Name:- Rushikesh Karbhari Palve
Roll NO. 31258

: Date : / /20 | Page No :

Assignment NO. 3

DOP:- 03-09-2021 DOS:- 11-09-2021

Title:- SQL Queries of All types of Join.

Problem Definition:-

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

Learning Objectives:-

(i) To understand types of Joins.
(ii) To understand subquery and its types.
(iii) To understand complex views.

Learning Outcomes:-

Students will be able to -
(i) Understand types of Joins
(ii) Understand subquery and its types.

Theory:-

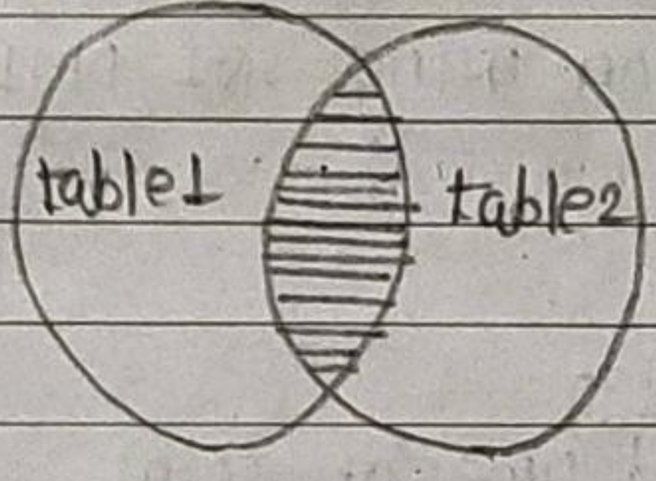
The ability of relational join operator is an important feature of relational system. A join makes it possible to select data from more than talk by means of single statement. This joining of tables may be done in many ways.

Types of JOIN :-

1. Inner
2. Outer (left, right, full)
3. Cross

1 INNER JOIN :-

The INNER JOIN keyword selects records that have matching values in both tables.



INNER JOIN

Syntax :-

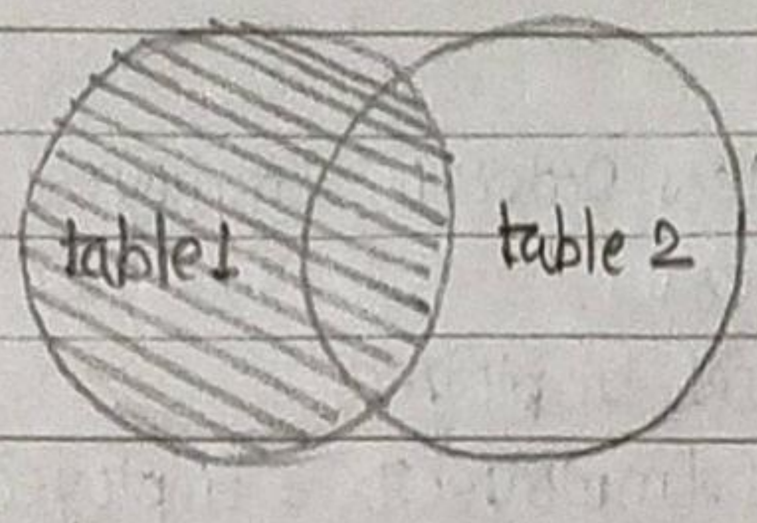
```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

Example :-

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;
```

2 LEFT JOIN :-

The LEFT JOIN keyword returns all records from the left table and the matching records from the right table.



LEFT JOIN

Syntax :-

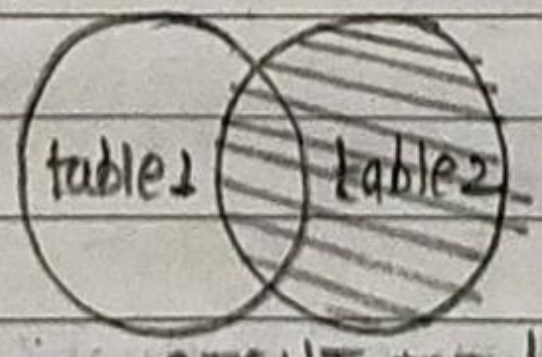
```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.columnName = table2.Column_name
```

Example :-

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;
```

③ RIGHT JOIN :-

The RIGHT JOIN keyword returns all records from the right table and the matching records from the left table.



RIGHT JOIN

Syntax :-

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.Column_name = table2.column_name;
```

Example :-

```
SELECT Orders. OrderID , Employees . LastName
FROM Orders
RIGHT JOIN Employees
ON Orders . EmployeeID = Employees . EmployeeID
ORDER BY Orders . OrderID ;
```

④ FULL JOIN :-

FULL JOIN lets you combine both LEFT JOIN and RIGHT JOIN into single compilation .

Syntax :-

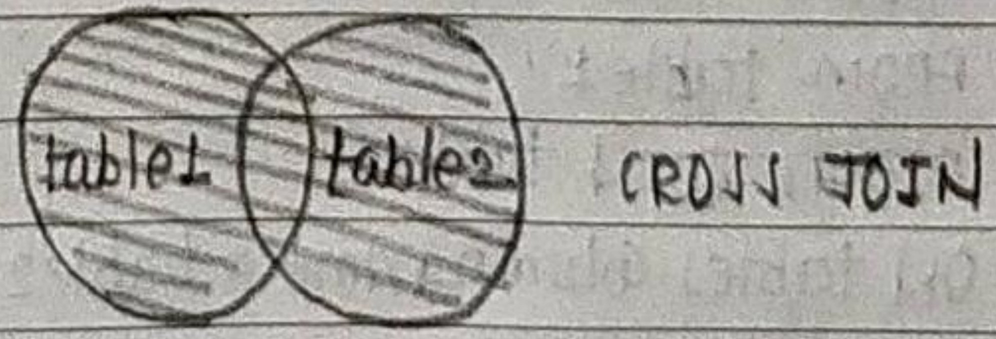
```
SELECT column_name(s)
FROM table1
FULL JOIN table2
ON table1 . Column_name = table2 . Column_name ;
```

Example :-

```
SELECT s1.id , s1.name , s2.stream
FROM student_name AS s1
FULL JOIN student_stream AS s2
ON s1.id = s2.id ;
```

⑤ CROSS JOIN :-

The CROSS JOIN keyword returns all records from both tables .



Syntax:-

```
SELECT column_name(s)
FROM table1
CROSS JOIN table2 ;
```

Example:-

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
CROSS JOIN Orders ;
```

6 NATURAL JOIN :-

When we combine rows of two or more tables based on a common column between them, this operation is called joining. A Natural Join is a type of Join operation that creates an implicit join by combining tables based on columns with the same name and data type. It is similar to the INNER or LEFT JOIN. But we cannot use the ON or USING clause with Natural Join as we used in them.

Syntax:-

```
SELECT Column_name(s)
FROM table1
NATURAL JOIN table2 ;
```

Example:-

```
SELECT Customer.Customer_name, Balance.balance
FROM Customer
NATURAL JOIN Balance ;
```

Group By statement :-

The GROUP BY statement groups rows that have the same values into summary rows, like find the number of customers in each country.

The GROUP BY statement is often used with aggregate function to group the result-set by one or more columns.

Syntax :-

```
SELECT column_name(s)
FROM tablename
WHERE condition
GROUP BY column_name(s);
```

Example :-

```
SELECT COUNT (CustomerID), Country
FROM customers
GROUP BY Country;
```

Subquery :-

A subquery in MySQL is a query which is nested into another SQL query and embedded with SELECT, INSERT, UPDATE or DELETE statement along with the various operators. We can also nest the subquery with another subquery. A subquery is known as the inner query and the query that contains subquery is known as the outer query.

Syntax :-

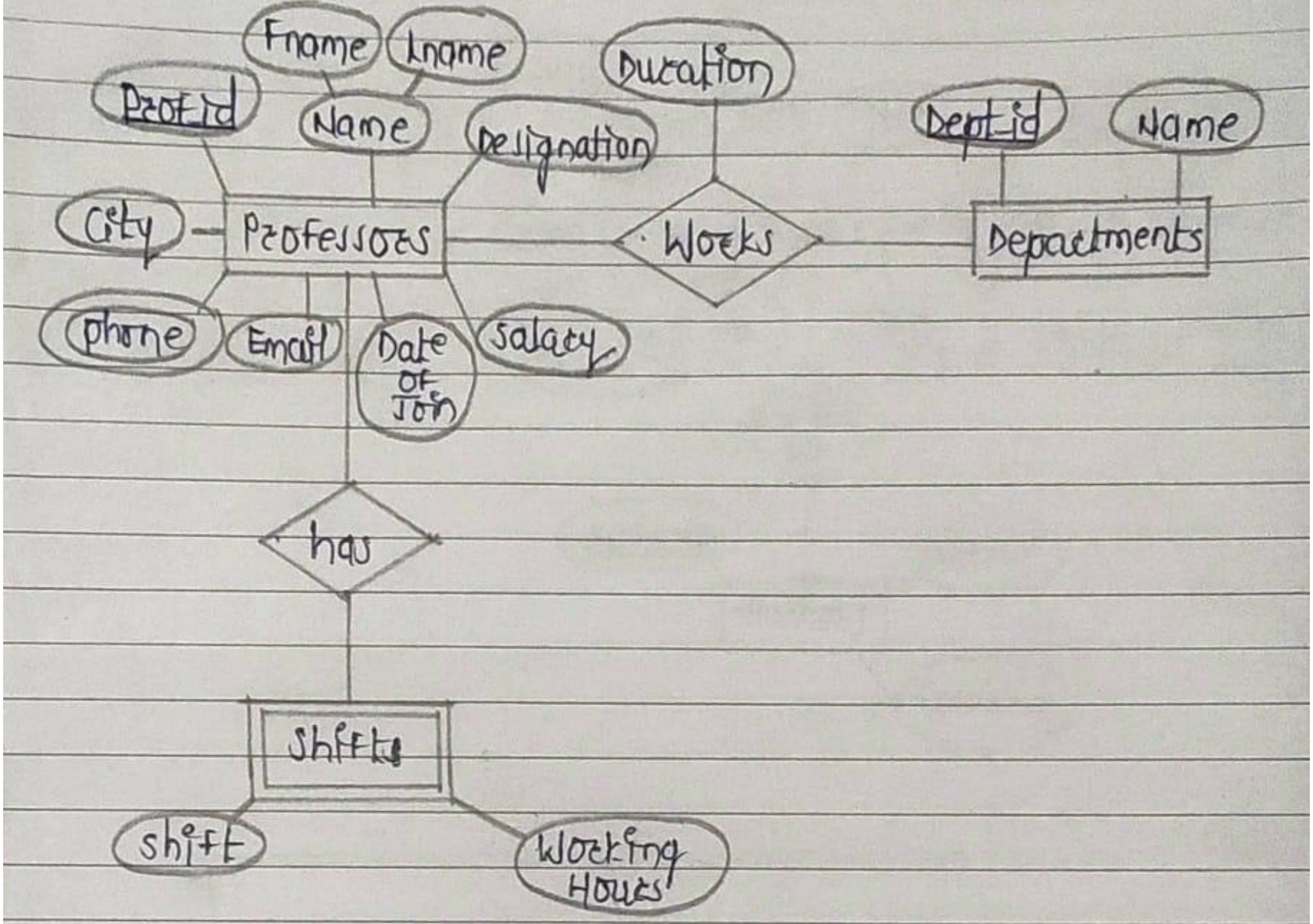
```
SELECT column_name
FROM table_name
WHERE column_name Operator
( SELECT column_name(s)
  FROM table_name
  WHERE condition
);
```

Example :-

```
SELECT Dept_name
FROM Departments
WHERE Dept_id IN
(
   SELECT Dept_id
   FROM PROFESSORS
   WHERE DOJ = '2015-01-01'
);
```

ER Diagram :-

ER Diagram :-



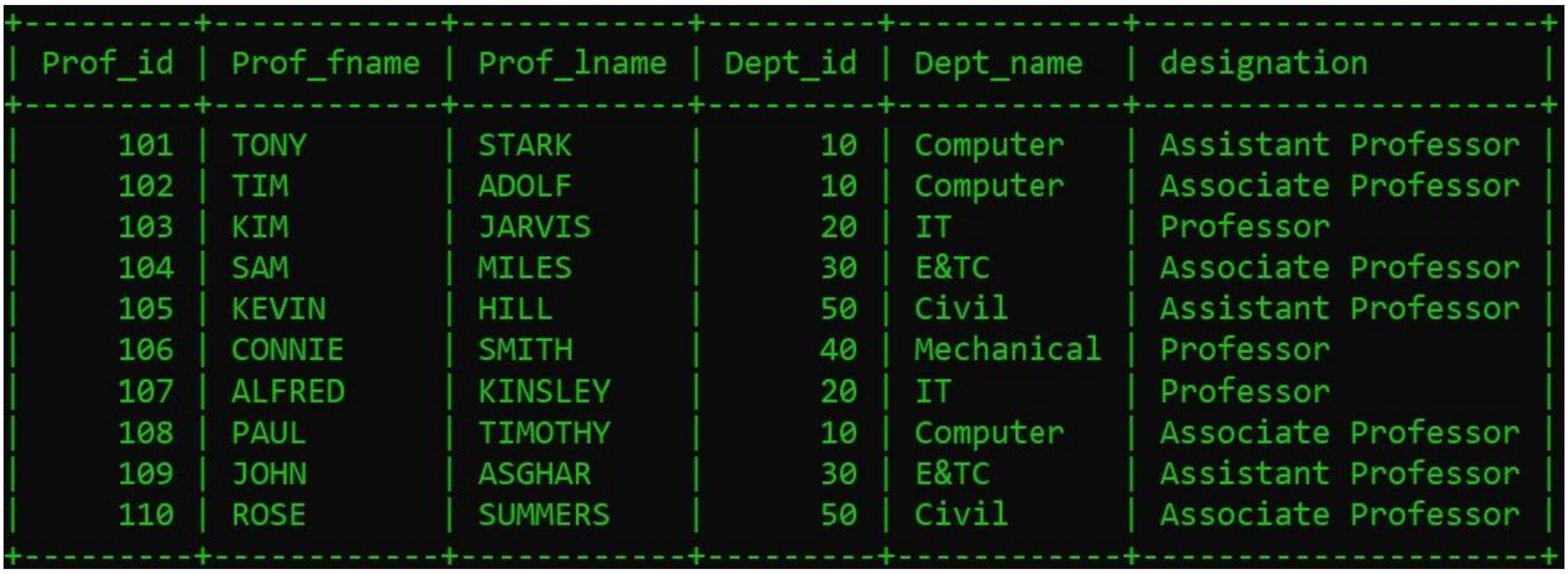
CONCLUSION :-

thus we have studied to use and implement various join operations with nested queries

OUTPUT :-

-- 1. Find the professor details and department details using NATURAL JOIN.

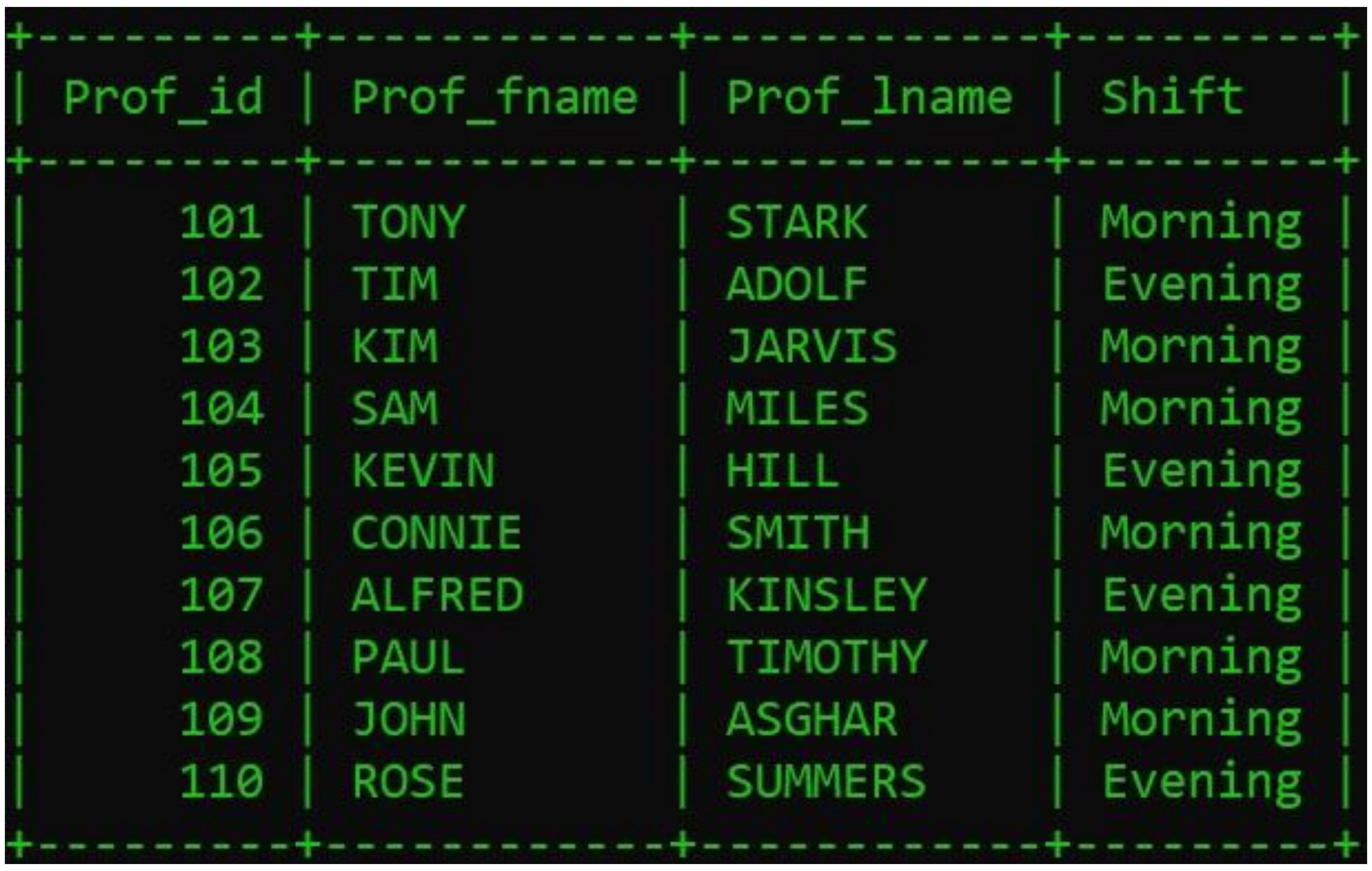
```
SELECT P.Prof_id, Prof_fname, Prof_lname, D.Dept_id, D.Dept_name, P.designation
    FROM Professors P
    NATURAL JOIN Departments D;
```



```
+---------+------------+------------+---------+------------+---------------------+
| Prof_id | Prof_fname | Prof_lname | Dept_id | Dept_name  | designation         |
+---------+------------+------------+---------+------------+---------------------+
|     101 | TONY       | STARK      |      10 | Computer   | Assistant Professor |
|     102 | TIM        | ADOLF      |      10 | Computer   | Associate Professor |
|     103 | KIM        | JARVIS     |      20 | IT         | Professor           |
|     104 | SAM        | MILES      |      30 | E&TC       | Associate Professor |
|     105 | KEVIN      | HILL       |      50 | Civil      | Assistant Professor |
|     106 | CONNIE     | SMITH      |      40 | Mechanical | Professor           |
|     107 | ALFRED     | KINSLEY    |      20 | IT         | Professor           |
|     108 | PAUL       | TIMOTHY    |      10 | Computer   | Associate Professor |
|     109 | JOHN       | ASGHAR     |      30 | E&TC       | Assistant Professor |
|     110 | ROSE       | SUMMERS    |      50 | Civil      | Associate Professor |
+---------+------------+------------+---------+------------+---------------------+
```

-- 2. Find the prof_id, prof_name and shift. (INNER JOIN)

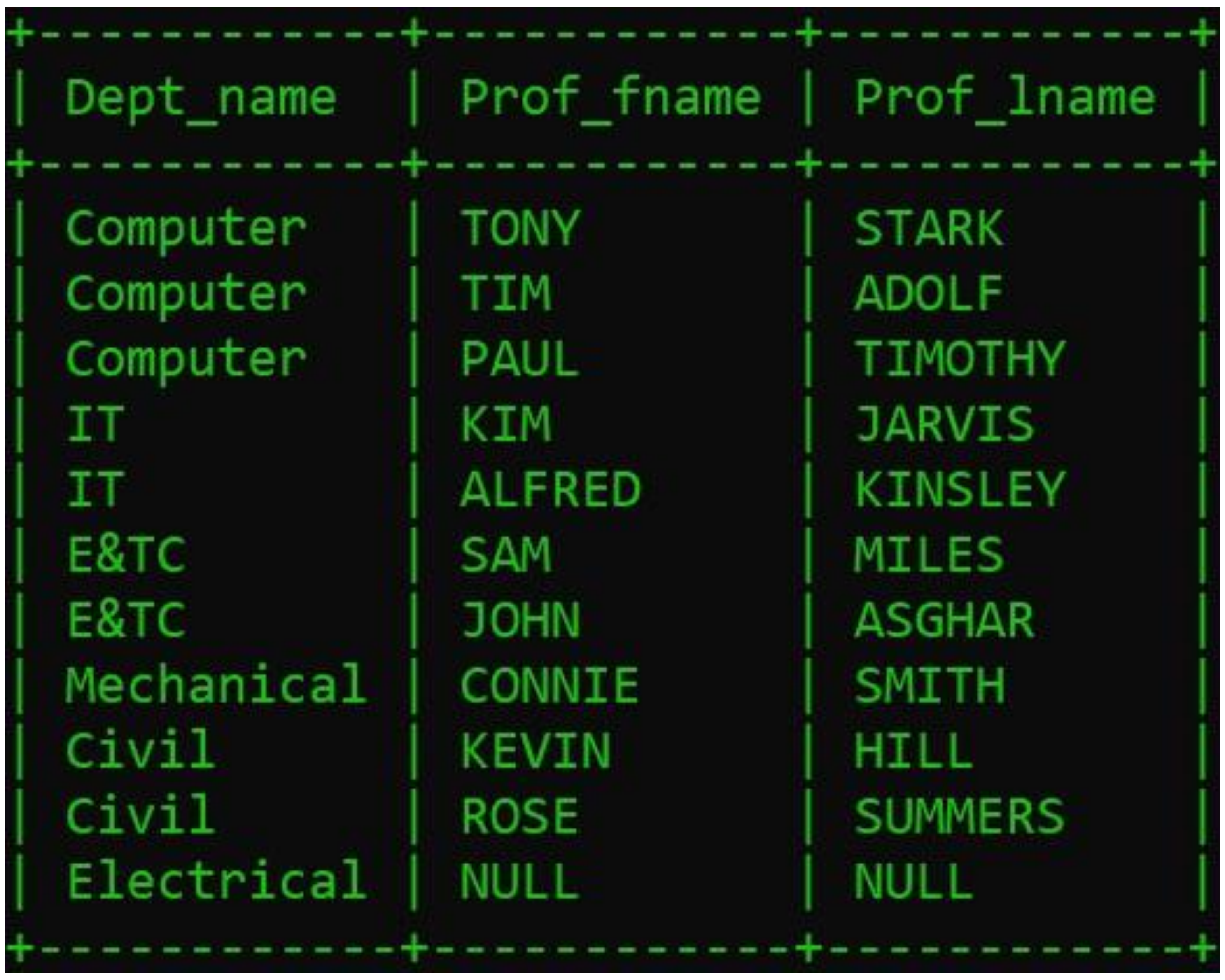
```
SELECT P.Prof_id, Prof_fname, Prof_lname, S.Shift FROM
Professors P INNER JOIN Shift S ON P.Prof_id = S.Prof_id;
```



```
+---------+------------+------------+---------+
| Prof_id | Prof_fname | Prof_lname | Shift   |
+---------+------------+------------+---------+
|     101 | TONY       | STARK      | Morning |
|     102 | TIM        | ADOLF      | Evening |
|     103 | KIM        | JARVIS     | Morning |
|     104 | SAM        | MILES      | Morning |
|     105 | KEVIN      | HILL       | Evening |
|     106 | CONNIE     | SMITH      | Morning |
|     107 | ALFRED     | KINSLEY    | Evening |
|     108 | PAUL       | TIMOTHY    | Morning |
|     109 | JOHN       | ASGHAR     | Morning |
|     110 | ROSE       | SUMMERS    | Evening |
+---------+------------+------------+---------+
```

-- 3. List all the department details and the corresponding names of professors -- in the same department.(left outer join)

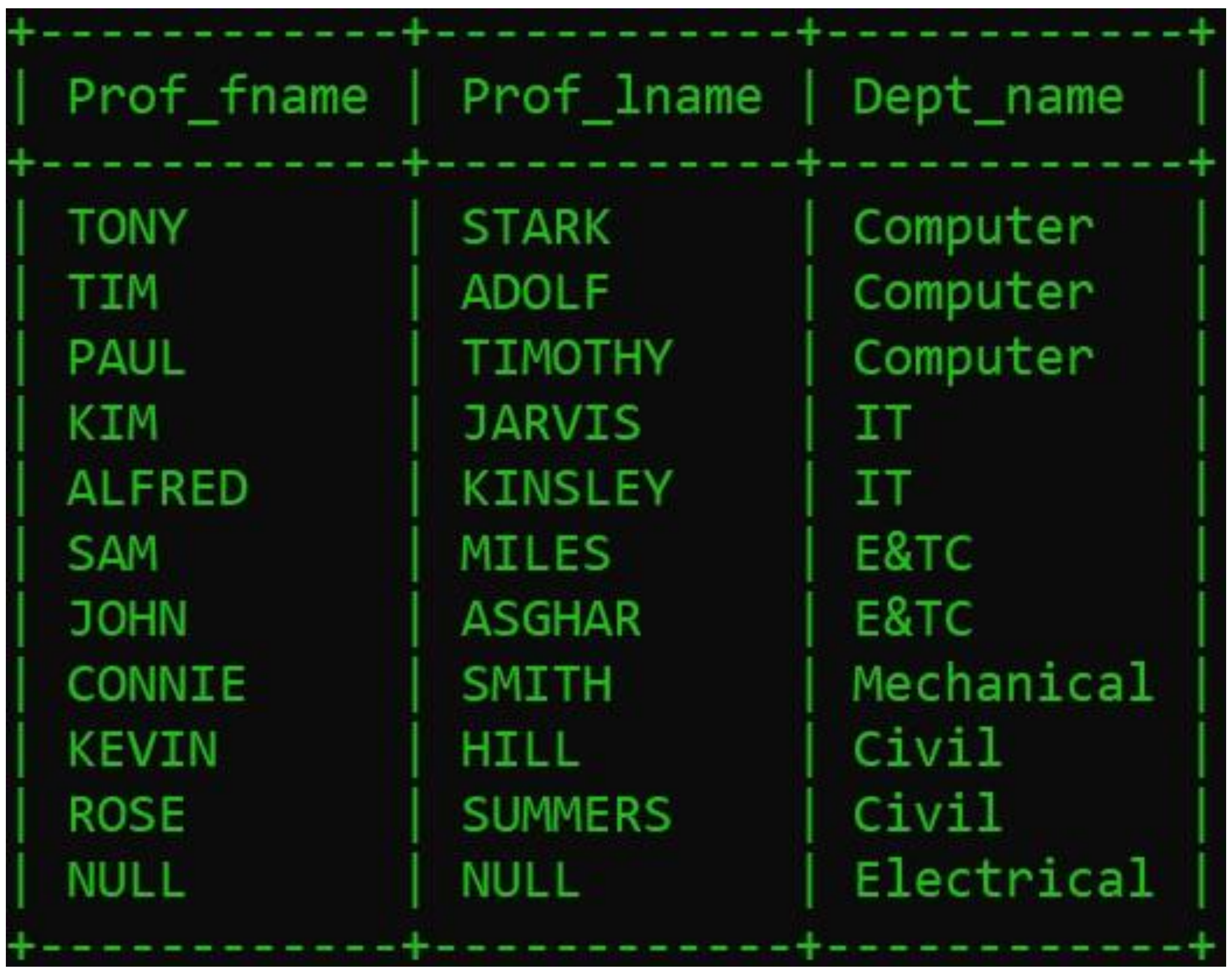
SELECT D.Dept_name, P.Prof_fname, P.Prof_lname
FROM Departments D LEFT JOIN Professors P ON
D.Dept_id = P.Dept_id;



```
+------------+------------+------------+
| Dept_name  | Prof_fname | Prof_lname |
+------------+------------+------------+
| Computer   | TONY       | STARK      |
| Computer   | TIM        | ADOLF      |
| Computer   | PAUL       | TIMOTHY    |
| IT         | KIM        | JARVIS     |
| IT         | ALFRED     | KINSLEY    |
| E&TC       | SAM        | MILES      |
| E&TC       | JOHN       | ASGHAR     |
| Mechanical | CONNIE     | SMITH      |
| Civil      | KEVIN      | HILL       |
| Civil      | ROSE       | SUMMERS    |
| Electrical | NULL       | NULL       |
+------------+------------+------------+
```

-- 4. List all the professors and the corresponding names of department. -- (right outer join)

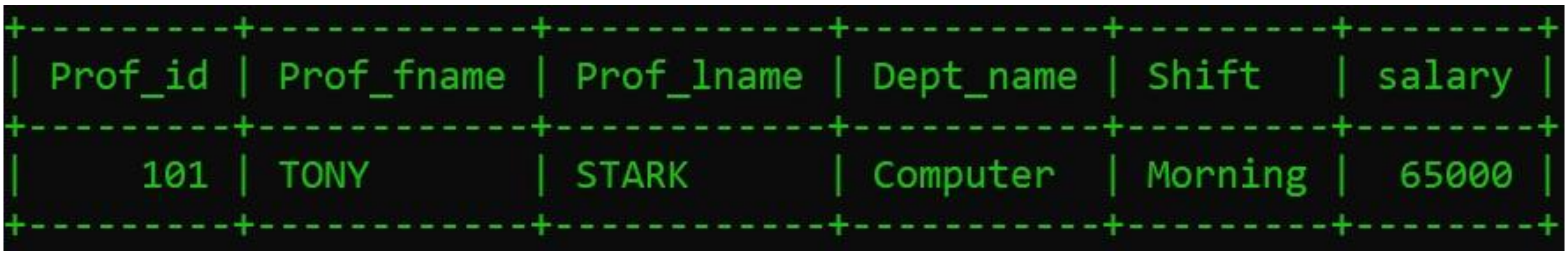
SELECT P.Prof_fname, P.Prof_lname, D.Dept_name
FROM Professors P RIGHT JOIN Departments D ON
P.Dept_id = D.Dept_id ;



```
+------------+------------+------------+
| Prof_fname | Prof_lname | Dept_name  |
+------------+------------+------------+
| TONY       | STARK      | Computer   |
| TIM        | ADOLF      | Computer   |
| PAUL       | TIMOTHY    | Computer   |
| KIM        | JARVIS     | IT         |
| ALFRED     | KINSLEY    | IT         |
| SAM        | MILES      | E&TC       |
| JOHN       | ASGHAR     | E&TC       |
| CONNIE     | SMITH      | Mechanical |
| KEVIN      | HILL       | Civil      |
| ROSE       | SUMMERS    | Civil      |
| NULL       | NULL       | Electrical |
+------------+------------+------------+
```

-- 5. Display professor name, dept_name, shift, salary where prof_id = 101 -- (multitable join)

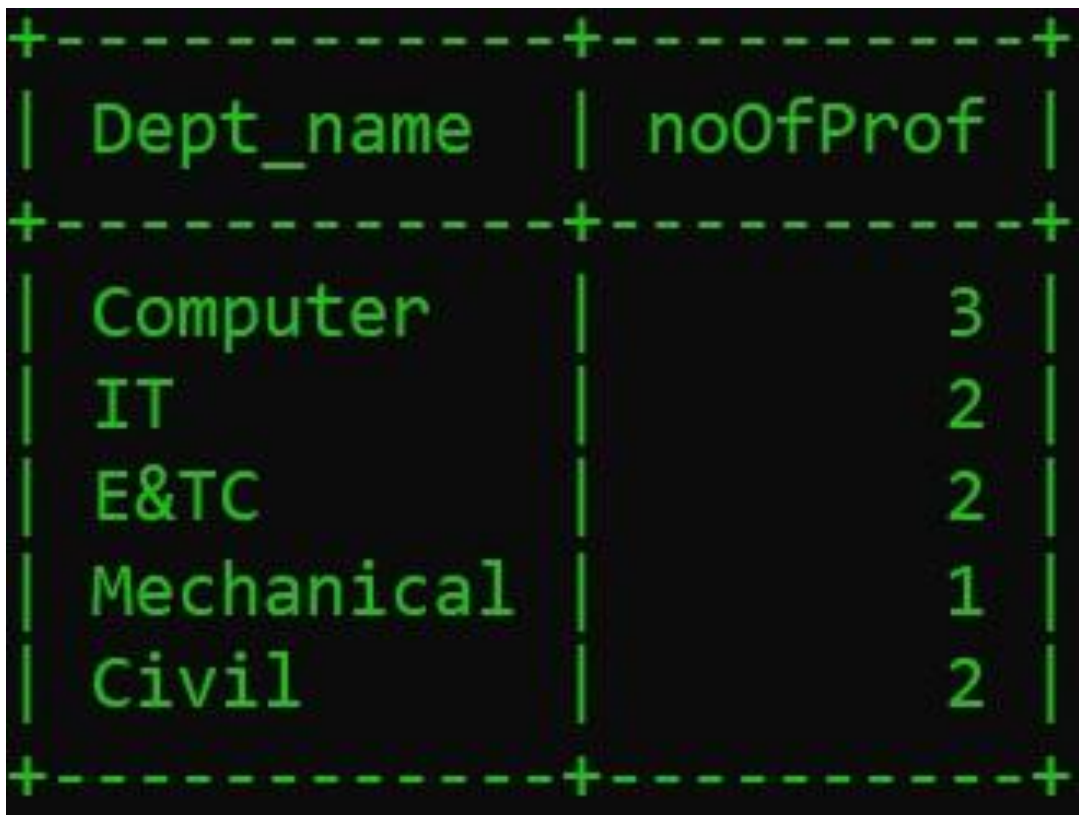
SELECT P.Prof_id, P.Prof_fname, P.Prof_lname, D.Dept_name, S.Shift, P.salary FROM Professors P JOIN Departments D ON P.prof_id = 101 AND D.Dept_id = P.Dept_id JOIN Shift S ON S.prof_id = 101;



```
+---------+------------+------------+-----------+---------+--------+
| Prof_id | Prof_fname | Prof_lname | Dept_name | Shift   | salary |
+---------+------------+------------+-----------+---------+--------+
|     101 | TONY       | STARK      | Computer  | Morning |  65000 |
+---------+------------+------------+-----------+---------+--------+
```

-- 6. list the total number of professor in each department.
-- (count and any join, groupby)

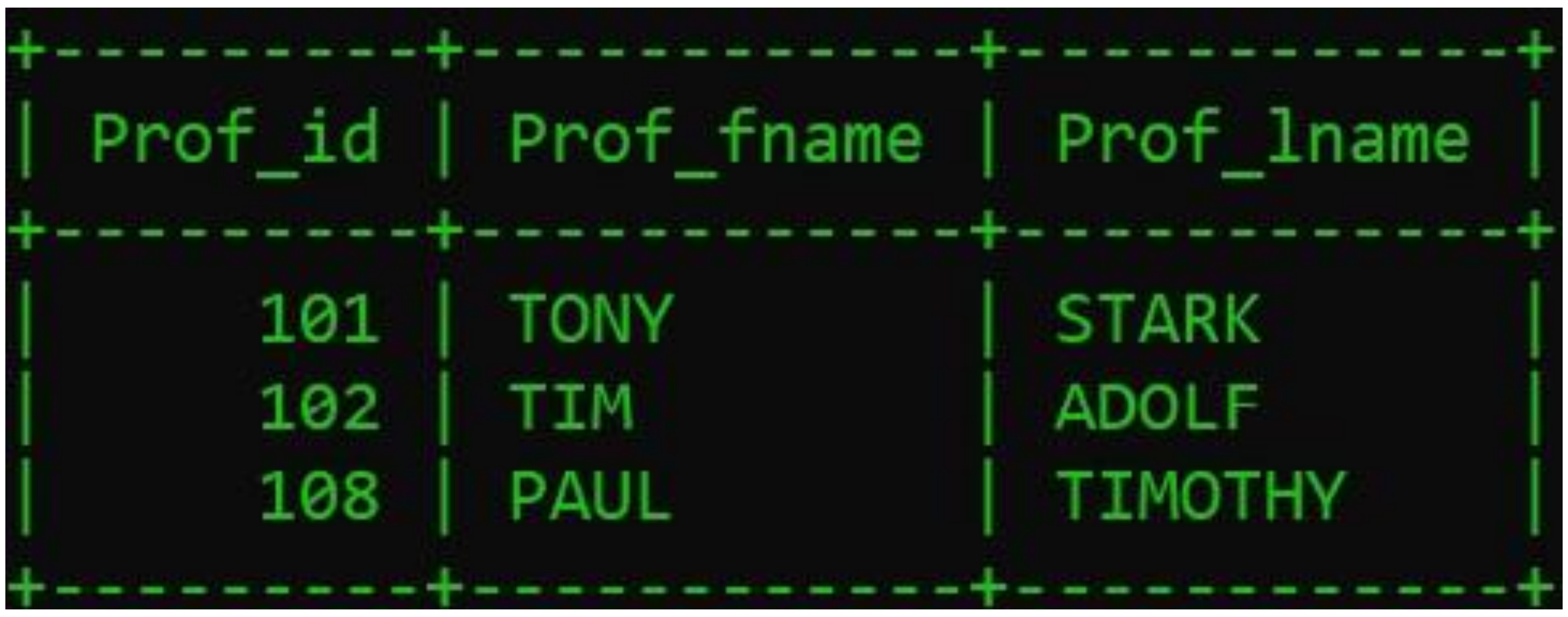
SELECT D.Dept_name, COUNT(P.Prof_id) AS noOfProf
FROM (Professors P INNER JOIN Departments D ON
P.Dept_id = D.Dept_id) GROUP BY Dept_name;



```
+------------+----------+
| Dept_name  | noOfProf |
+------------+----------+
| Computer   |        3 |
| IT         |        2 |
| E&TC       |        2 |
| Mechanical |        1 |
| Civil      |        2 |
+------------+----------+
```

-- 7. List the prof_id associated department and the dept_name having name -- 'computer' (subquery)

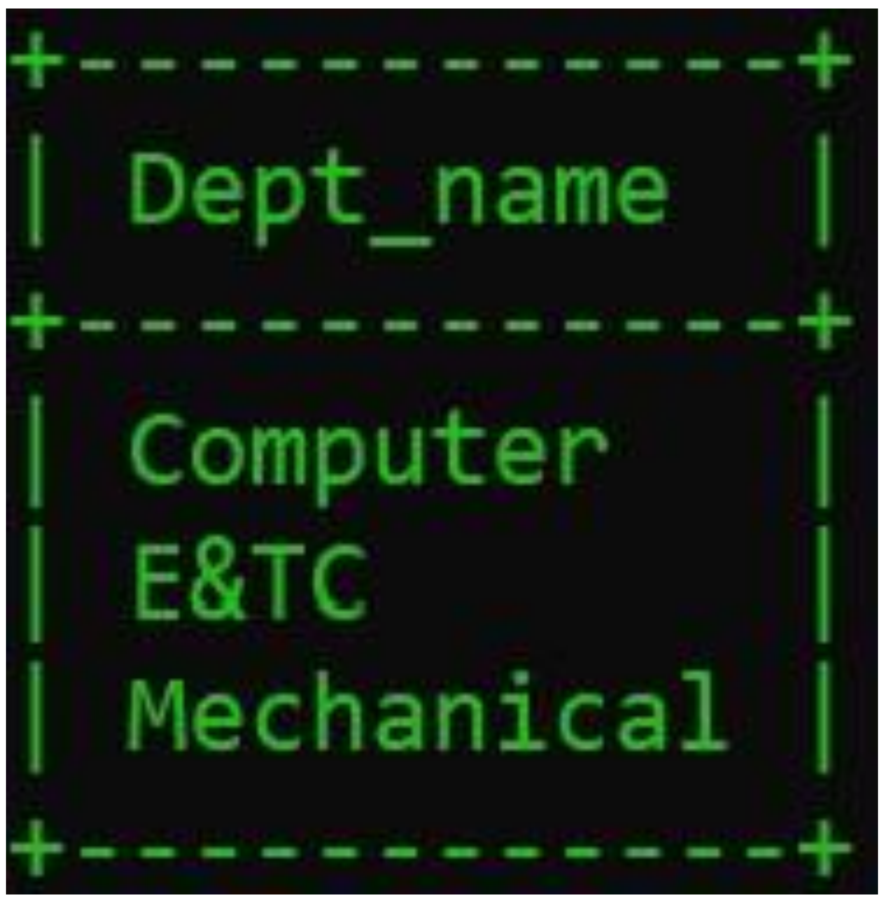
SELECT Prof_id, Prof_fname, Prof_lname FROM
Professors WHERE Dept_id = (SELECT Dept_id FROM
Departments WHERE Dept_name = 'Computer');



```
+---------+------------+------------+
| Prof_id | Prof_fname | Prof_lname |
+---------+------------+------------+
|     101 | TONY       | STARK      |
|     102 | TIM        | ADOLF      |
|     108 | PAUL       | TIMOTHY    |
+---------+------------+------------+
```

-- 8. Find the names of all departments where the professors joined in year 2015 -- (or date of joining is 1-1-2015).(subquery)

```
SELECT DISTINCT Dept_name
    FROM Departments NATURAL
    JOIN professors WHERE
    YEAR(Date_of_join) = '2015';
```



```
+------------+
| Dept_name  |
+------------+
| Computer   |
| E&TC       |
| Mechanical |
+------------+
```