Lab Assignment 5

CIS430/CIS530
Dr. Sunnie S. Chung

Lab5 has two parts.

Part1: View

There are two ways to save a SQL query result permanently in a SQL Server so that the saved query result can be used in the From clause of any SQLs.

1) **Create View as Select ...**
2) **Insert Into ... Select ...**

1. Saving a Result of a SQL Permanently in a SQL Server to Be Used later

1-1) View Creation from SQL

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 the query with the view (Select * from **VDept_Budget**;)

1-2) Table Creation from the same SQL as in 1-1)

Create a Table named **Dept_Budget** with the Same Column names as in 1-1) and Populated the Table from the same SQL Result in 1-1) using "Insert Into … Select …" to Compare with the content of View in 1-1).
Show the content of the table through SQL (Select * From **Dept_Budget;)**

2. Testing to see whether the View is automatically updated by the server when the state of the underlying base table is changed while the user table that that was created and physically stored in the file is not.

Add 2 new employees (yourself and one more new employee) to the department **dno = 1** with each **salary $50,000** to change the state of the base table Employee in the database.

2-1) Then Show the content of your View again through (Select * from **VDept_Budget**;) to see if your view is updated according to the changes that you just made in the base table Employee.

2-2) Show the content of the table through SQL (Select * From **Dept_Budget;)**
Compare the difference between the content of the View (**VDept_Budget**) and the content of the user table **Dept_Budget** for update when the content of the underlying base table has been changed.

3. Difference in Changing the Scheme of the Existing View and the Table

Then Change your existing view to add two more columns – 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.

Do the same for the Table **Dept_Budget**.
Show the content of your changed Table to report the updated info after changes in the table.

Show your SQL statements and the result of each step in a word document. Add screenshots showing your SQLs and the results to show:

1) Whether your view is updated after the changes in the underlying base table
2) Whether your table is updated after the changes in the underlying base table

Part2: Stored Procedure and 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:

1) It creates a new table **NEW_Dept_Budget as below:**

   NEW_Dept_Budget has 5 columns

   Dept_No (Int)
   Dept_Name (Char(30))
   COUNT_Emp (INT)
   New_SUM_Salary (INT)
   New_AVE_Salary (INT)

2) Check if the view **VDept_Budget** is empty or not (by counting rows from the view).

3) If not empty,

   For each row of the view **VDept_Budget,**

   populate (Insert) the new table **NEW_Dept_Budget** from the view **VDept_Budget**

and New_SUM_Salary, New_AVE_Salary with the newly calculated two columns as calculated below.

Calculate New_SUM_Salary, New_AVE_Salary as follow:

If Dept = 1, Increase SUM (Salary) by 10%
If Dept = 4, Increase SUM (Salary) by 20%
If Dept = 5, Increase SUM (Salary) by 30%
If Dept = 7, Increase SUM (Salary) by 40%

You can also do this task in your stored procedure by populating (insert) the new table directly from the view first then update two columns of the new table with the newly calculated new Sum and new Ave later.

Output in screen captures for the Stored Procedure created and the execution of the Stored Procedure by showing the contents of the View and the new table in Select statements in your output.

COMPANY DATABASE

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	09-Jan-55	731 Fondren, Houston, TX	M	30000	987654321	5
Franklin	T	Wong	333445555	08-Dec-45	638 Voss, Houston, TX	M	40000	888665555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice, Houston, TX	F	25000	333445555	5
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak, Humble, TX	M	38000	333445555	5
James	E	Borg	888665555	10-Nov-27	450 Stone, Houston, TX	M	55000		1
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry, Bellaire, TX	F	43000	888665555	4
Ahmad	V	Jabbar	987987987	29-Mar-59	980 Dallas, Houston, TX	M	25000	987654321	4
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle, SPring, TX	F	25000	987654321	4

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
Headquarters	1	888665555	19-Jun-71
Administration	4	987654321	01-Jan-85
Research	5	333445555	22-May-78
Automation	7	123456789	06-Oct-05

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
123456789	Alice	F	31-Dec-78	Daughter
123456789	Elizabeth	F	05-May-57	Spouse
123456789	Michael	M	01-Jan-78	Son
333445555	Alice	F	05-Apr-76	Daughter
333445555	Joy	F	03-May-48	Spouse
333445555	Theodore	M	25-Oct-73	Son
987654321	Abner	M	29-Feb-32	Spouse

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

WORKS_ON

ESSN	PNO	Hours
123456789	1	32.5
123456789	2	7.5
333445555	2	10
333445555	3	10
333445555	10	10
333445555	20	10
453453453	1	20
453453453	2	20
666884444	3	40
888665555	20	
987654321	20	15
987654321	30	20
987987987	10	35
987987987	30	5
999887777	10	10
999887777	30	30