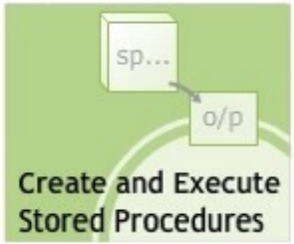
# Module Overview

Welcome to the module **Introducing Stored Procedures**. A stored procedure is a set of Transact-SQL statements that are executed as a single block of code. Stored procedures can be created to carry out repetitive tasks. SQL Server 2005 provides various system stored procedures that assist in viewing information about database objects as well as carrying out other administrative activities.

In this module, you will learn about:

- Stored Procedures
- Create and Execute Stored Procedures

Towards the end of the module, there are demonstrations and/ or simulations for reinforcing the theoretical concepts.

# Lesson Overview

In this first lesson, **Stored Procedures**, you will learn to:

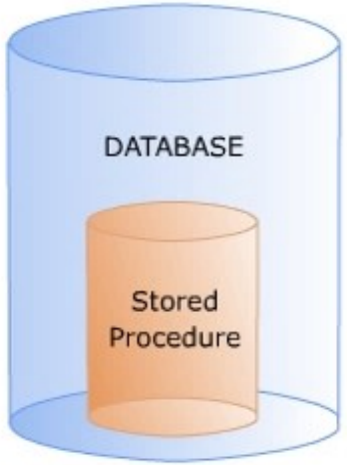
- Explain how to define stored procedures.
- Explain system stored procedures.
- Describe extended stored procedures.
- Describe temporary stored procedures.
- Explain remote stored procedures.
- Explain local stored procedures.
- Describe custom stored procedures.

# Stored Procedures

A stored procedure is a group of Transact-SQL statements that act as a single block of code that performs a specific task. This block of code is identified by an assigned name and is stored in the database in a compiled form.

Stored procedures are useful for performing repetitive tasks. This eliminates the need for repetitively typing out multiple Transact-SQL statements and then repetitively compiling them.

Stored procedures support user-declared variables, conditional execution and other programming features. Also, parameters can be passed between the stored procedure and the calling program.



# Advantages

Database developers or administrators write stored procedures to perform a variety of tasks related to database access and management. This eliminates the need to write Transact-SQL statements every time the same task is to be repeated.

Using stored procedures offers numerous advantages over using Transact-SQL statements. These are:

- Improved Security
- Precompiled Execution
- Reduced Client/Server Traffic
- Reuse of Code

Click on each link to learn more.

# Advantages

Database developers or administrators write stored procedures to perform a variety of tasks related to database access and management. This eliminates the need to write Transact-SQL statements every time the same task is to be repeated.

Using stored procedures offers numerous advantages over using Transact-SQL statements. These are:

- Improved Security
- Precompiled Execution
- Reduced Client/Server Traffic
- Reuse of Code

Click on each link to learn more.

**Improved Security**

The database administrator can improve the security by associating database privileges with stored procedures. Users can be given permission to execute a stored procedure even if the user does not have permission to access the tables or views.

# Advantages

Database developers or administrators write stored procedures to perform a variety of tasks related to database access and management. This eliminates the need to write Transact-SQL statements every time the same task is to be repeated.

Using stored procedures offers numerous advantages over using Transact-SQL statements. These are:

- Improved Security
- Precompiled Execution
- Reduced Client/Server Traffic
- Reuse of Code

Click on each link to learn more.


**Precompiled Execution**

Stored procedures are compiled during the first execution. For every subsequent execution, SQL Server reuses this precompiled version. This reduces the time and resources required for compilation.

# Advantages

Database developers or administrators write stored procedures to perform a variety of tasks related to database access and management. This eliminates the need to write Transact-SQL statements every time the same task is to be repeated.

Using stored procedures offers numerous advantages over using Transact-SQL statements. These are:

- Improved Security
- Precompiled Execution
- Reduced Client/Server Traffic
- Reuse of Code

Click on each link to learn more.


**Reduced Client/Server Traffic**

Stored procedures help in reducing network traffic. When Transact-SQL statements are executed individually, there is network usage separately for execution of each statement. When a stored procedure is executed, the Transact-SQL statements are executed together as a single unit. Network path is not used separately for execution of each individual statement. This reduces network traffic.

# Advantages

Database developers or administrators write stored procedures to perform a variety of tasks related to database access and management. This eliminates the need to write Transact-SQL statements every time the same task is to be repeated.

Using stored procedures offers numerous advantages over using Transact-SQL statements. These are:

- Improved Security
- Precompiled Execution
- Reduced Client/Server Traffic
- Reuse of Code

Click on each link to learn more.

**Reuse of Code**

Stored procedures can be used multiple times. This eliminates the need to repetitively type out hundreds of Transact-SQL statements every time a similar task is to be performed.

# System Stored Procedures

A system stored procedure is a set of pre-compiled Transact-SQL statements executed as a single unit. System procedures are used in database administrative and informational activities. These procedures provide easy access to the metadata information about database objects such as system tables, user-defined tables, views, and indexes.

System stored procedures logically appear in the `sys` schema of system and user-defined databases. When referencing a system stored procedure, the `sys` schema identifier is used. The system stored procedures are stored physically in the `Resource` database and have the `'sp_'` prefix.

**System Stored Procedure**

| | Text |
|---|---|
| 1 | create procedure sys.sp_helpindex |
| 2 | @objname nvarchar(776) -- the table to check for indexes |
| 3 | as |
| 4 | -- PRELIM |
| 5 | set nocount on |
| 6 | |
| 7 | declare @objid int, -- the object id of the table |
| 8 | @indid smallint, -- the index id of an index |
| 9 | @groupid int, -- the filegroup id of an index |
| 10 | @indname sysname, |
| 11 | @groupname sysname, |

More

# System Stored Procedures

A system stored procedure is a set of pre-compiled Transact-SQL statements executed as a single unit. System procedures are used in database administrative and informational activities. These procedures provide easy access to the metadata information about database objects such as system tables, user-defined tables, views, and indexes.

System stored procedures logically appear in the `sys` schema of system and user-defined databases. When referencing a system stored procedure, the `sys` schema identifier is used. The system stored procedures are stored physically in the `Resource` database and have the `'sp_'` prefix.

**System Stored Procedure**

| | Text |
|---|---|
| 1 | create procedure sys.sp_helpindex |
| 2 | @objname nvarchar(776) -- the table to check for indexes |
| 3 | as |
| 4 | -- PRELIM |
| 5 | set nocount on |
| 6 | |
| 7 | declare @objid int, -- the object id of the table |
| 8 | @indid smallint, -- the index id of an index |
| | -- the filegroup id of an index |
| | name, |
| | ysname, |

**More**

System tables are created by default at the time of creating a new database. These tables store the metadata information about user-defined objects such as tables, views etc. Users cannot access or update the system tables using system stored procedures except through permissions granted by a database administrator.

System stored procedures are owned by the database administrator.

More

# Types of System Stored Procedures

System stored procedures can be classified into different categories depending on the tasks they perform. Some of the important categories are:

- Catalog Stored Procedures
- Security Stored Procedures
- Cursor Stored Procedures
- Distributed Query Stored Procedures
- Database Mail and SQL Mail Stored Procedures

Click on each link to learn more.

| Catalog Stored Procedures |
|---|
| sp_columns |
| sp_database |
| sp_statistics |

| Security Stored Procedures |
|---|
| sp_addalias |
| sp_addapprole |
| sp_addserver |

| Cursor Stored Procedures |
|---|
| sp_cursor_list |
| sp_describe_cursor |
| sp_describe_columns |

| Distributed Query Stored Procedure |
|---|
| sp_indexes |
| sp_catalogs |
| sp_primarykeys |

| Database Mail and SQL Mail Stored Procedure |
|---|
| sp_send_dbmail |
| sysmail_add_profile_sp |
| sysmail_help_configure_sp |

# Types of System Stored Procedures

System stored procedures can be classified into different categories depending on the tasks they perform. Some of the important categories are:

- Catalog Stored Procedures
- Security Stored Procedures
- Cursor Stored Procedures
- Distributed Query Stored Procedures
- Database Mail and SQL Mail Stored Procedures

Click on each link to learn more.

| Catalog Stored Procedures |
|---|
| sp_columns |
| sp_database |
| sp_statistics |

| Security Stored Procedures |
|---|
| sp_addalias |
| sp_addapprole |
| sp_addserver |

| Cursor Stored Procedures |
|---|
| sp_cursor_list |
| sp_describe_cursor |
| sp_describe_columns |

| Distributed Query Stored Procedure |
|---|
| sp_indexes |
| sp_catalogs |
| sp_primarykeys |

| Database Mail and SQL Mail Stored Procedure |
|---|
| sp_send_dbmail |
| sysmail_add_profile_sp |
| sysmail_help_configure_sp |

**Catalog Stored Procedures**

All information about tables in the user database is stored in a set of tables called the system catalog. Information from the system catalog can be accessed using catalog procedures. For example, the `sp_tables` catalog stored procedure displays the list of all the tables in the current database.

# Types of System Stored Procedures

System stored procedures can be classified into different categories depending on the tasks they perform. Some of the important categories are:

- Catalog Stored Procedures
- Security Stored Procedures
- Cursor Stored Procedures
- Distributed Query Stored Procedures
- Database Mail and SQL Mail Stored Procedures

Click on each link to learn more.

**Security Stored Procedures**

Security stored procedures are used to manage the security of the database. For example, the `sp_changedbowner` security stored procedure is used to change the owner of the current database.

| Catalog Stored Procedures |
|---|
| sp_columns |
| sp_database |
| sp_statistics |

| Security Stored Procedures |
|---|
| sp_addalias |
| sp_addapprole |
| sp_addserver |

| Cursor Stored Procedures |
|---|
| sp_cursor_list |
| sp_describe_cursor |
| sp_describe_columns |

| Distributed Query Stored Procedure |
|---|
| sp_indexes |
| sp_catalogs |
| sp_primarykeys |

| Database Mail and SQL Mail Stored Procedure |
|---|
| sp_send_dbmail |
| sysmail_add_profile_sp |
| sysmail_help_configure_sp |

# Types of System Stored Procedures

System stored procedures can be classified into different categories depending on the tasks they perform. Some of the important categories are:

- Catalog Stored Procedures
- Security Stored Procedures
- Cursor Stored Procedures
- Distributed Query Stored Procedures
- Database Mail and SQL Mail Stored Procedures

Click on each link to learn more.

| Catalog Stored Procedures |
|---|
| sp_columns |
| sp_database |
| sp_statistics |

| Security Stored Procedures |
|---|
| sp_addalias |
| sp_addapprole |
| sp_addserver |

| Cursor Stored Procedures |
|---|
| sp_cursor_list |
| sp_describe_cursor |
| sp_describe_columns |

| Distributed Query Stored Procedure |
|---|
| sp_indexes |
| sp_catalogs |
| sp_primarykeys |

| Database Mail and SQL Mail Stored Procedure |
|---|
| sp_send_dbmail |
| sysmail_add_profile_sp |
| sysmail_help_configure_sp |

**Cursor Stored Procedures**

Cursor procedures are used to implement the functionality of a cursor. For example, the `sp_cursor_list` cursor stored procedure lists all the cursors opened by the connection and describes their attributes.

# Types of System Stored Procedures

System stored procedures can be classified into different categories depending on the tasks they perform. Some of the important categories are:

- Catalog Stored Procedures
- Security Stored Procedures
- Cursor Stored Procedures
- Distributed Query Stored Procedures
- Database Mail and SQL Mail Stored Procedures

Click on each link to learn more.

| Catalog Stored Procedures |
|---|
| sp_columns |
| sp_database |
| sp_statistics |

| Security Stored Procedures |
|---|
| sp_addalias |
| sp_addapprole |
| sp_addserver |

| Cursor Stored Procedures |
|---|
| sp_cursor_list |
| sp_describe_cursor |
| sp_describe_columns |

| Distributed Query Stored Procedure |
|---|
| sp_indexes |
| sp_catalogs |
| sp_primarykeys |

| Database Mail and SQL Mail Stored Procedure |
|---|
| sp_send_dbmail |
| sysmail_add_profile_sp |
| sysmail_help_configure_sp |

**Distributed Query Stored Procedures**

Distributed stored procedures are used in the management of distributed queries. For example, the `sp_indexes` distributed query stored procedure returns index information for the specified remote table.

# Types of System Stored Procedures

System stored procedures can be classified into different categories depending on the tasks they perform. Some of the important categories are:

- Catalog Stored Procedures
- Security Stored Procedures
- Cursor Stored Procedures
- Distributed Query Stored Procedures
- Database Mail and SQL Mail Stored Procedures

Click on each link to learn more.

| Catalog Stored Procedures |
|---|
| sp_columns |
| sp_database |
| sp_statistics |

| Security Stored Procedures |
|---|
| sp_addalias |
| sp_addapprole |
| sp_addserver |

| Cursor Stored Procedures |
|---|
| sp_cursor_list |
| sp_describe_cursor |
| sp_describe_columns |

| Distributed Query Stored Procedure |
|---|
| sp_indexes |
| sp_catalogs |
| sp_primarykeys |

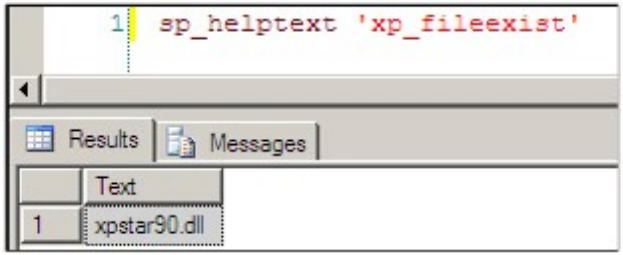
| Database Mail and SQL Mail Stored Procedure |
|---|
| sp_send_dbmail |
| sysmail_add_profile_sp |
| sysmail_help_configure_sp |

**Database Mail and SQL Mail Stored Procedures**

Database Mail and SQL Mail stored procedures are used to perform e-mail operations from within the SQL Server. For example, the `sp_send_dbmail` database mail stored procedure sends e-mail messages to specified recipients. The message may include a query result set or file attachments or both.

# Extended Stored Procedures

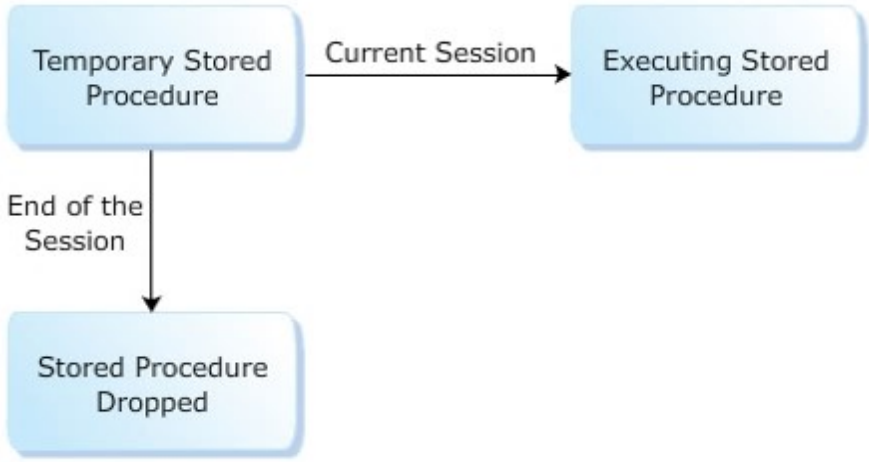
The extended stored procedures are not residents of SQL Server. They are procedures that are implemented as dynamic-link libraries (DLL) executed outside the SQL Server environment. Extended stored procedures use the 'xp_' prefix. Tasks that are complicated or cannot be executed using Transact-SQL statements are performed using extended stored procedures.



# Temporary Stored Procedures

Stored procedures created for temporary use within a session are called temporary stored procedures. These procedures are stored in the `tempdb` database.

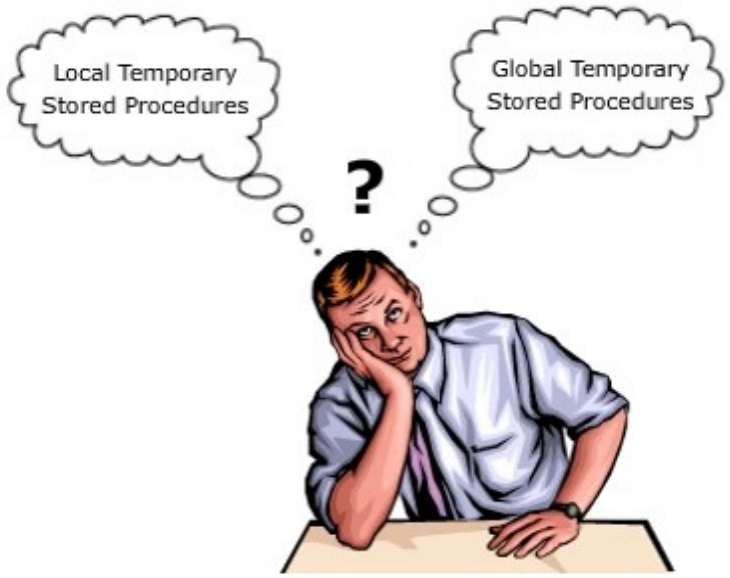
The `tempdb` system database is a global resource available to all users connected to an instance of SQL Server. It holds all temporary tables and temporary stored procedures.



# Types of Temporary Stored Procedure

SQL Server 2005 supports two types of temporary stored procedures namely, local and global. The differences between the two types are given in the table.

| Local Temporary Procedure | Global Temporary Procedure |
|---|---|
| Visible only to the user that created it. | Visible to all users. |
| Dropped at the end of the current session. | Dropped at the end of the last session. |
| Can only be used by its owner. | Can be used by any user. |
| Uses the # prefix before the procedure name. | Uses the ## prefix before the procedure name. |

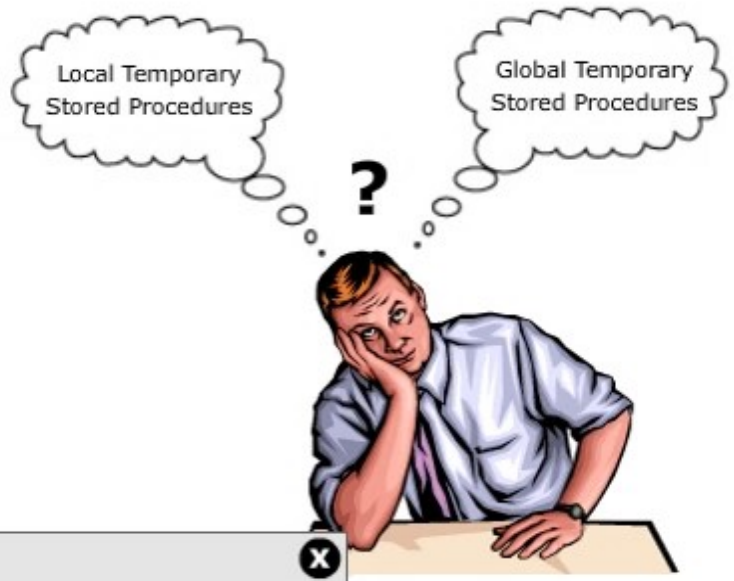



More

# Types of Temporary Stored Procedure

SQL Server 2005 supports two types of temporary stored procedures namely, local and global. The differences between the two types are given in the table.

| Local Temporary Procedure | Global Temporary Procedure |
|---|---|
| Visible only to the user that created it. | Visible to all users. |
| Dropped at the end of the current session. | Dropped at the end of the last session. |
| Can only be used by its owner. | Can be used by any user. |
| Uses the # prefix before the procedure name. | Uses the ## prefix before the procedure name. |




More

**More**

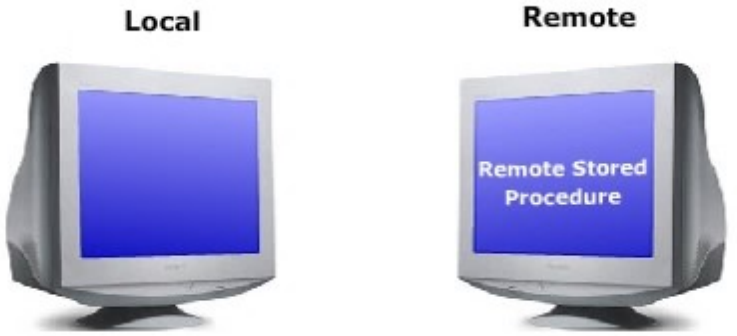
A session is established when a user connects to the database and is ended when the user disconnects.

The complete name of a global temporary stored procedure including the prefix ## cannot exceed 128 characters. The complete name of a local temporary stored procedure including the prefix # cannot exceed 116 characters.

# Remote Stored Procedures

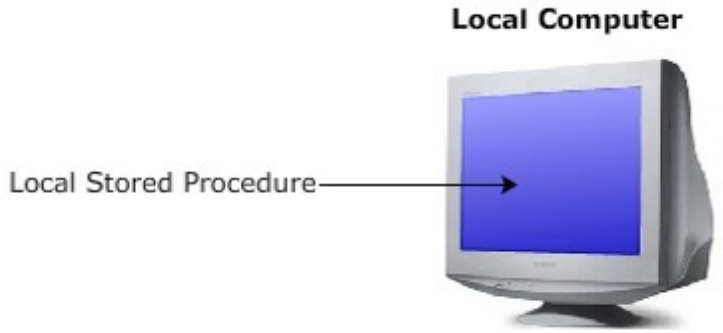
Stored procedures that run on remote SQL Servers are known as remote stored procedures. Remote stored procedures can be used only when the remote server allows remote access.

When a remote stored procedure is executed from a local instance of SQL Server to a client computer, a statement abort error might be encountered. When such an error occurs, the statement that caused the error is terminated but the remote procedure continues to be executed.



# Local Stored Procedures

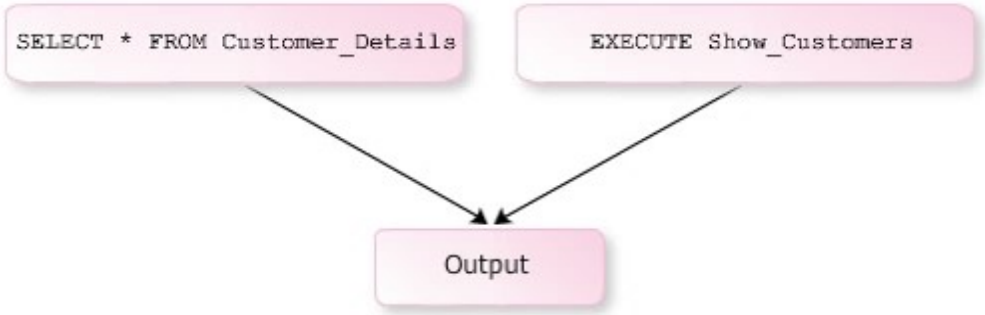
Local stored procedures are created in individual user databases. A local stored procedure cannot be accessed by any user other than the one who has created it. If the local procedure is a temporary procedure, it is dropped at the end of the session.



# Custom Stored Procedures

In SQL Server 2005, users are allowed to create customized stored procedures for performance of various tasks. Such stored procedures are referred to as user-defined or custom stored procedures.

For example, consider a table `Customer_Details` that stores the details about all the customers. You would need to type out Transact -SQL statements every time you wished to view the details about the customers. Instead, you could create a custom stored procedure that would display these details whenever the procedure is executed.




Syntax

Snippet

More

# Custom Stored Procedures

In SQL Server 2005, users are allowed to create customized stored procedures for performance of various tasks. Such stored procedures are referred to as user-defined or custom stored procedures.

For example, consider a table `Customer_Details` that stores the details about all the customers. You would need to type out Transact -SQL statements every time you wished to view the details about the customers. Instead, you could create a custom stored procedure that would display these details whenever the procedure is executed.

```
SELECT * FROM Customer_Details
```

```
EXECUTE Show_Customers
```

The following syntax is used to create a custom stored procedure.

```
CREATE { PROC | PROCEDURE } procedure_name
    [ { @parameter data_type }  ]
AS <sql_statement>
```

where,

`procedure_name`: Specifies the name of the procedure.

`@parameter`: Specifies the input/output parameters in the procedure.

`data_type`: Specifies the data types of the parameters.

`sql_statement`: Specifies one or more Transact-SQL statements to be included in the procedure.

Syntax

Syntax

Snippet

More

# Custom Stored Procedures

In SQL Server 2005, users are allowed to create customized stored procedures for performance of various tasks. Such stored procedures are referred to as user-defined or custom stored procedures.

For example, consider a table `Customer_Details` that stores the details about all the customers. You would need to type out Transact -SQL statements every time you wished to view the details about the customers. Instead, you could create a custom stored procedure that would display these details whenever the procedure is executed.

`SELECT * FROM Customer Details`

`EXECUTE Show Customers`

The following code creates a custom stored procedure, `Show_Customers` which will display the details of customers.

```
CREATE PROCEDURE Show_Customers
AS
SELECT * FROM Customer_Details
```

The following code is used to execute the `Show_Customers` stored procedure, which will display the details about the customers.

```
EXECUTE Show_Customers
```

Snippet

Syntax

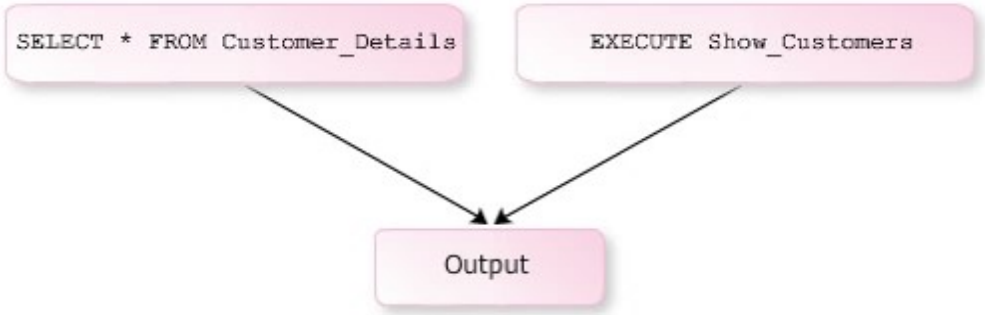
Snippet

More

# Custom Stored Procedures

In SQL Server 2005, users are allowed to create customized stored procedures for performance of various tasks. Such stored procedures are referred to as user-defined or custom stored procedures.

For example, consider a table `Customer_Details` that stores the details about all the customers. You would need to type out Transact -SQL statements every time you wished to view the details about the customers. Instead, you could create a custom stored procedure that would display these details whenever the procedure is executed.




Syntax

Snippet

More

**More**

Custom stored procedures can be set to override the default system procedures.

## Knowledge Check


Which of these statements about stored procedures are true and which statements are false?

Select an option for each statement and then click on **Submit**.

| | Statements | True | False |
|---|---|---|---|
| (A) | A stored procedure is a group of TRANSACT-SQL statements that act as a block of code used to perform a particular task. | ○ | ○ |
| (B) | All system stored procedures are identified by the 'xp_' prefix. | ○ | ○ |
| (C) | A distributed stored procedure is used in the management of distributed queries. | ○ | ○ |
| (D) | Database Mail and SQL mail procedures are used to perform e-mail operations within SQL Server. | ○ | ○ |
| (E) | User-defined stored procedures are also known as custom stored procedures. | ○ | ○ |

Submit

33 | 8

Score

## Knowledge Check

Which of these statements about stored procedures are true and which statements are false?

Select an option for each statement and then click on **Submit**.

| | Statements | True | False |
|---|---|---|---|
| (A) | A stored procedure is a group of TRANSACT-SQL statements that act as a block of code used to perform a particular task. | ◉ | ○ |
| (B) | All system stored procedures are identified by the 'xp_' prefix. | ◉ | ○ |
| (C) | A distributed stored procedure is used in the management of distributed queries. | ◉ | ○ |
| (D) | Database Mail and SQL mail procedures are used to perform e-mail operations within SQL Server. | ◉ | ○ |
| (E) | User-defined stored procedures are also known as custom stored procedures. | ◉ | ○ |

**Correct**

The correct answers are displayed.

Submit

33 | 8

Score

## Knowledge Check

Can you match the SQL Server stored procedures against their corresponding description?

Select an option from each drop-down list box and then click on **Submit**.

| | Description | Stored Procedures |
|---|---|---|
| (A) | Stored procedure created for temporary use within a session. | Select Step... |
| (B) | Stored procedure created in individual user database. | Select Step... |
| (C) | Stored procedure that runs on remote SQL Server. | Select Step... |
| (D) | Stored procedure that is dropped at the end of the last session. | Select Step... |
| (E) | Stored procedure that is stored in the `tempdb` database. | Select Step... |

Submit

View Answer

34 | 8

Score

## Knowledge Check

Can you match the SQL Server stored procedures against their corresponding description?

Select an option from each drop-down list box and then click on **Submit**.

| | Description | Stored Procedures |
|---|---|---|
| (A) | Stored procedure created for temporary use within a session. | Temporary |
| (B) | Stored procedure created in individual user database. | Local |
| (C) | Stored procedure that runs on remote SQL Server. | Remote |
| (D) | Stored procedure that is dropped at the end of the last session. | Global |
| (E) | Stored procedure that is stored in the `tempdb` database. | Temporary |

**Correct**

The correct answers are displayed.

Submit

View Answer

# Lesson Review

In this first lesson, **Stored Procedures**, you learnt to:

- Explain how to define stored procedures.
- Explain system stored procedures.
- Describe extended stored procedures.
- Describe temporary stored procedures.
- Explain remote stored procedures.
- Explain local stored procedures.
- Describe custom stored procedures.

# Lesson Overview

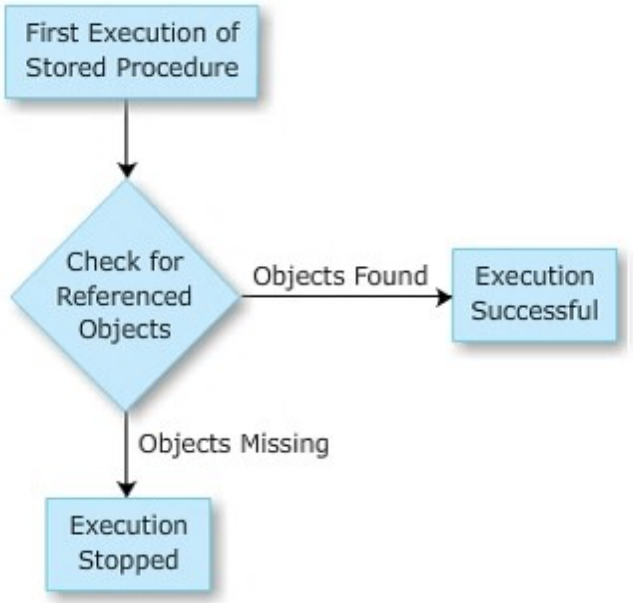
In this second lesson, **Create and Execute Stored Procedures**, you will learn to:

- Define deferred name resolution.
- Describe how to create Transact-SQL stored procedures.
- Describe how to execute extended stored procedures.

# Deferred Name Resolution

Statements in a stored procedure are checked for errors during the creation of the stored procedure. If an error is found, an error message is returned and the procedure is not created. If there are no errors, the procedure is created and the text of the procedure is stored in the `sys.sql_modules` catalog view.

At the first instance of the stored procedure execution, the query processor reads the statements in the stored procedure from the `sys.sql_modules` catalog view. The processor then checks for the name of all objects used by the procedure. This check is known as deferred name resolution. If a referenced object is missing, the execution stops.

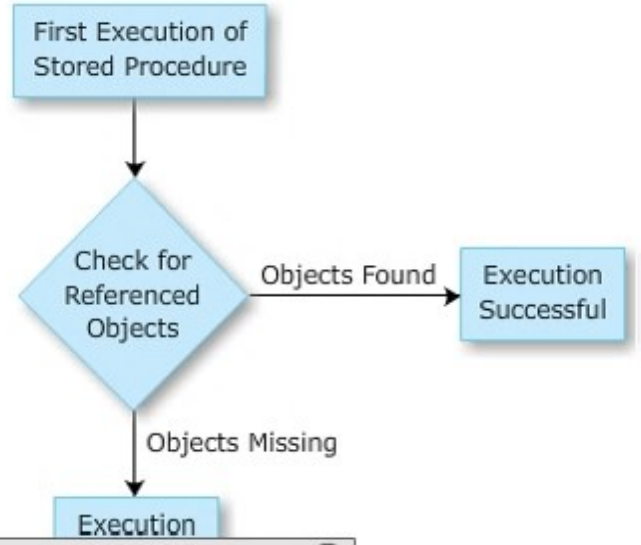



More

# Deferred Name Resolution

Statements in a stored procedure are checked for errors during the creation of the stored procedure. If an error is found, an error message is returned and the procedure is not created. If there are no errors, the procedure is created and the text of the procedure is stored in the `sys.sql_modules` catalog view.

At the first instance of the stored procedure execution, the query processor reads the statements in the stored procedure from the `sys.sql_modules` catalog view. The processor then checks for the name of all objects used by the procedure. This check is known as deferred name resolution. If a referenced object is missing, the execution stops.




**More**

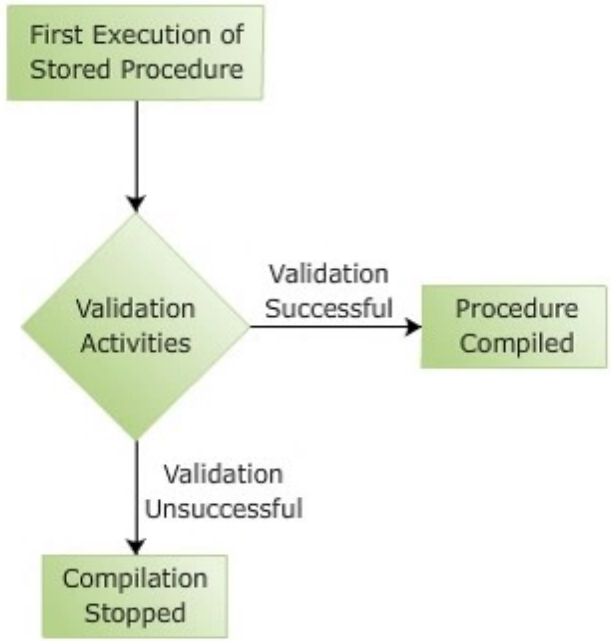
Deferred name resolution is carried out at the time of the first execution of the procedure, not at the time of creation. Thus, objects that do not exist at the time of creation can be added at any time prior to the first execution of the procedure.

More

# Deferred Name Resolution Process

Validation activities other than checking for referenced objects are also carried out in the deferred name resolution stage. For example, variables are checked for compatibility with the column data types. If an error is found in the resolution stage, SQL Server returns an error.

If the resolution stage of the execution process is successfully completed, the procedure is compiled. For every subsequent call to the procedure, this compiled version is used.



# Transact-SQL Stored Procedures

A stored procedure can be created either for permanent use or for temporary use. When a stored procedure is created for temporary use within a session, it is referred to as a local temporary procedure. When a stored procedure is created for temporary use within all sessions, it is referred to as a global temporary procedure.

A stored procedure is created using the `CREATE PROCEDURE` statement.

**Stored Procedure Definitions**

- CREATE PROCEDURE Statement
- Input/Output Parameters
- Select Statement



# Transact-SQL Stored Procedures

A stored procedure can be created either for permanent use or for temporary use. When a stored procedure is created for temporary use within a session, it is referred to as a local temporary procedure. When a stored procedure is created for temporary use within all sessions, it is referred to as a global temporary procedure.

A stored procedure is created using the `CREATE PROCEDURE` statement.

Syntax

Snippet

More

The following syntax is used to create a stored procedure.

```
CREATE { PROC | PROCEDURE } procedure_name
    [ { @parameter data_type }
        [ = DEFAULT ] [ OUTPUT ]
    ]
[ WITH ENCRYPTION ]
AS <sql_statement>
```

where,

`DEFAULT`: Specifies a default value for the parameter.

`OUTPUT`: Specifies that the parameter is an output parameter.

`WITH ENCRYPTION`: Specifies that the definition of the stored procedure will not be visible.

Syntax

# Transact-SQL Stored Procedures

A stored procedure can be created either for permanent use or for temporary use. When a stored procedure is created for temporary use within a session, it is referred to as a local temporary procedure. When a stored procedure is created for temporary use within all sessions, it is referred to as a global temporary procedure.

A stored procedure is created using the `CREATE PROCEDURE` statement.

Syntax

Snippet

More

The following code creates a stored procedure named `Display_Customers`. This stored procedure displays the values in the `CustID`, `AccNo`, `AccName`, `City`, `State`, and `Country` columns from the `Customer_Details` table.

```
CREATE PROCEDURE Display_Customers
AS
     SELECT CustID, AccNo, AccName, City, State,
Country
     FROM Customer_Details
```

The following code is used to execute the `Display_Customers` procedure.

```
EXECUTE Display_Customers
```

Snippet

# Transact-SQL Stored Procedures

A stored procedure can be created either for permanent use or for temporary use. When a stored procedure is created for temporary use within a session, it is referred to as a local temporary procedure. When a stored procedure is created for temporary use within all sessions, it is referred to as a global temporary procedure.

A stored procedure is created using the `CREATE PROCEDURE` statement.

**Stored Procedure Definitions**

- CREATE PROCEDURE Statement
- Input/Output Parameters
- Select Statement

Syntax

Snippet

More

**More**

A stored procedure can be up to 128MB in size.

# Using "OUTPUT" Clause

The OUTPUT clause returns information from each row on which the `INSERT`, `UPDATE` and `DELETE` statements have been executed. This clause is useful to retrieve the value of an identity or computed column after an `INSERT` or `UPDATE` operation.

```
1  CREATE PROCEDURE Max_Salary
2      @max_sal int OUTPUT
3  AS
4  SELECT @max_sal = MAX(Salary) FROM Employee_Details
5
6  DECLARE @max_sal int
7  EXECUTE Max_Salary @max_sal OUTPUT
8  PRINT @max_sal
```

Messages

```
40000
```

Snippet

# Using "OUTPUT" Clause

The OUTPUT clause returns information from each row on which the `INSERT`, `UPDATE` and `DELETE` statements have been executed. This clause is useful to retrieve the value of an identity or computed column after an `INSERT` or `UPDATE` operation.

Snippet

The following code creates a procedure `Max_Salary` with an `OUTPUT` parameter `max_sal`. The procedure retrieves the maximum salary from the `Employee_Details` table.

```
CREATE PROCEDURE Max_Salary
     @max_sal int OUTPUT
AS
SELECT @max_sal = MAX(Salary) FROM Employee_Details

DECLARE @max_sal int;
EXECUTE Max_Salary @max_sal OUTPUT;
PRINT @max_sal
```

The above code retrieves the maximum salary from the `Employee_Details` table through the `OUTPUT` parameter and prints it.

Snippet

# Guidelines for Creating Stored Procedures

Consider the following guidelines when you create stored procedures:

- When a local temporary table is created inside a stored procedure, the table disappears when the procedure is exited.
- Stored procedures can reference tables, views, user-defined functions and other stored procedures.
- When a stored procedure calls another stored procedure, the called procedure can access all objects created by the calling procedure.
- Changes made by a remote stored procedure on a remote instance of Microsoft SQL Server 2005 cannot be rolled back.
- A stored procedure can have a maximum of 2100 parameters.
- The number of local variables in a stored procedure depends upon available memory.

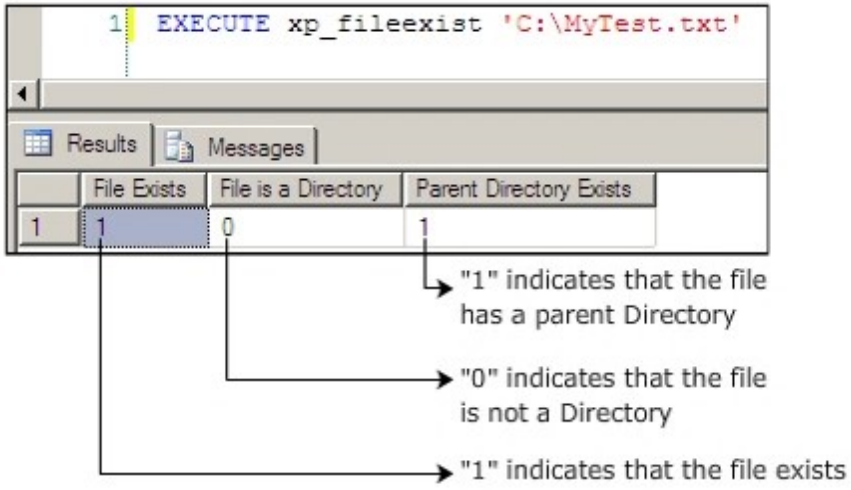
# Extended Stored Procedures

Extended stored procedures are used to perform tasks that are unable to be performed using standard Transact-SQL statements. Extended stored procedures use the `'xp_'` prefix. These stored procedures are contained in the `dbo` schema of the `master` database.

The following points need to be considered while executing extended stored procedures:

- Extended stored procedures can be executed under the security limitations specified by the SQL Server.
- Extended stored procedures execute in the process space of the SQL Server.
- When an extended stored procedure is executed, the extended stored procedure's DLL remains in the address space of the server unless it is unloaded by the administrator.

Extended stored procedures are executed using the `EXECUTE` statement.




Syntax

Snippet

More

# Extended Stored Procedures

Extended stored procedures are used to perform tasks that are unable to be performed using standard Transact-SQL statements. Extended stored procedures use the `'xp_'` prefix. These stored procedures are contained in the `dbo` schema of the `master` database.

The following syntax is used to execute an extended stored procedure:

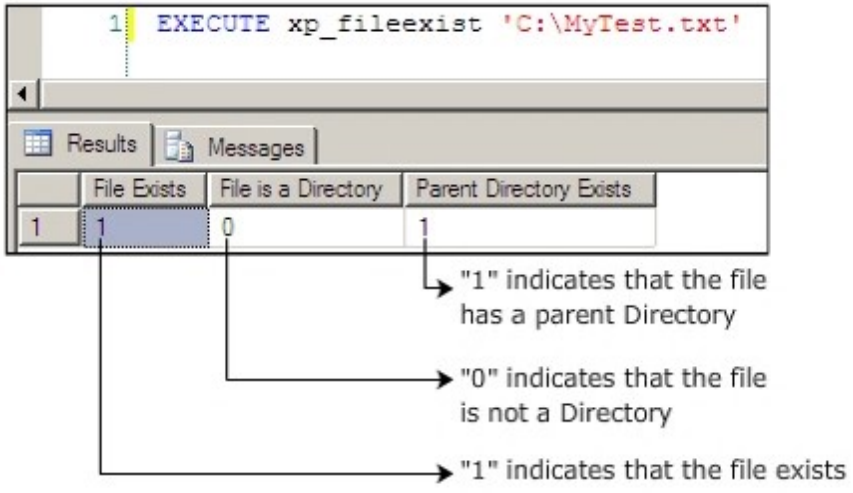
```
EXECUTE <procedure_name>
```

Syntax

Extended stored procedures are executed using the `EXECUTE` statement.

Syntax | Snippet

More



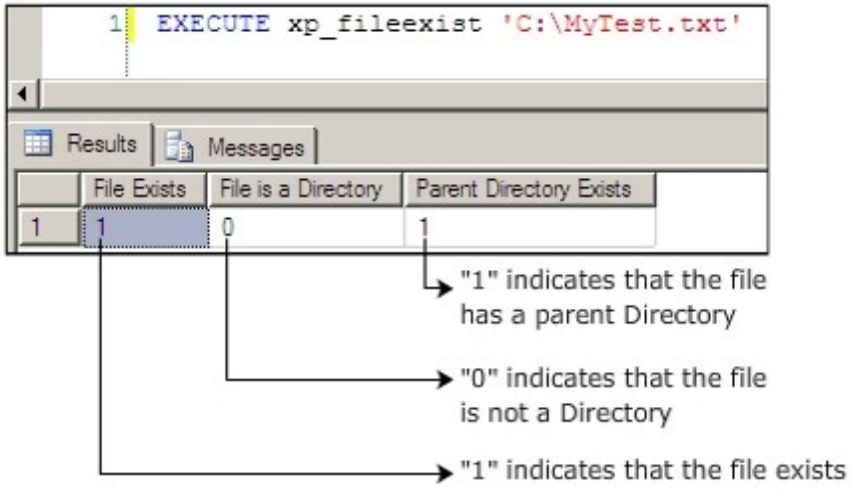
# Extended Stored Procedures

Extended stored procedures are used to perform tasks that are unable to be performed using standard Transact-SQL statements. Extended stored procedures use the 'xp_' prefix. These stored procedures are contained in the dbo schema of the master database.

The following code executes the extended stored procedure xp_fileexist to check whether the MyTest.txt file exists or not:

```
EXECUTE xp_fileexist 'c:\MyTest.txt'
```

Snippet




Extended stored procedures are executed using the EXECUTE statement.

Syntax

Snippet

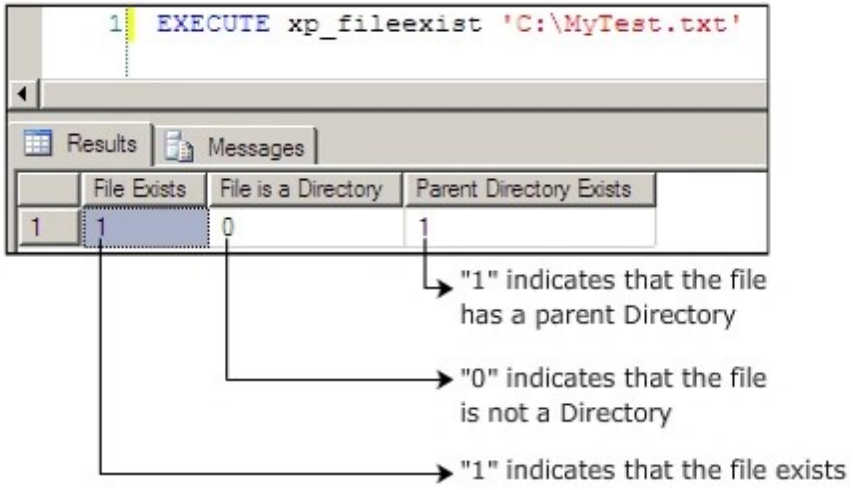
More

# Extended Stored Procedures

Extended stored procedures are used to perform tasks that are unable to be performed using standard Transact-SQL statements. Extended stored procedures use the 'xp_' prefix. These stored procedures are contained in the dbo schema of the master database.

The following points need to be considered while executing extended stored procedures:

- Extended stored procedures can be executed under the security limitations specified by the SQL Server.
- Extended stored procedures execute in the process space of the SQL Server.
- When an extended stored procedure is executed, the extended stored procedure's DLL remains in the address space of the server unless it is unloaded by the administrator.




Extended stored procedures are executed using statement.

Syntax

Snippet

More

**More**

When you execute an extended stored procedure, either in a batch or a module, qualify the stored procedure name with master.dbo.

## Knowledge Check


Which of these statements about deferred name resolution and creating and executing stored procedures are true and which statements are false?

Select an option for each statement and then click on **Submit**.

| | Statements | True | False |
|---|---|---|---|
| (A) | Extended stored procedures are executed using the `EXECUTE` statement. | ○ | ○ |
| (B) | Extended stored procedures are executed under the security limitations specified by the SQL Server. | ○ | ○ |
| (C) | Variables are checked for compatibility with the column data types during the deferred resolution stage. | ○ | ○ |
| (D) | The stored procedure execution process stops when an object referenced by the procedure is missing. | ○ | ○ |
| (E) | Statements in the stored procedure are checked for errors when creating a stored procedure. | ○ | ○ |

Submit

35 | 8

Score

## Knowledge Check

Which of these statements about deferred name resolution and creating and executing stored procedures are true and which statements are false?

Select an option for each statement and then click on **Submit**.

| | Statements | True | False |
|---|---|---|---|
| (A) | Extended stored procedures are executed using the `EXECUTE` statement. | ◉ | ○ |
| (B) | Extended stored procedures are executed under the security limitations specified by the SQL Server. | ◉ | ○ |
| (C) | Variables are checked for compatibility with the column data types during the deferred resolution stage. | ◉ | ○ |
| (D) | The stored procedure execution process stops when an object referenced by the procedure is missing. | ◉ | ○ |
| (E) | Statements in the stored procedure are checked for errors when creating a stored procedure. | ◉ | ○ |

**Correct**

The correct answers are displayed.

Submit

35 | 8

Score

## Knowledge Check

Which of these statements about deferred name resolution and creating and executing stored procedures are true and which statements are false?

Select an option for each statement and then click on **Submit**.

| | Statements | True | False |
|---|---|---|---|
| (A) | Extended stored procedures are executed using the `EXECUTE` statement. | ◉ | ○ |
| (B) | Extended stored procedures are executed under the security limitations specified by the SQL Server. | ◉ | ○ |
| (C) | Variables are checked for compatibility with the column data types during the deferred resolution stage. | ◉ | ○ |
| (D) | The stored procedure execution process stops when an object referenced by the procedure is missing. | ◉ | ○ |
| (E) | Statements in the stored procedure are checked for errors when creating a stored procedure. | ◉ | ○ |

**Correct**

The correct answers are displayed.

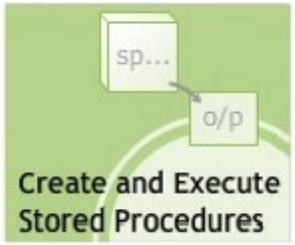
Submit

# Module Summary

In this module, **Introducing Stored Procedures**, you learnt about:

- Stored Procedures
- Create and Execute Stored Procedures

Click on each link for a summary of the lesson.

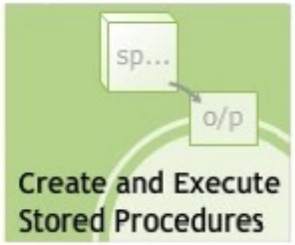



# Module Summary

In this module, **Introducing Stored Procedures**, you learnt about:

- Stored Procedures
- Create and Execute Stored Procedures

Click on each link for a summary of the lesson.


**Stored Procedures**

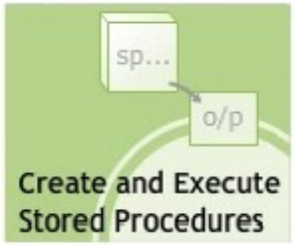
A stored procedure is a group of Transact-SQL statements executed as a single block of code. SQL Server 2005 provides various system stored procedures that assist in database administrative activities. Also, a user can create custom stored procedures for performing various tasks.

# Module Summary

In this module, **Introducing Stored Procedures**, you learnt about:

- Stored Procedures
- Create and Execute Stored Procedures

Click on each link for a summary of the lesson.


**Create and Execute Stored Procedure**

Deferred name resolution is the stage where the processor checks for names of the objects referenced by the procedure. This check is done during the execution of the procedure. When a procedure is executed, parameters can be passed between the calling program and the stored procedure.