Note: This file will be updated in a few days. Will add more content.

**DriverManager**

* Manages a list of Driver objects that are registered with it.
* Used to establish a connection with the database.

***Methods:-***

* public static [Connection](https://docs.oracle.com/javase/7/docs/api/java/sql/Connection.html) getConnection([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) url,

[String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) user,

[String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) password)

throws [SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)

Description:

Attempts to establish a connection to the given database URL. The DriverManager attempts to select an appropriate driver from the set of registered JDBC drivers.

Parameters:

url - a database url of the form jdbc:subprotocol:subname

user - the database user on whose behalf the connection is being made

password - the user's password

Returns:

a connection to the URL

Throws:

[SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html) - if a database access error occurs

**Connection**

* Represents a session between our application and a database server.
* Provides methods to handle a database connection.

***Methods:-***

* [Statement](https://docs.oracle.com/javase/7/docs/api/java/sql/Statement.html) createStatement()

throws [SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)

Description:

Creates a Statement object for sending SQL statements to the database. SQL statements without parameters are normally executed using Statement objects. If the same SQL statement is executed many times, it may be more efficient to use a PreparedStatement object.

**Returns:**

a new default Statement object

**Throws:**

[SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed connection

* [PreparedStatement](https://docs.oracle.com/javase/7/docs/api/java/sql/PreparedStatement.html) prepareStatement([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) sql)

throws [SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)

Description:

Creates a PreparedStatement object for sending parameterized SQL statements to the database.

A SQL statement with or without IN parameters can be pre-compiled and stored in a PreparedStatement object. This object can then be used to efficiently execute this statement multiple times.

**Parameters:**

sql - an SQL statement that may contain one or more '?' IN parameter placeholders

**Returns:**

a new default PreparedStatement object containing the pre-compiled SQL statement

**Throws:**

[SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed connection

* [CallableStatement](https://docs.oracle.com/javase/7/docs/api/java/sql/CallableStatement.html) prepareCall([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) sql)

throws [SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)

Description:

Creates a CallableStatement object for calling database stored procedures. The CallableStatement object provides methods for setting up its IN and OUT parameters, and methods for executing the call to a stored procedure.

**Parameters:**

sql - an SQL statement that may contain one or more '?' parameter placeholders. Typically this statement is specified using JDBC call escape syntax.

**Returns:**

a new default CallableStatement object containing the pre-compiled SQL statement

**Throws:**

[SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html) - if a database access error occurs or this method is called on a closed connection

**Statement**

* Used to execute SQL queries and return results.

***Methods:-***

* [ResultSet](https://docs.oracle.com/javase/7/docs/api/java/sql/ResultSet.html) executeQuery([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) sql)

throws [SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)

Description:

Executes the given SQL statement, which returns a single ResultSet object.

**Note:**This method cannot be called on a PreparedStatement or CallableStatement.

Parameters:

sql - an SQL statement to be sent to the database, typically a static SQL SELECT statement

Returns:

a ResultSet object that contains the data produced by the given query; never null

Throws:

[*SQLException*](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)*-* if a database access error occurs, this method is called on a closed Statement, the given SQL statement produces anything other than a single ResultSet object, the method is called on a PreparedStatement or CallableStatement

[*SQLTimeoutException*](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLTimeoutException.html)- when the driver has determined that the timeout value that was specified by the setQueryTimeout method has been exceeded and has at least attempted to cancel the currently running Statement

**PreparedStatement**

* Also used to execute SQL queries and return results. It is more efficient compared to Statement.
* PreparedStatement object is used to store a precompiled SQL statement.

***Methods:-***

* int executeUpdate()

throws [SQLException](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)

Description:

Executes the SQL statement in this PreparedStatement object, which must be an SQL Data Manipulation Language (DML) statement, such as INSERT, UPDATE or DELETE; or an SQL statement that returns nothing, such as a DDL statement.

Returns:

either (1) the row count for SQL Data Manipulation Language (DML) statements or (2) 0 for SQL statements that return nothing

Throws:

[*SQLException*](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)- if a database access error occurs; this method is called on a closed PreparedStatement or the SQL statement returns a ResultSet object

[*SQLTimeoutException*](https://docs.oracle.com/javase/7/docs/api/java/sql/SQLTimeoutException.html)- when the driver has determined that the timeout value that was specified by the setQueryTimeout method has been exceeded and has at least attempted to cancel the currently running Statement

**ResultSet**

* Used to represent data retrieved from a database in a tabular format.
* Maintains a cursor pointing to its current row. Initially the cursor is positioned before the first row. When we call next() method on the ResultSet object, the cursor will be moved to the next row.

**CallableStatement**

* Used to execute SQL stored procedures and functions.

[Reference:- Java Docs by Oracle]