

JDBC和ODBC(转) -java基础 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/461/2021\\_2022\\_JDBC\\_E5\\_92\\_8CODB\\_c104\\_461613.htm](https://www.100test.com/kao_ti2020/461/2021_2022_JDBC_E5_92_8CODB_c104_461613.htm) 数据库系统是由多个异构的成员数据库系统组成的数据库系统，异构性体现为各个成员数据库之间在硬件平台、操作系统或数据库管理系统等方面的不同。Internet上大量信息必须通过数据库系统才能有效管理。由于历史等原因，Internet上的数据库系统不少是异构的。为了在Internet环境下，基于异种系统平台实现对异构数据库的查询和联合使用，必须提供一个独立于特定的数据库管理系统的统一编程界面。目前许多数据库系统均支持SQL，对于由SQL数据库组成的异构数据库系统，JDBC和ODBC为访问其异构成员提供了统一的方式，也为各异构成员之间的协作和多个成员之上的操作打下了基础。本文简单介绍了JDBC和ODBC对各个异构的数据库进行统一访问和协作的原理及其应用。ODBC (Open DataBase Connectivity) 是微软倡导的、当前被业界广泛接受的、用于数据库访问的应用程序编程接口 (API)，它以X/Open和ISO/IEC的调用级接口(CLI)规范为基础，并使用结构化查询语言 (SQL) 作为其数据库访问语言。ODBC总体结构有四个组件：应用程序执行处理并调用ODBC API函数，以提交SQL语句并检索结果。驱动程序管理器 (Driver Manager) 根据应用程序需要加载/卸载驱动程序，处理ODBC函数调用，或把它们传送到驱动程序。驱动程序处理ODBC函数调用，提交SQL请求到一个指定的数据源，并把结果返回到应用程序。如果有必要，驱动程序修改一个应用程序请求，以使请求与相关的DBMS支

持的语法一致。数据源包括用户要访问的数据及其相关的操作系统、DBMS及用于访问DBMS的网络平台。ODBC驱动程序的使用把应用程序从具体的数据库调用中隔离开来，驱动程序管理器针对特定数据库的各个驱动程序进行集中管理，并向应用程序提供统一的标准接口，这就为ODBC的开放性奠定了基础。数据库独立性 ODBC是为最大的互用性而设计的，要求一个应用程序有用相同的源代码（不用重新编译或重新链接）访问不同的数据库管理系统(DBMS)的能力。ODBC定义了一个标准的调用层接口（CLI）。这包含X/Open和ISO/IEC的CLI规范中的所有函数，并提供应用程序普遍需要的附加函数。每个支持ODBC的DBMS需要不同的库或驱动程序，驱动程序实现ODBC API中的函数。当需要改变驱动程序时，应用程序不需要重新编译或者重新链接，只是动态加载新的驱动程序，并调用其中的函数即可。如果要同时访问多个DBMS系统，应用程序可加载多个驱动程序。如何支持驱动程序取决于操作系统，例如，在Windows操作系统上，驱动程序是动态链接库（DLL）。使用相同源代码的应用程序访问不同的DBMS，体现了ODBC的数据库独立性。对数据库特殊功能的支持 各个DBMS参照的标准、提供的功能不尽相同，应用程序如何以统一的接口使用各个DBMS特有的功能呢？一方面，ODBC为所有DBMS功能都定义了公共接口。这些DBMS功能比多数DBMS支持的更多，但只要要求驱动程序实现这些功能的一个子集。另一方面，ODBC定义了API和SQL语法一致层，它规定驱动程序应支持的基本功能。ODBC还提供两个函数（SQLGetInfo和SQLGetFunctions）返回关于驱动程序和DBMS能力的一般信息及驱动程序支持

的函数列表。因此，应用程序可以检查DBMS支持的特殊功能。这样，编写应用程序时，就可以检查并自动使用各个驱动程序对应的DBMS所支持的特殊功能。这样做的优点是当增加DBMS支持的功能时，应用程序不需要改变，只需安装更新的驱动程序，应用程序便可以自动发现并使用这些功能。

**互操作能力** 通过使用多个驱动程序可以同时访问多个DBMS系统。ODBC提供的Driver Manager实现所有的ODBC函数，多数是传递调用给驱动程序中的ODBC函数，并静态链接应用程序，或在应用程序运行时加载它。这样，应用程序在Driver Manager中按名调用驱动ODBC函数，而不是通过每个驱动程序中的指针。当应用程序需要通过特定的驱动程序时，它首先需要有一个标识驱动程序的连接句柄。Driver Manager加载驱动程序，并存储每个驱动程序中的函数地址。要使驱动程序调用一个ODBC函数，可在应用程序调用Driver Manager中的函数，并为驱动程序传送连接句柄，然后Driver Manager使用以前存储的地址来调用函数。ODBC可以同时连接到多个DBMS，解决了同时访问多个DBMS的问题，提供了异构成员数据库之间互操作的能力。

**JDBC** JDBC(Java DataBase Connectivity)是Java与数据库的接口规范，JDBC定义了一个支持标准SQL功能的通用低层的应用程序编程接口(API)，它由Java语言编写的类和接口组成，旨在让各数据库开发商为Java程序员提供标准的数据库API。JDBC API定义了若干Java中的类，表示数据库连接、SQL指令、结果集、数据库元数据等。它允许Java程序员发送SQL指令并处理结果。通过驱动程序管理器，JDBC API可利用不同的驱动程序连接不同的数据库系统。JDBC与ODBC都是基于X/Open的SQL调用

级接口，JDBC的设计在思想上沿袭了ODBC，同时在其主要抽象和SQL CLI实现上也沿袭了ODBC，这使得JDBC容易被接受。JDBC的总体结构类似于ODBC，也有四个组件：应用程序、驱动程序管理器、驱动程序和数据源。JDBC保持了ODBC的基本特性，也独立于特定数据库。使用相同源代码的应用程序通过动态加载不同的JDBC驱动程序，可以访问不同的DBMS。连接不同的DBMS时，各个DBMS之间仅通过不同的URL进行标识。JDBC的 DatabaseMetaData接口提供了一系列方法，可以检查DBMS对特定特性的支持，并相应确定有什么特性，从而能对特定数据库的特性予以支持。

与ODBC一样，JDBC也支持在应用程序中同时建立多个数据库连接，采用JDBC可以很容易地用SQL语句同时访问多个异构的数据库，为异构的数据库之间的互操作奠定基础。但是，JDBC除了具有ODBC的上述特点外，更具有对硬件平台、操作系统异构性的支持。这主要是因为ODBC使用的是C语言，而JDBC使用的是Java语言。Java语言具有与平台无关、移植性强、安全性高、稳定性好、分布式、面向对象等众多优点，而JDBC确保了“100%纯Java”的解决方案，利用Java的平台无关性，JDBC应用程序可以自然地实现跨平台特性，因而更适合于Internet上异构环境的数据库应用。此外，JDBC驱动程序管理器是内置的，驱动程序本身也可通过Web浏览器自动下载，无须安装、配置；而ODBC驱动程序管理器和ODBC驱动程序必须在每台客户机上分别安装、配置。JDBC和ODBC在Internet上的应用

JDBC和ODBC由于具有数据库独立性甚至平台无关性，因而对Internet上异构数据库的访问提供了很好的支持。在Internet上访问数据库通常采用三层模式

。以JDBC为例，在三层模式中客户端的Java Applet主要作为用户界面，它不直接与数据库交换信息，而是通过自定义的应用层网络协议与应用服务器交互，应用服务器通过JDBC与数据库服务器交换信息,并实现应用逻辑。DM3的JDBC和ODBC驱动程序支持目前流行的ASP和JSP技术，可以分别借助ODBC和JDBC同时访问Internet上多个异构的数据库。ASP是面向Web服务器的技术，客户端浏览器不需要任何附加的软件支持。ASP使用VBScript之类的脚本语言，它在HTML代码中嵌入某种程序代码，由HTML代码负责描述信息的显示样式，由嵌入的程序代码来描述处理逻辑。在ASP下，VBScript代码被ASP引擎在Web服务器端解释执行，执行结果被重新嵌入到HTML代码中，然后一起发送给浏览器。这里，VBScript代码可以通过ODBC访问多个异构的数据库。JSP是一种基于Java Servlet的Web开发技术，它和ASP非常相似，但又有区别：在JSP下，嵌入HTML页面的程序代码是Java代码；页面中嵌入的程序代码被编译成Servlet（这种编译操作仅在对JSP页面的第一次请求时发生）并由Java虚拟机执行。这里Java代码可以通过JDBC访问多个异构的数据库，其平台无关性特别好。当前，Internet上的数据库应用已越来越多，JDBC和ODBC必将在Internet上的异构数据库访问中发挥重要的作用。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)