

Views

Exercise Handout

Contents

Overview	2
Objectives.....	2
Setup: Launch SQL Server Management Studio (if necessary).....	2
Exercise 1: Use a predefined view	3
Task 1: Examine a complex query	3
Task 2: Use a view.....	3
Exercise 2: Create a telephone directory view	4
Task 1: Write a query that retrieves Customer contact details.....	4
Task 2: Create a new view in the database	4
Task 3: Write a query that uses your new view	4

Overview

In this set of exercises, you will first use an existing view to see just how much it simplifies a complex query.

In the second exercise, you will create your own view.

Objectives

At the end of this lab, you will be able to:

- use a predefined view.
- write and use a view.

Setup: Launch SQL Server Management Studio (if necessary)

1. Launch the virtual machine.
2. Launch SQL Server Management Studio.
3. Connect to the server.

Exercise 1: Use a predefined view

You want to write a query that retrieves full information about orders, including the name of the employee who placed the order, the name of the product that was ordered, details about the company name the order was for, which address the order was to be shipped to, calculations about the order total after applying discounts and much more. You will use a view to do this.

The main tasks for this exercise are as follows:

1. Examine a prewritten query and note its complexity.
2. Use a view that retrieves the same data.

Task 1: Examine a complex query

1. Open the provided script file 'Ex01Task01.sql'.
2. Execute the query and verify that 2155 rows are returned.
3. Add a where clause to filter the results only for UK customers (it's the Country column within the Customers – it's in there somewhere!). You'll need to ensure that you use the customers table's alias because there are at least two Country columns in the tables the query uses.
4. Confirm that 135 rows are returned.
5. Now imagine having to rewrite that query every time you wanted full details about an order or filter or sort it in a different way!

Task 2: Use a view

1. Create a new query called "UseView.sql".
2. Write a select statement that retrieves every row and column from the Invoices table.
3. Execute the query and verify that 2155 rows are returned – the same as the previous complex query. The Invoices "table" is actually a view, which is a stored version of the query you just looked at.
4. Add a where clause to filter the results only for UK customers. You don't need to worry about which table it's coming from this time, because as far as the view is concerned, there's only the one Country column.
5. Experiment with sorting and filtering options and only selecting some of the columns.

Exercise 2: Create a telephone directory view

In an earlier lab, you created a query to generate a combined contact directory for Northwind Traders. In this exercise, rather than having it stored in a script file, you will store it in the database as a view.

The main tasks for this exercise are as follows:

1. Create (or reuse) a query that retrieves the company name, contact name and telephone number of all customers, suppliers, and employees.
2. Store that query as a view in the database.
3. Use the view in another query.

Task 1: Write a query that retrieves Customer contact details

1. Create a new query and save it with a name of "DirectoryView.sql".
2. If you completed an earlier lab exercise on UNIONs, copy the directory query into your new script. If not, copy the content of the provided file Ex02Starter.txt.
3. Execute the query and verify that it returns 129 rows.

Task 2: Create a new view in the database

1. Immediately before the first SELECT statement of the contact directory query, add the appropriate commands to create a new view called 'ContactDirectory'.
2. Run the script and confirm that you get a message saying "command(s) completed successfully".

Task 3: Write a query that uses your new view

1. Create a new script file called "UseMyView.sql".
2. Write a query that selects all columns from the ContactDirectory view. Make sure it retrieves 129 rows.
3. Add a where clause to only select those contacts with a contact name that starts with the letter 'a'. You'll need to use a like operator.
4. Execute the query and verify that 14 contacts are returned, two of which are Northwind employees.



QA.com