

Practical 7- Consider the following database
Project(project_id,proj_name,chief_arch) , project_id is primary key
Employee(Emp_id,Emp_name) , Emp_id is primary key
Assigned-To(Project_id,Emp_id)
Find the SQL queries for the following:
1. Get the details of employees working on project C353
2. Get employee number of employees working on project C353
3. Obtain details of employees working on Database project
4. Get details of employees working on both C353 and C354
5. Get employee numbers of employees who do not work on project C453

```
C:\Users\ACER>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.39 MySQL Community Server - GPL
```

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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

```
mysql> CREATE DATABASE ProjectManagement;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> USE ProjectManagement;
Database changed
mysql> CREATE TABLE Project (
    project_id VARCHAR(10) PRIMARY KEY,
    proj_name VARCHAR(50),
    chief_arch VARCHAR(50)
);
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TABLE Employee (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(50)
);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE TABLE Assigned_To (
    Use project_id VARCHAR(10),
    emp_id INT,
    FOREIGN KEY (project_id) REFERENCES Project(project_id),
    FOREIGN KEY (emp_id) REFERENCES Employee(emp_id)
);
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> INSERT INTO Project (project_id, proj_name, chief_arch) VALUES
    ('C353', 'Database Project', 'John Doe'),
    ('C354', 'AI Project', 'Jane Smith'),
    ('C453', 'ML Project', 'Alice Johnson');
Query OK, 3 rows affected (0.02 sec)
```

Records: 3 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO Employee (emp_id, emp_name) VALUES
      (101, 'Alice'),
      (102, 'Bob'),
      (103, 'Charlie'),
      (104, 'David');
```

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO Assigned_To (project_id, emp_id) VALUES
      ('C353', 101),
      ('C353', 102),
      ('C354', 101),
      ('C354', 103),
      ('C453', 104);
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> SELECT E.emp_id, E.emp_name
      FROM Employee E
      JOIN Assigned_To A ON E.emp_id = A.emp_id
      WHERE A.project_id = 'C353';
```

```
+-----+-----+
| emp_id | emp_name |
+-----+-----+
|    101 | Alice    |
|    102 | Bob      |
+-----+-----+
2 rows in set (0.01 sec)
```

```
mysql> SELECT E.emp_id
      FROM Employee E
      JOIN Assigned_To A ON E.emp_id = A.emp_id
      WHERE A.project_id = 'C353';
```

```
+-----+
| emp_id |
+-----+
|    101 |
|    102 |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT E.emp_id, E.emp_name
      FROM Employee E
      JOIN Assigned_To A ON E.emp_id = A.emp_id
      JOIN Project P ON A.project_id = P.project_id
      WHERE P.proj_name = 'Database Project';
```

```
+-----+-----+
| emp_id | emp_name |
+-----+-----+
|    101 | Alice    |
|    102 | Bob      |
+-----+-----+
2 rows in set (0.01 sec)
```

```
mysql> SELECT E.emp_id, E.emp_name
      FROM Employee E
      WHERE E.emp_id IN (
```

```

        SELECT emp_id FROM Assigned_To WHERE project_id = 'C353'
    ) AND E.emp_id IN (
        SELECT emp_id FROM Assigned_To WHERE project_id = 'C354'
    );
+-----+-----+
| emp_id | emp_name |
+-----+-----+
|    101 | Alice    |
+-----+-----+
1 row in set (0.01 sec)

```

```

mysql>
mysql> SELECT E.emp_id
        FROM Employee E
        WHERE E.emp_id NOT IN (
            SELECT emp_id FROM Assigned_To WHERE project_id = 'C453'
        );
+-----+
| emp_id |
+-----+
|    101 |
|    102 |
|    103 |
+-----+
3 rows in set (0.01 sec)

```

Theory:

Explanation of Each Query:

Query 1: Joins Employee and Assigned_To tables to get employee details for project C353.

Query 2: Similar to Query 1 but only selects employee IDs for project C353.

Query 3: Joins all three tables (Employee, Assigned_To, Project) to get employees working on the "Database Project."

Query 4: Uses a nested query to find employees working on both projects C353 and C354.

Query 5: Uses a nested query with NOT IN to find employees not assigned to project C453.