CSSE333 Introduction to Databases – Lab Assignment

Lab 6: Views

Objective

In this lab you will learn how to create a database view, and also see an example of how a view can be useful. You will create a simple view in Server Management Studio using the new view wizard. After that, you will then use a template of a view in order to create another view. Finally, you will tackle a slightly more difficult exercise by creating a view in Transact-SQL.

Required Materials

- Your copy of Northwind.

Related Reading

- In Books Online, the section on the CREATE VIEW Transact-SQL statement.

Turn-in Instructions

Create a single zip or rar archive with 2 .sql files containing your query made from the View template in part (2) and CREATE VIEW query in part (3) and the CSV file from part (3) below. Upload that archive into the Lab 6 Drop Box.

Assignment Details

Views are a great way to give a user read access to only a portion of the rows in a table. For example, some newspapers and magazines will make articles from their current issue available online, but back issues can only be accessed for a fee. This would be analogous to reading a view constrained to a certain date range and presenting only those items for free access.

Your first task is to create a view that shows product information from a single supplier.

(1) Creating a View with the View Wizard

i. Open Server Management Studio and select your copy of Northwind.

ii. Right click the Views folder and select New View.

iii. Use the wizard to create a view on the Products table with the ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, and Discontinued columns. Only select rows for products supplied by Formaggi (SupplierID 14). Name the View “FormaggiProductsView”. On the last page of the wizard, take a moment to look over the CREATE VIEW SQL statement. You will be writing a similar statement in the last section. Finish the wizard.

iv. Right click on your new view in the Views list. Select Open View. This should open a window showing the contents of the view. You should see three products: Gorgonzola Telino, Mascarpone Fabioli, and Mozzarella di Giovanni.

As you saw in the last section, a View is a mix between a query and a table. Very often, you will store information in a very normalized state with related data spread across multiple tables to avoid duplication and to maintain structure. While this keeps your database from getting too big or corrupted, it also means that your queries will be burdened with several joins. Views provide a way to present common join queries as if they were a table, making other select and update queries simpler to write (though not necessarily any faster running since the same calculation must be done).

Next, you will use a template to create a view showing a certain number of orders that are being shipped to the United Kingdom. Your query will return rows from the Orders table, but there will also be columns indicating what shipping company aided with shipping the orders.

(2) Creating a View using SQL Server Management Studio's templates

i. On the right side of your SQL Server Management Studio should be a window showing the Solution Explorer. Select the Template Explorer.

ii. Find the Create View template and select it.

iii. You should now see the template for a view. Use this template to create a query that will return all rows from the Orders table where the OrderID is between 10400 and 10600 and the order is being shipped to the United Kingdom. The columns that should appear in your view are OrderID, CustomerID, CompanyName, ShipName, ShipCity, and ShipCountry. This query should return 17 rows. Be sure to save this query as "UKViewQuery_<*username*>".

Your last task is to create a view using Transact-SQL that joins multiple tables. You will create a view that returns rows from the Products table, but instead of SupplierID and CategoryID columns, the there will be columns containing the supplier name and category name. At the end of the lab you will turn in the SQL command for creating this view, along with the CSV file showing the results, and the SQL command for creating the view from the previous part.

(3) Creating a View With Transact-SQL

i. Open a new query window by hitting the "New Query" button.

ii. Use a CREATE VIEW statement to create a view called FriendlyProducts. Your view should contain the ProductID, ProductName, QuantityPerUnit and UnitPrice columns from the Products table. It should also contain the CompanyName column from the Suppliers table and the CategoryName column from the Categories table. Your view should only contain products that are not discontinued. This query should return 69 items. Be sure to save this query as "FriendlyProductsQuery_<*username*>".

iii. Using the sp_depends system stored procedure, list the dependencies of your new view. You should see the pertinent columns from the Products, Suppliers and Categories tables.

iv. Still in Query Analyzer, write a select statement that queries rows from the FriendlyProducts view where the UnitPrice column is greater than 100. This query should only return one item: Côte de Blaye. Save the results of this query as a CSV file. The name of the file should be

"UnitPriceQuery_<*username*>." Notice how having the view allows you to easily access the products' related information without having to perform a join in your query, but also without having to rewrite the join on the Products, Suppliers and Categories tables.

(4) Please complete the anonymous lab feedback survey for Lab 6 on Angel under Materials -> Lab Feedback. Your feedback will help us improve the labs for future students.

Turn-in Instructions

Create a single zip or rar archive with 2 .sql files containing your query made from the View template in part (2) and CREATE VIEW query in part (3) and the CSV file from part (3) above. Upload that archive into the Lab 6 Drop Box.

Revision History

December 8, 2006:	Updated for SQL Server 2005 and revised tasks, Chris Besier
Jan. 4, 2006:	Clarified instructions, Curt Clifton and Steve Chenoweth
Jan 6, 2005:	Added feedback survey instruction.
Jan 5, 2005:	Review and augmentations to "usefulness of views paragraph" by David Yip.
Jan 5, 2005:	Created by Andy Cooper.