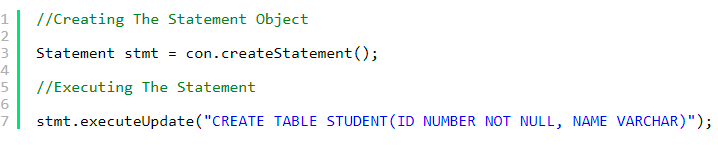
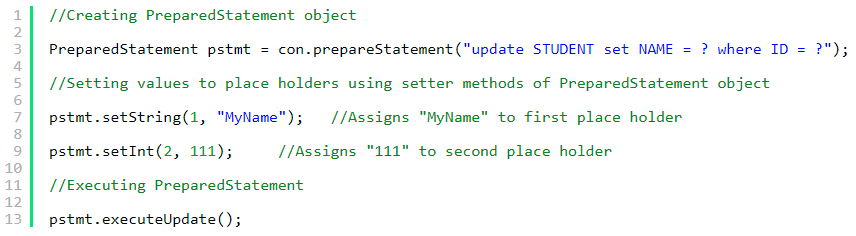
## **Statement**

Statement interface is used to execute normal SQL queries. You can’t pass the parameters to SQL query at run time using this interface. This interface is preferred over other two interfaces if you are executing a particular SQL query only once. The performance of this interface is also very less compared to other two interfaces. In most of time, Statement interface is used for DDL statements like **CREATE**, **ALTER**, **DROP** etc. For example,



## **Prepared Statement**

PreparedStatement is used to execute dynamic or parameterized SQL queries. PreparedStatement extends Statement interface. You can pass the parameters to SQL query at run time using this interface. It is recommended to use PreparedStatement if you are executing a particular SQL query multiple times. It gives better performance than Statement interface. Because, PreparedStatement are precompiled and the query plan is created only once irrespective of how many times you are executing that query. This will save lots of time.



## ****Callable Statement****

CallableStatement is used to execute the stored procedures. CallableStatement extends PreparedStatement. Using CallableStatement, you can pass 3 types of parameters to stored procedures. They are : ***IN*** – used to pass the values to stored procedure, **OUT** – used to hold the result returned by the stored procedure and **IN OUT** – acts as both IN and OUT parameter. Before calling the stored procedure, you must register OUT parameters using **registerOutParameter()** method of CallableStatement. The performance of this interface is higher than the other two interfaces. Because, it calls the stored procedures which are already compiled and stored in the database server.

