首先，需要知道使用Java开发任何数据库应用程序都需要4个主要接口：Driver，Connection，Statement和ResultSet。这些接口定义了使用SQL访问数据库的一般架构。访问数据库的Java程序主要有5个步骤，分别是：

**1）加载驱动程序。**

**2）建立连接。**

**3）创建语句。**

**4）执行语句。**

**5）处理ResultSet。**

**6）关闭连接。**

那么就简单说明这5个步骤如何实现：

**1）加载驱动程序**

在连接到数据库之前，需要加载一个合适的驱动程序。

**Class.forname("com.mysql.cj.jdbc.Driver");**

驱动程序是一个实现接口java.sql.Driver的具体类。因为我用的数据库是MySQL，所以在括号里填com.mysql.cj.jdbc.Driver，像其他的数据库，会有不同的程序驱动类。

**2）建立连接**

连接到数据库需要使用到DriverManager类中的静态方法getConnection(databaseURL)，其中databaseURL是数据库在Internet上的唯一标识符。

**Connection connection=DriverManager.getConnection("jdbc:mysql://localhost/Contacts?serverTimezone=UTC", "root", "Cc229654512");**

**3）创建语句**

Statement的对象可以为数据库传输SQL语句用于执行，并把运行结果返回程序。一旦创建Connection对象，就可以创建执行SQL语句的语句。

**Statement statement=connection.createStatement();**

**4）执行语句**

可以使用方法executeUpdate(String sql)来执行数据定义语言或者更新语句，可以使用executeQuery(String sql)来执行SQL查询语句。结果在ResultSet中返回。

**ResultSet resultSet=statement.executeQuery("select Name, PhoneNumber, Email, QQ, Note from Contacts");**

**5）处理ResultSet**

结果集ResultSet可以获得表的当前行，可以使用next方法移动行，可以使用各种get方法获取当前行的值。

**while(resultSet.next())**

**{  
        System.out.println(resultSet.getString(1)+"\t"+resultSet.getString(2)+"\t"+resultSet.getString(3));  
}**

6)关闭连接  
        rs.close();  
        state.close();  
        conn.close();

如下面所示代码，实现插入操作

1. **public**  **void** test\_insert()
2. {
3. String driver="oracle.jdbc.driver.OracleDriver";
4. String url="jdbc:oracle:thin:@127.0.0.1:1521:orcl";//orcl为sid
5. String user="briup";
6. String password="briup";
7. Connection conn=**null**;
8. Statement stat=**null**;
9. **try** {
10. //1、注册驱动
11. Class.forName(driver);
12. //2、获取连接
13. conn= DriverManager.getConnection(url, user, password);
14. //System.out.println(conn);
15. //3、创建statement对象
16. stat=conn.createStatement();
17. //4、执行sql语句
18. String sql="insert into lover values(5,'suxingxing',to\_date('21-9-2016','dd-mm-yyyy'))";
19. stat.execute(sql);
20. //System.out.println(stat.execute(sql));
21. //5、处理结果集,如果有的话就处理，没有就不用处理，当然insert语句就不用处理了
22. } **catch** (Exception e) {
23. e.printStackTrace();
24. }
25. **finally**{
26. //6、关闭资源
27. **try** {
28. **if**(stat!=**null**)stat.close();
29. } **catch** (SQLException e) {
30. e.printStackTrace();
31. }
32. **try** {
33. **if**(conn!=**null**)conn.close();
34. } **catch** (SQLException e) {
35. e.printStackTrace();
36. }
37. }
38. }