**Callable Statement:**

The **CallableStatement** of JDBC API is used to call a stored procedure from Java Program.

**Stored procedure** is a set of SQL statements to be executed to perform a specific task on a database. Stored procedures are beneficial when we are dealing with multiple tables with complex scenario and rather than sending multiple queries to the database,

**JDBC CallableStatement – Stored Procedure IN parameter example**:

Code snippets to show you how to call a Oracle stored procedure via JDBC CallableStatement, and how to pass IN parameters from Java to stored procedure.

//insertDBUSER is stored procedure

String insertStoreProc = "{call insertDBUSER(?,?,?,?)}";

callableStatement = dbConnection.prepareCall(insertStoreProc);

callableStatement.setInt(1, 1000);

callableStatement.setString(2, "Sanjoy");

callableStatement.setString(3, "Network Admin");

callableStatement.setDate(4, getCurrentDate());

callableStatement.executeUpdate();

## **Stored Procedure:**

A stored procedure in Oracle database. Later, calls it via JDBC.

CREATE OR REPLACE PROCEDURE insertDBUSER(

p\_userid IN DBUSER.USER\_ID%TYPE,

p\_username IN DBUSER.USERNAME%TYPE,

p\_createdby IN DBUSER.CREATED\_BY%TYPE,

p\_date IN DBUSER.CREATED\_DATE%TYPE)

IS

BEGIN

INSERT INTO DBUSER ("USER\_ID", "USERNAME", "CREATED\_BY", "CREATED\_DATE")

VALUES (p\_userid, p\_username,p\_createdby, p\_date);

COMMIT;

END;

/

**JDBC CallableStatement – Stored Procedure OUT parameter example:**

 **Code snippets**:

//getDBUSERByUserId is a stored procedure

String getDBUSERByUserIdSql = "{call getDBUSERByUserId(?,?,?,?)}";

callableStatement = dbConnection.prepareCall(getDBUSERByUserIdSql);

callableStatement.setInt(1, 10);

callableStatement.registerOutParameter(2, java.sql.Types.VARCHAR);

callableStatement.registerOutParameter(3, java.sql.Types.VARCHAR);

callableStatement.registerOutParameter(4, java.sql.Types.DATE);

// execute getDBUSERByUserId store procedure

callableStatement.executeUpdate();

String userName = callableStatement.getString(2);

String createdBy = callableStatement.getString(3);

Date createdDate = callableStatement.getDate(4);

## **Stored Procedure**

A stored procedure in Oracle database, with IN and OUT parameters. Later, calls it via JDBC.

CREATE OR REPLACE PROCEDURE getDBUSERByUserId(

p\_userid IN DBUSER.USER\_ID%TYPE,

o\_username OUT DBUSER.USERNAME%TYPE,

o\_createdby OUT DBUSER.CREATED\_BY%TYPE,

o\_date OUT DBUSER.CREATED\_DATE%TYPE)

IS

BEGIN

SELECT USERNAME , CREATED\_BY, CREATED\_DATE

INTO o\_username, o\_createdby, o\_date

from DBUSER WHERE USER\_ID = p\_userid;

END;

/

**JDBC CallableStatement – Stored Procedure CURSOR example:**

//getDBUSERCursor is a stored procedure

String getDBUSERCursorSql = "{call getDBUSERCursor(?,?)}";

callableStatement = dbConnection.prepareCall(getDBUSERCursorSql);

callableStatement.setString(1, "Sanjoy");

callableStatement.registerOutParameter(2, OracleTypes.CURSOR);

// execute getDBUSERCursor store procedure

callableStatement.executeUpdate();

// get cursor and cast it to ResultSet

rs = (ResultSet) callableStatement.getObject(2);

// loop it like normal

while (rs.next()) {

String userid = rs.getString("USER\_ID");

String userName = rs.getString("USERNAME");

}

## **Stored Procedure**

A Oracle stored procedure, with one IN and one OUT CURSOR parameter. Later, calls it via JDBC.

CREATE OR REPLACE PROCEDURE getDBUSERCursor(

p\_username IN DBUSER.USERNAME%TYPE,

c\_dbuser OUT SYS\_REFCURSOR)

IS

BEGIN

OPEN c\_dbuser FOR

SELECT \* FROM DBUSER WHERE USERNAME LIKE p\_username || '%';

END;

/