Assignment 12

```
use SAMPLE;
```

Data Profiling:

```
[2]   1   select * from employee;
```

SQL

(12 rows affected)

Total execution time: 00:00:00.065

	emp_no	emp_fname	emp_lname	dept_no
1	15000	John	Smith	D1
2	15001	Mark	Kelter	D2
3	15002	Peter	McDonalds	D3
4	15003	Ba	Tran	D2
5	15004	Rohit	Joshi	D3
6	15005	Lei	Zhou	D4
7	15006	Juan	Garcia	D5
8	15007	Deshaun	Jackson	D2
9	15008	Lionell	Messi	D3
10	25348	Luke	Smith	D2
11	28559	Matthew	Hoyer	D1
12	29346	Jay	Moser	D4

```
[3]   1   select * from department;
```

SQL

(5 rows affected)

Total execution time: 00:00:00.020

	dept_no	dept_name	location
1	D1	Accounting	Dallas
2	D2	Finance	Seattle
3	D3	IT	Boston
4	D4	Operatins	New York
5	D5	Sales	London

```
[4]  1  select * from project;
```

SQL

(7 rows affected)

Total execution time: 00:00:00.008

	project_no	project_name	budget
1	p1	Inventory system	60000
2	p2	CRM system	90000
3	p3	Order Management	135000
4	p4	Accounting System	202500
5	p5	ERP system	303750
6	p6	Data Warehouse	455625
7	p7	corporate website	683437.5

```
[5]  1  select * from works_on;
```

SQL

(12 rows affected)

Total execution time: 00:00:00.029

	emp_no	project_no	job	enter_date
1	15000	p1	Project Manager	2019-01-03
2	15000	p2	Project Manager	2024-01-03
3	15001	p2	*NULL*	2019-01-04
4	15001	p3	Data Engineer	2024-03-04
5	15001	p4	Data Engineer	2024-01-29
6	15002	p3	Software Developer	2020-01-03
7	15003	p4	Data Engineer	2020-01-04
8	15004	p5	Network Admin	2021-01-03
9	15005	p6	Data Analyst	2021-01-04
10	15006	p7	Data Engineer	2022-01-03
11	15007	p1	Data Architect	2022-01-04
12	15008	p2	DBA	2023-01-03

Use “sample” database to answer the following questions:

- **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,

TotalBudge

```
CREATE PROCEDURE GetEmployeeDetails
AS
BEGIN
    SELECT e.emp_no, e.emp_fname, e.emp_lname, d.dept_name,
           ISNULL(COUNT(w.project_no), 0) AS TotalProjects,
           ISNULL(SUM(p.budget), 0) AS TotalBudget
    FROM employee e
    LEFT 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
```

```
[12]    1    EXEC GetEmployeeDetails;
```

SQL

Warning: Null value is eliminated by an aggregate or other SET operation.

(12 rows affected)

Total execution time: 00:00:00.229

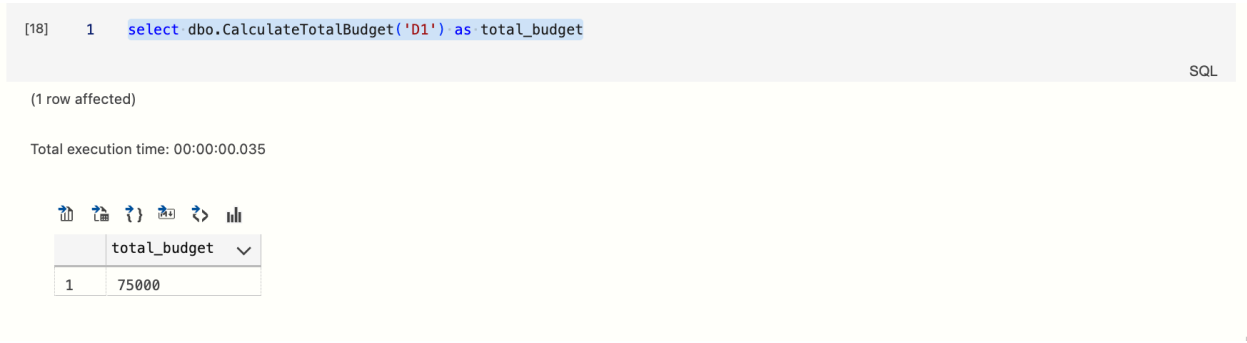
	emp_no	emp_fname	emp_lname	dept_name	TotalProjects	TotalBudget
1	15000	John	Smith	Accounting	2	150000
2	15001	Mark	Kelter	Finance	3	427500
3	15002	Peter	McDonalds	IT	1	135000
4	15003	Ba	Tran	Finance	1	202500
5	15004	Rohit	Joshi	IT	1	303750
6	15005	Lei	Zhou	Operatins	1	455625
7	15006	Juan	Garcia	Sales	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

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.**

```
CREATE FUNCTION CalculateTotalBudget (@dept_no CHAR(4))
RETURNS FLOAT
AS
BEGIN
    DECLARE @TotalBudget FLOAT
    SELECT @TotalBudget = AVG(p.budget)
    FROM project p
    WHERE p.project_no IN (SELECT w.project_no FROM works_on w JOIN employee e ON
w.emp_no = e.emp_no WHERE e.dept_no = @dept_no)
    RETURN ISNULL(@TotalBudget, -1)
END
```

Give D1 as input department

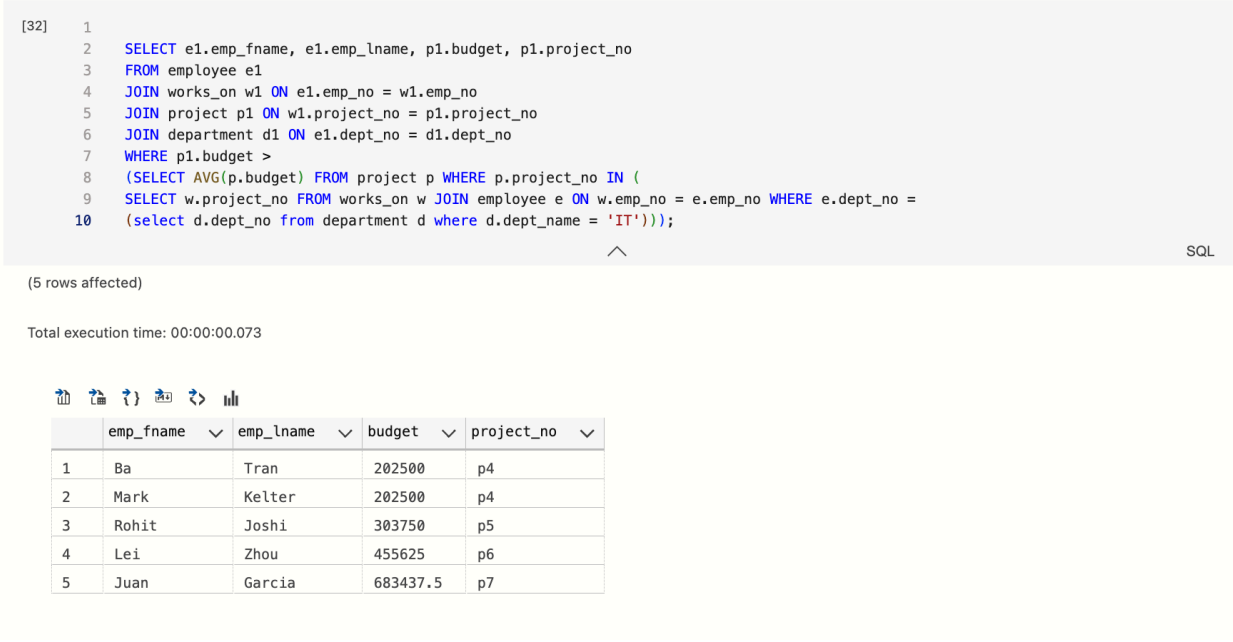
```
select dbo.CalculateTotalBudget('D1') as total_budget
```



- **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.**

```
SELECT e1.emp_fname, e1.emp_lname, p1.budget, p1.project_no
FROM employee e1
JOIN works_on w1 ON e1.emp_no = w1.emp_no
JOIN project p1 ON w1.project_no = p1.project_no
JOIN department d1 ON e1.dept_no = d1.dept_no
WHERE p1.budget >
(SELECT AVG(p.budget) FROM project p WHERE p.project_no IN (
```

```
SELECT w.project_no FROM works_on w JOIN employee e ON w.emp_no = e.emp_no WHERE
e.dept_no =
(select d.dept_no from department d where d.dept_name = 'IT')));
```



```
[32]
SELECT e1.emp_fname, e1.emp_lname, p1.budget, p1.project_no
FROM employee e1
JOIN works_on w1 ON e1.emp_no = w1.emp_no
JOIN project p1 ON w1.project_no = p1.project_no
JOIN department d1 ON e1.dept_no = d1.dept_no
WHERE p1.budget >
(SELECT AVG(p.budget) FROM project p WHERE p.project_no IN (
SELECT w.project_no FROM works_on w JOIN employee e ON w.emp_no = e.emp_no WHERE e.dept_no =
(select d.dept_no from department d where d.dept_name = 'IT')));
```

SQL

(5 rows affected)

Total execution time: 00:00:00.073

	emp_fname	emp_lname	budget	project_no
1	Ba	Tran	202500	p4
2	Mark	Kelter	202500	p4
3	Rohit	Joshi	303750	p5
4	Lei	Zhou	455625	p6
5	Juan	Garcia	683437.5	p7

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

```
SELECT d.dept_name, AVG(p.budget) Average_Budget
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;
```

[35]

```sql
SELECT d.dept_name, AVG(p.budget) Average_Budget
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;

```

SQL

(2 rows affected)

Total execution time: 00:00:00.048

	dept_name	Average_Budget
1	Operatins	455625
2	Sales	683437.5