**一、JDBC相关概念介绍**

**1.1、数据库驱动**

　　这里的驱动的概念和平时听到的那种驱动的概念是一样的，比如平时购买的声卡，网卡直接插到计算机上面是不能用的，必须要安装相应的驱动程序之后才能够使用声卡和网卡，同样道理，我们安装好数据库之后，我们的应用程序也是不能直接使用数据库的，必须要通过相应的数据库驱动程序，通过驱动程序去和数据库打交道

### 1.2、JDBC介绍

　　SUN公司为了简化、统一对数据库的操作，定义了一套Java操作数据库的规范（接口），称之为JDBC。这套接口由数据库厂商去实现，这样，开发人员只需要学习jdbc接口，并通过jdbc加载具体的驱动，就可以操作数据库。

JDBC全称为：Java Data Base Connectivity（java数据库连接），它主要由接口组成。  
　　组成JDBC的２个包：  
　　　java.sql  
　　　javax.sql  
　　开发JDBC应用需要以上2个包的支持外，还需要导入相应JDBC的数据库实现(即数据库驱动)。

## 二、编写JDBC程序

### 2.1、搭建实验环境

　　1、在mysql中创建一个库，并创建user表和插入表的数据。

　　SQL脚本如下：

create database jdbcStudy character set utf8 collate utf8\_general\_ci;

use jdbcStudy;

create table users(

id int primary key,

name varchar(40),

password varchar(40),

email varchar(60),

birthday date

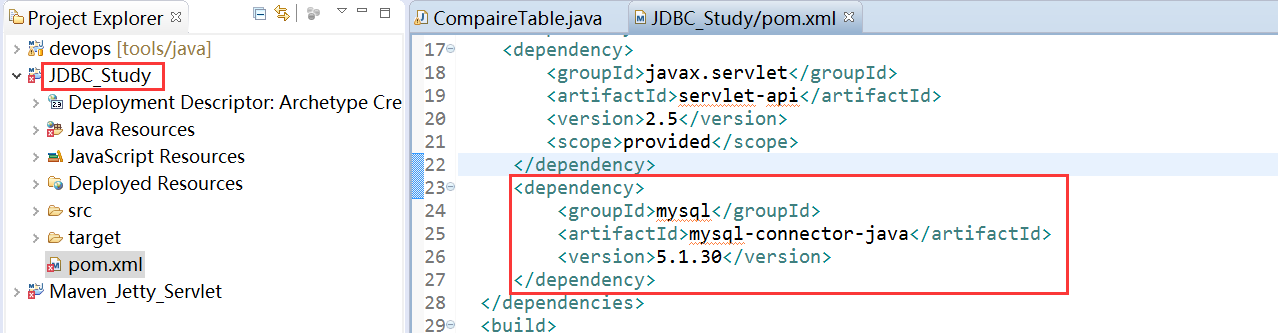
);

insert into users(id,name,password,email,birthday) values(1,'zhansan','123456','zs@sina.com','1980-12-04');

insert into users(id,name,password,email,birthday) values(2,'lisi','123456','lisi@sina.com','1981-12-04');

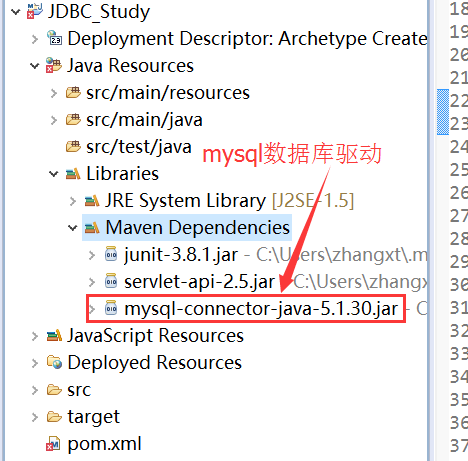
insert into users(id,name,password,email,birthday) values(3,'wangwu','123456','wangwu@sina.com','1979-12-04');

2、新建一个maven project，并在pom.xml文件中加入mysql数据库驱动的依赖，如图2-1。



2-1

对该工程进行Maven install，就会下载对应版本的mysql驱动，如图2-2



2-2

3、编写程序从user表中读取数据，并打印在命令行窗口中。

**package** com.test;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**public** **class** JdbcDemo {

**public** **static** **void** main(String[] args) **throws** Exception {

//要连接的数据库URL

String url = "jdbc:mysql://120.76.84.36:3306/jdbc\_study";

//连接的数据库时使用的用户名

String username = "root";

//连接的数据库时使用的密码

String password = "mysql!@#root";

//1.加载驱动

//DriverManager.registerDriver(new com.mysql.jdbc.Driver());不推荐使用这种方式来加载驱动

Class.*forName*("com.mysql.jdbc.Driver");//推荐使用这种方式来加载驱动

//2.获取与数据库的链接

Connection conn = DriverManager.*getConnection*(url, username, password);

//3.获取用于向数据库发送sql语句的statement

Statement st = conn.createStatement();

String sql = "select id,name,password,email,birthday from users";

//4.向数据库发sql,并获取代表结果集的resultset

ResultSet rs = st.executeQuery(sql);

//5.取出结果集的数据

**while**(rs.next()){

System.***out***.println("id=" + rs.getObject("id"));

System.***out***.println("name=" + rs.getObject("name"));

System.***out***.println("password=" + rs.getObject("password"));

System.***out***.println("email=" + rs.getObject("email"));

System.***out***.println("birthday=" + rs.getObject("birthday"));

}

//6.关闭链接，释放资源

rs.close();

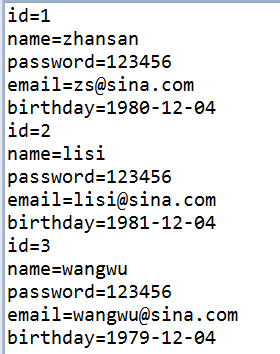
st.close();

conn.close();

}

}

运行结果如下：



### 2.2、DriverManager类讲解

　　Jdbc程序中的DriverManager用于加载驱动，并创建与数据库的链接，这个API的常用方法：

1. DriverManager.registerDriver(new Driver())
2. DriverManager.getConnection(url, user, password)，

注意：**在实际开发中并不推荐采用registerDriver方法注册驱动**。原因有二：

　　1、查看Driver的源代码可以看到，如果采用此种方式，会导致驱动程序注册两次，也就是在内存中会有两个Driver对象。

　　2、程序依赖mysql的api，脱离mysql的jar包，程序将无法编译，将来程序切换底层数据库将会非常麻烦。

推荐方式：**Class.forName("com.mysql.jdbc.Driver");**

采用此种方式不会导致驱动对象在内存中重复出现，并且采用此种方式，程序仅仅只需要一个字符串，不需要依赖具体的驱动，使程序的灵活性更高。

　　采用此种方式不会导致驱动对象在内存中重复出现，并且采用此种方式，程序仅仅只需要一个字符串，不需要依赖具体的驱动，使程序的灵活性更高。

**2.3、数据库URL讲解**

　　URL用于标识数据库的位置，通过URL地址告诉JDBC程序连接哪个数据库，URL的写法为：

　　MySql写法：jdbc:mysql://localhost:3306/sid

　　如果连接的是本地的Mysql数据库，并且连接使用的端口是3306，那么的url地址可以简写为： jdbc:mysql:///数据库

**2.4、Connection类讲解**

　　Jdbc程序中的Connection，它用于代表数据库的链接，Collection是数据库编程中最重要的一个对象，客户端与数据库所有交互都是通过connection对象完成的，这个对象的常用方法：

* createStatement()：创建向数据库发送sql的statement对象。
* prepareStatement(sql) ：创建向数据库发送预编译sql的PrepareSatement对象。
* prepareCall(sql)：创建执行存储过程的callableStatement对象。
* setAutoCommit(boolean autoCommit)：设置事务是否自动提交。
* commit() ：在链接上提交事务。
* rollback() ：在此链接上回滚事务。

**2.5、Statement类讲解**

　　Jdbc程序中的Statement对象用于向数据库发送SQL语句， Statement对象常用方法：

* executeQuery(String sql) ：用于向数据发送查询语句。
* executeUpdate(String sql)：用于向数据库发送insert、update或delete语句
* execute(String sql)：用于向数据库发送任意sql语句
* addBatch(String sql) ：把多条sql语句放到一个批处理中。
* executeBatch()：向数据库发送一批sql语句执行。

**2.6、ResultSet类讲解**

　　Jdbc程序中的ResultSet用于代表Sql语句的执行结果。Resultset封装执行结果时，采用的类似于表格的方式。ResultSet 对象维护了一个指向表格数据行的游标，初始的时候，游标在第一行之前，调用ResultSet.next() 方法，可以使游标指向具体的数据行，进行调用方法获取该行的数据。  
　　ResultSet既然用于封装执行结果的，所以该对象提供的都是用于获取数据的get方法：  
　　获取任意类型的数据  
　　　　getObject(int index)  
　　　　getObject(string columnName)  
　　获取指定类型的数据，例如：  
　　　　getString(int index)  
　　　　getString(String columnName)

　　ResultSet还提供了对结果集进行滚动的方法：

* next()：移动到下一行
* Previous()：移动到前一行
* absolute(int row)：移动到指定行
* beforeFirst()：移动resultSet的最前面。
* afterLast() ：移动到resultSet的最后面。

### 2.7、释放资源

　　Jdbc程序运行完后，切记要释放程序在运行过程中，创建的那些与数据库进行交互的对象，这些对象通常是ResultSet, Statement和Connection对象，特别是Connection对象，它是非常稀有的资源，用完后必须马上释放，如果Connection不能及时、正确的关闭，极易导致系统宕机。Connection的使用原则是尽量晚创建，尽量早的释放。  
　　为确保资源释放代码能运行，资源释放代码也一定要放在finally语句中。