

JDBC代表Java数据库连接(Java Database Connectivity)，它是用于Java编程语言和数据库之间的数据库无关连接的标准Java API，换句话说：JDBC是用于在Java语言编程中与数据库连接的API。

JDBC库包括通常与数据库使用相关，如下面提到的每个任务的API -

- 连接到数据库
- 创建SQL或MySQL语句
- 在数据库中执行SQL或MySQL查询
- 查看和修改结果记录

从根本上说，JDBC是一个规范，它提供了一整套接口，允许以一种可移植的访问底层数据库API。Java可以用它来编写不同类型的可执行文件，如 -

- Java应用程序
- Java Applet
- Java Servlets
- Java ServerPages
- 企业级JavaBeans(EJB)

所有这些不同的可执行文件都能够使用JDBC驱动程序来访问数据库，并用于存储数据到数据库中。

JDBC提供与ODBC相同的功能，允许Java程序包含与数据库无关的代码(同样的代码，只需要指定使用的数据库类型，不需要重修改数据库查询或操作代码)。

先决条件

在进一步了解之前，需要对以下两个Java和数据库SQL语句有很好的了解 -

- JAVA编程
- SQL或MySQL数据库

JDBC架构

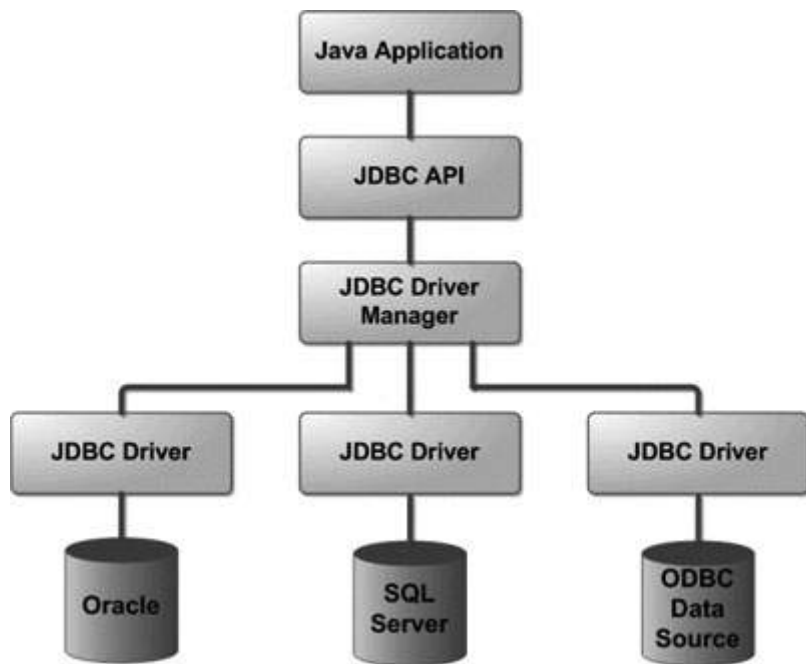
JDBC API支持用于数据库访问的两层和三层处理模型，但通常，JDBC体系结构由两层组成：

- JDBC API：提供应用程序到JDBC管理器连接。
- JDBC驱动程序API：支持JDBC管理器到驱动程序连接。

JDBC API使用驱动程序管理器并指定数据库的驱动程序来提供与异构数据库的透明连接。

JDBC驱动程序管理器确保使用正确的驱动程序来访问每个数据源。驱动程序管理器能够支持连接到多个异构数据库的多个并发驱动程序。

以下是架构图，它显示了驱动程序管理器相对于JDBC驱动程序和Java应用程序的位置 -



常见的JDBC组件

JDBC API提供以下接口和类 -

- **DriverManager**: 此类管理数据库驱动程序列表。使用通信子协议将来自java应用程序的连接请求与适当的数据库驱动程序进行匹配。在JDBC下识别某个子协议的第一个驱动程序将用于建立数据库连接。
- **Driver**: 此接口处理与数据库服务器的通信。我们很少会直接与Driver对象进行交互。但会使用DriverManager对象来管理这种类型的对象。它还提取与使用Driver对象相关的信息。
- **Connection**: 此接口具有用于联系数据库的所有方法。连接(Connection)对象表示通信上下文, 即, 与数据库的所有通信仅通过连接对象。
- **Statement**: 使用从此接口创建的对象将SQL语句提交到数据库。除了执行存储过程之外, 一些派生接口还接受参数。
- **ResultSet**: 在使用Statement对象执行SQL查询后, 这些对象保存从数据库检索的数据。它作为一个迭代器并可移动ResultSet对象查询的数据。
- **SQLException**: 此类处理数据库应用程序中发生的任何错误。

JDBC 4.0包

java.sql和javax.sql是JDBC 4.0的主要包。这是编写本教程时最新的JDBC版本。它提供了与数据源进行交互的主要类。

这些包中的新功能包括以下更改(增强) -

- 自动数据库驱动程序加载
- 异常处理改进
- 增强的BLOB/CLOB功能
- Connection 和 Statement接口的增强
- 国家字符集支持
- SQL ROWID访问
- SQL 2003 XML数据类型的支持
- 注解支持

