

# JDBC 基础

## 一、实验介绍

### 1.1 实验内容

本实验将学习 JDBC 驱动类型以及 JDBC 连接数据库的相关知识。

### 1.2 实验知识点

- JDBC 结构
- JDBC 驱动类型
- JDBC 连接数据库

### 1.3 实验环境

- JDK1.7
- Eclipse
- Mysql 5.5.50

## 二、实验步骤

上一个实验，我们简单地介绍了一下 JDBC，同时也编写了第一个 JDBC 程序，同学们是不是对 JDBC 有了一个基础性的了解呢？今天我们将继续深入认识 JDBC，看一看它是如何实现 Java 程序与数据库交互的。

### 2.1 JDBC 结构

首先我们先来回顾一下上节实验课的内容。JDBC 全称为Java Database Connectivity，中文名称为 Java 数据库连接，它是一种用于执行 SQL 语句的 Java API,它由一组用 Java 编程语言编写的类和接口组成。JDBC 为数据库开发人员提供了一个标准的 API,使他们能够用纯 Java API 来编写数据库应用程序。

JDBC API 是 Java 开发工具包(JDK)的组成部份,由三部分组成:

- JDBC 驱动程序管理器

- JDBC 驱动程序测试工具包
- JDBC 入门教程 (/courses/110)

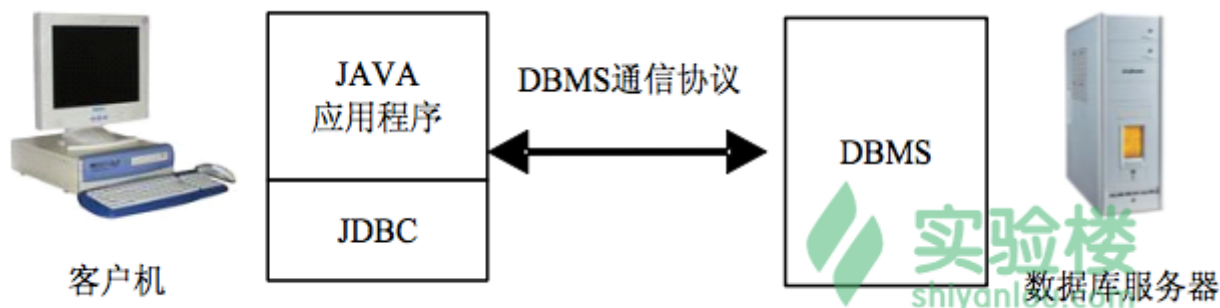
a. JDBC 驱动程序管理器是 JDBC 体系结构的支柱,其主要作用是把 Java 应用程序连接到正确的 JDBC 驱动程序上。

b. JDBC 驱动程序测试工具包为 JDBC 驱动程序的运行提供一定的可信度,只有通过 JDBC 驱动程序测试包的驱动程序才被认为是符合 JDBC 标准的。

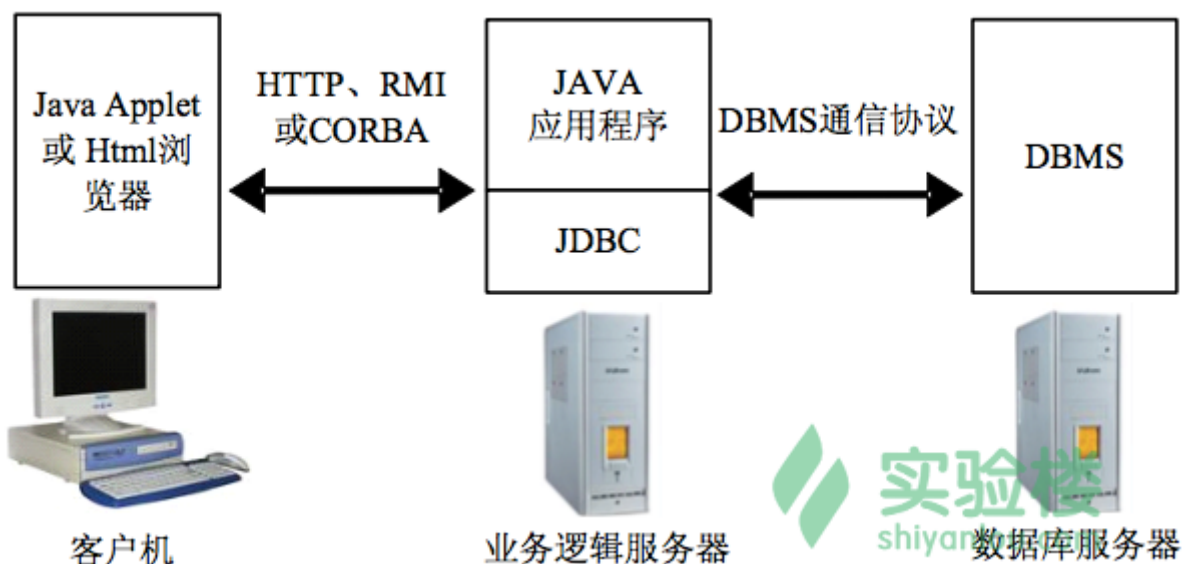
c. JDBC-ODBC 桥使 ODBC 驱动程序可被用作 JDBC 驱动程序。其目标是方便实现访问某些不常见的 DBMS (数据库管理系统),它的实现为 JDBC 的快速发展提供了一条途径。

JDBC既支持数据库访问的两层模型,也支持三层模型。

### 1、数据库访问的两层模型



### 2、数据库访问的三层模型

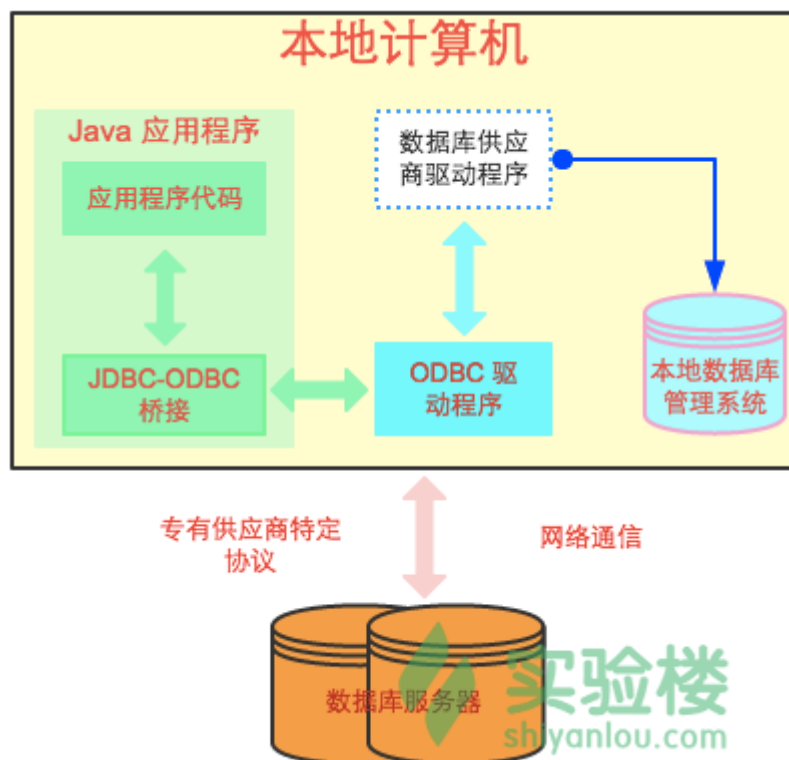


## 2.2 JDBC 驱动类型

JDBC 驱动程序实现 JDBC API 中定义的接口,用于与数据库服务器进行交互。JDBC 驱动程序可以打开数据库连接,并通过发送 SQL 或数据库命令,然后在收到结果与 Java 进行交互。

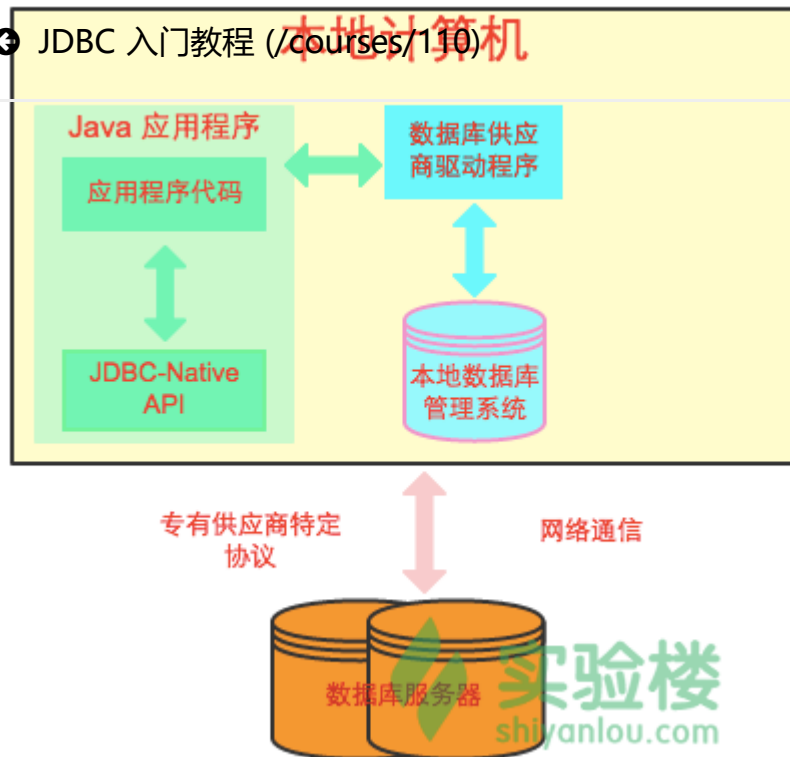
JDBC 驱动程序的实现因为各种各样的操作系统和 Java 运行在不同的硬件平台上而不同。JDBC 驱动类型可以归结为以下几类:

- JDBC-ODBC 桥接 ODBC 驱动程序: 它是将 JDBC 翻译成 ODBC,然后使用一个 ODBC 驱动程序与数据库进行通信。当 Java 刚出来时, 这是一个有用的驱动程序, 因为大多数的数据库只支持 ODBC 访问, 但现在建议使用此类型的驱动程序仅用于实验用途或在没有其他选择的情况。

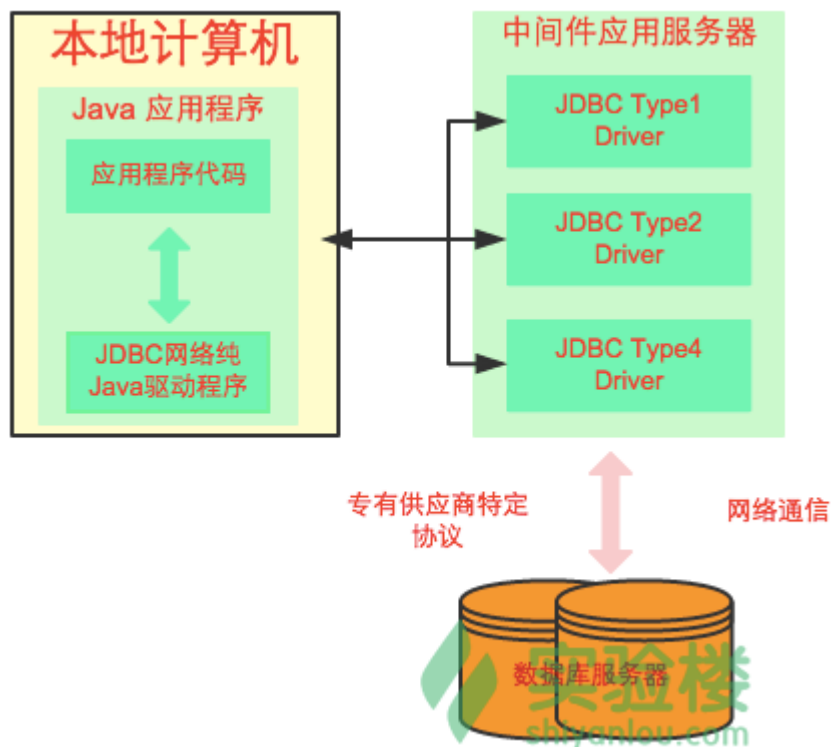


- 本地 API 用 Java 来编写的驱动程序: 这种类型的驱动程序把客户机 API 上的 JDBC 调用转换为 Oracle、Sybase、Informix、DB2 或其它 DBMS 的调用。

## JDBC 入门教程 (/courses/110)

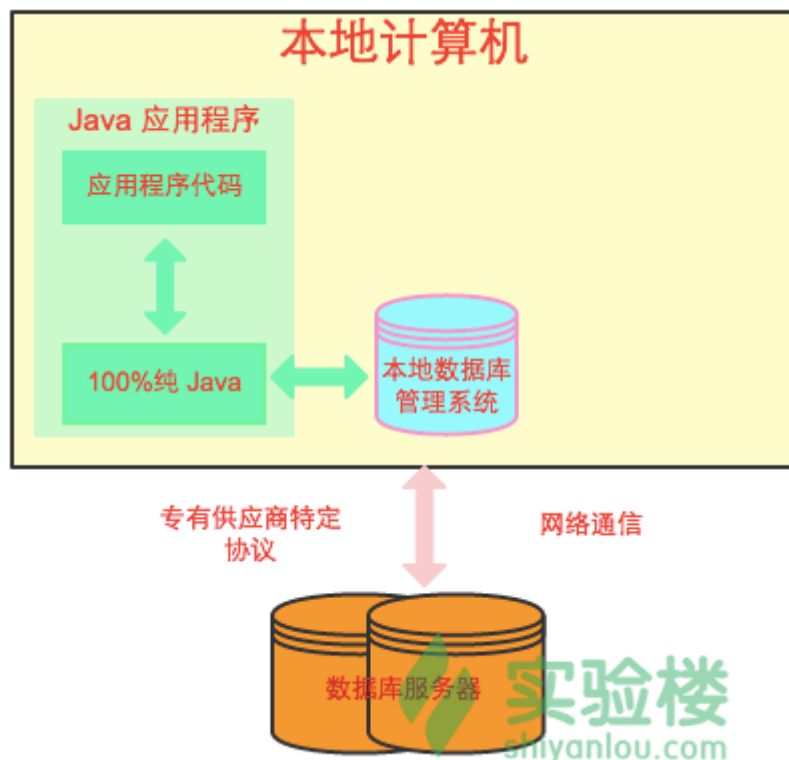


- JDBC 网络纯 Java 驱动程序：这种驱动程序将 JDBC 转换为与 DBMS 无关的网络协议,这是最为灵活的 JDBC 驱动程序。它是一个三层的方法来访问数据库，在 JDBC 客户端使用标准的网络套接字与中间件应用服务器进行通信。然后由中间件应用服务器进入由 DBMS 所需要的的调用格式转换，并转发到数据库服务器。



- 本地协议纯 Java 驱动程序：这种类型的驱动程序将 JDBC 调用直接转换为 DBMS 所使用的专用网络协议。是 Intranet 访问的一个很实用的解决方法。它是直接与供应商的数据库进行通

- 信, 通过 socket 连接一个纯粹的基于 Java 的驱动程序。这是可用于数据库的最高性能的驱动程序, 并且通常由供应商本身提供。
- ⑤ JDBC 入门教程 (/courses/110)



MySQL 的 Connector/Java 的驱动程序是一个类型4驱动程序。因为他们的网络协议的专有性的, 数据库厂商通常提供类型4的驱动程序。

通常情况下如果正在访问一个类型的数据库, 如Oracle, Sybase或IBM, 首选驱动程序是类型4。

如果Java应用程序同时访问多个数据库类型, 类型3是首选的驱动程序。

第2类驱动程序是在类型3或类型4驱动程序还没有提供数据库的情况下显得非常有用。

类型1驱动程序不被认为是部署级别的驱动程序, 它通常仅用于开发和测试目的。

## 2.3 JDBC 连接数据库

涉及到建立一个 JDBC 连接的编程主要有四个步骤:

1. 导入 JDBC 驱动: 只有拥有了驱动程序我们才可以注册驱动程序完成连接的其他步骤。
2. 注册 JDBC 驱动程序: 这一步会导致 JVM 加载所需的驱动类实现到内存中, 然后才可以实现 JDBC 请求。
3. 数据库 URL 指定: 创建具有正确格式的地址, 指向到要连接的数据库。
4. 创建连接对象: 最后, 代码调用 DriverManager 对象的 getConnection() 方法来建立实际的数据库连接。

接下来我们便详细讲解这四步。

📍 JDBC 入门教程 (/courses/110)

### 2.3.1 导入 JDBC 驱动

我们已经在上一节课为同学们就相应的方法做了详细的介绍，不清楚的同学可以查看上一节实验课的内容。

### 2.3.2 注册 JDBC 驱动程序

我们在使用驱动程序之前，必须注册你的驱动程序。注册驱动程序的本质就是将我们将要使用的数据库的驱动类文件动态的加载到内存中，然后才能进行数据库。比如我们使用的 Mysql 数据库。我们可以通过以下两种方式来注册我们的驱动程序。

1、方法1——Class.forName():

动态加载一个类最常用的方法是使用Java的Class.forName()方法，通过使用这个方法将数据库的驱动类动态加载到内存中，然后我们就可以使用。

使用Class.forName()来注册 Mysql 驱动程序：

```
try {  
    Class.forName("com.mysql.jdbc.Driver");  
}  
catch(ClassNotFoundException ex) {  
    System.out.println("Error: unable to load driver class!");  
    System.exit(1);  
}
```

2、方法2——DriverManager.registerDriver():

```
Driver driver = new com.mysql.jdbc.Driver();  
DriverManager.registerDriver(driver);
```

### 2.3.3 指定数据库连接URL

当加载了驱动程序，便可以使用 DriverManager.getConnection() 方法连接到数据库了。

这里给出 DriverManager.getConnection() 三个重载方法：

## 🔗 JDBC 入门教程 (/courses/110)

getConnection(String url)

getConnection(String url, Properties prop)

getConnection(String url, String user, String password)

数据库的URL是指向数据库地址。下表列出了下来流行的JDBC驱动程序名和数据库的URL。

RDBMS	JDBC驱动程序名称	URL
Mysql	com.mysql.jdbc.Driver	jdbc:mysql://hostname/ databaseName
Oracle	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@hostname:port Number:databaseName
DB2	COM.ibm.db2.jdbc.net.DB2Driver	jdbc:db2:hostname:port Number/databaseName
Sybase	com.sybase.jdbc.SybDriver	jdbc:sybase:Tds:hostname: port Number/databaseName

### 2.3.4 创建连接对象

下面三种形式DriverManager.getConnection()方法来创建一个连接对象，以 Mysql 为例。getConnection()最常用形式要求传递一个数据库URL，用户名 username和密码 password。

#### 1、使用数据库URL的用户名和密码

```
String URL = "jdbc:mysql://localhost/EXAMPLE";  
String USER = "username";  
String PASS = "password"  
Connection conn = DriverManager.getConnection(URL, USER, PASS);
```

#### 2、只使用一个数据库URL

然而，在这种情况下，数据库的URL，包括用户名和密码。

```
String URL = "jdbc:mysql://localhost/EXAMPLE?user=root&password=0909";  
//Mysql URL的参数设置详细可以查阅相关资料  
Connection conn = DriverManager.getConnection(URL);
```

#### 3、使用数据库的URL和一个Properties对象

### JDBC 入门教程\* (/courses/110)

```
String URL = "jdbc:mysql://localhost/EXAMPLE";
Properties pro = new Properties( );

//Properties对象，保存一组关键字-值对
pro.put( "user", "root" );
pro.put( "password", "" );

Connection conn = DriverManager.getConnection(URL, pro);
```

#### 4、关闭JDBC 连接

```
conn.close();
```

## 三、实验总结

本次课程介绍了 JDBC 驱动类型以及 JDBC 连接数据库的相关知识，下一节我们将进入 JDBC 接口的学习。

## 四、课后习题

今天再来看昨天我们写的代码，是不是感觉简单了许多？我们今天讲了许多关于连接的方法，你可以将上一节的代码再在 Eclipse 上测试一下。试着运用不同的方法，学会调试自己的程序！

\*本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。

上一节：[认识 JDBC \(/courses/110/labs/1192/document\)](/courses/110/labs/1192/document)

下一节：[JDBC 接口 \(/courses/110/labs/1194/document\)](/courses/110/labs/1194/document)



