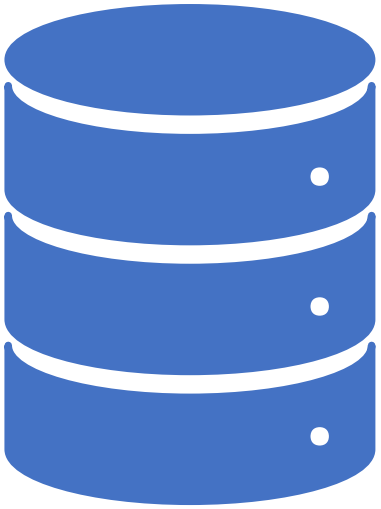
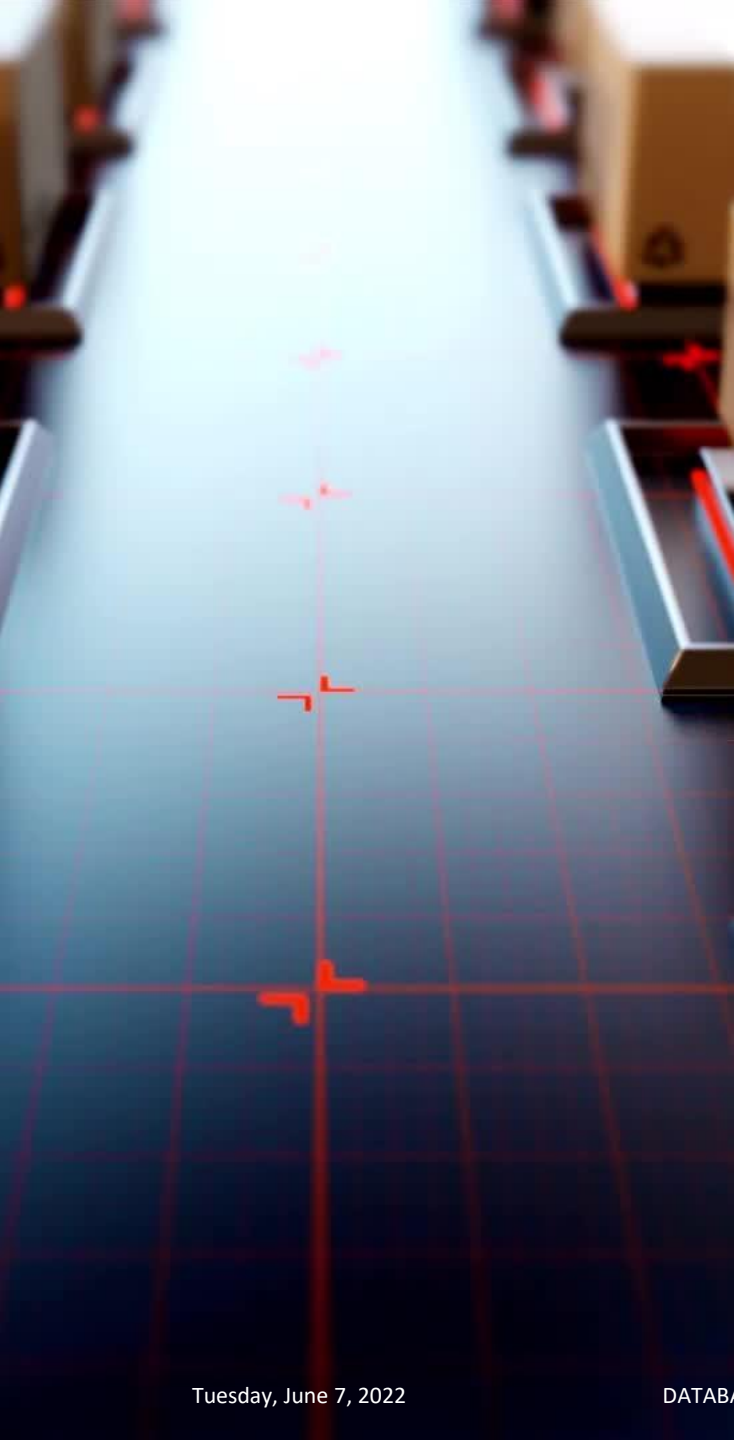
Module 10: Implementing Stored Procedures



Database Design & Business Application Development



Overview

Introduction to Stored Procedures

Creating, Executing, Modifying, and Dropping Stored Procedures

Using Parameters in Stored Procedures

Executing Extended Stored Procedures

Handling Error Messages

Introduction to Stored Procedures

Defining Stored Procedures

Initial Processing of Stored Procedures

Subsequent Processing of Stored Procedures

Advantages of Stored Procedures

Defining Stored Procedures

Named Collections of Transact-SQL Statements

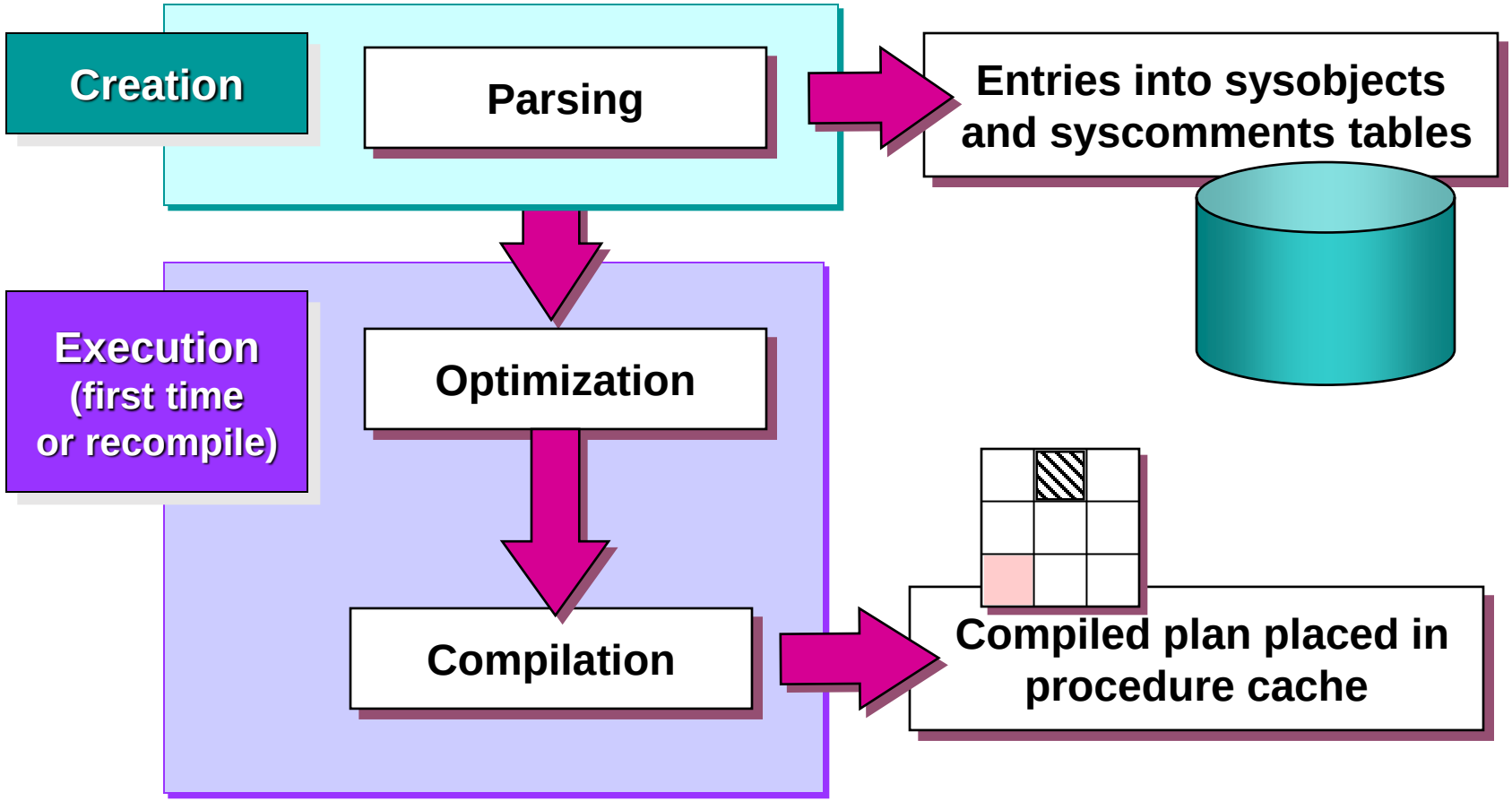
Encapsulate Repetitive Tasks

Five Types (System, Local, Temporary, Remote, and Extended)

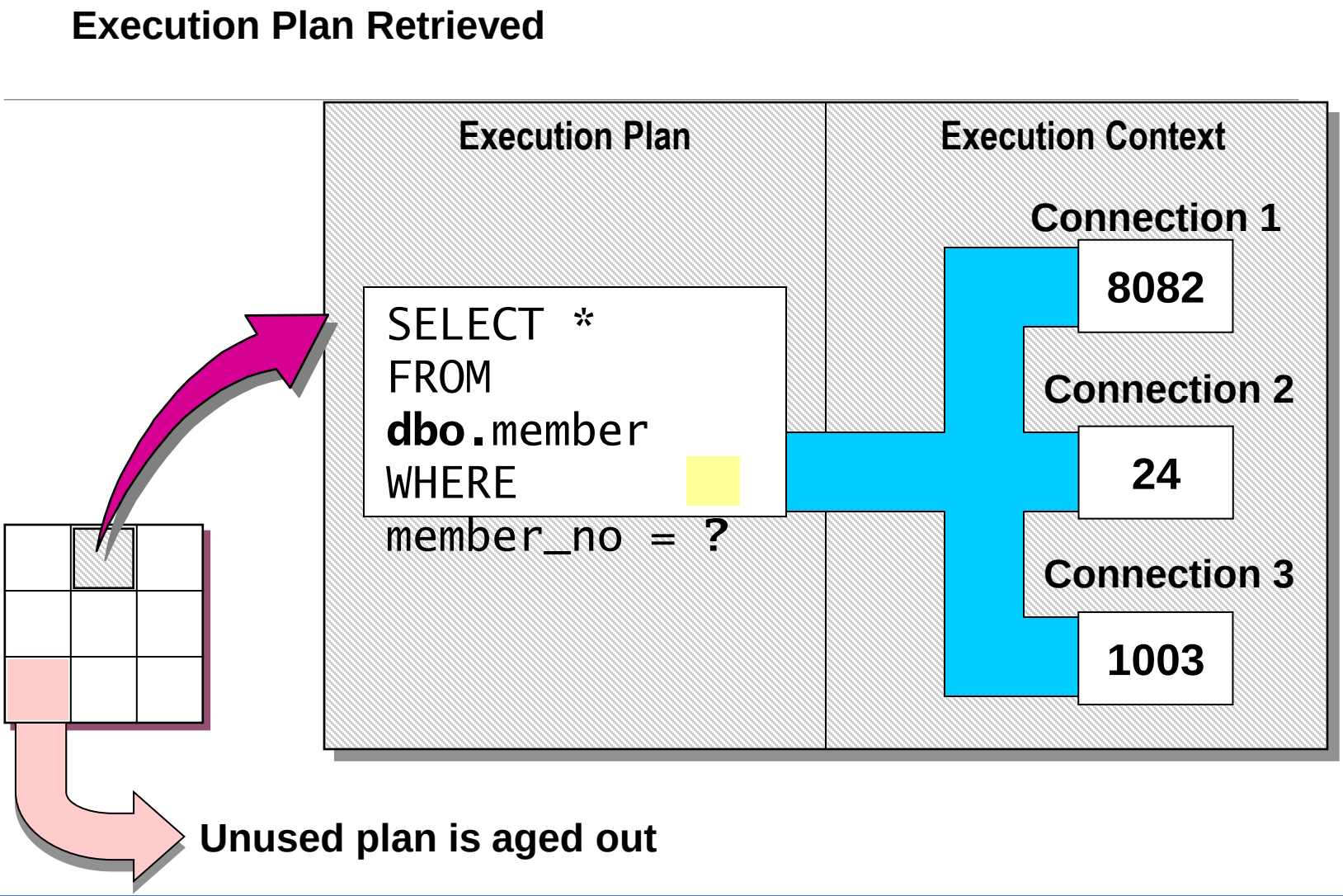
Accept Input Parameters and Return Values

Return Status Value to Indicate Success or Failure

Initial Processing of Stored Procedures



Subsequent Processing of Stored Procedures



Advantages of Stored Procedures

SHARE APPLICATION LOGIC

SHIELD DATABASE SCHEMA DETAILS

PROVIDE SECURITY MECHANISMS

IMPROVE PERFORMANCE

REDUCE NETWORK TRAFFIC

◆ Creating, Executing, Modifying, and Dropping Stored Procedures

Creating Stored Procedures

Guidelines for Creating Stored Procedures

Executing Stored Procedures

Altering and Dropping Stored Procedures

Creating Stored Procedures

Create in Current Database Using the CREATE PROCEDURE Statement

```
USE Northwind
GO
CREATE PROC dbo.OverdueOrders
AS
  SELECT *
   FROM dbo.Orders
   WHERE RequiredDate < GETDATE() AND ShippedDate IS Null
GO
```

Guidelines for Creating Stored Procedures

- dbo User Should Own All Stored Procedures
- One Stored Procedure for One Task
- Create, Test, and Troubleshoot
- Avoid sp_ Prefix in Stored Procedure Names
- Use Same Connection Settings for All Stored Procedures
- Minimize Use of Temporary Stored Procedures
- Never Delete Entries Directly From Syscomments

Executing Stored Procedures

Executing a Stored Procedure by Itself

```
EXEC OverdueOrders
```

Executing a Stored Procedure Within an INSERT Statement

```
INSERT INTO Customers
EXEC EmployeeCustomer
```

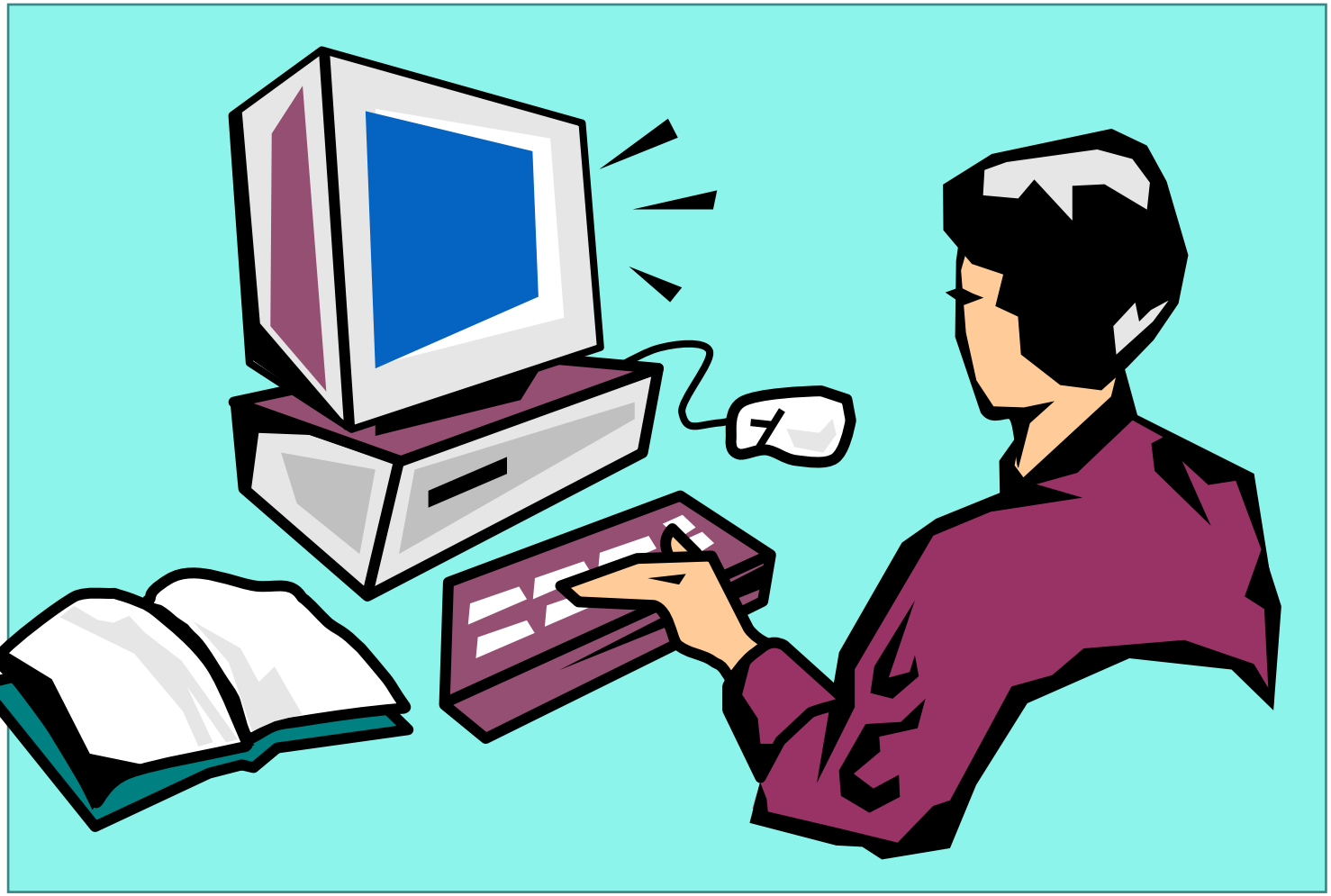
Altering and Dropping Stored Procedures

Altering Stored Procedures

- Include any options in ALTER PROCEDURE
- Does not affect nested stored procedures

```
USE Northwind
GO
ALTER PROC dbo.OverdueOrders
AS
SELECT CONVERT(char(8), RequiredDate, 1) RequiredDate,
  CONVERT(char(8), OrderDate, 1) OrderDate,
  OrderID, CustomerID, EmployeeID
  FROM Orders
WHERE RequiredDate < GETDATE() AND ShippedDate IS Null
ORDER BY RequiredDate
GO
```

Lab: Creating Stored Procedures



Using Parameters in Stored Procedures

- Using Input Parameters
- Executing Stored Procedures Using Input Parameters
- Returning Values Using Output Parameters
- Explicitly Recompiling Stored Procedures

Using Input Parameters

Validate All Incoming Parameter Values First

Provide Appropriate Default Values and Include Null Checks

```
CREATE PROCEDURE dbo.[Year to Year Sales]
  @BeginningDate DateTime, @EndingDate DateTime
AS
IF @BeginningDate IS NULL OR @EndingDate IS NULL
BEGIN
   RAISERROR('NULL values are not allowed', 14, 1)
   RETURN
END
SELECT O.ShippedDate,
       O.OrderID,
       OS.Subtotal,
       DATENAME(yy,ShippedDate) AS Year
FROM ORDERS O INNER JOIN [Order Subtotals] OS
   ON O.OrderID = OS.OrderID
WHERE O.ShippedDate BETWEEN @BeginningDate AND @EndingDate
GO
```

Executing Stored Procedures Using Input Parameters

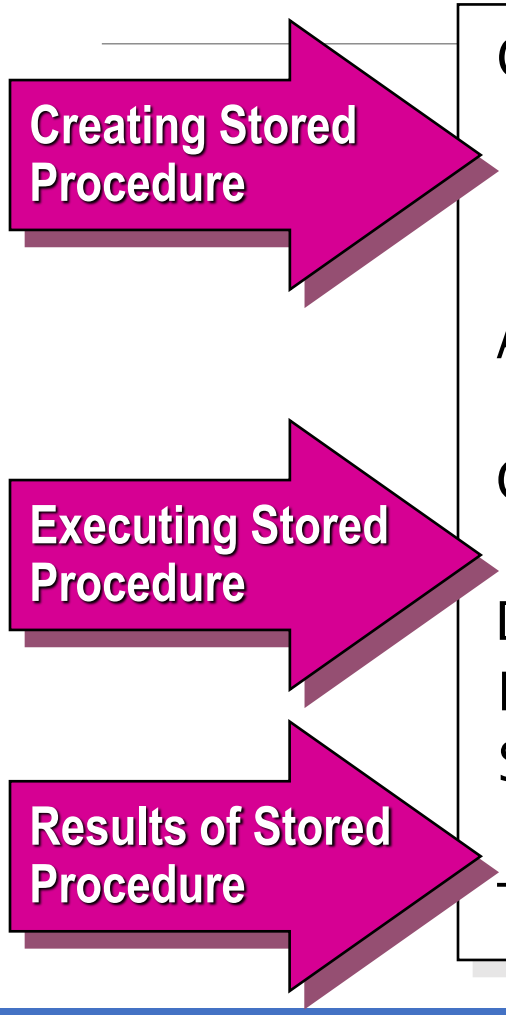
Passing Values by Parameter Name

```
EXEC AddCustomer
    @CustomerID = 'ALFKI',
    @ContactName = 'Maria Anders',
    @CompanyName = 'Alfreds Futterkiste',
    @ContactTitle = 'Sales Representative',
    @Address = 'Obere Str. 57',
    @City = 'Berlin',
    @PostalCode = '12209',
    @Country = 'Germany',
    @Phone = '030-0074321'
```

Passing Values by Position

```
EXEC AddCustomer 'ALFKI2', 'Alfreds Futterkiste',
'Maria Anders', 'Sales Representative', 'Obere Str.
57', 'Berlin', NULL, '12209', 'Germany', '030-
0074321'
```

Returning Values Using Output Parameters




```
CREATE PROCEDURE dbo.MathTutor
    @m1 smallint,
    @m2 smallint,
    @result smallint OUTPUT
AS
    SET @result = @m1* @m2
GO

DECLARE @answer smallint
EXECUTE MathTutor 5,6, @answer OUTPUT
SELECT 'The result is: ', @answer

The result is:  30
```

Explicitly Recompiling Stored Procedures

Recompile When

- Stored procedure returns widely varying result sets
- A new index is added to an underlying table
- The parameter value is atypical

Recompile by Using

- CREATE PROCEDURE [WITH RECOMPILE]
- EXECUTE [WITH RECOMPILE]
- **sp_recompile**

Executing Extended Stored Procedures

Are Programmed Using Open Data Services API

Can Include C and C++ Features

Can Contain Multiple Functions

Can Be Called from a Client or SQL Server

Can Be Added to the master Database Only

```
EXEC master..xp_cmdshell 'dir c:\'
```

Handling Error Messages

RETURN Statement Exits Query or Procedure Unconditionally

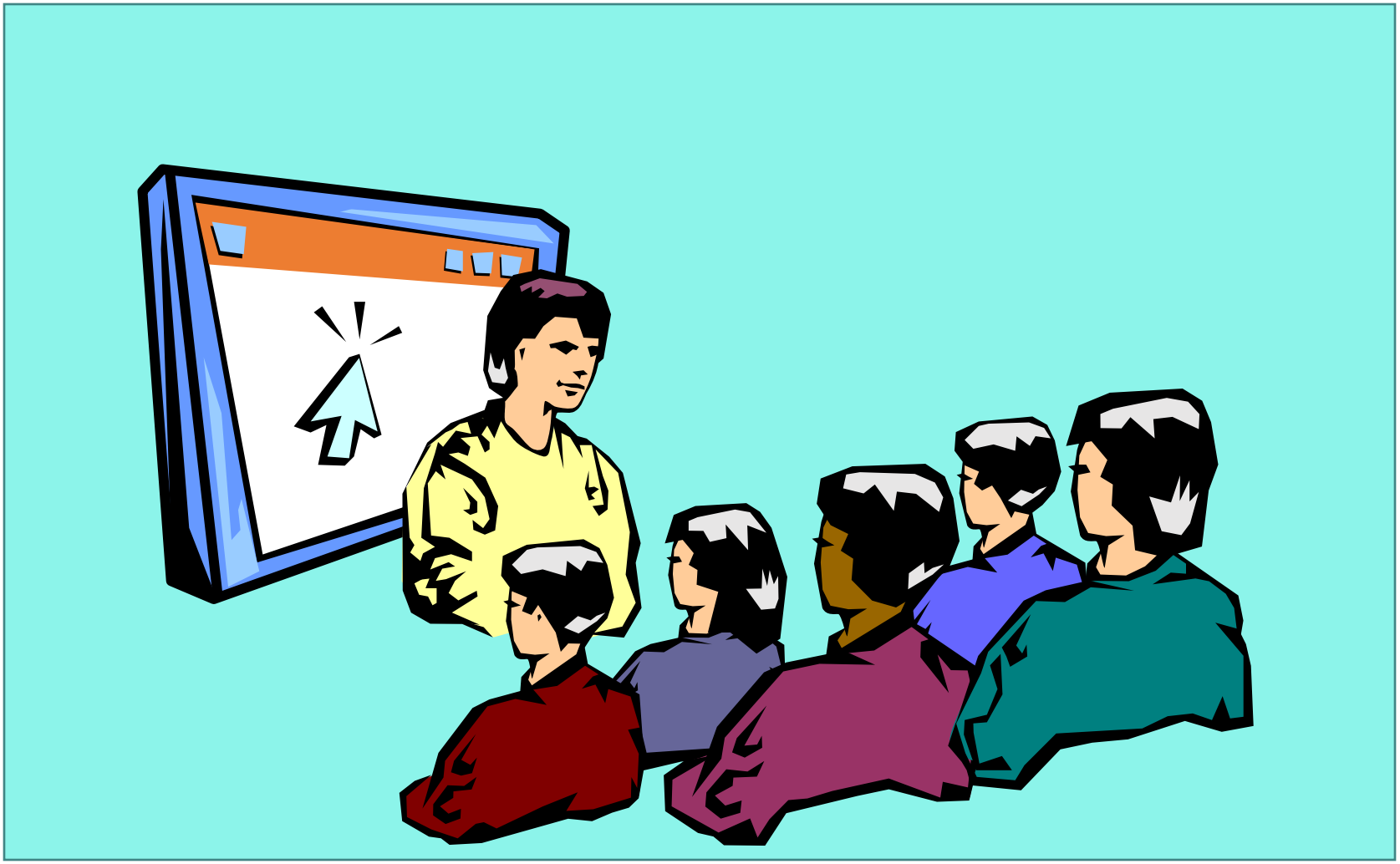
sp_addmessage Creates Custom Error Messages

@@error Contains Error Number for Last Executed Statement

RAISERROR Statement

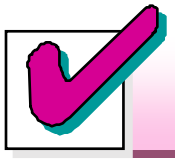
- Returns user-defined or system error message
- Sets system flag to record error

Demonstration: Handling Error Messages



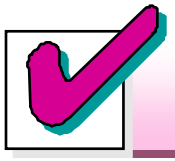
Recommended Practices

Verify Input Parameters

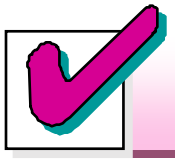


Design Each Stored Procedure to Accomplish a Single Task

Validate Data Before You Begin Transactions

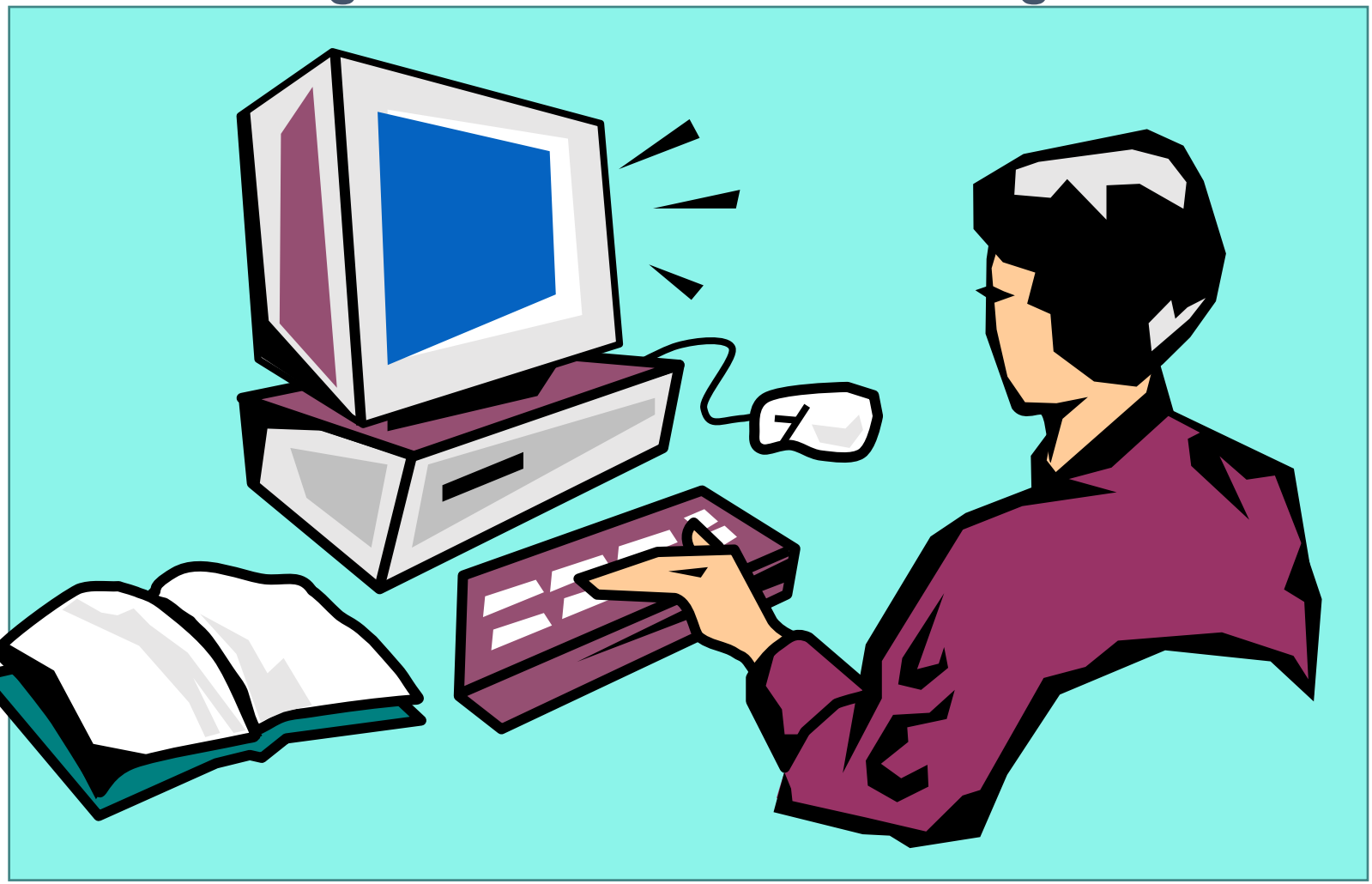


Use the Same Connection Settings for All Stored Procedures



Use WITH ENCRYPTION to Hide Text of Stored Procedures

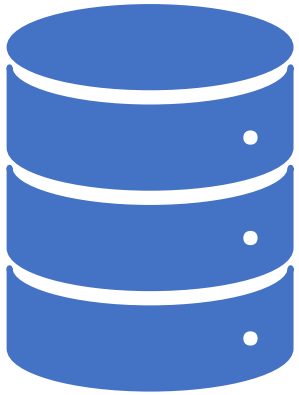
Lab: Creating Stored Procedures Using Parameters



Review

- Introduction to Stored Procedures
- Creating, Executing, Modifying, and Dropping Stored Procedures
- Using Parameters in Stored Procedures
- Executing Extended Stored Procedures
- Handling Error Messages

Module References



- Create Procedure (Transact-SQL) - https://docs.microsoft.com/en-us/sql/t-sql/statements/create-procedure-transact-sql?f1url=%3FappId%3DDev15IDEF1%26l%3DEN-US%26k%3Dk(create_procedure_TSQL);k(sql13.swb.tsqlresults.f1);k(sql13.swb.tsqlquery.f1);k(MiscellaneousFilesProject);k(DevLang-TSQL)%26rd%3Dtrue&view=sql-server-ver15
- Create Procedure Oracle Database - https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6009.htm
- Create Procedure MySQL Database - https://dev.mysql.com/doc/refman/5.7/en/create-procedure.html