

Lab Manual

Database Management Laboratory

Semester IV

Course Code: BCS403

Teaching Hours/Week - 2

CIE Marks 50

SEE Marks 50

Credits 04

Exam Hours:

(Applicable for 4th Sem ISE)

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



Course Objectives (Defined by the university):

- CLO 1. To Provide a strong foundation in database concepts, technology, and practice.
- CLO 2. To Practice SQL programming through a variety of database problems.
- CLO 3. To Understand the relational database design principles
- CLO 4. To Demonstrate the use of concurrency and transactions in database.
- CLO 5. To Design and build database applications for real world problems.
- CLO 6. To become familiar with database storage structures and access techniques.

Note: A two-hour tutorial is suggested for each laboratory session.

Pedagogy: For the above experiments the following pedagogy can be considered. Problem-based learning, Active learning, MOOC, Chalk & Talk

PART A – List of problems for which students should develop programs and execute in the Laboratory.

Course outcomes (Course Skill Set):

At the end of the course, the student will be able to:

- CO 1. Describe the basic elements of a relational database management system
- CO 2. Design entity relationship for the given scenario.
- CO 3. Apply various Structured Query Language (SQL) statements for database manipulation.
- CO 4. Analyse various normalization forms for the given application.
- CO 5. Develop database applications for the given real world problem
- CO 6. Understand the concepts related to NoSQL databases.

List of Problems/Experiments

Experiments	
	List of problems for which students should develop the program and execute it in the laboratory
1	<p>Create a table called Employee & execute the following. Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION)</p> <ol style="list-style-type: none"> 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	<p>Create a table called Employee that contain attributes EMPNO, ENAME, JOB, MGR,SAL & execute the following.</p> <ol style="list-style-type: none"> 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 Emp no is 105.
3	<p>Queries using aggregate functions (COUNT, AVG, MIN, MAX,SUM),Group by, Orderby. Employee(E_id, E_name, Age, Salary)</p> <ol style="list-style-type: none"> 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	<p>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</p>

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



	<p>salary difference between the old & new Salary. CUSTOMERS(ID , NAME , AGE , ADDRESS, SALARY)</p>
5	<p>Create cursor for Employee table & extract the values from the table. Declare the variables ,Open the cursor & extrect the values from the cursor. Close the cursor. Employee(E_id, E_name, Age, Salary)</p>
6	<p>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.</p>
7	<p>Install an Open Source NoSQL Data base MangoDB & perform basic CRUD(Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.</p>

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



Experiment 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.

Code:

1. CREATE USER 'sharma'@'localhost' IDENTIFIED BY 'tiger';

```
Database changed
MariaDB [arpit]> CREATE USER 'sharma'@'localhost' IDENTIFIED BY 'tiger';
Query OK, 0 rows affected (0.004 sec)

C:\xampp\mysql\bin>mysql -u sharma -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
GRANT ALL PRIVILEGES ON *.* TO 'sharma'@'localhost';

MariaDB [arpit]> GRANT ALL PRIVILEGES ON *.* TO 'sharma'@'localhost';
Query OK, 0 rows affected (0.005 sec)
```

Creation of Table:

```
create table employee(empno int, ename varchar(10),job varchar(10),manager_no int,sal int,
commission int);
```

2. insert into employee(ename,job,manager_no,sal,commission) values ("Ram","Developer",123,6000,20);
insert into employee(ename,job,manager_no,sal,commission) values ("John","Tester",456,7000,10);

CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme

BCS403 - Database Management System Laboratory

insert into employee(ename,job,manager_no,sal,commission) values
("Mahesh","Consultant",789,8000,15);

Rollback;

```
MariaDB [arpit]> Rollback;
Query OK, 0 rows affected (0.000 sec)

MariaDB [arpit]> select * from Employee;
+-----+-----+-----+-----+-----+-----+
| empno | ename | job      | manager_no | sal | commission |
+-----+-----+-----+-----+-----+-----+
| NULL  | Ram   | Developer | 123        | 6000 | 20          |
| NULL  | John  | Tester   | 456        | 7000 | 10          |
| NULL  | Mahesh | Consultant | 789        | 8000 | 15          |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [arpit]> select * from Employee where ename="Ram";
+-----+-----+-----+-----+-----+-----+
| empno | ename | job      | manager_no | sal | commission |
+-----+-----+-----+-----+-----+-----+
| NULL  | Ram   | Developer | 123        | 6000 | 20          |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [arpit]> SELECT COUNT(*) FROM Employee;
+-----+
| COUNT(*) |
+-----+
| 3        |
+-----+
```

3. alter table employee add primary key(ename);

```
MariaDB [arpit]> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno      | int(11)       | NO   |     | NULL    |       |
| ename      | varchar(10)   | NO   | PRI | NULL    |       |
| job        | varchar(10)   | YES  |     | NULL    |       |
| manager_no | int(11)       | YES  |     | NULL    |       |
| sal        | int(11)       | YES  |     | NULL    |       |
| commission | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.014 sec)
```

alter table employee modify empno int not null;

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```
Query OK, 0 rows affected, 3 warnings (0.064 sec)
Records: 0 Duplicates: 0 Warnings: 3

MariaDB [arpit]> select * from Employee;
+-----+-----+-----+-----+-----+-----+
| empno | ename  | job          | manager_no | sal   | commission |
+-----+-----+-----+-----+-----+-----+
| 0     | John   | Tester       | 456         | 7000  | 10          |
| 0     | Mahesh | Consultant    | 789         | 8000  | 15          |
| 0     | Ram    | Developer    | 123         | 6000  | 20          |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)
```

4. insert into employee values(2,"Gokul",null,421,9800,34);

```
MariaDB [arpit]> select * from Employee;
+-----+-----+-----+-----+-----+-----+
| empno | ename  | job          | manager_no | sal   | commission |
+-----+-----+-----+-----+-----+-----+
| 2     | Gokul  | NULL         | 421         | 9800  | 34          |
| 0     | John   | Tester       | 456         | 7000  | 10          |
| 0     | Mahesh | Consultant    | 789         | 8000  | 15          |
| 0     | Ram    | Developer    | 123         | 6000  | 20          |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)
```

Experiment 2: Create a table called Employee that contain attributes EMPNO, ENAME, JOB,

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



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 Emp no is 105.

Code:

Creation of Table: create table Emp(empno int, name varchar(30),job varchar(30),mgr int, sal int);

```
MariaDB [arpit]> create table Emp(empno int, name varchar(30),job varchar(30),mgr int, sal int);
Query OK, 0 rows affected (0.019 sec)
```

1. ALTER TABLE Emp ADD COLUMN COMMISSION DECIMAL(10, 2);

```
MariaDB [arpit]> ALTER TABLE Emp ADD COLUMN COMMISSION DECIMAL(10, 2);
Query OK, 0 rows affected (0.022 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

2. insert into Emp values (101, 'John Doe', 'Manager', 100, 5000, 1000),(102, 'Jane Smith', 'Developer', 101, 4000, 500),(103, 'Alice Johnson', 'Analyst', 102, 3500, NULL),(104, 'Bob Brown', 'Assistant', 103, 3000, 200),(105, 'Emily Davis', 'Clerk', 102, 2500, 300);

```
MariaDB [arpit]> insert into Emp values (101, 'John Doe', 'Manager', 100, 5000, 1000.00),
(102, 'Jane Smith', 'Developer', 101, 4000, 500.00),
(103, 'Alice Johnson', 'Analyst', 102, 3500, NULL),
(104, 'Bob Brown', 'Assistant', 103, 3000, 200.00),
(105, 'Emily Davis', 'Clerk', 102, 2500, 300.00);
Query OK, 5 rows affected (0.021 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
MariaDB [arpit]> select * from Emp;
```

empno	name	job	mgr	sal	COMMISSION
101	John Doe	Manager	100	5000	1000.00
102	Jane Smith	Developer	101	4000	500.00
103	Alice Johnson	Analyst	102	3500	NULL
104	Bob Brown	Assistant	103	3000	200.00
105	Emily Davis	Clerk	102	2500	300.00

```
5 rows in set (0.000 sec)
```

3. UPDATE Emp SET job = 'Senior Developer' WHERE empno = 102;

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```

MariaDB [arpit]> UPDATE Emp SET job = 'Senior Developer' WHERE empno = 102;
Query OK, 1 row affected (0.015 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [arpit]> select * from Emp;
+-----+-----+-----+-----+-----+-----+
| empno | name       | job           | mgr  | sal  | COMMISSION |
+-----+-----+-----+-----+-----+-----+
| 101   | John Doe   | Manager       | 100  | 5000 | 1000.00    |
| 102   | Jane Smith | Senior Developer | 101  | 4000 | 500.00     |
| 103   | Alice Johnson | Analyst       | 102  | 3500 | NULL       |
| 104   | Bob Brown  | Assistant     | 103  | 3000 | 200.00     |
| 105   | Emily Davis | Clerk         | 102  | 2500 | 300.00     |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)

```

4. ALTER TABLE Emp change mgr supervisor_no int;

```

MariaDB [arpit]> ALTER TABLE Emp change mgr supervisor_no int;
Query OK, 0 rows affected (0.013 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [arpit]> select * from Emp;
+-----+-----+-----+-----+-----+-----+
| empno | name       | job           | supervisor_no | sal  | COMMISSION |
+-----+-----+-----+-----+-----+-----+
| 101   | John Doe   | Manager       | 100          | 5000 | 1000.00    |
| 102   | Jane Smith | Senior Developer | 101          | 4000 | 500.00     |
| 103   | Alice Johnson | Analyst       | 102          | 3500 | NULL       |
| 104   | Bob Brown  | Assistant     | 103          | 3000 | 200.00     |
| 105   | Emily Davis | Clerk         | 102          | 2500 | 300.00     |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

```

5. DELETE FROM Emp WHERE empno = 105;

```

MariaDB [arpit]> DELETE FROM Emp WHERE empno = 105;
Query OK, 1 row affected (0.010 sec)

MariaDB [arpit]> select * from Emp;
+-----+-----+-----+-----+-----+-----+
| empno | name       | job           | supervisor_no | sal  | COMMISSION |
+-----+-----+-----+-----+-----+-----+
| 101   | John Doe   | Manager       | 100          | 5000 | 1000.00    |
| 102   | Jane Smith | Senior Developer | 101          | 4000 | 500.00     |
| 103   | Alice Johnson | Analyst       | 102          | 3500 | NULL       |
| 104   | Bob Brown  | Assistant     | 103          | 3000 | 200.00     |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

```

Experiment 3: Queries using aggregate functions (COUNT, AVG, MIN, MAX,SUM),Group by, Orderby.

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.

Code:

1. CREATE TABLE Employee1(E_id INT,E_name VARCHAR(50),Age INT, Salary DECIMAL(10, 2));

```
MariaDB [arpit]> CREATE TABLE Employee1(
->   E_id INT,
->   E_name VARCHAR(50),
->   Age INT,
->   Salary DECIMAL(10, 2)
-> );
Query OK, 0 rows affected (0.021 sec)
```

```
INSERT INTO Employee VALUES (1, 'John Doe', 30, 50000),(2, 'Jane Smith', 25,
45000),(3, 'Alice Johnson', 35, 60000),(4, 'Bob Brown', 28, 48000),(5, 'Emily Davis', 32,
55000);
```

```
MariaDB [arpit]> INSERT INTO Employee1 VALUES (1, 'John Doe', 30
', 35, 60000),(4, 'Bob Brown', 28, 48000),(5, 'Emily Davis', 32,
Query OK, 5 rows affected (0.006 sec)
Records: 5 Duplicates: 0 Warnings: 0

MariaDB [arpit]> select * from Employee1;
+-----+-----+-----+-----+
| E_id | E_name      | Age | Salary |
+-----+-----+-----+-----+
| 1 | John Doe    | 30 | 50000.00 |
| 2 | Jane Smith  | 25 | 45000.00 |
| 3 | Alice Johnson | 35 | 60000.00 |
| 4 | Bob Brown   | 28 | 48000.00 |
| 5 | Emily Davis | 32 | 55000.00 |
+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```

2. SELECT COUNT(E_name) AS TotalEmployees FROM Employee1;

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```
MariaDB [arpit]> SELECT COUNT(E_name) AS TotalEmployees FROM Employee1;
+-----+
| TotalEmployees |
+-----+
|           5 |
+-----+
1 row in set (0.000 sec)
```

3. SELECT MAX(Age) AS MaxAge FROM Employee1;

```
+-----+
| MaxAge |
+-----+
|      35 |
+-----+
1 row in set (0.000 sec)
```

4. SELECT MIN(Age) AS MinAge FROM Employee1;

```
+-----+
| MinAge |
+-----+
|      25 |
+-----+
1 row in set (0.001 sec)
```

5. SELECT E_name, Salary FROM Employee1 ORDER BY Salary ASC;

```
+-----+-----+
| E_name      | Salary |
+-----+-----+
| Jane Smith  | 45000.00 |
| Bob Brown   | 48000.00 |
| John Doe    | 50000.00 |
| Emily Davis | 55000.00 |
| Alice Johnson | 60000.00 |
+-----+-----+
5 rows in set (0.001 sec)
```

6. SELECT Salary, COUNT(*) AS EmployeeCount FROM Employee1 GROUP BY Salary;

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```
+-----+-----+
| Salary | EmployeeCount |
+-----+-----+
| 45000.00 | 1 |
| 48000.00 | 1 |
| 50000.00 | 1 |
| 55000.00 | 1 |
| 60000.00 | 1 |
+-----+-----+
5 rows in set (0.001 sec)
```

Experiment 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)

Code:

Creation of Table:

```
CREATE TABLE customers (ID INT,NAME VARCHAR(50),AGE INT,ADDRESS  
VARCHAR(100),SALARY INT);
```

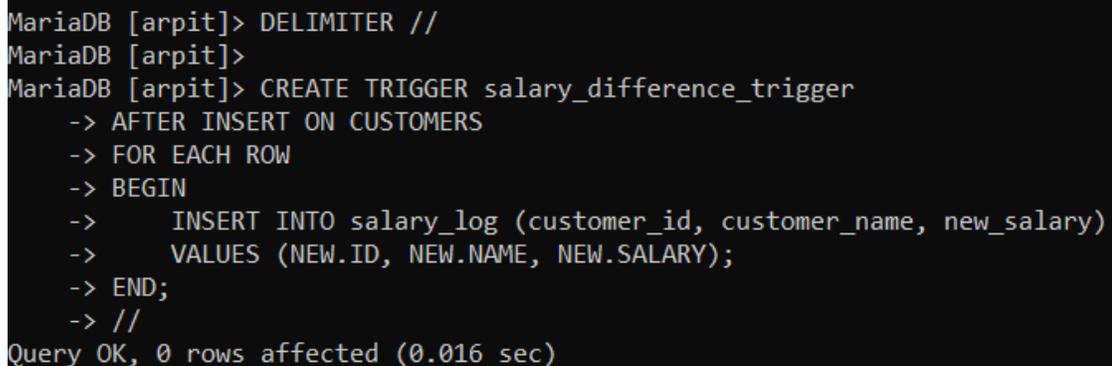
```
CREATE TABLE salary_log (customer_id INT,customer_name  
VARCHAR(50),new_salary INT);
```

Trigger

```
DELIMITER //
```

```
CREATE TRIGGER salary_difference_trigger  
AFTER INSERT ON CUSTOMERS  
FOR EACH ROW  
BEGIN  
    INSERT INTO salary_log (customer_id, customer_name, new_salary)  
    VALUES (NEW.ID, NEW.NAME, NEW.SALARY);  
END;  
//
```

```
DELIMITER ;
```



```
MariaDB [arpit]> DELIMITER //  
MariaDB [arpit]>  
MariaDB [arpit]> CREATE TRIGGER salary_difference_trigger  
-> AFTER INSERT ON CUSTOMERS  
-> FOR EACH ROW  
-> BEGIN  
->     INSERT INTO salary_log (customer_id, customer_name, new_salary)  
->     VALUES (NEW.ID, NEW.NAME, NEW.SALARY);  
-> END;  
-> //  
Query OK, 0 rows affected (0.016 sec)
```

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme**

BCS403 - Database Management System Laboratory



```
INSERT INTO customers (NAME, AGE, ADDRESS, SALARY) VALUES ('John', 30,  
'123 Main St', 50000.00);
```

```
INSERT INTO customers VALUES (1,'Himesh', 25, 'Delhi', 40000);
```

```
MariaDB [arpit]> INSERT INTO customers VALUES (1,'Himesh', 25, 'Delhi', 40000);  
Query OK, 1 row affected (0.006 sec)  
  
MariaDB [arpit]> select * from salary_log;  
+-----+-----+-----+  
| customer_id | customer_name | new_salary |  
+-----+-----+-----+  
|          NULL | John          |      50000 |  
|           1 | Himesh       |      40000 |  
+-----+-----+-----+  
2 rows in set (0.000 sec)
```

Experiment 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)

Code:

```
DELIMITER //
CREATE PROCEDURE fetch_employee_data()
BEGIN
    DECLARE emp_id INT;
    DECLARE emp_name VARCHAR(100);
    DECLARE emp_age INT;
    DECLARE emp_salary INT;
    DECLARE done BOOLEAN DEFAULT FALSE;
    DECLARE emp_cursor CURSOR FOR
        SELECT E_id, E_name, Age, Salary FROM Employee1;
    OPEN emp_cursor;
emp_loop: LOOP
    FETCH emp_cursor INTO emp_id, emp_name, emp_age, emp_salary;
    IF done THEN
        LEAVE emp_loop;
    END IF;
    SELECT CONCAT('Employee ID: ', emp_id, ', Name: ', emp_name, ', Age: ',
emp_age, ', Salary: ', emp_salary);

    END LOOP emp_loop;
    CLOSE emp_cursor;

END //
DELIMITER ;
```

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```
MariaDB [arpit]> DELIMITER //
MariaDB [arpit]>
MariaDB [arpit]> CREATE PROCEDURE fetch_employee_data()
  -> BEGIN
  ->   DECLARE emp_id INT;
  ->   DECLARE emp_name VARCHAR(100);
  ->   DECLARE emp_age INT;
  ->   DECLARE emp_salary INT;
  ->   DECLARE done BOOLEAN DEFAULT FALSE;
  ->   DECLARE emp_cursor CURSOR FOR
  ->     SELECT E_id, E_name, Age, Salary FROM Employee1;
  ->   OPEN emp_cursor;
  ->   emp_loop: LOOP
  ->     FETCH emp_cursor INTO emp_id, emp_name, emp_age, emp_salary;
  ->     IF done THEN
  ->       LEAVE emp_loop;
  ->     END IF;
  ->     SELECT CONCAT('Employee ID: ', emp_id, ', Name: ', emp_name, ', Age: ', emp_age, ', Salary: ', emp_salary);
  ->
  ->   END LOOP emp_loop;
  ->   CLOSE emp_cursor;
  ->
  -> END //
Query OK, 0 rows affected (0.013 sec)
```

CALL fetch_employee_data();

```
+-----+
| CONCAT('Employee ID: ', emp_id, ', Name: ', emp_name, ', Age: ', emp_age, ', Salary: ', emp_salary) |
+-----+
| Employee ID: 1, Name: John Doe, Age: 30, Salary: 50000 |
+-----+
1 row in set (0.001 sec)

+-----+
| CONCAT('Employee ID: ', emp_id, ', Name: ', emp_name, ', Age: ', emp_age, ', Salary: ', emp_salary) |
+-----+
| Employee ID: 2, Name: Jane Smith, Age: 25, Salary: 45000 |
+-----+
1 row in set (0.007 sec)

+-----+
| CONCAT('Employee ID: ', emp_id, ', Name: ', emp_name, ', Age: ', emp_age, ', Salary: ', emp_salary) |
+-----+
| Employee ID: 3, Name: Alice Johnson, Age: 35, Salary: 60000 |
+-----+
1 row in set (0.011 sec)

+-----+
| CONCAT('Employee ID: ', emp_id, ', Name: ', emp_name, ', Age: ', emp_age, ', Salary: ', emp_salary) |
+-----+
| Employee ID: 4, Name: Bob Brown, Age: 28, Salary: 48000 |
+-----+
1 row in set (0.016 sec)

+-----+
| CONCAT('Employee ID: ', emp_id, ', Name: ', emp_name, ', Age: ', emp_age, ', Salary: ', emp_salary) |
+-----+
| Employee ID: 5, Name: Emily Davis, Age: 32, Salary: 55000 |
+-----+
1 row in set (0.024 sec)
```

Experiment 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.

Code:

```
CREATE TABLE O_RollCall (id INT,name VARCHAR(100));
INSERT INTO O_RollCall (id, name) VALUES(1, 'John'),(2, 'Alice'),(3, 'Bob');
CREATE TABLE N_RollCall (id INT,name VARCHAR(100));
```

PL/SQL Code:

```
DELIMITER //

CREATE PROCEDURE MergeRollCallData()
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE o_id INT;
    DECLARE o_name VARCHAR(100);
    DECLARE n_count INT;

    DECLARE o_cursor CURSOR FOR
        SELECT id, name FROM O_RollCall;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN o_cursor;

    read_loop: LOOP
        FETCH o_cursor INTO o_id, o_name;
        IF done THEN
            LEAVE read_loop;
        END IF;
        SELECT COUNT(*) INTO n_count FROM N_RollCall WHERE id = o_id;
        IF n_count = 0 THEN
            INSERT INTO N_RollCall (id, name) VALUES (o_id, o_name);
        END IF;
    END LOOP;
END //
```

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```
END IF;  
END LOOP;
```

```
CLOSE o_cursor;  
END //
```

```
DELIMITER ;
```

```
MariaDB [arpit]> CREATE PROCEDURE MergeRollCallData()  
-> BEGIN  
->   DECLARE done INT DEFAULT FALSE;  
->   DECLARE o_id INT;  
->   DECLARE o_name VARCHAR(100);  
->   DECLARE n_count INT;  
->  
->   DECLARE o_cursor CURSOR FOR  
->     SELECT id, name FROM O_RollCall;  
->  
->   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;  
->  
->   OPEN o_cursor;  
->  
->   read_loop: LOOP  
->     FETCH o_cursor INTO o_id, o_name;  
->     IF done THEN  
->       LEAVE read_loop;  
->     END IF;  
->     SELECT COUNT(*) INTO n_count FROM N_RollCall WHERE id = o_id;  
->     IF n_count = 0 THEN  
->       INSERT INTO N_RollCall (id, name) VALUES (o_id, o_name);  
->     END IF;  
->   END LOOP;  
->  
->   CLOSE o_cursor;  
-> END //  
Query OK, 0 rows affected (0.017 sec)
```

```
CALL MergeRollCallData();
```

CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory



```
MariaDB [arpit]> CALL MergeRollCallData();
Query OK, 5 rows affected (0.017 sec)

MariaDB [arpit]> select * from N_RollCall;
+-----+-----+
| id    | name  |
+-----+-----+
| 1     | Alice |
| 2     | Bob   |
| 3     | Charlie |
+-----+-----+
3 rows in set (0.000 sec)
```

Experiment 7: Install an Open Source NoSQL Data base MongoDB & perform basic CRUD(Create, Read, Update & Delete) operations.
Execute MongoDB basic Queries using CRUD operations.

Code:

How to Connect Mongo database:

Mongosh

```
C:\Users\DELL>mongosh
Current Mongosh Log ID: 661b50e0441a784b1316c9b4
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.3
Using MongoDB:      7.0.8
Using Mongosh:      2.2.3

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2024-04-14T09:05:07.740+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
```

Basic Operations:

Creating Database:

use databasename;

```
test> use sharma;
switched to db sharma
```

Existing Database:

Show dbs;

```
sharma> show dbs;
admin    40.00 KiB
arpit    8.00 KiB
config   72.00 KiB
local    72.00 KiB
sharma> 
```

Creating Collection: a collection is a grouping of MongoDB documents. It is the equivalent of a table in a relational database.

```
db.createCollection("cmrit");
```

```
sharma> db.createCollection("cmrit");  
{ ok: 1 }
```

Create Operation:

For inserting one: `db.collectionname.insertOne({name:"joy",age:20});`

```
arpit> db.Students.insertOne({ name: "John", age: 30 });  
{  
  acknowledged: true,  
  insertedId: ObjectId('661b5f45da0b4b0eab16c9b5')  
}
```

For inserting many:

```
db.Students.insertMany([  
  { name: "Mack", age: 29 },  
  { name: "Alice", age: 25 },  
  { name: "Bob", age: 35 }  
]);
```

```
arpit> db.Students.insertMany([  
...   { name: "Mack", age: 29 },  
...   { name: "Alice", age: 25 },  
...   { name: "Bob", age: 35 }  
... ]);  
{  
  acknowledged: true,  
  insertedIds: {  
    '0': ObjectId('661b6277da0b4b0eab16c9b6'),  
    '1': ObjectId('661b6277da0b4b0eab16c9b7'),  
    '2': ObjectId('661b6277da0b4b0eab16c9b8')  
  }  
}
```

Read Operation:

```
db.Students.find();
```

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```
arpit> db.Students.find();
[
  { _id: ObjectId('661b5f45da0b4b0eab16c9b5'), name: 'John', age: 30 },
  { _id: ObjectId('661b6277da0b4b0eab16c9b6'), name: 'Mack', age: 29 },
  { _id: ObjectId('661b6277da0b4b0eab16c9b7'), name: 'Alice', age: 25 },
  { _id: ObjectId('661b6277da0b4b0eab16c9b8'), name: 'Bob', age: 35 }
]
```

To find One element:

```
db.Students.findOne({age:30});
```

```
arpit> db.Students.findOne({age:30});
{ _id: ObjectId('661b5f45da0b4b0eab16c9b5'), name: 'John', age: 30 }
```

Update Operation:

```
db.Students.updateOne({name:'John'},{$set: {age:40}});
```

```
arpit> db.Students.updateOne({name:'John'}, {$set: {age:40}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
arpit> db.Students.updateOne({name:'Alice'},{$set: {age:30}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
db.Students.updateMany(
  { name: "John" },
  { $set: { age: 35 } }
);
```

The above command will update the "age" field to 35 for all documents in the "users" collection where the "name" field is "John"

CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory



```
arpit> db.Students.updateMany(  
...   { name: "John" },  
...   { $set: { age: 35 } }  
... );  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

```
db.Students.updateMany(  
  { name: { $in: ["John", "Alice"] } },  
  { $set: { age: 45 } }  
);
```

```
arpit> db.Students.updateMany( { name: { $in: ["John", "Alice"] } }, { $set: { age: 45 } } );  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 2,  
  modifiedCount: 0,  
  upsertedCount: 0  
}  
arpit> db.Students.find();  
{  
  { _id: ObjectId('661b5f45da0b4b0eab16c9b5'), name: 'John', age: 45 },  
  { _id: ObjectId('661b6277da0b4b0eab16c9b6'), name: 'Mack', age: 29 },  
  { _id: ObjectId('661b6277da0b4b0eab16c9b7'), name: 'Alice', age: 45 },  
  { _id: ObjectId('661b6277da0b4b0eab16c9b8'), name: 'Bob', age: 35 }  
}
```

Deleting Operation:

```
db.Students.deleteOne({ name: "Alice" });
```

```
{ acknowledged: true, deletedCount: 1 }
```

```
arpit> db.Students.deleteOne({name:"Bob"});  
{ acknowledged: true, deletedCount: 1 }
```

To delete multiple elements:

```
{ acknowledged: true, deletedCount: 1 }  
arpit> db.Students.deleteMany({ name: { $in: ["John", "Mack"] } });  
{ acknowledged: true, deletedCount: 2 }
```

To delete Collection:

**CMR Institute of Technology,
Bengaluru Department of ISE
Lab Manual, IV Semester, 2022 Scheme
BCS403 - Database Management System Laboratory**



```
db.cmrit.drop();  
arpit> db.cmrit.drop();  
true  
arpit>
```

To delete database:

```
arpit> use sharma;  
switched to db sharma  
sharma> db.deleteDatabase();  
TypeError: db.deleteDatabase is not a function  
sharma> db.dropDatabase();  
{ ok: 1, dropped: 'sharma' }
```