

SCHEME :K

Name : _____
Roll No.: _____ Year : 20 ____ 20 ____
Exam Seat No. : _____

LABORATORY MANUAL FOR ADVANCE DATABASE MANAGEMENT (315324)



COMPUTER ENGINEERING GROUP



**MAHARASHTRA STATE BOARD OF
TECHNICAL EDUCATION, MUMBAI**
(Autonomous)(ISO21001:2018)(ISO/IEC27001:2013)

Vision

To ensure that the Diploma level Technical Education constantly matches the latest requirements of Technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

Mission

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the challenging technological & environmental challenges.

Quality Policy

We, at MSBTE are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

Core Values

MSBTE believes in the following:

- Skill development in line with industry requirements
- Industry readiness and improved employability of Diploma holders
- Synergistic relationship with industry
- Collective and Cooperative development of all stake holders
- Technological interventions in societal development
- Access to uniform quality technical education

**A Practical Manual
for
Advance Database Management
(315324)**

Semester-V

AI/ AN/ DS/ IF/ IH



**Maharashtra State Board of Technical
Education, Mumbai**

**(Autonomous) (ISO 21001:2018)(ISO/IEC 27001:2013)
'K' Scheme Curriculum**



Maharashtra State Board of Technical Education, Mumbai

(Autonomous) (ISO 21001:2018)(ISO/IEC 27001:2013)

4th Floor, Government Polytechnic Building
49, Kherwadi, Bandra (East), Mumbai – 400051



Maharashtra State Board of Technical Education Certificate

This is to certify that Mr. /Ms. Roll No.....
of the Fifth Semester of Diploma in..... Engineering/Technology
(Program Code -5K) of the Institute
(Inst. Code.....) has completed the practical work satisfactorily for the course
Advance Database Management (Course Code: 315324) for the academic year 20..... –
20..... as prescribed in the curriculum.

Place

Enrollment No.....

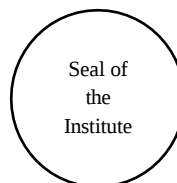
Date:

Exam Seat No.

Course Teacher

Head of the Department

Principal



Preface

The primary focus of any engineering laboratory/field work in the technical education system is to develop the much-needed industry relevant competencies and skills. Therefore, for the successful implementation of this curriculum, every practical has been designed to serve as instrumental resources to develop this industry identified competency in every student. The practical skills are difficult to develop through 'chalk and duster' activity in the classroom situation. Accordingly, the 'K' scheme laboratory manual development team designed the practical to focus on outcomes, rather than the traditional age-old practice of conducting practical's to 'verify the theory' (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the pre-determined outcomes. It is expected from each student that at least a day in advance, they have to thoroughly read the concerned practical procedure that they will do the next day and understand minimum theoretical background associated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. Students will then become aware about the skills they will achieve through procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students. Advance Database Management Systems (ADBMS) encompass a wide range of topics related to database systems, including their design and management. This lab manual extensively covers parallel and distributed database systems, database transactions, and recent developments in database technologies, providing knowledge of both structured and unstructured databases like MongoDB, SQL, and XML, while emphasizing the importance of database architecture, data mining, and techniques for managing large datasets in today's information-driven business world. By the end of the course, students will be able to design and implement efficient database systems, analyze and optimize complex queries, and work with a wide range of modern database technologies and platforms.

Although all care has been taken to check for mistakes in this laboratory manual, yet it is impossible to claim perfection especially as this is the first edition. We sincerely hope that this manual proves to be an instrumental resource in your professional journey towards becoming software developer.

Program Outcomes (POs) to be achieved through Course:

PO1	Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.
PO2	Problem analysis: Identify and analyses well-defined engineering problems using codified standard methods.
PO3	Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
PO4	Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
PO5	Engineering practices for society, sustainability and environment: Apply appropriate technology in context of society, sustainability, environment and ethical practices.
PO6	Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
PO7	Life-long learning: Ability to analyses individual needs and engage in updating in the context of technological changes.

Practical Course Outcome Matrix**Course Outcomes (COs)**

CO1	Apply the concept of concurrency control.
CO2	Analyze various database architectures.
CO3	Use Object Oriented and XML queries on Database.
CO4	Manipulate data using NoSQL commands.
CO5	Use data mining and warehousing concepts.

Sr. No.	Title of the Experiment	CO1	CO2	CO3	CO4	CO5
1	1. Install suitable Database. 2. Configure a server-based database and establish client-server connections.	✓				
2	*Execute query to implement Locked Based Protocols.	✓				
3	Study Parallel and Distributed system using Case.		✓			
4	Create database using XML 1. Create a xml file for given Application 2. Create database using xml file 3. Confirm database path 4. Show database			✓		
5	*4.1 Implement queries based on FLWOR expressions 1. Create a xml file 2. Confirm the path expression 3. Use FLWOR expression for given criteria to display result from xml file 4. Execute Join queries *4.2 Implement queries based on nested queries and sorting of results using XQuery 1. Create a xml file 2. Execute queries based on Nested queries and sorting of results using XQuery			✓		
6	*Execute query using type inheritance and table inheritance 1. Create Parent Table and child table for given application 2. Execute queries using inheritance approach by combining a data from parent, child tables				✓	
7	*Execute query using Array and Multiset types in SQL 1. Create an array Type and Multiset type 2. Use array type and Multiset type as a column				✓	

	name in table 3. Insert and display the data from table					
8	*Execute MongoDB Query using basic operations 1. Create a database for given application 2. Use DATABASE statement 3. Insert, update and delete the record for given application				✓	
9	*9.1 : Implement aggregation queries 1. Write MongoDB queries using aggregate function for given application *9.2: Execute query using find() function 1. Write MongoDB queries using find () for given application				✓	
10	*Use Data warehousing tool (ETL) 1. Extract the relevant data from the source database 2. Transform the data so that it is better suited for analytics 3. Load the data into the target database					✓
11	Implement Classification Techniques in Data Mining					✓

List of Relevant Skills

The following industry relevant skills of the competency “Apply advanced Database Management Concepts” are expected to be developed in you by performing practical’s of this laboratory manual.

1. Writing efficient queries, indexing strategies, and understanding execution plans.
2. Working with databases like MongoDB, Cassandra, or Redis for non-relational data storage.
3. Automating tasks and enforcing business rules at the database level.
4. Structuring databases for efficiency and scalability.

Guidelines to Teachers

1. Teachers should align the explanation of the topic to teaching learning outcome (TLOs).
2. Refer to laboratory learning outcome (LLOs) for the execution of the practical to focus on the defined objectives.
3. Promote life-long learning by training the students to equip themselves with essential knowledge, skills and attitudes.
4. If required, provide demonstration for the practical emphasizing on the skills that the student should achieve.
5. Teachers should give opportunity to the students for exhibiting their skills after the demonstration.
6. Provide feedback and/or suggestions and share insights to improve effectiveness.
7. Assess students' skill achievement related to COs of each unit.

Instructions for Students

1. 100% attendance is compulsory for all practical sessions.
2. Students must adhere to ethical practices.
3. All the students must follow the schedule of practical sessions, complete the assigned work/activity and submit the assignment in stipulated time as instructed by the course teacher.
4. Students should listen carefully the lecture given by teacher about importance of subject, learning structure, course outcomes.
5. Students should understand the purpose of experiment and its practical implementation.
6. Students should write the answers of the questions during practical.
7. Student should feel free to discuss any difficulty faced during the conduct of practical.
8. Students should develop web based and window-based applications as expected by the industries.
9. Student should attempt to develop related hands-on skills and gain confidence.
10. Students should develop habit to submit the write-ups on the scheduled dates and time.

Content Page**List of Practical and Formative Assessment Sheet**

Sr. No	Practical Title	Date of Performance	Date of Submission	Assessment Marks (25)	Dated Sign of Teacher	Remark
1	1. Install suitable Database. 2. Configure a server-based database and establish client-server connections.					
2	*Execute query to implement Locked Based protocols.					
3	Study Parallel and Distributed system using Case.					
4	Create database using XML 1. Create a xml file for given Application 2. Create database using xml file 3. Confirm database path 4. Show database					
5	*4.1 Implement queries based on FLWOR expressions 1. Create a xml file 2. Confirm the path expression 3. Use FLWOR expression for given criteria to display result from xml file 4. Execute Join queries *4.2 Implement queries based on nested queries and sorting of results using XQuery 1. Create a xml file 2. Execute queries based on Nested queries and sorting of results using XQuery					

6	*Execute query using type inheritance and table inheritance 1. Create Parent Table and child table for given application 2. Execute queries using inheritance approach by combining a data from parent, child tables					
7	*Execute query using Array and Multiset types in SQL 1. Create an array Type and Multiset type 2. Use array type and Multiset type as a column name in table 3. Insert and display the data from table					
8	*Execute MongoDB Query using basic operations 1. Create a database for given application 2. Use DATABASE statement 3. Insert, update and delete the record for given application					
9	*9.1 : Implement aggregation queries 1. Write MongoDB queries using aggregate function for given application *9.2: Execute query using find() function 1. Write MongoDB queries using find () for given application					
10	*Use Data warehousing tool (ETL) 1. Extract the relevant data from the source database					

	2. Transform the data so that it is better suited for analytics 3. Load the data into the target database					
11	Implement Classification Techniques in Data Mining					
Total						

***Total marks to be transferred to proforma published by MSBTE Note:**

- '*' Marked Practical's (LLOs) Are mandatory.
- Minimum 80% of above list of lab experiment are to be performed.
- Judicial mix of LLOs are to be performed to achieve desired outcomes.

Practical No. 1**i. Install suitable Database****ii. Configure a server-based database and establish client-server connections.****I. Practical Significance**

An Advanced Database Management System (ADBMS) is essential for managing large-scale, complex, and distributed data environments across modern industries. Applications of ADBMS span cloud computing, data analytics, banking systems, enterprise applications, e-commerce, healthcare, and telecommunications.

In this practical, students will gain hands-on experience with modern database technologies and understand how advanced systems handle real-time, large-volume data efficiently. They should understand the basic functioning of ADBMS and how it differs from traditional systems in terms of scalability, flexibility, and performance. Installing MongoDB, a leading NoSQL database, and exploring its document-oriented data model. Configuring a server-based database environment, enabling the deployment of databases on a centralized server and to establish client-server connections, allowing multiple users or applications to access and interact with the database concurrently.

II. Industry/Employer Expected Program Outcomes (POs)

Manage both structured and unstructured data using various tools for Database.

III. Course level learning outcomes

CO1: Apply the concept of concurrency control.

IV. Laboratory Learning Outcome(s)

LLO 1.1: Understanding Server System Architecture in Databases

V. Relevant Affective domain related Outcome(s)

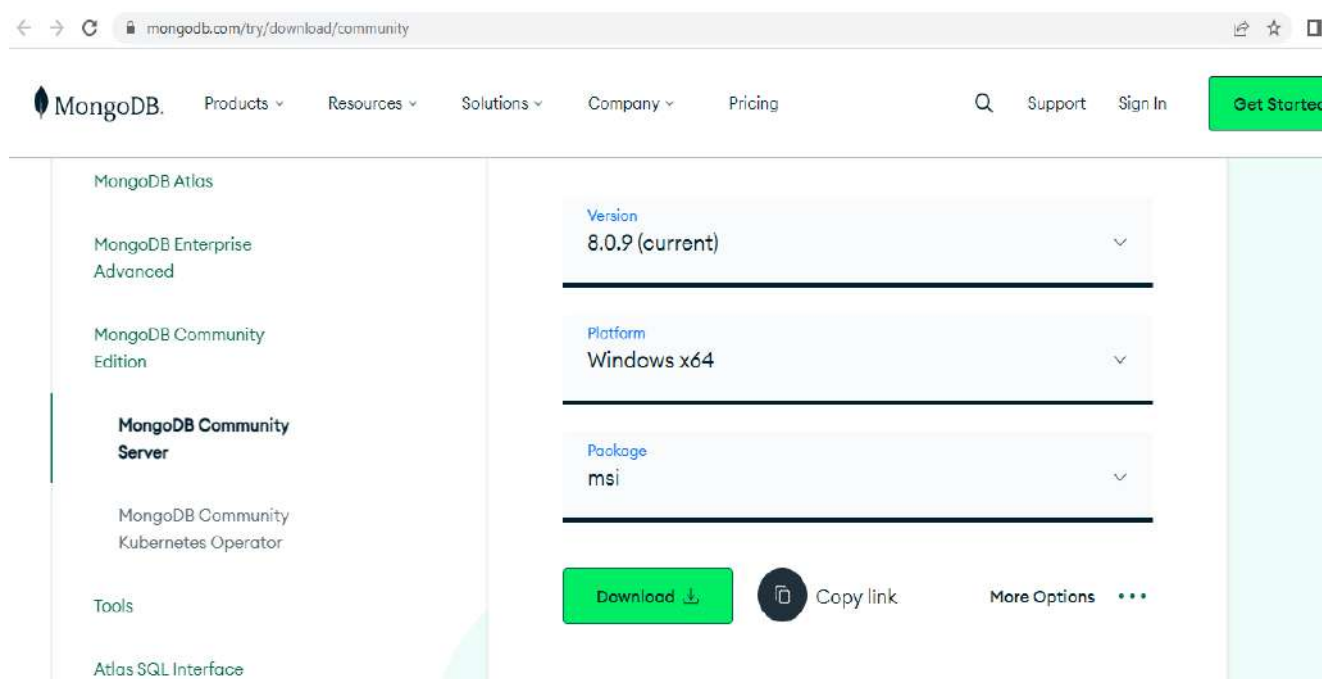
1. Follow ethical practices.
2. Install appropriate MONGODB software.

VI. Relevant Theoretical Background

MongoDB is a popular open-source NoSQL database known for its flexible document-oriented data model and scalability. Unlike traditional relational databases, MongoDB stores data in JSON-like documents, allowing for flexible data structures and easier integration with dynamic applications

Step1:

Open any browser and navigate to the MongoDB Download Center.

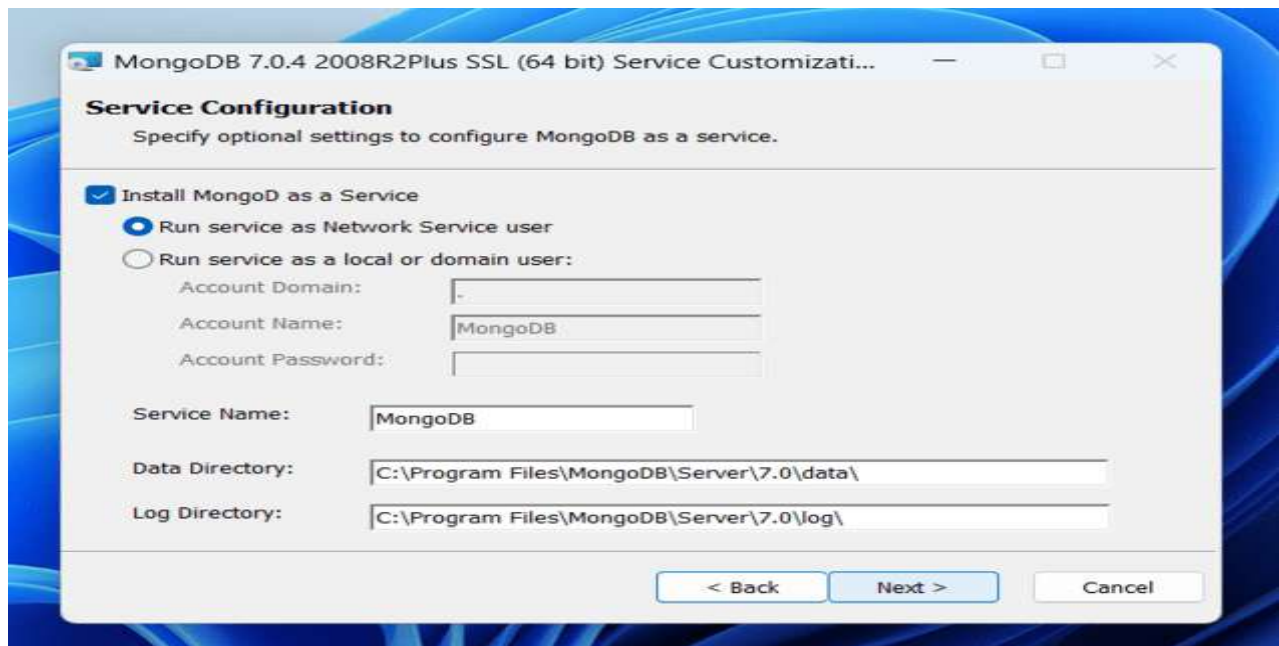


Step 2: Install MongoDB

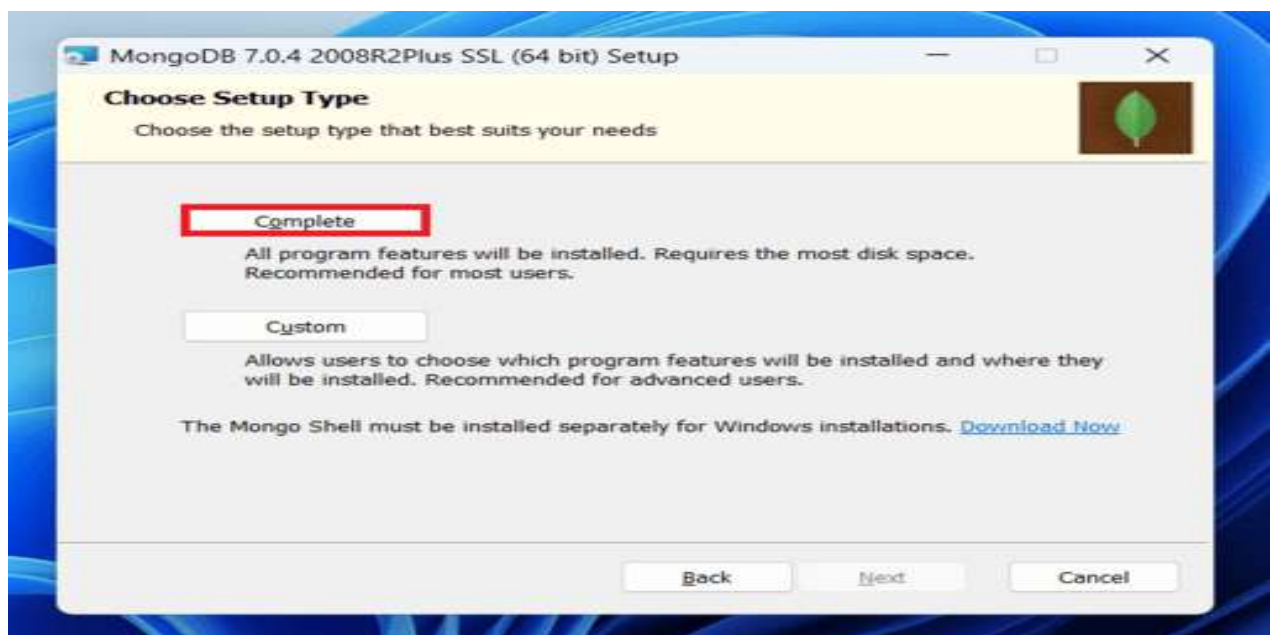
- When the download is complete open the **msi file** and click the **next button** in the startup screen:



- Now accept the **End-User License Agreement** and click the next button:

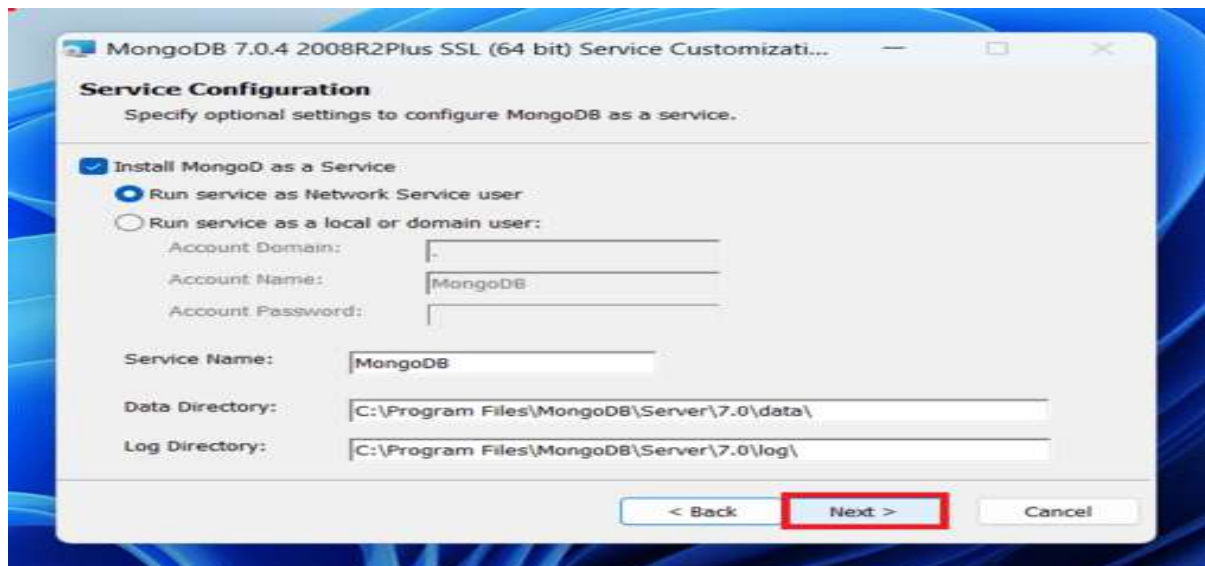


- Now select the **complete option** to install all the program features. Here, if you can want to install only selected program features and want to select the location of the installation, then use the **Custom option**:

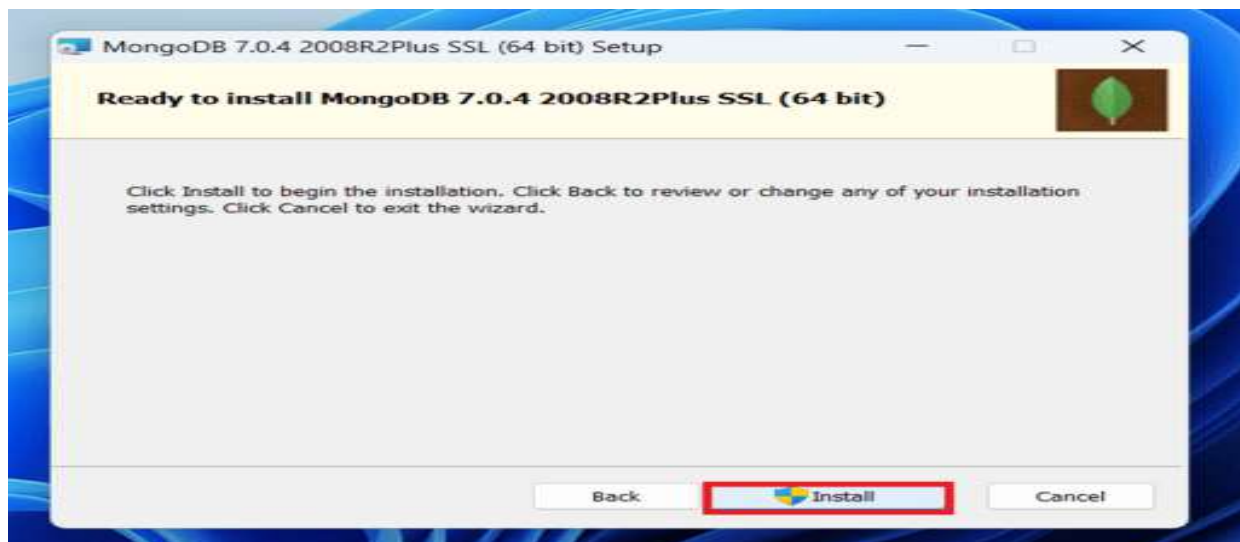


Step 3: Configure MongoDB Service

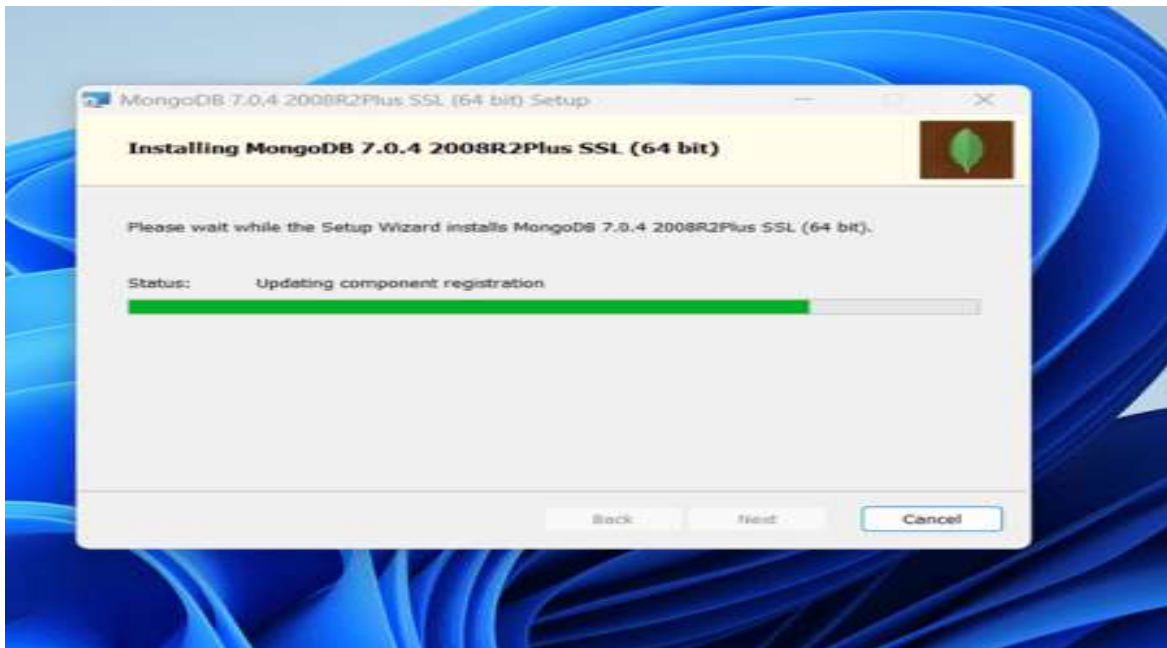
- Select “**Run service as Network Service user**” and copy the path of the data directory.
Click Next:



- Click the **Install** button to start the MongoDB installation process:



- After clicking on the install button installation of MongoDB begins:

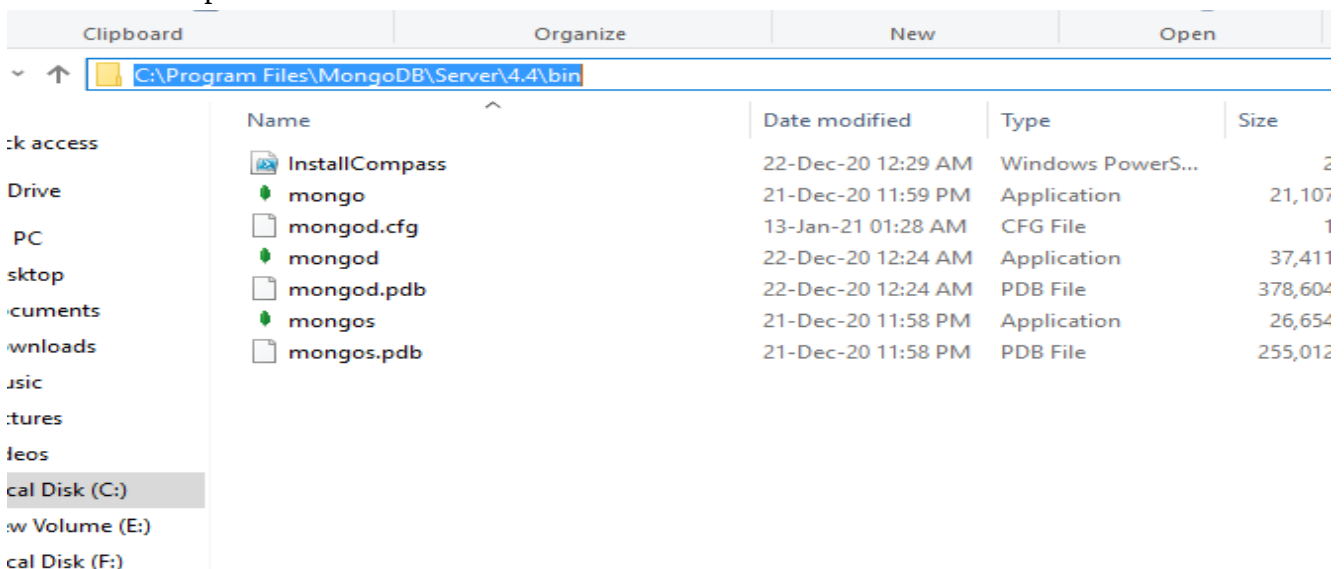


Step 4: Complete Installation

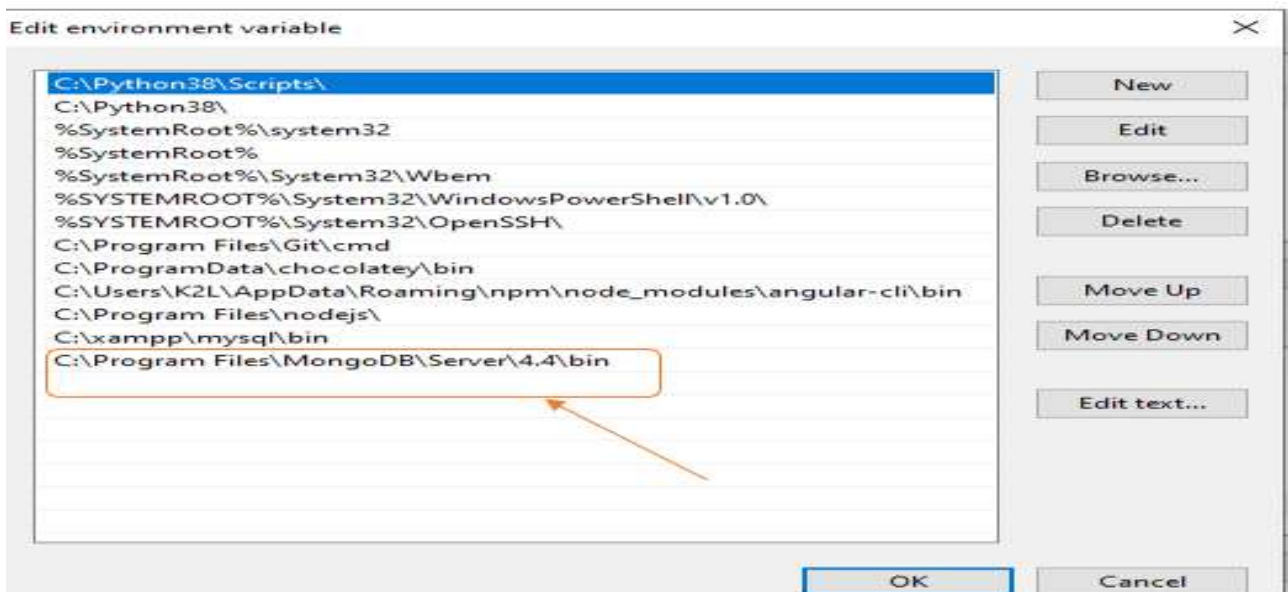
- Now click the **Finish** button to complete the MongoDB installation process:

Step 5: Set Environment Variables

- Now we go to the location where MongoDB installed in step 5 in your system and copy the **bin** path:



- Now, to create an environment variable open system **properties** >> **Environment Variable** >> **System variable** >> **path** >> **Edit Environment variable**
- paste the copied link to your environment system and **click Ok**:

**Step 6:**

Start the MongoDB Server and also Verify Installation and Connection.

Run MongoDB Server (mongod)**Step 1. Start MongoDB Service**

- After setting the environment variable, we will run the MongoDB server, i.e. **mongod**.
- So, open the **command prompt** and run the following command:

```
mongod
```

When you run this command you will get an error i.e. **C:/data/db/ not found**.

Step 2. Create Required Folders

- Now, Open **C** drive and create a folder named "**data**"
- Inside the **data** folder create another folder named "**db**".

Step 3. Restart MongoDB

After creating these folders. Again open the command prompt and run the following command:

```
mongod
```

Now, this time the MongoDB server(i.e., mongod) will run successfully.

```

C:\Users\Nikhil Chhipa>mongo
{"t":{"$date":"2021-01-31T00:56:54.081+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx"
ify --sslDisabledProtocols 'none'}}
{"t":{"$date":"2021-01-31T00:56:54.087+05:30"},"s":"W", "c":"ASIO", "id":22601, "ctx"
}
{"t":{"$date":"2021-01-31T00:56:54.088+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx"
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I", "c":"STORAGE", "id":4615611, "ctx"
bPath":"C:/data/db/", "architecture":"64-bit", "host":"DESKTOP-L9MUQ7N"}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I", "c":"CONTROL", "id":23398, "ctx"
rgetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I", "c":"CONTROL", "id":23403, "ctx"
gitVersion":"913d6b62acfb344dde1b116f4161360acd8fd13","modules":[],"allocator":"tcmalloc",
}}}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I", "c":"CONTROL", "id":51765, "ctx"
ndows 10","version":"10.0 (build 14393)"}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I", "c":"CONTROL", "id":21951, "ctx"
{"t":{"$date":"2021-01-31T00:56:54.157+05:30"},"s":"I", "c":"STORAGE", "id":22270, "ctx"
:{"dbpath":"C:/data/db/", "storageEngine":"wiredTiger"}}
{"t":{"$date":"2021-01-31T00:56:54.158+05:30"},"s":"I", "c":"STORAGE", "id":22315, "ctx"
ize=1491M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statisti
le_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statisti
ess],"}}
{"t":{"$date":"2021-01-31T00:56:54.395+05:30"},"s":"I", "c":"STORAGE", "id":22430, "ctx"
95788][3708:140713908197088], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 20 thr
{"t":{"$date":"2021-01-31T00:56:54.631+05:30"},"s":"I", "c":"STORAGE", "id":22430, "ctx"

```

VII. Resources required (Additional)

Sr. No	Instrum ent/ Object	Specification	Quantity
1	Operatin g system	Windows 10 or higher	01 system for each student
2	Software	MongoDB Software: ~200 MB for binaries Data Storage: At least 2 GB free space (more depending on data volume) Journal Files and Logs: Additional 1–2 GB recommended minimum of 2 GB RAM	

VIII. Precautions to be followed

1. Check System Requirements
2. Download from Official Source
3. Create Required Folders and Set Permissions.
4. Avoid Port Conflicts

IX. Conclusion

.....

.....

.....

X. References/ Suggestions for further Reading

1. <https://www.mongodb.com/docs/manual/installation/>
2. <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows>
3. <https://www.geeksforgeeks.org/how-to-install-mongodb-on-windows/>

XI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 2: *Execute query to implement Locked Based protocols.**I. Practical Significance**

Implementing lock-based protocols in database systems is crucial for ensuring data consistency and integrity. Query to implement locked Based Protocol, ensures data consistency in concurrent environments. Preventing anomalies in multi-user transactions, managing concurrent access in Real-Time Systems

II. Industry / Employer Expected Outcome(s)

Manage both structured and unstructured data using various tools for Database.

III. Course Level Learning Outcomes(s)

CO1 - Apply the concept of concurrency control

IV. Laboratory Learning Outcome(s)

LLO 2.1 Implement Locked Based protocols

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Involve teamwork, good communication and cooperation skills.
3. Approach challenges with curiosity and adaptability.

VI. Relevant Theoretical Background**Lock Based Protocol:**

A lock is a mechanism to control concurrent access to a data item .Data items can be locked in two modes:

1. **Exclusive (X) mode.** Data item can be both read as well as written. X-lock is requested using lock-X instruction.
2. **Shared (S) mode.** Data item can only be read. S-lock is requested using lock-S instruction. Lock requests are made to concurrency-control manager. Transaction can proceed only after request is granted.

Steps to implement sql query on single phase lock based protocol:

1. **Design the Table**
2. **Begin a Transaction** (BEGIN TRANSACTION)
3. **Acquire All Locks at Once (Initial Phase)**
For exclusive locks (X-locks)- SELECT ... FOR UPDATE
For shared locks (S-locks)-SELECT ... FOR SHARE
4. **Perform the Desired Operations** (e.g., UPDATE, INSERT, etc.)
5. **Release the Locks (At Commit)**

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	MySQL/Oracle/SQL server/MongoDB or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

[illegible]

Practical No. 3: Study Parallel and Distributed system using Case**I. Practical Significance**

Studying Parallel and Distributed Systems using case studies has significant practical benefits, especially in fields like computing, engineering, and data science. Case studies illustrate how real-world applications leverage parallel and distributed computing, making concepts easier to grasp. By analyzing existing systems, students can identify best practices for efficiency, scalability, and fault tolerance.

II. Industry / Employer Expected Outcome(s)

Manage both structured and unstructured data using various tools for Database.

III. Course Level Learning Outcomes(s)

CO2 - Analyze various database architectures

IV. Laboratory Learning Outcome(s)

LLO 3.1 Understand Parallel and Distributed Systems through Case Study

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Involve teamwork, good communication and cooperation skills.
3. Approach challenges with curiosity and adaptability.

VI. Relevant Theoretical Background

- **Parallel System:** - A Parallel system is designed to improve performance by executing multiple operations simultaneously. It uses multiple processors and storage devices to handle large-scale data efficiently.

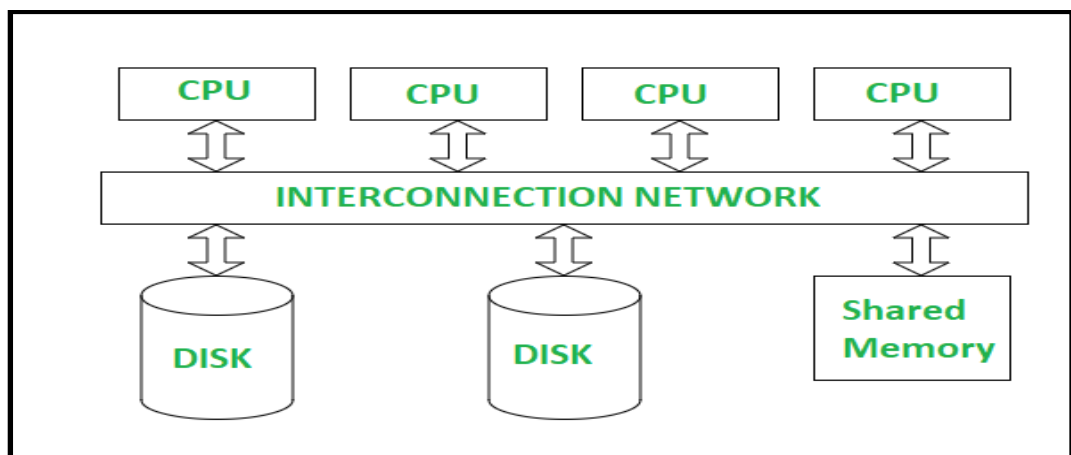


Fig no 3.1: Parallel System

- **Distributed System:** - A Distributed System is a system where data is stored across multiple locations and managed collectively. It ensures that users can access and manipulate data as if it were stored in a single database.

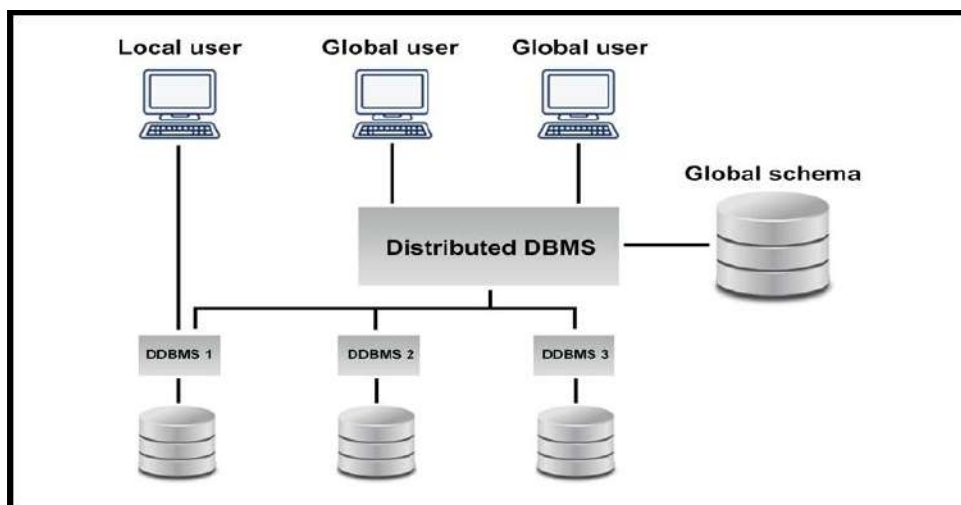


Fig no 3.2: Distributed System

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	MySQL/Oracle/SQL server/MongoDB or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

VIII. Conclusion

.....

.....

.....

IX. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. What is a Parallel System?
2. What is a Distributed System?
3. Write a case study for Parallel Systems?

4. Write a case study for a Distributed System?

[illegible]

.....

.....

.....

.....

.....

.....

.....

.....

X. References:

1. <https://www.geeksforgeeks.org/introduction-of-parallel-database/>
2. <https://www.geeksforgeeks.org/distributed-database-system/>
3. <https://www.geeksforgeeks.org/operating-system-difference-between-distributed-system-and-parallel-system/>

XI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 4:**i. Create a xml file for given Application****ii. Create database using xml file****iii. Confirm database path, Show database****I. Practical Significance**

This experiment serves as an entry point for students to understand how modern database management systems (DBMS) work. The use of BaseX—an XML database—introduces students to a specialized kind of DBMS used for storing hierarchical and semi-structured data, which is different from traditional relational database systems like MySQL or PostgreSQL. By working with an XML database, students can compare and contrast the different data models (relational vs. hierarchical/document-based) and understand how each one is suited for different types of data storage and retrieval.

With this experiment, students gain hands-on experience in managing XML-based data, which is a common format for web services, data exchange, and configuration files. They learn how to use tools like BaseX to efficiently store, query, and manipulate hierarchical data.

II. Industry/Employer Expected Program Outcomes (POs)

Manage both structured and unstructured data using various tools for Database.

III. Course level learning outcomes

CO3: Use Object Oriented and XML queries on Database

IV. Laboratory Learning Outcome(s)

LLO 4.1: Create a database using XML Attributes and Elements.

V. Relevant Affective domain related Outcome(s)

1. Appreciation for XML and NoSQL Technologies
2. Confidence in Working with Advanced Database Tools.

VI. Relevant Theoretical Background

Creating a database using XML involves understanding both XML (eXtensible Markup Language) and database management systems that support hierarchical or semi-structured data, such as BaseX. The theoretical background encompasses the principles of XML, NoSQL databases, and the relationship between structured, semi-structured, and unstructured data.

BaseX is an XML Database that provides efficient storage and querying of XML data. It is designed for managing large XML datasets, offering a powerful query language called XQuery for querying XML data efficiently. BaseX is often used in situations where hierarchical or semi-structured data is prevalent, such as content management systems, web scraping, and data integration tasks.

1. Create a xml file for given Application

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<bookstore>
  <book category="Fiction">
    <title lang="en">The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
    <year>1925</year>
    <price>10.99</price>
  </book>
  <book category="Programming">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

This XML file represents an online store where we have products with attributes like id, name, category, price, and stock.

2. Create a Database Using the XML File in BaseX

Once you have the bookstore.xml file, follow these steps to create a database in BaseX using this file.

1. Open BaseX: Launch BaseX and open the BaseX GUI or BaseX command-line interface (CLI).
2. Create the Database:
 - i. In the BaseX GUI, click on Database > New....
 - ii. Select the bookstore.xml file and provide a name for the database (e.g., bookstore).
 - iii. Click OK to create the database.

3. Confirm Database Path

To confirm the database path (where the database is stored on your system), follow these steps:

1. In BaseX GUI:
 - Open the Database menu.
 - Select Properties from the drop-down.
 - This will show the file path of the bookstore database.
- or
- To check where the database is stored on your file system, use the following command:

info database

This will display metadata about the currently opened database, such as:

```

Database Properties
NAME: bookstore
SIZE: 5553 b
NODES: 37
DOCUMENTS: 1
BINARIES: 0
VALUES: 0
TIMESTAMP: 2025-05-12T20:35:36.438Z
UPTODATE: true

Resource Properties
INPUTPATH: C:/Users/comp/Documents/bookstore.xml
INPUTSIZE: 455 b
INPUTDATE: 2025-05-12T20:34:15.287Z

```

4. Show database

To list all the available databases in BaseX, you should use the **list** command:



Result

```

Name      Resources  Size  Input Path
-----
bookstore  1           5553  C:/Users/comp/Documents/bookstore.xml

1 database(s).

```

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1	Operating system	Windows 10 or higher Minimum: 200 MB for BaseX installation Minimum of 2 GB RAM	01 System for each student
2	Software	BaseX	

VIII. Precautions to be followed

1. Always validate your XML file before importing.
2. Use proper opening and closing tags.
3. Ensure that the XML file path and file name are correct when creating a database in BaseX.
4. Make sure BaseX is properly installed and the Java Runtime Environment (JRE) is available.

IX. Conclusion

.....

.....

.....

X. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. What is XML and how can it be used in database creation?
2. What are the advantages of using XML to define or populate a database?
3. What are the differences between storing XML in a native XML database vs. a relational database?
4. What are the advantages and disadvantages of using XML as a data source for databases?
5. Design an XML file that stores student records for a school management system. Include fields for: Student ID, Name, Class, Address, Department.
6. What BaseX commands are used for database management and user control?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[illegible]

VII. Exercise

1. Write an XML file for a hospital management system to record Patient ID, Name, Gender, Age, Diagnosis, Admitted Ward.

Then, using BaseX:

- 1.1 Confirm the database path where the XML file is stored or created.
- 1.2 Create a BaseX database using the XML file.
- 1.3 Display the contents of the database using an appropriate XQuery expression.

VIII. References/ Suggestions for further Reading

1. https://www.tutorialspoint.com/xml/xml_databases.html
2. https://docs.basex.org/12/Commands#show_options
3. <https://www.geeksforgeeks.org/what-is-xml-data-model-in-dbms/>

IX. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 5:***5.1 Implement queries based on FLWOR expressions*****5.2 Implement queries based on nested queries and sorting of results using XQuery****I. Practical Significance**

In the context of **FLWOR** expressions (short for **For, Let, Where, Order by, Return**) in XML query languages, the goal is to implement queries that extract or manipulate data based on certain conditions. FLWOR expressions are commonly associated with **XQuery**, a language for querying XML data.

Nested queries and sorting in XQuery have significant real-world applications, especially when dealing with complex XML data structures such as databases, large datasets, and hierarchical documents. These techniques are commonly used in scenarios where data must be filtered, aggregated, or transformed in meaningful ways.

II. Industry/Employer Expected Program Outcomes (POs)

Manage both structured and unstructured data using various tools for Database.

III. Course level learning outcomes

CO3: Use Object Oriented and XML queries on Database

IV. Laboratory Learning Outcome(s)

LLO 5.1 Implement queries based on FLWOR expressions using XQuery.

LLO 5.2 Implement joins queries using XQuery.

LLO 5.3 Implement nested queries using XQuery.

V. Relevant Affective domain related Outcome(s)

1. Appreciation for XML and NoSQL Technologies
2. Confidence in Working with Advanced Database Tools.

VI. Relevant Theoretical Background

FLWOR expressions are part of the XQuery language, commonly used for querying XML data. The term "FLWOR" stands for **For, Let, Where, Order by, and Return**, which are the core components that help in retrieving and manipulating data from XML documents. FLWOR stands for the five components of an **XQuery** expression, and it is used for querying XML data. Here's the breakdown:

For: Loops through items in an XML document.

Let: Defines and assigns values to variables.

Where: Filters the items, keeping only those that match a condition.

Order by: Sorts the results based on a specified condition.

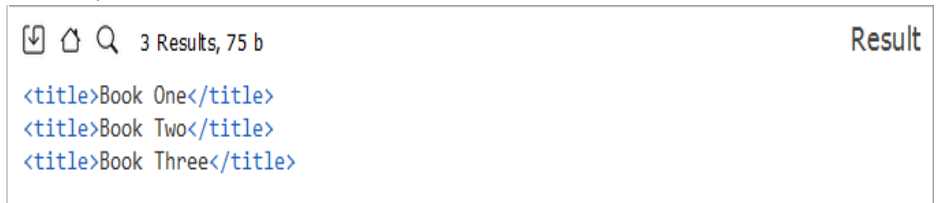
Return: Defines what the output of the query should be.

In XQuery, nested queries allow you to perform complex operations where one query is used within another. Additionally, sorting can be applied to the results of these nested queries to order the output

based on specific criteria. Let's walk through how you can implement both nested queries and sorting in XQuery.

Create a xml file for given Application

```
<library>
  <book>
    <title>Book One</title>
    <author>Author A</author>
    <year>2021</year>
    <genre>Fiction</genre>
  </book>
  <book>
    <title>Book Two</title>
    <author>Author B</author>
    <year>2020</year>
    <genre>Non-fiction</genre>
  </book>
  <book>
    <title>Book Three</title>
    <author>Author A</author>
    <year>2019</year>
    <genre>Fiction</genre>
  </book>
</library>
```

FOR	<pre>for \$book in doc("C:\Users\comp\Documents\books.xml")//book return \$book/title</pre> 
LET	<pre>let \$avgYear := avg(doc("C:\Users\comp\Documents\books.xml")//year) return <average>{\$avgYear}</average></pre> 
WHERE	<pre>for \$book in doc("C:\Users\comp\Documents\books.xml")//book where \$book/year > 2020 return <result>{\$book/title,\$book/year}</result></pre>

	   1 Result, 57 b Result <pre><result><title>Book One</title><year>2021</year></result></pre>
ORDER	for \$book in doc("C:\Users\comp\Documents\books.xml")//book order by \$book/title return \$book/title    3 Results, 75 b Result <pre><title>Book One</title> <title>Book Three</title> <title>Book Two</title></pre>
RETURN	for \$book in doc("C:\Users\comp\Documents\books.xml")//book return <bookInfo> <name>{\$book/title/text()}</name> <cost>{\$book/year/text()}</cost> </bookInfo>    3 Results, 183 b Result <pre><bookInfo><name>Book One</name><cost>2021</cost></bookInfo> <bookInfo><name>Book Two</name><cost>2020</cost></bookInfo> <bookInfo><name>Book Three</name><cost>2019</cost></bookInfo></pre>

Join Query

Just like in relational databases, **joins** in XML help you **combine data** from two related sets of XML elements based on a shared value (usually like a key). While XML doesn't support joins natively like SQL, you can **simulate joins** using languages like **XQuery**, **XPath**, or in code using **Python**, **Java**, etc.

Create a xml file for given Application

library.xml	author.xml
<pre><library> <book> <title>Book A</title> <author>John Doe</author> <price>20</price> </book> <book> <title>Book B</title> <author>Jane Smith</author></pre>	<pre><authors> <author> <name>John Doe</name> <country>USA</country> </author> <author> <name>Jane Smith</name> <country>UK</country> </author></pre>

<pre> <price>15</price> </book> <book> <title>Book C</title> <author>John Doe</author> <price>25</price> </book> <book> <title>Book D</title> <author>John Doe</author> <price>30</price> </book> <book> <title>Book E</title> <author>Jane Smith</author> <price>10</price> </book> </library> </pre>	<pre> </authors> </pre>
--	-------------------------------

Types of Join	XML Query	Output
Inner Join: Returns only records that match in both datasets.	<pre> for \$book in /library/book let \$author := /authors/author[name = \$book/author] return <bookInfo> <title>{\$book/title}</title> <author>{\$book/author}</author> <country>{\$author/country}</country> </bookInfo> </pre>	<pre> <bookInfo> <title>Book A</title> <author>John Doe</author> <country>USA</country> </bookInfo> <bookInfo> <title>Book B</title> <author>Jane Smith</author> <country>UK</country> </bookInfo> </pre>

Left Outer Join Returns all records from the left dataset (books) , with data from the right (authors) if available . If no match, fills with default (e.g., "Unknown").	<pre> for \$book in /library/book let \$author := /authors/author[name = \$book/author] return <bookInfo> <title>{\$book/title}</title> <author>{\$book/author}</author> <country>{if (\$author) then \$author/country else "Unknown"}</country> </bookInfo> </pre>	<pre> <bookInfo> <title>Book A</title> <author>John Doe</author> <country>USA</country> </bookInfo> <bookInfo> <title>Book B</title> <author>Jane Smith</author> <country>UK</country> </bookInfo> <bookInfo> <title>Book C</title> <author>Unknown Author</author> <country>Unknown</country> </bookInfo> </pre>
Right Outer Join: Returns all records from the right dataset (authors) , even if they don't match a book.	<pre> for \$author in /authors/author let \$books := /library/book[author = \$author/name] return <authorInfo> <name>{\$author/name}</name> <country>{\$author/country}</country> <books> { for \$b in \$books return <title>{\$b/title}</title> } </books> </authorInfo> </pre>	<pre> <authorInfo> <name>John Doe</name> <country>USA</country> <books> <title>Book A</title> </books> </authorInfo> <authorInfo> <name>Jane Smith</name> <country>UK</country> <books> <title>Book B</title> </books> </authorInfo> </pre>
Full Outer Join Includes all records from both datasets. Matches where possible, but includes	<pre> (for \$book in /library/book let \$author := /authors/author[name = \$book/author] return <bookInfo> <title>{\$book/title}</title> <author>{\$book/author}</author> <country>{if (\$author) then \$author/country else </pre>	<pre> <bookInfo> <title>Book A</title> <author>John Doe</author> <country>USA</country> </bookInfo> <bookInfo> <title>Book B</title> <author>Jane Smith</author> <country>UK</country> </bookInfo> </pre>

unmatched records from both sides.	<pre> "Unknown"}</country> </bookInfo>) union (for \$author in /authors/author let \$matched := /library/book[author = \$author/name] where empty(\$matched) return <authorInfo> <name>{\$author/name}</name> <country>{\$author/country}</country> <note>No books found</note> </authorInfo>) </pre>	<pre> <bookInfo> <title>Book C</title> <author>Unknown Author</author> <country>Unknown</country> </bookInfo> <!-- Author entries without books --> <!-- (none in this case, but this part would appear if unmatched authors existed) --> </pre>
Cross Join (Cartesian Product) Each element from one dataset is paired with every element from another. Rarely useful unless filtering.	<pre> for \$book in /library/book for \$author in /authors/author return <combo> <book>{\$book/title}</book> <author>{\$author/name}</author> </combo> </pre>	<pre> <combo><book>Book A</book><author>John Doe</author></combo> <combo><book>Book A</book><author>Jane Smith</author></combo> <combo><book>Book B</book><author>John Doe</author></combo> <combo><book>Book B</book><author>Jane Smith</author></combo> <combo><book>Book C</book><author>John Doe</author></combo> <combo><book>Book C</book><author>Jane Smith</author></combo> </pre>

Nested Query

In XQuery, **nested queries** allow you to perform operations where one query is embedded within another, enabling more complex data manipulations. This is useful when you need to work with data from multiple levels or when one query depends on the result of another. Below are a few examples demonstrating how to use nested queries in XQuery.

XML Query	Explanation:
<pre> for \$author in distinct- values(/library/book/author) let \$booksByAuthor := /library/book[author = \$author] where count(\$booksByAuthor) > 1 return <author> <name>{ \$author }</name> <books> { for \$book in \$booksByAuthor return <book> <title>{ \$book/title }</title> <price>{ \$book/price }</price> } </books> </author> </pre>	Outer Query: The query iterates over all distinct authors using distinct-values(/library/book/author). Nested Query: For each author, the query retrieves all books written by that author using the let \$booksByAuthor := /library/book[author = \$author] clause. Filtering: The where clause filters the authors, keeping only those who have written more than one book (count(\$booksByAuthor) > 1). Return Clause: For each qualifying author, the query returns a new XML structure with the author's name and their books, including the title and price for each book.

Output

```

<author>
  <name>John Doe</name>
  <books>
    <book>
      <title>Book A</title>
      <price>20</price>
    </book>
    <book>
      <title>Book C</title>
      <price>25</price>
    </book>
    <book>
      <title>Book D</title>
      <price>30</price>
    </book>
  </books>
</author>
<author>
  <name>Jane Smith</name>
  <books>

```

```

<book>
  <title>Book B</title>
  <price>15</price>
</book>
<book>
  <title>Book E</title>
  <price>10</price>
</book>
</books>
</author>

```

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1	Operating system	Windows 10 or higher Minimum: 200 MB for BaseX installation Minimum of 2 GB RAM	01 System for each student
2	Software	BaseX	

VIII. Precautions to be followed

1. Correct XML Structure
2. Check Query Compatibility
3. Test with Small Data Sets First
4. Ensure Proper Data Types
5. Optimization and Performance

IX. Conclusion

.....

.....

.....

X. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. What is a FLWOR expression in XQuery?
2. Explain the role of each component in a FLWOR expression.

3. What is the difference between the For and Let clauses in a FLWOR expression?
4. What are some practical use cases of FLWOR expressions in real-world applications?
5. What is a nested query in XQuery? How does it differ from a regular query?
6. Explain the concept of "sorting results" in XQuery. How would you sort data in descending order?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

This image shows a full page of primary-ruled paper. It features multiple horizontal rows, each defined by two parallel dashed lines. The rows are evenly spaced across the entire page, providing a guide for handwriting practice. There are no margins, text, or other markings present.

VIII. Exercise

1. **Implement a FLWOR Query for an E-Commerce Store. The data is stored in an XML format, with each product having attributes like id, name, category, price, and rating.**
2. **Implementing Nested Queries and Sorting Results. Write an XQuery that:**
 - 2.1. **Uses a nested query to find the average price of Electronics products.**
 - 2.2. **Selects only those products in the Electronics category with a price higher than the average.**
 - 2.3. **Sorts the result by rating in descending order.**
 - 2.4. **Returns the name, price, and rating of those products.**

IX. References/ Suggestions for further Reading

1. https://www.tutorialspoint.com/xml/xml_databases.html
2. https://docs.basex.org/12/Commands#show_options
3. <https://www.geeksforgeeks.org/what-is-xml-data-model-in-dbms/>

IX. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 6 *Execute query using type inheritance and table inheritance**1. Create Parent Table and child table for given application****2. Execute queries using inheritance approach by combining a data from parent, child tables****I. Practical Significance**

Type inheritance applies to named row types only. You can use inheritance to group named row types into a type hierarchy in which each subtype inherits the representation (data fields) and the behavior (UDRs, aggregates, and operators) of the supertype under which it is defined. A type hierarchy provides. It encourages modular implementation of your data model also ensures consistent reuse of schema components and no data fields are accidentally left out.

Table inheritance is the property that allows a table to inherit the behavior (constraints, storage options, triggers) from the supertable above it in the table hierarchy. A table hierarchy is the relationship that you can define among tables in which subtables inherit the behavior of supertables.

II. Industry / Employer Expected Outcome(s)

Develop Database which implements inheritance and Supports Reusability.

III. Course Level Learning Outcomes(s)

CO4 - Manipulate data using NoSQL commands

IV. Laboratory Learning Outcome(s)

LLO 6.1 Execute queries using type inheritance and table inheritance in SQL

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Use appropriate DBMS software.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Relevant Theoretical Background

Type inheritance applies to named row types only. You can use inheritance to group named row types into a *type hierarchy* in which each subtype inherits the representation (data fields) and the behavior (UDRs, aggregates, and operators) of the supertype under which it is defined.

Type inheritance applies to named row types only. You can use inheritance to group named row types into a type hierarchy in which each subtype inherits the representation (data fields) and the behavior (aggregates and operators) of the supertype under which it is defined.

Row type is used to specify the type of a composite attribute of a relation. A row type can be specified using the syntax given below:

```
CREATE TYPE <Type_Name> AS [ROW] (<attribute1> <data_type1>,  
                                <attribute2> <data_type2>, :<attribute_n"> <data_type>);
```

Table inheritance is the property that allows a table to inherit the behavior (constraints, storage options, triggers) from the supertable above it in the *table hierarchy*. In a table

hierarchy, a subtable automatically inherits the properties from its supertable are All constraint definitions (primary key, unique, and referential constraints), Storage option, All triggers, Indexes, Access method.

```
CREATE TABLE cities
(
    name      text,
    population float,
    elevation  int  -- in feet
);
CREATE TABLE capitals
(
    state      char(2)
) INHERITS (cities);
```

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	MySQL/Oracle/SQL server/MongoDB or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

VIII. Conclusion

.....

.....

.....

IX. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State the advantages of table inheritance
2. State the advantages of type inheritance
3. Which properties are inherited subtable
4. Write syntax of type inheritance
5. Implement table inheritance for employee table

.....

.....

[illegible]

X. Exercise:

1. Create two table student and city. And implement table inheritance
2. Perform type inheritance on address and employee table
3. What are the basic difference between type and table inheritance

XI. References:

1. www.w3school.com
2. https://help.hcl-software.com/onedb/1.0.1/ddi/ids_ddi_119.html
3. <https://www.postgresql.org/docs/current/ddl-inherit.html>

XII. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 7 Execute query using Array and Multiset types in SQL

- 1. Create an array Type and Multiset type**
- 2. Use array type and Multiset type as a column name in table**
- 3. Insert and display the data from table**

I. Practical Significance

In the world of databases, SQL (Structured Query Language) is a widely used language for managing and manipulating data. One of the key features of SQL DBMS (Database Management Systems) is its support for various data types, including arrays and multisets. Arrays and multisets provide powerful capabilities for storing & querying structured data in SQLDBMS. By using these types, you can organize & manipulate data more efficiently, allowing for more flexible and expressive database design.

II. Industry / Employer Expected Outcome(s)

Develop Database which implements Array and Multiset types in SQL

III. Course Level Learning Outcomes(s)

CO4 - Manipulate data using NoSQL commands

IV. Laboratory Learning Outcome(s)

LLO 7.1 Implement queries using Array and Multiset types in SQL.

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Use appropriate DBMS software.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Relevant Theoretical Background

An array is an ordered set of elements of a single built-in data type. Arrays can be used only in the following contexts:

- RETURN data types from SQL functions
- Parameters to SQL procedures
- SQL variables that are declared in SQL functions
- SQL variables that are declared in SQL procedures

Example:- `CREATE TABLE Students ( ID SERIAL PRIMARY KEY, Name VARCHAR (50),`

`Age INTEGER, Grades INTEGER [] );`

The **MULTISET** data type is a collection type that stores a non-ordered set that can include duplicate element values. The elements in a **MULTISET** have no ordinal position. That is, there is no concept of a first, second, or third element in a **MULTISET**.

Example:- `CREATE TABLE shopping_cart(id integer, items fruits_muliset);`

INSERT into shopping_cart (1,fruits_multiset('Apple','Banana','Apple'));

SELECT DISTINCT item from TABLE(shopping_cart.item);

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	MySQL/Oracle/SQL server/MongoDB or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

VIII. Conclusion

.....

.....

.....

IX. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State the use of Array and Multiset.
2. Write syntax and example Array.
3. Write syntax and example of multiset.
4. Create one table to implement array.

.....

.....

.....

.....

.....

.....

.....

.....

.....

[illegible]

X. Exercise:

- 1. Create table employee with members id,name and salary.**
 - i. Use array for salary which accept 3 value as Basic,Hra,Da.**
 - ii. Insert 5 record.**
 - iii. Display all record.**
- 2. Create table diploma_courses with data member id,branch.**
 - i. Create branch as a multiset.**
 - ii. Insert id as 1 and add any 4 branch related to computer(for ex. CO,IF,AI).**
 - iii. Display all Multiset value**

3. Compare between Array and Multiset

XI. References:

1. <https://www.theknowledgeacademy.com/blog/sql-array/>
2. <https://edurev.in/t/267712/Array-and-Multiset-Types-in-SQL>
3. <https://www.postgresql.org/docs/current/arrays.html>

XII. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 8 Execute MongoDB query using basic operation

- 1. Create a database for given application**
- 2. Use DATABASE statement**
- 3. Insert update and delete the record for given application**

I. Practical Significance

MongoDB is a powerful, open-source NoSQL database that offers a document-oriented data model, providing a flexible alternative to traditional relational databases. Unlike SQL databases, MongoDB stores data in BSON format, which is similar to JSON, enabling efficient and scalable data storage and retrieval.

MongoDB allows a highly flexible and scalable document structure. For example, one data document of a collection in MongoDB can have two fields whereas the other document in the same collection can have four. MongoDB is faster as compared to RDBMS due to efficient indexing and storage techniques.

MongoDB currently provides official driver support for all popular programming languages like C, C++, Rust, C#, Java, Node.js, Perl, PHP, Python, Ruby, Scala, Go and Erlang.

II. Industry / Employer Expected Outcome(s)

Develop Database using MongoDB which supports different operation like insert update delete.

III. Course Level Learning Outcomes(s)

CO4 - Manipulate data using NoSQL commands

IV. Laboratory Learning Outcome(s)

LLO 8.1 Develop MongoDB Queries using Basic operation.

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Use appropriate DBMS software.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Relevant Theoretical Background

MongoDB is a document database. It stores data in a type of JSON format called BSON. Records in a MongoDB database are called documents, and the field values may include numbers, strings, booleans, arrays, or even nested documents.

Key Differences Between Databases, Collections, and Documents

- **Database:** A container for collections, providing structure and logical isolation for data.
- **Collection:** A group of documents within a database, similar to a table in relational databases.
- **Document:** A single data record within a collection, stored as a BSON object.

Basic Operation with MongoDBShell:-

1. **show dbs:-** Use this command to get list of all databases
2. **use <database>:-** this command is used to create a new database or to select present database as current database.
3. **insert():-** this command is used to insert single document in selected collection.
4. **remove():-** it delete or remove the document from collection.
5. **update():-** it is used to modify existing documents

example:-

```
>use student //create new database
>db.student.insert({rn:01,name:"abc",per:69.25}) //Insert a record
>db.student.update({rn:01},{ $set:{per:80.50}}) // Update record
>db.student.remove({rn:01}) // Delete/remove record
```

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	MySQL/Oracle/SQL server/MongoDB or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

VIII. Conclusion

.....

.....

.....

IX. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State the importance of NoSQL database
2. Compare between SQL and NoSQL database
3. Write syntax of insert, remove command.
4. What is database, collection and documents in mongoDB
5. What is MongoDB shell.

.....

.....

This image shows a full page of primary-ruled paper. It features multiple horizontal rows, each defined by two parallel dotted lines. The rows are evenly spaced across the entire page, providing a guide for handwriting practice. There are no margins, text, or other markings present.

X. Exercise:

1. Create database employee with members id,name and salary.
 - i. Insert 5 record.
 - ii. Update 1 record

- iii. Delete 1 record
- 2. Create database student with members rn,name and per.
 - i. Insert 5 record.
 - ii. Update marks of student 85 where rn = 2
 - iii. Delete record of student whose name="ram"

XI. References:

1. <https://blog.e-zest.com/basic-commands-for-mongodb>
2. <https://www.mongodb.com/docs/manual/reference/command/>
3. <https://www.w3schools.com/mongodb/>

XII. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 9**9.1 *Implements aggregation queries****9.2. Execute query using find() function**

1. Write mongoDB queries using aggregate function for given application
2. Write MongoDB Queries using find() for given application

I. Practical Significance

Aggregation in MongoDB is a powerful framework that allows developers to perform complex data transformations, computations and analysis on collections of documents. By utilizing the aggregation pipeline, users can efficiently group, filter, sort, reshape, and perform calculations on data to generate meaningful insights.

find() method in MongoDB is a tool for retrieving documents from a collection. It supports various query operators and enabling complex queries. It also allows selecting specific fields to optimize data transfer and benefits from automatic indexing for better performance. Industry / Employer Expected Outcome(s).

II. Industry/Employer Expected Program Outcomes (POs)

Manage both structured and unstructured data using various tools for Database.

III. Course Level Learning Outcomes(s)

CO4 - Manipulate data using NoSQL commands

IV. Laboratory Learning Outcome(s)

LLO 9.1 Implement aggregation queries using MongoDB

LLO 9.2. Implement MongoDB queries using find() function.

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Use appropriate DBMS software.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Relevant Theoretical Background

MongoDB Aggregation is a database process that allows us to perform complex data transformations and computations on collections of documents or rows. It enables us to group, filter, and manipulate data to produce summarized results. MongoDB Aggregation is typically carried out using the aggregation pipeline which is a framework for data aggregation modeled on the concept of data processing pipelines.

Example:-

- `db.users.aggregate([{$group: {_id: "$city", totalUsers: {$sum: 1}}}]])`
- `db.user.aggregate ([{$group: {_id: "$dept", per: {$max: "$per"}}}]])`
- `db.user.aggregate ([{$group: {_id: "$dept", per: {$min: "$per"}}}]])`

The find() method is the primary way to retrieve documents from a collection in MongoDB. It provides extensive flexibility through its query operators, projection options, and additional parameters to customize the results. It allows us to specify which fields to include or exclude in the result set and reduce the amount of data transferred.

- db.Collection_name.find(): Specifies the database (db) and the collection (Collection_name) from which to retrieve documents.
- With specific condition:-db.student.find({age:18}) it Find all the students whose age is exactly 18
- Using Nested Documents in Queries :- db.student.find({score:{math: 230, science: 234}})

It find list of the student's who's math score is 230 and science score is 234.

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	MySQL/Oracle/SQL server/MongoDB or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

VIII. Conclusion

.....

.....

.....

IX. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

6. State the advantages of Aggregate function
7. State the role of find()
8. List and state role of pipeline aggregate function
9. Write syntax find()
10. Write syntax of following functions
 - i. \$sum
 - ii. \$avg
 - iii. \$max

.....

.....

.....

This image shows a full page of primary-ruled paper. It features approximately 20 horizontal dotted lines spaced evenly apart, providing a guide for handwriting practice. The lines are black dots on a white background, and there are no margins or other markings present.

X. Exercise:

1. Create database student with members rn,name and per.
 - i. Insert 5 record.
 - ii. Display information of student having name is “ram”
 - iii. Display information of student having per>=75
 - iv. Display all record
 - v. Display information of student having rn=5
2. Create database Employee with members id,name dept and salary.
 - i. Insert 5 record.
 - ii. Display avg salary of each dept
 - iii. Display min salary of each department
 - iv. Display max salary of each department

XI. References:

1. <https://www.geeksforgeeks.org/mongodb-db-collection-find-method/>
2. <https://www.geeksforgeeks.org/aggregation-in-mongodb/>
3. <https://www.mongodb.com/docs/manual/reference/method/db.collection.aggregate/>

XII. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 10 *Use Data warehousing tool(ETL)**10.1. Extract the relevant data from the source database****10.2. Transform the data so that it is better suited for analytics****10.3. Load the into the target database****I. Practical Significance**

The ETL (Extract, Transform, Load) process plays an important role in data warehousing by ensuring seamless integration and preparation of data for analysis. This method involves extracting data from multiple sources, transforming it into a uniform format, and loading it into a centralized data warehouse or data lake. ETL is essential for businesses to consolidate vast amounts of data, enhancing decision-making processes and enabling accurate business insights. In today's digital ecosystem, where data comes from various sources and formats, the ETL process ensures that organizations can efficiently clean, standardize, and organize this data for advanced analytics. It provides a structured foundation for data analytics, improving the quality, security, and accessibility of enterprise data.

II. Industry/Employer Expected Program Outcomes (POs)

Manage both structured and unstructured data using various tools for Database.

III. Course Level Learning Outcomes(s)

CO5- Use data mining and warehousing concepts

IV. Laboratory Learning Outcome(s)

LLO 10.1 Use extract, transform and load (ETL) data warehousing tool

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Use appropriate Data Warehousing tool.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Relevant Theoretical Background

The ETL process, which stands for Extract, Transform, and Load, is a critical methodology used to prepare data for storage, analysis, and reporting in a data warehouse. It involves three distinct stages that help to streamline raw data from multiple sources into a clean, structured, and usable form. Here's a detailed breakdown of each phase:

1. Extraction: The Extract phase is the first step in the ETL process, where raw data is collected from various data sources. These sources can be diverse, ranging from structured sources like databases (SQL, NoSQL), to semi-structured data like JSON, XML, or unstructured data such as emails or flat files. The main goal of extraction is to gather data without altering its format, enabling it to be further processed in the next stage.

Following Steps for Extraction of Data

- i. Step 1: Configure the QueryDatabaseTable
- ii. Step 2: Configure the ConvertAvroToJson

- iii. Step 3: Configure the QueryRecord
 - iv. Step 4: Configure the UpdateAttribute to update the filename
 - v. Step 5: Configure the UpdateAttribute to update file extension
2. Transformation: The Transform phase is where the magic happens. Data extracted in the previous phase is often raw and inconsistent. During transformation, the data is cleaned, aggregated, and formatted according to business rules. Common Transformation include Data filtering, data sorting, data aggregating.
- i. ExecuteSQL Processor: Executes the SQL query to fetch data from the database.
 - ii. ConvertAvroToJson Processor: Converts the extracted Avro-formatted data into JSON.
 - iii. SplitJson Processor: Splits large JSON files into smaller pieces for easier processing.
 - iv. UpdateAttribute Processor: Adds custom attributes to the flow files for better tracking and processing.
3. Loading:- Once data has been cleaned and transformed, it is ready for the final step: Loading. This phase involves transferring the transformed data into a data warehouse, data lake, or another target system for storage. Depending on the use case, there are two types of loading methods:
- Full Load:** All data is loaded into the target system, often used during the initial population of the warehouse.
- Incremental Load:** Only new or updated data is loaded, making this method more efficient for ongoing data updates.
- Final step after transformation:-** PutDatabaseRecord Processor: Inserts the processed data into the target database.
- Different ETL tools are Apache Nifi, Talend, Microsoft SSIS, Hevo, Oracle Warehouse Builder.

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	MySQL/Oracle/SQL server/MongoDB or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

VIII. Conclusion

.....

.....

.....

IX. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List the Challenges in ETL process
2. What are the solutions to overcome ETL challenges.
3. Which factors are help to choose the right ETL tool
4. Compare structure, semi structure and unstructured data.
5. List any 4 open source ETL tools

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. Exercise:

1. Perform Extract, Transform and load operation using any ETL tool.

XI. References:

1. <https://www.projectpro.io/recipes/fetch-data-from-mysql-database-and-perform-query-operation-on-flowfile-nifi>
2. <https://pierrevillard.com/2016/2016-03-09-transform-data-with-apache-nifi/>
3. <https://www.projectpro.io/recipes/fetch-data-from-mysql-database-table-and-store-it-into-postgres-nifi>

XII. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 11 Implement classification techniques in data mining**I. Practical Significance**

Data mining is the process of extracting and finding patterns in massive data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal of extracting information (with intelligent methods) from a data set and transforming the information into a comprehensible structure for further use.

II. Industry/Employer Expected Program Outcomes (POs)

Manage both structured and unstructured data using various tools for Database.

III. Course Level Learning Outcomes(s)

CO5- Use data mining and warehousing concepts

IV. Laboratory Learning Outcome(s)

LLO 11.1 Understand the concept of classification in data mining

V. Relevant Affective Domain related Outcomes

1. Follow ethical practices.
2. Use appropriate Data mining tool.
3. Demonstrate analytical and logical knowledge as an Individual.
4. Participate in team problem solving activities.
5. Prioritizes time effectively to meet the needs of the team and self.

VI. Relevant Theoretical Background

Data mining is the process of extracting knowledge or insights from large amounts of data using various statistical and computational techniques. The data can be structured, semi-structured or unstructured, and can be stored in various forms such as databases, data warehouses, and data lakes.

The primary goal of data mining is to discover hidden patterns and relationships in the data that can be used to make informed decisions or predictions. This involves exploring the data using various techniques such as clustering, classification, regression analysis, association rule mining, and anomaly detection.

Data mining process includes Data gathering, Data staging, Data storage component, Information delivery component, management and control component.

Different open source data mining software are as follows

- Weka - an open-source software for data mining
- RapidMiner - an open-source system for data and text mining
- KNIME - an open-source data integration, processing, analysis, and exploration platform
- The Mahout machine learning library - mining large data sets. It supports recommendation mining, clustering, classification and frequent itemset mining.
- Rattle - a GUI for data mining using R

Steps for classification of data using weka software

- i. Setting data set:- Open the saved file by using the Open file ... option under the Preprocess tab, click on the Classify tab

- ii. Selecting_classifier:-weka→classifiers→trees.
Click on the Start button to start the classification process. After a while, the classification results would be presented on your screen
- iii. Visualize result

VII. Resources required (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity
1.	Any DBMS software	Weka or any suitable database software	01 System for each student
2.	Computer system	Any computer system with basic configuration	

VIII. Conclusion

.....

.....

.....

IX. Practical related Questions

Note: Below given are a few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List & explain different type of data mining
2. List application of data mining.
3. What is data staging?
4. Compare between classification and clustering.
5. List any 4 data mining tools

.....

.....

.....

.....

.....

.....

X. Exercise:

- ## XI. References:

- 55

4. <https://www.youtube.com/watch?v=U63ExiTJMic>

XII. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 30%	Marks-10
1.	Tool Selection Ability:20%	
2.	Follow Ethical Practices:10%	
	Weightage- Product related: 70%	Marks-15
3.	Verifying System Requirement for Installation:20%	
4.	Correctness in Use of Command Tools:15%	
5.	Use of Attribute/s Options:15%	
6.	Timely Submission:10%	
7.	Answering Sample Questions:10%	
	Total 25	
	Dated Signature of Course Teacher	