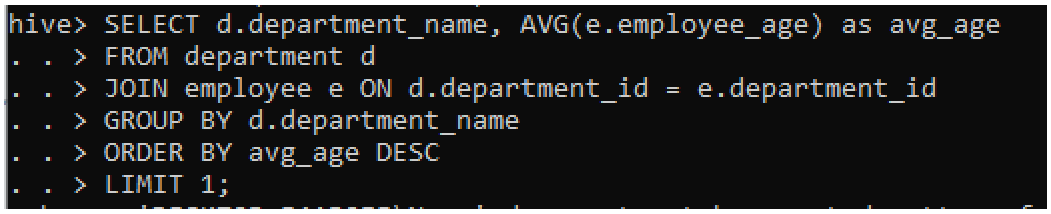
```
hive> SELECT COUNT(e.employee_id) as num_employees
. . > FROM employee e
. . > JOIN salary s ON e.employee_id = s.employee_id
. . > WHERE s.salary > 90000;
```



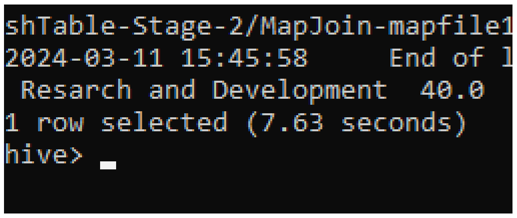
```
ashTable-Stage-2/MapJoin-mapfile1
2024-03-11 15:39:28     End of lo
8
1 row selected (6.664 seconds)
hive>
```

Figure 15: Counting the number of employees with salaries above 90000

This image represents the total number of employees employed by the company who have a salary count higher than the number 900000, and it can be seen that 8 employees have a salary count higher than the number 900000.



```
hive> SELECT d.department_name, AVG(e.employee_age) as avg_age
. . > FROM department d
. . > JOIN employee e ON d.department_id = e.department_id
. . > GROUP BY d.department_name
. . > ORDER BY avg_age DESC
. . > LIMIT 1;
```



```
shTable-Stage-2/MapJoin-mapfile1
2024-03-11 15:45:58     End of l
 Resarch and Development  40.0
1 row selected (7.63 seconds)
hive>
```

Figure 16: Finding the department with the highest average employee age

This is the query search for the department with the highest average employee age. It can be seen that the Research and Development department has the highest average employee age.

Task B: MapReduce Programming

Task b.4: Output the average number of authors per paper for each year.

MapReduce is used to process large amounts of data, it splits larger chunks of data into smaller chunks which helps the system to process and analyses huge amounts of data (Wu. 2022). Below is a MapReduce programming based on the average number of authors per paper for each year.

Mapper

```
Map(String inputKey, String inputValue):
    # Splitting the input line into fields
    fields = split(inputValue, '|')

    # Extracting the relevant information
    authors = split(fields[0], ',')
    year = fields[3]

    # Emitting key-value pair: (year, (number_of_authors, 1))
    for author in authors:
        emitIntermediate(year, (1, 1))

# Function to emit intermediate key-value pairs
function emitIntermediate(key, value):
    // Emitting the intermediate key-value pair
    output(key, value)
```

The above represents the Mapper code used for this pseudo-code, it can be seen that “split(inputValue, '|')” represents the code where input values will be split using the “|” and will be stored in the array named “fields”. The “ split(fields[0],‘,’”) stores the value from the first field into the authors. The emitIntermediate function is called to output the intermediate key-value pair. Where the key is the “year” and the value is “(1,1)” where the first element represents the number of authors and the second element is always 1.

Reducer

```
Reduce(String key, List<Values> values):
    total_authors = 0
    total_papers = 0

    # Aggregating the values for the same year
    for value in values:
        total_authors += value[0]  # Number of authors
        total_papers += value[1]   # Count of papers

    # Calculate the average number of authors per paper
    average_authors_per_paper = total_authors / total_papers

    # Emitting the final key-value pair: (year, average_authors_per_paper)
    output(key, average_authors_per_paper)
```

The Input Parameters in the above reduced code are key represented by the “year” and values represented by a list of values for the same year, where each value is a tuple “(number_of_authors, 1)”. The “ total_authors” and “total_papers” are 0 meaning that the initial value is set to 0. A loop iterates over the values, initializing total_authors and total_papers based on the values in the tuple. The “ average_authors_per_paper = total_authors / total_papers” calculates the average number of authors per paper for the year. The output is based on the key-value pair. This map reducer program calculates the average number of authors per paper each year.

Task C: Big Data Project Analysis

Task C.1

The Business Head of ABC Investment Bank Ltd wants to create a data warehouse to apply social media, news articles, and other information methods. Its main goal is to assemble a trading method and collection rebalancing decisions to give the back a competitive edge against its rivals (Verma *et al.* 2022). A data warehouse is a category of data management system created to enable and assist business intelligence schemes.

Although the Business Head of ABC Investment Bank Ltd has the right idea in mind, for this project instead of using a data warehouse a data lake has been used for various reasons. Reasons such as a data lake being able to store structured, semi-structured, and nanostructured data while a data warehouse is only able to store structured data (Saddad *et al.* 2020). A data lake is able to store a wide variety of data whereas a data warehouse is modeled to store only a specific type of data. A data lake is much faster when storing data due to it not having the requirement to sort data, while a data warehouse needs to sort data according to its requirements which is much time consuming. Due to all these reasons utilizing a data lake is able to fulfill the bank requirements of creating a business intelligence system that is able to give them an edge over their rivals.

Proper implementation of a data lake system needs proper utilization of various methods as the first step is to create a proper storage system for the raw data that needs to be stored. Data Lakes are mainly stored in distributed object stores, each storing key points to a file. Hadoop Distributed File system more commonly known as the open-source one, mother to all other systems. The next part is data ingestion referring to the process of importing large, assorted data files from multiple sources into a single, cloud-based storage medium such as the Hadoop framework. Handling the large database of this economic solution needs the implementation of a proper query management system such as Hadoop Hive. Finally, the implementation of all the necessary policies has been processed to make sure that the model complies with the existing laws.

Task C.2

It has been suggested by the business head of ABC Investment Bank Ltd to build a map-reduce function for the parallel distributed processing on a cluster. The aim of doing this is to analyze real time data processing which will give the company and its client enough time to make financial decisions that will keep the company ahead of its competitors (Galvão *et al.* 2020). MapReduce is a distributed execution framework used in Apache Hadoop Framework and based on Java. MapReduce is used for processing a large set of databases it splits large chunks of data into smaller chunks and processes them by utilizing the Apache Hadoop Servers.

However, the problem of using map reduction in this business scenario is not an efficient solution since map reduction lacks the capability of live data processing. It suffers from high latency which is a major problem for real-time data or near real-time data processing (Maddikunta *et al*. 2022). Even if one can ignore the above problems the complexity that needs to be implemented to be able to analyze real-time data using MapReduce is simply too high and inefficient for this business intelligent model.

A suitable alternative for MapReduce is Apache Spark which can be implemented appropriately for this business intelligent model created for ABC Investment Bank Ltd (Ngo *et al.* 2020). Apache Spark has the ability to perform in-memory processing which reduces the processing speed significantly and enables the model to perform real-time data analysis. Spark also allows batch processing, supports interactive queries, and enables stream processing allowing the model to process a constant stream of data. It also supports an extended set of libraries and tools that can be used for data processing and visualization.

Task C.3

This is a detailed hosting strategy implemented for ABC Investment Bank Ltd. for its requirement of big data analysis utilizing Hadoop. It will give the company and its clients a competitive advantage when deciding whether they should buy, sell, or hold a Financial Instrument (Goyal *et al* 2020). This Business intelligence system will be able to analyze historical data from social media like Twitter, Facebook, or from any news article, and will provide the bank about the financial market beforehand. The steps that need to be followed for a successful implementation of the systems, and the first step is the proper implementation of a server. The bank has branches at key financial points of the world such as London, New York, Tokyo, Hong Kong, and Sydney (Rudniy. 2022). The data server has been implemented in the key locations and branches of the bank for a faster web of servers. Subscription-based services such as Amazon web services, Google Cloud, or Azure can be utilized that exchange and analyze data which will provide the bank with a global presence all over the world and will allow the system to collect data faster than others (Priyanka *et al.* 2021). Apache Spark cluster can be utilized to successfully store a large amount of data and will allow efficient analysis of the stored data. Apache Spark needs to be properly configured for efficient data analysis on a global scale. For the next step, the system needs to be properly configured to store data using Amazon web service or Azure. The system must be implemented with a proper encryption algorithm and security measures to ensure data safety. A proper backup server should be implemented with configuration for regular data backups (Jain *et al.* 2021). Then the international and local security and safety policies are implemented to ensure compliance with the authorities. All these steps will ensure the security and safety of the system and will be able to execute the requirements of ABC Investment Bank Ltd.

Reference

Galvão, J., Leon, A., Costa, C., Santos, M.Y. and López, Ó.P., 2020, November. Towards Designing Conceptual Data Models for Big Data Warehouses: The Genomics Case. In European, Mediterranean, and Middle Eastern Conference on Information Systems (pp. 3-19). Cham: Springer International Publishing.

Goyal, D., Goyal, R., Rekha, G., Malik, S. and Tyagi, A.K., 2020, February. Emerging trends and challenges in data science and big data analytics. In 2020 International conference on emerging trends in information technology and engineering (ic-ETITE) (pp. 1-8). IEEE.

Khajonmote, W., Chinsook, K., Klintawon, S., Sakulthai, C., Leamsakul, W., Jansawang, N. and Jantakoon, T., 2022. System Architecture of Big Data in Massive Open Online Courses (BD-MOOCs System Architecture). Journal of Education and Learning, 11(3), pp.105-118.

Maddikunta, P.K.R., Fang, F. and Pathirana, P.N., 2022. A survey on blockchain for big data: Approaches, opportunities, and future directions. Future Generation Computer Systems, 131, pp.209-226.

Ngo, V.M., Le-Khac, N.A. and Kechadi, M.T., 2020. Data warehouse and decision support on integrated crop big data. International Journal of Business Process Integration and Management, 10(1), pp.17-28.

Priyanka, E.B., Thangavel, S., Meenakshipriya, B., Prabu, D.V. and Sivakumar, N.S., 2021. Big data technologies with computational model computing using hadoop with scheduling Jain, N. and Tiwari, P., 2021. MULTI-JOIN-ORDERING QUERY OPTIMIZATION ALGORITHM FOR HIVE WAREHOUSE WITH MAPREDUCE. OORJA-International Journal of Management & IT, 19(1).

Rudniy, A., 2022. Data Warehouse Design for Big Data in Academia. Computers, Materials & Continua, 71(1).

Saddad, Emad, Ali El-Bastawissy, Hoda MO Mokhtar, and Maryam Hazman. "Lake data warehouse architecture for big data solutions." International Journal of Advanced Computer Science and Applications 11, no. 8 (2020).

Verma, A., Lamsal, K. and Verma, P., 2022. An investigation of skill requirements in artificial intelligence and machine learning job advertisements. Industry and Higher Education, 36(1), pp.63-73.

Wu, W., 2022, July. Investigating internship experiences of data science students for curriculum enhancement. In Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1 (pp. 505-511).

Zhang, H., Zhang, J., Tian, M. and Qiao, B., 2021, December. Design of Offline Analysis System for Remote Sensing Data Service Based on Hive. In 2021 International Conference on Digital Society and Intelligent Systems (DSInS) (pp. 300-303). IEEE.