

**Name: LAVU MEGHA SRAVANI**

**CSU ID: 2762646**

**Description: Creating View and stored procedure in Microsoft SQL**

**Part1 on View:**

**1-1) Create a view named VDept\_Budget that reports headcount for each department.**

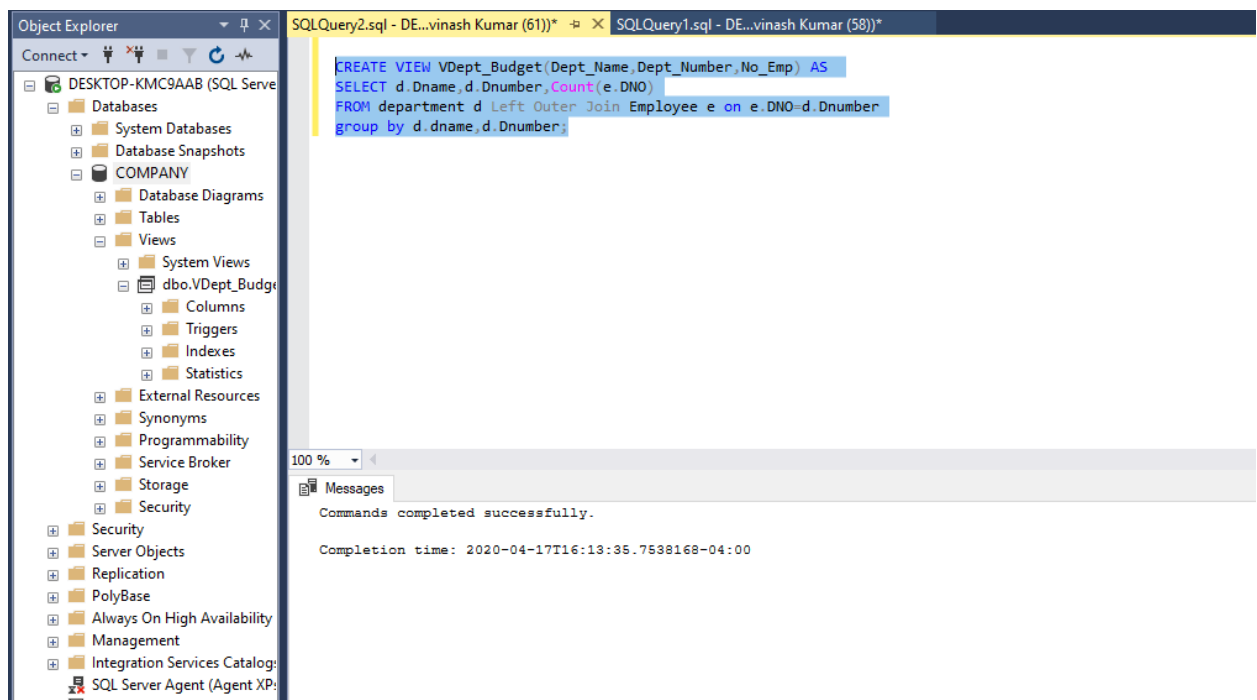
**The report includes 3 columns as follow:**

**Dept\_Name, Dept\_Number, No\_Emp.**

**Include all the departments.**

**Show the content of the view through SQL (Select \* from VDept\_Budget;)**

```
CREATE VIEW VDept_Budget(Dept_Name,Dept_Number,No_Emp) AS  
SELECT d.Dname,d.Dnumber,Count(e.DNO)  
FROM department d Left Outer Join Employee e on e.DNO=d.Dnumber  
group by d.dname,d.Dnumber;
```



```
select * from VDept_Budget v;
```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'DESKTOP-KMC9AAB (SQL Server)'. The central query window, titled 'SQLQuery2.sql - DE...vinash Kumar (61))', contains the following SQL code:

```

select * from Employee;
select * from department;
select * from PROJECT;
select * from DEPENDENT;
select * from DEPT_LOCATIONS;
select * from WORKS_ON;

DELETE from DEPENDENT where ESSN='987654321';
DELETE from WORKS_ON where ESSN='987654321';
DELETE from DEPENDENT where ESSN='453453453';

INSERT into DEPENDENT values('987654321','Abner','M','29-Feb-32','Spouse');
INSERT into WORKS_ON values('987654321','20','15');
INSERT into WORKS_ON values('987654321','30','20');

select * from VDept_Budget v;

DROP VIEW IF EXISTS VDept_Budget;

```

At the bottom, the 'Results' tab is active, displaying a table with the following data:

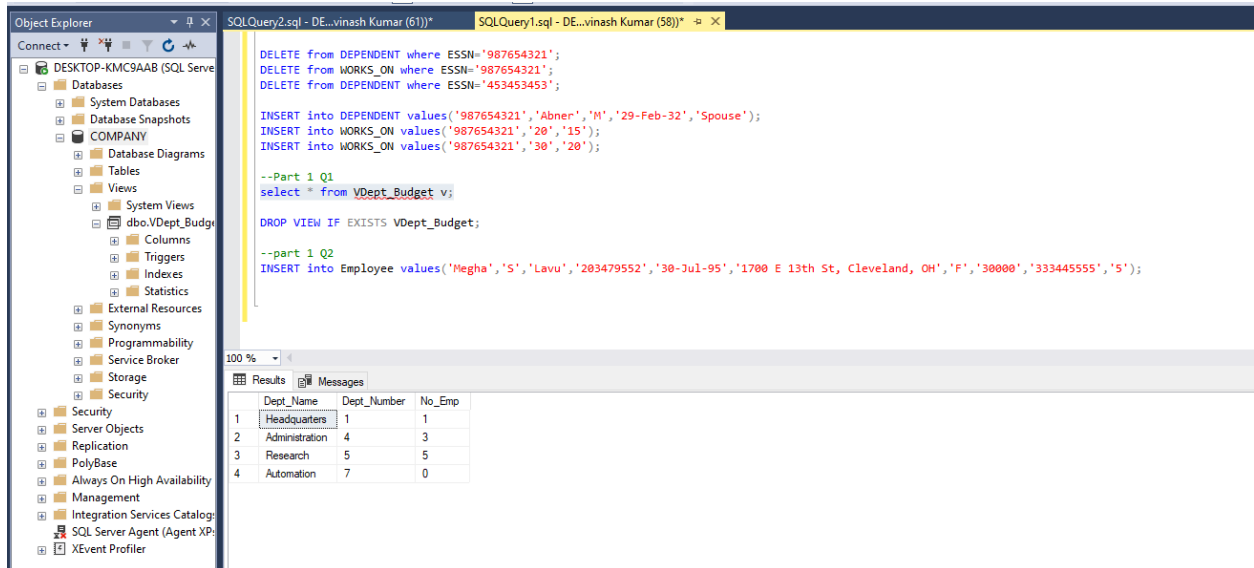
	Dept_Name	Dept_Number	No_Emp
1	Headquarters	1	1
2	Administration	4	3
3	Research	5	4
4	Automation	7	0

**1-2) Add yourself or anyone to Employee table in the database . Then Show the content of your view again through SQL (Select \* from VDept\_Budget;) to see if your view is updated according to the changes you just made in the base table Employee.**

--part 1 Q2

INSERT into Employee values('Megha','S','Lavu','203479552','30-Jul-95','1700 E 13th St, Cleveland, OH','F','30000','333445555','5');

select \* from VDept\_Budget v;



**1-3) Then Change your view to add two more info – Sum\_Salary, Ave\_Salary for each department. Include all the departments. Your report (view) lists 5 Columns as follow: Dept\_Name, Dept\_Number, No\_Emp, Sum\_Salary, Ave\_Salary**

**Show the content of your changed view to report the updated info after changes in the view and the database.**

**Show your SQL statements and the result of each step in a Word document. Add screenshots showing your SQLs and the results to show your report is updated before and after the changes in the database**

--part 1 Q3

```
ALTER VIEW VDept_Budget(Dept_Name,Dept_Number,No_Emp,Sum_Salary,Ave_Salary)
AS
SELECT d.Dname,d.Dnumber,Count(e.DNO),Sum(e.employeesalary),Avg(e.employeesalary)
FROM department d Left Outer Join Employee e on e.DNO=d.Dnumber
group by d.dname,d.Dnumber;
```

## SS CHUNG LAB ASSIGNMENT 5

Object Explorer

Connect

DESKTOP-KMC9AAB (SQL Server)

Databases

System Databases

Database Snapshots

COMPANY

Database Diagrams

Tables

Views

System Views

dbo.VDept\_Budget

Columns

Triggers

Indexes

Statistics

External Resources

Synonyms

Programmability

Service Broker

Storage

Security

Server Objects

Replication

PolyBase

Always On High Availability

Management

Integration Services Catalog

SQL Server Agent (Agent XP)

SQLQuery3.sql - DE...vinash Kumar (54))\*

SQLQuery2.sql - DE...vinash Kumar (61))\*

SQLQuery1.sql - DE...vinash Kumar (58))\*

```
--part 1 Q3
ALTER VIEW VDept_Budget (Dept_Name, Dept_Number, No_Emp, Sum_Salary, Ave_Salary) AS
SELECT d.Dname, d.Dnumber, Count(e.DNO), Sum(e.employeeesalary), Avg(e.employeeesalary)
FROM department d Left Outer Join Employee e on e.DNO=d.Dnumber
group by d.dname, d.Dnumber;
```

100 %

Messages

Commands completed successfully.

Completion time: 2020-04-17T16:36:53.9401287-04:00

`select * from VDept_Budget v;`

Object Explorer

Connect

DESKTOP-KMC9AAB (SQL Server)

Databases

System Databases

Database Snapshots

COMPANY

Database Diagrams

Tables

Views

System Views

dbo.VDept\_Budget

Columns

Triggers

Indexes

Statistics

External Resources

Synonyms

Programmability

Service Broker

Storage

Security

Server Objects

Replication

PolyBase

Always On High Availability

Management

Integration Services Catalog

SQL Server Agent (Agent XP)

XEvent Profiler

SQLQuery3.sql - DE...vinash Kumar (54))\*

SQLQuery2.sql - DE...vinash Kumar (61))\*

SQLQuery1.sql - DE...vinash Kumar (58))\*

```
select * from Employee;
select * from department;
select * from PROJECT;
select * from DEPENDENT;
select * from DEPT_LOCATIONS;
select * from WORKS_ON;

DELETE from DEPENDENT where ESSN='987654321';
DELETE from WORKS_ON where ESSN='987654321';
DELETE from DEPENDENT where ESSN='453453453';

INSERT into DEPENDENT values('987654321', 'Abner', 'M', '29-Feb-32', 'Spouse');
INSERT into WORKS_ON values('987654321', '20', '15');
INSERT into WORKS_ON values('987654321', '30', '20');

--Part 1 Q1
select * from VDept_Budget v;

DROP VIEW IF EXISTS VDept_Budget;

--part 1 Q2
```

100 %

Results

	Dept_Name	Dept_Number	No_Emp	Sum_Salary	Ave_Salary
1	Headquarters	1	1	55000	55000.000000
2	Administration	4	3	93000	31000.000000
3	Research	5	5	163000	32600.000000
4	Automation	7	0	NULL	NULL

## **Part2 on Stored Procedure using Cursor**

**Write a Stored Procedure SP\_Report\_NEW\_Budget using the view you created in Part 1-3).**

**Use CURSOR to write the stored Procedure for the tasks below.**

**Your Stored Procedure SP\_Report\_NEW\_Budget does the following tasks:**

**It creates a new table NEW\_Dept\_Budget as follow:**

**NEW\_Dept\_Budget has 5 columns**

**Dept\_No (Int)**

**Dept\_Name (Char(30))**

**COUNT\_Emp (INT)**

**New\_SUM\_Salary (INT)**

**New\_AVE\_Salary (INT)**

- 1) Check if the view VDept\_Budget is empty or not (by counting rows from the view).**
- 2) If not empty , For each row of the view VDept\_Budget, Calculate New\_SUM\_Salary, New\_AVE\_Salary as follow:**
  - If Dept = 1, Increase SUM(salary) 10%**
  - If Dept = 4, Increase SUM(salary) 20%**
  - If Dept = 5, Increase SUM(salary) 30%**
  - If Dept = 7, Increase SUM(salary) 40%**
- 3) Insert each column value into the new table NEW\_Dept\_Budget.**

**Output (snapshot of screen) each Function and Procedure created and Show all the tables contents with Select statement in your output.**

```
CREATE PROCEDURE SP_Report_NEW_Budget
AS
BEGIN
```

```
DECLARE @depno Int
DECLARE @depname Char(30)
DECLARE @count Int
DECLARE @sum_salary Int
DECLARE @avg_salary Int
Declare @rowcount Int
```

```
CREATE TABLE NEW_Dept_Budget
```

```
(
    Dept_No Int,
    Dept_Name char(30),
    COUNT_Emp Int,
    NEW_SUM_Salary Int,
    NEW_AVE_Salary Int,
)
IF EXISTS (select * from VDept_Budget v)
```

```
-- checks whether the view is empty or not
Select @rowcount=count(*)
from VDept_Budget v
--PRINT @rowcount
```

```
IF @rowcount > 0
```

```
BEGIN
```

```
DECLARE deptno_cursor cursor For
select v.Dept_Name, v.Dept_Number,v.No_Emp,v.Sum_Salary,v.Ave_Salary
from VDept_Budget v
```

```
OPEN deptno_Cursor
BEGIN
```

```
FETCH NEXT FROM deptno_cursor into
@depname,@depno,@count,@sum_salary,@avg_salary
while @@FETCH_STATUS =0
BEGIN
    if @depno=1
    begin
        SET @sum_salary= @sum_salary*1.1
        SET @avg_salary= @avg_salary*1.1
```

```
        end
        else if @depno=4
        begin
            SET @sum_salary= @sum_salary*1.2
            SET @avg_salary= @avg_salary*1.2
        end
        else if @depno=5
        begin
            SET @sum_salary= @sum_salary*1.3
            SET @avg_salary= @avg_salary*1.3
        end
        else if @depno=7
        begin
            SET @sum_salary= @sum_salary*1.4
            SET @avg_salary= @avg_salary*1.4
        end
        INSERT INTO NEW_Dept_Budget
values(@depno,@depname,@count,@sum_salary,@avg_salary)
        FETCH NEXT FROM deptno_cursor into
        @depname,@depno,@count,@sum_salary,@avg_salary

        END
        END
        CLOSE deptno_Cursor
        END

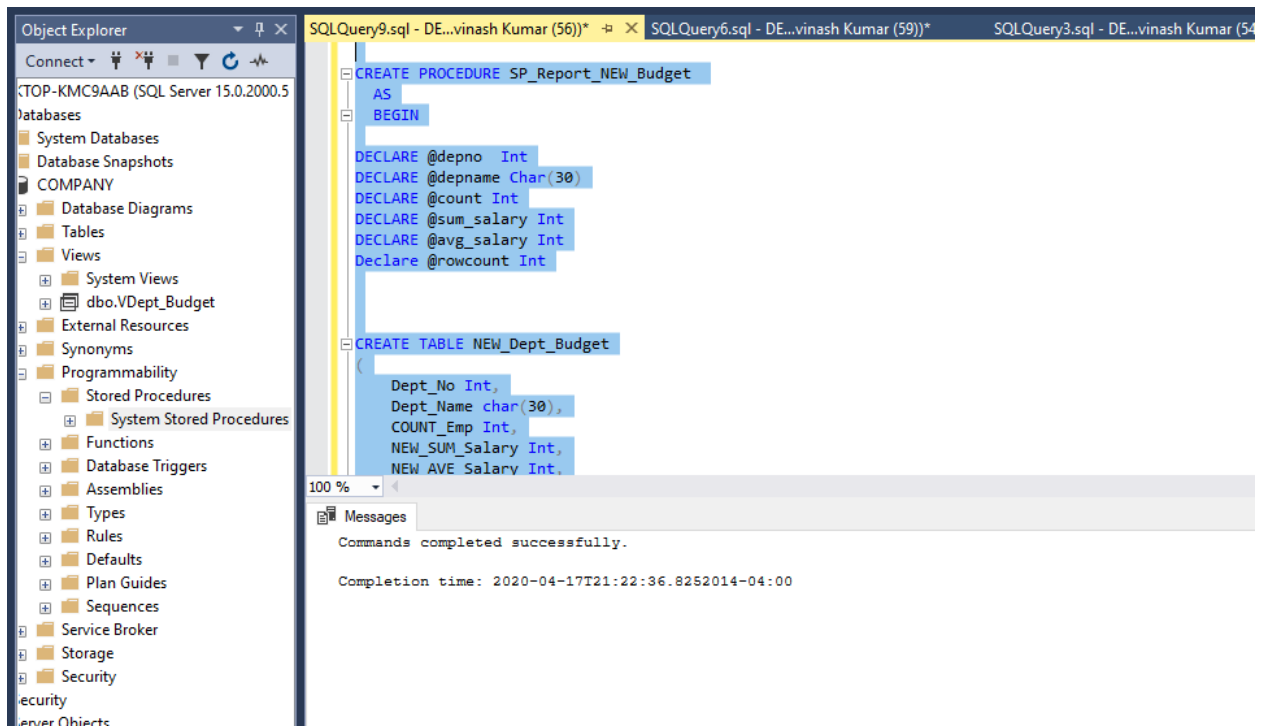
        ELSE
        PRINT 'TABLE IS EMPTY'

        DEALLOCATE deptno_Cursor

        Select * from NEW_Dept_Budget

    END
```

SS CHUNG  
LAB ASSIGNMENT 5



EXEC SP\_Report\_NEW\_Budget



SS CHUNG  
LAB ASSIGNMENT 5

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'COMPANY', including tables, views, and stored procedures. The central query window, titled 'SQLQuery9.sql - DE...vinash Kumar (56))', contains the following T-SQL code:

```
END  
END  
CLOSE deptno_Cursor  
END  
  
ELSE  
PRINT 'TABLE IS EMPTY'  
  
DEALLOCATE deptno_Cursor  
  
Select * from NEW_Dept_Budget  
END  
  
DROP TABLE NEW_Dept_Budget  
  
EXEC SP_Report_NEW_Budget
```

Below the query window, the 'Results' tab is active, showing a grid with the following data:

	Dept_No	Dept_Name	COUNT_Emp	NEW_SUM_Salary	NEW_AVE_Salary
1	1	Headquarters	1	60500	60500
2	4	Administration	3	111600	37200
3	5	Research	5	211900	42380
4	7	Automation	0	NULL	NULL