

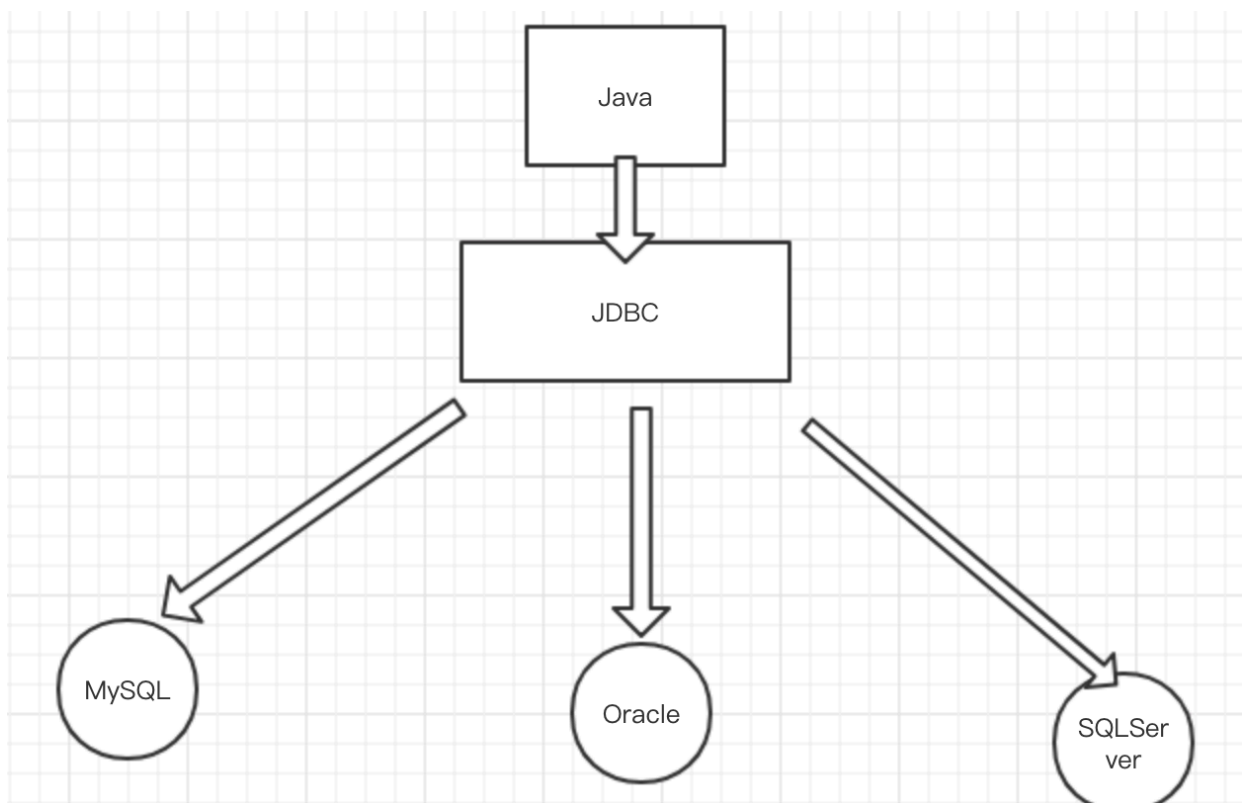
JDBC

Java Database Connectivity是一个独立于特定数据库管理系统，通用的SQL数据库存取和操作的公共接口。

定义了一组标准，为访问不同的数据库提供了统一的途径。

可以用Java程序直接调用JDBC的相关接口完成数据持久化的工作，Java程序只需要关注JDBC的接口，面向JDBC接口编程，不需要关注底层数据库，无论是MySQL，Oracle，SQLServer等，Java程序调用JDBC的代码不需要改变，只需要通过JDBC的不同配置，不同的驱动去连接到不同的底层数据库。

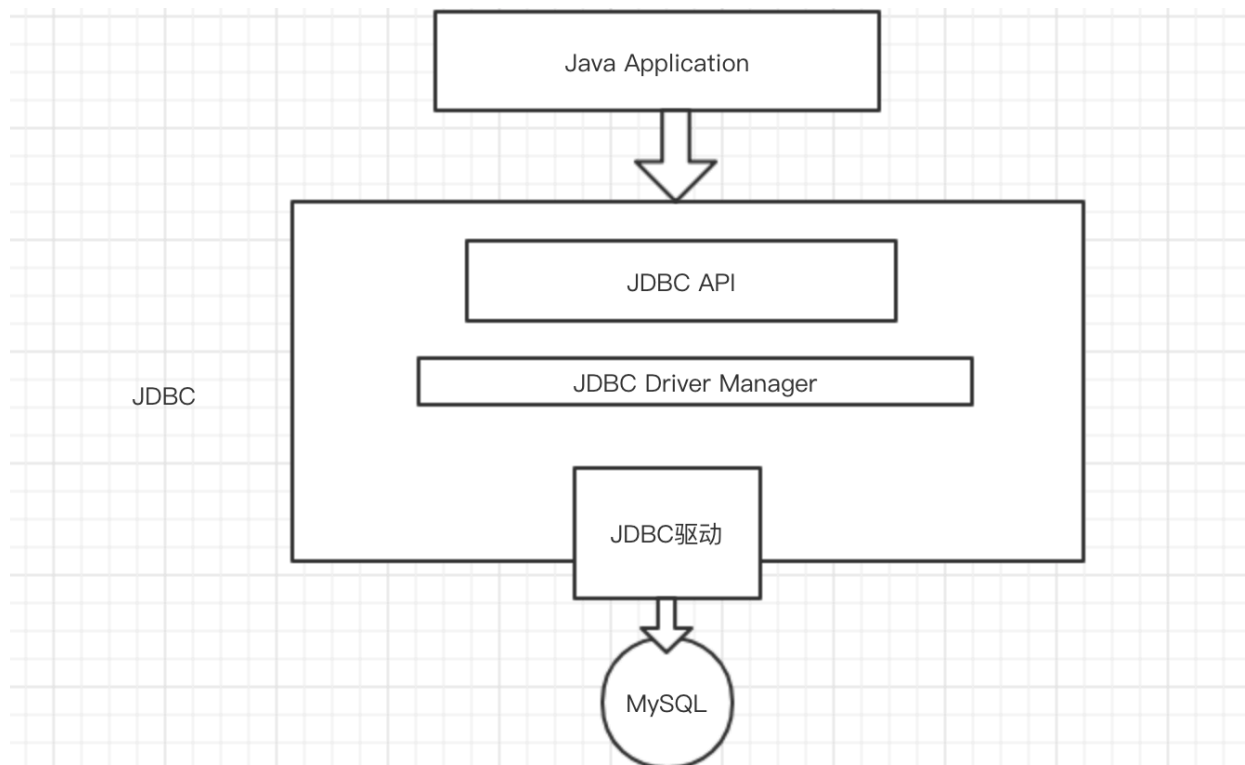
JDBC将底层数据库的操作进行了屏蔽，使得Java程序只需要在业务层/应用层，调用同一套接口进行编程即可。



JDBC的体系结构

JDBC接口包括两个层面：

- 面向应用的API（Application Interface），供开发人员调用。
- 面向数据库的API，供数据库开发厂商调用，来开发数据库驱动。



JDBC API是由Java的开发者来提供的，内容包括供开发者调用的各种接口，java.sql和javax.sql包中，DriverManager类、Connection接口、Statement接口、ResultSet接口。

DirverManager类的作用是管理各种不同的JDBC驱动。

Connection接口负责连接数据库。

Statement接口负责执行SQL语句。

ResultSet接口负责解析结果集（从数据库中查询到的结果）。

JDBC驱动是由数据库厂商来提供的，负责让Java程序去连接各种厂商对应的数据库。

使用JDBC的步骤：

- 加载数据库驱动，Java程序和MySQL的桥梁。
- 获取Connection，表示一次连接。
- 通过Statement来执行SQL语句，进而对数据库进行管理。
- ResultSet保存Statement执行后所产生的结果（只针对查询操作）。

```
package com.southwind.jdbc;

import java.sql.*;

public class Test {
    public static void main(String[] args) {
        Connection connection = null;
```

```

Statement statement = null;
try {
    //1.加载驱动
    Class.forName("com.mysql.cj.jdbc.Driver");
    //2.获取连接
    String url = "jdbc:mysql://localhost:3306/demo";
    String user = "root";
    String password = "19900310";
    connection = DriverManager.getConnection(url,user,password);
    //新增数据
    //3.定义SQL语句
    //      String sql = "insert into user(name,age,money) values ('张三',22,500)";
    //      //4.执行SQL
    //      Statement statement = connection.createStatement();
    //      int row = statement.executeUpdate(sql);
    //      if(row == 1){
    //          System.out.println("添加成功! ");
    //      }

    //修改数据
    //      String sql = "update user set name = '李四',age = 23,money = 1000 where id = 10";
    //      Statement statement = connection.createStatement();
    //      int row = statement.executeUpdate(sql);
    //      if(row == 1){
    //          System.out.println("修改成功! ");
    //      }

    //删除数据
    //      String sql = "delete from user where id = 10";
    //      Statement statement = connection.createStatement();
    //      int row = statement.executeUpdate(sql);
    //      if(row == 1){
    //          System.out.println("删除成功! ");
    //      }

    //查询
    //      String sql = "select * from user where id = 11";
    //      Statement statement = connection.createStatement();
    //      ResultSet resultSet = statement.executeQuery(sql);
    //      while(resultSet.next()){
    //          int id = resultSet.getInt(1);
    //          String name = resultSet.getString(2);
    //          int age = resultSet.getInt(3);
    //          int money = resultSet.getInt(4);
    //          System.out.println(id+"-"+name+"-"+age+"-"+money);
    //          id = resultSet.getInt("id");
    //          name = resultSet.getString("name");

```

```

//            age = resultSet.getInt("age");
//            money = resultSet.getInt("money");
//            System.out.println(id+"-"+name+"-"+age+"-"+money);
//        }

//            String sql = "select * from user where id = 11";
//            String sql = "insert into user(name,age,money) values('王
五',25,1000)";
//            String sql = "update user set name = '小明' where id = 13";
String sql = "delete from user where id = 13";
statement = connection.createStatement();
boolean flag = statement.execute(sql);
System.out.println(flag);
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException e){
    e.printStackTrace();
} finally {
    try {
        connection.close();
        statement.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}
}

```

执行SQL通过Statement对象的方法来完成的

- int executeUpdate(String sql) 适用于新增，修改，删除，返回数据的操作行数。
- ResultSet executeQuery(String sql) 适用于查询，并且将查询结果封装成一个ResultSet对象，需要对该对象进行解析。
- boolean execute(String sql)适用于新增，修改，删除，查询，true表示返回结果是ResultSet对象，false表示返回结果不是ResultSet对象。

JDBCTools

```

package com.southwind.jdbc;

import com.southwind.entity.User;

import java.sql.*;

public class JDBCTools {
    private String driver = "com.mysql.cj.jdbc.Driver";
}

```

```

private String url = "jdbc:mysql://localhost:3306/demo";
private String username = "root";
private String password = "19900310";
private Connection connection = null;

public JDBCTools(){
    try {
        Class.forName(driver);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}

/**
 * 新增
 */
public int save(User user){
    int row = 0;
    Statement statement = null;
    try {
        connection =
DriverManager.getConnection(url,username,password);
        String sql = "insert into user(name,age,money)
values('"+user.getName()+"','"+user.getAge()+"','"+user.getMoney()+"')";
        statement = connection.createStatement();
        row = statement.executeUpdate(sql);
    } catch (SQLException e){
        e.printStackTrace();
    }finally {
        try {
            connection.close();
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return row;
}

/**
 * 修改
 */
public int update(User user){
    int row = 0;
    Statement statement = null;
    try{
        connection =
DriverManager.getConnection(url,username,password);

```

```

        String sql = "update user set name = '"+user.getName()+"',age =
"+user.getAge()+" ,money = "+user.getMoney()+" where id = "+user.getId();
        statement = connection.createStatement();
        row = statement.executeUpdate(sql);
    }catch (SQLException e){
        e.printStackTrace();
    }finally {
        try {
            connection.close();
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return row;
}

/**
 * 查询
 */
public User find(int id){
    Statement statement = null;
    User user = null;
    ResultSet resultSet = null;
    try{
        connection =
DriverManager.getConnection(url,username,password);
        String sql = "select * from user where id = "+id;
        statement = connection.createStatement();
        resultSet = statement.executeQuery(sql);
        if(resultSet.next()){
            int id2 = resultSet.getInt(1);
            String name = resultSet.getString(2);
            int age = resultSet.getInt(3);
            int money = resultSet.getInt(4);
            user = new User(id2,name,age,money);
        }
    }catch (SQLException e){
        e.printStackTrace();
    }finally {
        try {
            connection.close();
            statement.close();
            resultSet.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return user;
}

```

```

    }

    /**
     * 删除
     */
    public int delete(int id){
        int row = 0;
        Statement statement = null;
        try{
            connection =
DriverManager.getConnection(url,username,password);
            String sql = "delete from user where id = "+id;
            statement = connection.createStatement();
            row = statement.executeUpdate(sql);
        }catch (SQLException e){
            e.printStackTrace();
        }finally {
            try {
                connection.close();
                statement.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return row;
    }
}

```