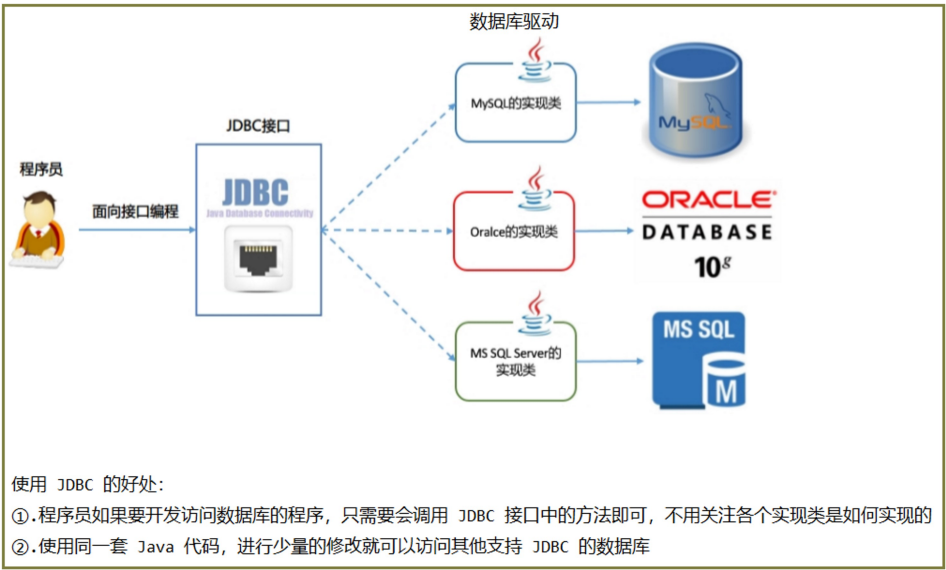


通过命令行、第三方客户端工具与 MySQL 数据库软件的通信，可以访问、操作 MySQL 数据库中的数据，而 Java 程序与数据库之间的交互是通过 JDBC 来实现的。

JDBC(Java DataBase Connectivity)：是官方定义的一套操作、访问所有关系型数据库的标准规范，即一系列的接口。真正怎么操作数据库还需要具体的实现类，各个数据库厂商会根据自家数据库的通信格式去实现这套接口并提供具体的实现类，即：数据库驱动。所以我们只需要会调用 JDBC 接口中的方法即可，真正执行的代码是数据库驱动中的实现类。(面向接口编程：以多态的形式调用 JDBC 接口中的方法，真正调用的还是数据库驱动中的实现类，但不用关心各个实现类是如何实现的，只需要会调用接口中的方法即可)



一.JDBC 的基本使用

```
// JDBC快速入门程序:
public class DemoMain {
    public static void main(String[] args) throws ClassNotFoundException, SQLException {

        /*
        1. 导入数据库驱动jar包: (第三方类库的使用方法与其类似)
            ①. 复制mysql-connector-java-5.1.37-bin.jar包到lib目录下
            ②. 右键 --> Add As Library 将其添加成库, 就可以使用jar包里面的内容了
        */

        // 2. 注册数据库驱动
        Class.forName("com.mysql.jdbc.Driver");

        // 3. 获取数据库连接对象
        Connection conn = DriverManager.getConnection(url: "jdbc:mysql://localhost:3306/test", user: "root", password: "root");

        // 4. 定义sql语句
        String sql = "update account set balance = 2000 where id = 1";

        // 5. 获取执行sql语句的对象
        Statement stmt = conn.createStatement();

        // 6. 执行sql语句并接收返回结果
        int count = stmt.executeUpdate(sql);

        // 7. 处理sql语句的返回结果
        System.out.println(count);

        // 8. 释放资源
        stmt.close();
        conn.close();
    }
}
```

开发一个 JDBC 程序的基本步骤如上所示, 下面我们就对 JDBC 中的接口、类进行详解。

## 1.DriverManager 类: 驱动管理对象

- 注册数据库驱动:
- 获取数据库连接对象:

```
static Connection getConnection(String url, String user, String password) throws SQLException
```

调用 DriverManager 类中的静态方法 getConnection, 可以返回一个 Connection 接口的实现类对象, 即: 数据库连接对象

## 2.Connection 接口: 数据库连接对象