Assignment 2
DBMSL
Ansh Bhutada

```
MariaDB [Practice]> CREATE TABLE Employee(Emp_Id INT PRIMARY KEY NOT
NULL,Dept_Id INT NOT NULL,Emp_Fname VARCHAR(255) NOT NULL,Emp_Lname
VARCHAR(255) NOT NULL,Emp_Position VARCHAR(255) NOT NULL,Emp_Salary
DECIMAL(10,2) NOT NULL,Emp_JoinDate DATE NOT NULL,FOREIGN KEY (Dept_Id)
REFERENCES Dept(Dept_Id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.019 sec)

MariaDB [Practice]> CREATE TABLE Project(Project_Id INT PRIMARY KEY NOT
NULL,Dept_Id INT NOT NULL,Project_Name VARCHAR(255) NOT
NULL,Project_Location VARCHAR(255) NOT NULL,Project_Cost DECIMAL(10,2)
NOT NULL,Project_Year INT NOT NULL,FOREIGN KEY (Dept_Id) REFERENCES
Dept(Dept_Id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.019 sec)

MariaDB [Practice]> INSERT INTO Dept(Dept_ID,Dept_Name,Location) VALUES
(1,'Sales','New York'),(2,'IT','San Francisco'),(3,'Engineering','Los
Angeles'),(4,'Marketing','Chicago');
Query OK, 4 rows affected (0.002 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [Practice]> INSERT INTO
Employee(Emp_Id,Dept_Id,Emp_Fname,Emp_Lname,Emp_Position,Emp_Salary,Emp_
JoinDate) VALUES (1,1,'John','Smith','Manager',60000,'2022-05-15'),
(2,1,'Alice','Johnson','Analyst',45000,'2021-10-03'),
(3,3,'David','Lee','Developer',55000,'2023-01-20'),
(4,2,'Mary','Brown','Designer',48000,'2022-08-11'),
(5,3,'Robert','Davis','Engineer',58000,'2020-12-04'),
(6,4,'Sarah','Wilson','Manager',62000,'2019-06-19'),
(7,2,'Michael','Turner','Analyst',52000,'2021-07-25'),
(8,2,'Laura','Adams','Developer',46000,'2022-03-10'),
(9,3,'James','White','Engineer',59000,'2020-12-14'),
(10,2,'Emily','Harris','Manager',63000,'2019-11-08');
Query OK, 10 rows affected (0.002 sec)
Records: 10  Duplicates: 0  Warnings: 0

MariaDB [Practice]> INSERT INTO
Project(Project_Id,Dept_Id,Project_Name,Project_Location,Project_Cost,Pr
oject_Year) VALUES (1,1,'Sales Expansion','Miami',50000,2022),
(2,3,'Product Development','Los Angeles',60000,2023),(3,2,'IT
Infrastructure Upgrade','San Francisco',75000,2023),(4,4,'Customer
Survey','Boston',30000,2021),(5,3,'Testing and Debugging','Cleveland
Ohio',67500,2020);
Query OK, 5 rows affected (0.002 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [Practice]> SELECT * FROM Dept;
+---------+-------------+---------------+
| Dept_Id | Dept_Name   | Location      |
+---------+-------------+---------------+
|       1 | Sales       | New York      |
|       2 | IT          | San Francisco |
|       3 | Engineering | Los Angeles   |
|       4 | Marketing   | Chicago       |
+---------+-------------+---------------+
4 rows in set (0.001 sec)

MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+-----------+-----------+--------------+------------
+--------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |
|      5 |       3 | Robert    | Davis     | Engineer     |   58000.00 |
2020-12-04   |
|      6 |       4 | Sarah     | Wilson    | Manager      |   62000.00 |
2019-06-19   |
|      7 |       2 | Michael   | Turner    | Analyst      |   52000.00 |
2021-07-25   |
|      8 |       2 | Laura     | Adams     | Developer    |   46000.00 |
2022-03-10   |
|      9 |       3 | James     | White     | Engineer     |   59000.00 |
2020-12-14   |
|     10 |       2 | Emily     | Harris    | Manager      |   63000.00 |
2019-11-08   |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
10 rows in set (0.001 sec)

MariaDB [Practice]> SELECT * FROM Project;
+------------+---------+---------------------------+-----------------
+--------------+--------------+
| Project_Id | Dept_Id | Project_Name              | Project_Location |
Project_Cost | Project_Year |
+------------+---------+---------------------------+-----------------
+--------------+--------------+
|          1 |       1 | Sales Expansion           | Miami            |
50000.00 |         2022 |
|          2 |       3 | Product Development       | Los Angeles      |
60000.00 |         2023 |
|          3 |       2 | IT Infrastructure Upgrade | San Francisco    |
75000.00 |         2023 |
|          4 |       4 | Customer Survey           | Boston           |
30000.00 |         2021 |
|          5 |       3 | Testing and Debugging     | Cleveland Ohio   |
67500.00 |         2020 |
+------------+---------+---------------------------+-----------------
+--------------+--------------+
5 rows in set (0.001 sec)
```

```
MariaDB [Practice]> UPDATE Employee SET Dept_Id=4 WHERE Emp_Id=8;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [Practice]> SELECT * FROM Employee WHERE  Dept_Id IN (SELECT
Dept_Id FROM Dept WHERE Dept_Name IN ('Sales','Engineering')) AND
(Emp_Fname LIKE ('D%') OR Emp_Fname LIKE('J%'));
+--------+---------+-----------+-----------+--------------+------------
+--------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |
|      9 |       3 | James     | White     | Engineer     |   59000.00 |
2020-12-14   |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
3 rows in set (0.001 sec)

MariaDB [Practice]> SELECT DISTINCT Emp_Position FROM Employee;
+--------------+
| Emp_Position |
+--------------+
| Manager      |
| Analyst      |
| Developer    |
| Designer     |
| Engineer     |
+--------------+
5 rows in set (0.001 sec)

MariaDB [Practice]> SELECT COUNT(DISTINCT Emp_Position) AS
Employee_Positions FROM Employee;
+--------------------+
| Employee_Positions |
+--------------------+
|                  5 |
+--------------------+
1 row in set (0.001 sec)

MariaDB [Practice]> UPDATE Employee SET Emp_Salary = Emp_Salary*1.1
WHERE Emp_JoinDate < '2021-07-30';
Query OK, 5 rows affected (0.002 sec)
Rows matched: 5  Changed: 5  Warnings: 0

MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+-----------+-----------+--------------+------------
+--------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |
|      5 |       3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |
|      6 |       4 | Sarah     | Wilson    | Manager      |   68200.00 |
2019-06-19   |
|      7 |       2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |
|      8 |       4 | Laura     | Adams     | Developer    |   46000.00 |
2022-03-10   |
|      9 |       3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |
|     10 |       2 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
10 rows in set (0.001 sec)

MariaDB [Practice]> DELETE FROM Dept WHERE Location='Chicago';
Query OK, 1 row affected (0.002 sec)

MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+-----------+-----------+--------------+------------
+--------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |
|      5 |       3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |
|      7 |       2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |
|      9 |       3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |
|     10 |       2 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
8 rows in set (0.001 sec)
```

```
MariaDB [Practice]> SELECT * FROM Project;
+------------+---------+----------------------------+------------------
+--------------+--------------+
| Project_Id | Dept_Id | Project_Name               | Project_Location |
Project_Cost | Project_Year |
+------------+---------+----------------------------+------------------
+--------------+--------------+
|          1 |       1 | Sales Expansion            | Miami            |
50000.00 |         2022 |
|          2 |       3 | Product Development        | Los Angeles      |
60000.00 |         2023 |
|          3 |       2 | IT Infrastructure Upgrade | San Francisco    |
75000.00 |         2023 |
|          5 |       3 | Testing and Debugging      | Cleveland Ohio   |
67500.00 |         2020 |
+------------+---------+----------------------------+------------------
+--------------+--------------+
```
4 rows in set (0.001 sec)

```
MariaDB [Practice]> SELECT Project_Name FROM Project WHERE
Project_Location = 'Los Angeles';
+---------------------+
| Project_Name        |
+---------------------+
| Product Development |
+---------------------+
```
1 row in set (0.001 sec)

```
MariaDB [Practice]> SELECT * FROM Project WHERE Project_Cost > 61000 AND
Project_Cost < 81000;
+------------+---------+----------------------------+------------------
+--------------+--------------+
| Project_Id | Dept_Id | Project_Name               | Project_Location |
Project_Cost | Project_Year |
+------------+---------+----------------------------+------------------
+--------------+--------------+
|          3 |       2 | IT Infrastructure Upgrade | San Francisco    |
75000.00 |         2023 |
|          5 |       3 | Testing and Debugging      | Cleveland Ohio   |
67500.00 |         2020 |
+------------+---------+----------------------------+------------------
+--------------+--------------+
```
2 rows in set (0.003 sec)

```
MariaDB [Practice]> SELECT * FROM Project WHERE Project_Cost = SELECT
MAX(Project_Cost) FROM Project_Cost;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right
syntax to use near 'SELECT MAX(Project_Cost) FROM Project_Cost' at line
1
MariaDB [Practice]> SELECT * FROM Project WHERE Project_Cost = SELECT
MAX(Project_Cost) FROM Project;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right
syntax to use near 'SELECT MAX(Project_Cost) FROM Project' at line 1
```

```
MariaDB [Practice]> SELECT * FROM Project WHERE Project_Cost = (SELECT
MAX(Project_Cost) FROM Project);
+------------+---------+----------------------------+------------------
+--------------+--------------+
| Project_Id | Dept_Id | Project_Name               | Project_Location |
Project_Cost | Project_Year |
+------------+---------+----------------------------+------------------
+--------------+--------------+
|          3 |       2 | IT Infrastructure Upgrade | San Francisco    |
75000.00 |         2023 |
+------------+---------+----------------------------+------------------
+--------------+--------------+
1 row in set (0.003 sec)

MariaDB [Practice]> SELECT AVG(Project_Cost) AS Average_Project_Cost
FROM Project;
+----------------------+
| Average_Project_Cost |
+----------------------+
|         63125.000000 |
+----------------------+
1 row in set (0.001 sec)

MariaDB [Practice]> SELECT * FROM Employee ORDER BY Emp_Lname DESC;
+--------+---------+-----------+-----------+--------------+------------
+--------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
|      9 |       3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |
|      7 |       2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |
|     10 |       2 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |
|      5 |       3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |
+--------+---------+-----------+-----------+--------------+------------
+--------------+
8 rows in set (0.001 sec)

MariaDB [Practice]> SELECT Project_Name,Project_Cost,Project_Location
FROM Project WHERE Project_Year IN (2020,2022);
+-----------------------+--------------+------------------+
| Project_Name          | Project_Cost | Project_Location |
+-----------------------+--------------+------------------+
| Sales Expansion       |     50000.00 | Miami            |
| Testing and Debugging |     67500.00 | Cleveland Ohio   |
+-----------------------+--------------+------------------+
2 rows in set (0.001 sec)

MariaDB [Practice]> CREATE VIEW Employee_View AS SELECT
Emp_Id,Emp_Fname,Emp_Salary FROM Employee;
Query OK, 0 rows affected (0.026 sec)

MariaDB [Practice]> SELECT * FROM Employee_View;
+--------+-----------+------------+
| Emp_Id | Emp_Fname | Emp_Salary |
+--------+-----------+------------+
|      1 | John      |   60000.00 |
|      2 | Alice     |   45000.00 |
|      3 | David     |   55000.00 |
|      4 | Mary      |   48000.00 |
|      5 | Robert    |   63800.00 |
|      7 | Michael   |   57200.00 |
|      9 | James     |   64900.00 |
|     10 | Emily     |   69300.00 |
+--------+-----------+------------+
8 rows in set (0.002 sec)

MariaDB [Practice]> CREATE VIEW Dept_View AS SELECT Dept_Id,Location
FROM Dept;
Query OK, 0 rows affected (0.016 sec)

MariaDB [Practice]> SELECT * FROM Dept_View;
+---------+---------------+
| Dept_Id | Location      |
+---------+---------------+
|       1 | New York      |
|       2 | San Francisco |
|       3 | Los Angeles   |
+---------+---------------+
3 rows in set (0.001 sec)

MariaDB [Practice]> SELECT * FROM Dept;
+---------+-------------+---------------+
| Dept_Id | Dept_Name   | Location      |
+---------+-------------+---------------+
|       1 | Sales       | New York      |
|       2 | IT          | San Francisco |
|       3 | Engineering | Los Angeles   |
+---------+-------------+---------------+
3 rows in set (0.001 sec)

MariaDB [Practice]> CREATE VIEW Project_View AS SELECT
Project_Id,Project_Name,Project_Location FROM Project;
Query OK, 0 rows affected (0.016 sec)

MariaDB [Practice]> SELECT * FROM Project_View;
+------------+----------------------------+------------------+
| Project_Id | Project_Name               | Project_Location |
+------------+----------------------------+------------------+
|          1 | Sales Expansion            | Miami            |
|          2 | Product Development       | Los Angeles      |
|          3 | IT Infrastructure Upgrade | San Francisco    |
|          5 | Testing and Debugging     | Cleveland Ohio   |
+------------+---------------------------+------------------+
4 rows in set (0.001 sec)

MariaDB [Practice]> CREATE VIEW Merged_View AS SELECT
Emp_Id,Emp_Fname,Emp_Position,(SELECT Dept_Name FROM Dept WHERE Dept_Id
= Employee.Dept_Id) AS Dept_Name FROM Employee;
Query OK, 0 rows affected (0.016 sec)

MariaDB [Practice]> SELECT * FROM Merged_View;
+--------+-----------+--------------+-------------+
| Emp_Id | Emp_Fname | Emp_Position | Dept_Name   |
+--------+-----------+--------------+-------------+
|      1 | John      | Manager      | Sales       |
|      2 | Alice     | Analyst      | Sales       |
|      3 | David     | Developer    | Engineering |
|      4 | Mary      | Designer     | IT          |
|      5 | Robert    | Engineer     | Engineering |
|      7 | Michael   | Analyst      | IT          |
|      9 | James     | Engineer     | Engineering |
|     10 | Emily     | Manager      | IT          |
+--------+-----------+--------------+-------------+
8 rows in set (0.003 sec)

MariaDB [Practice]> CREATE INDEX Ind_Emp_Lname ON Employee(Lname);
ERROR 1072 (42000): Key column 'Lname' doesn't exist in table
MariaDB [Practice]> CREATE INDEX Ind_Emp_Lname ON Employee(Emp_Lname);
Query OK, 0 rows affected (0.038 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Practice]> SHOW INDEX FROM Employee WHERE Key_Name =
'Ind_Emp_Lname';
+----------+------------+---------------+--------------+-------------
+-----------+-------------+----------+--------+------+------------
+---------+---------------+---------+
| Table    | Non_unique | Key_name      | Seq_in_index | Column_name |
Collation | Cardinality | Sub_part | Packed | Null | Index_type |
Comment | Index_comment | Ignored |
+----------+------------+---------------+--------------+-------------
+-----------+-------------+----------+--------+------+------------
+---------+---------------+---------+
| employee |          1 | Ind_Emp_Lname |            1 | Emp_Lname   | A
|          8 |     NULL | NULL   |      | BTREE      |         |
| NO      |
+----------+------------+---------------+--------------+-------------
+-----------+-------------+----------+--------+------+------------
+---------+---------------+---------+
1 row in set (0.003 sec)

MariaDB [Practice]> SHOW CREATE TABLE Employee;
+----------
+--------------------------------------------------------------------
--------------------------------------------------------------------
--------------------------------------------------------------------
```

```
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------+
| Table    | Create Table
|
+----------
+-----------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------+
| Employee | CREATE TABLE `Employee` (
  `Emp_Id` int(11) NOT NULL,
  `Dept_Id` int(11) NOT NULL,
  `Emp_Fname` varchar(255) NOT NULL,
  `Emp_Lname` varchar(255) NOT NULL,
  `Emp_Position` varchar(255) NOT NULL,
  `Emp_Salary` decimal(10,2) NOT NULL,
  `Emp_JoinDate` date NOT NULL,
  PRIMARY KEY (`Emp_Id`),
  KEY `Dept_Id` (`Dept_Id`),
  KEY `Ind_Emp_Lname` (`Emp_Lname`),
  CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`Dept_Id`) REFERENCES `Dept`
(`Dept_Id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci |
+----------
+-----------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------+
1 row in set (0.000 sec)
```

Assignment 3
DBMSL
Ansh Bhutada

```
MariaDB [Practice]> DESCRIBE Employee;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| Emp_Id       | int(11)       | NO   | PRI | NULL    |       |
| Dept_Id      | int(11)       | NO   | MUL | NULL    |       |
| Emp_Fname    | varchar(255)  | NO   |     | NULL    |       |
| Emp_Lname    | varchar(255)  | NO   | MUL | NULL    |       |
| Emp_Position | varchar(255)  | NO   |     | NULL    |       |
| Emp_Salary   | decimal(10,2) | NO   |     | NULL    |       |
| Emp_JoinDate | date          | NO   |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
7 rows in set (0.008 sec)

MariaDB [Practice]> ALTER TABLE Employee ADD COLUMN Project_Id INT;
Query OK, 0 rows affected (0.038 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Practice]> ALTER TABLE Employee ADD CONSTRAINT
fk_Employee_Project_Id FOREIGN KEY (Project_Id) REFERENCES
Project(Project_Id);
Query OK, 8 rows affected (0.044 sec)
Records: 8  Duplicates: 0  Warnings: 0

MariaDB [Practice]> DESCRIBE Employee;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| Emp_Id       | int(11)       | NO   | PRI | NULL    |       |
| Dept_Id      | int(11)       | NO   | MUL | NULL    |       |
| Emp_Fname    | varchar(255)  | NO   |     | NULL    |       |
| Emp_Lname    | varchar(255)  | NO   | MUL | NULL    |       |
| Emp_Position | varchar(255)  | NO   |     | NULL    |       |
| Emp_Salary   | decimal(10,2) | NO   |     | NULL    |       |
| Emp_JoinDate | date          | NO   |     | NULL    |       |
| Project_Id   | int(11)       | YES  | MUL | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
8 rows in set (0.005 sec)

MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate | Project_Id |
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |       NULL |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |       NULL |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |       NULL |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |       NULL |
|      5 |       3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |       NULL |
|      7 |       2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |       NULL |
|      9 |       3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |       NULL |
|     10 |       2 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |       NULL |
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
8 rows in set (0.001 sec)

MariaDB [Practice]> UPDATE Employee JOIN Project ON Employee.Dept_Id =
Project.Dept_Id SET Employee.Project_Id = Project.Project_Id;
Query OK, 8 rows affected (0.008 sec)
Rows matched: 8  Changed: 8  Warnings: 0

MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate | Project_Id |
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |          1 |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |          1 |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |          2 |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |          3 |
|      5 |       3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |          2 |
|      7 |       2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |          3 |
|      9 |       3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |          2 |
|     10 |       2 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |          3 |
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
8 rows in set (0.001 sec)


MariaDB [Practice]> UPDATE Employee SET Project_Id = 5 WHERE Emp_Id =
10;
Query OK, 1 row affected (0.028 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+----------+----------+--------------+-----------
+--------------+------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate | Project_Id |
+--------+---------+----------+----------+--------------+-----------
+--------------+------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |          1 |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |          1 |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |          2 |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |          3 |
|      5 |       3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |          2 |
|      7 |       2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |          3 |
|      9 |       3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |          2 |
|     10 |       2 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |          5 |
+--------+---------+----------+----------+--------------+-----------
+--------------+------------+
```
8 rows in set (0.001 sec)

```
MariaDB [Practice]> SELECT * FROM Employee NATURAL JOIN Dept;
+---------+--------+----------+----------+--------------+-----------
+--------------+------------+-------------+---------------+
| Dept_Id | Emp_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate | Project_Id | Dept_Name   | Location      |
+---------+--------+----------+----------+--------------+-----------
+--------------+------------+-------------+---------------+
|       1 |      1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |          1 | Sales       | New York      |
|       1 |      2 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |          1 | Sales       | New York      |
|       3 |      3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |          2 | Engineering | Los Angeles   |
|       2 |      4 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |          3 | IT          | San Francisco |
|       3 |      5 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |          2 | Engineering | Los Angeles   |
|       2 |      7 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |          3 | IT          | San Francisco |
|       3 |      9 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |          2 | Engineering | Los Angeles   |
|       2 |     10 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |          5 | IT          | San Francisco |
+---------+--------+----------+----------+--------------+-----------
+--------------+------------+-------------+---------------+
```
8 rows in set (0.001 sec)

```
MariaDB [Practice]> SELECT
E.Emp_Fname,E.Emp_Position,D.Location,E.Emp_JoinDate FROM Employee E
JOIN Dept D ON E.Dept_Id=D.Dept_Id WHERE D.Dept_Id=1;
+-----------+--------------+----------+--------------+
| Emp_Fname | Emp_Position | Location | Emp_JoinDate |
+-----------+--------------+----------+--------------+
| John      | Manager      | New York | 2022-05-15   |
| Alice     | Analyst      | New York | 2021-10-03   |
+-----------+--------------+----------+--------------+
```
2 rows in set (0.003 sec)

```
MariaDB [Practice]> SELECT
E.Emp_Fname,E.Emp_Position,D.Location,E.Emp_JoinDate FROM Employee E
JOIN Dept D ON E.Dept_Id=D.Dept_Id WHERE D.Dept_Id=2;
+-----------+--------------+---------------+--------------+
| Emp_Fname | Emp_Position | Location      | Emp_JoinDate |
+-----------+--------------+---------------+--------------+
| Mary      | Designer     | San Francisco | 2022-08-11   |
| Michael   | Analyst      | San Francisco | 2021-07-25   |
| Emily     | Manager      | San Francisco | 2019-11-08   |
+-----------+--------------+---------------+--------------+
```
3 rows in set (0.001 sec)

```
MariaDB [Practice]> SELECT
E.Emp_Fname,E.Emp_Position,D.Location,E.Emp_JoinDate FROM Employee E
JOIN Dept D ON E.Dept_Id=D.Dept_Id WHERE D.Dept_Id=3;
+-----------+--------------+-------------+--------------+
| Emp_Fname | Emp_Position | Location    | Emp_JoinDate |
+-----------+--------------+-------------+--------------+
| David     | Developer    | Los Angeles | 2023-01-20   |
| Robert    | Engineer     | Los Angeles | 2020-12-04   |
| James     | Engineer     | Los Angeles | 2020-12-14   |
+-----------+--------------+-------------+--------------+
```
3 rows in set (0.001 sec)

```
MariaDB [Practice]> SELECT
E.Emp_Fname,E.Emp_Position,D.Location,E.Emp_JoinDate FROM Employee E
JOIN Dept D ON E.Dept_Id=D.Dept_Id WHERE D.Dept_Id=4;
```
Empty set (0.001 sec)

```
MariaDB [Practice]> SELECT
E.Emp_Id,E.Emp_Fname,E.Emp_Lname,E.Emp_Position,E.Emp_Salary,E.Emp_JoinD
ate,P.Project_Id,P.Project_Cost FROM Employee E LEFT JOIN Project P ON
E.Project_Id = P.Project_Id AND P.Project_Location <> 'San Francisco'
WHERE P.Project_Id IS NOT NULL;
+--------+-----------+-----------+--------------+-----------
+--------------+------------+--------------+
| Emp_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate | Project_Id | Project_Cost |
+--------+-----------+-----------+--------------+-----------
+--------------+------------+--------------+
|      1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |          1 |     50000.00 |
|      2 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |          1 |     50000.00 |
|      3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |          2 |    60000.00 |
|      5 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |          2 |    60000.00 |
|      9 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |          2 |    60000.00 |
|     10 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |          5 |    67500.00 |
+--------+-----------+-----------+--------------+------------
+--------------+-----------+-------------+
```
6 rows in set (0.003 sec)

```
MariaDB [Practice]> SELECT * FROM Project;
+------------+---------+----------------------------+------------------
+-------------+-------------+
| Project_Id | Dept_Id | Project_Name               | Project_Location |
Project_Cost | Project_Year |
+------------+---------+----------------------------+------------------
+-------------+-------------+
|          1 |       1 | Sales Expansion            | Miami            |
50000.00 |         2022 |
|          2 |       3 | Product Development        | Los Angeles      |
60000.00 |         2023 |
|          3 |       2 | IT Infrastructure Upgrade  | San Francisco    |
75000.00 |         2023 |
|          5 |       3 | Testing and Debugging      | Cleveland Ohio   |
67500.00 |         2020 |
+------------+---------+----------------------------+------------------
+-------------+-------------+
```
4 rows in set (0.001 sec)

```
MariaDB [Practice]> SELECT
D.Dept_Name,E.Emp_Fname,E.Emp_Lname,E.Emp_Position FROM Employee E JOIN
Project P ON E.Project_Id = P.Project_Id JOIN Dept D ON E.Dept_Id =
D.Dept_Id  WHERE P.Project_Year = 2020;
+-----------+-----------+-----------+--------------+
| Dept_Name | Emp_Fname | Emp_Lname | Emp_Position |
+-----------+-----------+-----------+--------------+
| IT        | Emily     | Harris    | Manager      |
+-----------+-----------+-----------+--------------+
```
1 row in set (0.002 sec)

```
MariaDB [Practice]> SELECT E.Emp_Position,D.Dept_Name FROM Employee E
JOIN Dept D ON E.Dept_Id = D.Dept_Id JOIN Project P ON E.Project_Id =
P.Project_Id WHERE P.Project_Cost > 61000;
+--------------+-------------+
| Emp_Position | Dept_Name   |
+--------------+-------------+
| Designer     | IT          |
| Analyst      | IT          |
| Manager      | Engineering |
+--------------+-------------+
3 rows in set (0.001 sec)


MariaDB [Practice]> SELECT
D.Dept_Name,E.Emp_Fname,E.Emp_Lname,E.Emp_Position FROM Employee E JOIN
Dept D ON E.Dept_Id = D.Dept_Id JOIN Project P ON E.Project_Id =
P.Project_Id WHERE P.Project_Year = 2023;
+-------------+-----------+-----------+--------------+
| Dept_Name   | Emp_Fname | Emp_Lname | Emp_Position |
+-------------+-----------+-----------+--------------+
| IT          | Mary      | Brown     | Designer     |
| IT          | Michael   | Turner    | Analyst      |
| Engineering | David     | Lee       | Developer    |
| Engineering | Robert    | Davis     | Engineer     |
| Engineering | James     | White     | Engineer     |
+-------------+-----------+-----------+--------------+
5 rows in set (0.001 sec)

MariaDB [Practice]> SELECT
D.Dept_Name,E.Emp_Fname,E.Emp_Lname,E.Emp_Position FROM Employee E JOIN
Dept D ON E.Dept_Id = D.Dept_Id JOIN Project P ON E.Project_Id =
P.Project_Id WHERE P.Project_Year = 2020;
+-------------+-----------+-----------+--------------+
| Dept_Name   | Emp_Fname | Emp_Lname | Emp_Position |
+-------------+-----------+-----------+--------------+
| Engineering | Emily     | Harris    | Manager      |
+-------------+-----------+-----------+--------------+
1 row in set (0.001 sec)

MariaDB [Practice]> SELECT Project_Name FROM Project WHERE Project_Year
> 2020;
+---------------------------+
| Project_Name              |
+---------------------------+
| Sales Expansion           |
| Product Development       |
| IT Infrastructure Upgrade |
+---------------------------+
3 rows in set (0.001 sec)

MariaDB [Practice]> SELECT D.Dept_Name FROM Dept D JOIN Employee E ON
D.Dept_Id = E.Dept_Id GROUP BY D.Dept_Name HAVING COUNT(*) >= 3;
+-------------+
| Dept_Name   |
+-------------+
| Engineering |
+-------------+
1 row in set (0.006 sec)

MariaDB [Practice]> SELECT D.Dept_Name FROM Dept D JOIN Employee E ON
D.Dept_Id = E.Dept_Id GROUP BY D.Dept_Name HAVING COUNT(*) >= 2;
+-------------+
| Dept_Name   |
+-------------+
| Engineering |
| IT          |
| Sales       |
+-------------+
3 rows in set (0.001 sec)

MariaDB [Practice]> SELECT D.Dept_Name FROM Dept D JOIN Employee E ON
D.Dept_Id = E.Dept_Id GROUP BY D.Dept_Name HAVING COUNT(*) = 3;
Empty set (0.001 sec)



MariaDB [Practice]> SELECT COUNT(*) as Total_Employee_Before_2022 FROM
Employee E JOIN Project P ON E.Project_Id = P.Project_Id WHERE
P.Project_Year < 2022;
+----------------------------+
| Total_Employee_Before_2022 |
+----------------------------+
|                          1 |
+----------------------------+
1 row in set (0.001 sec)

MariaDB [Practice]> SELECT COUNT(*) as Total_Employee_Before_2022 FROM
Employee E JOIN Project P ON E.Project_Id = P.Project_Id WHERE
P.Project_Year < 2023;
+----------------------------+
| Total_Employee_Before_2022 |
+----------------------------+
|                          3 |
+----------------------------+
1 row in set (0.001 sec)

MariaDB [Practice]> SELECT COUNT(*) as Total_Employee_Before_2023 FROM
Employee E JOIN Project P ON E.Project_Id = P.Project_Id WHERE
P.Project_Year < 2023;
+----------------------------+
| Total_Employee_Before_2023 |
+----------------------------+
|                          3 |
+----------------------------+
1 row in set (0.001 sec)

MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+-----------+-----------+--------------+------------
+-------------+------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate | Project_Id |
+--------+---------+-----------+-----------+--------------+------------
+-------------+------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |          1 |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |          1 |
|      3 |       3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |          2 |
|      4 |       2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |          3 |
|      5 |       3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |          2 |
|      7 |       2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |          3 |
|      9 |       3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |          2 |
|     10 |       3 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |          5 |
+--------+---------+-----------+-----------+-------------+------------
+-------------+-----------+
8 rows in set (0.001 sec)

MariaDB [Practice]> SELECT * FROM Project;
+------------+---------+----------------------------+-----------------
+-------------+-------------+
| Project_Id | Dept_Id | Project_Name               | Project_Location |
Project_Cost | Project_Year |
+------------+---------+----------------------------+-----------------
+-------------+-------------+
|          1 |       1 | Sales Expansion            | Miami            |
50000.00 |         2022 |
|          2 |       3 | Product Development        | Los Angeles      |
60000.00 |         2023 |
|          3 |       2 | IT Infrastructure Upgrade | San Francisco    |
75000.00 |         2023 |
|          5 |       3 | Testing and Debugging      | Cleveland Ohio   |
67500.00 |         2020 |
+------------+---------+----------------------------+-----------------
+-------------+-------------+
4 rows in set (0.001 sec)

MariaDB [Practice]> CREATE VIEW Employee_Dept_View AS SELECT
E.Emp_Id,Emp_Fname,Emp_Lname,E.Emp_Position,E.Emp_JoinDate,D.Dept_Name
FROM Employee E JOIN Dept D ON E.Dept_Id = D.Dept_Id;
Query OK, 0 rows affected (0.020 sec)

MariaDB [Practice]> SELECT * FROM Employee_Dept_View;
+--------+-----------+-----------+--------------+-------------
+-------------+
| Emp_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_JoinDate |
Dept_Name   |
+--------+-----------+-----------+--------------+-------------
+-------------+
|      1 | John      | Smith     | Manager      | 2022-05-15   | Sales
|
|      2 | Alice     | Johnson   | Analyst      | 2021-10-03   | Sales
|
|      3 | David     | Lee       | Developer    | 2023-01-20   |
Engineering |
|      4 | Mary      | Brown     | Designer     | 2022-08-11   | IT
|
|      5 | Robert    | Davis     | Engineer     | 2020-12-04   |
Engineering |
|      7 | Michael   | Turner    | Analyst      | 2021-07-25   | IT
|
|      9 | James     | White     | Engineer     | 2020-12-14   |
Engineering |
|     10 | Emily     | Harris    | Manager      | 2019-11-08   |
Engineering |
+--------+----------+----------+------------+-------------
+-------------+
8 rows in set (0.005 sec)



MariaDB [Practice]> CREATE VIEW Employee_View AS SELECT
E.Emp_Id,Emp_Fname,E.Emp_Position,E.Emp_Salary FROM Employee E WHERE
Emp_Position IN ('Manager','Engineer');
Query OK, 0 rows affected (0.016 sec)

MariaDB [Practice]> SELECT * FROM Employee_View;
+--------+-----------+--------------+------------+
| Emp_Id | Emp_Fname | Emp_Position | Emp_Salary |
+--------+-----------+--------------+------------+
|      1 | John      | Manager      |   60000.00 |
|      5 | Robert    | Engineer     |   63800.00 |
|      9 | James     | Engineer     |   64900.00 |
|     10 | Emily     | Manager      |   69300.00 |
+--------+-----------+--------------+------------+
4 rows in set (0.001 sec)

MariaDB [Practice]> INSERT INTO
Employee(Emp_Id,Dept_Id,Emp_Fname,Emp_Lname,Emp_Position,Emp_Salary,Emp_
JoinDate,Project_Id) VALUES
(11,2,'Gary','Williams','Manager',57500,'2021-08-07',5);
Query OK, 1 row affected (0.004 sec)

MariaDB [Practice]> SELECT * FROM Employee_View;
+--------+-----------+--------------+------------+
| Emp_Id | Emp_Fname | Emp_Position | Emp_Salary |
+--------+-----------+--------------+------------+
|      1 | John      | Manager      |   60000.00 |
|      5 | Robert    | Engineer     |   63800.00 |
|      9 | James     | Engineer     |   64900.00 |
|     10 | Emily     | Manager      |   69300.00 |
|     11 | Gary      | Manager      |   57500.00 |
+--------+-----------+--------------+------------+
5 rows in set (0.001 sec)

MariaDB [Practice]> SELECT * FROM Employee;
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
| Emp_Id | Dept_Id | Emp_Fname | Emp_Lname | Emp_Position | Emp_Salary |
Emp_JoinDate | Project_Id |
+--------+---------+-----------+-----------+--------------+------------
+--------------+------------+
|      1 |       1 | John      | Smith     | Manager      |   60000.00 |
2022-05-15   |          1 |
|      2 |       1 | Alice     | Johnson   | Analyst      |   45000.00 |
2021-10-03   |          1 |
|      3 |        3 | David     | Lee       | Developer    |   55000.00 |
2023-01-20   |          2 |
|      4 |        2 | Mary      | Brown     | Designer     |   48000.00 |
2022-08-11   |          3 |
|      5 |        3 | Robert    | Davis     | Engineer     |   63800.00 |
2020-12-04   |          2 |
|      7 |        2 | Michael   | Turner    | Analyst      |   57200.00 |
2021-07-25   |          3 |
|      9 |        3 | James     | White     | Engineer     |   64900.00 |
2020-12-14   |          2 |
|     10 |        3 | Emily     | Harris    | Manager      |   69300.00 |
2019-11-08   |          5 |
|     11 |        2 | Gary      | Williams  | Manager      |   57500.00 |
2021-08-07   |          5 |
+--------+---------+-----------+-----------+-------------+------------
+--------------+------------+
9 rows in set (0.001 sec)

MariaDB [Practice]> UPDATE Employee_View SET Emp_Salary = 59500 WHERE
Emp_Id = 11;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [Practice]> SELECT * FROM Employee_View;
+--------+-----------+--------------+------------+
| Emp_Id | Emp_Fname | Emp_Position | Emp_Salary |
+--------+-----------+--------------+------------+
|      1 | John      | Manager      |   60000.00 |
|      5 | Robert    | Engineer     |   63800.00 |
|      9 | James     | Engineer     |   64900.00 |
|     10 | Emily     | Manager      |   69300.00 |
|     11 | Gary      | Manager      |   59500.00 |
+--------+-----------+--------------+------------+
5 rows in set (0.001 sec)

MariaDB [Practice]> DELETE FROM Employee_View WHERE Emp_Id = 11;
Query OK, 1 row affected (0.002 sec)

MariaDB [Practice]> SELECT * FROM Employee_View;
+--------+-----------+--------------+------------+
| Emp_Id | Emp_Fname | Emp_Position | Emp_Salary |
+--------+-----------+--------------+------------+
|      1 | John      | Manager      |   60000.00 |
|      5 | Robert    | Engineer     |   63800.00 |
|      9 | James     | Engineer     |   64900.00 |
|     10 | Emily     | Manager      |   69300.00 |
+--------+-----------+--------------+------------+
4 rows in set (0.001 sec)

MariaDB [Practice]> DROP VIEW Employee_View;
Query OK, 0 rows affected (0.021 sec)

MariaDB [Practice]> SELECT * FROM Employee_View;
ERROR 1146 (42S02): Table 'practice.employee_view' doesn't exist
```

Assignment 4
DBMSL
Ansh Bhutada

```
MariaDB [Practice]> CREATE TABLE Area(radius DECIMAL(10,2),area
DECIMAL(10,2));
Query OK, 0 rows affected (0.026 sec)

MariaDB [Practice]> ALTER TABLE Area RENAME TO Areas;
Query OK, 0 rows affected (0.023 sec)

MariaDB [Practice]> DELIMITER //
MariaDB [Practice]> CREATE PROCEDURE CalcArea()
    -> BEGIN
    -> DECLARE v_radius DECIMAL(10,2);
    -> DECLARE v_area DECIMAL(10,2);
    -> SET v_radius = 5.00;
    -> WHILE v_radius <= 9.00 DO
    -> SET v_area = 3.14159 * v_radius * v_radius;
    -> INSERT INTO Areas(radius,area) VALUES (v_radius,v_area);
    -> SET v_radius = v_radius + 1;
    -> END WHILE;
    -> END //
Query OK, 0 rows affected (0.019 sec)

MariaDB [Practice]> DELIMITER //
MariaDB [Practice]> CALL CalcArea();
    -> END//
Query OK, 5 rows affected (0.009 sec)

ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right
syntax to use near 'END' at line 1
MariaDB [Practice]> DELIMITER ;
MariaDB [Practice]> CALL CalcArea();
Query OK, 5 rows affected (0.002 sec)

MariaDB [Practice]> SELECT * FROM Areas;
+--------+--------+
| radius | area   |
+--------+--------+
|   5.00 |  78.54 |
|   6.00 | 113.10 |
|   7.00 | 153.94 |
|   8.00 | 201.06 |
|   9.00 | 254.47 |
|   5.00 |  78.54 |
|   6.00 | 113.10 |
|   7.00 | 153.94 |
|   8.00 | 201.06 |
|   9.00 | 254.47 |
+--------+--------+
10 rows in set (0.002 sec)
```

// idhar multiple entries aye kyun ki pahle procedure me >30 pe (* 50) likhna rah gya the

```
DELIMITER //

CREATE PROCEDURE CalcFineWithUpdateStatus(IN in_roll_no INT, IN
in_book_name VARCHAR(100))
BEGIN
    DECLARE v_issue_date DATE;
    DECLARE v_status VARCHAR(1);
    DECLARE v_days_overdue INT;
    DECLARE v_fine_amt INT DEFAULT 0;

    -- Check for negative roll number
    IF in_roll_no < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Roll number cannot be negative';
        RETURN;
    END IF;

    -- Get issue date and status
    SELECT Date_Of_Issue, Status
    INTO v_issue_date, v_status
    FROM Borrower
    WHERE Roll_No = in_roll_no AND Name_Of_Book = in_book_name;

    -- Calculate days overdue
    SET v_days_overdue = DATEDIFF(CURDATE(), v_issue_date);

    -- Calculate fine amount
    IF v_days_overdue > 30 THEN
        SET v_fine_amt = 75 + (v_days_overdue - 30) * 50;
    ELSEIF v_days_overdue > 15 THEN
        SET v_fine_amt = (v_days_overdue - 15) * 5;
    END IF;

    -- Insert fine if applicable
    IF v_fine_amt > 0 THEN
        INSERT INTO Fine (Roll_No, Date, Amount)
        VALUES (in_roll_no, CURDATE(), v_fine_amt);
    END IF;

    -- Update borrower status
    UPDATE Borrower
    SET Status = 'R'
    WHERE Roll_No = in_roll_no AND Name_Of_Book = in_book_name;

    -- Return fine amount, old status, and new status
    SELECT v_fine_amt AS Fine_Amount, v_status AS Old_Status, 'R' AS
New_Status;
END//

DELIMITER ;
```

```
Query OK, 0 rows affected (0.014 sec)

MariaDB [Practice]> CALL CalcFineWithUpdateStatus(5,'Book5');
-> Exit//
+-------------+------------+------------+
| Fine_Amount | Old_Status | New_Status |
+-------------+------------+------------+
|          50 | R          | R          |
+-------------+------------+------------+
1 row in set (0.003 sec)

Query OK, 2 rows affected (0.003 sec)

ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right
syntax to use near 'Exit' at line 1
MariaDB [Practice]> DELIMITER ;
MariaDB [Practice]> CALL CalcFineWithUpdateStatus(5,'Book5');
+-------------+------------+------------+
| Fine_Amount | Old_Status | New_Status |
+-------------+------------+------------+
|          50 | R          | R          |
+-------------+------------+------------+
1 row in set (0.002 sec)

Query OK, 2 rows affected (0.002 sec)

MariaDB [Practice]> CALL CalcFineWithUpdateStatus(4,'Book4');
+-------------+------------+------------+
| Fine_Amount | Old_Status | New_Status |
+-------------+------------+------------+
|         925 | I          | R          |
+-------------+------------+------------+
1 row in set (0.002 sec)

Query OK, 3 rows affected (0.002 sec)

MariaDB [Practice]> CALL CalcFineWithUpdateStatus(3,'Book3');
+-------------+------------+------------+
| Fine_Amount | Old_Status | New_Status |
+-------------+------------+------------+
|           0 | R          | R          |
+-------------+------------+------------+
1 row in set (0.001 sec)

Query OK, 1 row affected (0.001 sec)

MariaDB [Practice]> CALL CalcFineWithUpdateStatus(1,'Book1');
+-------------+------------+------------+
| Fine_Amount | Old_Status | New_Status |
+-------------+------------+------------+
|        3075 | R          | R          |
+-------------+------------+------------+
1 row in set (0.002 sec)
```

```
Query OK, 2 rows affected (0.002 sec)

MariaDB [Practice]> CALL CalcFineWithUpdateStatus(2,'Book2');
+-------------+------------+------------+
| Fine_Amount | Old_Status | New_Status |
+-------------+------------+------------+
|          65 | I          | R          |
+-------------+------------+------------+
1 row in set (0.002 sec)

Query OK, 3 rows affected (0.002 sec)

MariaDB [Practice]> SELECT * FROM Fine;
+---------+------------+--------+
| Roll_No | Date       | Amount |
+---------+------------+--------+
|       5 | 2023-10-30 |     50 |
|       1 | 2023-10-30 |    135 |
|       5 | 2023-10-30 |     50 |
|       5 | 2023-10-30 |     50 |
|       4 | 2023-10-30 |    925 |
|       1 | 2023-10-30 |   3075 |
|       2 | 2023-10-30 |     65 |
+---------+------------+--------+
7 rows in set (0.001 sec)

MariaDB [Practice]>
```

ASSIGNMENT 6
CURSOR

```
 CREATE TABLE OldTable(Id PRIMARY KEY INT NOT NULL,Name VARCHAR(100) NOT
NULL);
ERROR 4161 (HY000): Unknown data type: 'PRIMARY'
MariaDB [Practice]> CREATE TABLE OldTable(Id INT NOT NULL PRIMARY
KEY,Name VARCHAR(100) NOT NULL);
Query OK, 0 rows affected (0.019 sec)

MariaDB [Practice]> CREATE TABLE NewTable(Id INT NOT NULL PRIMARY
KEY,Name VARCHAR(100) NOT NULL);
Query OK, 0 rows affected (0.015 sec)

MariaDB [Practice]> INSERT INTO OldTable(Id,Name) VALUES (1,'John'),
(2,'Jonathan'),(3,'James'),(4,'Jimmy'),(5,'Janine');
Query OK, 5 rows affected (0.004 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [Practice]> SELECT * FROM OldTable;
+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
|  3 | James    |
|  4 | Jimmy    |
|  5 | Janine   |
+----+----------+
5 rows in set (0.000 sec)

MariaDB [Practice]> DELIMITER //
MariaDB [Practice]> CREATE PROCEDURE CursorTry(R INT)
    -> BEGIN
    -> DECLARE N VARCHAR(100);
    -> DECLARE VAR1,F1 INT DEFAULT 0;
    -> DECLARE C1 CURSOR FOR SELECT Id,Name FROM OldTable WHERE Id=R;
    -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET VAR1:=1;
    -> OPEN C1;
    -> LABEL:
    -> LOOP FETCH C1 INTO R,N;
    -> SELECT EXISTS(SELECT Id,Name FROM NewTable WHERE Id=R) INTO F1;
    -> SELECT F1;
    -> IF F1=0 THEN INSERT INTO NewTable VALUES(R,N);
    -> END IF;
    -> IF VAR1=1 THEN CLOSE C1;
    -> LEAVE LABEL;
    -> END IF;
    -> END LOOP;
    -> SELECT * FROM NewTable;
    -> END //
Query OK, 0 rows affected (0.012 sec)



 CALL CursorTry(1);
    -> exit //
+------+
| F1   |
+------+
|    0 |
+------+
1 row in set (0.003 sec)

+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.004 sec)

+----+------+
| Id | Name |
+----+------+
|  1 | John |
+----+------+
1 row in set (0.004 sec)

Query OK, 3 rows affected (0.004 sec)
```

```
DELIMITER ;
MariaDB [Practice]> CALL CursorTry(2);
+------+
| F1   |
+------+
|    0 |
+------+
1 row in set (0.003 sec)

+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.006 sec)

+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
+----+----------+
2 rows in set (0.006 sec)

Query OK, 3 rows affected (0.006 sec)

MariaDB [Practice]> CALL CursorTry(1);
+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.003 sec)

+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.003 sec)

+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
+----+----------+
2 rows in set (0.003 sec)

Query OK, 2 rows affected (0.003 sec)

MariaDB [Practice]> CALL CursorTry(3);
+------+
| F1   |
+------+
|    0 |
+------+
1 row in set (0.001 sec)

+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.002 sec)

+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
|  3 | James    |
+----+----------+
3 rows in set (0.002 sec)

Query OK, 3 rows affected (0.002 sec)

MariaDB [Practice]> CALL CursorTry(4);
+------+
| F1   |
+------+
|    0 |
+------+
1 row in set (0.001 sec)

+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.005 sec)

+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
|  3 | James    |
|  4 | Jimmy    |
+----+----------+
4 rows in set (0.005 sec)

Query OK, 3 rows affected (0.006 sec)

MariaDB [Practice]> CALL CursorTry(3);
+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.001 sec)

+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.001 sec)

+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
|  3 | James    |
|  4 | Jimmy    |
+----+----------+
4 rows in set (0.001 sec)

Query OK, 2 rows affected (0.001 sec)

MariaDB [Practice]> SELECT * FROM NewTable;
+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
|  3 | James    |
|  4 | Jimmy    |
+----+----------+
4 rows in set (0.001 sec)

MariaDB [Practice]> CALL CursorTry(5);
+------+
| F1   |
+------+
|    0 |
+------+
1 row in set (0.001 sec)

+------+
| F1   |
+------+
|    1 |
+------+
1 row in set (0.002 sec)

+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
|  3 | James    |
|  4 | Jimmy    |
|  5 | Janine   |
+----+----------+
5 rows in set (0.002 sec)

Query OK, 3 rows affected (0.002 sec)

MariaDB [Practice]> SELECT * FROM NewTable;
+----+----------+
| Id | Name     |
+----+----------+
|  1 | John     |
|  2 | Jonathan |
|  3 | James    |
|  4 | Jimmy    |
|  5 | Janine   |
+----+----------+
5 rows in set (0.001 sec)
```

Ansh Bhutada
Assignment 7

```
CREATE TABLE Library (
    -> book_id INT AUTO_INCREMENT PRIMARY KEY,
    -> book_title VARCHAR(255),
    -> book_author VARCHAR(255),
    -> publication_year INT,
    -> ISBN VARCHAR(13)
    -> );
Query OK, 0 rows affected (0.061 sec)

MariaDB [Practice]> INSERT INTO Library (book_title, book_author,
publication_year, ISBN) VALUES
    -> ('To Kill a Mockingbird', 'Harper Lee', 1960,
'978-0-06-112008-4'),
    -> ('1984', 'George Orwell', 1949, '978-0-452-28423-4'),
    -> ('Pride and Prejudice', 'Jane Austen', 1813,
'978-0-486-42261-0'),
    -> ('The Great Gatsby', 'F. Scott Fitzgerald', 1925,
'978-0-7432-7356-5'),
    -> ('The Catcher in the Rye', 'J.D. Salinger', 1951,
'978-0-316-76948-0');
ERROR 1406 (22001): Data too long for column 'ISBN' at row 1
MariaDB [Practice]> INSERT INTO Library (book_title, book_author,
publication_year, ISBN) VALUES ('To Kill a Mockingbird', 'Harper Lee',
1960, '9780061120084'), ('1984', 'George Orwell', 1949,
'9780452284234'), ('Pride and Prejudice', 'Jane Austen', 1813,
'9780486422610'), ('The Great Gatsby', 'F. Scott Fitzgerald', 1925,
'9780743273565'), ('The Catcher in the Rye', 'J.D. Salinger', 1951,
'9780316769480');
Query OK, 5 rows affected (0.002 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [Practice]> CREATE TABLE IF NOT EXISTS Library_Audit(
    -> audit_id INT AUTO_INCREMENT PRIMARY KEY,
    -> book_id INT.
    -> book_title VARCHAR(255),
    -> book_author VARCHAR(255),
    -> action_type ENUM('UPDATE','DELETE'),
    -> action_timestamp TIMESTAMP
    -> );
ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right
syntax to use near '.
book_title VARCHAR(255),
book_author VARCHAR(255),
action_type ENUM('UPDATE...' at line 3
MariaDB [Practice]> CREATE TABLE IF NOT EXISTS Library_Audit (
    ->     audit_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     book_id INT,
    ->     book_title VARCHAR(255),
    ->     book_author VARCHAR(255),
    ->     action_type ENUM('UPDATE', 'DELETE'),
    ->     action_timestamp TIMESTAMP
    -> );
Query OK, 0 rows affected (0.025 sec)

MariaDB [Practice]> DELIMITER //
MariaDB [Practice]> CREATE TRIGGER Library_Update_Audit
    -> AFTER UPDATE ON Library
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO
Library_Audit(book_id,book_title,book_author,action_type,action_timestam
p)
    -> VALUES
(OLD.book_id,OLD.book_title,OLD.book_author,'UPDATE',NOW());
    -> END;
    -> //
Query OK, 0 rows affected (0.048 sec)

MariaDB [Practice]> CREATE TRIGGER Library_Delete_Audit
    -> AFTER DELETE ON Library
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO
Library_Audit(book_id,book_title,book_author,action_type,action_timestam
p)
    -> VALUES
(OLD.book_id,OLD.book_title,OLD.book_author,'DELETE',NOW());
    -> END;
    -> //
Query OK, 0 rows affected (0.048 sec)

MariaDB [Practice]> DELIMITER ;
MariaDB [Practice]> DELETE FROM Library WHERE book_id = 2;
Query OK, 1 row affected (0.011 sec)

MariaDB [Practice]> SELECT * FROM Library;
+---------+----------------------+---------------------
+------------------+---------------+
| book_id | book_title           | book_author         |
publication_year | ISBN          |
+---------+-------------------------+---------------------
+------------------+---------------+
|       1 | To Kill a Mockingbird   | Harper Lee          |
1960 | 9780061120084 |
|       3 | Pride and Prejudice     | Jane Austen         |
1813 | 9780486422610 |
|       4 | The Great Gatsby        | F. Scott Fitzgerald |
1925 | 9780743273565 |
|       5 | The Catcher in the Rye | J.D. Salinger        |
1951 | 9780316769480 |
+---------+-------------------------+---------------------
+------------------+---------------+
4 rows in set (0.003 sec)

MariaDB [Practice]> SELECT * FROM Library_Audit;
+----------+---------+------------+---------------+------------
+---------------------+
| audit_id | book_id | book_title | book_author   | action_type |
action_timestamp    |
+----------+---------+------------+---------------+------------
+---------------------+
|        1 |       2 | 1984       | George Orwell | DELETE      |
2023-11-02 19:56:58 |
+----------+---------+------------+---------------+------------
+---------------------+
1 row in set (0.001 sec)

MariaDB [Practice]> UPDATE Library SET publication_year = 1928 WHERE
book_id = 4;
Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [Practice]> SELECT * FROM Library_Audit;
+----------+---------+------------------+---------------------
+-------------+---------------------+
| audit_id | book_id | book_title       | book_author         |
action_type | action_timestamp    |
+----------+---------+------------------+---------------------
+-------------+---------------------+
|        1 |       2 | 1984             | George Orwell       | DELETE
| 2023-11-02 19:56:58 |
|        2 |       4 | The Great Gatsby | F. Scott Fitzgerald | UPDATE
| 2023-11-02 20:10:54 |
+----------+---------+------------------+---------------------
+-------------+---------------------+
2 rows in set (0.001 sec)
```

Ansh Bhutada
Assignment 9

```
db.Products.insertMany([ { name: "HeadPhones", brand: "Sony", price:
100, quantity: 200, discount: 0.15 }, { name: "Keyboard", brand: "i-
ball", price: 45, quantity: 325, discount: 0.10 }])
```

```
db.Products.insertOne( { name: "Monitor", brand: "Dell", price: 175,
quantity: 275, discount: 0.05 } )


db.Products.find({brand: “Dell"})


db.Products.find({ brand: { $ne: "Dell" } })

db.Products.find({ $and: [ { discount: { $gt: 0.10 } }, { brand: { $in:
["Sony", "Dell"] } }] })


db.Products.find({discount: {$gt: 0.17}})

db.Products.find({discount: {$lt: 0.17}})


db.Products.updateMany({}, { $set: { features: [] } });


db.Products.updateOne( { _id: ObjectId("65449a9b0ad66f85680a41f8") },
{ $set: { features: [ 'High-quality sound', 'Noise cancellation',
'Wireless connectivity', 'Comfortable design'] } } );



db.Products.updateMany( { brand: "Dell" }, { $set: { quantity:
300 } } );


db.Products.update( { brand: 'Dell' }, { $set: { discount: 0.1 } },
{ multi: false } );

db.Products.deleteOne( { price: 112 })


db.Products.deleteMany({ $and: [ { price: { $lt: 90 } }, { discount:
{ $gt: 0.20 } }] })



db.Products.find().pretty()

upsert()  not written read from document DBMS on drive
```

Array waali queries:

Dataset :

```
[
{
  _id: ObjectId("65449a9b0ad66f85680a41f9"),
  name: 'Keyboard',
  brand: 'i-ball',
  price: 45,
  quantity: 325,
  discount: 0.1,
  features: [
    'Full-size layout',
    'Quiet keys',
    'Wired connection',
    'Durable build'
  ]
},
{
  _id: ObjectId("65449b030ad66f85680a41fa"),
  name: 'Monitor',
  brand: 'Dell',
  price: 175,
  quantity: 300,
  discount: 0.19,
  features: [
    'Full HD resolution',
    '24-inch screen',
    'HDMI and VGA inputs',
    'Slim design'
  ]
},
{
  _id: ObjectId("65449bfe0ad66f85680a41fc"),
  name: 'CPU',
  brand: 'Dell',
  price: 95,
  quantity: 300,
  discount: 0.14,
  features: [
    'Intel Core i5 processor',
    '8GB RAM',
    '512GB SSD',
    'Compact form factor'
  ]
},
{
  _id: ObjectId("6545f68c777e411b71a6d6fd"),
  name: 'Mouse',
  brand: 'I-Tech',
  price: 25,
  quantity: 325,
  discount: 0.16,
  features: [
    'Wireless connectivity',
    'High-precision sensor',
    'Enorganic shape',
    'Programmable buttons'
  ]
},
  {
    _id: ObjectId("6545f7b3777e411b71a6d6ff"),
    name: 'HeadPhones',
    brand: 'Sony',
    price: 112,
    quantity: 275,
    discount: 0.22,
    features: [
      'High-quality sound',
      'Noise cancellation',
      'Wireless connectivity',
      'Comfortable Design'
    ]
  }
]
```

```
 db.Products.find({_id: ObjectId("6545f7b3777e411b71a6d6ff")},{features:
{$slice: 2}})
[
  {
    _id: ObjectId("6545f7b3777e411b71a6d6ff"),
    name: 'HeadPhones',
    brand: 'Sony',
    price: 112,
    quantity: 275,
    discount: 0.22,
    features: [ 'High-quality sound', 'Noise cancellation' ]
  }
]
```

```
db.Products.aggregate([
... {
... $match: { _id: ObjectId("6545f7b3777e411b71a6d6ff") }},
... {
... $project: { featureCount: { $size: "$features"}}}
... ])
[ { _id: ObjectId("6545f7b3777e411b71a6d6ff"), featureCount: 4 } ]
```

```
 db.Products.findOne({ _id:
ObjectId("65449b030ad66f85680a41fa") }).features
[
  'Full HD resolution',
  '24-inch screen',
  'HDMI and VGA inputs',
  'Slim design'
]
```

```
 db.Products.find({}, { _id: 0, features: 1 })
```

```
[
  {
    features: [
      'Full-size layout',
      'Quiet keys',
      'Wired connection',
      'Durable build'
    ]
  },
  {
    features: [
      'Full HD resolution',
      '24-inch screen',
      'HDMI and VGA inputs',
      'Slim design'
    ]
  },
  {
    features: [
      'Intel Core i5 processor',
      '8GB RAM',
      '512GB SSD',
      'Compact form factor'
    ]
  },
  {
    features: [
      'Wireless connectivity',
      'High-precision sensor',
      'Enorganic shape',
      'Programmable buttons'
    ]
  },
  {
    features: [
      'High-quality sound',
      'Noise cancellation',
      'Wireless connectivity',
      'Comfortable Design'
    ]
  }
]
```

```
db.Products.update(
... { _id: ObjectId("6545f7b3777e411b71a6d6ff") },
... { $push: {features:"Comfortable"}}
... )
DeprecationWarning: Collection.update() is deprecated. Use updateOne,
updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}


{
    _id: ObjectId("6545f7b3777e411b71a6d6ff"),
    name: 'HeadPhones',
    brand: 'Sony',
    price: 112,
    quantity: 275,
    discount: 0.22,
    features: [
      'High-quality sound',
      'Noise cancellation',
      'Wireless connectivity',
      'Comfortable Design',
      'Comfortable'
    ]
  }


db.Products.aggregate([
... {
... $match: { _id: ObjectId("6545f7b3777e411b71a6d6ff")}},
... {
... $project: { featureCount: {$size: "$features"}}}
... ])
[ { _id: ObjectId("6545f7b3777e411b71a6d6ff"), featureCount: 5 } ]




//yeh implement nhi kia tha
db.collection.aggregate([
  { $match: { _id: ObjectId("6545f7b3777e411b71a6d6ff") } },
  { $unwind: "$features" }
])


{
  "_id" : ObjectId("6545f7b3777e411b71a6d6ff"),
  "name" : "HeadPhones",
  "brand" : "Sony",
  "price" : 112,
  "quantity" : 275,
  "discount" : 0.22,
  "features" : "High-quality sound"
}
{
  "_id" : ObjectId("6545f7b3777e411b71a6d6ff"),
  "name" : "HeadPhones",
  "brand" : "Sony",
  "price" : 112,
  "quantity" : 275,
  "discount" : 0.22,
  "features" : "Noise cancellation"
```

```
}
{
  "_id" : ObjectId("6545f7b3777e411b71a6d6ff"),
  "name" : "HeadPhones",
  "brand" : "Sony",
  "price" : 112,
  "quantity" : 275,
  "discount" : 0.22,
  "features" : "Wireless connectivity"
}
{
  "_id" : ObjectId("6545f7b3777e411b71a6d6ff"),
  "name" : "HeadPhones",
  "brand" : "Sony",
  "price" : 112,
  "quantity" : 275,
  "discount" : 0.22,
  "features" : "Comfortable Design"
}
{
  "_id" : ObjectId("6545f7b3777e411b71a6d6ff"),
  "name" : "HeadPhones",
  "brand" : "Sony",
  "price" : 112,
  "quantity" : 275,
  "discount" : 0.22,
  "features" : "Comfortable"
}
```

Ansh Bhutada
Assignment 10

Dataset:

```
[
  {
    _id: ObjectId("65449a9b0ad66f85680a41f9"),
    name: 'Keyboard',
    brand: 'i-ball',
    price: 45,
    quantity: 325,
    discount: 0.1,
    features: [
      'Full-size layout',
      'Quiet keys',
      'Wired connection',
      'Durable build'
    ]
  },
  {
    _id: ObjectId("65449b030ad66f85680a41fa"),
    name: 'Monitor',
    brand: 'Dell',
    price: 175,
    quantity: 300,
    discount: 0.19,
    features: [
      'Full HD resolution',
      '24-inch screen',
      'HDMI and VGA inputs',
      'Slim design'
    ]
  },
  {
    _id: ObjectId("65449bfe0ad66f85680a41fc"),
    name: 'CPU',
    brand: 'Dell',
    price: 95,
    quantity: 300,
    discount: 0.14,
    features: [
      'Intel Core i5 processor',
      '8GB RAM',
      '512GB SSD',
      'Compact form factor'
    ]
  },
  {
    _id: ObjectId("6545f68c777e411b71a6d6fd"),
    name: 'Mouse',
    brand: 'I-Tech',
    price: 25,
    quantity: 325,
    discount: 0.16,
    features: [
      'Wireless connectivity',
      'High-precision sensor',
      'Enorganic shape',
      'Programmable buttons'
    ]
  },
  {
    _id: ObjectId("6545f7b3777e411b71a6d6ff"),
    name: 'HeadPhones',
    brand: 'Sony',
    price: 112,
    quantity: 275,
    discount: 0.22,
    features: [
      'High-quality sound',
      'Noise cancellation',
      'Wireless connectivity',
      'Comfortable Design'
    ]
  }
]
```

Indexing:

Execution stats + simple index

```
db.Products.explain("executionStats").find()
{
  explainVersion: '2',
  queryPlanner: {
    namespace: 'practice.Products',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: 'E475932B',
    planCacheKey: '8AE93992',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      queryPlan: {
        stage: 'COLLSCAN',
        planNodeId: 1,
        filter: {},
        direction: 'forward'
      },
      slotBasedPlan: {
        slots: '$$RESULT=s4 env: { s1 = TimeZoneDatabase(America/
Indiana/Indianapolis...Africa/Libreville) (timeZoneDB), s2 = Nothing
(SEARCH_META), s3 = 1699098599215 (NOW) }',
        stages: '[1] scan s4 s5 none none none none lowPriority []
@"146ba6fc-501e-485c-aa87-47a9c5b76fd1" true false '
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 5,
    executionTimeMillis: 11,
    totalKeysExamined: 0,
    totalDocsExamined: 5,
    executionStages: {
      stage: 'scan',
      planNodeId: 1,
      nReturned: 5,
      executionTimeMillisEstimate: 0,
      opens: 1,
      closes: 1,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      numReads: 5,
      recordSlot: 4,
      recordIdSlot: 5,
      fields: [],
      outputSlots: []
    }
  },
  command: { find: 'Products', filter: {}, '$db': 'practice' },
  serverInfo: {
    host: 'ANSHs-MacBook-Air.local',
    port: 27017,
    version: '7.0.2',
    gitVersion: '02b3c655e1302209ef046da6ba3ef6749dd0b62a'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeEngine'
  },
  ok: 1
}
practice> db.Products.createIndex({brand: 1})
brand_1
practice> db.Products.find({brand: 'I-Tech'}).explain("executionStats")
{
  explainVersion: '2',
  queryPlanner: {
    namespace: 'practice.Products',
    indexFilterSet: false,
    parsedQuery: { brand: { '$eq': 'I-Tech' } },
    queryHash: '454FBA40',
    planCacheKey: '507EDC9D',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      queryPlan: {
        stage: 'FETCH',
        planNodeId: 2,
        inputStage: {
          stage: 'IXSCAN',
          planNodeId: 1,
          keyPattern: { brand: 1 },
          I ndexName: 'brand_1',
          isMultiKey: false,
          multiKeyPaths: { brand: [] },
          isUnique: false,
          isSparse: false,
          isPartial: false,
          indexVersion: 2,
          direction: 'forward',
          indexBounds: { brand: [ '["I-Tech", "I-Tech"]' ] }
        }
      },
      slotBasedPlan: {
        slots: '$$RESULT=s11 env: { s2 = Nothing (SEARCH_META), s1 =
TimeZoneDatabase(America/Indiana/Indianapolis...Africa/Libreville)
```

```
(timeZoneDB), s3 = 1699101520023 (NOW), s6 = KS(3C492D5465636800FE04),
s10 = {"brand" : 1}, s5 = KS(3C492D54656368000104) }',
        stages: '[2] nlj inner [] [s4, s7, s8, s9, s10] \n' +
          '    left \n' +
          '        [1] cfilter {(exists(s5) && exists(s6))} \n' +
          '        [1] ixseek s5 s6 s9 s4 s7 s8 [] @"146ba6fc-501e-485c-
aa87-47a9c5b76fd1" @"brand_1" true \n' +
          '    right \n' +
          '        [2] limit 1 \n' +
          '        [2] seek s4 s11 s12 s7 s8 s9 s10 []
@"146ba6fc-501e-485c-aa87-47a9c5b76fd1" true false \n'
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 15,
    totalKeysExamined: 1,
    totalDocsExamined: 1,
    executionStages: {
      stage: 'nlj',
      planNodeId: 2,
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      opens: 1,
      closes: 1,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      totalDocsExamined: 1,
      totalKeysExamined: 1,
      collectionScans: 0,
      collectionSeeks: 1,
      indexScans: 0,
      indexSeeks: 1,
      indexesUsed: [ 'brand_1' ],
      innerOpens: 1,
      innerCloses: 1,
      outerProjects: [],
      outerCorrelated: [ Long("4"), Long("7"), Long("8"), Long("9"),
Long("10") ],
      outerStage: {
        stage: 'cfilter',
        planNodeId: 1,
        nReturned: 1,
        executionTimeMillisEstimate: 0,
        opens: 1,
        closes: 1,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        numTested: 1,
        filter: '(exists(s5) && exists(s6)) ',
        inputStage: {
```

```
    stage: 'ixseek',
    planNodeId: 1,
    nReturned: 1,
    executionTimeMillisEstimate: 0,
    opens: 1,
    closes: 1,
    saveState: 0,
    restoreState: 0,
    isEOF: 1,
    indexName: 'brand_1',
    keysExamined: 1,
    seeks: 1,
    numReads: 2,
    indexKeySlot: 9,
    recordIdSlot: 4,
    snapshotIdSlot: 7,
    indexIdentSlot: 8,
    outputSlots: [],
    indexKeysToInclude: '00000000000000000000000000000000',
    seekKeyLow: 's5 ',
    seekKeyHigh: 's6 '
  }
},
innerStage: {
  stage: 'limit',
  planNodeId: 2,
  nReturned: 1,
  executionTimeMillisEstimate: 0,
  opens: 1,
  closes: 1,
  saveState: 0,
  restoreState: 0,
  isEOF: 1,
  limit: 1,
  inputStage: {
    stage: 'seek',
    planNodeId: 2,
    nReturned: 1,
    executionTimeMillisEstimate: 0,
    opens: 1,
    closes: 1,
    saveState: 0,
    restoreState: 0,
    isEOF: 0,
    numReads: 1,
    recordSlot: 11,
    recordIdSlot: 12,
    seekKeySlot: 4,
    snapshotIdSlot: 7,
    indexIdentSlot: 8,
    indexKeySlot: 9,
    indexKeyPatternSlot: 10,
    fields: [],
    outputSlots: []
  }
}
```

```
    }
  },
  command: { find: 'Products', filter: { brand: 'I-Tech' }, '$db':
'practice' },
  serverInfo: {
    host: 'ANSHs-MacBook-Air.local',
    port: 27017,
    version: '7.0.2',
    gitVersion: '02b3c655e1302209ef046da6ba3ef6749dd0b62a'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeEngine'
  },
  ok: 1
}
```

Execution stats + complex index

```
 db.products.createIndex({ name: 1, price: 1})
name_1_price_1
practice> db.Products.find({name : 'Keyboard' , price: { $gte:
45} }).explain("executionStats")
{
  explainVersion: '2',
  queryPlanner: {
    namespace: 'practice.Products',
    indexFilterSet: false,
    parsedQuery: {
      '$and': [ { name: { '$eq': 'Keyboard' } }, { price: { '$gte': 45 }
} ]
    },
    queryHash: '99266D5C',
    planCacheKey: 'A440F309',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      queryPlan: {
        stage: 'COLLSCAN',
        planNodeId: 1,
        filter: {
          '$and': [
            { name: { '$eq': 'Keyboard' } },
            { price: { '$gte': 45 } }
          ]
        },
        direction: 'forward'
      },
      slotBasedPlan: {
        slots: '$$RESULT=s6 env: { s2 = Nothing (SEARCH_META), s1 =
TimeZoneDatabase(America/Indiana/Indianapolis...Africa/Libreville)
(timeZoneDB), s3 = 1699102857301 (NOW), s8 = "Keyboard", s9 = 45 }',
        stages: '[1] filter {(traverseF(s4, lambda(l1.0) { ((l1.0 == s8)
?: false) }, false) && traverseF(s5, lambda(l2.0) { ((l2.0 >= s9) ?:
false) }, false))} \n' +
          '[1] scan s6 s7 none none none none lowPriority [s4 = name, s5
= price] @"146ba6fc-501e-485c-aa87-47a9c5b76fd1" true false '
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 4,
    totalKeysExamined: 0,
    totalDocsExamined: 5,
    executionStages: {
      stage: 'filter',
      planNodeId: 1,
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      opens: 1,
      closes: 1,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      numTested: 5,
      filter: '(traverseF(s4, lambda(l1.0) { ((l1.0 == s8) ?: false) },
false) && traverseF(s5, lambda(l2.0) { ((l2.0 >= s9) ?: false) },
false)) ',
      inputStage: {
        stage: 'scan',
        planNodeId: 1,
        nReturned: 5,
        executionTimeMillisEstimate: 0,
        opens: 1,
        closes: 1,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        numReads: 5,
        recordSlot: 6,
        recordIdSlot: 7,
        fields: [ 'name', 'price' ],
        outputSlots: [ Long("4"), Long("5") ]
      }
    }
  },
  command: {
    find: 'Products',
    filter: { name: 'Keyboard', price: { '$gte': 45 } },
    '$db': 'practice'
  },
  serverInfo: {
    host: 'ANSHs-MacBook-Air.local',
    port: 27017,
    version: '7.0.2',
    gitVersion: '02b3c655e1302209ef046da6ba3ef6749dd0b62a'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeEngine'
  },
  ok: 1
}
practice> db.Products.find({name : 'Mouse' , price:
25 }).explain("executionStats")
{
  explainVersion: '2',
  queryPlanner: {
    namespace: 'practice.Products',
    indexFilterSet: false,
    parsedQuery: {
      '$and': [ { name: { '$eq': 'Mouse' } }, { price: { '$eq': 25 } } ]
    },
    queryHash: 'EA7C602B',
    planCacheKey: '85961E02',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      queryPlan: {
        stage: 'COLLSCAN',
        planNodeId: 1,
        filter: {
          '$and': [ { name: { '$eq': 'Mouse' } }, { price: { '$eq': 25 }
} ]
        },
        direction: 'forward'
      },
      slotBasedPlan: {
        slots: '$$RESULT=s6 env: { s9 = 25, s3 = 1699102979839 (NOW), s2
= Nothing (SEARCH_META), s1 = TimeZoneDatabase(America/Indiana/
Indianapolis...Africa/Libreville) (timeZoneDB), s8 = "Mouse" }',
        stages: '[1] filter {(traverseF(s4, lambda(l1.0) { ((l1.0 == s8)
?: false) }, false) && traverseF(s5, lambda(l2.0) { ((l2.0 == s9) ?:
false) }, false))} \n' +
          '[1] scan s6 s7 none none none none lowPriority [s4 = name, s5
= price] @"146ba6fc-501e-485c-aa87-47a9c5b76fd1" true false '
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 5,
    executionStages: {
      stage: 'filter',
      planNodeId: 1,
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      opens: 1,
      closes: 1,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      numTested: 5,
      filter: '(traverseF(s4, lambda(l1.0) { ((l1.0 == s8) ?: false) },
false) && traverseF(s5, lambda(l2.0) { ((l2.0 == s9) ?: false) },
false)) ',
      inputStage: {
        stage: 'scan',
        planNodeId: 1,
        nReturned: 5,
        executionTimeMillisEstimate: 0,
        opens: 1,
        closes: 1,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        numReads: 5,
        recordSlot: 6,
        recordIdSlot: 7,
        fields: [ 'name', 'price' ],
        outputSlots: [ Long("4"), Long("5") ]
      }
    }
  },
  command: {
    find: 'Products',
    filter: { name: 'Mouse', price: 25 },
    '$db': 'practice'
  },
  serverInfo: {
    host: 'ANSHs-MacBook-Air.local',
    port: 27017,
    version: '7.0.2',
    gitVersion: '02b3c655e1302209ef046da6ba3ef6749dd0b62a'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeEngine'
  },
  ok: 1
}
```

Execution Stats + unique Index

```
practice> db.products.createIndex({ name: 1 }, { unique: true })
name_1
practice> db.Products.find({name:
'HeadPhones'}).explain("executionStats")
{
  explainVersion: '2',
  queryPlanner: {
    namespace: 'practice.Products',
    indexFilterSet: false,
    parsedQuery: { name: { '$eq': 'HeadPhones' } },
    queryHash: '1AD872C6',
    planCacheKey: 'B630900B',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      queryPlan: {
        stage: 'COLLSCAN',
        planNodeId: 1,
        filter: { name: { '$eq': 'HeadPhones' } },
        direction: 'forward'
      },
      slotBasedPlan: {
        slots: '$$RESULT=s5 env: { s2 = Nothing (SEARCH_META), s1 =
TimeZoneDatabase(America/Indiana/Indianapolis...Africa/Libreville)
(timeZoneDB), s3 = 1699103246210 (NOW), s7 = "HeadPhones" }',
        stages: '[1] filter {traverseF(s4, lambda(l1.0) { ((l1.0 ==
s7) ?: false) }, false)} \n' +
          '[1] scan s5 s6 none none none none lowPriority [s4 = name]
@"146ba6fc-501e-485c-aa87-47a9c5b76fd1" true false '
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 1,
    totalKeysExamined: 0,
    totalDocsExamined: 5,
    executionStages: {
      stage: 'filter',
      planNodeId: 1,
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      opens: 1,
      closes: 1,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      numTested: 5,
      filter: 'traverseF(s4, lambda(l1.0) { ((l1.0 == s7) ?: false) },
false) ',
      inputStage: {
        stage: 'scan',
        planNodeId: 1,
        nReturned: 5,
        executionTimeMillisEstimate: 0,
        opens: 1,
        closes: 1,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        numReads: 5,
        recordSlot: 5,
        recordIdSlot: 6,
        fields: [ 'name' ],
        outputSlots: [ Long("4") ]
      }
    }
  },
  command: {
    find: 'Products',
    filter: { name: 'HeadPhones' },
    '$db': 'practice'
  },
  serverInfo: {
    host: 'ANSHs-MacBook-Air.local',
    port: 27017,
    version: '7.0.2',
    gitVersion: '02b3c655e1302209ef046da6ba3ef6749dd0b62a'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeEngine'
  },
  ok: 1
}
```

Returns simple index only

```
practice> db.Products.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { brand: 1 }, name: 'brand_1' }
]
```

Aggregation:

```
 db.Products.aggregate([
...   {
...     $group: {
...       _id: "$brand",
...       averagePrice: {
...         $avg: "$price"
...       }
...     }
...   },
...   {
...     $sort: {
...       averagePrice: -1
...     }
...   }
... ])
[
  { _id: 'Dell', averagePrice: 135 },
  { _id: 'Sony', averagePrice: 112 },
  { _id: 'i-ball', averagePrice: 45 },
  { _id: 'I-Tech', averagePrice: 25 }
]
```

I tried to calculate Total price like price*quantity*(1-discount)

```
db.Products.aggregate([
...   {
...     $addFields: {
...       totalPrice: {
...         $multiply: [
...           "$price",
...           "$quantity",
...           { $subtract: [1, "$discount"] }
...         ]
...       }
...     }
...   },
...   {
...     $group: {
...       _id: "$name",
...       total: { $sum: "$totalPrice" }
...     }
...   },
...   {
```

```
...     $sort: {
...       total: -1
...     }
...   }
... ])
[
  { _id: 'Monitor', total: 42525 },
  { _id: 'CPU', total: 24510 },
  { _id: 'HeadPhones', total: 24024 },
  { _id: 'Keyboard', total: 13162.5 },
  { _id: 'Mouse', total: 6825 }
]
```

Skip

```
db.Products.aggregate([
... {
...   $project: {
...      name: 1,
...      price: 1,
...      brand: 1
...    }
... },
... {
...   $skip: 2
... }
... ])
[
  {
    _id: ObjectId("65449bfe0ad66f85680a41fc"),
    name: 'CPU',
    brand: 'Dell',
    price: 95
  },
  {
    _id: ObjectId("6545f68c777e411b71a6d6fd"),
    name: 'Mouse',
    brand: 'I-Tech',
    price: 25
  },
  {
    _id: ObjectId("6545f7b3777e411b71a6d6ff"),
    name: 'HeadPhones',
    brand: 'Sony',
    price: 112
  }
]
```

Without skip

```
db.Products.aggregate([ { $project: { name: 1, price: 1, brand: 1 } }] )
[
  {
    _id: ObjectId("65449a9b0ad66f85680a41f9"),
    name: 'Keyboard',
    brand: 'i-ball',
    price: 45
  },
  {
    _id: ObjectId("65449b030ad66f85680a41fa"),
    name: 'Monitor',
    brand: 'Dell',
    price: 175
  },
  {
    _id: ObjectId("65449bfe0ad66f85680a41fc"),
    name: 'CPU',
    brand: 'Dell',
    price: 95
  },
  {
    _id: ObjectId("6545f68c777e411b71a6d6fd"),
    name: 'Mouse',
    brand: 'I-Tech',
    price: 25
  },
  {
    _id: ObjectId("6545f7b3777e411b71a6d6ff"),
    name: 'HeadPhones',
    brand: 'Sony',
    price: 112
  }
]
```

Limit

```
 db.Products.aggregate([
... {
...   $match: { price: { $gt: 40 } }
... },
... {
...   $limit: 3
... }
... ])
[
  {
    _id: ObjectId("65449a9b0ad66f85680a41f9"),
    name: 'Keyboard',
    brand: 'i-ball',
    price: 45,
    quantity: 325,
    discount: 0.1,
    features: [
      'Full-size layout',
      'Quiet keys',
      'Wired connection',
      'Durable build'
    ]
  },
  {
    _id: ObjectId("65449b030ad66f85680a41fa"),
    name: 'Monitor',
    brand: 'Dell',
    price: 175,
    quantity: 300,
    discount: 0.19,
    features: [
      'Full HD resolution',
      '24-inch screen',
      'HDMI and VGA inputs',
      'Slim design'
    ]
  },
  {
    _id: ObjectId("65449bfe0ad66f85680a41fc"),
    name: 'CPU',
    brand: 'Dell',
    price: 95,
    quantity: 300,
    discount: 0.14,
    features: [
      'Intel Core i5 processor',
      '8GB RAM',
      '512GB SSD',
      'Compact form factor'
    ]
  }
]
```

Count

```
db.Products.aggregate([
... {
... $match: { discount: { $gt: 0.15 } }
... },
... {
... $count: "totalProducts_WithDiscountGreaterThen_15_percent"
... }
... ])
[ { totalProducts_WithDiscountGreaterThen_15_percent: 3 } ]
```

Count

```
db.Products.find({ brand: "Dell"}).count()
2
```

Distinct

```
db.Products.distinct("brand")
```

```
[ 'Dell', 'I-Tech', 'Sony', 'i-ball' ]

distinctBrands = db.Products.distinct("brand");
[ 'Dell', 'I-Tech', 'Sony', 'i-ball' ]
practice> distinctBrandsCount = distinctBrands.length;
4
```

Ansh Bhutada
Assignment 11

```
 var mapFunction = function() {
... emit(this.name,this.discount * this.quantity * this.price);
... }

practice> var reduceFunction = function(key,values) {
... return Array.sum(values);
... }


 db.Products.mapReduce( mapFunction, reduceFunction, { out:
"Name_Discount_Value" } )

{ result: 'Name_Discount_Value', ok: 1 }

db.Name_Discount_Value.find().pretty()
[
  { _id: 'Monitor', value: 9975 },
  { _id: 'HeadPhones', value: 6776 },
  { _id: 'Keyboard', value: 1462.5 },
  { _id: 'Mouse', value: 1300 },
  { _id: 'CPU', value: 3990.0000000000005 }
]
```

```
var mapFunction = function() {
... if(this.price > 40) {
... emit(this.name,this.price);
... }
... }

var reduceFunction = function(key,values) {
... return Array.sum(values)
... }


db.Products.mapReduce( mapFunction, reduceFunction, { query: { price:
{$gt: 40}}, out: "brand_price_gt_40" } )
{ result: 'brand_price_gt_40', ok: 1 }
```

```
 db.brand_price_gt_40.find().pretty()
[
  { _id: 'Keyboard', value: 45 },
  { _id: 'CPU', value: 95 },
  { _id: 'Monitor', value: 175 },
  { _id: 'HeadPhones', value: 112 }
]
```

```
var mapFunction = function() {
... emit(this.brand, {
... count: 1,
... total: this.price
... })
... }

practice> var reduceFunction = function(key,values) {
... var reduced =
... {
... count: 0,
... total: 0
... }
... for(var i=0;i<values.length;i++){
... reduced.count += values[i].count;
... reduced.total += values[i].total;
... }
... if(reduced.count > 0) {
... reduced.average = reduced.total / reduced.count;
... }
... return reduced
... }
```

```
db.Products.mapReduce( mapFunction, reduceFunction, { out:
"brand_average_price" } )
{ result: 'brand_average_price', ok: 1 }

 db.brand_average_price.find().pretty()
[
  { _id: 'i-ball', value: { count: 1, total: 45, average: 45 } },
  { _id: 'Dell', value: { count: 2, total: 270, average: 135 } },
  { _id: 'I-Tech', value: { count: 1, total: 25, average: 25 } },
  { _id: 'Sony', value: { count: 1, total: 112, average: 112 } }
]
```

Ansh Bhutada
Assignment 8

```
import java.sql.*;
import java.util.*;

public class Main {

    public static void create_table(String table_name) {
        Connection connection = null;
        String url = "jdbc:mariadb://10.10.13.128:3306/31111_db";
        String user = "te31111";
        String pass = "te31111";

        try {
            connection = DriverManager.getConnection(url, user, pass);
        } catch (SQLException e) {
            e.printStackTrace();
        }

        try {
            String sql = "create table " + table_name + "(" + "roll_no
int primary key," + "name varchar(25))";
            PreparedStatement ps = connection.prepareStatement(sql);
            ps.executeUpdate();
            System.out.println("Created table successfully");
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void insert_data(int roll_no, String name, String
table_name) {
        Connection connection = null;
        String url = "jdbc:mariadb://10.10.13.128:3306/31111_db";
        String user = "te31111";
        String pass = "te31111";

        try {
            connection = DriverManager.getConnection(url, user, pass);
        } catch (SQLException e) {
            e.printStackTrace();
        }

        try {
            String sql = "insert into " + table_name + " values (" +
roll_no + ", '" + name + "')";
            PreparedStatement ps = connection.prepareStatement(sql);
            ps.executeUpdate();
            System.out.println("Inserted successfully");
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void delete_data(int id, String table_name) {
        Connection connection = null;
        String url = "jdbc:mariadb://10.10.13.128:3306/31111_db";
        String user = "te31111";
        String pass = "te31111";

        try {
            connection = DriverManager.getConnection(url, user, pass);
        } catch (SQLException e) {
            e.printStackTrace();
        }

        try {
            String sql = "delete from " + table_name + " where roll_no =
" + id + ";";
            PreparedStatement ps = connection.prepareStatement(sql);
            ps.executeUpdate();
            System.out.println("Deleted successfully");
        } catch (SQLException e) {
            System.out.println(e);
        }
    }

    public static void main(String[] args) {
        Connection connection = null;
        String url = "jdbc:mariadb://10.10.13.128:3306/31111_db";
        String user = "te31111";
        String pass = "te31111";

        try {
            connection = DriverManager.getConnection(url, user, pass);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        System.out.println("Successfully Connected");

        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("Menu:");
            System.out.println("1. Create Table");
            System.out.println("2. Insert Data");
            System.out.println("3. Delete Data");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");

            choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character
            switch (choice) {
                case 1:
                    System.out.print("Enter table name: ");
                    String tableName = scanner.nextLine();
                    create_table(tableName);
                    break;
                case 2:
                    System.out.print("Enter roll number: ");
                    int rollNo = scanner.nextInt();
                    scanner.nextLine(); // Consume the newline character
                    System.out.print("Enter name: ");
                    String name = scanner.nextLine();
                    System.out.print("Enter table name: ");
                    String insertTableName = scanner.nextLine();
                    insert_data(rollNo, name, insertTableName);
                    break;
                case 3:
                    System.out.print("Enter ID to delete: ");
                    int idToDelete = scanner.nextInt();
                    scanner.nextLine(); // Consume the newline character
                    System.out.print("Enter table name: ");
                    String deleteTableName = scanner.nextLine();
                    delete_data(idToDelete, deleteTableName);
                    break;
                case 4:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice. Please select a
valid option.");
            }
        } while (choice != 4);

        scanner.close();
    }
}
```

Ansh Bhutada
Assignment 12

```
import java.util.*;
import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;
import com.mongodb.client.model.Updates;
import org.bson.*;

public class Assignment12 {

    public static void createDocument(MongoCollection<Document>
collection) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the name to insert: ");
        String name = scanner.nextLine();
        Document doc = new Document("name", name);
        collection.insertOne(doc);
        System.out.println("Document created successfully.");
    }

    public static void readDocuments(MongoCollection<Document>
collection) {
        System.out.println("Reading documents:");
        FindIterable<Document> documents = collection.find();
        for (Document doc : documents) {
            System.out.println(doc);
        }
    }

    public static void updateDocument(MongoCollection<Document>
collection) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the name to update: ");
        String nameToUpdate = scanner.nextLine();
        System.out.print("Enter the new name: ");
        String newName = scanner.nextLine();

        collection.updateOne(Filters.eq("name", nameToUpdate),
Updates.set("name", newName));
        System.out.println("Document updated successfully.");
    }

    public static void deleteDocument(MongoCollection<Document>
collection) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the name to delete: ");
        String nameToDelete = scanner.nextLine();

        collection.deleteMany(Filters.eq("name", nameToDelete));
        System.out.println("Document(s) deleted successfully.");
    }

    public static void main(String[] args) {
        System.out.println("Hello World");

        MongoClientURI connUri = new MongoClientURI("mongodb://
10.10.10.176");
        MongoClient mongoClient = new MongoClient(connUri);
        MongoDatabase db = mongoClient.getDatabase("31112_db");
        MongoCollection<Document> collection =
db.getCollection("connectivity_test");

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Create Document");
            System.out.println("2. Read Documents");
            System.out.println("3. Update Document");
            System.out.println("4. Delete Document");
            System.out.println("5. Exit");

            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    createDocument(collection);
                    break;
                case 2:
                    readDocuments(collection);
                    break;
                case 3:
                    updateDocument(collection);
                    break;
                case 4:
                    deleteDocument(collection);
                    break;
                case 5:
                    mongoClient.close();
                    System.out.println("Goodbye!");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please try
again.");
            }
        }
    }
}
```