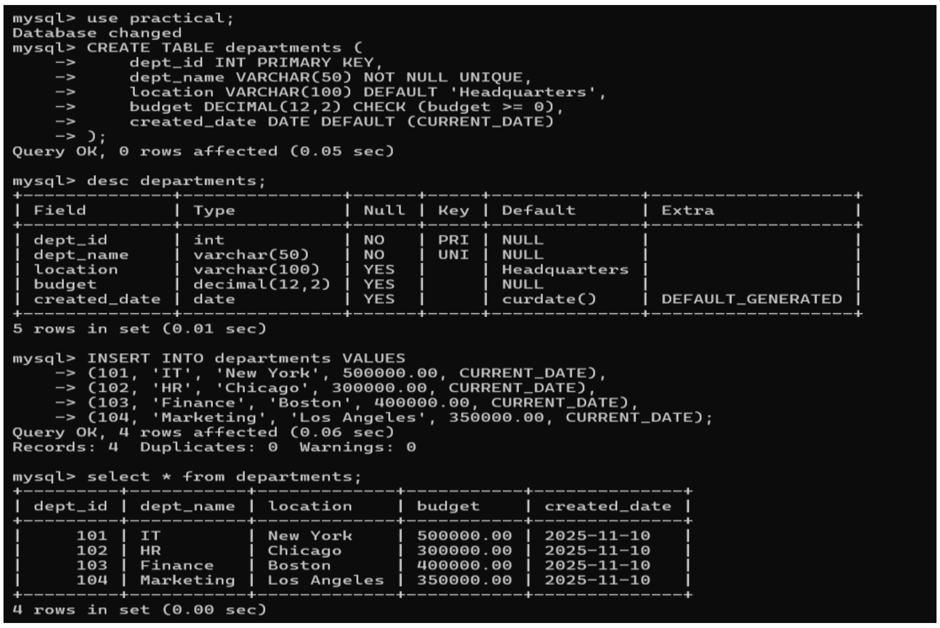
# 2A

SQL Queries:

aDesign and Develop SQLDDL statements which demonstrate the use of SQL objects suchas Table, View, Index, Sequence, Synonym, different constraints etc.

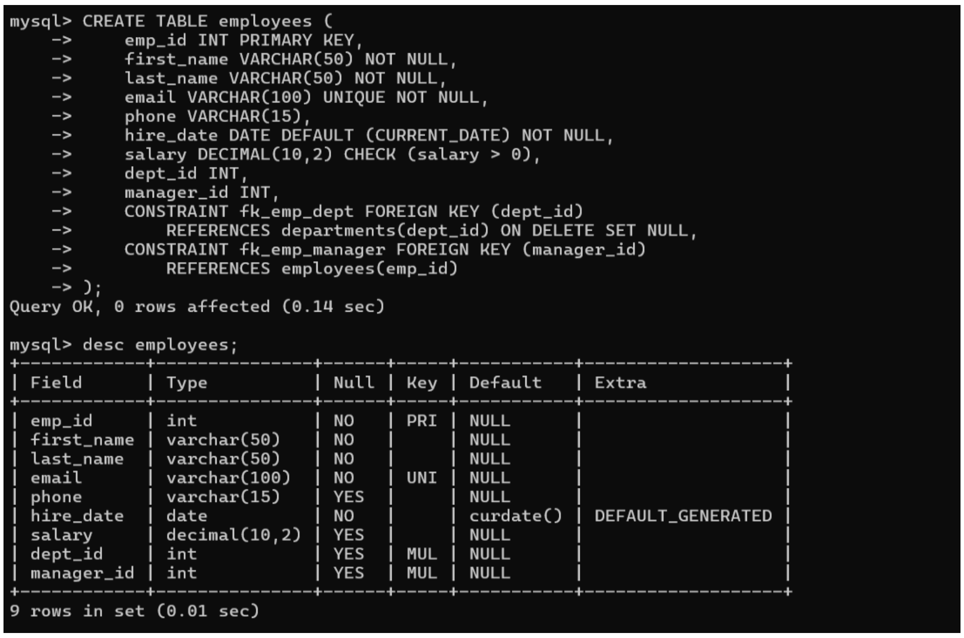
```
-- 1. CREATE TABLE with Constraints
CREATE TABLE departments (
    dept_id INT AUTO_INCREMENT PRIMARY KEY,
    dept_name VARCHAR(50) NOT NULL,
    location VARCHAR(100) DEFAULT 'Headquarters',
    budget DECIMAL(12,2) CHECK (budget >= 0),
    created_date DATE DEFAULT (CURRENT_DATE)
);
```



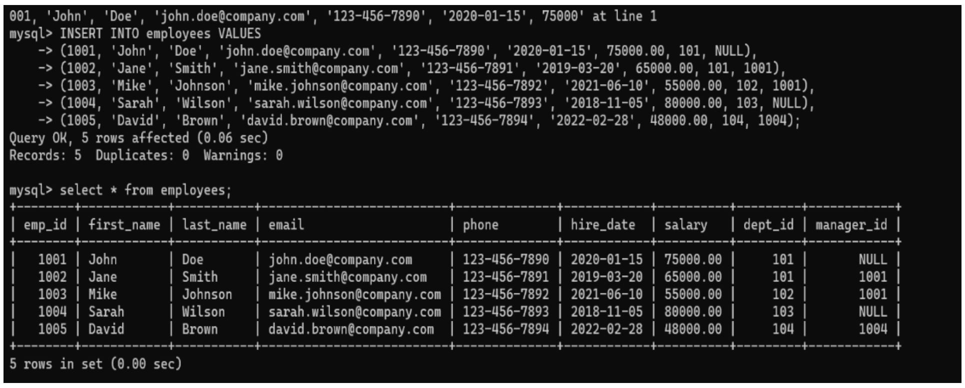
```
mysql> use practical;
Database changed
mysql> CREATE TABLE departments (
    ->     dept_id INT PRIMARY KEY,
    ->     dept_name VARCHAR(50) NOT NULL UNIQUE,
    ->     location VARCHAR(100) DEFAULT 'Headquarters',
    ->     budget DECIMAL(12,2) CHECK (budget >= 0),
    ->     created_date DATE DEFAULT (CURRENT_DATE)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc departments;
+--------------+---------------+------+-----+--------------+-------------------+
| Field        | Type          | Null | Key | Default      | Extra             |
+--------------+---------------+------+-----+--------------+-------------------+
| dept_id      | int           | NO   | PRI | NULL         |                   |
| dept_name    | varchar(50)   | NO   | UNI | NULL         |                   |
| location     | varchar(100)  | YES  |     | Headquarters |                   |
| budget       | decimal(12,2) | YES  |     | NULL         |                   |
| created_date | date          | YES  |     | curdate()    | DEFAULT_GENERATED |
+--------------+---------------+------+-----+--------------+-------------------+
5 rows in set (0.01 sec)

mysql> INSERT INTO departments VALUES
    -> (101, 'IT', 'New York', 500000.00, CURRENT_DATE),
    -> (102, 'HR', 'Chicago', 300000.00, CURRENT_DATE),
    -> (103, 'Finance', 'Boston', 400000.00, CURRENT_DATE),
    -> (104, 'Marketing', 'Los Angeles', 350000.00, CURRENT_DATE);
Query OK, 4 rows affected (0.06 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from departments;
+---------+-----------+-------------+-----------+--------------+
| dept_id | dept_name | location    | budget    | created_date |
+---------+-----------+-------------+-----------+--------------+
|     101 | IT        | New York    | 500000.00 | 2025-11-10   |
|     102 | HR        | Chicago     | 300000.00 | 2025-11-10   |
|     103 | Finance   | Boston      | 400000.00 | 2025-11-10   |
|     104 | Marketing | Los Angeles | 350000.00 | 2025-11-10   |
+---------+-----------+-------------+-----------+--------------+
4 rows in set (0.00 sec)
```

```
-- 2. CREATE TABLE with Foreign Key
CREATE TABLE employees (
    emp_id NUMBER(8) PRIMARY KEY,
    first_name VARCHAR2(50) NOT NULL,
    last_name VARCHAR2(50) NOT NULL,
    email VARCHAR2(100) UNIQUE NOT NULL,
    phone VARCHAR2(15),
    hire_date DATE DEFAULT SYSDATE NOT NULL,
    salary NUMBER(10,2) CHECK (salary > 0),
    dept_id NUMBER(5),
    manager_id NUMBER(8),
    CONSTRAINT fk_emp_dept FOREIGN KEY (dept_id)
```

```
        REFERENCES departments(dept_id) ON DELETE SET NULL,
    CONSTRAINT fk_emp_manager FOREIGN KEY (manager_id)  REFERENCES employees(emp_id));
```

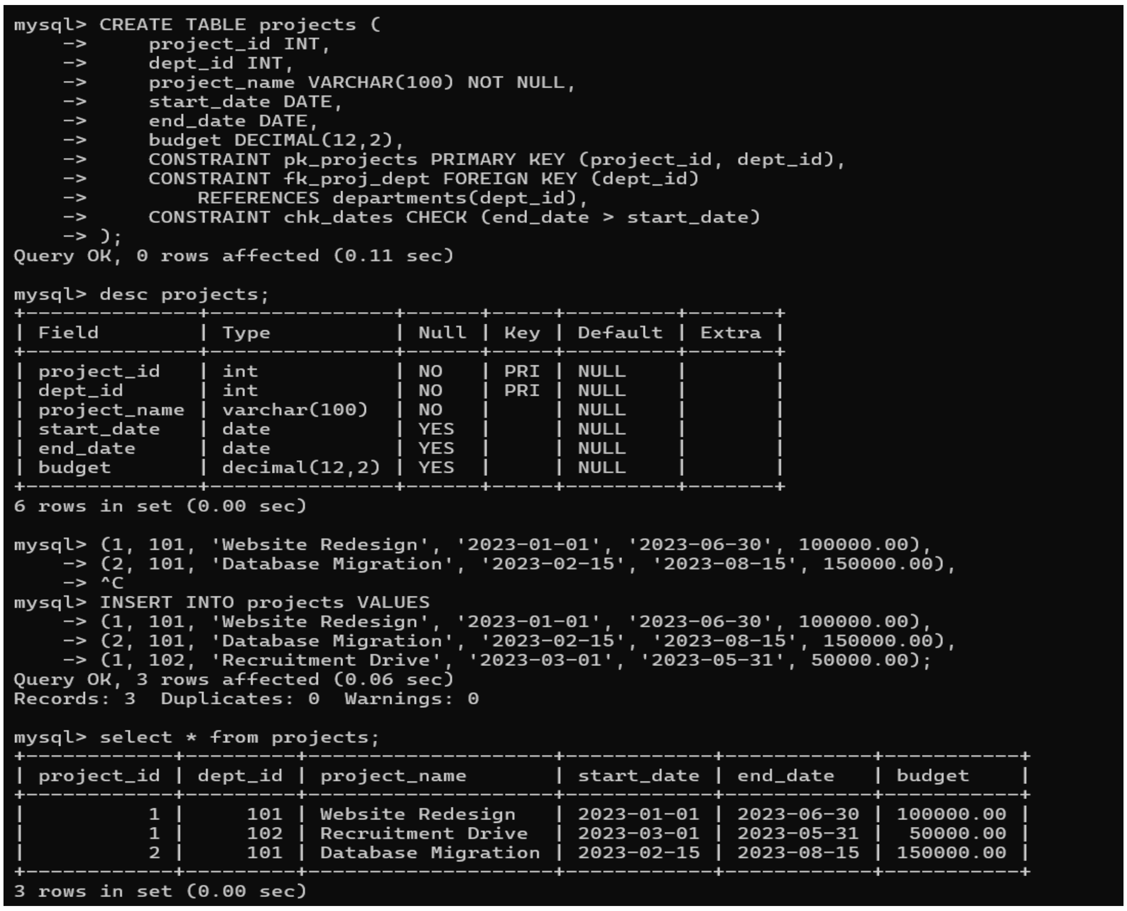


```
mysql> CREATE TABLE employees (
    ->     emp_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50) NOT NULL,
    ->     last_name VARCHAR(50) NOT NULL,
    ->     email VARCHAR(100) UNIQUE NOT NULL,
    ->     phone VARCHAR(15),
    ->     hire_date DATE DEFAULT (CURRENT_DATE) NOT NULL,
    ->     salary DECIMAL(10,2) CHECK (salary > 0),
    ->     dept_id INT,
    ->     manager_id INT,
    ->     CONSTRAINT fk_emp_dept FOREIGN KEY (dept_id)
    ->         REFERENCES departments(dept_id) ON DELETE SET NULL,
    ->     CONSTRAINT fk_emp_manager FOREIGN KEY (manager_id)
    ->         REFERENCES employees(emp_id)
    -> );
Query OK, 0 rows affected (0.14 sec)

mysql> desc employees;
+------------+---------------+------+-----+-----------+-------------------+
| Field      | Type          | Null | Key | Default   | Extra             |
+------------+---------------+------+-----+-----------+-------------------+
| emp_id     | int           | NO   | PRI | NULL      |                   |
| first_name | varchar(50)   | NO   |     | NULL      |                   |
| last_name  | varchar(50)   | NO   |     | NULL      |                   |
| email      | varchar(100)  | NO   | UNI | NULL      |                   |
| phone      | varchar(15)   | YES  |     | NULL      |                   |
| hire_date  | date          | NO   |     | curdate() | DEFAULT_GENERATED |
| salary     | decimal(10,2) | YES  |     | NULL      |                   |
| dept_id    | int           | YES  | MUL | NULL      |                   |
| manager_id | int           | YES  | MUL | NULL      |                   |
+------------+---------------+------+-----+-----------+-------------------+
9 rows in set (0.01 sec)
```



```
001, 'John', 'Doe', 'john.doe@company.com', '123-456-7890', '2020-01-15', 75000' at line 1
mysql> INSERT INTO employees VALUES
    -> (1001, 'John', 'Doe', 'john.doe@company.com', '123-456-7890', '2020-01-15', 75000.00, 101, NULL),
    -> (1002, 'Jane', 'Smith', 'jane.smith@company.com', '123-456-7891', '2019-03-20', 65000.00, 101, 1001),
    -> (1003, 'Mike', 'Johnson', 'mike.johnson@company.com', '123-456-7892', '2021-06-10', 55000.00, 102, 1001),
    -> (1004, 'Sarah', 'Wilson', 'sarah.wilson@company.com', '123-456-7893', '2018-11-05', 80000.00, 103, NULL),
    -> (1005, 'David', 'Brown', 'david.brown@company.com', '123-456-7894', '2022-02-28', 48000.00, 104, 1004);
Query OK, 5 rows affected (0.06 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from employees;
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
| emp_id | first_name | last_name | email                    | phone        | hire_date  | salary   | dept_id | manager_id |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
|   1001 | John       | Doe       | john.doe@company.com     | 123-456-7890 | 2020-01-15 | 75000.00 |     101 |       NULL |
|   1002 | Jane       | Smith     | jane.smith@company.com   | 123-456-7891 | 2019-03-20 | 65000.00 |     101 |       1001 |
|   1003 | Mike       | Johnson   | mike.johnson@company.com | 123-456-7892 | 2021-06-10 | 55000.00 |     102 |       1001 |
|   1004 | Sarah      | Wilson    | sarah.wilson@company.com | 123-456-7893 | 2018-11-05 | 80000.00 |     103 |       NULL |
|   1005 | David      | Brown     | david.brown@company.com  | 123-456-7894 | 2022-02-28 | 48000.00 |     104 |       1004 |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
5 rows in set (0.00 sec)
```

```
-- 3. CREATE TABLE with Composite Primary Key
CREATE TABLE projects (
    project_id NUMBER(6),
    dept_id NUMBER(5),
    project_name VARCHAR2(100) NOT NULL,
    start_date DATE,
    end_date DATE,
    budget NUMBER(12,2),
    CONSTRAINT pk_projects PRIMARY KEY (project_id, dept_id),
    CONSTRAINT fk_proj_dept FOREIGN KEY (dept_id)
        REFERENCES departments(dept_id),
    CONSTRAINT chk_dates CHECK (end_date > start_date)
);
```



```
mysql> CREATE TABLE projects (
    ->     project_id INT,
    ->     dept_id INT,
    ->     project_name VARCHAR(100) NOT NULL,
    ->     start_date DATE,
    ->     end_date DATE,
    ->     budget DECIMAL(12,2),
    ->     CONSTRAINT pk_projects PRIMARY KEY (project_id, dept_id),
    ->     CONSTRAINT fk_proj_dept FOREIGN KEY (dept_id)
    ->         REFERENCES departments(dept_id),
    ->     CONSTRAINT chk_dates CHECK (end_date > start_date)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> desc projects;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| project_id   | int           | NO   | PRI | NULL    |       |
| dept_id      | int           | NO   | PRI | NULL    |       |
| project_name | varchar(100)  | NO   |     | NULL    |       |
| start_date   | date          | YES  |     | NULL    |       |
| end_date     | date          | YES  |     | NULL    |       |
| budget       | decimal(12,2) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> (1, 101, 'Website Redesign', '2023-01-01', '2023-06-30', 100000.00),
    -> (2, 101, 'Database Migration', '2023-02-15', '2023-08-15', 150000.00),
    -> ^C
mysql> INSERT INTO projects VALUES
    -> (1, 101, 'Website Redesign', '2023-01-01', '2023-06-30', 100000.00),
    -> (2, 101, 'Database Migration', '2023-02-15', '2023-08-15', 150000.00),
    -> (1, 102, 'Recruitment Drive', '2023-03-01', '2023-05-31', 50000.00);
Query OK, 3 rows affected (0.06 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from projects;
+------------+---------+--------------------+------------+------------+-----------+
| project_id | dept_id | project_name       | start_date | end_date   | budget    |
+------------+---------+--------------------+------------+------------+-----------+
|          1 |     101 | Website Redesign   | 2023-01-01 | 2023-06-30 | 100000.00 |
|          1 |     102 | Recruitment Drive  | 2023-03-01 | 2023-05-31 |  50000.00 |
|          2 |     101 | Database Migration | 2023-02-15 | 2023-08-15 | 150000.00 |
+------------+---------+--------------------+------------+------------+-----------+
3 rows in set (0.00 sec)
```

-- 4. CREATE SEQUENCE

CREATE TABLE sequences (

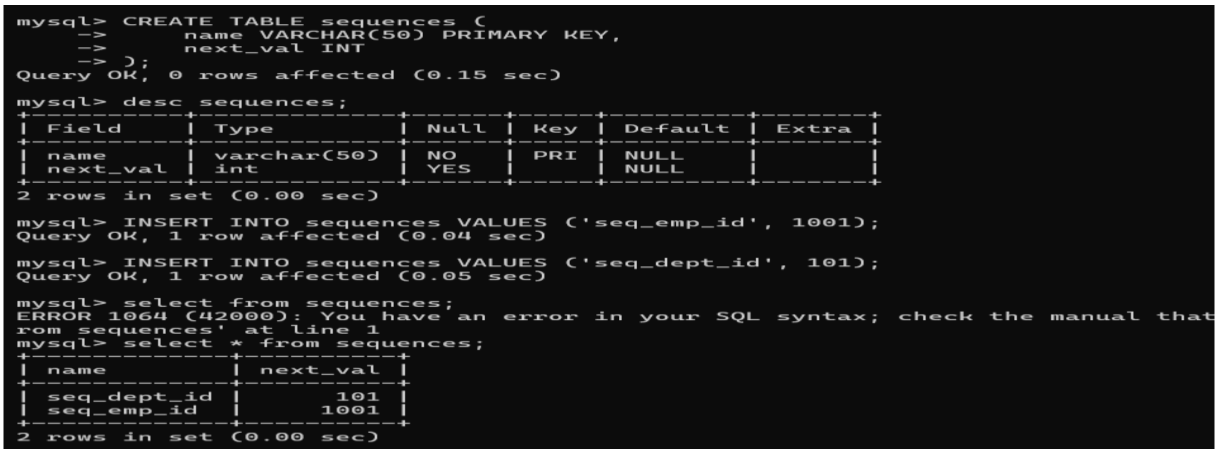
    name VARCHAR(50) PRIMARY KEY,

    next_val INT

);

INSERT INTO sequences VALUES ('seq_emp_id', 1001);

INSERT INTO sequences VALUES ('seq_dept_id', 101);



```
mysql> CREATE TABLE sequences (
    ->     name VARCHAR(50) PRIMARY KEY,
    ->     next_val INT
    -> );
Query OK, 0 rows affected (0.15 sec)

mysql> desc sequences;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| name     | varchar(50) | NO   | PRI | NULL    |       |
| next_val | int         | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> INSERT INTO sequences VALUES ('seq_emp_id', 1001);
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO sequences VALUES ('seq_dept_id', 101);
Query OK, 1 row affected (0.05 sec)

mysql> select from sequences;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
rom sequences' at line 1
mysql> select * from sequences;
+-------------+----------+
| name        | next_val |
+-------------+----------+
| seq_dept_id |      101 |
| seq_emp_id  |     1001 |
+-------------+----------+
2 rows in set (0.00 sec)
```

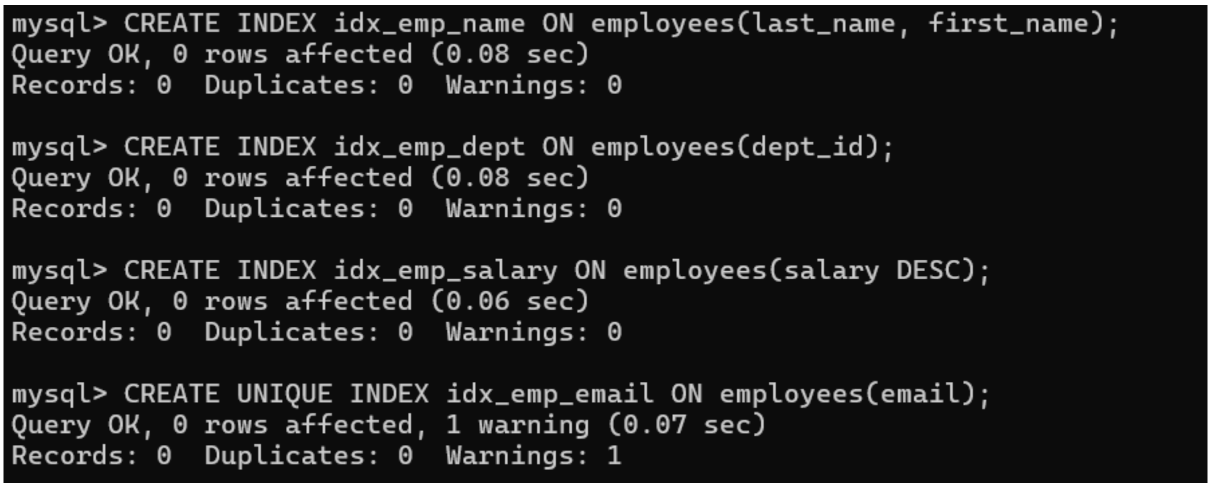
-- 5. CREATE INDEX

CREATE INDEX idx_emp_name ON employees(last_name, first_name);

CREATE INDEX idx_emp_dept ON employees(dept_id);

CREATE INDEX idx_emp_salary ON employees(salary DESC);

CREATE UNIQUE INDEX idx_emp_email ON employees(email);



```
mysql> CREATE INDEX idx_emp_name ON employees(last_name, first_name);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_emp_dept ON employees(dept_id);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX idx_emp_salary ON employees(salary DESC);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE UNIQUE INDEX idx_emp_email ON employees(email);
Query OK, 0 rows affected, 1 warning (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 1
```

-- 6. CREATE VIEW

CREATE VIEW employee_details AS

SELECT

    e.emp_id,

    e.first_name || ' ' || e.last_name AS full_name,

    e.email,

    e.salary,

    d.dept_name,

    d.location,

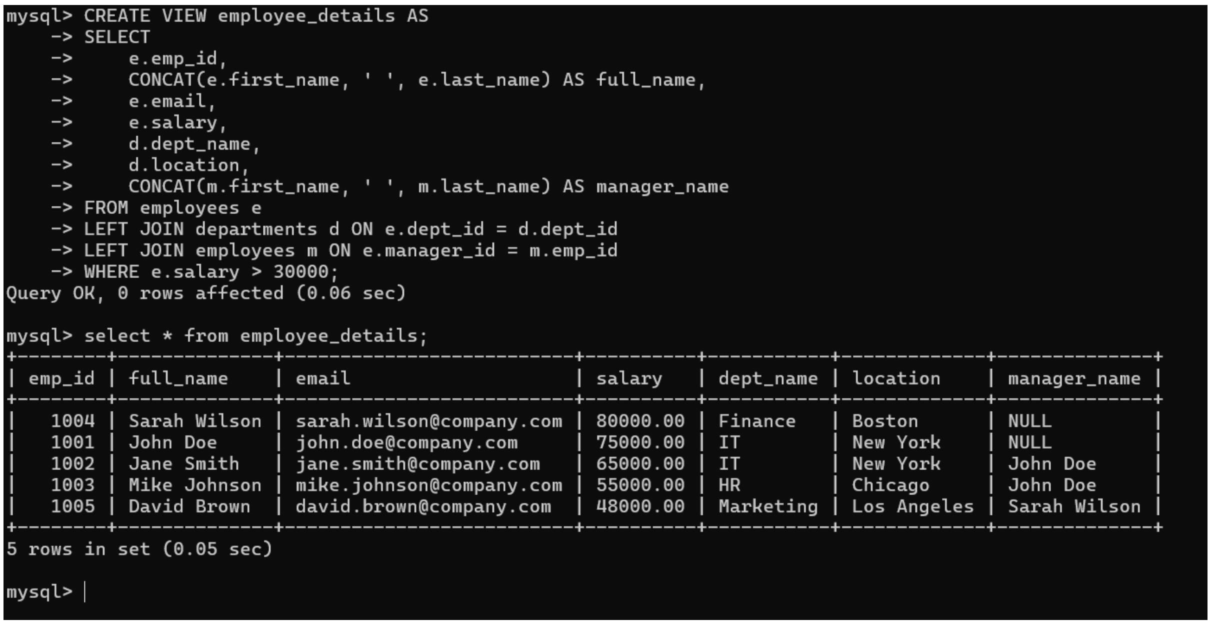
    m.first_name || ' ' || m.last_name AS manager_name

FROM employees e

LEFT JOIN departments d ON e.dept_id = d.dept_id

LEFT JOIN employees m ON e.manager_id = m.emp_id

WHERE e.salary > 30000;



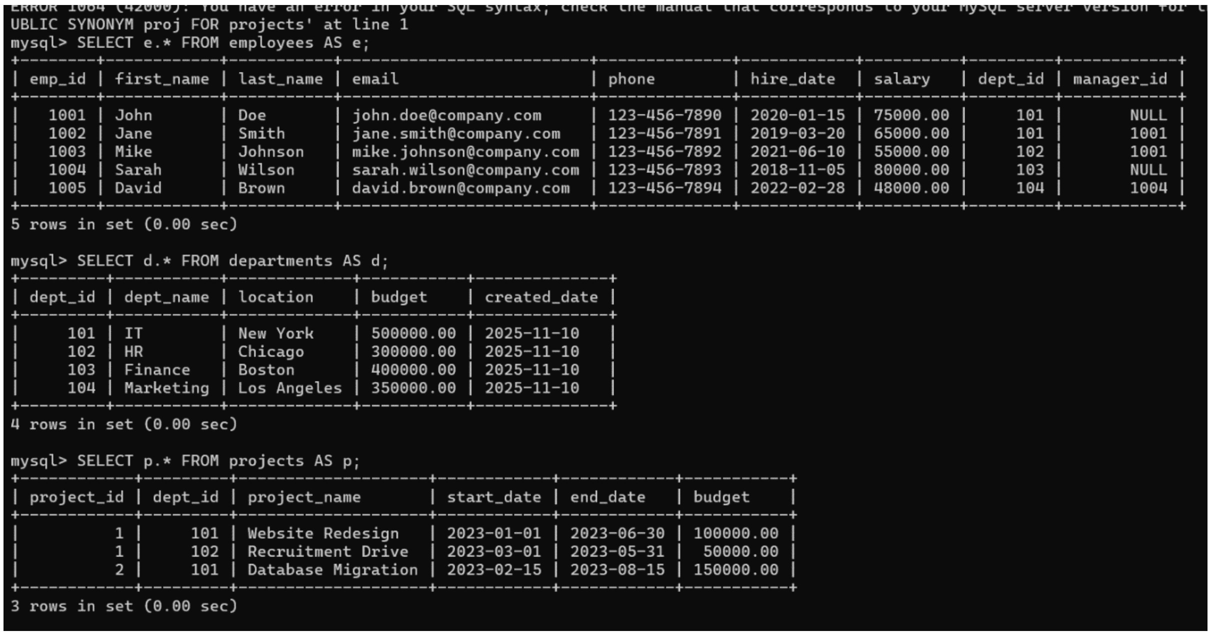
```
mysql> CREATE VIEW employee_details AS
    -> SELECT
    ->     e.emp_id,
    ->     CONCAT(e.first_name, ' ', e.last_name) AS full_name,
    ->     e.email,
    ->     e.salary,
    ->     d.dept_name,
    ->     d.location,
    ->     CONCAT(m.first_name, ' ', m.last_name) AS manager_name
    -> FROM employees e
    -> LEFT JOIN departments d ON e.dept_id = d.dept_id
    -> LEFT JOIN employees m ON e.manager_id = m.emp_id
    -> WHERE e.salary > 30000;
Query OK, 0 rows affected (0.06 sec)

mysql> select * from employee_details;
+--------+--------------+--------------------------+----------+-----------+-------------+--------------+
| emp_id | full_name    | email                    | salary   | dept_name | location    | manager_name |
+--------+--------------+--------------------------+----------+-----------+-------------+--------------+
|   1004 | Sarah Wilson | sarah.wilson@company.com | 80000.00 | Finance   | Boston      | NULL         |
|   1001 | John Doe     | john.doe@company.com     | 75000.00 | IT        | New York    | NULL         |
|   1002 | Jane Smith   | jane.smith@company.com   | 65000.00 | IT        | New York    | John Doe     |
|   1003 | Mike Johnson | mike.johnson@company.com | 55000.00 | HR        | Chicago     | John Doe     |
|   1005 | David Brown  | david.brown@company.com  | 48000.00 | Marketing | Los Angeles | Sarah Wilson |
+--------+--------------+--------------------------+----------+-----------+-------------+--------------+
5 rows in set (0.05 sec)

mysql> |
```

-- 7. CREATE SYNONYM

SELECT e.* FROM employees AS e;

SELECT d.* FROM departments AS d;

SELECT p.* FROM projects AS p;



```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for t
UBLIC SYNONYM proj FOR projects' at line 1
mysql> SELECT e.* FROM employees AS e;
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
| emp_id | first_name | last_name | email                    | phone        | hire_date  | salary   | dept_id | manager_id |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
|   1001 | John       | Doe       | john.doe@company.com     | 123-456-7890 | 2020-01-15 | 75000.00 |     101 |       NULL |
|   1002 | Jane       | Smith     | jane.smith@company.com   | 123-456-7891 | 2019-03-20 | 65000.00 |     101 |       1001 |
|   1003 | Mike       | Johnson   | mike.johnson@company.com | 123-456-7892 | 2021-06-10 | 55000.00 |     102 |       1001 |
|   1004 | Sarah      | Wilson    | sarah.wilson@company.com | 123-456-7893 | 2018-11-05 | 80000.00 |     103 |       NULL |
|   1005 | David      | Brown     | david.brown@company.com  | 123-456-7894 | 2022-02-28 | 48000.00 |     104 |       1004 |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
5 rows in set (0.00 sec)

mysql> SELECT d.* FROM departments AS d;
+---------+-----------+-------------+-----------+--------------+
| dept_id | dept_name | location    | budget    | created_date |
+---------+-----------+-------------+-----------+--------------+
|     101 | IT        | New York    | 500000.00 | 2025-11-10   |
|     102 | HR        | Chicago     | 300000.00 | 2025-11-10   |
|     103 | Finance   | Boston      | 400000.00 | 2025-11-10   |
|     104 | Marketing | Los Angeles | 350000.00 | 2025-11-10   |
+---------+-----------+-------------+-----------+--------------+
4 rows in set (0.00 sec)

mysql> SELECT p.* FROM projects AS p;
+------------+---------+--------------------+------------+------------+-----------+
| project_id | dept_id | project_name       | start_date | end_date   | budget    |
+------------+---------+--------------------+------------+------------+-----------+
|          1 |     101 | Website Redesign   | 2023-01-01 | 2023-06-30 | 100000.00 |
|          1 |     102 | Recruitment Drive  | 2023-03-01 | 2023-05-31 |  50000.00 |
|          2 |     101 | Database Migration | 2023-02-15 | 2023-08-15 | 150000.00 |
+------------+---------+--------------------+------------+------------+-----------+
3 rows in set (0.00 sec)
```

-- 8. Insert sample data

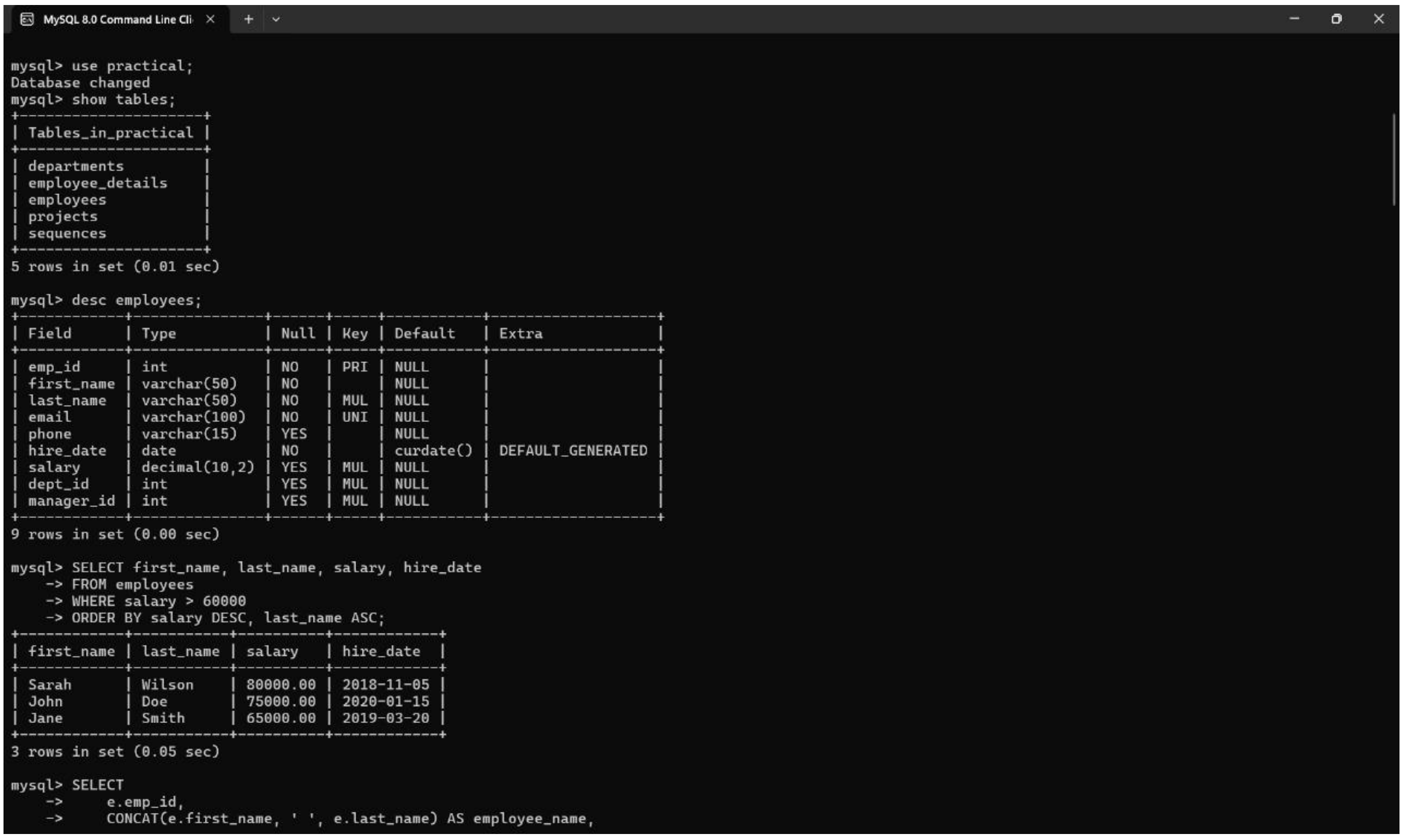
```
INSERT INTO departments VALUES (101, 'IT', 'New York', 500000, SYSDATE);

INSERT INTO departments VALUES (102, 'HR', 'Chicago', 300000, SYSDATE);

INSERT INTO departments VALUES (103, 'Finance', 'Boston', 400000, SYSDATE);

INSERT INTO departments VALUES (104, 'Marketing', 'Los Angeles', 350000, SYSDATE);


INSERT INTO employees VALUES (seq_emp_id.NEXTVAL, 'John', 'Doe', 'john.doe@company.com',

                 '123-456-7890', DATE '2020-01-15', 75000, 101, NULL);

INSERT INTO employees VALUES (seq_emp_id.NEXTVAL, 'Jane', 'Smith', 'jane.smith@company.com',

                 '123-456-7891', DATE '2019-03-20', 65000, 101, 1001);

INSERT INTO employees VALUES (seq_emp_id.NEXTVAL, 'Mike', 'Johnson', 'mike.johnson@company.com',

                 '123-456-7892', DATE '2021-06-10', 55000, 102, 1001);

INSERT INTO employees VALUES (seq_emp_id.NEXTVAL, 'Sarah', 'Wilson', 'sarah.wilson@company.com',

                 '123-456-7893', DATE '2018-11-05', 80000, 103, NULL);

INSERT INTO employees VALUES (seq_emp_id.NEXTVAL, 'David', 'Brown', 'david.brown@company.com',

                 '123-456-7894', DATE '2022-02-28', 48000, 104, 1004);


INSERT INTO projects VALUES (1, 101, 'Website Redesign', DATE '2023-01-01', DATE '2023-06-30', 100000);

INSERT INTO projects VALUES (2, 101, 'Database Migration', DATE '2023-02-15', DATE '2023-08-15', 150000);

INSERT INTO projects VALUES (1, 102, 'Recruitment Drive', DATE '2023-03-01', DATE '2023-05-31', 50000);
```

# 2B

Write at least 10 SQL queries on the suitable database application using SQL DML statements.

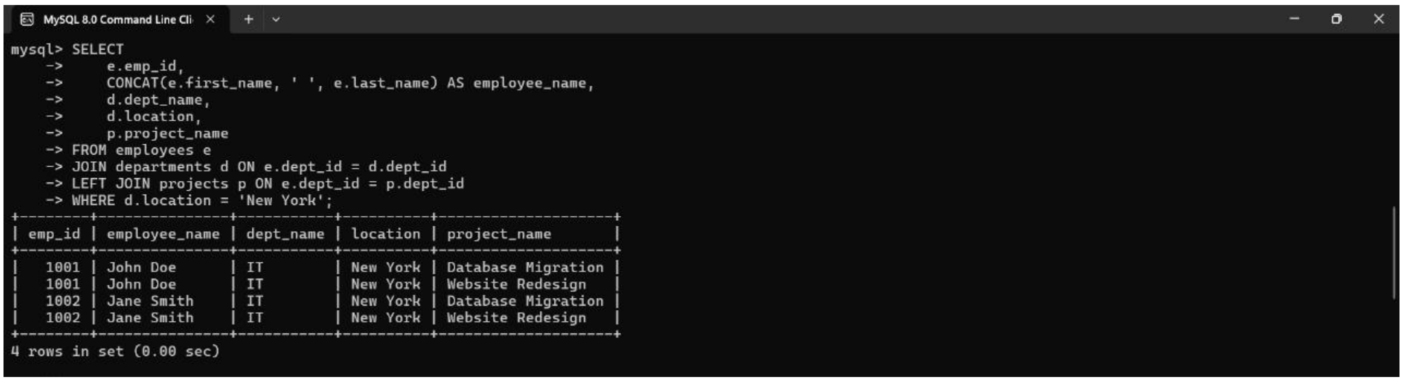
Note: Instructor will design the queries which demonstrate the use of concepts like Insert. Select, Update, Delete with operators, functions, and set operator etc.

```
-- 1. Basic SELECT with WHERE and ORDER BY

SELECT first_name, last_name, salary, hire_date

FROM employees

WHERE salary > 60000

ORDER BY salary DESC, last_name ASC;
```



```
mysql> use practical;
Database changed
mysql> show tables;
+---------------------+
| Tables_in_practical |
+---------------------+
| departments         |
| employee_details    |
| employees           |
| projects            |
| sequences           |
+---------------------+
5 rows in set (0.01 sec)

mysql> desc employees;
+------------+---------------+------+-----+-----------+-------------------+
| Field      | Type          | Null | Key | Default   | Extra             |
+------------+---------------+------+-----+-----------+-------------------+
| emp_id     | int           | NO   | PRI | NULL      |                   |
| first_name | varchar(50)   | NO   |     | NULL      |                   |
| last_name  | varchar(50)   | NO   | MUL | NULL      |                   |
| email      | varchar(100)  | NO   | UNI | NULL      |                   |
| phone      | varchar(15)   | YES  |     | NULL      |                   |
| hire_date  | date          | NO   |     | curdate() | DEFAULT_GENERATED |
| salary     | decimal(10,2) | YES  | MUL | NULL      |                   |
| dept_id    | int           | YES  | MUL | NULL      |                   |
| manager_id | int           | YES  | MUL | NULL      |                   |
+------------+---------------+------+-----+-----------+-------------------+
9 rows in set (0.00 sec)

mysql> SELECT first_name, last_name, salary, hire_date
    -> FROM employees
    -> WHERE salary > 60000
    -> ORDER BY salary DESC, last_name ASC;
+------------+-----------+----------+------------+
| first_name | last_name | salary   | hire_date  |
+------------+-----------+----------+------------+
| Sarah      | Wilson    | 80000.00 | 2018-11-05 |
| John       | Doe       | 75000.00 | 2020-01-15 |
| Jane       | Smith     | 65000.00 | 2019-03-20 |
+------------+-----------+----------+------------+
3 rows in set (0.05 sec)

mysql> SELECT
    ->     e.emp_id,
    ->     CONCAT(e.first_name, ' ', e.last_name) AS employee_name,
```

```
-- 2. JOIN between multiple tables

SELECT

    e.emp_id,

    CONCAT(e.first_name, ' ', e.last_name) AS employee_name,

    d.dept_name,

    d.location,

    p.project_name

FROM employees e

JOIN departments d ON e.dept_id = d.dept_id

LEFT JOIN projects p ON e.dept_id = p.dept_id

WHERE d.location = 'New York';
```



```
mysql> SELECT
    ->     e.emp_id,
    ->     CONCAT(e.first_name, ' ', e.last_name) AS employee_name,
    ->     d.dept_name,
    ->     d.location,
    ->     p.project_name
    -> FROM employees e
    -> JOIN departments d ON e.dept_id = d.dept_id
    -> LEFT JOIN projects p ON e.dept_id = p.dept_id
    -> WHERE d.location = 'New York';
+--------+---------------+-----------+----------+--------------------+
| emp_id | employee_name | dept_name | location | project_name       |
+--------+---------------+-----------+----------+--------------------+
|   1001 | John Doe      | IT        | New York | Database Migration |
|   1001 | John Doe      | IT        | New York | Website Redesign   |
|   1002 | Jane Smith    | IT        | New York | Database Migration |
|   1002 | Jane Smith    | IT        | New York | Website Redesign   |
+--------+---------------+-----------+----------+--------------------+
4 rows in set (0.00 sec)
```

--3. JOIN using RIGHT

SELECT

    e.emp_id,

    CONCAT(e.first_name, ' ', e.last_name) AS employee_name,

    d.dept_name,

    d.location,

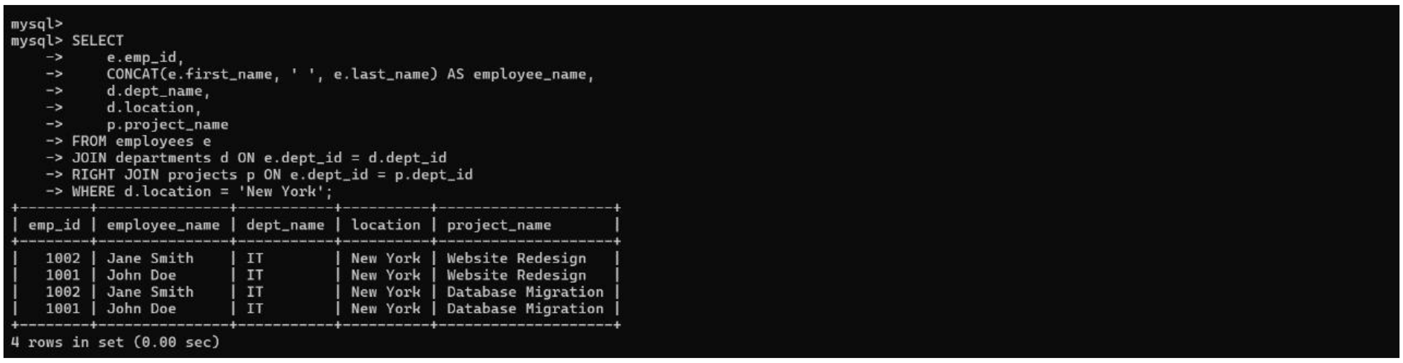
    p.project_name

FROM employees e

JOIN departments d ON e.dept_id = d.dept_id

RIGHT JOIN projects p ON e.dept_id = p.dept_id

WHERE d.location = 'New York';



```
mysql>
mysql> SELECT
    ->     e.emp_id,
    ->     CONCAT(e.first_name, ' ', e.last_name) AS employee_name,
    ->     d.dept_name,
    ->     d.location,
    ->     p.project_name
    -> FROM employees e
    -> JOIN departments d ON e.dept_id = d.dept_id
    -> RIGHT JOIN projects p ON e.dept_id = p.dept_id
    -> WHERE d.location = 'New York';
+--------+---------------+-----------+----------+--------------------+
| emp_id | employee_name | dept_name | location | project_name       |
+--------+---------------+-----------+----------+--------------------+
|   1002 | Jane Smith    | IT        | New York | Website Redesign   |
|   1001 | John Doe      | IT        | New York | Website Redesign   |
|   1002 | Jane Smith    | IT        | New York | Database Migration |
|   1001 | John Doe      | IT        | New York | Database Migration |
+--------+---------------+-----------+----------+--------------------+
4 rows in set (0.00 sec)
```

-- 4. Aggregate functions with GROUP BY and HAVING

SELECT

    d.dept_name,

    COUNT(e.emp_id) AS employee_count,

    AVG(e.salary) AS avg_salary,

    MAX(e.salary) AS max_salary,

    MIN(e.salary) AS min_salary

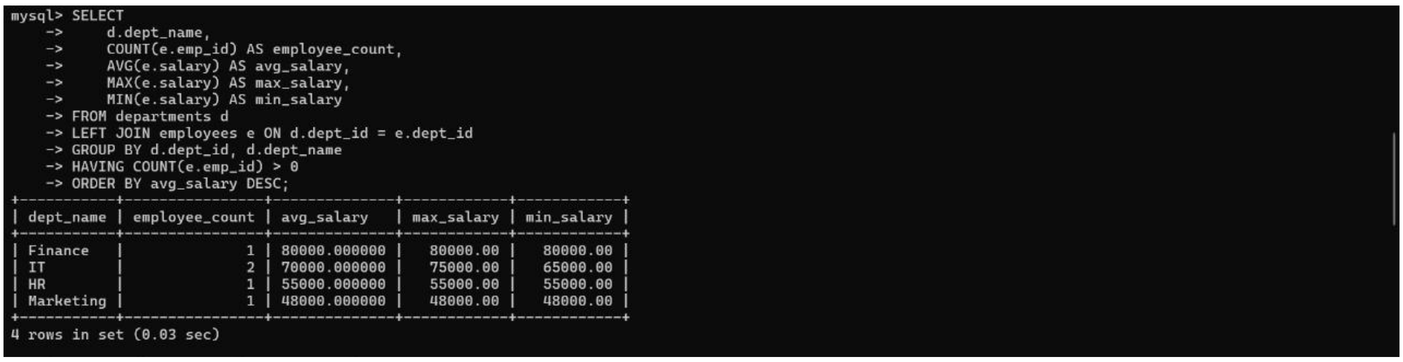
FROM departments d

LEFT JOIN employees e ON d.dept_id = e.dept_id

GROUP BY d.dept_id, d.dept_name

HAVING COUNT(e.emp_id) > 0

ORDER BY avg_salary DESC;



```
mysql> SELECT
    ->     d.dept_name,
    ->     COUNT(e.emp_id) AS employee_count,
    ->     AVG(e.salary) AS avg_salary,
    ->     MAX(e.salary) AS max_salary,
    ->     MIN(e.salary) AS min_salary
    -> FROM departments d
    -> LEFT JOIN employees e ON d.dept_id = e.dept_id
    -> GROUP BY d.dept_id, d.dept_name
    -> HAVING COUNT(e.emp_id) > 0
    -> ORDER BY avg_salary DESC;
+-----------+----------------+--------------+------------+------------+
| dept_name | employee_count | avg_salary   | max_salary | min_salary |
+-----------+----------------+--------------+------------+------------+
| Finance   |              1 | 80000.000000 |   80000.00 |   80000.00 |
| IT        |              2 | 70000.000000 |   75000.00 |   65000.00 |
| HR        |              1 | 55000.000000 |   55000.00 |   55000.00 |
| Marketing |              1 | 48000.000000 |   48000.00 |   48000.00 |
+-----------+----------------+--------------+------------+------------+
4 rows in set (0.03 sec)
```

-- 5. Subquery in WHERE clause

SELECT first_name, last_name, salary

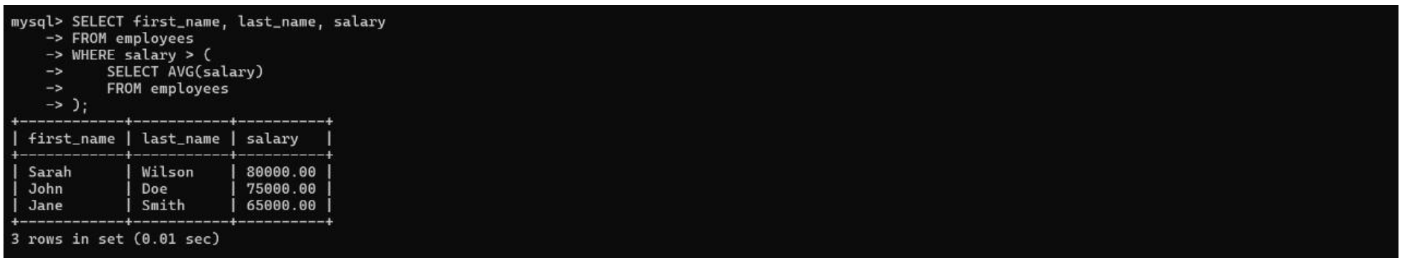
FROM employees

WHERE salary > (

    SELECT AVG(salary)

    FROM employees

);



```
mysql> SELECT first_name, last_name, salary
    -> FROM employees
    -> WHERE salary > (
    ->     SELECT AVG(salary)
    ->     FROM employees
    -> );
+------------+-----------+----------+
| first_name | last_name | salary   |
+------------+-----------+----------+
| Sarah      | Wilson    | 80000.00 |
| John       | Doe       | 75000.00 |
| Jane       | Smith     | 65000.00 |
+------------+-----------+----------+
3 rows in set (0.01 sec)
```

-- 6. UPDATE with subquery

UPDATE employees

SET salary = salary * 1.10

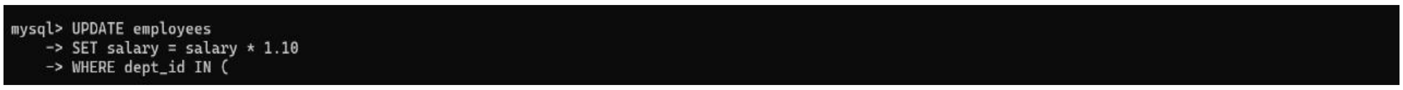
WHERE dept_id IN (

    SELECT dept_id

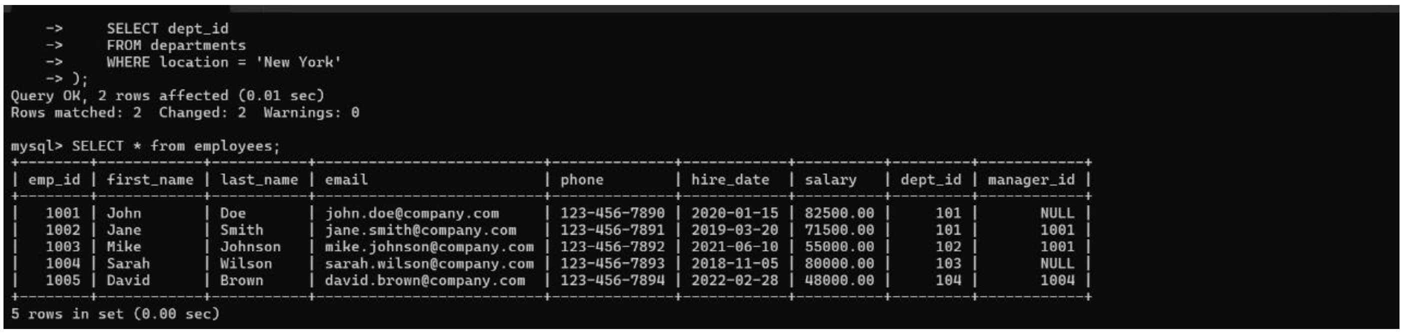
    FROM departments

    WHERE location = 'New York'

);



```
mysql> UPDATE employees
    -> SET salary = salary * 1.10
    -> WHERE dept_id IN (
```



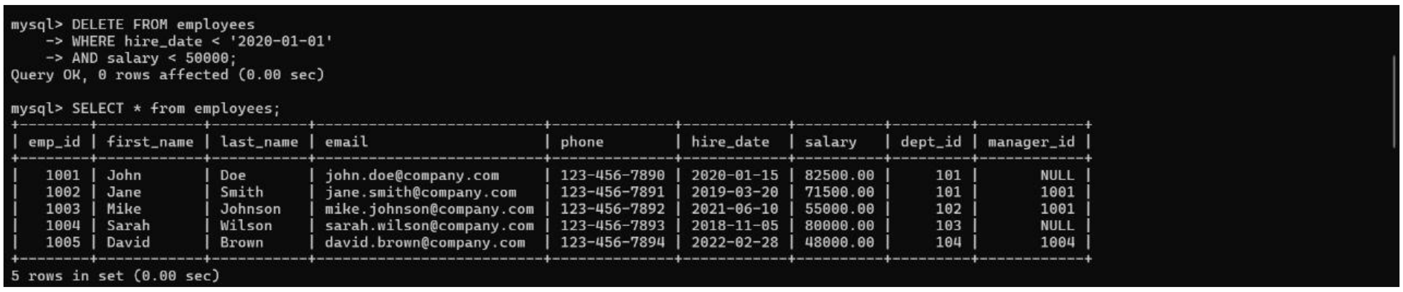
```
    ->     SELECT dept_id
    ->     FROM departments
    ->     WHERE location = 'New York'
    -> );
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT * from employees;
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
| emp_id | first_name | last_name | email                    | phone        | hire_date  | salary   | dept_id | manager_id |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
|   1001 | John       | Doe       | john.doe@company.com     | 123-456-7890 | 2020-01-15 | 82500.00 |     101 |       NULL |
|   1002 | Jane       | Smith     | jane.smith@company.com   | 123-456-7891 | 2019-03-20 | 71500.00 |     101 |       1001 |
|   1003 | Mike       | Johnson   | mike.johnson@company.com | 123-456-7892 | 2021-06-10 | 55000.00 |     102 |       1001 |
|   1004 | Sarah      | Wilson    | sarah.wilson@company.com | 123-456-7893 | 2018-11-05 | 80000.00 |     103 |       NULL |
|   1005 | David      | Brown     | david.brown@company.com  | 123-456-7894 | 2022-02-28 | 48000.00 |     104 |       1004 |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
5 rows in set (0.00 sec)
```

-- 7. DELETE with condition

DELETE FROM employees

WHERE hire_date < '2020-01-01'

AND salary < 50000;

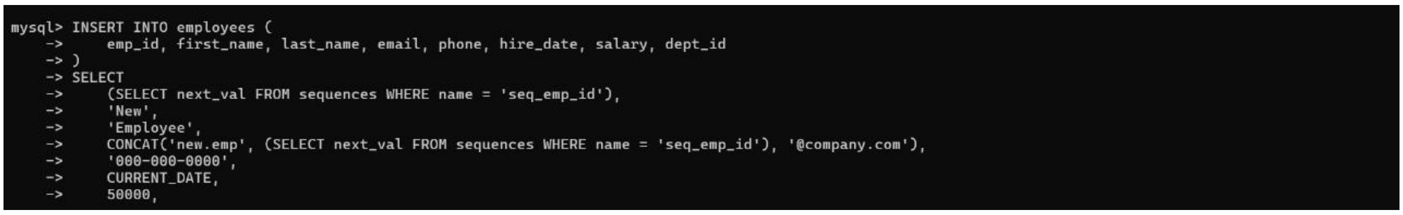


```
mysql> DELETE FROM employees
    -> WHERE hire_date < '2020-01-01'
    -> AND salary < 50000;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * from employees;
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
| emp_id | first_name | last_name | email                    | phone        | hire_date  | salary   | dept_id | manager_id |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
|   1001 | John       | Doe       | john.doe@company.com     | 123-456-7890 | 2020-01-15 | 82500.00 |     101 |       NULL |
|   1002 | Jane       | Smith     | jane.smith@company.com   | 123-456-7891 | 2019-03-20 | 71500.00 |     101 |       1001 |
|   1003 | Mike       | Johnson   | mike.johnson@company.com | 123-456-7892 | 2021-06-10 | 55000.00 |     102 |       1001 |
|   1004 | Sarah      | Wilson    | sarah.wilson@company.com | 123-456-7893 | 2018-11-05 | 80000.00 |     103 |       NULL |
|   1005 | David      | Brown     | david.brown@company.com  | 123-456-7894 | 2022-02-28 | 48000.00 |     104 |       1004 |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
5 rows in set (0.00 sec)
```

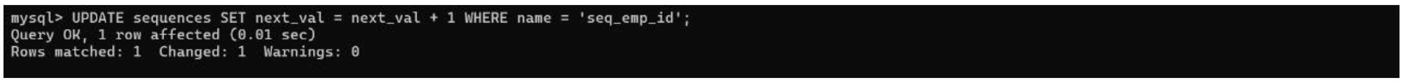
--8.  Update sequence after insert

INSERT INTO employees (
emp_id, first_name, last_name, email, phone, hire date, salary, dept_id
SELECT
(SELECT next_val FROM sequences WHERE name 'seq_emp_id'),
New',
'Employee',
CONCAT('new.emp, (SELECT next_val FROM sequences WHERE name = 'seq_emp_id'), '@company.com'),
1006-006-0000',
CURRENT DATE,
50000,

UPDATE sequences SET next_val = next_val + 1 WHERE name = 'seq_emp_id';



```
mysql> INSERT INTO employees (
    ->     emp_id, first_name, last_name, email, phone, hire_date, salary, dept_id
    -> )
    -> SELECT
    ->     (SELECT next_val FROM sequences WHERE name = 'seq_emp_id'),
    ->     'New',
    ->     'Employee',
    ->     CONCAT('new.emp', (SELECT next_val FROM sequences WHERE name = 'seq_emp_id'), '@company.com'),
    ->     '000-000-0000',
    ->     CURRENT_DATE,
    ->     50000,
```



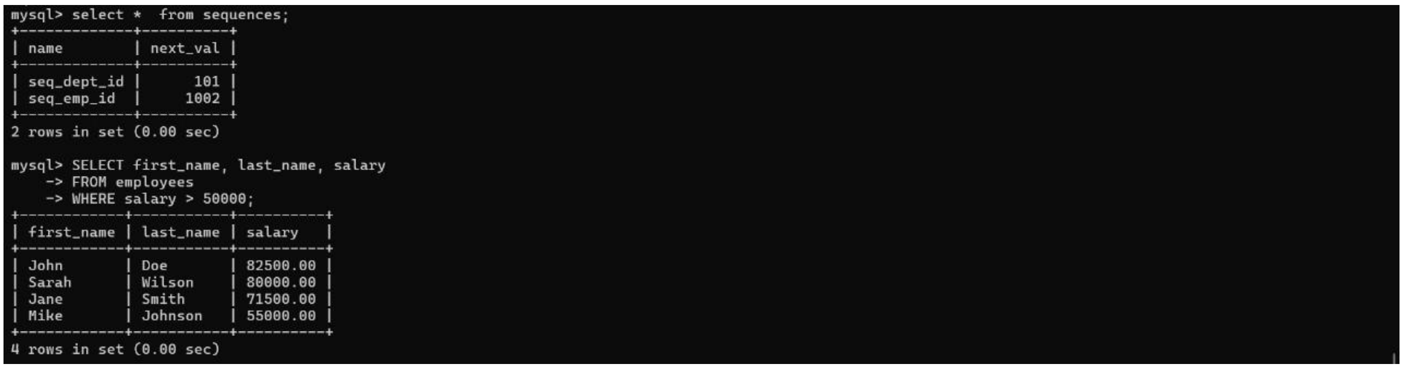
```
mysql> UPDATE sequences SET next_val = next_val + 1 WHERE name = 'seq_emp_id';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

--9. SELECT with WHERE

SELECT first_name, last_name, salary

FROM employees

WHERE salary > 50000;



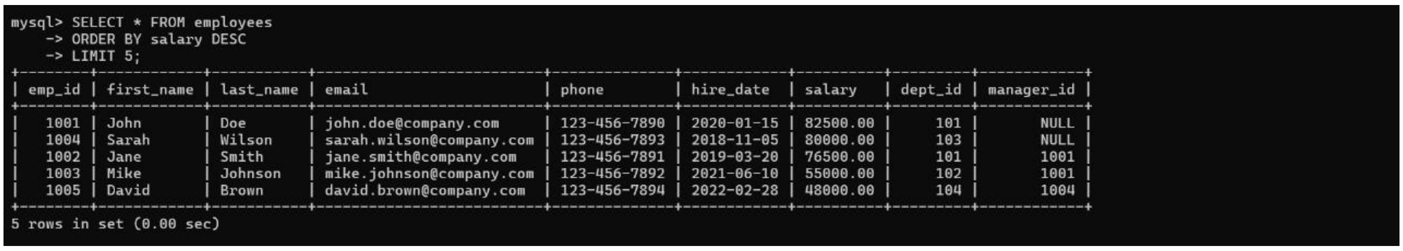
```
mysql> select *  from sequences;
+-------------+----------+
| name        | next_val |
+-------------+----------+
| seq_dept_id |      101 |
| seq_emp_id  |     1002 |
+-------------+----------+
2 rows in set (0.00 sec)

mysql> SELECT first_name, last_name, salary
    -> FROM employees
    -> WHERE salary > 50000;
+------------+-----------+----------+
| first_name | last_name | salary   |
+------------+-----------+----------+
| John       | Doe       | 82500.00 |
| Sarah      | Wilson    | 80000.00 |
| Jane       | Smith     | 71500.00 |
| Mike       | Johnson   | 55000.00 |
+------------+-----------+----------+
4 rows in set (0.00 sec)
```

--10. SELECT with ORDER BY

SELECT first_name, last_name, hire_date

FROM employees

ORDER BY hire_date DESC;

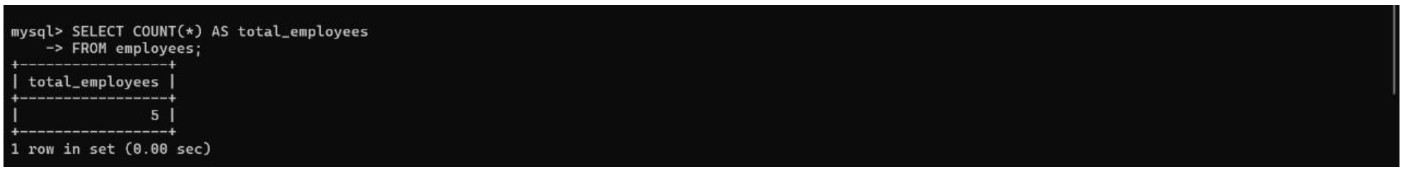


```
mysql> SELECT * FROM employees
    -> ORDER BY salary DESC
    -> LIMIT 5;
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
| emp_id | first_name | last_name | email                    | phone        | hire_date  | salary   | dept_id | manager_id |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
|   1001 | John       | Doe       | john.doe@company.com     | 123-456-7890 | 2020-01-15 | 82500.00 |     101 |       NULL |
|   1004 | Sarah      | Wilson    | sarah.wilson@company.com | 123-456-7893 | 2018-11-05 | 80000.00 |     103 |       NULL |
|   1002 | Jane       | Smith     | jane.smith@company.com   | 123-456-7891 | 2019-03-20 | 76500.00 |     101 |       1001 |
|   1003 | Mike       | Johnson   | mike.johnson@company.com | 123-456-7892 | 2021-06-10 | 55000.00 |     102 |       1001 |
|   1005 | David      | Brown     | david.brown@company.com  | 123-456-7894 | 2022-02-28 | 48000.00 |     104 |       1004 |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
5 rows in set (0.00 sec)
```

--11. COUNT records

SELECT COUNT(*) AS total_employees

FROM employees;



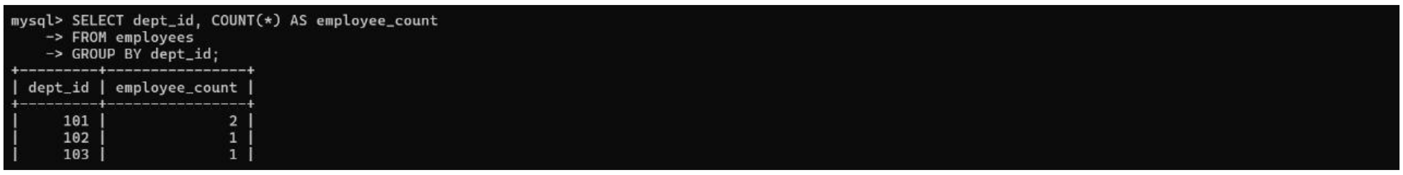
```
mysql> SELECT COUNT(*) AS total_employees
    -> FROM employees;
+-----------------+
| total_employees |
+-----------------+
|               5 |
+-----------------+
1 row in set (0.00 sec)
```

--12. GROUP BY with COUNT

SELECT dept_id, COUNT(*) AS employee_count

FROM employees

GROUP BY dept_id;



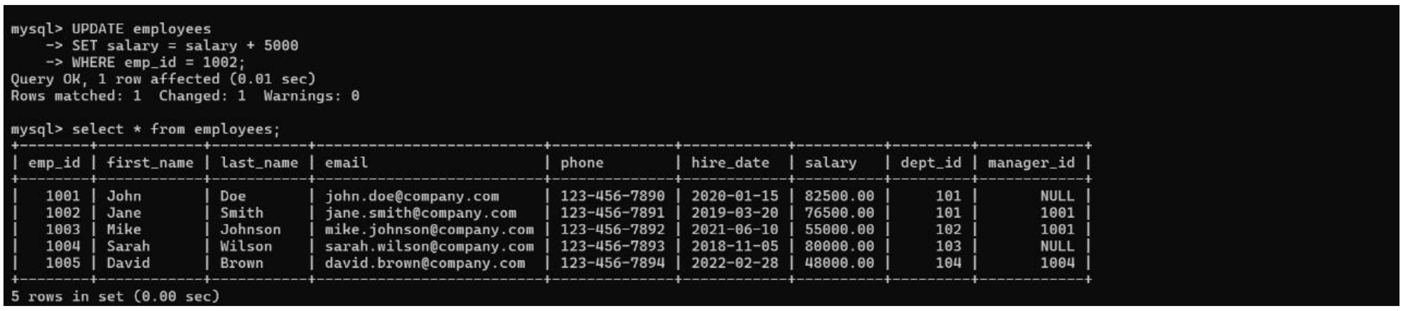
```
mysql> SELECT dept_id, COUNT(*) AS employee_count
    -> FROM employees
    -> GROUP BY dept_id;
+---------+----------------+
| dept_id | employee_count |
+---------+----------------+
|     101 |              2 |
|     102 |              1 |
|     103 |              1 |
```

--13. UPDATE specific record

UPDATE employees

SET salary = salary + 5000

WHERE emp_id = 1002;



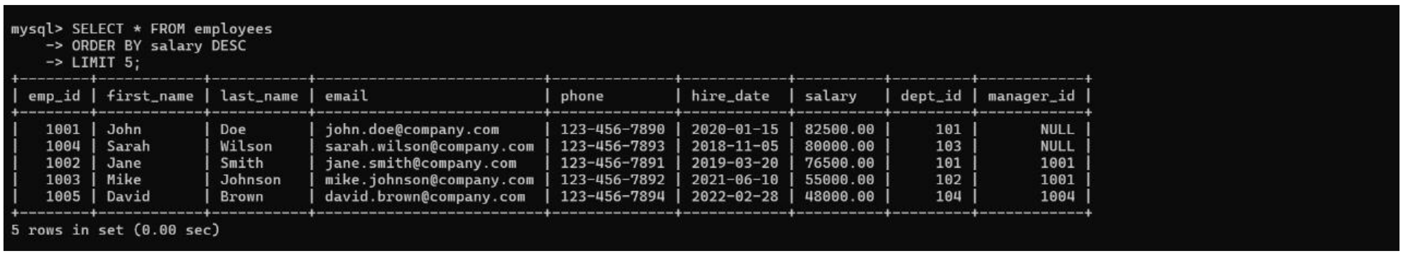
```
mysql> UPDATE employees
    -> SET salary = salary + 5000
    -> WHERE emp_id = 1002;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employees;
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
| emp_id | first_name | last_name | email                    | phone        | hire_date  | salary   | dept_id | manager_id |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
|   1001 | John       | Doe       | john.doe@company.com     | 123-456-7890 | 2020-01-15 | 82500.00 |     101 |       NULL |
|   1002 | Jane       | Smith     | jane.smith@company.com   | 123-456-7891 | 2019-03-20 | 76500.00 |     101 |       1001 |
|   1003 | Mike       | Johnson   | mike.johnson@company.com | 123-456-7892 | 2021-06-10 | 55000.00 |     102 |       1001 |
|   1004 | Sarah      | Wilson    | sarah.wilson@company.com | 123-456-7893 | 2018-11-05 | 80000.00 |     103 |       NULL |
|   1005 | David      | Brown     | david.brown@company.com  | 123-456-7894 | 2022-02-28 | 48000.00 |     104 |       1004 |
+--------+------------+-----------+--------------------------+--------------+------------+----------+---------+------------+
5 rows in set (0.00 sec)
```

--14. SELECT with LIMIT

SELECT * FROM employees

ORDER BY salary DESC

LIMIT 5;



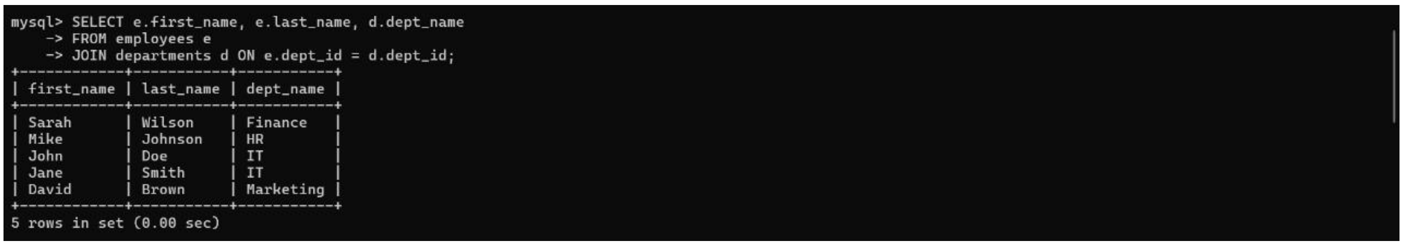
```
mysql> SELECT * FROM employees
    -> ORDER BY salary DESC
    -> LIMIT 5;
+--------+------------+-----------+---------------------------+--------------+------------+----------+---------+------------+
| emp_id | first_name | last_name | email                     | phone        | hire_date  | salary   | dept_id | manager_id |
+--------+------------+-----------+---------------------------+--------------+------------+----------+---------+------------+
|   1001 | John       | Doe       | john.doe@company.com      | 123-456-7890 | 2020-01-15 | 82500.00 |     101 |       NULL |
|   1004 | Sarah      | Wilson    | sarah.wilson@company.com  | 123-456-7893 | 2018-11-05 | 80000.00 |     103 |       NULL |
|   1002 | Jane       | Smith     | jane.smith@company.com    | 123-456-7891 | 2019-03-20 | 76500.00 |     101 |       1001 |
|   1003 | Mike       | Johnson   | mike.johnson@company.com  | 123-456-7892 | 2021-06-10 | 55000.00 |     102 |       1001 |
|   1005 | David      | Brown     | david.brown@company.com   | 123-456-7894 | 2022-02-28 | 48000.00 |     104 |       1004 |
+--------+------------+-----------+---------------------------+--------------+------------+----------+---------+------------+
5 rows in set (0.00 sec)
```

--15. JOIN two tables

SELECT e.first_name, e.last_name, d.dept_name

FROM employees e

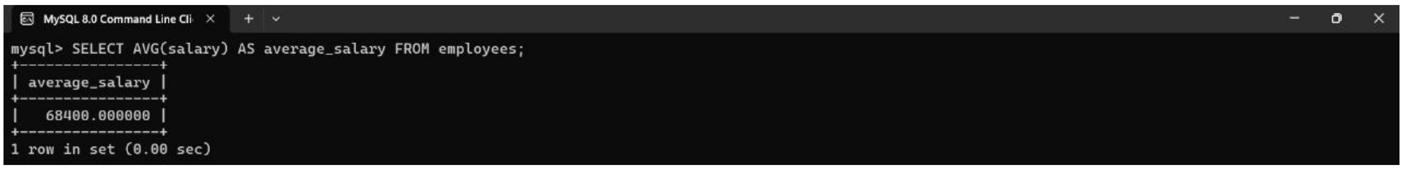
JOIN departments d ON e.dept_id = d.dept_id;



```
mysql> SELECT e.first_name, e.last_name, d.dept_name
    -> FROM employees e
    -> JOIN departments d ON e.dept_id = d.dept_id;
+------------+-----------+-----------+
| first_name | last_name | dept_name |
+------------+-----------+-----------+
| Sarah      | Wilson    | Finance   |
| Mike       | Johnson   | HR        |
| John       | Doe       | IT        |
| Jane       | Smith     | IT        |
| David      | Brown     | Marketing |
+------------+-----------+-----------+
5 rows in set (0.00 sec)
```

--16. Find average salary

SELECT AVG(salary) AS average_salary FROM employees;

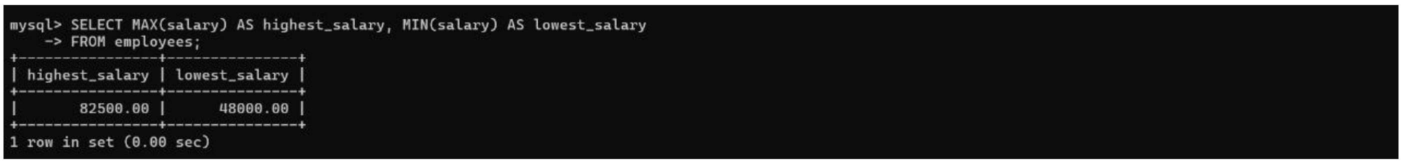


```
mysql> SELECT AVG(salary) AS average_salary FROM employees;
+----------------+
| average_salary |
+----------------+
|   68400.000000 |
+----------------+
1 row in set (0.00 sec)
```

--17. Find highest and lowest salary

SELECT MAX(salary) AS highest_salary, MIN(salary) AS lowest_salary

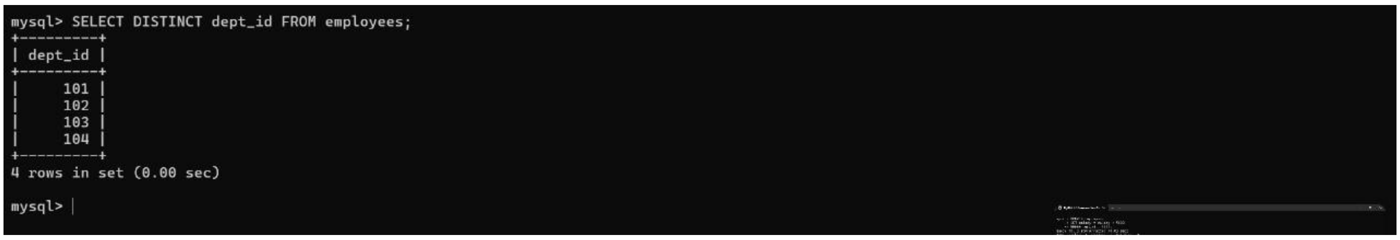
FROM employees;



```
mysql> SELECT MAX(salary) AS highest_salary, MIN(salary) AS lowest_salary
    -> FROM employees;
+----------------+---------------+
| highest_salary | lowest_salary |
+----------------+---------------+
|       82500.00 |      48000.00 |
+----------------+---------------+
1 row in set (0.00 sec)
```

--18. Select distinct departments

SELECT DISTINCT dept_id FROM employees;



```
mysql> SELECT DISTINCT dept_id FROM employees;
+---------+
| dept_id |
+---------+
|     101 |
|     102 |
|     103 |
|     104 |
+---------+
4 rows in set (0.00 sec)

mysql> |
```

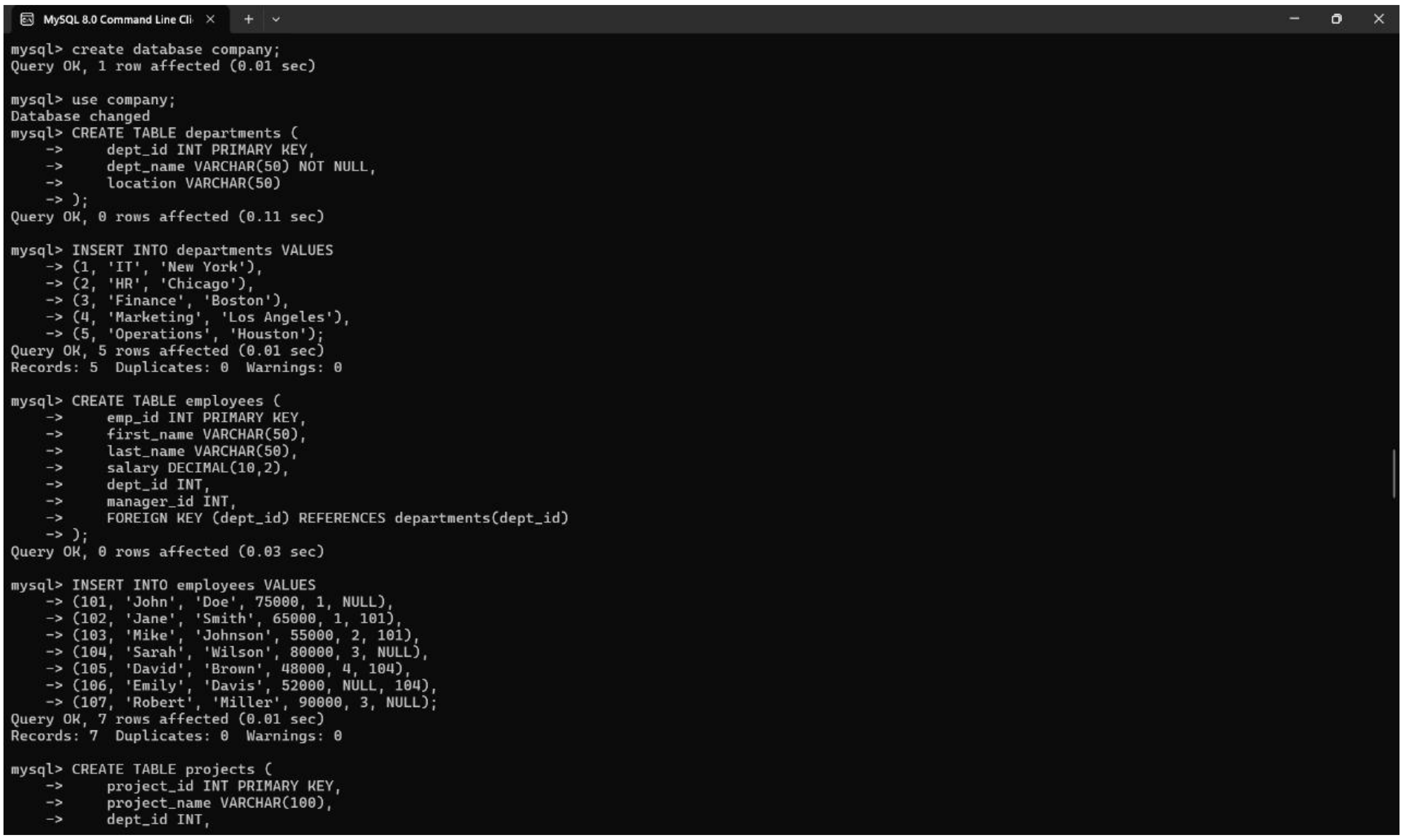
# 3 prac

SQL Queries all types of Join, Sub-Query and View:

Write at least10 SQL queries for suitable database application using SQL DML statements.

Note: Instructor will design the queries which demonstrate the use of concepts like all types of

Join Sub-Query and View



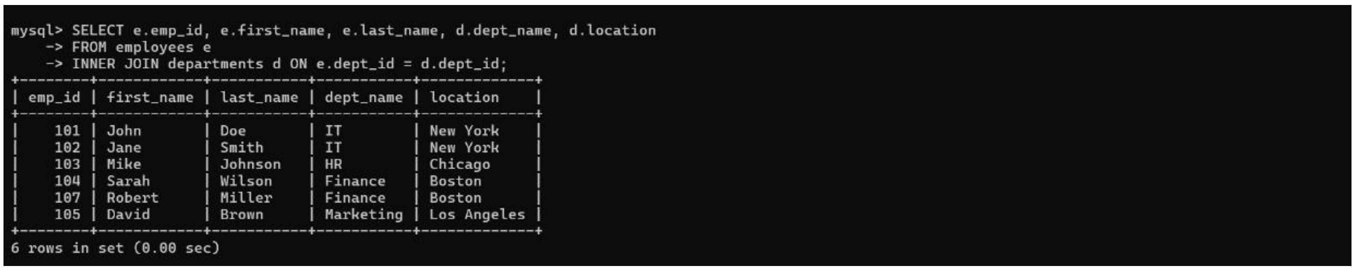
```
mysql> create database company;
Query OK, 1 row affected (0.01 sec)

mysql> use company;
Database changed
mysql> CREATE TABLE departments (
    ->     dept_id INT PRIMARY KEY,
    ->     dept_name VARCHAR(50) NOT NULL,
    ->     location VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> INSERT INTO departments VALUES
    -> (1, 'IT', 'New York'),
    -> (2, 'HR', 'Chicago'),
    -> (3, 'Finance', 'Boston'),
    -> (4, 'Marketing', 'Los Angeles'),
    -> (5, 'Operations', 'Houston');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> CREATE TABLE employees (
    ->     emp_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     salary DECIMAL(10,2),
    ->     dept_id INT,
    ->     manager_id INT,
    ->     FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO employees VALUES
    -> (101, 'John', 'Doe', 75000, 1, NULL),
    -> (102, 'Jane', 'Smith', 65000, 1, 101),
    -> (103, 'Mike', 'Johnson', 55000, 2, 101),
    -> (104, 'Sarah', 'Wilson', 80000, 3, NULL),
    -> (105, 'David', 'Brown', 48000, 4, 104),
    -> (106, 'Emily', 'Davis', 52000, NULL, 104),
    -> (107, 'Robert', 'Miller', 90000, 3, NULL);
Query OK, 7 rows affected (0.01 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> CREATE TABLE projects (
    ->     project_id INT PRIMARY KEY,
    ->     project_name VARCHAR(100),
    ->     dept_id INT,
```

--1. INNER JOIN - Employees with their departments

SELECT e.emp_id, e.first_name, e.last_name, d.dept_name, d.location

FROM employees e

INNER JOIN depar tments d ON e.dept_id = d.dept_id;



```
mysql> SELECT e.emp_id, e.first_name, e.last_name, d.dept_name, d.location
    -> FROM employees e
    -> INNER JOIN departments d ON e.dept_id = d.dept_id;
+--------+------------+-----------+-----------+-------------+
| emp_id | first_name | last_name | dept_name | location    |
+--------+------------+-----------+-----------+-------------+
|    101 | John       | Doe       | IT        | New York    |
|    102 | Jane       | Smith     | IT        | New York    |
|    103 | Mike       | Johnson   | HR        | Chicago     |
|    104 | Sarah      | Wilson    | Finance   | Boston      |
|    107 | Robert     | Miller    | Finance   | Boston      |
|    105 | David      | Brown     | Marketing | Los Angeles |
+--------+------------+-----------+-----------+-------------+
6 rows in set (0.00 sec)
```

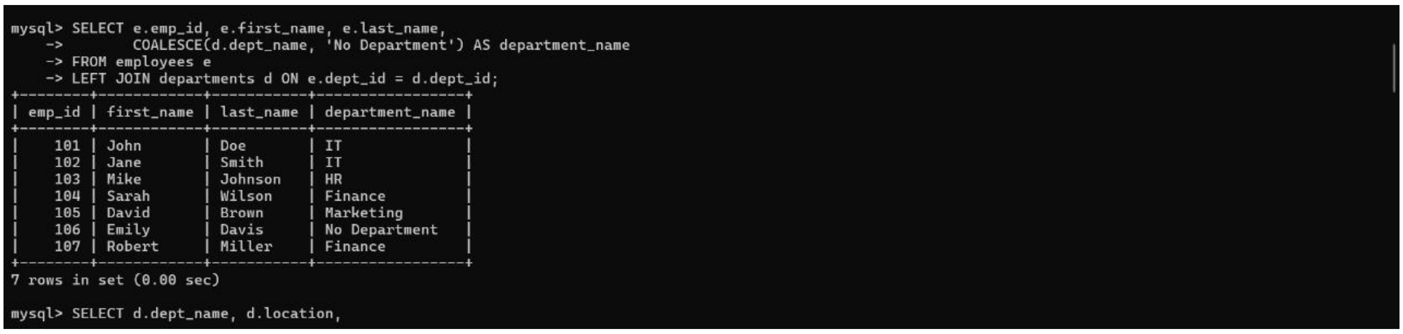
--2. LEFT JOIN - All employees with their department info

SELECT e.emp_id, e.first_name, e.last_name,

    COALESCE(d.dept_name, 'No Department') AS department_name

FROM employees e

LEFT JOIN departments d ON e.dept_id = d.dept_id;


```
mysql> SELECT e.emp_id, e.first_name, e.last_name,
    ->        COALESCE(d.dept_name, 'No Department') AS department_name
    -> FROM employees e
    -> LEFT JOIN departments d ON e.dept_id = d.dept_id;
+--------+------------+-----------+-----------------+
| emp_id | first_name | last_name | department_name |
+--------+------------+-----------+-----------------+
|    101 | John       | Doe       | IT              |
|    102 | Jane       | Smith     | IT              |
|    103 | Mike       | Johnson   | HR              |
|    104 | Sarah      | Wilson    | Finance         |
|    105 | David      | Brown     | Marketing       |
|    106 | Emily      | Davis     | No Department   |
|    107 | Robert     | Miller    | Finance         |
+--------+------------+-----------+-----------------+
7 rows in set (0.00 sec)

mysql> SELECT d.dept_name, d.location,
```

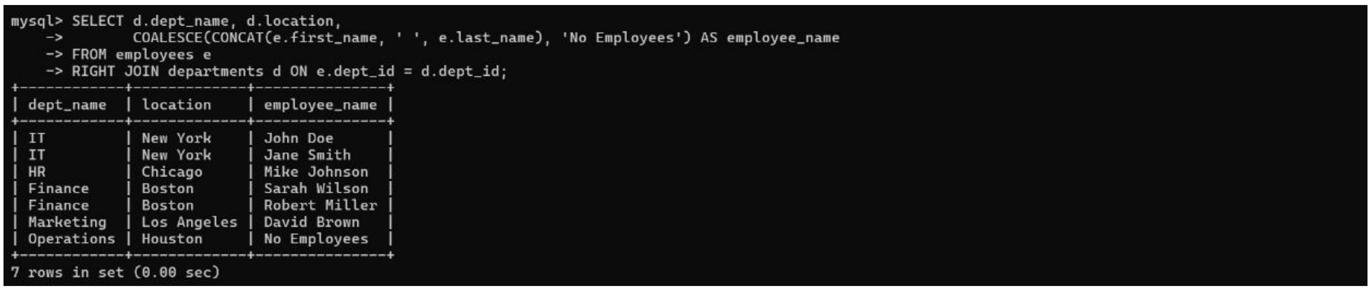
--3. RIGHT JOIN - All departments with their employees

SELECT d.dept_name, d.location,

COALESCE(CONCAT(e.first_name, ' ', e.last_name), 'No Employees') AS employee_name

FROM employees e

RIGHT JOIN departments d ON e.dept_id = d.dept_id;


```
mysql> SELECT d.dept_name, d.location,
    ->        COALESCE(CONCAT(e.first_name, ' ', e.last_name), 'No Employees') AS employee_name
    -> FROM employees e
    -> RIGHT JOIN departments d ON e.dept_id = d.dept_id;
+------------+-------------+---------------+
| dept_name  | location    | employee_name |
+------------+-------------+---------------+
| IT         | New York    | John Doe      |
| IT         | New York    | Jane Smith    |
| HR         | Chicago     | Mike Johnson  |
| Finance    | Boston      | Sarah Wilson  |
| Finance    | Boston      | Robert Miller |
| Marketing  | Los Angeles | David Brown   |
| Operations | Houston     | No Employees  |
+------------+-------------+---------------+
7 rows in set (0.00 sec)
```

--4. FULL OUTER JOIN (using UNION)

SELECT e.emp_id, e.first_name, e.last_name, d.dept_name, d.location

FROM employees e

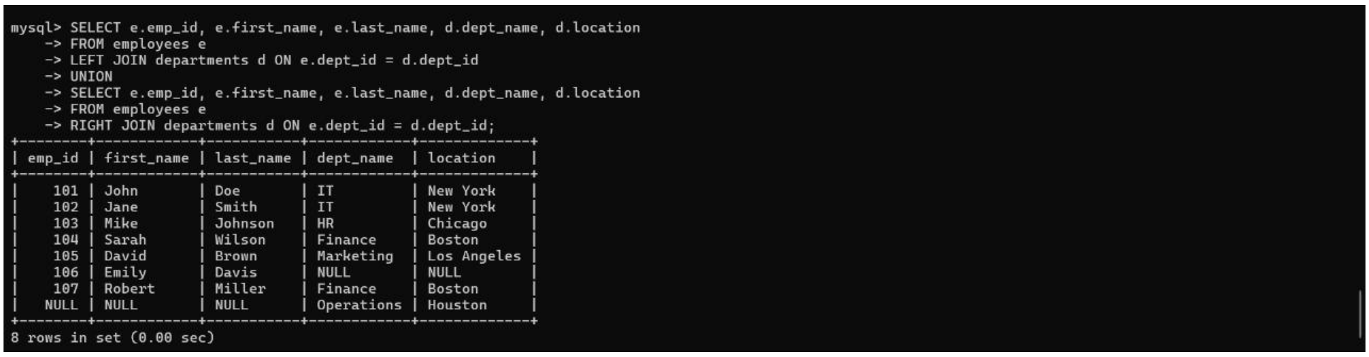
LEFT JOIN departments d ON e.dept_id = d.dept_id

UNION

SELECT e.emp_id, e.first_name, e.last_name, d.dept_name, d.location

FROM employees e

RIGHT JO IN departments d ON e.dept_id = d.dept_id;


```
mysql> SELECT e.emp_id, e.first_name, e.last_name, d.dept_name, d.location
    -> FROM employees e
    -> LEFT JOIN departments d ON e.dept_id = d.dept_id
    -> UNION
    -> SELECT e.emp_id, e.first_name, e.last_name, d.dept_name, d.location
    -> FROM employees e
    -> RIGHT JOIN departments d ON e.dept_id = d.dept_id;
+--------+------------+-----------+------------+-------------+
| emp_id | first_name | last_name | dept_name  | location    |
+--------+------------+-----------+------------+-------------+
|    101 | John       | Doe       | IT         | New York    |
|    102 | Jane       | Smith     | IT         | New York    |
|    103 | Mike       | Johnson   | HR         | Chicago     |
|    104 | Sarah      | Wilson    | Finance    | Boston      |
|    105 | David      | Brown     | Marketing  | Los Angeles |
|    106 | Emily      | Davis     | NULL       | NULL        |
|    107 | Robert     | Miller    | Finance    | Boston      |
|   NULL | NULL       | NULL      | Operations | Houston     |
+--------+------------+-----------+------------+-------------+
8 rows in set (0.00 sec)
```

--5. SELF JOIN

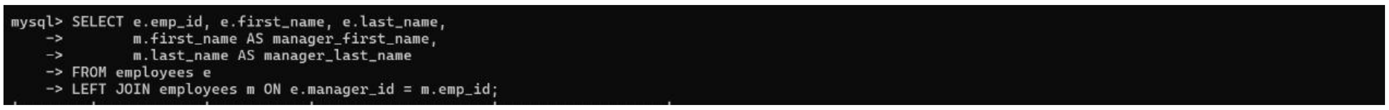
SELECT e.emp_id, e.first_name, e.last_name,

    m.first_name AS manager_first_name,

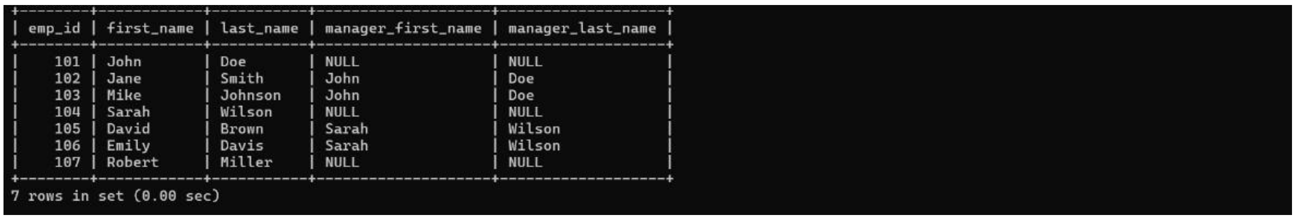
    m.last_name AS manager_last_name

FROM employees e

LEFT JOIN employees m ON e.manager_id = m.emp_id;



```
mysql> SELECT e.emp_id, e.first_name, e.last_name,
    ->        m.first_name AS manager_first_name,
    ->        m.last_name AS manager_last_name
    -> FROM employees e
    -> LEFT JOIN employees m ON e.manager_id = m.emp_id;
```



```
+--------+------------+-----------+--------------------+-------------------+
| emp_id | first_name | last_name | manager_first_name | manager_last_name |
+--------+------------+-----------+--------------------+-------------------+
|    101 | John       | Doe       | NULL               | NULL              |
|    102 | Jane       | Smith     | John               | Doe               |
|    103 | Mike       | Johnson   | John               | Doe               |
|    104 | Sarah      | Wilson    | NULL               | NULL              |
|    105 | David      | Brown     | Sarah              | Wilson            |
|    106 | Emily      | Davis     | Sarah              | Wilson            |
|    107 | Robert     | Miller    | NULL               | NULL              |
+--------+------------+-----------+--------------------+-------------------+
7 rows in set (0.00 sec)
```

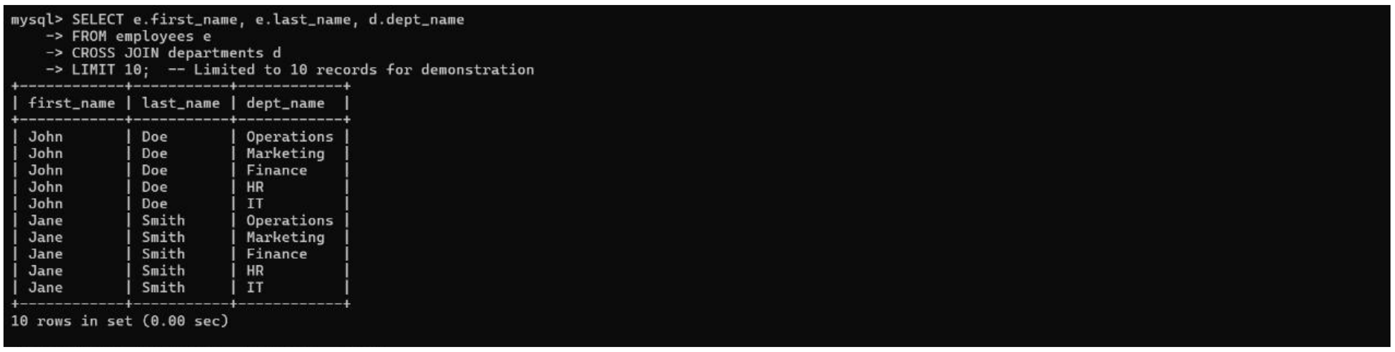
--6. CROSS JOIN

SELECT e.first_name, e.last_name, d.dept_name

FROM employees e

CROSS JOIN departments d

LIMIT 10;



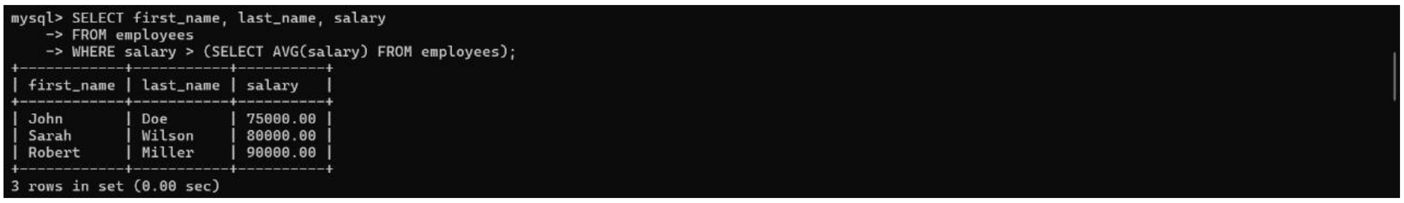
```
mysql> SELECT e.first_name, e.last_name, d.dept_name
    -> FROM employees e
    -> CROSS JOIN departments d
    -> LIMIT 10;  -- Limited to 10 records for demonstration
+------------+-----------+------------+
| first_name | last_name | dept_name  |
+------------+-----------+------------+
| John       | Doe       | Operations |
| John       | Doe       | Marketing  |
| John       | Doe       | Finance    |
| John       | Doe       | HR         |
| John       | Doe       | IT         |
| Jane       | Smith     | Operations |
| Jane       | Smith     | Marketing  |
| Jane       | Smith     | Finance    |
| Jane       | Smith     | HR         |
| Jane       | Smith     | IT         |
+------------+-----------+------------+
10 rows in set (0.00 sec)
```

--7. SUBQUERY in WHERE clause

SELECT first_name, last_name, salary

FROM employees

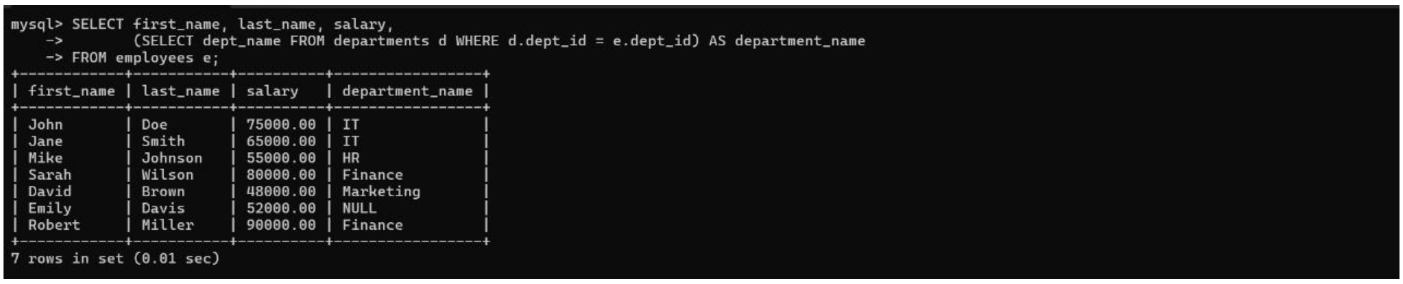
WHERE salary > (SELECT AVG(salary) FROM employees);



```
mysql> SELECT first_name, last_name, salary
    -> FROM employees
    -> WHERE salary > (SELECT AVG(salary) FROM employees);
+------------+-----------+----------+
| first_name | last_name | salary   |
+------------+-----------+----------+
| John       | Doe       | 75000.00 |
| Sarah      | Wilson    | 80000.00 |
| Robert     | Miller    | 90000.00 |
+------------+-----------+----------+
3 rows in set (0.00 sec)
```

--8. SUBQUERY in SELECT clause

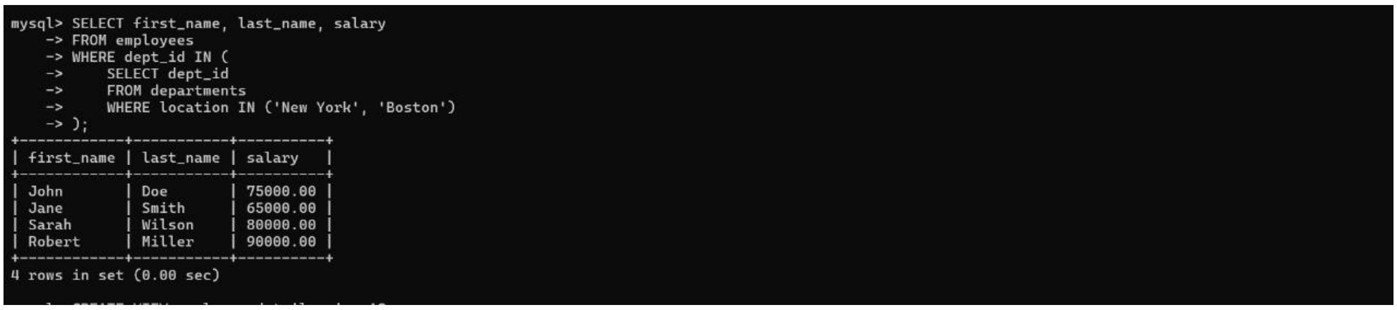
```
SELECT first_name, last_name, salary,
       (SELECT dept_name FROM departments d WHERE d.dept_id = e.dept_id) AS department_name
FROM employees e;
```



```
mysql> SELECT first_name, last_name, salary,
    ->        (SELECT dept_name FROM departments d WHERE d.dept_id = e.dept_id) AS department_name
    -> FROM employees e;
+------------+-----------+----------+-----------------+
| first_name | last_name | salary   | department_name |
+------------+-----------+----------+-----------------+
| John       | Doe       | 75000.00 | IT              |
| Jane       | Smith     | 65000.00 | IT              |
| Mike       | Johnson   | 55000.00 | HR              |
| Sarah      | Wilson    | 80000.00 | Finance         |
| David      | Brown     | 48000.00 | Marketing       |
| Emily      | Davis     | 52000.00 | NULL            |
| Robert     | Miller    | 90000.00 | Finance         |
+------------+-----------+----------+-----------------+
7 rows in set (0.01 sec)
```

--9. SUBQUERY with IN

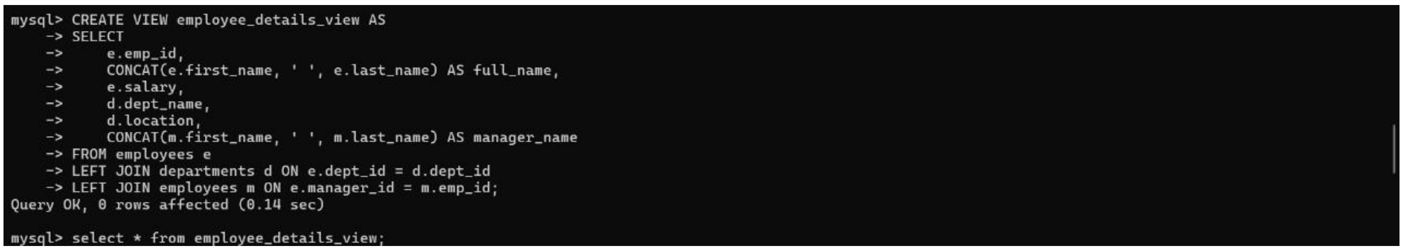
```
SELECT first_name, last_name, salary
FROM employees
WHERE dept_id IN (
   SELECT dept_id
   FROM departments
   WHERE location IN ('New York', 'Boston')
);
```



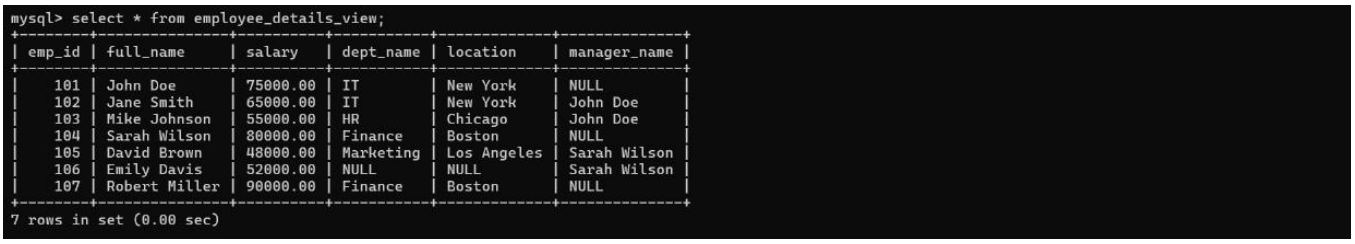
```
mysql> SELECT first_name, last_name, salary
    -> FROM employees
    -> WHERE dept_id IN (
    ->     SELECT dept_id
    ->     FROM departments
    ->     WHERE location IN ('New York', 'Boston')
    -> );
+------------+-----------+----------+
| first_name | last_name | salary   |
+------------+-----------+----------+
| John       | Doe       | 75000.00 |
| Jane       | Smith     | 65000.00 |
| Sarah      | Wilson    | 80000.00 |
| Robert     | Miller    | 90000.00 |
+------------+-----------+----------+
4 rows in set (0.00 sec)
```

--10. CREATE VIEW

```
CREATE VIEW employee_details_view AS
SELECT
   e.emp_id,
   CONCAT(e.first_name, ' ', e.last_name) AS full_name,
   e.salary,
   d.dept_name,
   d.location,
   CONCAT(m.first_name, ' ', m.last_name) AS manager_name
FROM employees e
LEFT JOIN departments d ON e.dept_id = d.dept_id
LEFT JOIN employees m ON e.manager_id = m.emp_id;
```



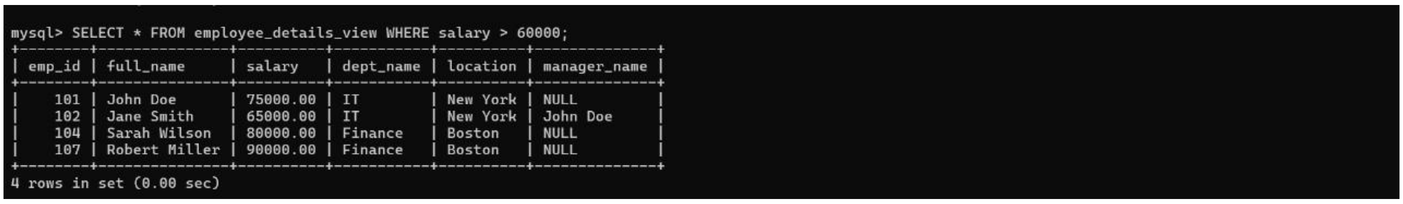
```
mysql> CREATE VIEW employee_details_view AS
    -> SELECT
    ->     e.emp_id,
    ->     CONCAT(e.first_name, ' ', e.last_name) AS full_name,
    ->     e.salary,
    ->     d.dept_name,
    ->     d.location,
    ->     CONCAT(m.first_name, ' ', m.last_name) AS manager_name
    -> FROM employees e
    -> LEFT JOIN departments d ON e.dept_id = d.dept_id
    -> LEFT JOIN employees m ON e.manager_id = m.emp_id;
Query OK, 0 rows affected (0.14 sec)

mysql> select * from employee_details_view;
```



```
mysql> select * from employee_details_view;
+--------+---------------+----------+-----------+-------------+--------------+
| emp_id | full_name     | salary   | dept_name | location    | manager_name |
+--------+---------------+----------+-----------+-------------+--------------+
|    101 | John Doe      | 75000.00 | IT        | New York    | NULL         |
|    102 | Jane Smith    | 65000.00 | IT        | New York    | John Doe     |
|    103 | Mike Johnson  | 55000.00 | HR        | Chicago     | John Doe     |
|    104 | Sarah Wilson  | 80000.00 | Finance   | Boston      | NULL         |
|    105 | David Brown   | 48000.00 | Marketing | Los Angeles | Sarah Wilson |
|    106 | Emily Davis   | 52000.00 | NULL      | NULL        | Sarah Wilson |
|    107 | Robert Miller | 90000.00 | Finance   | Boston      | NULL         |
+--------+---------------+----------+-----------+-------------+--------------+
7 rows in set (0.00 sec)
```

--11. Query using VIEW

SELECT * FROM employee_details_view WHERE salary > 60000;



```
mysql> SELECT * FROM employee_details_view WHERE salary > 60000;
+--------+---------------+----------+-----------+----------+--------------+
| emp_id | full_name     | salary   | dept_name | location | manager_name |
+--------+---------------+----------+-----------+----------+--------------+
|    101 | John Doe      | 75000.00 | IT        | New York | NULL         |
|    102 | Jane Smith    | 65000.00 | IT        | New York | John Doe     |
|    104 | Sarah Wilson  | 80000.00 | Finance   | Boston   | NULL         |
|    107 | Robert Miller | 90000.00 | Finance   | Boston   | NULL         |
+--------+---------------+----------+-----------+----------+--------------+
4 rows in set (0.00 sec)
```

--12. Complex JOIN with multiple tables

SELECT

    e.first_name,

    e.last_name,

    d.dept_name,

    p.project_name,

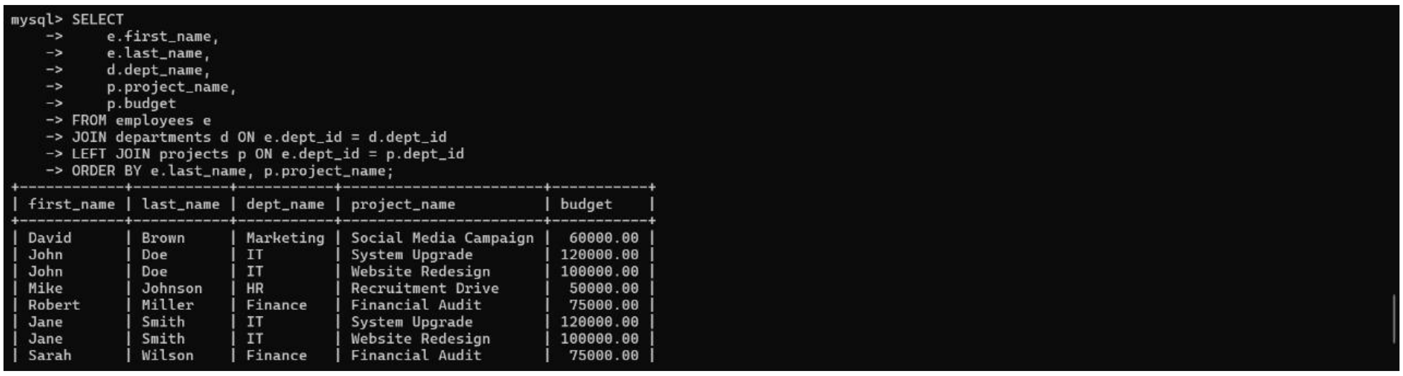
    p.budget

FROM employees e

JOIN departments d ON e.dept_id = d.dept_id

LEFT JOIN projects p ON e.dept_id = p.dept_id

ORDER BY e.last_name, p.project_name;



```
mysql> SELECT
    ->     e.first_name,
    ->     e.last_name,
    ->     d.dept_name,
    ->     p.project_name,
    ->     p.budget
    -> FROM employees e
    -> JOIN departments d ON e.dept_id = d.dept_id
    -> LEFT JOIN projects p ON e.dept_id = p.dept_id
    -> ORDER BY e.last_name, p.project_name;
+------------+-----------+-----------+-------------------------+-----------+
| first_name | last_name | dept_name | project_name            | budget    |
+------------+-----------+-----------+-------------------------+-----------+
| David      | Brown     | Marketing | Social Media Campaign   |  60000.00 |
| John       | Doe       | IT        | System Upgrade          | 120000.00 |
| John       | Doe       | IT        | Website Redesign        | 100000.00 |
| Mike       | Johnson   | HR        | Recruitment Drive       |  50000.00 |
| Robert     | Miller    | Finance   | Financial Audit         |  75000.00 |
| Jane       | Smith     | IT        | System Upgrade          | 120000.00 |
| Jane       | Smith     | IT        | Website Redesign        | 100000.00 |
| Sarah      | Wilson    | Finance   | Financial Audit         |  75000.00 |
```

# 5 Prac

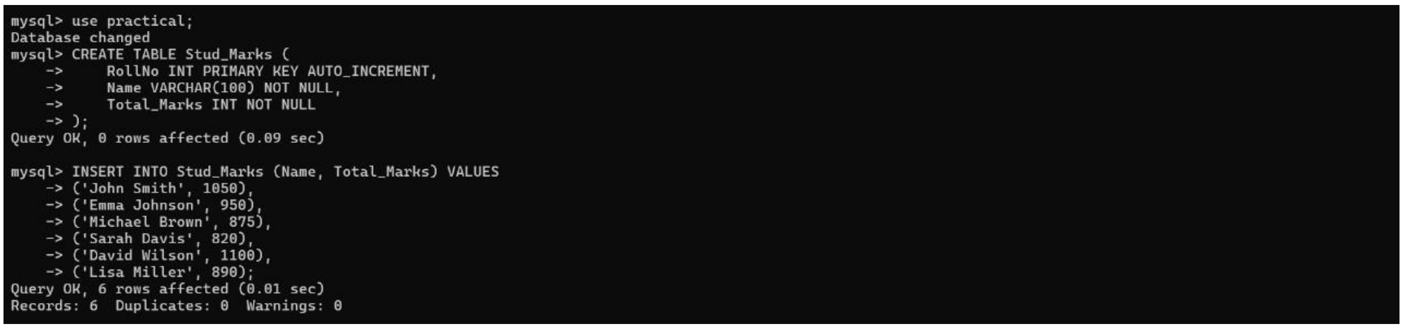
Named PL/SQL Block: PL/SQL Stored Procedure and Stored Function.

Write a Stored Procedure namely proc Grade for the categorization of student. If marks scored by students in examination is 1500 and marks> 990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks 899 and 825 category is Higher Second Class.

Write a PL/SQLblock to use procedure created with above requirement.

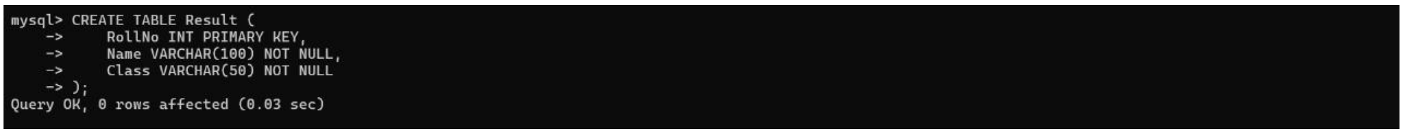
Stud Marks(name, total marks) Result (Roll, Name, Class)

```
-- Create the tables first

CREATE TABLE Stud_Marks (

    RollNo INT PRIMARY KEY AUTO_INCREMENT,

    Name VARCHAR(100) NOT NULL,

    Total_Marks INT NOT NULL

);

-- Insert sample data

INSERT INTO Stud_Marks (Name, Total_Marks) VALUES

('John Smith', 1050),

('Emma Johnson', 950),

('Michael Brown', 875),

('Sarah Davis', 820),

('David Wilson', 1100),

('Lisa Miller', 890);
```



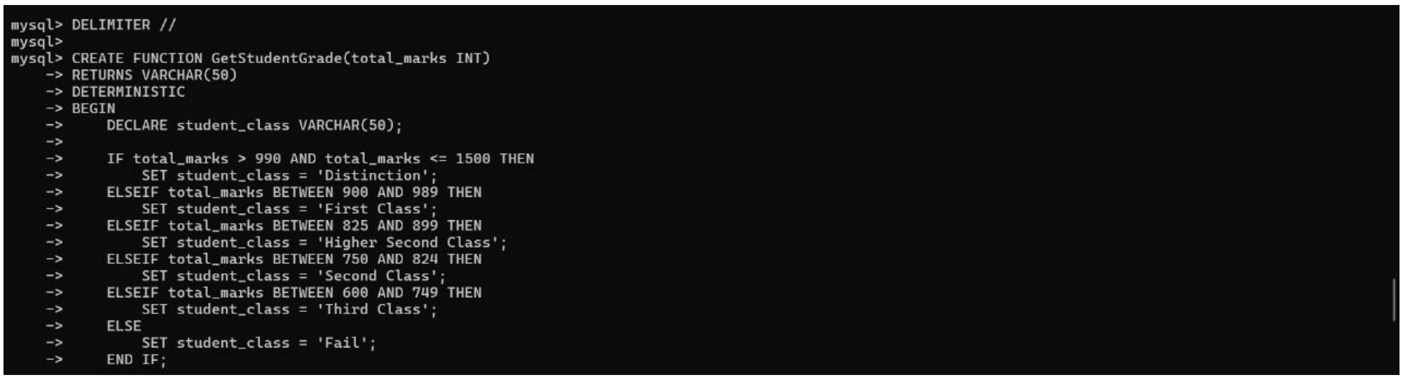
```
mysql> use practical;
Database changed
mysql> CREATE TABLE Stud_Marks (
    ->     RollNo INT PRIMARY KEY AUTO_INCREMENT,
    ->     Name VARCHAR(100) NOT NULL,
    ->     Total_Marks INT NOT NULL
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO Stud_Marks (Name, Total_Marks) VALUES
    -> ('John Smith', 1050),
    -> ('Emma Johnson', 950),
    -> ('Michael Brown', 875),
    -> ('Sarah Davis', 820),
    -> ('David Wilson', 1100),
    -> ('Lisa Miller', 890);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

```
CREATE TABLE Result (

    RollNo INT PRIMARY KEY,

    Name VARCHAR(100) NOT NULL,

    Class VARCHAR(50) NOT NULL

);
```

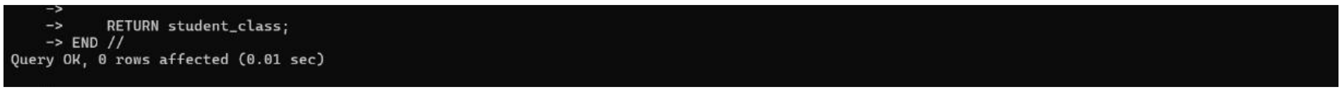


```
mysql> CREATE TABLE Result (
    ->     RollNo INT PRIMARY KEY,
    ->     Name VARCHAR(100) NOT NULL,
    ->     Class VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
DELIMITER //

CREATE FUNCTION GetStudentGrade(total_marks INT)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
  DECLARE student_class VARCHAR(50);

  IF total_marks > 990 AND total_marks <= 1500 THEN
    SET student_class = 'Distinction';
  ELSEIF total_marks BETWEEN 900 AND 989 THEN
    SET student_class = 'First Class';
  ELSEIF total_marks BETWEEN 825 AND 899 THEN
    SET student_class = 'Higher Second Class';
  ELSEIF total_marks BETWEEN 750 AND 824 THEN
    SET student_class = 'Second Class';
  ELSEIF total_marks BETWEEN 600 AND 749 THEN
    SET student_class = 'Third Class';
  ELSE
    SET student_class = 'Fail';
  END IF;

  RETURN student_class;
END //
```

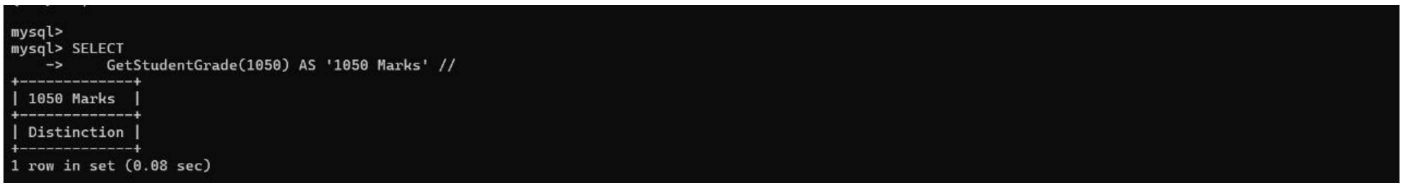


```
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION GetStudentGrade(total_marks INT)
    -> RETURNS VARCHAR(50)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE student_class VARCHAR(50);
    ->
    ->     IF total_marks > 990 AND total_marks <= 1500 THEN
    ->         SET student_class = 'Distinction';
    ->     ELSEIF total_marks BETWEEN 900 AND 989 THEN
    ->         SET student_class = 'First Class';
    ->     ELSEIF total_marks BETWEEN 825 AND 899 THEN
    ->         SET student_class = 'Higher Second Class';
    ->     ELSEIF total_marks BETWEEN 750 AND 824 THEN
    ->         SET student_class = 'Second Class';
    ->     ELSEIF total_marks BETWEEN 600 AND 749 THEN
    ->         SET student_class = 'Third Class';
    ->     ELSE
    ->         SET student_class = 'Fail';
    ->     END IF;
```



```
    ->
    ->     RETURN student_class;
    -> END //
Query OK, 0 rows affected (0.01 sec)
```

SELECT GetStudentGrade(1050) AS '1050 Marks'//



```
mysql>
mysql> SELECT
    ->     GetStudentGrade(1050) AS '1050 Marks' //
+-------------+
| 1050 Marks  |
+-------------+
| Distinction |
+-------------+
1 row in set (0.08 sec)
```

SELECT

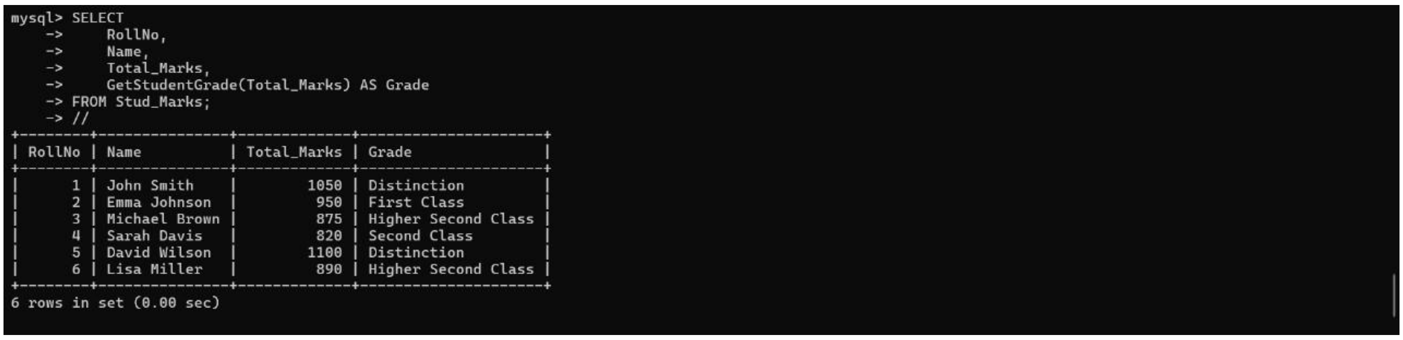
RollNo,

Name,

Total_Marks,

GetStudentGrade(Total_Marks) AS Grade

FROM Stud_Marks //



```
mysql> SELECT
    ->     RollNo,
    ->     Name,
    ->     Total_Marks,
    ->     GetStudentGrade(Total_Marks) AS Grade
    -> FROM Stud_Marks;
    -> //
+--------+---------------+-------------+---------------------+
| RollNo | Name          | Total_Marks | Grade               |
+--------+---------------+-------------+---------------------+
|      1 | John Smith    |        1050 | Distinction         |
|      2 | Emma Johnson  |         950 | First Class         |
|      3 | Michael Brown |         875 | Higher Second Class |
|      4 | Sarah Davis   |         820 | Second Class        |
|      5 | David Wilson  |        1100 | Distinction         |
|      6 | Lisa Miller   |         890 | Higher Second Class |
+--------+---------------+-------------+---------------------+
6 rows in set (0.00 sec)
```

# 7 Prac

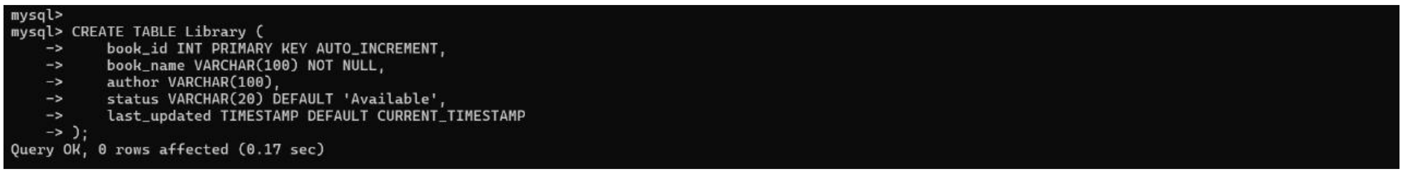
Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).

Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library Audit table.

Note: Instructor will Frame the problem statement for writing PL/SQLblock for all types of Triggers in line with above statement.

-- Create tables

```
CREATE TABLE Library (
    book_id INT PRIMARY KEY AUTO_INCREMENT,
    book_name VARCHAR(100) NOT NULL,
    author VARCHAR(100),
    status VARCHAR(20) DEFAULT 'Available',
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

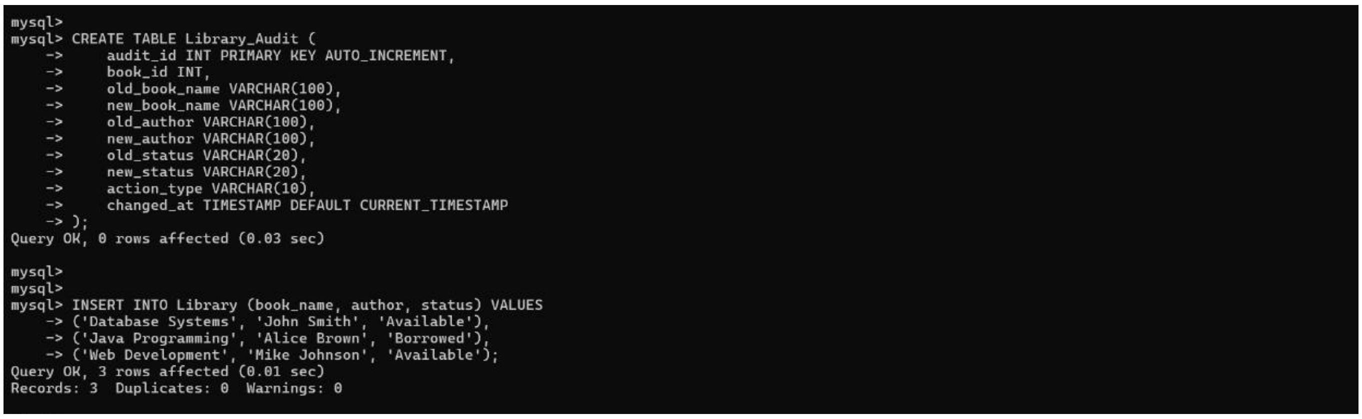


```
mysql>
mysql> CREATE TABLE Library (
    ->     book_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     book_name VARCHAR(100) NOT NULL,
    ->     author VARCHAR(100),
    ->     status VARCHAR(20) DEFAULT 'Available',
    ->     last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.17 sec)
```

```
CREATE TABLE Library_Audit (
    audit_id INT PRIMARY KEY AUTO_INCREMENT,
    book_id INT,
    old_book_name VARCHAR(100),
    new_book_name VARCHAR(100),
    old_author VARCHAR(100),
    new_author VARCHAR(100),
    old_status VARCHAR(20),
    new_status VARCHAR(20),
    action_type VARCHAR(10),
    changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

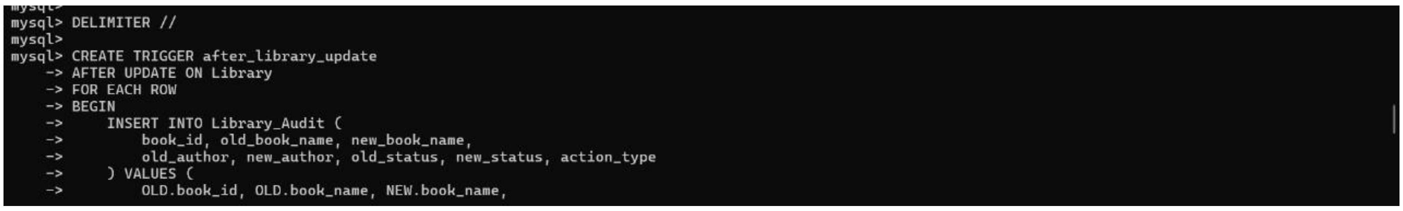
-- Insert sample data

```
INSERT INTO Library (book_name, author, status) VALUES
('Database Systems', 'John Smith', 'Available'),
('Java Programming', 'Alice Brown', 'Borrowed'),
('Web Development', 'Mike Johnson', 'Available');
```

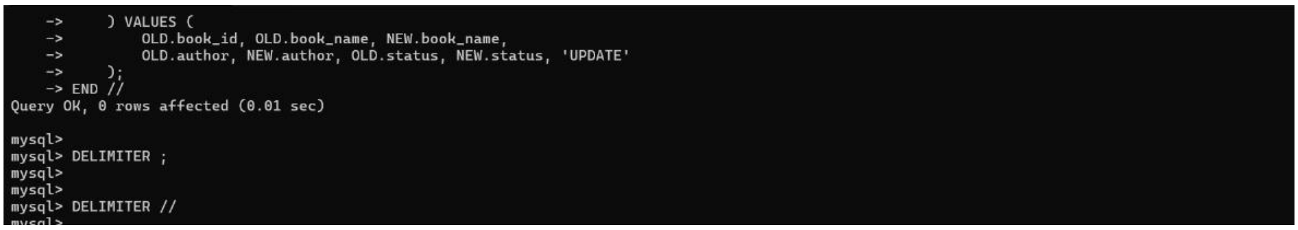


```
mysql>
mysql> CREATE TABLE Library_Audit (
    ->     audit_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     book_id INT,
    ->     old_book_name VARCHAR(100),
    ->     new_book_name VARCHAR(100),
    ->     old_author VARCHAR(100),
    ->     new_author VARCHAR(100),
    ->     old_status VARCHAR(20),
    ->     new_status VARCHAR(20),
    ->     action_type VARCHAR(10),
    ->     changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql>
mysql> INSERT INTO Library (book_name, author, status) VALUES
    -> ('Database Systems', 'John Smith', 'Available'),
    -> ('Java Programming', 'Alice Brown', 'Borrowed'),
    -> ('Web Development', 'Mike Johnson', 'Available');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
-- Create UPDATE trigger

DELIMITER //


CREATE TRIGGER after_library_update

AFTER UPDATE ON Library

FOR EACH ROW

BEGIN

    INSERT INTO Library_Audit (

        book_id, old_book_name, new_book_name,

        old_author, new_author, old_status, new_status, action_type

    ) VALUES (

        OLD.book_id, OLD.book_name, NEW.book_name,

        OLD.author, NEW.author, OLD.status, NEW.status, 'UPDATE'

    );

END //


DELIMITER ;
```



```
mysql> DELIMITER //
mysql>
mysql> CREATE TRIGGER after_library_update
    -> AFTER UPDATE ON Library
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Library_Audit (
    ->         book_id, old_book_name, new_book_name,
    ->         old_author, new_author, old_status, new_status, action_type
    ->     ) VALUES (
    ->         OLD.book_id, OLD.book_name, NEW.book_name,
```



```
    ->     ) VALUES (
    ->         OLD.book_id, OLD.book_name, NEW.book_name,
    ->         OLD.author, NEW.author, OLD.status, NEW.status, 'UPDATE'
    ->     );
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
mysql>
mysql> DELIMITER //
```

```
-- Create DELETE trigger

DELIMITER //
```

CREATE TRIGGER after_library_delete

AFTER DELETE ON Library

FOR EACH ROW

BEGIN

    INSERT INTO Library_Audit (

        book_id, old_book_name, old_author, old_status, action_type

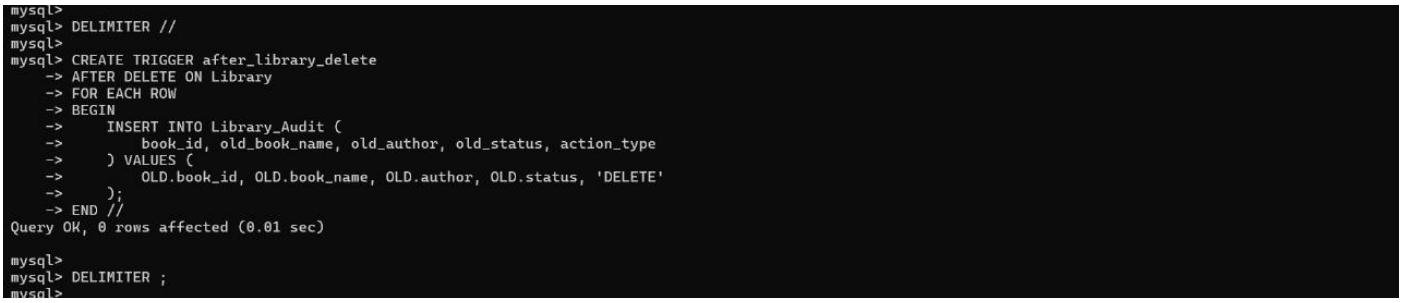
    ) VALUES (

        OLD.book_id, OLD.book_name, OLD.author, OLD.status, 'DELETE'

    );

END //

DELIMITER ;

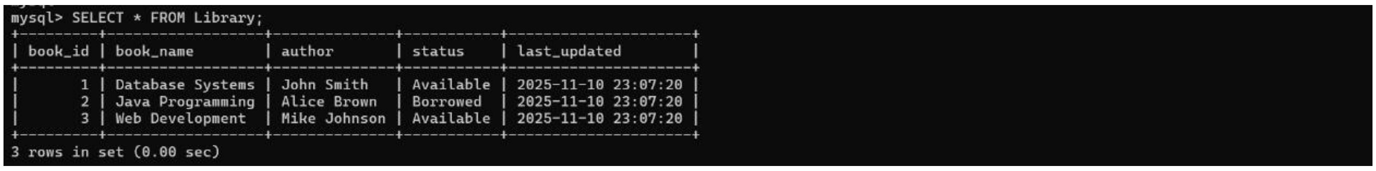


```
mysql>
mysql> DELIMITER //
mysql>
mysql> CREATE TRIGGER after_library_delete
    -> AFTER DELETE ON Library
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Library_Audit (
    ->         book_id, old_book_name, old_author, old_status, action_type
    ->     ) VALUES (
    ->         OLD.book_id, OLD.book_name, OLD.author, OLD.status, 'DELETE'
    ->     );
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

-- Test the triggers

SELECT '=== BEFORE ANY CHANGES ===' AS '';

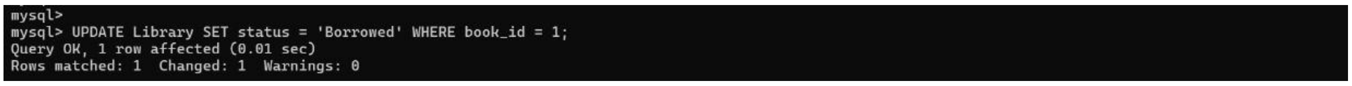
SELECT * FROM Library;



```
mysql> SELECT * FROM Library;
+---------+------------------+--------------+-----------+---------------------+
| book_id | book_name        | author       | status    | last_updated        |
+---------+------------------+--------------+-----------+---------------------+
|       1 | Database Systems | John Smith   | Available | 2025-11-10 23:07:20 |
|       2 | Java Programming | Alice Brown  | Borrowed  | 2025-11-10 23:07:20 |
|       3 | Web Development  | Mike Johnson | Available | 2025-11-10 23:07:20 |
+---------+------------------+--------------+-----------+---------------------+
3 rows in set (0.00 sec)
```

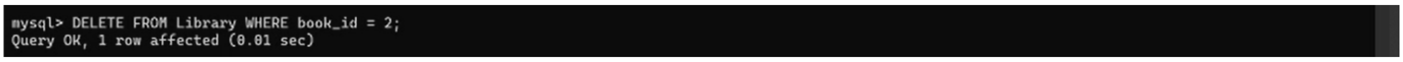
-- Perform operations that will trigger the triggers

UPDATE Library SET status = 'Borrowed' WHERE book_id = 1;



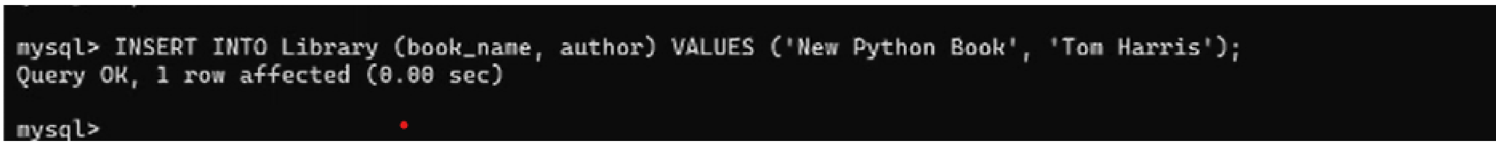
```
mysql>
mysql> UPDATE Library SET status = 'Borrowed' WHERE book_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

DELETE FROM Library WHERE book_id = 2;



```
mysql> DELETE FROM Library WHERE book_id = 2;
Query OK, 1 row affected (0.01 sec)
```

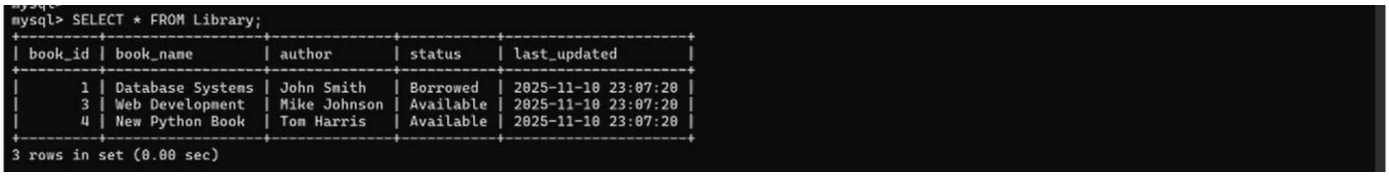
INSERT INTO Library (book_name, author) VALUES ('New Python Book', 'Tom Harris');



```
mysql> INSERT INTO Library (book_name, author) VALUES ('New Python Book', 'Tom Harris');
Query OK, 1 row affected (0.00 sec)

mysql>
```

SELECT '=== AFTER CHANGES ===' AS '';

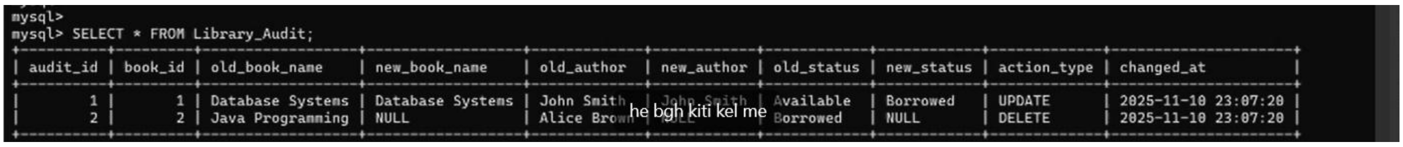
SELECT * FROM Library;



```
mysql> SELECT * FROM Library;
+---------+-----------------+--------------+-----------+---------------------+
| book_id | book_name       | author       | status    | last_updated        |
+---------+-----------------+--------------+-----------+---------------------+
|       1 | Database Systems | John Smith  | Borrowed  | 2025-11-10 23:07:20 |
|       3 | Web Development | Mike Johnson | Available | 2025-11-10 23:07:20 |
|       4 | New Python Book | Tom Harris   | Available | 2025-11-10 23:07:20 |
+---------+-----------------+--------------+-----------+---------------------+
3 rows in set (0.00 sec)
```

SELECT '=== AUDIT TRAIL SHOWING ALL CHANGES ===' AS '';

SELECT * FROM Library_Audit;



```
mysql>
mysql> SELECT * FROM Library_Audit;
+----------+---------+------------------+------------------+-------------+------------+------------+------------+-------------+---------------------+
| audit_id | book_id | old_book_name    | new_book_name    | old_author  | new_author | old_status | new_status | action_type | changed_at          |
+----------+---------+------------------+------------------+-------------+------------+------------+------------+-------------+---------------------+
|        1 |       1 | Database Systems | Database Systems | John Smith  | John Smith | Available  | Borrowed   | UPDATE      | 2025-11-10 23:07:20 |
|        2 |       2 | Java Programming | NULL             | Alice Brown | NULL       | Borrowed   | NULL       | DELETE      | 2025-11-10 23:07:20 |
+----------+---------+------------------+------------------+-------------+------------+------------+------------+-------------+---------------------+
```

he bgh kiti kel me

## MonqoDB Queries

# Practical 9

Design and Develop MonqoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.)

```
> show dbs
< admin  40.00 KiB
  config  12.00 KiB
  local  40.00 KiB

> use Aaku
< switched to db Aaku
> db.student.inse# Practical No. 9rt(['Rollno':'1', 'Name':'Aaku', 'Class':'TE COMP']);
< {
  acknowledgcd: true,
  insertedIds: {
  '@': ObjectId('68e00d24765c378352d35e3b')
  }
}

> db.student.insert(['Rollno':'2', 'Name':'Miku', 'Class':'TE COMP']);
< {
  acknowledgcd: true,
    insertedIds: {
    '@': ObjectId('68e00d24765c378352d35e3b')
    }
}

> db.student.insert(['Rollno':'4', 'Name':'radha', 'Class':'TE COMP']);
< {
    acknowledgcd: true,
    insertedIds: {
    '@': ObjectId('68e00d48765c378352d35e3b')
    }
}

> db.student.insert(['Rollno':'5', 'Name':'rutu', 'Class':'TE COMP']);
< {
    acknowledgcd: true,
    insertedIds: {
    '@': ObjectId('68e00d56765c378352d35e3a')
    }
}

> db.student.insert(['Rollno':'6', 'Name':'praju', 'Class':'TE COMP']);
< {
    acknowledgcd: true,
    insertedIds: {
    '@': ObjectId('68e00d66765c378352d35e3b')
    }
}

` db.student.find(); `

` { _id: ObjectId('68e90cdb765c378352d35e35'), Rollno: '1', Name: 'Aaku', Class: 'TE COMP' } `

` _id: ObjectId('68e90cf4765c378352d35e36'), Rollno: '1', Name: 'Aaku', Class: 'TE COMP' } `

` _id: ObjectId('68e90d24765c378352d35e37'), Rollno: '2', Name: 'Miku', Class: 'TE COMP' } `

` _id: ObjectId('68e90d35765c378352d35e38'), Rollno: '3', Name: 'shradha', Class: 'TE COMP' } `
```

` _id: ObjectId('68e90d48765c378352d35e39'), Rollno: '4', Name: 'radha', Class: 'TE COMP' } `

` _id: ObjectId('68e90d56765c378352d35e3a'), Rollno: '5', Name: 'rutu', Class: 'TE COMP' } `

` _id: ObjectId('68e90d66765c378352d35e3b'), Rollno: '6', Name: 'praju', Class: 'TE COMP' } `

` db.student.find().pretty(); `

` { _id: ObjectId('68e90cdb765c378352d35e35'), Rollno: '1', Name: 'Aaku', Class: 'TE COMP' }

[ _id: ObjectId('68e@0e64765c378352d35e36'), Rollno: '1', Name: 'Aaku', Class: 'TE COMP' ] { _id: ObjectId('68e@0d24765c378352d35e37'), Rollno: '2', Name: 'Miku', Class: 'TE COMP' ] { _id: ObjectId('68e@0d35765c378352d35e38'), Rollno: '3', Name: 'shradha', Class: 'TE COMP' ] { _id: ObjectId('68e@0d48765c378352d35e39'), Rollno: '4', Name: 'radha', Class: 'TE COMP' ] { _id: ObjectId('68e@0d56765c378352d35e3a'), Rollno: '5', Name: 'rutu', Class: 'TE COMP' ] { _id: ObjectId('68e@0d66765c378352d35e3b'), Rollno: '6', Name: 'praju', Class: 'TE COMP' ] show dbs; Aaku 64.00 KiB admin 40.00 KiB config 92.00 KiB local 40.00 KiB db.Student.update(['Name':'praju'],{$set:['Name':'simran']}); DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkwrite. { acknowledged: true, insertedId: null, matchedCount: 0, modifiedCount: 0, upsertedCount: 0 }

```
> db.student.find().pretty();

< {

  _id: ObjectId('68e@0edb765c378352d35e38'),
  Rollno: '1',
  Name: 'Aaku',
  Class: 'TE COMP'

}

{
  _id: ObjectId('68e@0ef4765c378352d35e36'),
  Rollno: '1',
  Name: 'Aaku',
  Class: 'TE COMP'

}

{
  _id: ObjectId('68e@0d24765c378352d35e37'),
  Rollno: '2',
  Name: 'Miku',
  Class: 'TE COMP'

}

{
  _id: ObjectId('68e@0d35765c378352d35e38'),
  Rollno: '3',
  Name: 'shradha',
  Class: 'TE COMP'

}

{
  _id: ObjectId('68e@0d48765c378352d35e39'),
  Rollno: '4',
  Name: 'radha',
  Class: 'TE COMP'

}

{
  _id: ObjectId('68e@0d56765c378352d35e38'),
  Rollno: '5',
  Name: 'rutu',
  Class: 'TE COMP'

}

{
  _id: ObjectId('68e@0d66765c378352d35e39'),
  Rollno: '6',
  Name: 'praju',
  Class: 'TE COMP'

}
```

```
db.Student.remove(['Name': 'Miku'],]);

DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulWrite.

{

  acknowledged: true,
  deletedCount: 0

> db.student.find().pretty();

< {

 _id: ObjectId('6aee0cdb765c378352d35e3s'),
  Rollno: '1',
  Name: 'Aaku',
  Class: 'TE COMP'

}

{
 _id: ObjectId('6aee0cfa765c378352d35e3s'),
  Rollno: '1',
  Name: 'Aaku',
  Class: 'TE COMP'

}

{
 _id: ObjectId('6aee0d24765c378352d35e37'),
  Rollno: '2',
  Name: 'Niku',
  Class: 'TE COMP'

}

{
 _id: ObjectId('6aee0d35765c378352d35e3s'),
  Rollno: '3',
  Name: 'shradha',
  Class: 'TE COMP'

}

{
 _id: ObjectId('6aee0d48765c378352d35e39'),
  Rollno: '4',
  Name: 'radha',

Class: 'TE COMP'

}

{
 _id: ObjectId('6aee0d56765c378352d35e3a'),
  Rollno: '5',
  Name: 'rutu',
  Class: 'TE COMP'

}

{
 _id: ObjectId('6aee0d66765c378352d35e3b'),
  Rollno: '6',
  Name: 'pragu',
  Class: 'TE COMP'

}

db.Student.drop();true
true
db.Student.drop();
true
```

# # Practical No. 10

MongoDB - Aggregation and Indexing:  Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.

```
> use Aaku;

< switched to db Aaku

> db.createCollection('student');

< [ ok; ]

> db.Student.insert(['Rno':'1', 'Name':'Aaku','Class':'TE COMP']);

< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or butWrite.

< {

   acknowledged: true,

   insertedIds: {

   '0': ObjectId('68e0154457da2524c90c6d9e')

   }

   }

> db.Student.insert(['Rno':'2', 'Name':'niku','Class':'TE COMP']);

< {

   acknowledged: true,

   insertedIds: {

   '0': ObjectId('68e0154957da2524c90c6d9f')

   }

   }

> db.Student.insert(['Rno':'3', 'Name':'radha','Class':'TE COMP']);

< {

   acknowledged: true,

   insertedIds: {

   '0': ObjectId('68e0158957da2524c90c6da0')

   }

   }

> db.Student.insert(['Rno':'4', 'Name':'rutu','Class':'TE COMP']);

< {

   acknowledged: true,

   insertedIds: {

   '0': ObjectId('68e0159657da2524c90c6da1')
```

```
    }
    }
> db.Student.insert(['Rno':'5', 'Name':'shradha','Class':'TE COMP']);
< {
    acknowledged: true,
    insertedIds: {
    '0': ObjectId('68e015b357da2524c90c6da2')
    }
    }
> db.Student.insert(['Rno':'6', 'Name':'sonu','Class':'TE COMP']);
< {
    acknowledged: true,
    insertedIds: {
    '0': ObjectId('68e015c157da2524c90c6da3')
    }
    }
> db.Student.find();
< {
    _id: ObjectId('68e0154457da2524c90c6d9e'),
    Rno: '1',
    }

Name: 'Aaku',
Class: 'TE COMP'
}
{
    _id: ObjectId('68e6157857da2524c99cc6d9f'),
    Rno: '2',
    Name: 'niku',
    Class: 'TE COMP'
}

{
    _id: ObjectId('68e6158957da2524c99cc6a0'),
    Rno: '3',
    Name: 'radha',
    Class: 'TE COMP'
}
```

```
{
   _id: ObjectId('68e6159657da2524c99cc6da1'),
   Rno: '4',
   Name: 'rutu',
   Class: 'TE COMP'
}

{
   _id: ObjectId('68e6159357da2524c99cc6a21'),
   Rno: '5',
   Name: 'shradha',
   Class: 'TE COMP'
}

{
   _id: ObjectId('68e6154157da2524c99cc6a31'),
   Rno: '6',
   Name: 'sonu',
   Class: 'TE COMP'
}

db.Student.find().pretty();
{
   _id: ObjectId('68e6154457da2524c99cc6d9e'),
   Rno: '1',
   Name: 'Aaku',
   Class: 'TE COMP'
}

{
   _id: ObjectId('68e6157857da2524c99cc6d9f'),
   Rno: '2',
   Name: 'niku',
   Class: 'TE COMP'
}

{
    _id: ObjectId('68e6158957da2524c99cc6a0'),
    Rno: '3',
    Name: 'radha',
    Class: 'TE COMP'

}

{

_id: ObjectId('68e@15be57da2524c99c6da1'),
Rno: '4',
Name: 'rutu',
Class: 'TE COMP'

}

{

_id: ObjectId('68e@15b357da2524c99c6da2'),
Rno: '5',
Name: 'shradha',
Class: 'TE COMP'

}

{

_id: ObjectId('68e@15c157da2524c99c6da3'),
Rno: '6',
Name: 'sonu',
Class: 'TE COMP'

}

} db.Student.update(['Name':'sonu'],($set: {'Name':'monu'}));
< DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
< {
    acknowledged: true,
   insertedId: null,

   matchedCount: 1,

   modifiedCount: 1,

   upsertedCount: 0

}

> db.Student.find().pretty();

< {

   _id: ObjectId('68e@154457da2524c99c6d9e'),

Rno: '1',

Name: 'Aaku',

Class: 'TE COMP'

}

{

   _id: ObjectId('68e@157857da2524c99c6d9f'),

Rno: '2',

Name: 'niku',

Class: 'TE COMP'

}

{

   _id: ObjectId('68e@158957da2524c99c6da8'),

Rno: '3',

Name: 'radha',

Class: 'TE COMP'

}

}


[

 ]

 {

 _id: ObjectId('68e@159e57da2524c99c6da1'),

 Rno: '4',

 Name: 'rutu',

 Class: 'TE COMP'

 }

 {

 _id: ObjectId('68e@159e57da2524c99c6da2'),

 Rno: '5',

 Name: 'shradha',
 Class: 'TE COMP'

 }

 {

 _id: ObjectId('68e@156157da2524c99c6da3'),

 Rno: '6',

 Name: 'monu',

 Class: 'TE COMP'

 }

 > db.Student.remove(['Name':'niku']);

 < {

 _acknowledged: true,

 _deletedCount:

 }

 > db.Student.find().pretty();

 < {

 _id: ObjectId('68e@154457da2524c99c6d9e'),

 Rno: '1',

 Name: 'Aaku',

 Class: 'TE COMP'

 }

 {

 _id: ObjectId('68e@158957da2524c99c6de0'),

 Rno: '3',

 Name: 'radha',

 Class: 'TE COMP'

 }

 {

 _id: ObjectId('68e@159e57da2524c99c6da1'),

 Rno: '4',

 Name: 'rutu',

 Class: 'TE COMP'

 }

 {

 _id: ObjectId('68e@159e57da2524c99c6da2'),

 Rno: '5',

 Name: 'shradha',

 Class: 'TE COMP'

 }
```

```
{
_id: ObjectId('68e@156157da2524c99c6da3'),
Rno: '6',
Name: 'monu',
Class: 'TE COMP'
}
```

# # Practical No. 11

MongoDB -- Map-reduces operations:

Implement Map reduces operation with suitable example using MongoDB.

```
> use Aaku;
< switched to db Aaku
> db.createCollection('website');
< { ok; }
> <db.website.insert(['roll':'1','name':'Aaku','amount':5800,'url':'www.yahoo.com']);
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkwrite.
< {
   acknowledged: true,
   insertedIds: {
   '0': ObjectId('68e91c448d55562cf00426f')
   }
> db.website.insert(['roll':'2','name':'niku','amount':4200,'url':'www.yahoo.com']);
< {
   acknowledged: true,
   insertedIds: {
   '0': ObjectId('68e91d7289d55562cf00426f')
   }
> db.website.insert(['roll':'3','name':'shradha','amount':9850,'url':'www.google.com']);
< {
   acknowledged: true,
   insertedIds: {
```

```
   '0': ObjectId('68e91e1b89d55562cf00426f')
   }
> db.website.insert(['roll':'4','name':'radha','amount':7581,'url':'www.gmail.com']);
< {
   acknowledged: true,
   insertedIds: {
   '0': ObjectId('68e91e5489d55562cf00426f')
   }
> db.website.insert(['roll':'5','name':'rutu','amount':3215,'url':'www.org.com']);
< {
   acknowledged: true,
   insertedIds: {
   '0': ObjectId('68e91ffc89d55562cf00426f')
   }
> db.website.aggregate({$group:{_id:"$name","total":{$sum:"$amount"}}});
< {
   _id: 'niku',
   total: 4200
   }
> {
   _id: 'Aaku',
   total: 5800
   }
> }
```

-

-id: 'niku', total: 4260 ) { _id: 'Aaku', total: 5860 } { _id: 'shradha', total: 9850 } { _id: 'rutu', total: 3215 } { _id: 'radha', total: 7581 } db.website.aggregate({$group:{_id:'$name','total':{${irst:'$amount'}}}); { _id: 'radha', total: 7581 } { _id: 'niku', total: 4260 } { _id: 'Aaku', total: 5860 } { _id: 'shradha', total: 9850 } { _id: 'rutu', total: 3215 } db.website.aggregate({$group:{_id:'$name','total':{${last:'$amount'}}}); { _id: 'shradha', total: 9850 } { _id: 'Aaku', total: 5860 } { _id: 'niku', total: 4260

}

{

__id: 'rutu',

total: 3215

```
}

{

__id: 'radha',

total: 7581

}

db.website.aggregate({$group:(_id:'$name','total':{$push:'$amount'}}});

{

__id: 'niku',

total: [

   4200

]

{

__id: 'Aaku',

total: [

   5800

]

{

__id: 'shradha',

total: [
   9850
]
{
__id: 'rutu',
total: [
   3215
]
{
__id: 'radha',
total: [
   7581
]
} db.website.aggregate({$group:(_id:'$name','total':{$sum:1})});
< {
__id: 'radha',
total: 1
}
}
_id: 'shradha',
total: 1

}

_id: 'Aaku',

total: 1

{

__id: 'niku',
total: 1
}

{
__id: 'rutu',
total: 1
}

} db.website.aggregate({$group:{_id:'$name','total':{$addToSet:'$amount'}}});
< {
__id: 'radha',
total: [
   7581
   ]
}
{
__id: 'niku',
total: [
   4260
   ]
}
{
__id: 'Aaku',
total: [
   5800
   ]
```

```
}
{
__id: 'shradha',
total: [
   9850
   ]
}
{
__id: 'rutu',
total: [
   3215
   ]
} db.createCollection('website1');
< { ok: 1 }
} db.website1.insert(['r':1,'name':'Aniket'])
< {
   acknowledged: true,
   insertedIds: {
   '@': ObjectId('68e0232889d55562cf004269')
   }
}
} db.website1.find().pretty()
< {
__id: ObjectId('68e0232889d55562cf004269'),
   r: 1,
   name: 'Aniket'
   }
} db.website.dropIndex(['name':-1])
```