Automation Testing

JDBC Stored Procedures with Exceptions

simplilearn

Learning Objectives

By the end of this lesson, you will be able to:

- Illustrate stored procedures
- List the types of stored procedures
- Identify the exceptions in stored procedures

Stored Procedures

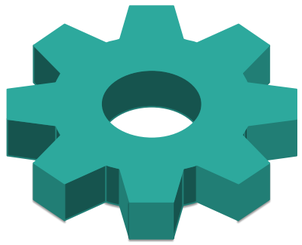
Stored Procedures

A stored procedure is a logical unit made up of SQL statements to accomplish a certain activity. It is used to encapsulate a set of actions or queries to run on a database server.

They can include any combination of input or output parameters and can be compiled and performed with varying parameters and outputs.

Parameter Modes

There are three types of parameters in a stored procedure:



IN

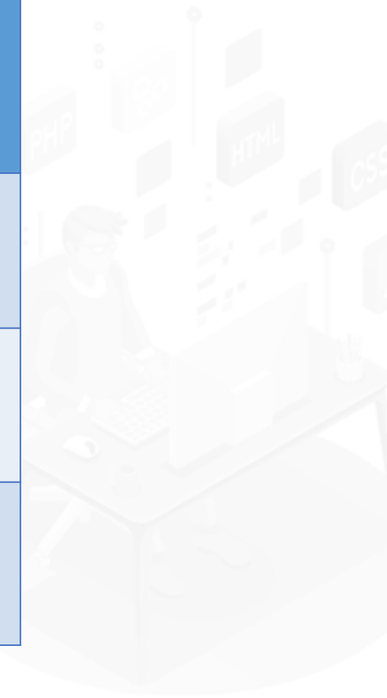
OUT

INOUT

The PreparedStatement object only uses the IN parameter, and the CallableStatement object can use all of them.

Parameter Modes

Parameter	Description
IN	A parameter whose value is unknown when the SQL statement is created.
OUT	A parameter whose value is supplied by the SQL statement it returns.
INOUT	A parameter that provides both input and output values.



Stored Procedures Types

Types of Stored Procedure

The stored procedures are divided into two groups based on the transactions in which they are used:

Types of Stored Procedure

Nested connections

The transaction that this type of procedure uses is the same as the transaction that the SQL statement that called it utilized.

Sample Code

```
Connection c =
DriverManager.getConnection("jdbc:default:connection");
```

Types of Stored Procedure

Non-nested connections

A fresh database connection is used in this procedure. The procedure is run in a separate transaction than the SQL that calls itself.

Sample Code

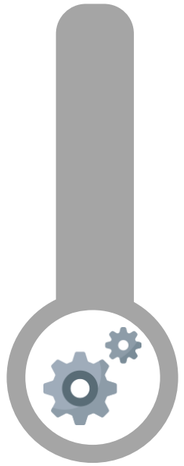
```
CREATE PROCEDURE PROC_NAME_4(OUT paramname INTEGER)
    LANGUAGE JAVA
    EXTERNAL NAME 'JavaStoredProcs.testProc4'
    PARAMETER STYLE JAVA
    READS SQL DATA;
```

Steps to Call and Execute a Stored Procedure

A stored procedure is executed using a CallableStatement object.

Prepare a CallableStatement

Register the output parameters



Set the input parameters

Run the CallableStatement and get the result sets

Prepare a CallableStatement

Preparing a callable statement

Create an object of the CallableStatement (interface) using the prepare Call() method of the connection interface.

Suppose users have a procedure name myProcedure in the database, they can prepare a callable statement as:

Sample Code

```
CallableStatement cstmt = con.prepareCall("{call
myprocedure(?,?,?)}");
```

Set the Input parameters

Setting up the Callable Statement

Users can set values to the input parameters of the procedure call using the setter methods.

Instead of index they can also pass the name of the parameter in String format

Sample Code

```
cstmt.setString(1, “Raghav”);
Cstmt.setInt(2, 3000);
Cstmt.setString(3, “Hyderabad”);
```

Run the Callable Statement

Executing the Callable Statement

Once users have created the CallableStatement( ), they can execute it using one of the execute() method.

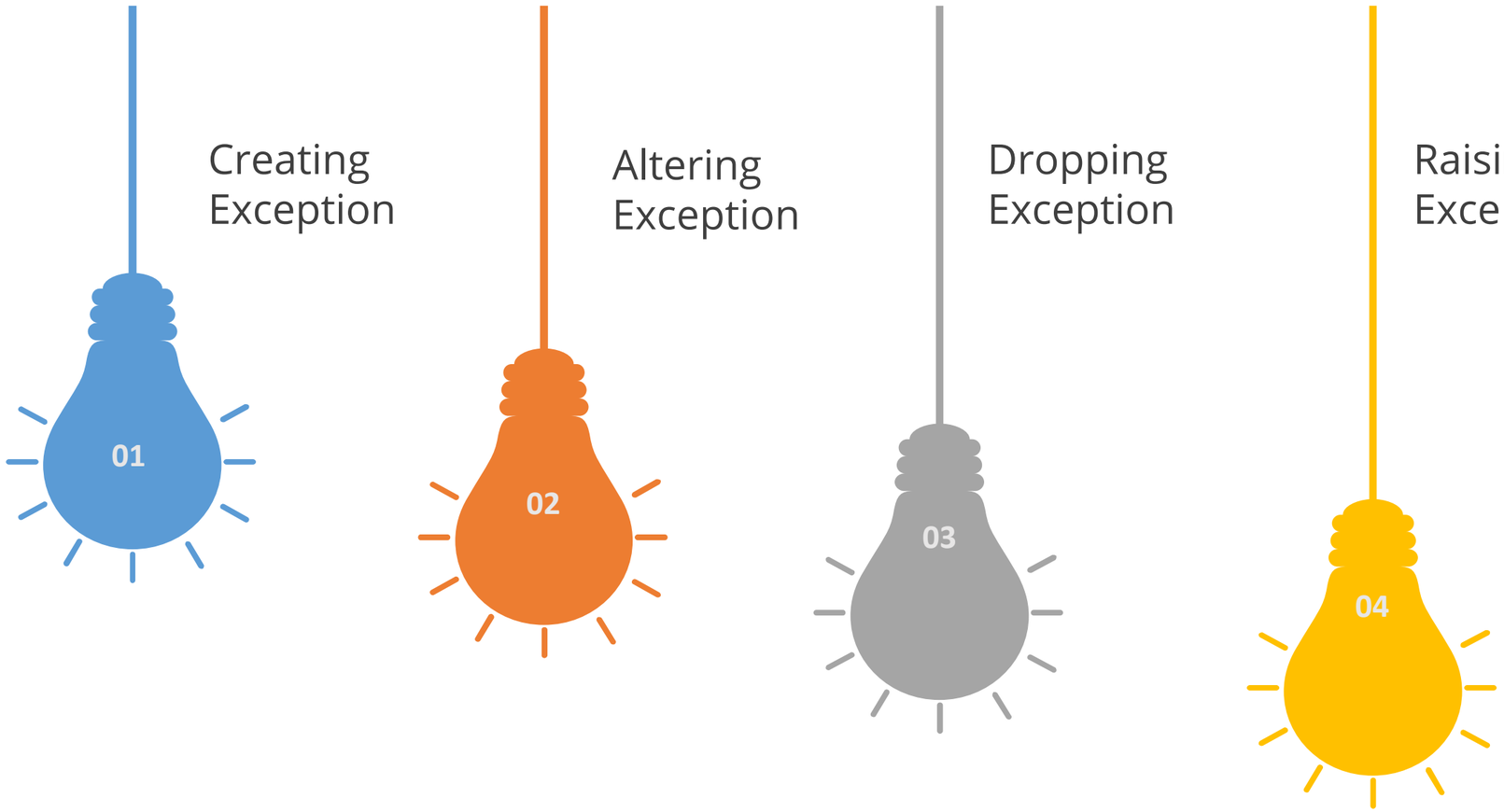
Sample Code

```
cstmt.execute();
```

Exceptions in Stored Procedure

Stored Procedure Exception

An exception is a named error message that can be raised from a stored procedure. The different types of stored procedure exception are:



Creating Exceptions

To create an exception, use the following create exception syntax:

Sample Code

```
CREATE EXCEPTION name '<message>';
```

Altering Exceptions

To alter the message returned by an exception, use the following syntax:

Sample Code

```
ALTER EXCEPTION name '<message>';
```

Dropping Exceptions

To delete an exception, the following restrictions need to be applied:

- Only the creator of an exception can drop it.
- Exceptions used in existing procedures and triggers cannot be dropped.
- Exceptions currently in use cannot be dropped.

Sample Code

```
Drop EXCEPTION name;
```

Raise an Exception

When an exception is raised it does the following :

- Terminates the procedure in which it was raised.
- Undoes any actions performed (directly or indirectly) by the procedure.
- Returns an error message to the calling application.
- Returns an error message to the calling application.
- Where <name> is the name of an exception that already exists in the database.

Sample Code

```
EXCEPTION name;
```

Key Takeaways

- A stored procedure is a prepared SQL code that can be reused.
- A stored procedure supports security through data access control because end users may try to change but do not write procedures.
- The procedure calls are quick and efficient as stored procedures are compiled once and stored in an executable form.
- The stored procedures provide an important layer of security between the user interface and the database.