LAKSHMI KUMAR
002887298

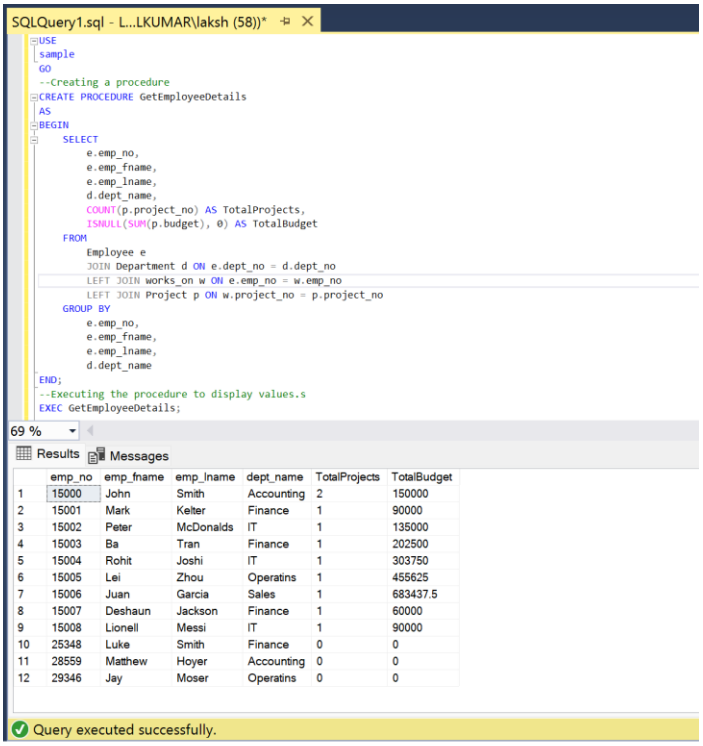
DATA MANAGEMENT AND DATABASE DESIGN
HOMEWORK: WEEK- 10

1. Create **GetEmployeeDetails** stored procedure return result set of the total number of projects each employee is working on and the total budget for these projects. This will involve joining multiple tables and using aggregate functions. All employees must be returned even if no projects were assigned.
   Columns to be returned:
   - emp_no,
   - emp_fname,
   - emp_lname,
   - dept_name,
   - TotalProjects,
   - TotalBudget

QUERY:

```
USE
sample
GO
--Query to create a procedure
CREATE PROCEDURE GetEmployeeDetails
AS
BEGIN
    SELECT
        e.emp_no,
        e.emp_fname,
        e.emp_lname,
        d.dept_name,
        COUNT (p. project_no) AS TotalProjects,
        ISNULL (SUM (p. budget), 0) AS TotalBudget
    FROM
        Employee e
        JOIN Department d ON e. dept_no = d. dept_no
        LEFT JOIN works_on w ON e.emp_no = w.emp_no
        LEFT JOIN Project p ON w. project_no = p. project_no
    GROUP BY
        e.emp_no,
        e.emp_fname,
        e.emp_lname,
        d.dept_name
END;

--Executing the procedure to display values
EXEC GetEmployeeDetails;
```



```
USE
sample
GO
--Creating a procedure
CREATE PROCEDURE GetEmployeeDetails
AS
BEGIN
    SELECT
        e.emp_no,
        e.emp_fname,
        e.emp_lname,
        d.dept_name,
        COUNT(p.project_no) AS TotalProjects,
        ISNULL(SUM(p.budget), 0) AS TotalBudget
    FROM
        Employee e
        JOIN Department d ON e.dept_no = d.dept_no
        LEFT JOIN works_on w ON e.emp_no = w.emp_no
        LEFT JOIN Project p ON w.project_no = p.project_no
    GROUP BY
        e.emp_no,
        e.emp_fname,
        e.emp_lname,
        d.dept_name
END;
--Executing the procedure to display values.s
EXEC GetEmployeeDetails;
```

	emp_no	emp_fname	emp_lname	dept_name	TotalProjects	TotalBudget
1	15000	John	Smith	Accounting	2	150000
2	15001	Mark	Kelter	Finance	1	90000
3	15002	Peter	McDonalds	IT	1	135000
4	15003	Ba	Tran	Finance	1	202500
5	15004	Rohit	Joshi	IT	1	303750
6	15005	Lei	Zhou	Operatins	1	455625
7	15006	Juan	Garcia	Sales	1	683437.5
8	15007	Deshaun	Jackson	Finance	1	60000
9	15008	Lionell	Messi	IT	1	90000
10	25348	Luke	Smith	Finance	0	0
11	28559	Matthew	Hoyer	Accounting	0	0
12	29346	Jay	Moser	Operatins	0	0

Query executed successfully.

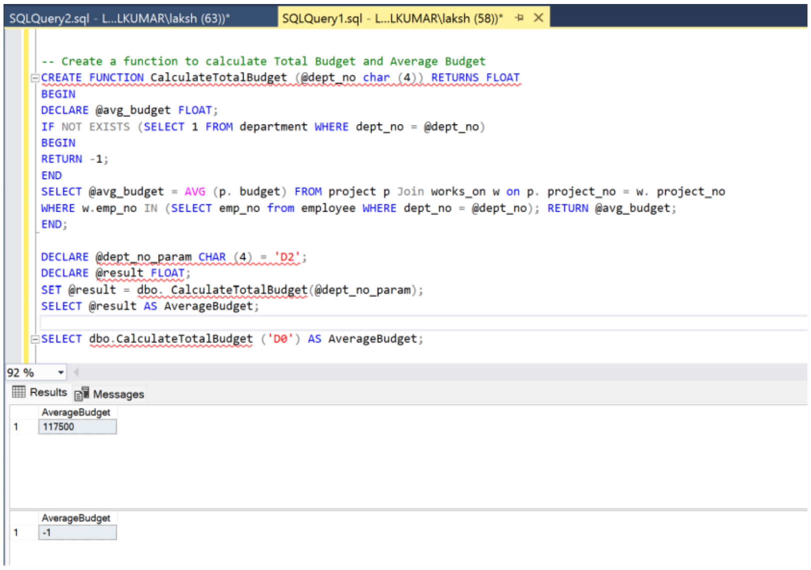
2. Create **CalculateTotalBudget** function that takes dept_no as input parameter and returns **-1** if the department does not exist. Also, make the function calculate the average budget per project for the specified department.

QUERY:

```
-- Create a function to calculate Total Budget and Average Budget
CREATE FUNCTION CalculateTotalBudget (@dept_no char (4)) RETURNS FLOAT
BEGIN
DECLARE @avg_budget FLOAT;
IF NOT EXISTS (SELECT 1 FROM department WHERE dept_no = @dept_no)
BEGIN
RETURN -1;
END
SELECT @avg_budget = AVG (p. budget) FROM project p Join works_on w on p.
project_no = w. project_no
WHERE w.emp_no IN (SELECT emp_no from employee WHERE dept_no =
@dept_no); RETURN @avg_budget;
END;

DECLARE @dept_no_param CHAR (4) = 'D2';
DECLARE @result FLOAT;
SET @result = dbo. CalculateTotalBudget(@dept_no_test);
SELECT @result AS AverageBudget;

SELECT dbo.CalculateTotalBudget(‘D0’) AS AverageBudget;
```



SQLQuery2.sql - L...LKUMAR\laksh (63))*
SQLQuery1.sql - L...LKUMAR\laksh (58))*
92 %
Results
Messages

```sql
-- Create a function to calculate Total Budget and Average Budget
CREATE FUNCTION CalculateTotalBudget (@dept_no char (4)) RETURNS FLOAT
BEGIN
DECLARE @avg_budget FLOAT;
IF NOT EXISTS (SELECT 1 FROM department WHERE dept_no = @dept_no)
BEGIN
RETURN -1;
END
SELECT @avg_budget = AVG (p. budget) FROM project p Join works_on w on p. project_no = w. project_no
WHERE w.emp_no IN (SELECT emp_no from employee WHERE dept_no = @dept_no); RETURN @avg_budget;
END;

DECLARE @dept_no_param CHAR (4) = 'D2';
DECLARE @result FLOAT;
SET @result = dbo. CalculateTotalBudget(@dept_no_param);
SELECT @result AS AverageBudget;

SELECT dbo.CalculateTotalBudget ('D0') AS AverageBudget;
```

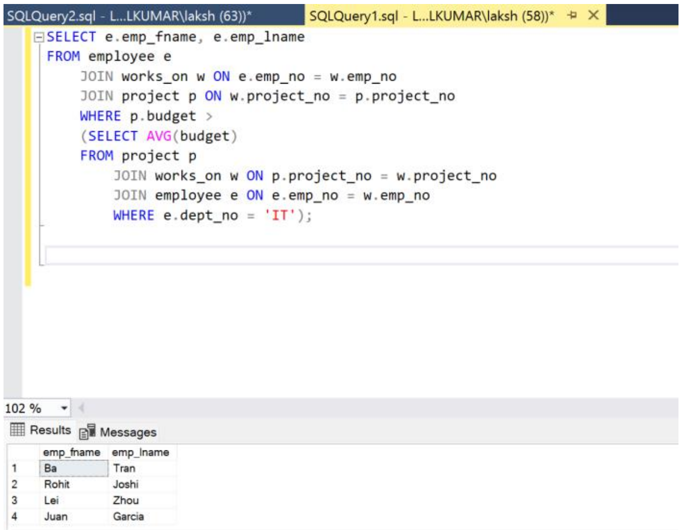
```
   AverageBudget
1  117500

   AverageBudget
1  -1
```

3. Formulate a query using a subquery to find the names of all employees who work on projects with a budget greater than the average budget of all projects in the 'IT' department.

QUERY:

```
SELECT e.emp_fname, e.emp_lname
FROM employee e
JOIN works_on w ON e.emp_no = w.emp_no
JOIN project p ON w. project_no = p. project_no
WHERE p. budget >
(SELECT AVG (budget)
FROM project p
JOIN works_on w ON p. project_no = w. project_no
JOIN employee e ON e.emp_no = w.emp_no
WHERE e. dept_no = 'IT');
```



SQLQuery2.sql - L...LKUMAR\laksh (63))*
SQLQuery1.sql - L...LKUMAR\laksh (58))*
102 %
Results
Messages

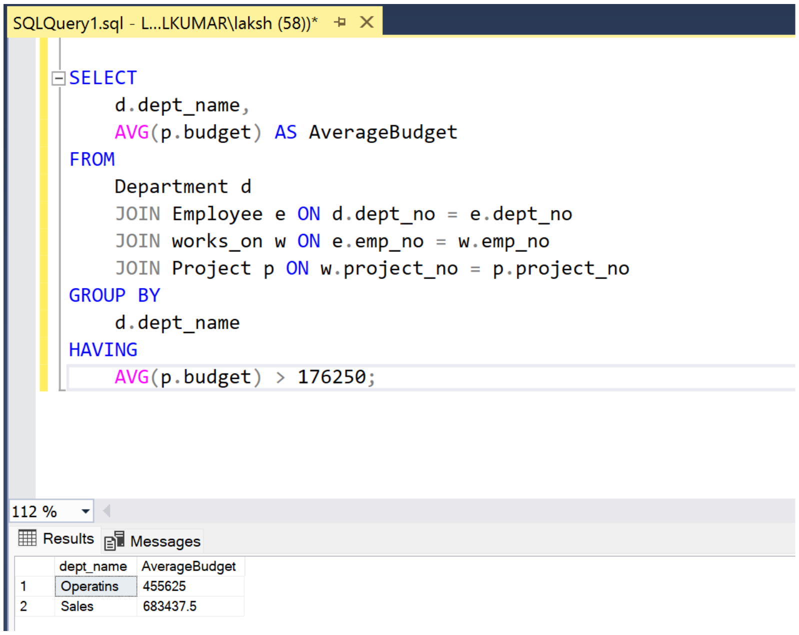
```sql
SELECT e.emp_fname, e.emp_lname
FROM employee e
    JOIN works_on w ON e.emp_no = w.emp_no
    JOIN project p ON w.project_no = p.project_no
    WHERE p.budget >
    (SELECT AVG(budget)
    FROM project p
        JOIN works_on w ON p.project_no = w.project_no
        JOIN employee e ON e.emp_no = w.emp_no
        WHERE e.dept_no = 'IT');
```

```
   emp_fname  emp_lname
1  Ba         Tran
2  Rohit      Joshi
3  Lei        Zhou
4  Juan       Garcia
```

4. Construct a query that shows the department names with average budget greater than $176,250.

QUERY:

```
SELECT d. dept_name,
AVG (p.budget) AS AverageBudget
FROM
Department d
   JOIN Employee e ON d. dept_no = e. dept_no
   JOIN works_on w ON e.emp_no = w.emp_no
   JOIN Project p ON w. project_no = p. project_no
GROUP BY d.dept_name
HAVING AVG (p.budget) > 176250;
```



```
SELECT
    d.dept_name,
    AVG(p.budget) AS AverageBudget
FROM
    Department d
    JOIN Employee e ON d.dept_no = e.dept_no
    JOIN works_on w ON e.emp_no = w.emp_no
    JOIN Project p ON w.project_no = p.project_no
GROUP BY
    d.dept_name
HAVING
    AVG(p.budget) > 176250;
```

	dept_name	AverageBudget
1	Operatins	455625
2	Sales	683437.5