

# Report on VIEWS IN SQL

- **Name:** Muhammad Ahmad
- **Roll No:** 2022-CS-144
- **Section:** C
- **Instructor:** Nazeeful-Haq

## Task Overview

This report outlines the execution and outcomes of various tasks performed on the Northwind database. Each task involved operations such as creating views, renaming views, dropping views, retrieving data from views, altering views, refreshing views, schema binding, encrypting views, and executing DML queries within views.

## Task 1: Creating SQL View

### SQL Code

```
--      Create SQL VIEW in SQL Server:

-- This view selects specific columns from the Customers table

CREATE VIEW CustomerView
AS
SELECT CustomerID, CompanyName, ContactName, ContactTitle
FROM Customers;

-- Select data from the CustomerView
SELECT * FROM CustomerView;

-- Create a view to display orders from customers in London
CREATE VIEW OrdersInLondonView
AS
-- Select order details along with customer information for customers in London
SELECT o.OrderID, o.OrderDate, c.CustomerID, c.CompanyName, c.ContactName
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE c.City = 'London';

-- Select data from the OrdersInLondonView
SELECT * FROM OrdersInLondonView;
```

```

-- Create SQL VIEW in SQL Server:
-- This view selects specific columns from the Customers table
CREATE VIEW CustomerView
AS
SELECT CustomerID, CompanyName, ContactName, ContactTitle
FROM Customers;

-- Select data from the CustomerView
SELECT * FROM CustomerView;

-- Create a view to display orders from customers in London
CREATE VIEW OrdersInLondonView
AS
-- Select order details along with customer information for customers in London
SELECT o.OrderID, o.OrderDate, c.CustomerID, c.CompanyName, c.ContactName
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE c.City = 'London';

```

Figure 1: Screenshot 1 for Creating SQL View

	OrderID	OrderDate	CustomerID	CompanyName	ContactName
1	10289	1996-08-26 00:00:00.000	BSBEV	B's Beverages	Victoria Ashworth
2	10355	1996-11-15 00:00:00.000	AROUT	Around the Horn	Thomas Hardy
3	10359	1996-11-21 00:00:00.000	SEVES	Seven Seas Imports	Hari Kumar
4	10364	1996-11-26 00:00:00.000	EASTC	Eastern Connection	Ann Devon
5	10377	1996-12-09 00:00:00.000	SEVES	Seven Seas Imports	Hari Kumar
6	10383	1996-12-16 00:00:00.000	AROUT	Around the Horn	Thomas Hardy
7	10388	1996-12-19 00:00:00.000	SEVES	Seven Seas Imports	Hari Kumar
8	10400	1997-01-01 00:00:00.000	EASTC	Eastern Connection	Ann Devon
9	10435	1997-02-04 00:00:00.000	CONSH	Consolidated Holdings	Elizabeth Brown
10	10453	1997-02-21 00:00:00.000	AROUT	Around the Horn	Thomas Hardy
11	10462	1997-03-03 00:00:00.000	CONSH	Consolidated Holdings	Elizabeth Brown
12	10471	1997-03-11 00:00:00.000	BSBEV	B's Beverages	Victoria Ashworth

Figure 2: Screenshot 2 for Creating SQL View

## Task 2: Renaming the View

### SQL Code

```
EXEC sp_rename 'CustomerView', 'NewCustomerView';
```

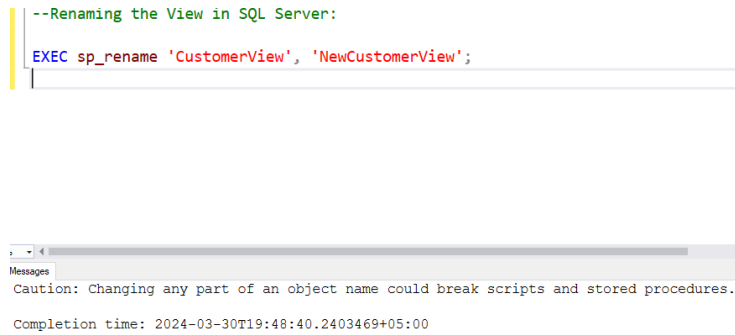
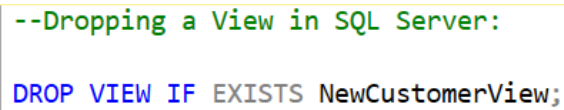


Figure 3: Screenshot for Renaming the View

## Task 3: Dropping a View

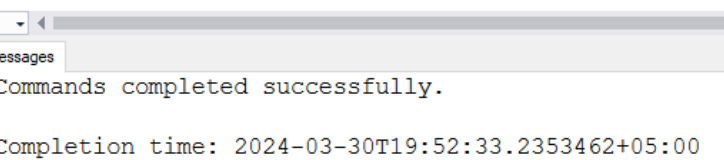
### SQL Code

```
DROP VIEW IF EXISTS NewCustomerView;
```

A screenshot of a text editor window with a yellow border. The text inside is: 

```
--Dropping a View in SQL Server:  
DROP VIEW IF EXISTS NewCustomerView;
```

```
--Dropping a View in SQL Server:  
DROP VIEW IF EXISTS NewCustomerView;
```

A screenshot of a command window. It has a title bar with a dropdown arrow and a left arrow. Below the title bar is a tab labeled 'essages'. The main area contains the text: 

```
Commands completed successfully.  
  
Completion time: 2024-03-30T19:52:33.2353462+05:00
```

```
Commands completed successfully.  
  
Completion time: 2024-03-30T19:52:33.2353462+05:00
```

Figure 4: Screenshot for Dropping a View

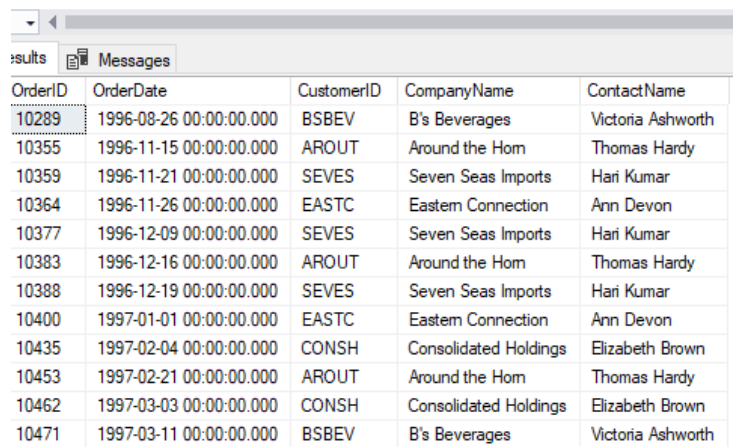
## Task 4: Retrieving Data From View

### SQL Code

```
SELECT * FROM OrdersInLondonView;
```

```
--Retrieve Data From View in SQL Server:
```

```
SELECT * FROM OrdersInLondonView;
```



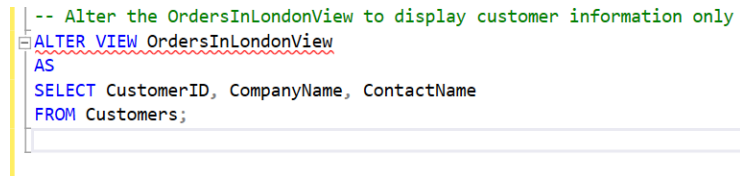
OrderID	OrderDate	CustomerID	CompanyName	ContactName
10289	1996-08-26 00:00:00.000	BSBEV	B's Beverages	Victoria Ashworth
10355	1996-11-15 00:00:00.000	AROUT	Around the Horn	Thomas Hardy
10359	1996-11-21 00:00:00.000	SEVES	Seven Seas Imports	Hari Kumar
10364	1996-11-26 00:00:00.000	EASTC	Eastern Connection	Ann Devon
10377	1996-12-09 00:00:00.000	SEVES	Seven Seas Imports	Hari Kumar
10383	1996-12-16 00:00:00.000	AROUT	Around the Horn	Thomas Hardy
10388	1996-12-19 00:00:00.000	SEVES	Seven Seas Imports	Hari Kumar
10400	1997-01-01 00:00:00.000	EASTC	Eastern Connection	Ann Devon
10435	1997-02-04 00:00:00.000	CONSH	Consolidated Holdings	Elizabeth Brown
10453	1997-02-21 00:00:00.000	AROUT	Around the Horn	Thomas Hardy
10462	1997-03-03 00:00:00.000	CONSH	Consolidated Holdings	Elizabeth Brown
10471	1997-03-11 00:00:00.000	BSBEV	B's Beverages	Victoria Ashworth

Figure 5: Screenshot for Retrieving Data From View

## Task 5: Altering View

### SQL Code

```
ALTER VIEW OrdersInLondonView  
AS  
SELECT CustomerID, CompanyName, ContactName  
FROM Customers;
```

A screenshot of a SQL development environment showing the code from the previous block. The code is color-coded: green for comments, blue for SQL keywords, and black for identifiers and literals. The text is: 

```
-- Alter the OrdersInLondonView to display customer information only  
ALTER VIEW OrdersInLondonView  
AS  
SELECT CustomerID, CompanyName, ContactName  
FROM Customers;
```

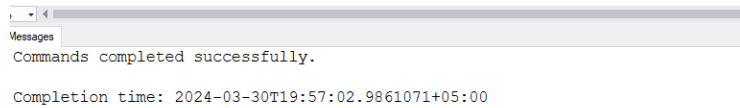
A screenshot of the SQL execution results pane. It shows a message box titled 'Messages' with the text 'Commands completed successfully.' and a completion time: 'Completion time: 2024-03-30T19:57:02.9861071+05:00'.

Figure 6: Screenshot for Altering View

## Task 6: Refreshing a View

### SQL Code

```
-- SQL code for refreshing view goes here
```

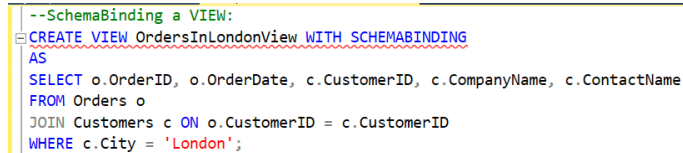
Figure 7: Screenshot for Refreshing a View



## Task 7: SchemaBinding a View

### SQL Code

```
CREATE VIEW OrdersInLondonView WITH SCHEMABINDING
AS
SELECT o.OrderID, o.OrderDate, o.CustomerID, c.CompanyName, c.ContactName
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE c.City = 'London';
```

A screenshot of a SQL editor window showing the code for creating a schema-bound view. The code is as follows:

```
--SchemaBinding a VIEW:
CREATE VIEW OrdersInLondonView WITH SCHEMABINDING
AS
SELECT o.OrderID, o.OrderDate, c.CustomerID, c.CompanyName, c.ContactName
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE c.City = 'London';
```

The text is color-coded: comments are green, keywords like CREATE, VIEW, WITH, SCHEMABINDING, AS, SELECT, FROM, JOIN, ON, and WHERE are blue, and string literals like 'London' are red.

Figure 8: Screenshot for SchemaBinding a View

## Task 8: Encrypting a View

### SQL Code

```
CREATE VIEW OrdersInLondonView  
WITH ENCRYPTION  
AS  
SELECT OrderID, OrderDate, CustomerID, EmployeeID, ShipCity  
FROM Orders;
```

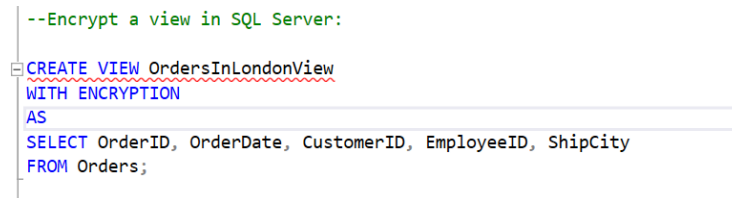


Figure 9: Screenshot for Encrypting a View