Big Data Report

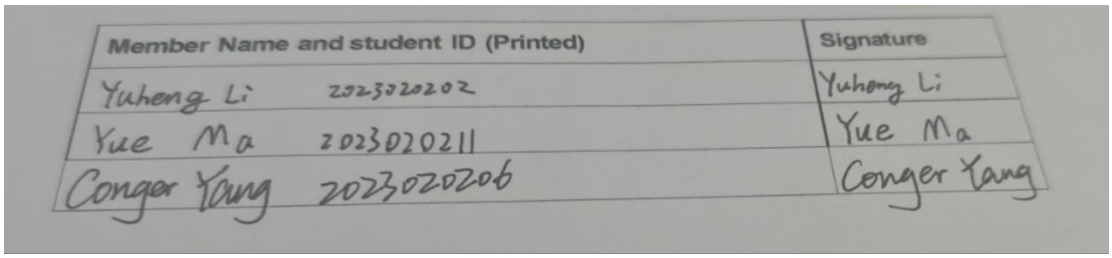
Reporter: Yuheng Li

Student ID: 2023020202

Group members:

Yue Ma (2023020211),

Conge Yang (2023020206)



Member Name and student ID (Printed)		Signature
Yuheng Li	2023020202	Yuheng Li
Yue Ma	2023020211	Yue Ma
Conger Yang	2023020206	Conger Yang

Our Report Github: https://github.com/Martin-Mystery/111

Module: Big Data

Lab: BD_Lab_01

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

This experiment aims to deepen the understanding of the basic concepts, characteristics, and applications of big data in business through a series of tasks. The experimental content includes the definition of big data, the requirements for data intensive systems, the distinction between structured and unstructured data, and the four key characteristics of big data (Volume, Velocity, Variety, Veracity).

Experimental content:

Basic tasks:

a) Definition of Big Data: Explain the meaning of big data through TikTok and Amazon case studies.

b) Business revenue: Big data analysis helps businesses optimize management and increase revenue.

c) Data type differentiation: Structured data is easy to query, while unstructured data is complex to process.

d) Data contribution diagram: shows the process of data from source to application.

e) Data intensive systems: Analyze user behavior and provide personalized services.

f) Technical Applications: Introduced HDFS, Tableau, Apache Spark, and Amazon Redshift.

Medium task:

a) Data value: Data is similar to oil and requires processing and analysis.

b) Data Authenticity: Discussed data inconsistency and quality issues.

Advanced tasks:

a) Dataset selection: The "Online Retail" dataset has been selected.

b) Dataset application: used to build data intensive systems for transactions between merchants and buyers.

Personal reflective:

In this experiment, we realized that big data is not just about having a large amount of data, but more importantly, how to extract valuable information from the data. Meanwhile, the authenticity and quality of data are crucial for the accuracy of data analysis.

Module: Big Data

Lab: BD_Lab_02

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

The aim of this experiment is to study different types of database systems, understand their applications in the real world, and recommend suitable database solutions for specific scenarios.

Experimental content:

Basic tasks:

a) Selected database instance:

Online and social games (entertainment)

Weather forecast

Library Management System (Document Management)

b) Recommended database solution:

Online and social games: using relational databases to track scores, inventory, and game status; Use NoSQL databases (key value stores) to store player settings and preferences.

Weather prediction: using relational databases to store weather data; Analyze large amounts of structured data using NoSQL databases (columnar storage).

Library management system: using relational databases to store books and student data; Use NoSQL databases (document storage) to store semi-structured data.

Medium task:

Comparison between relational databases and NoSQL databases:

Definition: A relational database is based on a relational model, where data is stored in tables that are interrelated through relationships (such as foreign keys). NoSQL databases are designed to handle large amounts of data, with flexible patterns and support for multiple data formats.

Advantages: Relational databases provide strong consistency, transaction support, and complex query capabilities. NoSQL databases have highly scalable, high-performance, and flexible data models.

Limitations: Relational databases have limited scalability and high maintenance costs. NoSQL databases may have issues with data consistency.

Software examples: MySQL, PostgreSQL, Oracle, SQL Server (relational); MongoDB、Cassandra、Redis(NoSQL).

Use case: Relational databases are suitable for financial transactions and customer relationship management. NoSQL databases are suitable for social media, real-time analytics, and big data storage.

Advanced tasks:

Comparison of database systems:

Features: Data storage capacity, standardization, simplicity, complexity and cost, database structure, integrity constraints, ACID transactions, performance (read/write), scalability, schema flexibility, reliability, storage requirements.

Comparison: Relational databases are suitable for medium-sized structured data, while NoSQL databases are suitable for large amounts of unstructured or semi-structured data.

Personal reflective:

In this experiment, we learned how to choose the appropriate database type based on the characteristics and application scenarios of the data. Choosing a suitable database requires comprehensive consideration of factors such as data model, scalability, data consistency, and performance.

Module: Big Data

Lab: BD_Lab_03

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

The purpose of this experiment is to investigate the data types and data exchange in real-time applications, understand the role of middleware in real-time systems, and select appropriate database solutions for specific scenarios.

Experimental content:

Basic tasks:

a) Autonomous Campus Security System:

Collaborative work of safety robots to protect campus safety.

Middleware execution activities: task allocation, information security, centralized data processing, efficient shift change, and collaboration with other systems.

Medium task:

a) Security/Emergency Scenarios:

Fire, illegal intrusion, natural disasters, health emergencies, etc.

Required data: video surveillance data, navigation data, energy usage data, real-time location data.

b) Database selection:

Structured data: relational databases (such as MySQL, PostgreSQL).

Unstructured data: NoSQL databases such as MongoDB and Cassandra.

Real time data: Real time databases (such as Redis, InfluxDB).

Advanced tasks:

a) Telecommunications and AI/machine learning applications:

Consider using Starlink communication and Microsoft Azure machine learning services.

b) Solution Display:

Create 2-3 slides to showcase the selected solution.

Personal reflective:

The experimental results indicate that middleware plays a crucial role in real-time systems, responsible for task allocation, information security, and data processing. When choosing a database solution, it is necessary to comprehensively consider factors such as dataset, scalability, data consistency, performance, security, and maintenance.

Module: Big Data

Lab: BD_Lab_04

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

The purpose of this experiment is to understand the basic concepts, structure, and functions of Apache Hadoop framework and HDFS, as well as the application of MapReduce programming model through practical operation.

Experimental content:

Basic tasks:

1. Copy files to multiple hosts:

Advantages: Reduce the load on the central server, improve data storage security, and minimize server failures caused by excessive load.

Potential drawbacks: There may be data consistency issues during data transmission, and problems such as disconnection and transmission failure may occur when transferring to other hosts; Increased maintenance costs, including the cost of purchasing and maintaining machines; Difficulty in maintaining hosts distributed in different regions leads to reduced efficiency.

2. Large file access on remote nodes:

Main disadvantages: There are risks to the integrity of data transmission, and data may be intercepted or maliciously tampered with during the transmission process.

Improvement method: Divide large files into small files and distribute them to multiple computing nodes for parallel processing.

Data locality: In distributed computing tasks, computing operations are physically close to data to reduce data transmission overhead and improve performance.

3. Sequential/Parallel Processing:

Sequential processing: Decompose matrices A and B into smaller sub matrices and assign them to different processors or computing nodes.

Parallel/Distributed Processing: Each processor or computing node independently calculates the product of its assigned submatrix.

4. Parallel execution of mathematical formulas:

Parallel executable parts: Each part of the formula can be independently calculated.

Limitations of parallel execution: The final subtraction operation needs to be coordinated, and data transmission between nodes may introduce delays, requiring synchronous parallel tasks to complete the computation.

5. Determine the maximum number in the large file:

Distributed/Parallel Method: Divide a large file into 50 small parts, with each computing node processing one part of the file, independently finding the local maximum, and then summarizing it to the main node to determine the global maximum.

6. Differences from the distributed/parallel approach in question 4:

Data complexity: The fifth question is about text data, which is relatively simple and can be directly addressed using distributed methods. The fourth question involves tree structure, which makes it difficult to use distributed partitioning from the beginning. It

is necessary to first find the parts that can be parallelized, and then use column based methods based on this part.

Medium task:

7. Hadoop Framework: HDFS

The relationship between NameNode and DataNode: NameNode manages the namespace of the file system, maintains the directory tree structure of the file system, and maps the relationships between files and data blocks. DataNode stores actual data and is managed and scheduled by NameNode.

File storage and replication: Files are divided according to block size, with multiple copies per block to improve data reliability and availability. The NameNode is responsible for allocating storage locations for data blocks and coordinating the replication process.

Advanced tasks:

8. Hadoop Framework: MapReduce

The relationship between JobTracker and TaskTracker: JobTracker is responsible for managing and coordinating the entire job process, while TaskTracker is responsible for executing actual tasks.

MapReduce function: The Map function processes input data and outputs intermediate key value pairs; The Reduce function processes the key value pairs output by the Map, merges values with the same key, and generates the final result.

9. Use MapReduce to calculate the total quantity of device orders:

Input and output keys: The input key is the order ID, and the output key is the device name.

10. Streaming media service billing:

Using MapReduce to calculate streaming media playback frequency and cost: The Map function parses each line of data and outputs key value pairs (program name, 1); The Reduce function summarizes the number of times each program is played and calculates the cost.

Personal reflective:

In this experiment, we learned how to use Hadoop for data storage and processing, as well as how to write MapReduce tasks to solve practical problems. For processing large-scale datasets, distributed computing and parallel processing are key.

Module: Big Data

Lab: BD_Lab_05_MongoDB

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

The purpose of this experiment is to understand the basic concepts, characteristics, and differences between MongoDB and relational SQL databases through practical operations, and to compare and understand the advantages of MongoDB as a "schema free" database.

Experimental content:

Basic tasks:

1. MongoDB handles document relationships:

Handle relationships between documents by embedding documents, referencing documents, and referencing databases.

2. The schema free feature of MongoDB:

MongoDB does not require identifying identifiers and inserting data types when creating documents, nor does it have strict definitions and formats like PostgreSQL. MongoDB allows for more flexible document editing and the ability to handle large amounts of data.

3. Dynamic Mode of MongoDB:

From the three examples, it can be seen that MongoDB's dynamic pattern is reflected in the flexibility and variability of keys. Documents in a collection can have different

fields and structures, and MongoDB automatically manages the structure of the collection based on the inserted documents.

Medium task:

4. Equivalent statement conversion between SQL and MongoDB:

Convert SQL Create/INSERT statements to MongoDB equivalent statements.

Convert SQL SELECT/FIND statements to MongoDB equivalent statements.

5. Characteristics of big data:

Big data, through its "V" characteristic (Volume, Velocity, Variety, Variability, Veracity, Value have changed the way we process and analyze information.

The diversity of big data refers to a wide range of data types, including structured, semi-structured, and unstructured data.

MongoDB handles data diversity through its flexible document model, dynamic schema, and powerful aggregation framework.

Advanced tasks:

6. MongoDB query operations:

Use the find operator to query documents in a collection of books.

Compare MongoDB operations with SQL equivalent statements and write MongoDB statements.

Personal reflective:

The experimental results indicate that MongoDB, as a NoSQL database, has significant advantages in handling big data due to its flexibility and scalability. The schema free nature of MongoDB enables it to easily handle data with constantly changing structures, while its dynamic schema allows documents in a collection to have different fields and structures. We realize that MongoDB's schema free and dynamic patterns are very useful for processing large-scale datasets.

Module: Big Data

Lab: BD_Lab_06_Visualisation

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

This experiment aims to learn and practice the basic concepts and techniques of data visualization by using two tools, Microsoft Power BI and Anaconda Spyder.

Experimental content:

Medium task:

1.Microsoft Power BI：

Completed the practical operation practice of Power BI.

2. Fundamentals of Data Visualization:

Explored the fundamental concepts of data visualization.

3.Anaconda Spyder：

Installed and started Spyder.

Downloaded and imported the Stroke dataset.

Use the Pandas library to load datasets and view data types.

Display the first five records of the dataset.

Created histograms for each numerical field in the dataset.

Added more 'bins' or intervals to more accurately identify peaks in the data.

Created histograms for multiple variables.

A bar chart was created using categorical data.

Created a bar chart using the functions in the matplotlib. pyplot library.

Tried different chart styles.

Created a scatter plot to compare the relationship between two or more variables.

Created a pie chart to display categorical data.

4. Spotify dataset:

Loaded the Spotify dataset using Spyder.

A line graph was drawn with multiple lines.

Added appropriate axis labels and set titles.

Advanced tasks:

19. Financial data analysis:

Completed Spyder's financial data analysis workshop.

20. Tool comparison:

Developed a comparison between Anaconda Spyder and Microsoft Power BI.

Personal reflective:

Microsoft Power BI and Anaconda Spyder are both powerful data visualization tools, but they differ in usage and functionality. Power BI provides rich visualization options and an easy-to-use interface, while Spyder offers more programming flexibility and control. We learned how to use two different tools for data visualization and understood their respective advantages and disadvantages. I realize that choosing the right tool depends on specific data analysis needs and personal preferences.

Module: Big Data

Lab: BD_Lab_07_GraphDB

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

The purpose of this experiment is to understand the basic concepts, characteristics, and differences between Neo4j graph database and relational databases and MongoDB through practical operations, and to create and query graph data using Cypher query language.

Experimental content:

Medium task:

1. ACID properties of Neo4j:

Neo4j ensures atomicity, consistency, isolation, and persistence through transaction management.

2. Data relationship processing:

Relational databases maintain data relationships through foreign key constraints, and querying relationships requires multi table join operations.

MongoDB allows nested documents and arrays, making nested relationships more flexible, but querying relationships requires querying the nested structure within the document.

Neo4j treats relationships as first-class citizens and stores them directly in the database, making accessing relationships as direct and fast as accessing data.

3. Graph data storage:

Non native storage requires storing graph data in a non specifically designed database, which may require additional indexing and query optimization.

Native storage is specifically designed for storing graph data, such as Neo4j, where data and relationships are stored together for fast access to relationships and better query performance.

4. Cypher script and graph generation:

Create user nodes and relationships.

Create an email sending relationship.

Explain the effect of Cypher script and return Jim's indirect contacts.

Advanced tasks:

5. Convert the relational table to a Neo4j graph:

Convert the relational table into a Neo4j graph and comment on both structures.

6. Neo4j graph query:

Obtain the friendship and timeline between Sally and John.

Write a Cypher script to query Moby Dick's average rating, author, Sally's age, and who read Moby Dick first.

Personal reflective:

We learned how to use Neo4j for creating and querying graph data, and understood the advantages of graph databases in handling complex relational data. Cypher query language provides powerful graph data query capabilities, making the creation and querying of graph data simple and intuitive.

Module: Big Data

Lab: BD_Lab_08_DataWarehouse

Name: Yuheng Li

Student ID: 2023020202

Group member: Yue Ma, Conge Yang

Experimental purpose:

This experiment aims to understand the basic concepts, characteristics, advantages and disadvantages of data warehouses and Lakehouses through self-study and practice, and explore these two data architectures through the Microsoft Fabric platform.

Experimental content:

Theoretical learning:

1. Data warehouse concept:

A data warehouse is an enterprise level solution used for storing, managing, and analyzing structured and semi-structured data.

The data warehouse supports ACID transactions and provides a high-performance query processing engine, suitable for storing and analyzing structured data.

2. Lakehouse concept:

Lakehouse is a data architecture platform used for storing, managing, and analyzing structured and unstructured data.

Lakehouse is built on a data lake and supports open data formats, providing seamless collaboration between data engineers and business users.

3. Advantages and disadvantages of data warehouse and Lakehouse:

Data warehouses are suitable for enterprise level applications that require high

performance, ease of management, and maintenance.

Lakehouse is suitable for scenarios that require processing large amounts of unstructured data and supports the use of tools such as Spark for data processing and analysis.

Practical operation:

Complete the recommended tutorial:

Completed the end-to-end analysis tutorial for Microsoft Fabric.

I have completed the introductory tutorial on Lakehouse and data warehouse in Microsoft Fabric.

Through practical operation, I have gained a deep understanding of the usage methods and application scenarios of data warehouses and Lakehouses.

Personal reflective:

Data warehouses and Lakehouses each have their own advantages and are suitable for different data management and analysis needs. Data warehouses are suitable for storing and analyzing structured data, while Lakehouse is suitable for handling large amounts of unstructured data. Through the Microsoft Fabric platform, it is easy to choose and switch between data warehouses and Lakehouses to meet different business needs. We learned how to use the Microsoft Fabric platform to create and manage data warehouses and Lakehouses.