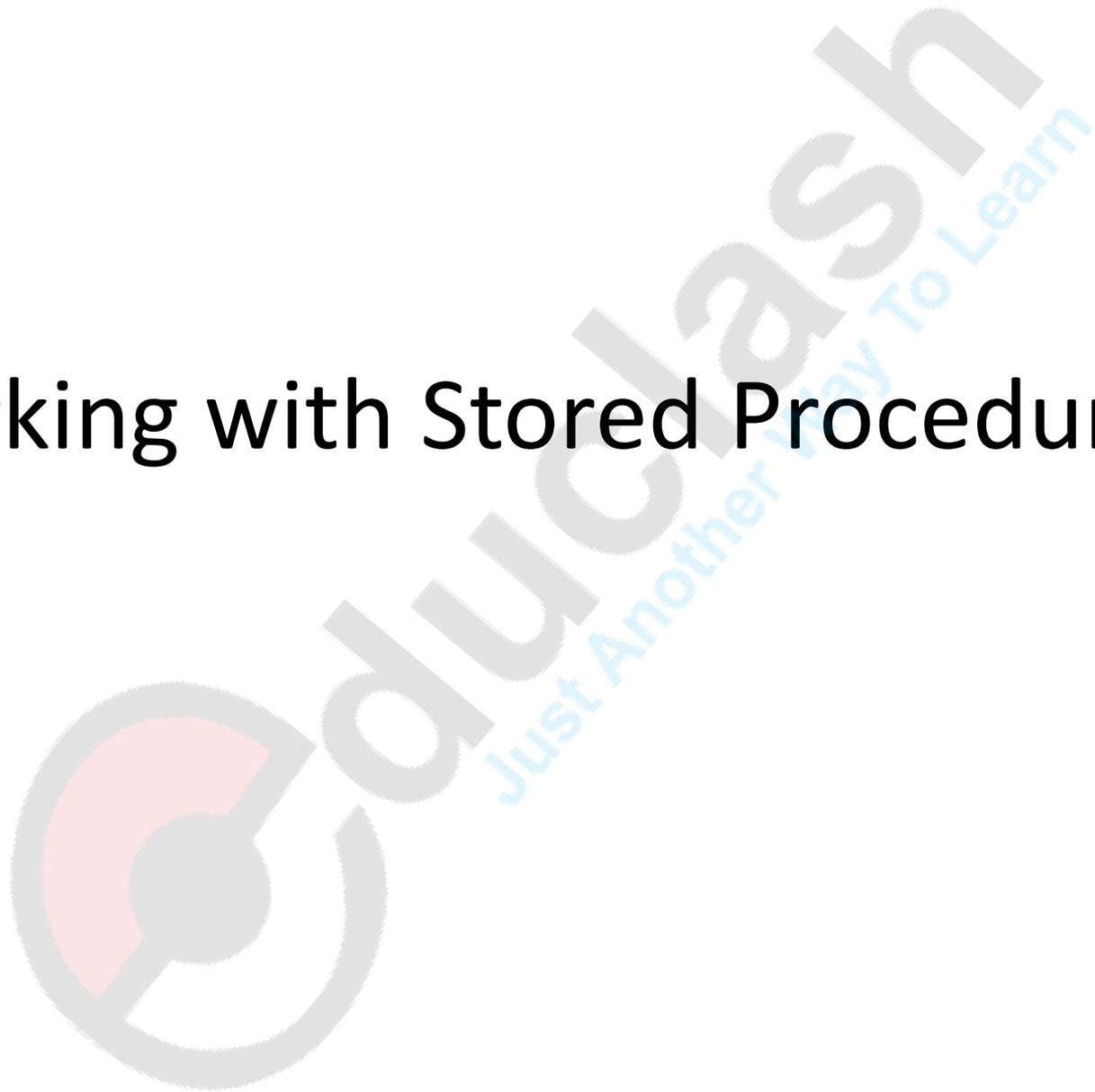


Working with Stored Procedures



Stored Procedures

- Stored Procedures are set of precompiled SQL statements stored in SQL Server database.
- Stored Procedure provides the facility to:
 - Accept input parameters and return multiple values in the form of output parameters to the calling program.
 - Contain programming statements that perform operations in the database. These include calling other procedures.
 - Return a status value to a calling program to indicate success or failure (and the reason for failure).

Example

```
CREATE PROCEDURE dbo.AddEditDeleteStudent
(
    @id int = 0, /* for update and delete */
    @f_name varchar(50),
    @l_name varchar(50),
    @dob date,
    @city varchar(50),
    @course varchar(50),
    @action varchar(10)
)
AS
if (@action = 'Insert')
insert into Student values(@f_name,@l_name,@dob,@city,@course)
    if (@action = 'Update')
update Student set FName = @f_name, LName=@l_name, DOB =@dob,
    City=@city, Course=@course where ID = @id
    if (@action = 'Delete')
delete from Student where ID = @id
select * from student
RETURN
```

Working of a stored Procedure

- The procedure is parsed into its component pieces.
- The components that reference other objects in the database (tables, views, functions, stored procedure) are checked for their existence. This process is known as **resolving**.
- Compilation continues, during which a blueprint for how to run the query is created. This blueprint is a normalized plan or a query tree.
- When the stored procedure is first executed, the query tree is read and fully optimized into a procedure plan and run.
- From next time onwards, only the query tree is executed saving a lot of time for reparsing, resolving and compiling.

Advantages

- Improved Performance - Since stored procedures are precompiled, they usually provide best performance of any type of query.
- Reduced server/client network traffic- The commands in a procedure are executed as a single batch of code. This can significantly reduce network traffic between the server and client because only the call to execute the procedure is sent across the network. Without the code encapsulation provided by a procedure, every individual line of code would have to cross the network.

Advantages(contd)

- Stronger security - Multiple users and client programs can perform operations on underlying database objects through a procedure, even if the users and programs do not have direct permissions on those underlying objects. The procedure controls what processes and activities are performed and protects the underlying database objects. This eliminates the requirement to grant permissions at the individual object level and simplifies the security layers.

When calling a procedure over the network, only the call to execute the procedure is visible. Therefore, malicious users cannot see table and database object names, embed Transact-SQL statements of their own, or search for critical data.

Advantages(contd)

- The EXECUTE AS clause can be specified in the CREATE PROCEDURE statement to enable impersonating another user, or enable users or applications to perform certain database activities without needing direct permissions on the underlying objects and commands.
- Using procedure parameters helps guard against SQL injection attacks. Since parameter input is treated as a literal value and not as executable code, it is more difficult for an attacker to insert a command into the Transact-SQL statement(s) inside the procedure and compromise security.

Advantages(contd)

- Reuse of code - The code for any repetitious database operation is the perfect candidate for encapsulation in procedures. This eliminates needless rewrites of the same code, decreases code inconsistency, and allows the code to be accessed and executed by any user or application possessing the necessary permissions.
- Easier maintenance - When client applications call procedures and keep database operations in the data tier, only the procedures must be updated for any changes in the underlying database. The application tier remains separate and does not have to know how about any changes to database layouts, relationships, or processes.

Types of Stored Procedure

- **User Defined** - Stored Procedures created by users and stored in the database.
- **System** - Many administrative activities in SQL Server are performed through a special kind of procedure known as a system stored procedure.

For eg. `sp_rename` is a commonly used system stored procedure which can be used to rename any database object like tables etc.