HKBK College of Engineering

(Affiliated to Visvesvaraya Technological University Belgaum and approved by AICTE, New Delhi and Govt.of Karnataka)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELGAVI-590018, KARNATAKA

LABORATORY MANUAL

DATABASE MANAGEMENT LABORATORY

BCS403

(Effective from the academic year 2024 -2025)

Prepared By:

Prof. SHRUTHI V KULKARNI

Verified By

DQAC	HOD

HKBK COLLEGE OF ENGINEERING

Opp to Manyata Techpark, Nagwara, Bangalore – 560 045

HKBK COLLEGE OF ENGINEERING
Established in 1997

Department of Artificial Intelligence and Machine Learning

Vision and Mission of the Institution

Vision

To empower students through wholesome education and enable the students to develop into highly qualified and trained professionals with ethics and emerge as responsible citizens with a broad outlook to build a vibrant nation.

Mission

M1. To achieve academic excellence in science, engineering, and technology through dedication to duty, innovation in teaching, and faith in human values.

M2. To enable our students to develop into outstanding professionals with high ethical standards to face the challenges of the 21st century.

M3. To provide educational opportunities to the deprived and weaker section of society to uplift their socioeconomic status.

Vision and Mission of the AIML Department

Vision

To advance the intellectual capacity of the nation and the international community by imparting knowledge to graduates who are globally recognized as innovators, entrepreneurs and competent professionals.

Mission

M1. To provide excellent technical knowledge and computing skills to make the graduates globally competitive with professional ethics.

M2. To be involved in research activities and be committed to lifelong learning to positive contributions to society.

Programme Educational Objectives

PEO-1	To provide students with a strong foundation in engineering fundamentals and in the computer science and engineering to work in the global scenario.
PEO-2	To provide sound knowledge of programming and computing techniques and good communication and interpersonal skills so that they will be capable of analyzing, designing and building innovative software systems.
PEO-3	To equip students in the chosen field of engineering and related fields to enable him to work in multidisciplinary teams.
PEO-4	To inculcate in students professional, personal and ethical attitude to relate engineering issues to broader social context and become responsible citizen.
PEO-5	To provide students with an environment for life-long learning which allow them to successfully adapt to the evolving technologies throughout their professional carrier and face the global challenges.

Programme Outcomes	
a	**Engineering Knowledge:** Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
b	**Problem Analysis: Identify,** formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences
c	**Design/ Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal and environmental considerations.
d	**Conduct investigations of complex problems** using research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.
e	**Modern Tool Usage**: Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an under- standing of the limitations.
f	**The Engineer and Society**: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.
g	**Environment and Sustainability**: Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.
h	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

i.	**Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
j.	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.
k	**Life-long Learning**: Recognize the need for and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.
l.	**Project Management and Finance**: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
Programme Specific Outcomes	
m	**Problem-Solving Skills:** An ability to investigate and solve a problem by analysis, interpretation of data, design and implementation through appropriate techniques, tools and skills.
n	**Professional Skills**: An ability to apply algorithmic principles, computing skills and computer science theory in the modelling and design of computer-based systems.
o	**Entrepreneurial Ability:** An ability to apply design, development principles and management skills in the construction of software product of varying complexity to become an entrepreneur

HKBK COLLEGE OF ENGINEERING
Established in 1997

Department of Artificial Intelligence and Machine Learning

Academic Year: 2024-2025 Semester: 4TH EVEN

LAB MANUAL

Subject: **DATABASE MANAGEMENT SYSTEM** Subject Code: **BCS403**

Period: February 2025 To May 2025

Course Learning Objectives:

- To Provide a strong foundation in database concepts, technology, and practice.
- To Practice SQL programming through a variety of database problems.
- To Understand the relational database design principles.
- To Demonstrate the use of concurrency and transactions in database.
- To Design and build database applications for real world problems.
- To become familiar with database storage structures and access techniques.

List of Experiments

Sl No.	Experiments
1.	Create a table called Employee & execute the following. **Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION)** 1. Create a user and grant all permissions to the user. 2. Insert the any three records in the employee table contains attributes EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert null values to the employee table and verify the result.
2.	Create a table called Employee that contain attributes EMPNO, ENAME, JOB, MGR, SAL & execute the following. 1. Add a column commission with domain to the Employee table. 2. Insert any five records into the table. 3. Update the column details of job 4. Rename the column of Employ table using alter command. 5. Delete the employee whose Empno is 105.
3.	Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by, Order by. **Employee (E_id, E_name, Age, Salary)** 1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employee table 3. Find the Maximum age from employee table. 4. Find the Minimum age from employee table. 5. Find salaries of employee in Ascending Order. 6. Find grouped salaries of employees.
4.	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary. **CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)**
5.	Create cursor for Employee table & extract the values from the table. Declare the variables, Open the cursor & extract the values from the cursor. Close the cursor. **Employee(E_id, E_name, Age, Salary)**
6.	Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.
7.	Install an Open Source NoSQL Data base MangoDB & perform basic CRUD (Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.

A. Create a table called Employee & execute the following.
Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION)
1. Create a user and grant all permissions to the user.
2. Insert the any three records in the employee table contains attributes EMPNO,ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result.
3. Add primary key constraint and not null constraint to the employee table.
4. Insert null values to the employee table and verify the result.

Query Solution:

1. CREATE DATABASE EXP1;

   USE EXP1;

   create user 'user1' identified by 'psd' password expire;

   grant ‘user1’ to ‘root’@’localhost’;

   grant all on EXP1.* to ‘user1’;

   CREATE TABLE Employee (EMPNO INT, ENAME VARCHAR(100), JOB VARCHAR(100), MANAGER_NO INT, SAL DECIMAL(10, 2), COMMISSION DECIMAL(10, 2));

   DESC Employee;

2. INSERT INTO Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION) VALUES (101, 'John Doe', 'Manager', 100, 50000, 5000);

   INSERT INTO Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION) VALUES (102, 'Jane Smith', 'Developer', 101, 60000, 6000);

   INSERT INTO Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION) VALUES (103, 'Jim Brown', 'Analyst', 101, 55000, 5500);

   SELECT * FROM Employee;

3. ALTER TABLE Employee ADD PRIMARY KEY (EMPNO);

   //Adding NOT NULL constraints

   ALTER TABLE Employee MODIFY ENAME VARCHAR(100) NOT NULL;

   ALTER TABLE Employee MODIFY JOB VARCHAR(100) NOT NULL;

   ALTER TABLE Employee MODIFY SAL DECIMAL(10, 2) NOT NULL;

   DESC Employee;

4. INSERT INTO Employee (EMPNO, ENAME, JOB, SAL) VALUES (104, NULL, 'Tester', NULL);
   INSERT INTO Employee (EMPNO, ENAME, JOB, SAL) VALUES (104, 'James', NULL ,60000);

Output:

1.
Query OK, 1 row affected
Default schema set to ` EXP1`.
Query OK, 1 row affected
Query OK, 1 row affected
Query OK, 1 row affected
Query OK, 1 row affected

Field	Type	Null	Key	Default	Extra
EMPNO	int	YES		NULL	
ENAME	varchar(100)	YES		NULL	
JOB	varchar(100)	YES		NULL	
MANAGER_NO	int	YES		NULL	
SAL	decimal(10,2)	YES		NULL	
COMMISSION	decimal(10,2)	YES		NULL	

6 rows in set (0.0026 sec)

2.

Query OK, 1 row affected
Query OK, 1 row affected
Query OK, 1 row affected

EMPNO	ENAME	JOB	MANAGER_NO	SAL	COMMISSION
101	John Doe	Manager	100	50000.00	5000.00
102	Jane Smith	Developer	101	60000.00	6000.00
103	Jim Brown	Analyst	102	55000.00	5500.00

3 rows in set (0.0008 sec)

3.

Query OK, 1 row affected
Query OK, 1 row affected
Query OK, 1 row affected
Query OK, 1 row affected

Field	Type	Null	Key	Default	Extra
EMPNO	int	NO	PRI	NULL	
ENAME	varchar(100)	NO		NULL	
JOB	varchar(100)	NO		NULL	
MANAGER_NO	int	YES		NULL	
SAL	decimal(10,2)	NO		NULL	
COMMISSION	decimal(10,2)	YES		NULL	

6 rows in set (0.0022 sec)

4.

Column 'ENAME' cannot be null
Column 'SAL' cannot be null
Column 'JOB' cannot be null

B. Create a table called Employee that contain attributes **EMPNO,ENAME,JOB, MGR,SAL** & execute the following.
1. Add a column commission with domain to the Employee table.
2. Insert any five records into the table.
3. Update the column details of job
4. Rename the column of Employ table using alter command.
5. Delete the employee whose Empno is 105.

Solution:

1. CREATE TABLE Employee (EMPNO INT, ENAME VARCHAR(100),JOB VARCHAR(100),MGR INT,SAL DECIMAL(10, 2));

   ALTER TABLE Employee ADD COMMISSION DECIMAL(10, 2) CHECK (COMMISSION >= 0 OR COMMISSION IS NULL);

2. INSERT INTO Employee (EMPNO, ENAME, JOB, MGR, SAL, COMMISSION) VALUES
   (101, 'John Doe', 'Manager', 100, 75000, 5000),
   (102, 'Jane Smith', 'Developer', 101, 60000, 3000),
   (103, 'Jim Beam', 'Analyst', 102, 55000, 2500),
   (104, 'Jill Hill', 'Salesperson', 103, 45000, 2000),
   (105, 'Jack Black', 'Support', 101, 40000, 1500);

3. UPDATE Employee SET JOB = 'Senior Salesperson' WHERE EMPNO = 104;

4. ALTER TABLE Employee RENAME COLUMN MGR TO MANAGER_ID;

5. DELETE FROM Employee WHERE EMPNO = 105;
   SELECT * FROM Employee;

C. Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by, Order by.

Employee(E_id, E_name, Age, Salary)

1. Create Employee table containing all Records E_id, E_name, Age, Salary.
2. Count number of employee names from employee table
3. Find the Maximum age from employee table.
4. Find the Minimum age from employee table.
5. Find salaries of employee in Ascending Order.
6. Find grouped salaries of employees.

Solution:

1. CREATE TABLE Employee (E_id INT PRIMARY KEY,E_name VARCHAR(100),Age INT,Salary DECIMAL(10, 2));

2. SELECT COUNT(E_name) FROM Employee;
   SELECT COUNT(DISTINCT E_name) FROM Employee;

3. SELECT MAX(Age) FROM Employee;

4. SELECT MIN(Age) FROM Employee;

5. SELECT Salary FROM Employee ORDER BY Salary ASC;

6. SELECT Salary, COUNT(*) AS Number_of_Employees FROM Employee GROUP BY Salary ORDER BY Salary;

D. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary.
CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)

Solution:

SQL>create table customers (id number(10), name varchar2(30), age number(10), address varchar2(50), salary number(10));

1. insert into customers (id, name, age, address, salary) values(101,'Kumar',45, 'Bengalure',20000);
2. insert into customers (id, name, age, address, salary) values(102,'Rohan',42, 'Tumkuru', 10000);
3. insert into customers (id, name, age, address, salary) values(103,'Kumar',38, 'Belagavi', 9000);
4. insert into customers (id, name, age, address, salary) values(104,'Chandu',35, 'Tiptur ', 9000);

SQL> select * from employee;

Results Explain Describe Saved SQL History

ID	NAME	AGE	ADDRESS	SALARY
101	Kumar	45	Bengalure	20000
102	Rohan	42	Tumkuru	10000
103	Kumar	38	Belagavi	9000
104	Chandu	35	Tiptur	9000

4 rows returned in 0.00 seconds CSV Export

SQL>

```
CREATE OR REPLACE TRIGGER display_salary_changes BEFORE DELETE OR INSERT OR UPDATE
    ON customers FOR EACH ROW WHEN (NEW.ID > 0)
DECLARE sal_diff number;
BEGIN
    sal_diff := :NEW.salary - :OLD.salary;
    dbms_output.put_line('Old salary: ' || :OLD.salary);
    dbms_output.put_line('New salary: ' || :NEW.salary);
    dbms_output.put_line('Salary difference: ' || sal_diff);
END;
```

Results Explain Describe Saved SQL History

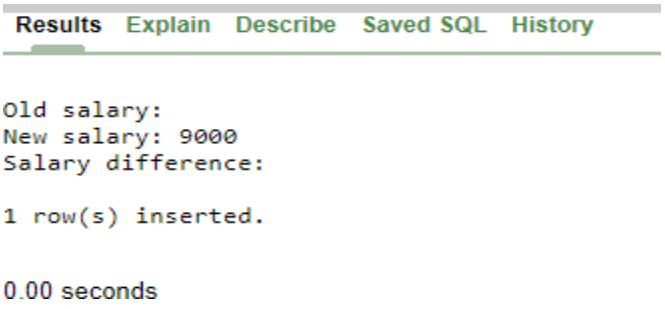
```
Trigger created.

0.12 seconds
```

To check whether trigger works during INSERT statement.

SQL>

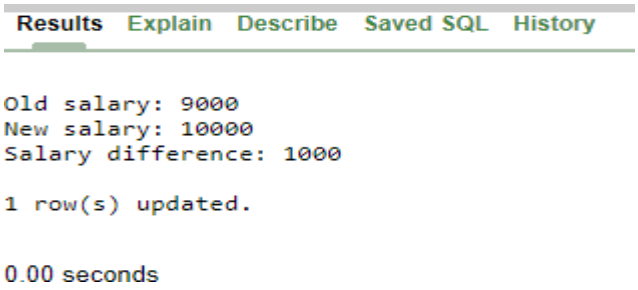
insert into customers (id, name, age, address, salary) values (105,'Keerthi',36, 'Mangaluru',9000);



To check whether trigger works during UPDATE statement.

SQL>

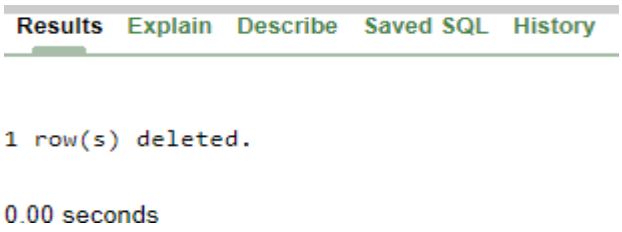
update customers SET salary=10000 where id=105;



To check whether trigger works during DELETE statement.

SQL>

delete from customers where id=105;



E. Create cursor for Employee table & extract the values from the table. Declare the variables, Open the cursor & extract the values from the cursor. Close the cursor.
Employee(E_id, E_name, Age, Salary)

Solution:

SQL> create table employee (e_id number(10), e_name varchar2(30), age number(10), salary number(10));

Insert atleast five rows into the employee table.

1. insert into employee (e_id, e_name, age, salary) values(101,'Kumar',45,20000);
2. insert into employee (e_id, e_name, age, salary) values(102,'Rohan',42, 10000);
3. insert into employee (e_id, e_name, age, salary) values(103,'Kumar',38, 9000);
4. insert into employee (e_id, e_name, age, salary) values(104,'Chandu',35, 9000);
5. insert into employee (e_id, e_name, age, salary) values(105,'Keerthi',36, 9000);

Check the result of insertion of rows in the employee table.

SQL> select * from employee;

Results Explain Describe Saved SQL History

E_ID	E_NAME	AGE	SALARY
101	Kumar	45	20000
102	Rohan	42	10000
103	Kumar	38	9000
104	Chandu	35	9000
105	Keerthi	36	9000

5 rows returned in 0.01 seconds CSV Export

SQL>

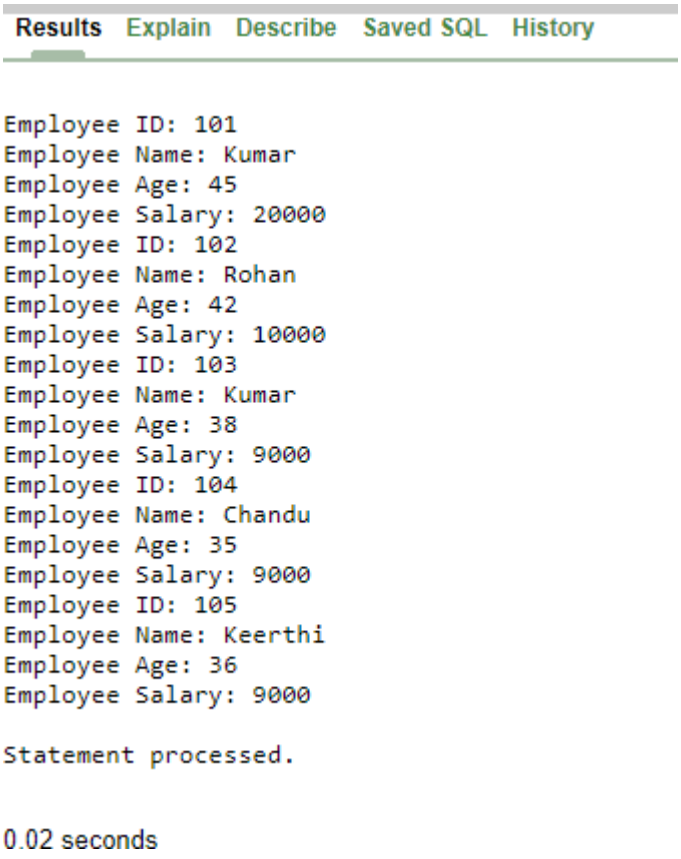
```
DECLARE   CURSOR emp_cursor IS

SELECT E_id, E_name, Age, Salary FROM Employee;

  v_e_id NUMBER;
  v_e_name VARCHAR2(20);
  v_age NUMBER;
  v_salary NUMBER;

BEGIN
  OPEN emp_cursor;
  LOOP
    FETCH emp_cursor INTO v_e_id, v_e_name, v_age, v_salary;
```

```
    EXIT WHEN emp_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || v_e_id); DBMS_OUTPUT.PUT_LINE('Employee Name: ' ||
    v_e_name); DBMS_OUTPUT.PUT_LINE('Employee Age: ' || v_age);
    DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || v_salary); END LOOP;
    CLOSE emp_cursor;
END;
```



F. Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.

Solution:

```
DECLARE
   -- Declare variables to store values from the cursor
   n_id N_RollCall.id%TYPE;
   n_name N_RollCall.name%TYPE;
   n_date N_RollCall.date%TYPE;

   -- Declare a cursor with parameters to fetch data from N_RollCall
   CURSOR merge_cursor IS
      SELECT id, name, date FROM N_RollCall;

BEGIN
   -- Open the cursor
   OPEN merge_cursor;

   -- Loop through the cursor records
   FOR merge_rec IN merge_cursor LOOP
      -- Check if the record already exists in O_RollCall
      SELECT id INTO n_id
      FROM O_RollCall
      WHERE id = merge_rec.id;

      -- If the record doesn't exist, insert it into O_RollCall
      IF n_id IS NULL THEN
         INSERT INTO O_RollCall (id, name, date)
         VALUES (merge_rec.id, merge_rec.name, merge_rec.date);
         DBMS_OUTPUT.PUT_LINE('Data merged for ID: ' || merge_rec.id);
      ELSE
         DBMS_OUTPUT.PUT_LINE('Data already exists for ID: ' || merge_rec.id || ', skipping...');
      END IF;
   END LOOP;

   -- Close the cursor
   CLOSE merge_cursor;

   -- Commit the transaction
   COMMIT;
EXCEPTION
   WHEN OTHERS THEN
      -- Handle exceptions if any
      DBMS_OUTPUT.PUT_LINE ('An error occurred: ' || SQLERRM);
      ROLLBACK;
END;
/
```

OR

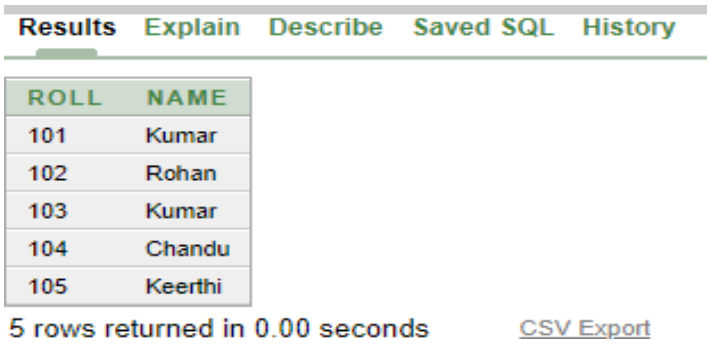
SQL> create table N_RollCall (roll number(10), name varchar2(30));

Insert atleast five rows into the N_RollCall table.

1. insert into N_RollCall (roll,name) values(101,'Kumar');

2. insert into N_RollCall (roll,name)values(102,'Rohan');
3. insert into N_RollCall (roll,name) values(103,'Kumar');
4. insert into N_RollCall (roll,name) values(104,'Chandu');
5. insert into N_RollCall (roll,name) values(105,'Keerthi');

SQL> select * from N_RollCall;



Results | Explain | Describe | Saved SQL | History

ROLL	NAME
101	Kumar
102	Rohan
103	Kumar
104	Chandu
105	Keerthi

5 rows returned in 0.00 seconds CSV Export

Create a table called O_RollCall & execute the following.

O_RollCall (ROLL,NAME)

SQL> create table O_RollCall (roll number(10), name varchar2(30));

Insert atleast five rows into the O_RollCall table.

1. insert into O_RollCall (roll,name) values(201,'Abhi');
2. insert into O_RollCall (roll,name)values(202,'Rohit');
3. insert into O_RollCall (roll,name) values(203,'Kumar');
4. insert into O_RollCall (roll,name) values(204,'Chandu');
5. insert into O_RollCall (roll,name) values(505,'Keerthi');

SQL> select * from O_RollCall;

Results Explain Describe Saved SQL History

ROLL	NAME
201	Abhi
202	Rohit
203	Kumar
204	Chandu
505	Keerthi

5 rows returned in 0.00 seconds CSV Export

SQL>

```
DECLARE
  CURSOR c1 IS SELECT * FROM O_RollCall;
   roll_no number; counter number(10);
BEGIN
  counter:=0;
  FOR c1_rec IN c1 LOOP
          select count(*) into roll_no from N_RollCall where roll=c1_rec.roll;
          if roll_no=0 then
        INSERT INTO N_RollCall VALUES (c1_rec.roll, c1_rec.name);
        counter:=counter+1;
        end if;
   END LOOP;
   COMMIT;
    DBMS_OUTPUT.PUT_LINE ('Number of Rows Inserted:'||counter);
  END;
```

Results Explain Describe Saved SQL History

```
Number of Rows Inserted:6

Statement processed.

0.00 seconds
```

G. Install an Open Source NoSQL Data base MongoDB & perform basic CRUD(Create, Read, Update & Delete) operations. Execute MongoDB basic Queries using CRUD operations.

Solution:

Procedure to install MongoDB:

- Visit MongoDB official website: https://www.mongodb.com/try/download/community
- Click the download button for the Windows
- Run the installer: Once downloaded, run the MSI file and follow the installation wizard. Choose the "Complete" installation.
- Download MongoDB Shell from official website: https://www.mongodb.com/try/download/shell
- Click the download button for the Windows
- Run the installer: Once downloaded, double click the zip folder and double click on mongosh.exe application
- To start MongoDB server and connect to MongoDB: Enter the MongoDB connection string in MongoDB shell command prompt
  mongo

MongoDB basic Queries using CRUD operations:

- To create Database
  mongo> use myNewDb
- create / insert operation
  myNewDb> db.myCollection.insertOne({name:"yourName",age:"27",profession:"Engineer" })
  myNewDb> db.myCollection.insertMany([{name: "Jane", age: 25, profession: "Designer"},
  {name: "Bob", age: 35, profession: "Manager"}])
- read query operations
  myNewDb>db.myCollection.findOne({name: "John"})
  myNewDb>db.myCollection.find({})
- update operations
  myNewDb>db.myCollection.updateOne({name: "John"}, {$set: {age: 31}})
  myNewDb>db.myCollection.updateMany({profession: "Engineer"}, {$set: {status: "Active"}})
- delete operations
  db.myCollection.deleteOne({name: "John"})
  db.myCollection.deleteMany({profession: "Engineer"})

OR

MongoDB is an open-source document-oriented database. It is categorized under the NoSQL(Not only SQL) database because the storage and retrieval of data in MongoDB are not in the form of tables.

Requirements to Install MongoDB on Windows:
MongoDB 4.4 and later only support 64-bit versions of Windows.
MongoDB 7.0 Community Edition supports the following 64-bit versions of Windows on x86_64 architectures:

Windows Server 2022
Windows Server 2019
Windows 11

Ensure that the user is running mongod and mongos has the necessary permissions from the following groups:

Performance Monitor Users

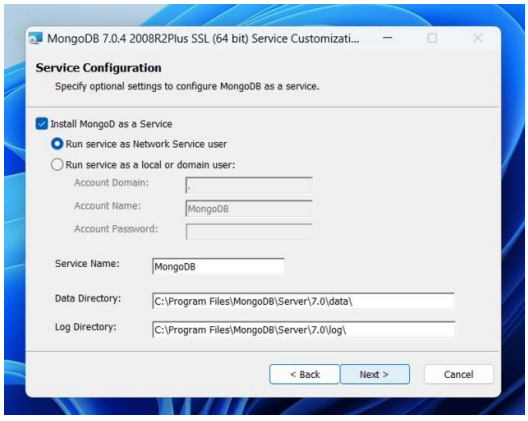
Performance Log Users

Install MangoDB:

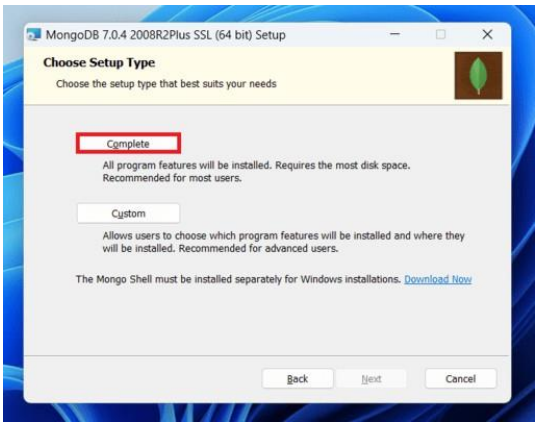
Step1: To install MongoDB on windows, first, download the MongoDB server and then install the MongoDB shell.

Step 2: When the download is complete open the msi file and click the next button in the startup screen:

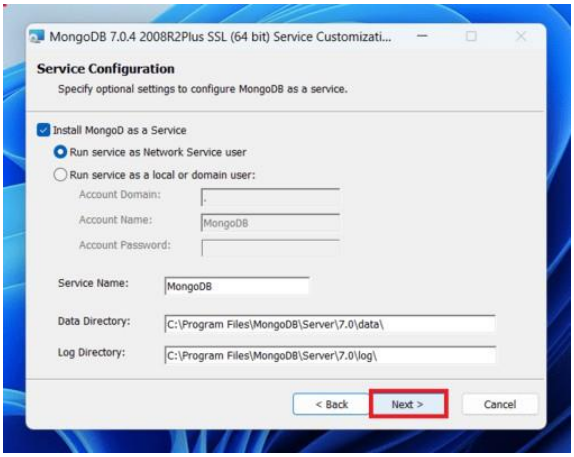
Step 3: Now accept the End-User License Agreement and click the next button:



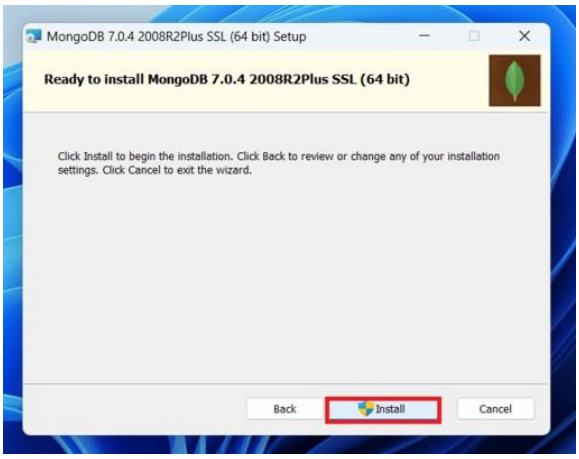
Step 4: Now select the complete option to install all the program features



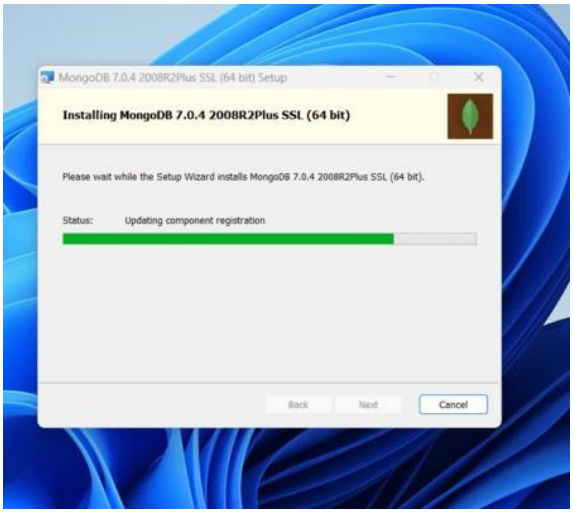
Step 5: Select “Run service as Network Service user” and copy the path of the data directory. Click Next:



Step 6: Click the Install button to start the MongoDB installation process:

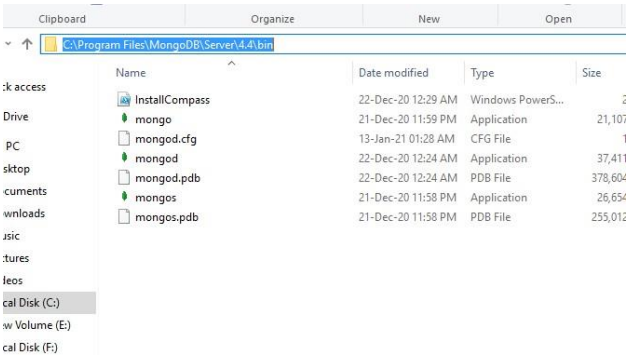


Step 7: After clicking on the install button installation of MongoDB begins:

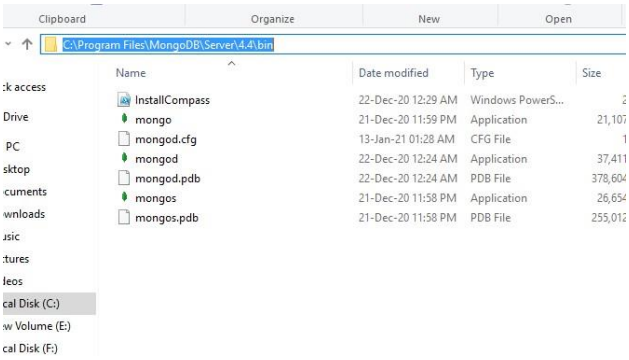


Step 8: Now click the Finish button to complete the MongoDB installation process:

Step 9: Now we go to the location where MongoDB installed in step 5 in your system and copy the bin path:



Step 10: Now, to create an environment variable open system properties << Environment Variable << System variable << path << Edit Environment variable and paste the copied link to your environment system and click Ok:



Step 11: After setting the environment variable, we will run the MongoDB server, i.e. mongod. So, open the command prompt and run the following command:

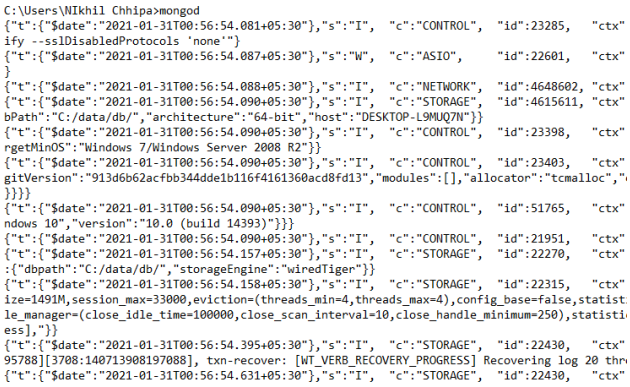
mongod

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

Step 12: Now, Open C drive and create a folder named “data” inside this folder create another folder named “db”. 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.



Run mongo Shell

Step 13: Now we are going to connect our server (mongod) with the mongo shell. So, keep that mongod window and open a new command prompt window and write mongo. Now, our mongo shell will successfully connect to the mongod.

Install an Open Source NoSQL Data base MangoDB.

SOLUTION:

SQL>

perform basic CRUD (Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.

CREATE OPERATION

SOLUTION:

Create a New Database Using

Mongo Shell use college

Show List of Databases show dbs

Check Current Database

db

Create Operations

Method	Description
db.collection.insertOne()	It is used to insert a single document in the collection.
db.collection.insertMany()	It is used to insert multiple documents in the collection.
db.createCollection()	It is used to create an empty collection.

Exercise1:

Insert Single Document db.student.insert({Name: "Akshay", Marks: 500}) db.student.find().pretty()

db.student.insertOne({Name: "Akshay", Marks: 500})

db.student.insertOne({_id: "Stu102", Name: "Vishal", Marks: 230})

db.student.insertMany([{name:"Ajay",age:20},

{name:"Bina",age:24},

{name:"Ram",age:23}]) db.student.insertMany([{_id:"stu200", name:"Ammu", age:18},

{_id:"stu201", name:"Priya", age:29}])

In MongoDB, the Bulk.insert() method is used to perform insert operations in bulk. var bulk = db.students.initializeUnorderedBulkOp();

bulk.insert({ first_name: "Sachin", last_name: "Tendulkar" });

bulk.insert({ first_name: "Virender", last_name: "Sehwag" });

bulk.insert({ first_name: "Shikhar", last_name: "Dhawan" });

bulk.insert({ first_name: "Mohammed", last_name: "Shami" }); bulk.insert({ first_name: "Shreyas", last_name: "Iyer" });

bulk.execute();

READ OPERATION

SOLUTION:

db.student.find({age:18})

db.student.find()

db.student.find({score:{math: 230, science: 234}})

db.student.findOne({language:"c++"})

db.student.findOne({name: "Avinash"}, {_id: 0, language:1})

UPDATE OPERATION

SOLUTION:

db.student.update({name:"avi"},{$set:{name:"helloword"}})

db.student.update({name:"prachi"},{$set:{age:20}}

db.student.updateOne({name: "Annu"}, {$set:{age:25}})

db.student.updateOne({name:"Bhannu"},{$set:{name:"Babita"}})

DELETE OPERATION

SOLUTION:

db.student.remove({name: "Akshay"})

db.student.remove({})

db.student.remove({age:{$eq:18}}, true) db.student.deleteOne({age:17})

db.student.deleteOne({age:{$lt:18}})