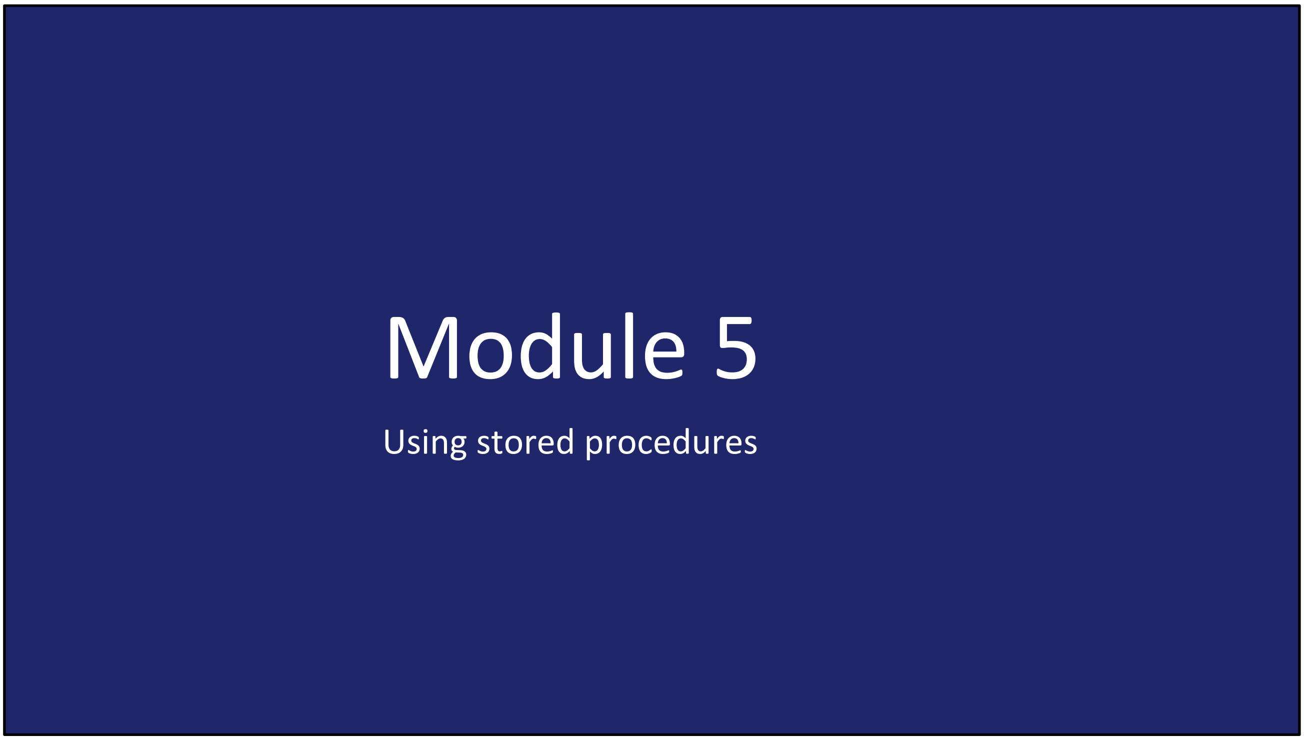
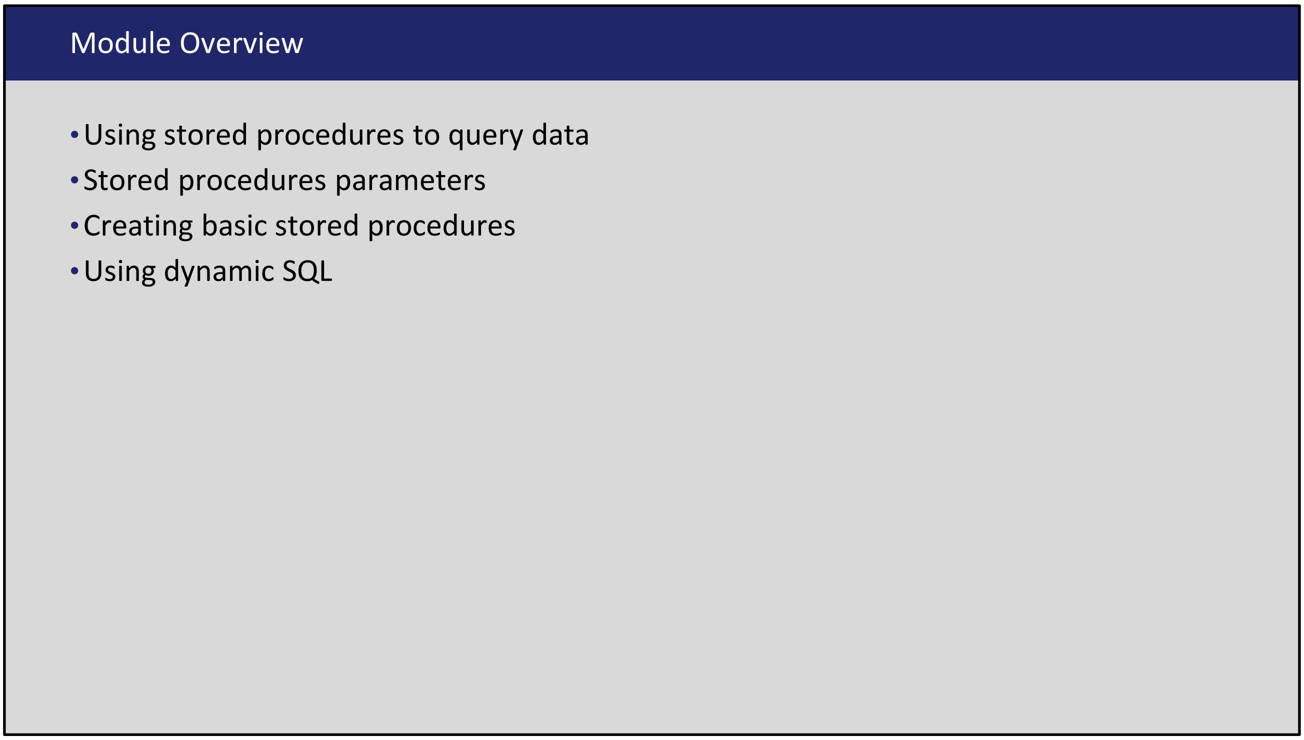


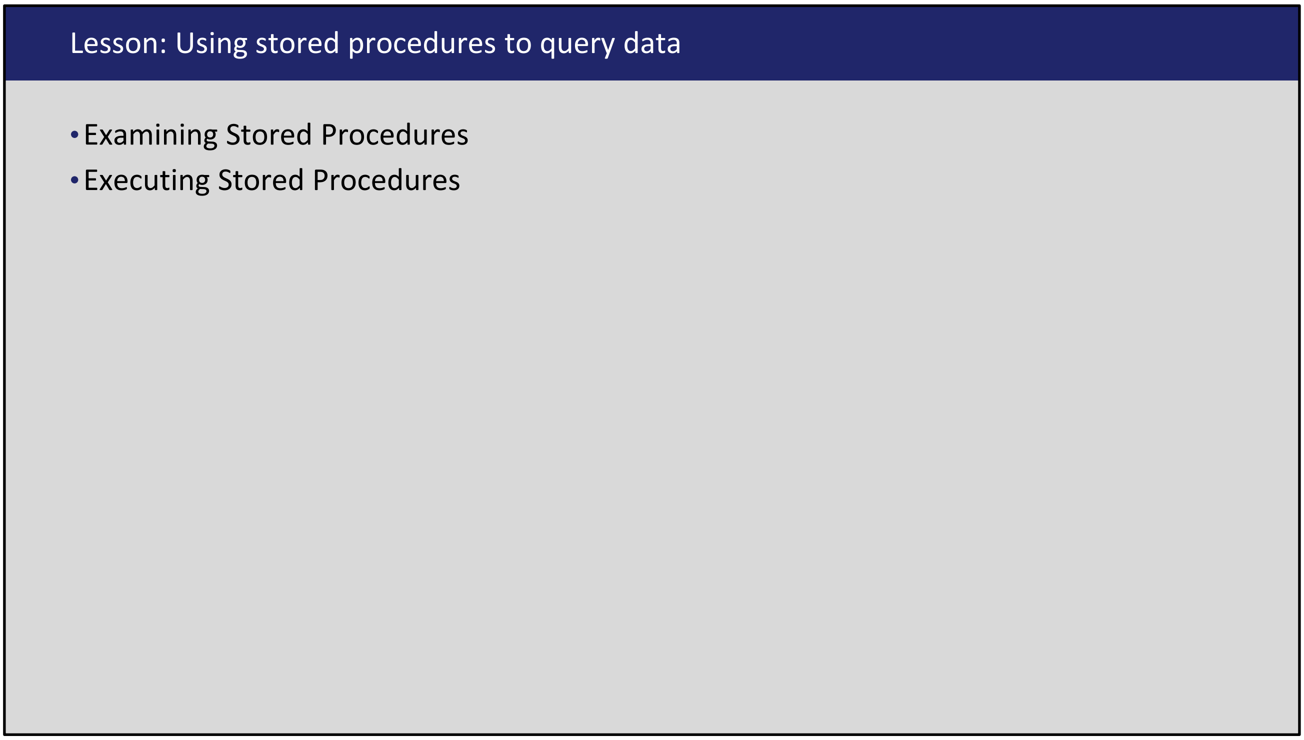
Module 5

Using stored procedures



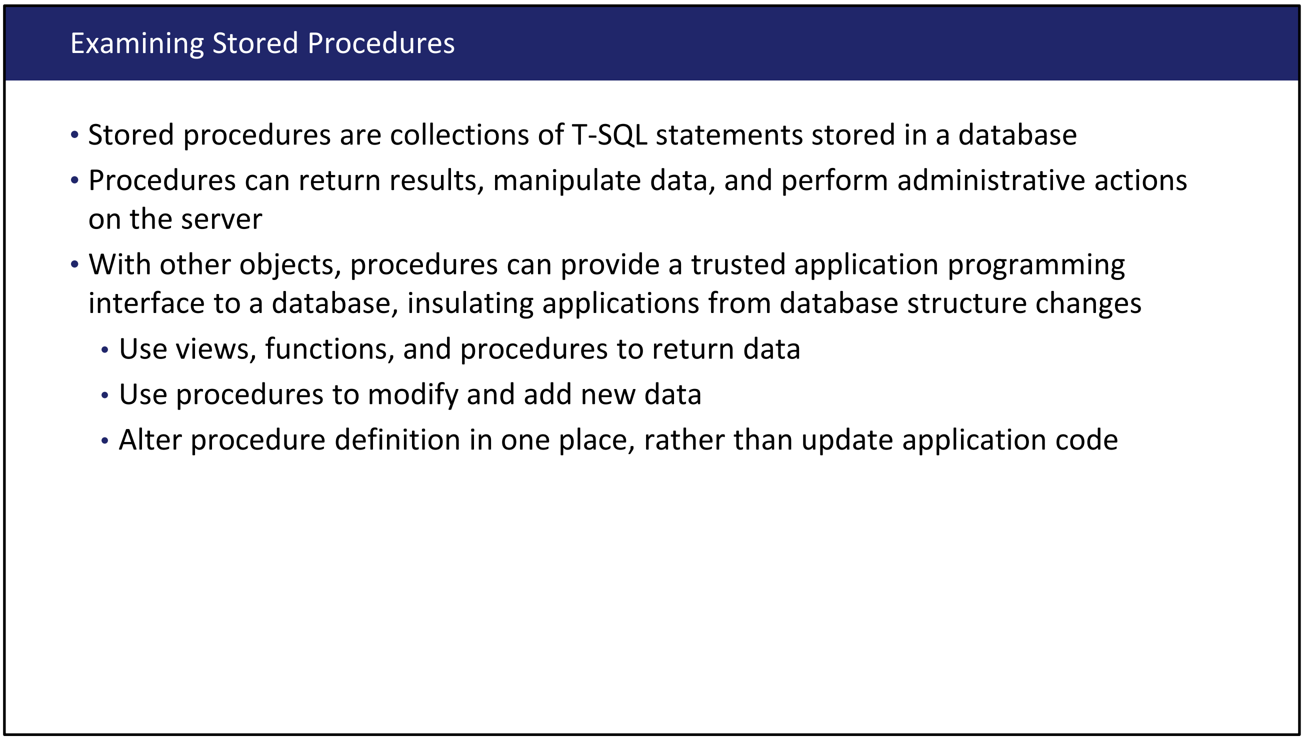
Module Overview

- Using stored procedures to query data
- Stored procedures parameters
- Creating basic stored procedures
- Using dynamic SQL



Lesson: Using stored procedures to query data

- Examining Stored Procedures
- Executing Stored Procedures



Examining Stored Procedures

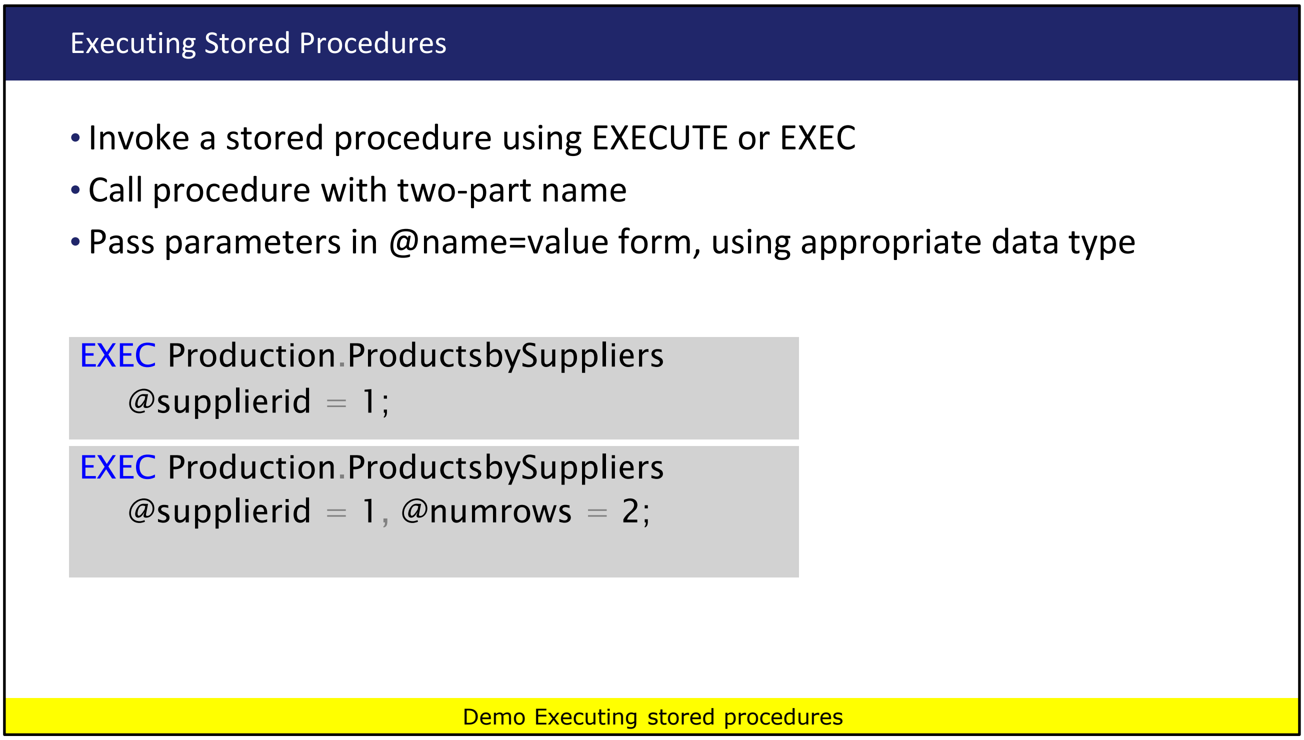
- Stored procedures are collections of T-SQL statements stored in a database
- Procedures can return results, manipulate data, and perform administrative actions on the server
- With other objects, procedures can provide a trusted application programming interface to a database, insulating applications from database structure changes
  - Use views, functions, and procedures to return data
  - Use procedures to modify and add new data
  - Alter procedure definition in one place, rather than update application code

SQL Stored Procedures for SQL Server
https://www.w3schools.com/sql/sql_stored_procedures.asp

SQL Server Stored Procedures
https://www.sqlservertutorial.net/sql-server-stored-procedures/

SQL Server stored procedures for beginners
https://www.sqlshack.com/sql-server-stored-procedures-for-beginners/

Create a stored procedure
https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/create-a-stored-procedure



Executing Stored Procedures

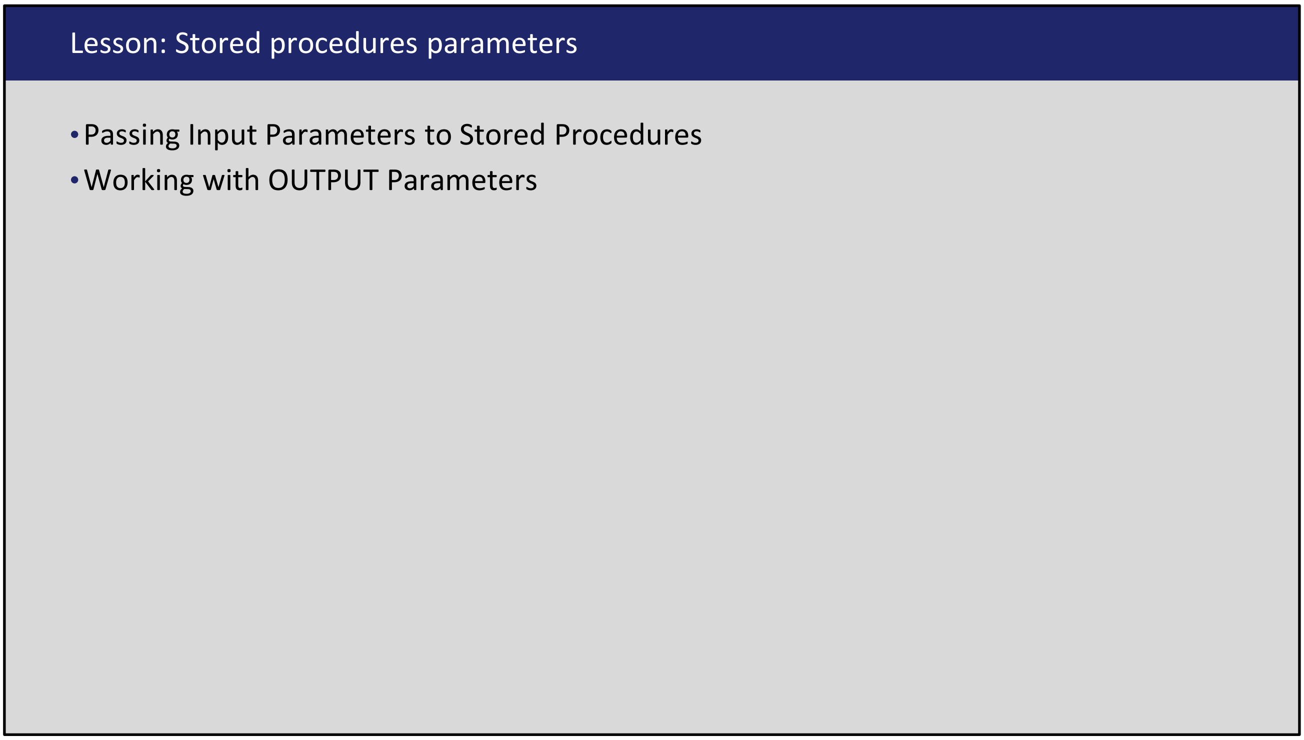
- Invoke a stored procedure using EXECUTE or EXEC
- Call procedure with two-part name
- Pass parameters in @name=value form, using appropriate data type

```
EXEC Production.ProductsbySuppliers
    @supplierid = 1;
```

```
EXEC Production.ProductsbySuppliers
    @supplierid = 1, @numrows = 2;
```

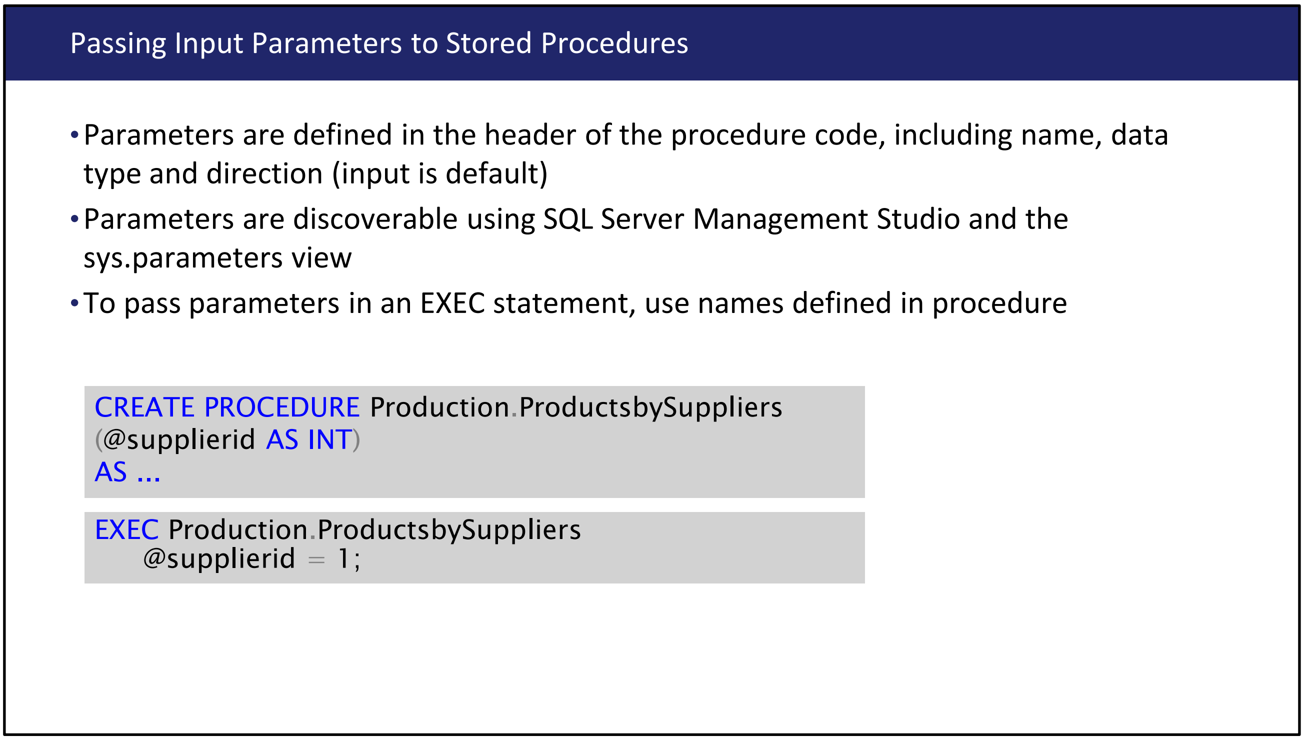
Demo Executing stored procedures

Execute a stored procedure
https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/execute-a-stored-procedure



Lesson: Stored procedures parameters

- Passing Input Parameters to Stored Procedures
- Working with OUTPUT Parameters



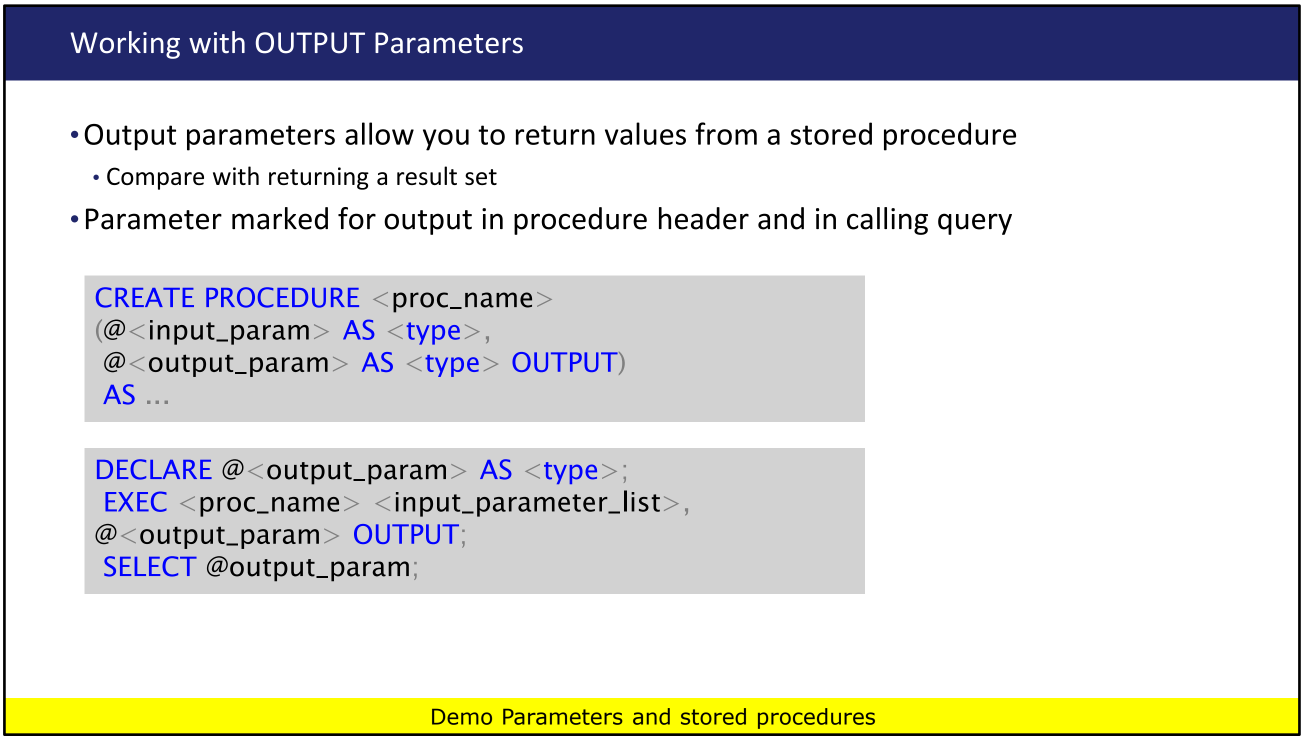
Passing Input Parameters to Stored Procedures

- Parameters are defined in the header of the procedure code, including name, data type and direction (input is default)
- Parameters are discoverable using SQL Server Management Studio and the sys.parameters view
- To pass parameters in an EXEC statement, use names defined in procedure

```
CREATE PROCEDURE Production.ProductsbySuppliers
(@supplierid AS INT)
AS ...
```

```
EXEC Production.ProductsbySuppliers
     @supplierid = 1;
```

Specify Parameters
https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/specify-parameters



Working with OUTPUT Parameters

- Output parameters allow you to return values from a stored procedure
  - Compare with returning a result set
- Parameter marked for output in procedure header and in calling query

```
CREATE PROCEDURE <proc_name>
(@<input_param> AS <type>,
 @<output_param> AS <type> OUTPUT)
 AS ...
```

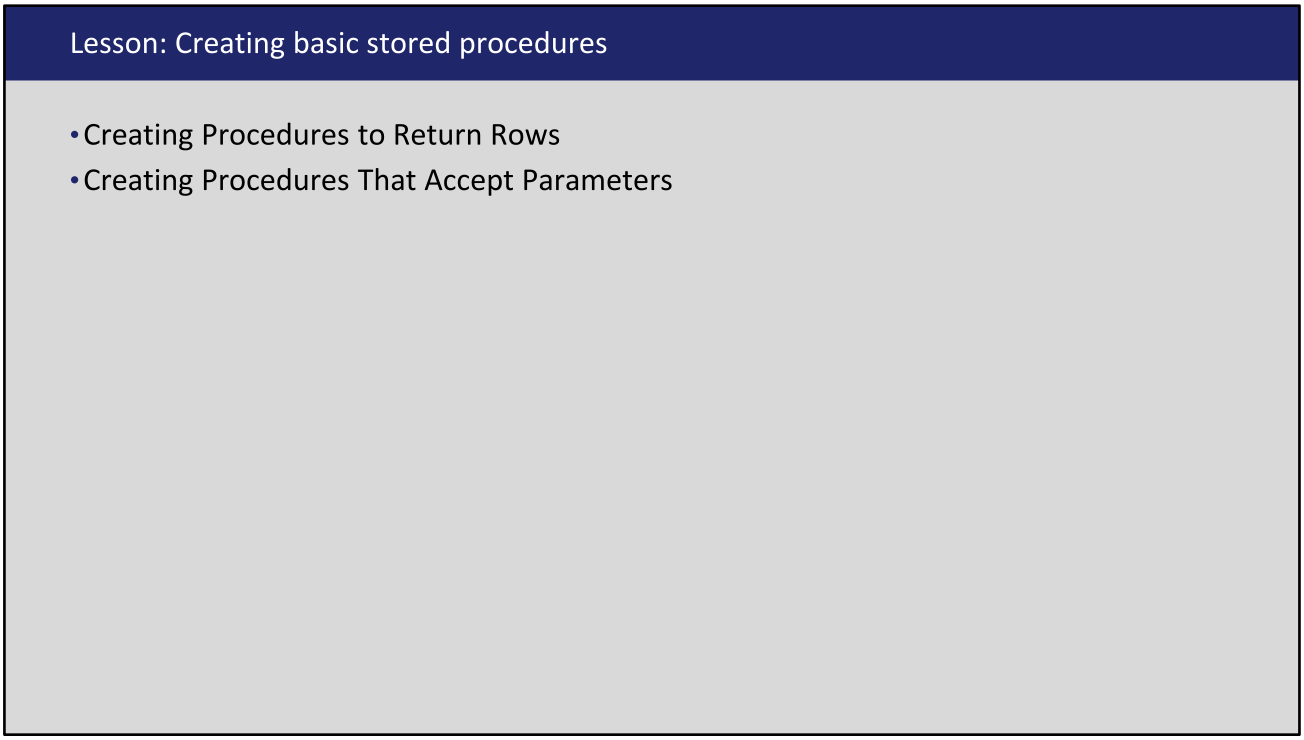
```
DECLARE @<output_param> AS <type>;
 EXEC <proc_name> <input_parameter_list>,
@<output_param> OUTPUT;
 SELECT @output_param;
```

Demo Parameters and stored procedures

Stored Procedure Output Parameters
https://www.sqlservertutorial.net/sql-server-stored-procedures/stored-procedure-output-parameters/

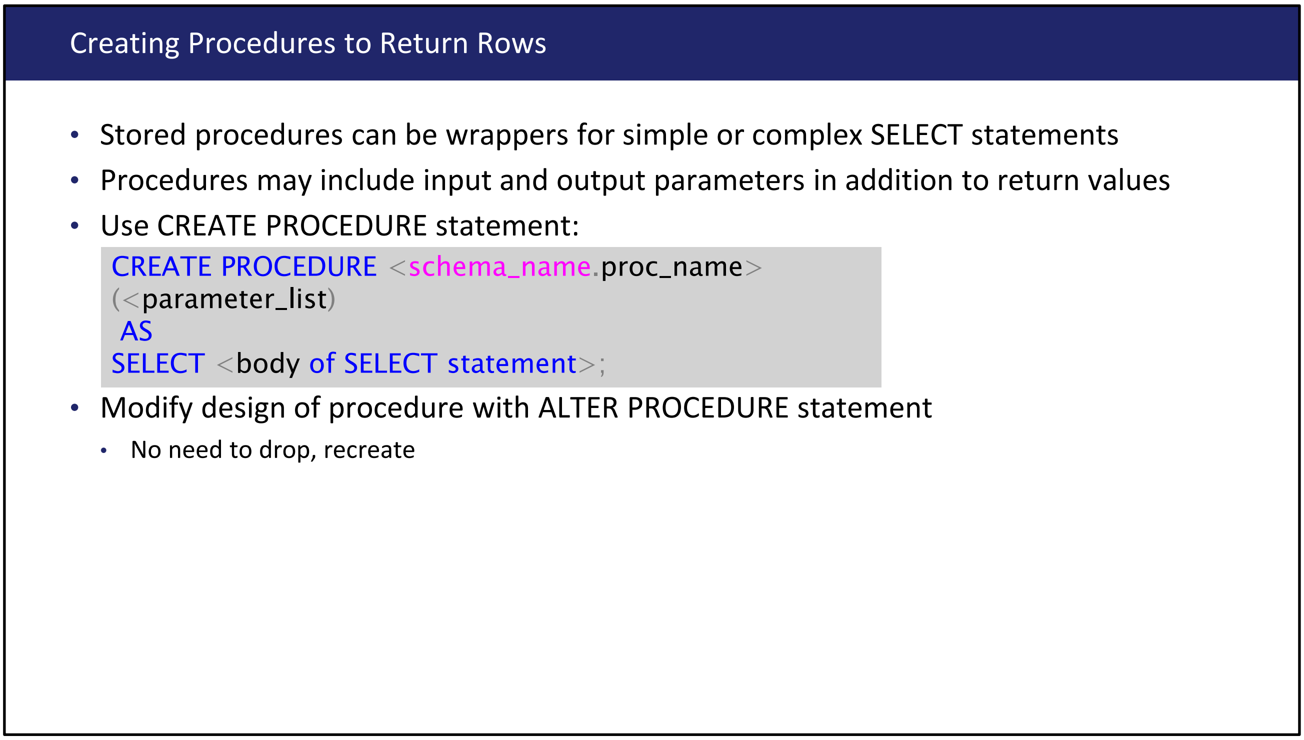
Using a stored procedure with output parameters
https://docs.microsoft.com/en-us/sql/connect/jdbc/using-a-stored-procedure-with-output-parameters

Return data from a stored procedure
https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/return-data-from-a-stored-procedure



Lesson: Creating basic stored procedures

- Creating Procedures to Return Rows
- Creating Procedures That Accept Parameters



Creating Procedures to Return Rows

- Stored procedures can be wrappers for simple or complex SELECT statements
- Procedures may include input and output parameters in addition to return values
- Use CREATE PROCEDURE statement:

```
CREATE PROCEDURE <schema_name.proc_name>
(<parameter_list)
 AS
SELECT <body of SELECT statement>;
```

- Modify design of procedure with ALTER PROCEDURE statement
  - No need to drop, recreate

Creating Procedures That Accept Parameters

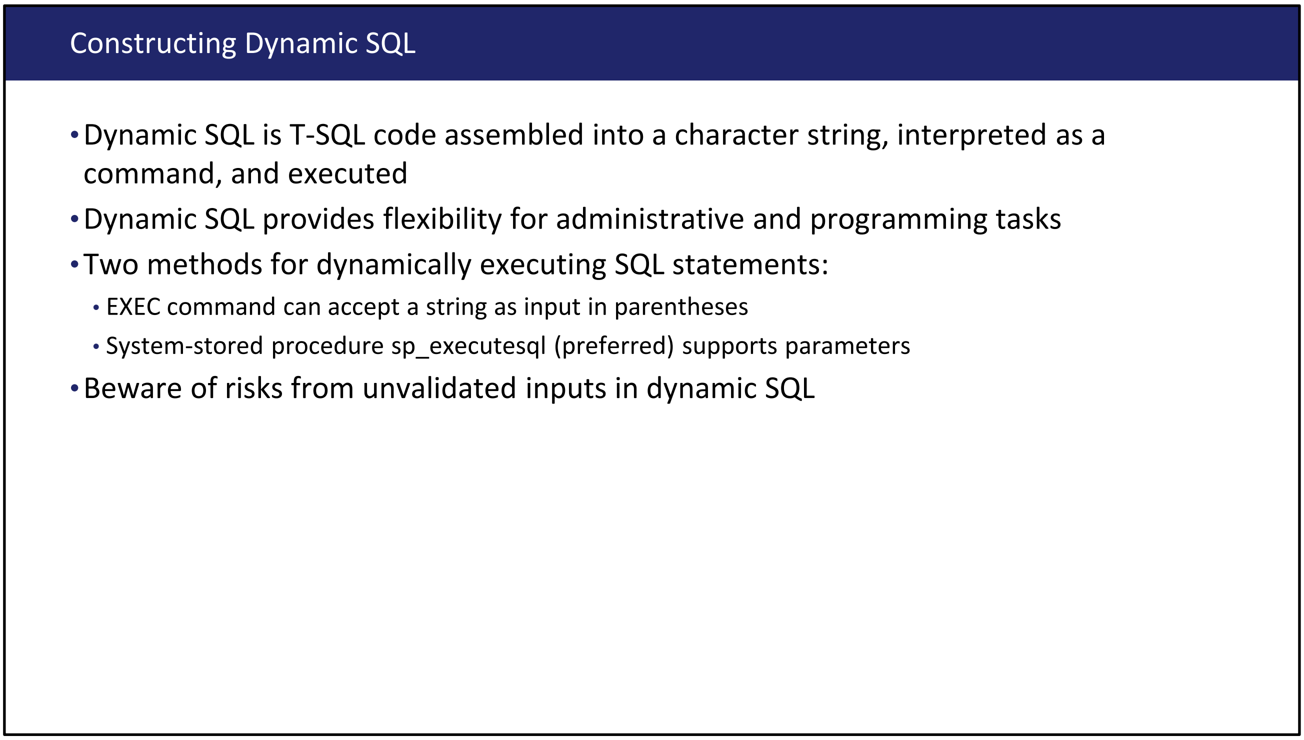
- Input parameters passed to procedure logically behave like local variables within procedure code
- Assign name with @ prefix, data type in procedure header
- Refer to parameter in body of procedure

```
CREATE PROCEDURE Production.ProdsByCategory
(@numrows AS int, @catid AS int)
 AS
SELECT TOP(@numrows) productid,
          productname, unitprice
FROM   Production.Products
WHERE           categoryid = @catid;
```

Demo Developing simple procedure

Lesson: Using dynamic SQL

- Constructing Dynamic SQL
- Writing Queries with Dynamic SQL



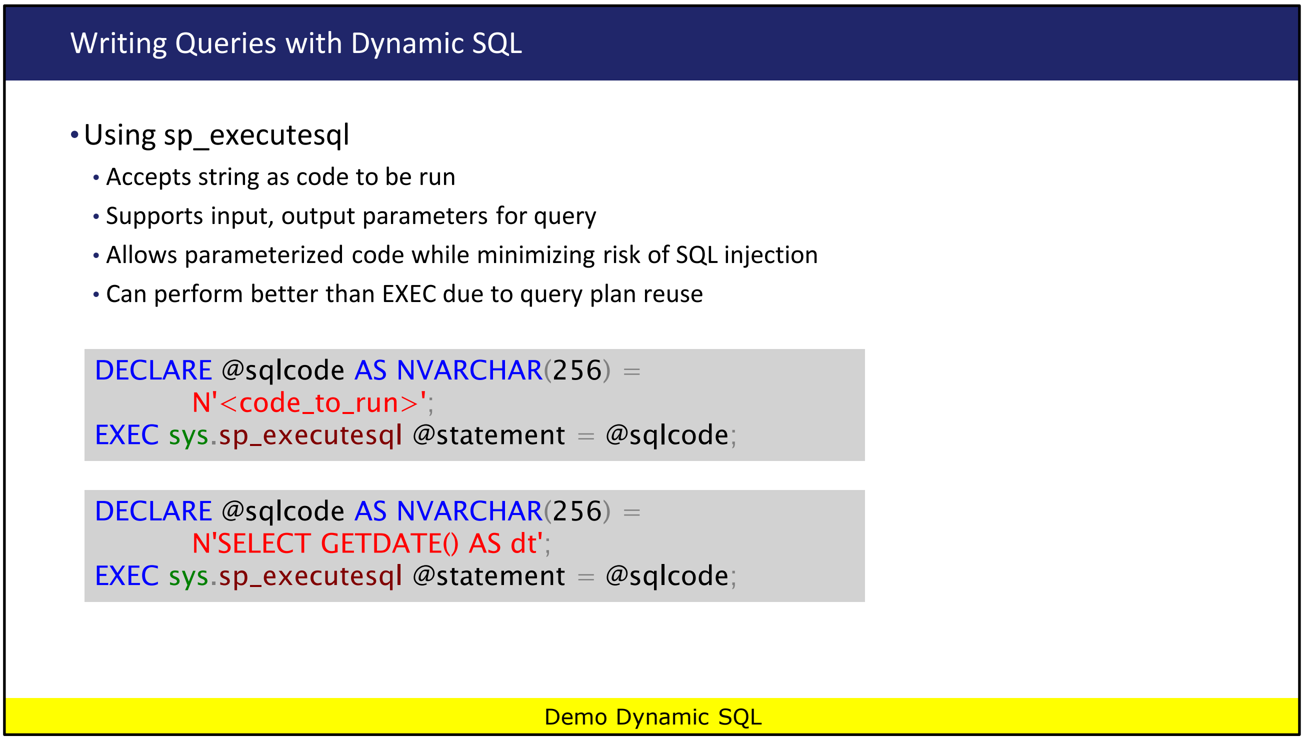
Constructing Dynamic SQL

- Dynamic SQL is T-SQL code assembled into a character string, interpreted as a command, and executed
- Dynamic SQL provides flexibility for administrative and programming tasks
- Two methods for dynamically executing SQL statements:
  - EXEC command can accept a string as input in parentheses
  - System-stored procedure sp_executesql (preferred) supports parameters
- Beware of risks from unvalidated inputs in dynamic SQL

The Curse and Blessings of Dynamic SQL
https://www.sommarskog.se/dynamic_sql.html

Execute Dynamic SQL commands in SQL Server
https://www.mssqltips.com/sqlservertip/1160/execute-dynamic-sql-commands-in-sql-server/

Dynamic SQL in SQL Server
https://www.sqlshack.com/dynamic-sql-in-sql-server/



Writing Queries with Dynamic SQL

- Using sp_executesql
  - Accepts string as code to be run
  - Supports input, output parameters for query
  - Allows parameterized code while minimizing risk of SQL injection
  - Can perform better than EXEC due to query plan reuse

```
DECLARE @sqlcode AS NVARCHAR(256) =
          N'<code_to_run>';
EXEC sys.sp_executesql @statement = @sqlcode;
```

```
DECLARE @sqlcode AS NVARCHAR(256) =
          N'SELECT GETDATE() AS dt';
EXEC sys.sp_executesql @statement = @sqlcode;
```

Demo Dynamic SQL

V1 Lab 5: Executing Stored Procedures

- ***Note carefully the folder name in the lab instructions!!!***
- Exercise 1: Using the EXECUTE Statement to Invoke Stored Procedures
- Exercise 2: Passing Parameters to Stored Procedures
- Exercise 3: Executing System Stored Procedures

Estimated Time: 30 minutes

V2 Lab 5: Using stored procedures

- Ex 1. Executing a stored procedure
- Ex 2. Write your own stored procedure

Estimated Time: 30 minutes