Qiaoyi Yang
Nov 22th, 2021
IT Foundations of Database Management
Assignment 06

Assignment 06 – Views, Functions, and Stored Procedures

Introduction

The sixth assignment of this class is about how to creating the view. One thing need to know is that creating the view does not run the select code, so the **Create statement results are its success**, not data.
The assignment questions were really help me to understand the concept. I have learned how to creating base views, including using schema binding to protect a parent table from being dropped, how functions can use parameters to change the results of the query as it is executed, and the scalars functions. I also learned how to creating reporting functions, and creating reporting stored procedures.

Questions

1. ***Explain when you would use a SQL View.***

   Views are virtual tables that can be a great way to optimize your database experience. Not only are views good for defining a table without using extra storage, but they also accelerate data analysis and can provide your data extra security.

2. ***Explain are the differences and similarities between a View, Function, and Stored Procedure.***

   View is simple showcasing data stored in the database tables whereas a stored procedure is a group of statements that can be executed.
   A view is faster as it displays data from the tables referenced whereas a store procedure executes SQL statements.
   In a function, it is mandatory to use the RETURNS and RETURN arguments, whereas in a stored procedure is not necessary. In few words, a stored procedure is more flexible to write any code that you want, while functions have a rigid structure and functionality.

```
-- Question 1 (5% pts): How can you create BACIC views to show
data from each table in the database.
-- NOTES: 1) Do not use a *, list out each column!
--        2) Create one view per table!
--        3) Use SchemaBinding to protect the views from being
orphaned!

Create View vCategories
With SchemaBinding
As
Select CategoryID, CategoryName
From dbo.Categories;
go
Select * From vCategories;
```

	CategoryID	CategoryName
1	1	Beverages
2	2	Condiments
3	3	Confections
4	4	Dairy Products
5	5	Grains/Cereals
6	6	Meat/Poultry
7	7	Produce
8	8	Seafood

```
Create
View vProducts
With SchemaBinding
As
Select ProductID, ProductName, CategoryID, UnitPrice
From dbo.Products;
go
Select * From vProducts;
```

	ProductID	ProductName	CategoryID	UnitPrice
1	1	Chai	1	18.0000
2	2	Chang	1	19.0000
3	3	Aniseed Syrup	2	10.0000
4	4	Chef Anton's Cajun Sea…	2	22.0000
5	5	Chef Anton's Gumbo Mix	2	21.3500
6	6	Grandma's Boysenberry …	2	25.0000
7	7	Uncle Bob's Organic Dr…	7	30.0000
8	8	Northwoods Cranberry S…	2	40.0000
9	9	Mishi Kobe Niku	6	97.0000
1…	10	Ikura	8	31.0000
1…	11	Queso Cabrales	4	21.0000

```
Create
View vEmployees
With SchemaBinding
As
Select EmployeeID, EmployeeFirstName, EmployeeLastName,
ManagerID
From dbo.Employees;
go
Select * From vEmployees;
```

	EmployeeID	EmployeeFirstName	EmployeeLastName	ManagerID
1	1	Nancy	Davolio	2
2	2	Andrew	Fuller	2
3	3	Janet	Leverling	2
4	4	Margaret	Peacock	2
5	5	Steven	Buchanan	2
6	6	Michael	Suyama	5
7	7	Robert	King	5
8	8	Laura	Callahan	2
9	9	Anne	Dodsworth	5

```
Create
View vInventories
With SchemaBinding
```

```
As
Select InventoryID, InventoryDate, EmployeeID, ProductID,
[Count]
From dbo.Inventories;
go
Select * From vInventories;
```

	InventoryID	InventoryDate	EmployeeID	ProductID	Count
1	1	2017-01-01	5	1	39
2	2	2017-01-01	5	2	17
3	3	2017-01-01	5	3	13
4	4	2017-01-01	5	4	53
5	5	2017-01-01	5	5	0
6	6	2017-01-01	5	6	120
7	7	2017-01-01	5	7	15
8	8	2017-01-01	5	8	6
9	9	2017-01-01	5	9	29
1…	10	2017-01-01	5	10	31
1…	11	2017-01-01	5	11	22
1…	12	2017-01-01	5	12	86
1…	13	2017-01-01	5	13	24
1…	14	2017-01-01	5	14	35
1…	15	2017-01-01	5	15	39

```
-- Question 2 (5% pts): How can you set permissions, so that the
public group CANNOT select data
-- from each table, but can select data from each view?
Use Assignment06DB_QiaoyiYang;

Deny Select On Categories to Public;
Grant Select On vCategories to Public;

Deny Select On Products to Public;
Grant Select On vProducts to Public;

Deny Select On Employees to Public;
Grant Select On vEmployees to Public;
```

```
Deny Select On Inventories to Public;
Grant Select On vInventories to Public;

go
```

```
-- Question 3 (10% pts): How can you create a view to show a
list of Category and Product names,
-- and the price of each product?
-- Order the result by the Category and Product!

-- Here is an example of some rows selected from the view:
-- CategoryName ProductName       UnitPrice
-- Beverages    Chai              18.00
-- Beverages    Chang             19.00
-- Beverages    Chartreuse verte  18.00

Create View vProductsByCategories
As
Select Top 100 Percent
CategoryName, ProductName, UnitPrice From Products as p
Join Categories as c
On p.CategoryID = c.CategoryID
Order By CategoryName, ProductName
go
Select * From vProductsByCategories;
```

	CategoryName	ProductName	UnitPrice
1	Beverages	Chai	18.0000
2	Beverages	Chang	19.0000
3	Condiments	Aniseed Syrup	10.0000
4	Condiments	Chef Anton's Cajun Sea…	22.0000
5	Condiments	Chef Anton's Gumbo Mix	21.3500
6	Condiments	Grandma's Boysenberry …	25.0000
7	Produce	Uncle Bob's Organic Dr…	30.0000
8	Condiments	Northwoods Cranberry S…	40.0000
9	Meat/Poultry	Mishi Kobe Niku	97.0000
1…	Seafood	Ikura	31.0000
1…	Dairy Products	Queso Cabrales	21.0000
1…	Dairy Products	Queso Manchego La Past…	38.0000
1…	Seafood	Konbu	6.0000

```
-- Question 4 (10% pts): How can you create a view to show a
list of Product names
-- and Inventory Counts on each Inventory Date?
-- Order the results by the Product, Date, and Count!

-- Here is an example of some rows selected from the view:
-- ProductName    InventoryDate Count
-- Alice Mutton   2017-01-01      0
-- Alice Mutton   2017-02-01      10
-- Alice Mutton   2017-03-01      20
-- Aniseed Syrup    2017-01-01    13
-- Aniseed Syrup    2017-02-01    23
-- Aniseed Syrup    2017-03-01    33

Create View vInventoriesByProductsByDates
As
Select Top 100 Percent
ProductName, InventoryDate, Count From Inventories as i
Join Products as p
On i.ProductID = p.ProductID
Order By ProductName, InventoryDate, Count
go
Select * From vInventoriesByProductsByDates;
```

	ProductName	InventoryDate	Count
1	Chai	2017-01-01	39
2	Chai	2017-02-01	49
3	Chai	2017-03-01	59
4	Chang	2017-03-01	37
5	Chang	2017-02-01	27
6	Chang	2017-01-01	17
7	Aniseed Syrup	2017-01-01	13
8	Aniseed Syrup	2017-02-01	23
9	Aniseed Syrup	2017-03-01	33
10	Chef Anton's Cajun Sea…	2017-03-01	73
11	Chef Anton's Cajun Sea…	2017-02-01	63

```
-- Question 5 (10% pts): How can you create a view to show a
list of Inventory Dates
-- and the Employee that took the count?
-- Order the results by the Date and return only one row per
date!

-- Here is are the rows selected from the view:

-- InventoryDate    EmployeeName
-- 2017-01-01       Steven Buchanan
-- 2017-02-01       Robert King
-- 2017-03-01       Anne Dodsworth

Create View vInventoriesByEmployeesByDates
As
Select DISTINCT Top 100 Percent
InventoryDate, [EmployeeName] = EmployeeFirstName + ' ' +
EmployeeLastName From Inventories as i
Join Employees as e
On i.EmployeeID = e.EmployeeID
Order By InventoryDate, EmployeeName;
go
Select * From vInventoriesByEmployeesByDates
```

	InventoryDate	EmployeeName
1	2017-01-01	Steven Buchanan
2	2017-02-01	Robert King
3	2017-03-01	Anne Dodsworth

```
-- Question 6 (10% pts): How can you create a view show a list
of Categories, Products,
-- and the Inventory Date and Count of each product?
-- Order the results by the Category, Product, Date, and Count!

-- Here is an example of some rows selected from the view:
-- CategoryName,ProductName,InventoryDate,Count
-- CategoryName ProductName InventoryDate   Count
-- Beverages      Chai        2017-01-01      39
-- Beverages      Chai        2017-02-01      49
-- Beverages      Chai        2017-03-01      59
-- Beverages      Chang       2017-01-01      17

Create View vInventoriesByProductsByCategories
As
Select Top 100 Percent
CategoryName,ProductName,InventoryDate,Count
From Inventories as i
Join Products as p
On i.ProductID = p.ProductID
Join Categories as c
On p.CategoryID = c.CategoryID
Order By CategoryName,ProductName,InventoryDate,Count;
go
Select * From vInventoriesByProductsByCategories
```

	CategoryName	ProductName	InventoryDate	Count
1	Beverages	Chai	2017-01-01	39
2	Beverages	Chang	2017-01-01	17
3	Condiments	Aniseed Syrup	2017-01-01	13
4	Condiments	Chef Anton's …	2017-01-01	53
5	Condiments	Chef Anton's …	2017-01-01	0
6	Condiments	Grandma's Boy…	2017-01-01	120
7	Produce	Uncle Bob's O…	2017-01-01	15
8	Condiments	Northwoods Cr…	2017-01-01	6
9	Meat/Poultry	Mishi Kobe Ni…	2017-01-01	29
1…	Seafood	Ikura	2017-01-01	31
1…	Dairy Products	Queso Cabrales	2017-01-01	22

```sql
-- Question 7 (10% pts): How can you create a view to show a list of Categories, Products,
-- the Inventory Date and Count of each product, and the EMPLOYEE who took the count?
-- Order the results by the Inventory Date, Category, Product and Employee!

Create View vInventoriesByProductsByEmployees
As
Select Top 100 Percent
CategoryName,ProductName,InventoryDate,Count, [EmployeeName] = EmployeeFirstName + ' ' + EmployeeLastName
From Inventories as i
Join Products as p
On i.ProductID = p.ProductID
Join Categories as c
On p.CategoryID = c.CategoryID
Join Employees as e
On i.EmployeeID = e.EmployeeID
Order By CategoryName,ProductName,InventoryDate,Count, EmployeeName;
go
--Select * From vInventoriesByProductsByEmployees
```

	CategoryName	ProductName	InventoryDate	Count	EmployeeName
1	Beverages	Chai	2017-01-01	39	Steven Buchanan
2	Beverages	Chang	2017-01-01	17	Steven Buchanan
3	Condiments	Aniseed Syrup	2017-01-01	13	Steven Buchanan
4	Condiments	Chef Anton's Cajun Sea…	2017-01-01	53	Steven Buchanan
5	Condiments	Chef Anton's Gumbo Mix	2017-01-01	0	Steven Buchanan
6	Condiments	Grandma's Boysenberry …	2017-01-01	120	Steven Buchanan
7	Produce	Uncle Bob's Organic Dr…	2017-01-01	15	Steven Buchanan
8	Condiments	Northwoods Cranberry S…	2017-01-01	6	Steven Buchanan
9	Meat/Poultry	Mishi Kobe Niku	2017-01-01	29	Steven Buchanan

```sql
-- Question 8 (10% pts): How can you create a view to show a
list of Categories, Products,
-- the Inventory Date and Count of each product, and the
Employee who took the count
-- for the Products 'Chai' and 'Chang'?

Create View vInventoriesForChaiAndChangByEmployees
As
Select Top 100 Percent
CategoryName,ProductName,InventoryDate,Count, [EmployeeName] =
EmployeeFirstName + ' ' + EmployeeLastName
From Inventories as i
Join Products as p
On i.ProductID = p.ProductID
Join Categories as c
On p.CategoryID = c.CategoryID
Join Employees as e
On i.EmployeeID = e.EmployeeID
Where
   i.ProductID in (Select ProductID From Products Where
ProductName in ('Chai','Chang'))
Order By CategoryName,ProductName,InventoryDate,Count,
EmployeeName;
go
--Select * From vInventoriesForChaiAndChangByEmployees
```

	CategoryName	ProductName	InventoryDate	Count	EmployeeName
1	Beverages	Chai	2017-01-01	39	Steven Buchanan
2	Beverages	Chang	2017-01-01	17	Steven Buchanan
3	Beverages	Chai	2017-02-01	49	Robert King
4	Beverages	Chang	2017-02-01	27	Robert King
5	Beverages	Chai	2017-03-01	59	Anne Dodsworth
6	Beverages	Chang	2017-03-01	37	Anne Dodsworth

```
-- Question 9 (10% pts): How can you create a view to show a
list of Employees and the Manager who manages them?
-- Order the results by the Manager's name!

Create View vEmployeesByManager
As
Select Top 100 Percent
[Manager] = m.EmployeeFirstName +' '+ m.EmployeeLastName,
[Employee] = e.EmployeeFirstName +' '+ e.EmployeeLastName
From Employees as e
Join Employees as m
On e.ManagerID = m.EmployeeID
Order By Manager, Employee;

Select * From vEmployeesByManager;
```

	Manager	Employee
1	Andrew Fuller	Nancy Davolio
2	Andrew Fuller	Andrew Fuller
3	Andrew Fuller	Janet Leverling
4	Andrew Fuller	Margaret Peacock
5	Andrew Fuller	Steven Buchanan
6	Steven Buchanan	Michael Suyama
7	Steven Buchanan	Robert King
8	Andrew Fuller	Laura Callahan
9	Steven Buchanan	Anne Dodsworth

```
-- Question 10 (20% pts): How can you create one view to show
all the data from all four
-- BASIC Views? Also show the Employee's Manager Name and order
the data by
-- Category, Product, InventoryID, and Employee.

Create View vInventoriesByProductsByCategoriesByEmployees
As
Select Top 100 Percent
c.CategoryID, c.CategoryName,
p.ProductID, p.ProductName, p.UnitPrice,
i.InventoryID, i.InventoryDate, i.Count,
e.EmployeeID, [Employee] = e.EmployeeFirstName + ' ' +
e.EmployeeLastName,
[Manager] = m.EmployeeFirstName +' '+ m.EmployeeLastName

From Categories as c
Join Products as p
On c.CategoryID = p.CategoryID
Join Inventories as i
On p.ProductID = i.ProductID
Join Employees as e
On e.EmployeeID = i.EmployeeID
Join Employees as m
On e.ManagerID = m.EmployeeID

Order By CategoryID,ProductID,InventoryID,Count, EmployeeID;
go
Select * From vInventoriesByProductsByCategoriesByEmployees;
```

CategoryID	CategoryName	ProductID	ProductName	UnitPrice	InventoryID	InventoryDate	Count	EmployeeID	Employee	Manager
1	Beverages	1	Chai	18.0000	1	2017-01-01	39	5	Steven Bucha n	Andrew Fuller
1	Beverages	2	Chang	19.0000	2	2017-01-01	17	5	Steven Bucha n	Andrew Fuller
2	Condiments	3	Aniseed Syrup	10.0000	3	2017-01-01	13	5	Steven Bucha n	Andrew Fuller
2	Condiments	4	Chef Anton's Cajun Sea…	22.0000	4	2017-01-01	53	5	Steven Bucha n	Andrew Fuller
2	Condiments	5	Chef Anton's Gumbo Mix	21.3500	5	2017-01-01	0	5	Steven Bucha n	Andrew Fuller
2	Condiments	6	Grandma's Boysenberry …	25.0000	6	2017-01-01	120	5	Steven Bucha n	Andrew Fuller
7	Produce	7	Uncle Bob's Organic Dr…	30.0000	7	2017-01-01	15	5	Steven Bucha n	Andrew Fuller
2	Condiments	8	Northwoods Cranberry S…	40.0000	8	2017-01-01	6	5	Steven Bucha n	Andrew Fuller
6	Meat/Poultry	9	Mishi Kobe Niku	97.0000	9	2017-01-01	29	5	Steven Bucha n	Andrew Fuller
8	Seafood	10	Ikura	31.0000	10	2017-01-01	31	5	Steven Bucha n	Andrew Fuller
4	Dairy Products	11	Queso Cabrales	21.0000	11	2017-01-01	22	5	Steven Bucha n	Andrew Fuller

Summary

This week's coding assignment was based on what we have learned from the past. I can create the report view depend on what I need. Also, I learned the different among a view, function, and stored procedure. These are all practical skills in the real life.