**The JDBC API** is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC API makes it possible to do three things:

* Establish a connection with a database or access any tabular data source
* Send SQL statements
* Process the results

**Connection pooling** - a connection pool is a cache of database connections that is maintained in memory, so that the connections may be reused

**Statement** interface

- Use the for general-purpose access to your database.

- Useful when you are using static SQL statements at runtime.

- The Statement interface cannot accept parameters.

**PreparedStatment** interface

extends the Statement interface

- more efficient (in the context of multiple executions) because the SQL statement that is sent gets pre-compiled (i.e. a query plan is prepared) in the DBMS.

- To safely provide values to the SQL parameters, through a range of setter methods (i.e. setInt(int,int), setString(int,String), etc.).

- Use the when you plan to use the SQL statements many times.

- The PreparedStatement interface accepts input parameters at runtime.

**CallableStatement** interface

- extends the PreparedStatement interface.

- used for executing the SQL stored procedures.

- the execution of stored procedures does not have to be DBMS-specific.

- the CallableStatement interface can also accept runtime input parameters.

- the output parameters need to be explicitly defined through the corresponding registerOutParameter() methods

See my Java Program examples for below

CALL STRORED PROCEDURE using IN Parameter

CALL STORED PROCEDURE using IN and OUT Parameters

Three types of parameters exist: IN, OUT, and INOUT.

The PreparedStatement object only uses the IN parameter.

The CallableStatement object can use all the three.

Here are the definitions of each −

Parameter Description

IN A parameter whose value is unknown when the SQL statement is created. You bind values to IN parameters with the setXXX() methods.

OUT A parameter whose value is supplied by the SQL statement it returns. You retrieve values from theOUT parameters with the getXXX() methods.

INOUT A parameter that provides both input and output values. You bind variables with the setXXX() methods and retrieve values with the getXXX() methods.