

## CIS 430: Lab Assignment 5

Name: Yuvaraj Vagula

ID: 2862494

Object: 2 parts; View and Stored Procedure using Cursor

**1-1) View Creation from SQL**

```
-- Create the view VDept_Budget
CREATE VIEW VDept_Budget AS
SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
GROUP BY D.DNAME, D.DNUMBER;

SELECT * FROM VDept_Budget;
```

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'COMPANY\_yVagula' database selected. The right pane shows the 'Lab 5.sql' script with the following SQL code:

```
1 USE master;
2 -- Create a db named COMPANY_yVagula. do NOT use master db.
3 IF DB_ID('COMPANY_yVagula') IS NULL
4 CREATE DATABASE COMPANY_yVagula;
5 GO
6 -- Use created db
7 USE COMPANY_yVagula;
8 GO
9
10 -- Create the view VDept_Budget
11 CREATE VIEW VDept_Budget AS
12 SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp
13 FROM DEPARTMENT D
14 LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
15 GROUP BY D.DNAME, D.DNUMBER;
16
17 SELECT * FROM VDept_Budget;
```

The bottom pane shows the 'Results' tab with the following data:

| Dept_Name      | Dept_Number | No_Emp |
|----------------|-------------|--------|
| Headquarters   | 1           | 1      |
| Administration | 4           | 3      |
| Research       | 5           | 5      |
| Automation     | 7           | 0      |

The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

## 1-2) Table creation from the same SQL as in 1.1

```
CREATE TABLE Dept_Budget (
    Dept_Name VARCHAR(15),
    Dept_Number INT,
    No_Emp INT
);
```

```
INSERT INTO Dept_Budget (Dept_Name, Dept_Number, No_Emp)
SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
GROUP BY D.DNAME, D.DNUMBER;
```

```
SELECT * FROM Dept_Budget;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
18 SELECT * FROM VDept_Budget;
19 -- (1-2)---
20 CREATE TABLE Dept_Budget (
21     Dept_Name VARCHAR(15),
22     Dept_Number INT,
23     No_Emp INT
24 );
25
26 INSERT INTO Dept_Budget (Dept_Name, Dept_Number, No_Emp)
27 SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp
28 FROM DEPARTMENT D
29 LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
30 GROUP BY D.DNAME, D.DNUMBER;
31
32 SELECT * FROM Dept_Budget;
33
34
35
```

The Results pane shows the output of the final query, displaying a table with 4 rows and 3 columns: Dept\_Name, Dept\_Number, and No\_Emp.

|   | Dept_Name      | Dept_Number | No_Emp |
|---|----------------|-------------|--------|
| 1 | Headquarters   | 1           | 1      |
| 2 | Administration | 4           | 3      |
| 3 | Research       | 5           | 5      |
| 4 | Automation     | 7           | 0      |

The status bar at the bottom indicates: Query executed successfully. LAPTOP-LSLNUDIE (15.0 RTM) | LAPTOP-LSLNUDIE\Yuvav ... COMPANY\_yVagula 00:00:00 4 rows

2.

**2-1 & 2-1) Adding myself and another employee, then showing the View VDEPT\_BUDGET content again.**

```
INSERT INTO EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO)
VALUES ('Yuvaraj', 'N', 'Vagula', '777777777', '2006-05-01', '2121 Euclid Ave, Cleveland, OH', 'M', 50000, NULL, 1),
       ('John', 'B', 'Doe', '222222222', '1991-02-02', '456 dog St, Shaker, OH', 'F', 50000, NULL, 1);
```

```
SELECT * FROM VDept_Budget;
```

```
SELECT * FROM Dept_Budget;
```

Lab 5.sql - LAPTOP-LSLNUDIE\COMPANY\_yVagula (LAPTOP-LSLNUDIE\Yuvar (70)) - Microsoft SQL Server Management Studio

Object Explorer: LAPTOP-LSLNUDIE (SQL Server 15.0.2116)

Query: Lab 5.sql - LAPTOP-LSLNUDIE\Yuvar (70)

```
24 );
25
26 INSERT INTO Dept_Budget (Dept_Name, Dept_Number, No_Emp)
27 SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp
28 FROM DEPARTMENT D
29 LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
30 GROUP BY D.DNAME, D.DNUMBER;
31
32 SELECT * FROM Dept_Budget;
33
34 -- (2-1) & (2-2) -----
35 INSERT INTO EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO)
36 VALUES ('Yuvaraj', 'N', 'Vagula', '777777777', '2006-05-01', '2121 Euclid Ave, Cleveland, OH', 'M', 50000, NULL, 1),
37        ('John', 'B', 'Doe', '222222222', '1991-02-02', '456 dog St, Shaker, OH', 'F', 50000, NULL, 1);
38
39 SELECT * FROM VDept_Budget;
40
41 SELECT * FROM Dept_Budget;
42
43
```

Results:

| Dept_Name      | Dept_Number | No_Emp |
|----------------|-------------|--------|
| Headquarters   | 1           | 3      |
| Administration | 4           | 3      |
| Research       | 5           | 5      |
| Automation     | 7           | 0      |

| Dept_Name      | Dept_Number | No_Emp |
|----------------|-------------|--------|
| Headquarters   | 1           | 1      |
| Administration | 4           | 3      |
| Research       | 5           | 5      |
| Automation     | 7           | 0      |

Query executed successfully.

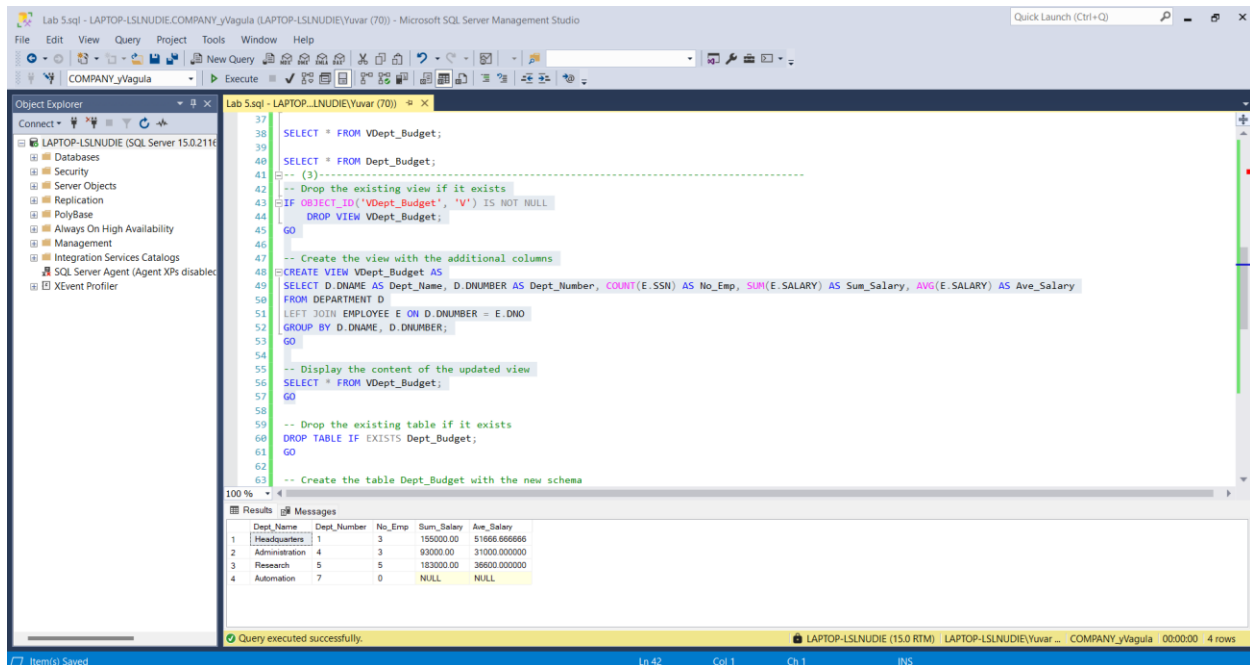
Item(s) Saved | Ln 38 | Col 1 | Ch 1 | INS | LAPTOP-LSLNUDIE (15.0 RTM) | LAPTOP-LSLNUDIE\Yuvar ... | COMPANY\_yVagula | 00:00:00 | 8 rows

### 3) Difference in Changing the Scheme of the Existing View and the Table

```
-- Drop the existing view if it exists
IF OBJECT_ID('VDept_Budget', 'V') IS NOT NULL
    DROP VIEW VDept_Budget;
GO

-- Create the view with the additional columns
CREATE VIEW VDept_Budget AS
SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp,
SUM(E.SALARY) AS Sum_Salary, AVG(E.SALARY) AS Ave_Salary
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
GROUP BY D.DNAME, D.DNUMBER;
GO

-- Display the content of the updated view
SELECT * FROM VDept_Budget;
GO
```



```
-- Drop the existing table if it exists
DROP TABLE IF EXISTS Dept_Budget;
GO

-- Create the table Dept_Budget with the new schema
CREATE TABLE Dept_Budget (
    Dept_Name VARCHAR(15),
    Dept_Number INT,
    No_Emp INT,
    Sum_Salary DECIMAL(10, 2),
    Ave_Salary DECIMAL(10, 2)
);
```

GO

```
-- Populate the table Dept_Budget with the new schema
INSERT INTO Dept_Budget (Dept_Name, Dept_Number, No_Emp, Sum_Salary, Ave_Salary)
SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp,
SUM(E.SALARY) AS Sum_Salary, AVG(E.SALARY) AS Ave_Salary
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
GROUP BY D.DNAME, D.DNUMBER;
GO
```

```
-- Display the content of the updated table
SELECT * FROM Dept_Budget;
GO
```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'COMPANY\_yVagula' database selected. The right pane shows a SQL query window with the following script:

```
58 -- Drop the existing table if it exists
59 DROP TABLE IF EXISTS Dept_Budget;
60 GO
61 -- Create the table Dept_Budget with the new schema
62 CREATE TABLE Dept_Budget (
63     Dept_Name VARCHAR(15),
64     Dept_Number INT,
65     No_Emp INT,
66     Sum_Salary DECIMAL(10, 2),
67     Ave_Salary DECIMAL(10, 2)
68 );
69 GO
70 -- Populate the table Dept_Budget with the new schema
71 INSERT INTO Dept_Budget (Dept_Name, Dept_Number, No_Emp, Sum_Salary, Ave_Salary)
72 SELECT D.DNAME AS Dept_Name, D.DNUMBER AS Dept_Number, COUNT(E.SSN) AS No_Emp, SUM(E.SALARY) AS Sum_Salary, AVG(E.SALARY) AS Ave_Salary
73 FROM DEPARTMENT D
74 LEFT JOIN EMPLOYEE E ON D.DNUMBER = E.DNO
75 GROUP BY D.DNAME, D.DNUMBER;
76 GO
77 -- Display the content of the updated table
78 SELECT * FROM Dept_Budget;
79 GO
```

Below the query window, the 'Results' pane shows the output of the final query:

| Dept_Name      | Dept_Number | No_Emp | Sum_Salary | Ave_Salary |
|----------------|-------------|--------|------------|------------|
| Headquarters   | 1           | 3      | 155000.00  | 51666.67   |
| Administration | 4           | 3      | 93000.00   | 31000.00   |
| Research       | 5           | 5      | 183000.00  | 36600.00   |
| Automation     | 7           | 0      | NULL       | NULL       |

The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

## Part 2) Stored Procedure and Cursor

```
-- Drop the existing stored procedure if it exists
DROP PROCEDURE IF EXISTS SP_Report_NEW_Budget;
GO

-- Create the stored procedure
CREATE PROCEDURE SP_Report_NEW_Budget
AS
BEGIN
    -- Create the new table NEW_Dept_Budget
    CREATE TABLE NEW_Dept_Budget (
        Dept_No INT,
        Dept_Name CHAR(30),
        COUNT_Emp INT,
        New_SUM_Salary INT,
        New_AVE_Salary INT
    );

    -- Check if the view VDept_Budget is empty
    IF (SELECT COUNT(*) FROM VDept_Budget) > 0
    BEGIN
        -- Declare cursor for the view VDept_Budget
        DECLARE cur CURSOR FOR
        SELECT Dept_Number, Dept_Name, No_Emp, Sum_Salary, Ave_Salary
        FROM VDept_Budget;

        -- Declare variables to hold the cursor data
        DECLARE @Dept_No INT, @Dept_Name CHAR(30), @COUNT_Emp INT, @Sum_Salary INT,
        @Ave_Salary INT;
        DECLARE @New_SUM_Salary INT, @New_AVE_Salary INT;

        -- Open the cursor
        OPEN cur;

        -- Fetch the cursor data
        FETCH NEXT FROM cur INTO @Dept_No, @Dept_Name, @COUNT_Emp, @Sum_Salary,
        @Ave_Salary;

        -- Loop through the cursor data
        WHILE @@FETCH_STATUS = 0
        BEGIN
            -- Calculate the new sum and average salary based on department number
            IF @Dept_No = 1
                SET @New_SUM_Salary = @Sum_Salary * 1.1;
            ELSE IF @Dept_No = 4
                SET @New_SUM_Salary = @Sum_Salary * 1.2;
            ELSE IF @Dept_No = 5
                SET @New_SUM_Salary = @Sum_Salary * 1.3;
            ELSE IF @Dept_No = 7
                SET @New_SUM_Salary = @Sum_Salary * 1.4;
            ELSE
                SET @New_SUM_Salary = @Sum_Salary;

            SET @New_AVE_Salary = @New_SUM_Salary / @COUNT_Emp;

            -- Insert data into the new table
```

```

INSERT INTO NEW_Dept_Budget (Dept_No, Dept_Name, COUNT_Emp, New_SUM_Salary,
New_AVE_Salary)
VALUES (@Dept_No, @Dept_Name, @COUNT_Emp, @New_SUM_Salary, @New_AVE_Salary);

-- Fetch the next row
FETCH NEXT FROM cur INTO @Dept_No, @Dept_Name, @COUNT_Emp, @Sum_Salary,
@Ave_Salary;
END;

-- Close and deallocate the cursor
CLOSE cur;
DEALLOCATE cur;
END;
END;
GO

EXEC SP_Report_NEW_Budget;

SELECT * FROM NEW_Dept_Budget;

```

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The main window displays a query script with the following content:

```

136
137
138 -- Insert data into the new table
139 INSERT INTO NEW_Dept_Budget (Dept_No, Dept_Name, COUNT_Emp, New_SUM_Salary, New_AVE_Salary)
140 VALUES (@Dept_No, @Dept_Name, @COUNT_Emp, @New_SUM_Salary, @New_AVE_Salary);
141
142 -- Fetch the next row
143 FETCH NEXT FROM cur INTO @Dept_No, @Dept_Name, @COUNT_Emp, @Sum_Salary, @Ave_Salary;
144 END;
145
146 -- Close and deallocate the cursor
147 CLOSE cur;
148 DEALLOCATE cur;
149 END;
150 GO
151
152 EXEC SP_Report_NEW_Budget;
153
154 SELECT * FROM NEW_Dept_Budget;

```

The bottom pane shows the query results in a table with 5 columns: Dept\_No, Dept\_Name, COUNT\_Emp, New\_SUM\_Salary, and New\_AVE\_Salary. The results are as follows:

| Dept_No | Dept_Name      | COUNT_Emp | New_SUM_Salary | New_AVE_Salary |
|---------|----------------|-----------|----------------|----------------|
| 1       | Headquarters   | 3         | 170500         | 56833          |
| 2       | Administration | 3         | 111600         | 37200          |
| 3       | Research       | 5         | 237900         | 47580          |
| 4       | Automation     | 0         | NULL           | NULL           |

The status bar at the bottom indicates: "Query executed successfully." and "LAPTOP-LSLNUDIE (15.0 RTM) | LAPTOP-LSLNUDIE\Yuvar ... COMPANY\_yVagula 00:00:00 4 rows".

# PROOF OF PATH AND FILENAME

The screenshot displays the Microsoft SQL Server Management Studio interface. The main window shows a T-SQL script being executed. The script includes an INSERT statement, a cursor loop to fetch and insert data, and a final SELECT statement. The results pane at the bottom shows the output of the SELECT statement, which is a table with 5 columns: Dept\_No, Dept\_Name, COUNT\_Emp, New\_SUM\_Salary, and New\_AVE\_Salary. The status bar at the bottom indicates that the query was executed successfully, returning 4 rows.

```
136
137
138 INSERT INTO NEW_Dept_Budget (Dept_No, Dept_Name, COUNT_Emp, New_SUM_Salary, New_AVE_Salary)
139 VALUES (@Dept_No, @Dept_Name, @COUNT_Emp, @New_SUM_Salary, @New_AVE_Salary);
140
141 -- Fetch the next row
142 FETCH NEXT FROM cur INTO @Dept_No, @Dept_Name, @COUNT_Emp, @Sum_Salary, @Ave_Salary;
143
144 END;
145
146 -- Close and deallocate the cursor
147 CLOSE cur;
148 DEALLOCATE cur;
149
150 END;
151
152 GO
153
154 EXEC SP_Report_NEW_Budget;
155
156 SELECT * FROM NEW_Dept_Budget;
```

| Dept_No | Dept_Name      | COUNT_Emp | New_SUM_Salary | New_AVE_Salary |
|---------|----------------|-----------|----------------|----------------|
| 1       | Headquarters   | 3         | 170500         | 56833          |
| 2       | Administration | 3         | 111600         | 37200          |
| 3       | Research       | 5         | 237900         | 47580          |
| 4       | Automation     | 0         | NULL           | NULL           |

Query executed successfully.