

MODULE 9: IMPLEMENTING VIEWS

OVERVIEW



Introduction to Views



Advantages of Views



Defining Views



Modifying Data Through Views



Optimizing Performance by Using Views

INTRODUCTION TO VIEWS

Employees

<i>EmployeeID</i>	<i>LastName</i>	<i>Firstname</i>	<i>Title</i>
1	Davolio	Nancy	~~~
2	Fuller	Andrew	~~~
3	Leverling	Janet	~~~

```
USE Northwind
GO
CREATE VIEW dbo.EmployeeView
AS
SELECT LastName, Firstname
FROM Employees
```

EmployeeView

<i>Lastname</i>	<i>Firstname</i>
Davolio	Nancy
Fuller	Andrew
Leverling	Janet

User's View



ADVANTAGES OF VIEWS

- Focus the Data for Users
 - Focus on important or appropriate data only
 - Limit access to sensitive data
- Mask Database Complexity
 - Hide complex database design
 - Simplify complex queries, including distributed queries to heterogeneous data
- Simplify Management of User Permissions
- Improve Performance
- Organize Data for Export to Other Applications

◆ DEFINING VIEWS



Creating Views



Example: View of
Joined Tables



Altering and
Dropping Views



Avoiding Broken
Ownership Chains



Locating View
Definition
Information



Hiding View
Definitions

CREATING VIEWS

■ Creating a View

```
CREATE VIEW dbo.OrderSubtotalsView (OrderID, Subtotal)
AS
SELECT OD.OrderID,
       SUM(CONVERT(money, (OD.UnitPrice*Quantity*(1-Discout)/100))*100)
FROM [Order Details] OD
GROUP BY OD.OrderID
GO
```

■ Restrictions on View Definitions

- Cannot include ORDER BY clause
- Cannot include INTO keyword

EXAMPLE:VIEW OF JOINED TABLES

Orders

OrderID	CustomerID	RequiredDate	ShippedDate
10663	BONAP	1997-09-24	1997-10-03
10827	BONAP	1998-01-26	1998-02-06
10427	PICCO	1997-02-24	1997-03-03
10451	QUICK	1997-03-05	1997-03-12
10515	QUICK	1997-05-07	1997-05-23

Customers

CustomerID	CompanyName	ContactName
BONAP	Bon app'	Laurence Lebihan
PICCO	Piccolo und mehr	Georg Pippis
QUICK	QUICK-Stop	Horst Kloss

```
USE Northwind
GO
CREATE VIEW dbo.ShipStatusView
AS
SELECT OrderID, RequiredDate, ShippedDate,
       ContactName
FROM Customers c INNER JOIN Orders o
  ON c.CustomerID = o.CustomerID
WHERE RequiredDate < ShippedDate
```

ShipStatusView

OrderID	ShippedDate	ContactName
10264	1996-08-23	Laurence Lebihan
10271	1996-08-30	Georg Pippis
10280	1996-09-12	Horst Kloss

ALTERING AND DROPPING VIEWS

- Altering Views

```
USE Northwind
GO
ALTER VIEW dbo.EmployeeView
AS
SELECT LastName, FirstName, Extension
FROM Employees
```

- Retains assigned permissions
- Causes new SELECT statement and options to replace existing definition

- Dropping Views

```
DROP VIEW dbo.ShipStatusView
```


AVOIDING BROKEN OWNERSHIP CHAINS

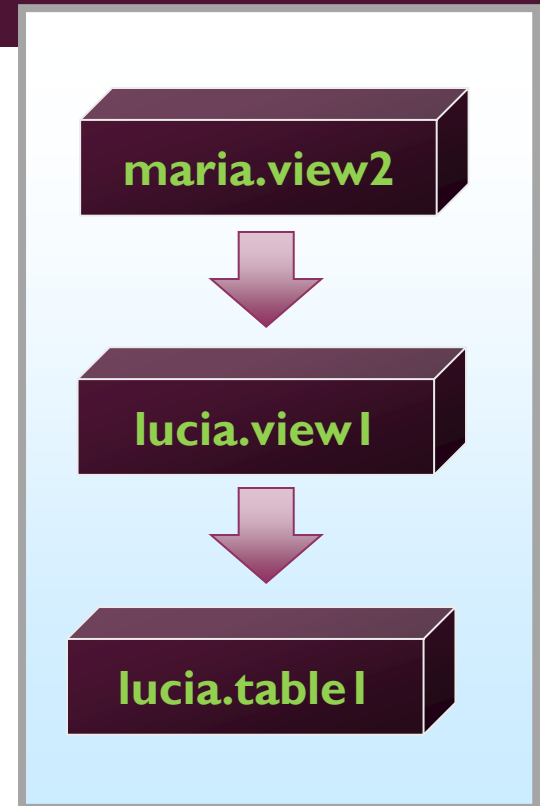
- Dependent Objects with Different Owners
- Example:

Maria executes:

```
GRANT SELECT ON view2 TO pierre
```

Pierre executes:

```
SELECT * FROM maria.view2
```



LOCATING VIEW DEFINITION INFORMATION

1

Locating View Definitions

- Not available if view was created using **WITH ENCRYPTION** option

2

Locating View Dependencies

- Lists objects upon which view depends
- Lists objects that depend on a view

HIDING VIEW DEFINITIONS

```
USE Northwind
GO
CREATE VIEW dbo.[Order Subtotals]
    WITH ENCRYPTION
AS
SELECT OrderID,
    Sum(CONVERT(money, (UnitPrice*Quantity*(1-Discount)/100))*100)
    AS Subtotal
FROM [Order Details]
GROUP BY OrderID
GO
```



MODIFYING DATA THROUGH VIEWS

- Cannot Affect More Than One Underlying Table
- Cannot Be Made to Certain Columns
- Can Cause Errors If They Affect Columns That Are Not Referenced in the View
- Are Verified If the **WITH CHECK OPTION** Has Been Specified



Performance Considerations



Using Indexed Views



Using Views to Partition Data

◆ OPTIMIZING PERFORMANCE BY USING VIEWS

PERFORMANCE CONSIDERATIONS

Customers

Orders

Order Details

1	1	~	~	~	~	~	~
2	2	~	~	~	~	~	~
3	3	~	~	~	~	~	~
4	4	~	~	~	~	~	~
5	5	~	~	~	~	~	~
6	6	~	~	~	~	~	~

USE Northwind

GO

CREATE VIEW dbo.TopSalesView

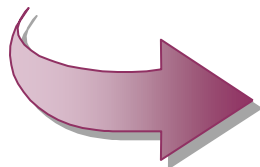
AS

SELECT *

FROM dbo.TotalPurchaseView

WHERE Subtotal > 50000

GO



TotalPurchaseView				
1	~	~	~	~
2	~	~	~	~
3	~	~	~	~
4	~	~	~	~
5	~	~	~	~
6	~	~	~	~



TopSalesView				
	~	~	~	
	~	~	~	
	~	~	~	

SELECT *

FROM dbo.TopSales

WHERE CompanyName = 'Ernst Handel'



USING INDEXED VIEWS

- Indexed Views Store the Result Sets in the Database
- Creating an Indexed View
- Guidelines for Creating Indexed Views
 - Use when:
 - Performance gains outweigh maintenance costs
 - Underlying data is infrequently updated
 - Queries perform many joins and aggregations
- Restrictions on Creating Indexed Views

USING VIEWS TO PARTITION DATA



YOU CAN USE VIEWS TO
PARTITION DATA ACROSS
MULTIPLE SERVERS OR
INSTANCES OF SQL SERVER



HOW SQL SERVER USES VIEWS
TO PARTITION DATA



HOW PARTITIONED VIEWS
IMPROVE PERFORMANCE

Recommended Practices



Use a Standard Naming Convention



dbo Should Own All Views



Verify Object Dependencies Before You Drop Objects

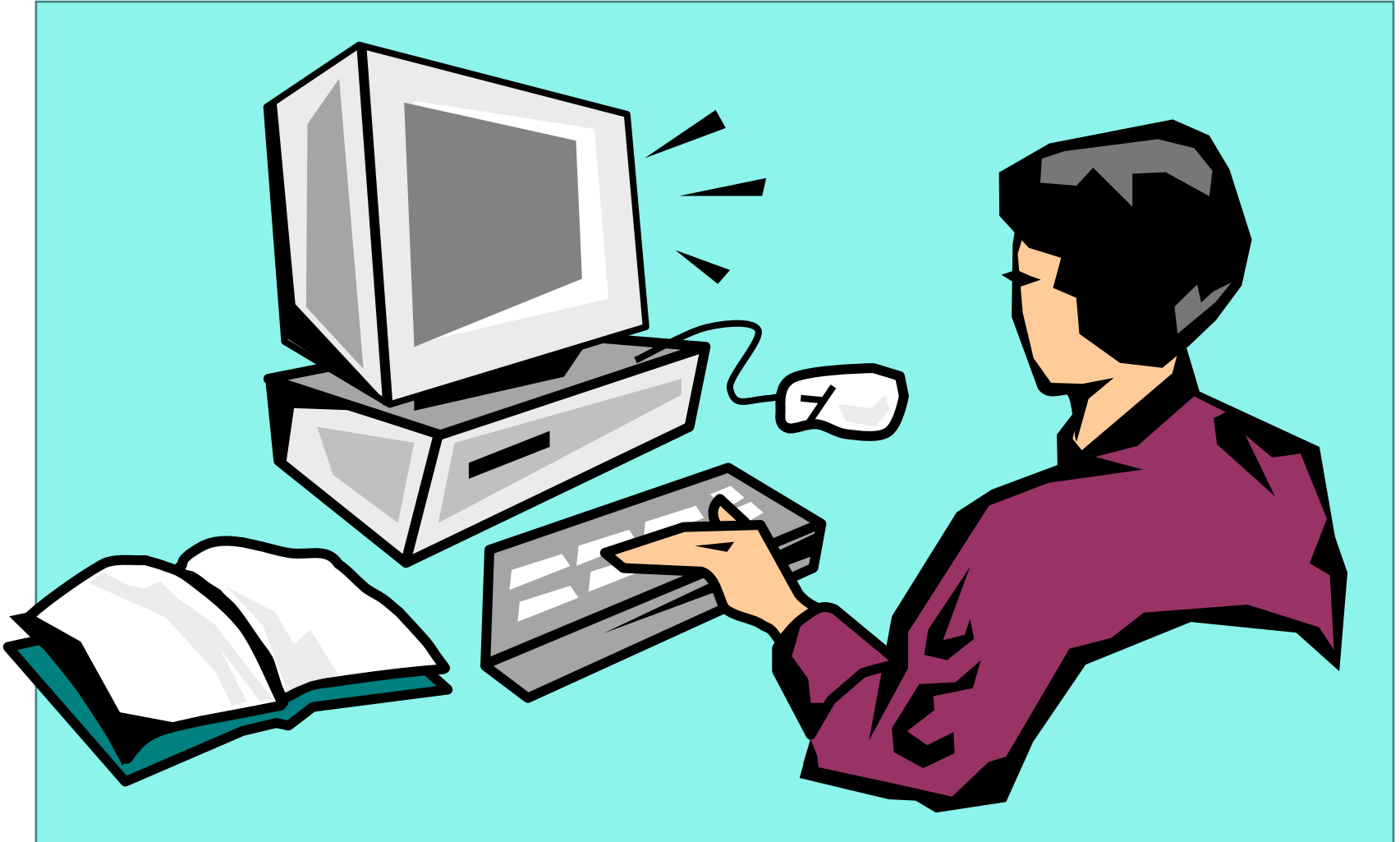


Never Delete Entries in the syscomments Table



Carefully Evaluate Creating Views Based on Views

Lab: Implementing Views



REVIEW



INTRODUCTION
TO VIEWS



ADVANTAGES OF
VIEWS



DEFINING VIEWS



MODIFYING
DATA THROUGH
VIEWS



OPTIMIZING
PERFORMANCE
BY USING VIEWS

MODULE REFERENCES

- Create View Microsoft SQL Server - [https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?f1url=%3FappId%3DDevI5IDEFI%26I%3DEN-US%26k%3Dk\(create_view_TSQL\);k\(sqlI3.swb.tsqldata.results.f1\);k\(sqlI3.swb.tsqldata.query.f1\);k\(MiscellaneousFilesProject\);k\(DevLang-TSQL\)%26rd%3Dtrue&view=sql-server-verI5](https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?f1url=%3FappId%3DDevI5IDEFI%26I%3DEN-US%26k%3Dk(create_view_TSQL);k(sqlI3.swb.tsqldata.results.f1);k(sqlI3.swb.tsqldata.query.f1);k(MiscellaneousFilesProject);k(DevLang-TSQL)%26rd%3Dtrue&view=sql-server-verI5)
- Create View Oracle Database - https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8004.htm
- Create View MySQL Database - <https://dev.mysql.com/doc/refman/8.0/en/create-view.html>