前　言

* *

Visual C++是一个复杂的开发环境。已经有许多 Visual C++的书通过论述像应用程序向导、工具条和菜单之类的 Visual Studio(可视工作室)环境给你介绍了 Visual C++。即使这些书涉及数据库开发，但也常常只在一章或几节中论述数据库开发。

有些数据库书确实重点论述 Visual C++。但是这些书常常使读者有点陷入困境。有些书只是很肤浅地谈到了数据库，但没有深入探讨开发者最需要了解的内容。这些书几乎没有涉及 OLE DB(对象链接和嵌入数据库)供应程序、特殊行集和 COM(组件对象模型)。更糟糕的是,有些 C++数据库的书不使用 Windows 图形用户接口(GUI)。在这样的书中调用数据库,然后使用 Print 语句在 DOS 屏幕上显示输出过程。这些书没有说明数据库如何能增强 Windows 开发的成果。

我的论点是 数据库开发是 99%的所有开发成果的最重要的组成部分，本书满足了涉及更有效数据库的需求，本书所提供的内容如下：

◆用于说明技术的源代码实例

◆描述 OLE DB 数据库开发，其中包括供应程序开发和特殊的 OLE DB用户程序。

◆ODBC(开放数据库互连)开发以及 ODBC 描述

◆专门的论题,例如 OLAP (联机分析处理技术)、SQL(结构化查询语言)以及 Visual Studio 数据库管理工具

◆在运行 Visual C++ Windows 应用程序的环境下开发数据库

本书为谁编写

本书不是为所有的读者编写的，而是为 Visual C++高级用户编写的。

◆本书不适合新程序员和 C++ 的初学者，本书的论题与数据库应用程序有关。你应该先找一本 Visual C++入门，然后再买此书。由于篇幅所限，在某种程度上入门书几乎不可能涉及到数据库开发，而本书论述了这部分内容。

◆虽然我建议应该买一本主要论述 Visual C++环境的书籍,而不是 C++语言的书籍，但本书也许有益于那些从来没有钻研 Visual C++现在的 C++程序员。

◆本书是专门为那些希望训练数据库技能，以及希望利用 OLE DB 新的工具进行开发的 Visual C++程序员编写的。书中有关数据库开发的内容在其他任何书都没有涉及。

本书采用 Visual C++ 6.0 环境，并对其深入分析以便了解如何编写高级 Visual C++数据库应用程序。本书与其他的 Visual C++书相衔接，涉及了其他书中没有涉及的内容(除非其他书的作者先读了本书)。我没有重复文档编辑,而是探究完整的 ODBC 和 OLE

需要全本请访问 www.ertongbook.com

DB组件,用源代码实例分解、检测、检查组件。本书不只是 Visual C++ 4.0 再加上几个功能部件的工具再版,Visual C++ 6.0 与早期的 Visual C++ 版本有天壤之别。OLE DB 对 Visual C++ 6.0 是完全陌生的数据库,看来 Microsoft 公司将把基点准确地定位在数据库访问的新工具上。那些不学习及其不懂得 OLE DB 的人将会落伍。

虽然本书的内容超前,但读起来并不困难。本书中有许多代码,每章中有多达 4个功能完整的确切注释的 Visual C++应用程序,比任何 Visual C++书的内容要多得多。包括下列内容:

- ◆ 超出如何绘制窗口的 AppWizard 指令
- ◆ 使用 OLE DB 用户程序的动态 HTML(超文本标记语言),给使用 Visual C++程序的 Web应用程序提供数据库的能力
- ◆ OLE DB 供应程序指令。你可以为诸如文本文件、目录结构或 Email 之类的非数据库数据源编写供应程序
- ◆ 说明 ODBC 内部结构及其如何使用 ODBC 为你工作的 ODBC 主题

你也会逐渐懂得 ADO(ActiveX 数据对象)和 DAO(数据访问对象)如何在 Visual C++ 环境中运行。你会获得实例、一步一步的指令、提示、警告、处理多种类别的参考以及关于数据库开发的注释。

OLE DB 的重要性

Visual C++ 6.0 采用了能够快速创建用户程序工具,该程序能够访问 OLE DB 供应程序的数据。

在 Visual C++ 6.0 以前,多数 OLE DB 数据库访问是通过 ADO实现的。在其他的语言中,ADO仍然是 OLE DB 访问的主要工具,而在 Visual C++ 6.0 版本中,OLE DB 用户程序是作为主要的数据库开发工具而取代 ADO。可以开发 OLE DB 用户程序,该程序使用有源模板库(ATL)容易开发快速的 Visual C++ 应用程序。OLE DB用户程序允许快速访问含有较大控件的 OLE DB 供应程序。

本书详细论述了 OLE DB。在本书中,你会看到许多的 ATL 实例以及 MFC(微软基础类库)的实例、如何编写 OLE DB 供应程序实例以及特殊的行集和 OLE DB特殊处理的实例。

如何使用本书

本书分为五部分:

第 1 部分 介绍 Visual C++数据库

本部分包括:论述了数据库基本原理和 Visual Studio 工具,包括 Microsoft 的通用数据库管理,并且还讨论了存储过程数据库、触发程序、视图、图表以及 New DataBase Wizard(新数据库向导)。

第 2 部分:ODBC 程序设计

本部分包括分解完整的 ODBC组件。本部分不仅论述了 ODBC API ,而且还论述了 ODBC 注册表条款;ODBC Windows 应用程序和报表;ODBC 应用程序的添加、更新和删除

能力 事物处理支持 表链接以及多重 ODBC 记录集。

第 3 部分:OLE DB 程序设计

第 3 部分探究了组件技术概述、OLE DB 用户程序结构、许多 MFC OLE DB 用户应用程序以及更多的 ATL OLE DB 用户应用程序。OLE DB错误处理一章非常有帮助，因为该章教你如何查明数据库应用程序确切的问题所在。本章还包括命令行集、表行集、书签以及 BLOB(二进制大对象)。本章还论述了像成批行集、数组行集、多结果行集和枚举元行集之类的特殊行集。最后，本章介绍了 OLE DB 供应程序。

第 4 部分 特殊数据库论题

本部分包括 利用动态 HTML 的 Visual C++ COM 对象，逐渐学会开发 Web 应用程序。本部分开发了更多的 OLE DB 供应程序,说明如何利用 OLE DB 用户应用程序访问文本和 Email。还介绍了作为未来趋势的 OLAP（联机分析处理技术）。该技术能够改变数据存储方法。本章还论述了应用程序的开发、ADO 和 DAO。

第 5部分 附录

虽然有些论题很重要，但似乎不适合作为正文部分。这就是本书有涉及这些重要论题的附录章节的原因。

- ◆ 在这本书中很难显示错误代码。而附录 A 显示 HRESUL 代码 SQL STATE 代码及其 ADO 错误代码。
- ◆ MFC AppWizard 准许使用“环绕式处理”ODBC API （应用程序设计接口）来创建 ODBC应用程序。但是你也许需要直接访问 ODBC API ,如本书第 2 部分所示。附录 B 给出了 ODBC API 函数所需要的各种文档、返回代码以及不良功能。
- ◆ OLE DB 属性准许 OLE DB 用户程序控制 OLE DB 供应程序函数。附录 C 列出了这些属性。
- ◆在附录 D 中列出了本书 CD-ROM 中的内容。

除了正常的正文和阐述外，本书还包括许多特别吸引人的图标。例如：注释、提示、警告、补充说明和代码程序。下面所示的图标用于特别吸引人的内容。



你也许能够找到参考章节或对本书当前论题的阐述。该图标用来告诉你本书中论题的其他阐述。



如果你能在 CD-ROM 中找到像源程序代码错误的内容，该标号告诉在哪查找。



警告图标告诉你这里是许多开发人员遇到的问题，以便避免犯别人犯过的错误。



提示图标描述你应该知道的某种技术，以便帮助你取得开发成果或者改进程序质量。



注释图标扩展正文中所提到的某一个论点，注释的确不是正文的组成部分，但能帮助读者全面地理解正文。

什么是补充说明

如果诀窍超出了本书正常范围，而且过长，就可能放在补充说明中。补充说明把过长的信息提取出来,因此在读者学习 Visual C++高级数据库开发时，不会分散注意力。

结束语

Microsoft 的 Visual C++ 已经成为了其他编译程序所参照的标准。 Visual C++ 比其他的 C++编译程序占有更多的市场份额和具有更健全的环境。

本书会帮助你踏上成为世界一流的 Windows 和 Internet 数据库开发人员征程。本书不仅是作者而且是出版公司作出的巨大努力的结果。

这本极其出色的书的确能帮助你成为一名优秀的 Visual C++数据库开发者。希望你喜欢这本书。

第Ⅰ部分

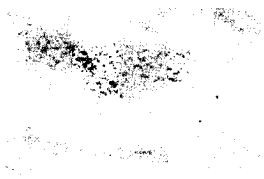
介绍 Visual C++ 数据库

第 1 章
数据库和 Visual C++

第 2 章
了解 SQL

第 3 章
使用 ODBC 数据库工程

需要全本请访问 www.ertongbook.com



第 1 章

数据库和 Visual C++

＊＊

本章包括

◆了解客户／服务器、SQL 语句和通用数据访问的重要性

◆了解 OLE DB、ODBC 和 ADO 在通用数据访问规划中的位置

◆了解当开发数据库应用程序时，应用程序设计的结果

◆了解 OLE DB、ADO 和 ODBC 之间的关系以及各自的结构

＊＊

看来，许多 Visual C++ 书，如果涉及到论述数据开发的内容，其内容陈旧。本书不同于这些书，因为本书的主题是论述使用 Visual C++，在 Visual Studio 内开发数据库。本章不仅阐明了 Visual C++所包括的一些基本概念，而且也充当本书其余章节的铺垫。

客户 / 服务器的重要性

在 20 世纪 60 年代直到 70 年代末，当计算机最初开始用于商业时，其价格太昂贵，多数人买不起。当时的计算机只有主机。在大公司里，主机进行所有的处理过程、文件共享和数据库管理。信息被传播到“哑终端”，其唯一的作用就是与大型主机交流。上述内容由图 1-1 说明。

20 世纪 80 年代初，人们采用了个人计算机。个人计算机实现了独立处理，尤其是用于电子表格分析。但这几乎不能连接到大型主机上或中央存储器上。但是，由于个人计算机成本费用更经济、功能更复杂并且更有效，所以工矿企业开始认为 PC 网络是使某些计算过程分散到每个工作站进行处理的一种方法。因此，减轻了中央大型计算机的压力。随着时间的推移，在许多领域中网络开始取代了主机(见图 1-2)。

网络产生了客户／服务器计算。在客户／服务器计算中，有些处理或存储发生在集中的服务器中，而有些出现在客户机中。通过把工作分布在最合适的地方，客户／服务器计算减少了网络的通讯量，也能实现更快、更灵活及更复杂的程序设计(例如 Windows 程序设计）。即使是集中式计算机也没有强大到像大型主机一样取代网络。

通过考虑在服务器中数据流量与在客户机中的处理容量一致，开发商不仅能给用户提供使用共享的集中数据的机会，而且还给用户提供使用先进的程序来完成工作的机会。当使用数据库服务器程序时，数据库引擎在服务器中运行，这样就给其他客户的机器提供所需

要的数据。

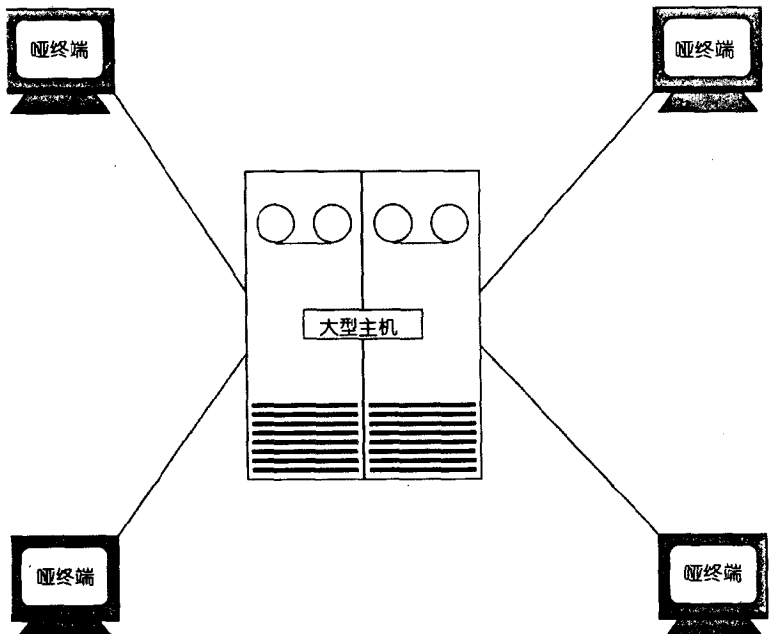


图 1-1　主机计算使所有处理和存储过程出现在集中的位置

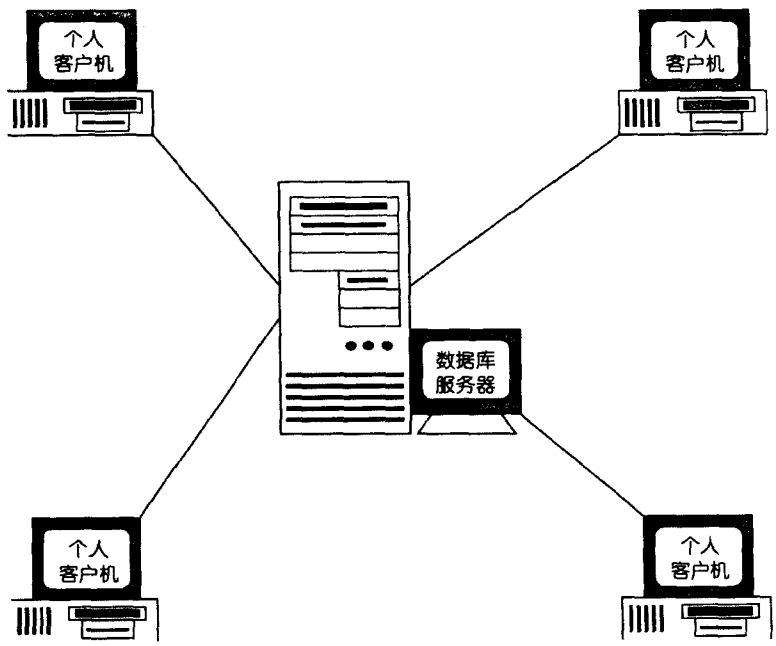


图 1-2　网络使得客户机与服务器共享信息，引发客户／服务器计算过程

这种配置带来了若干优点。这些优点包括：SQL语言。SQL是一种通用语言，它使数

据库开发者能够使用同一种技术来查询、更新数据库;ODBC实现了标准的程序访问任何数据库;OLE DB和ADO使标准化访问任何数据源成为可能，而不只是存在于集中数据库的数据源。

本节专门讨论现代客户／服务器技术的三个领域。这些论题是SQL语句,应用程序设计以及Web浏览器。

结构化查询语言(SQL)

SQL开发于20世纪70年代，但作为一种所有数据库中的标准语句至今仍在使用。SQL有两个功能：

- ◆ SQL提供了一种通用数据库访问语言，以便在某种数据库受过训练的开发者能够把相同的技术应用到另一种数据库中。
- ◆ 设计SQL用于关系数据库访问。不同于传统过程驱动语言的方法SQL的研制者认为SQL仅仅用于数据库访问。

在SQL没有被开发前，那些想访问文件或数据库信息的开发者必须使用某种复杂的代码。例如：比如表中或文件中的日期是一年多前的日期，你想把一个表中或文件中的日期嵌入现在的日期中。如果不用SQL，你必须编写一个与下列伪代码相似的程序。

```
open file or table
do
do while next record is locked by another user
  wait
loop
lock record
read record into data structure
if not EOF
  if date is over 1 year old、
   set date in data structure to today's date
   update date with data in data structure
  end if
end if
unlock record
loop until EOF
close file or table
```

甚至这种伪代码也相当复杂。实际上，实现这种伪代码也有许多困难。请看使用SQL：

```
Update MyTable set MyTable. date = today()
where MyTable. date - today() > 365
```



SQL 假设返回日期的函数存在。此日期可以从另一个日期减去以便返回天数，虽然各个数据库系统的函数和存储不同,但可使用 SQL 使数据库中一些函数和运算来达到用最少的代码就能达到目标。

正如你所看到的,SQL UPDATE 利用数据库的相关关系，不用费力就改变了数据。



第 2 章详细讨论 SQL。

应用程序设计的问题

应用程序设计一直很困难。但是，由于现代开发技术，劣质的应用程序设计会给你的软件销售或生产率带来负面影响。开发应用程序时，你应该考虑到以下设计问题：

- Web 怎么样？由于有了 Web，数据库问题不那么复杂。现在每天有数百人访问你的 Web 站点，运行你的应用程序。每个人也许操作不同的浏览器，但依靠击中浏览器的 EXIT键,离开主页。由于有了 Microsoft 的 Information Server(IIS),在任何操作系统中任一浏览器下传输可读性的 Active Server Page 。这要比传统的 JAVA 环境或客户端的脚本制作更适合于你。



关于使用 Visual C++来开发基于 Web 的数据库,见第 14 章。

- 你的客户在什么平台上运行？这个问题很重要。在很多系统中，甚至在当前数据库书籍中,许多作者认为基于本文的 DOS 或者 UNIX环境还在继续使用，而且他们的代码反映出这一点(例如：一些书广泛使用 Print 打印命令提示）。而本书作者认为，你要充分利用 Visual C++从 Windows 9X输送的所有的图表功能。因为在 Visual Studio 环境中编程,本书中没有基于文本的输出设备,这样一来,图形开发者不必费事把基于文本的实例变成基于图形的应用程序。
- 你用什么样的工具？我和编写 IDC 书的合作者经历千辛万苦发现了用于开发数据库的新技术,特别是通过 ATL(Active 样板库）开发。你会发现这些程序常常不紧凑，但是使用起来要比以前的 Visual C++样板中的 OLE DB/ADO 开发更简单。本

书也集中论述了 MFC。如果你仍然偏爱 ODBC 开发，MFC 是最好的方法。虽然我建议你使用 ATL，但你仍可以使用 MFC 编写 OLE DB 应用程序。本书按照读者的要求进行编写的，你们大多数都希望开发尽可能简单，但是又希望应用程序健壮而且具有商业级的属性。由于牢记这一点，本书采用向导输出，并把向导输出定制以适合本章的需要。

- 你需要什么技术来支持？本书不仅说明如何为传统的合作领域容易地开发数据库应用程序，而且还探讨其他书没有涉及的领域如 ATL 开发、Web 数据库应用程序、OLAP 数据库以及 OLE DB 供应程序开发等等。

由于开发者面临所有这些问题，所以应用程序设计非常重要。在开发工程之前，应该先提出这些问题，应用程序应该尽可能地利用技术以保证程序经久不衰。

了解通用数据访问(UDA)

当最初开发程序设计语言时，以文件的形式存储在磁带中，后来在硬盘中。这些文件只含有没有索引的原始数据，因此数据访问速度很慢，而且令人厌烦。在二十世纪六、七十年代，人们开发了存储数据容易、检索方便的数据库。数据库很先进以至于数据访问更容易，并且内装了数据保护系统。现在，如果没有追踪联合数据的数据库记录，商业系统就无法运行。从游戏到联合 Web 浏览器所有的应用程序都需要某种数据库。数据库使用户能够查看、更新或从数据库中删除数据。

Microsoft 公司用开放数据库互连（ODBC）标准、远程数据对象(RDO)、数据访问对象(DAO)支持数据库。ODBC 是一系列的标准函数，这些函数使开发者连接、处理数据库。数据库访问以指数增加，通常是归因于 ODBC 传送的工具和相互操作性。 RDO 通常使用 ODBC 数据源访问远程服务器的数据，虽然后来的 DAO 版本也能启动 ODBC 访问，但是 DAO 过去主要用于使用 Jet 数据库引擎的 Microsoft Access 数据库。

也存在着使用个人计算机的问题。过去常常有一种把所有的数据集中起来的趋势，而现在需要一个部门的每个成员以最适合个人的格式记录数据。有电子表格中的数据、文本文档数据。通过套接字／调制解调器连接所产生的数据和 E-mail 文件中的数据，这里仅仅提到了几种非关系的数据源（见图 1-3）。Microsoft 公司考虑到必须使所有的数据访问标准化（不管是否来自数据库）而不是把相关的数据库标准化。为了满足这一需求，Microsoft 开发了通用数据访问 (UDA)。

通用数据访问是 Microsoft 公司所采用的一种策略。它提供了访问所有类型的相关和不相关信息的通用程序。只要 OLE DB 的数据源的供应程序存在，通用数据访问使所有的 Visual Studio 工具能够访问任何平台上任何数据。通用数据访问协议使应用程序能够利用一组 API 类库调用访问无数的数据源。通用数据访问由以下三个核心技术组成。

- 对象链接和嵌入数据库（OLE DB）是用于与任何数据源互通的协议
- Active X 数据对象(ADO)是所有非 Visual C++ 程序使用的技术。该程序启动 OLE DB 数据访问。
- 开放数据库互连(ODBC）仍然用来连接任何数据源。Microsoft 包括一个访问任何数据源的 OLE DB 供应程序。

需要全本请访问 www.ertongbook.com

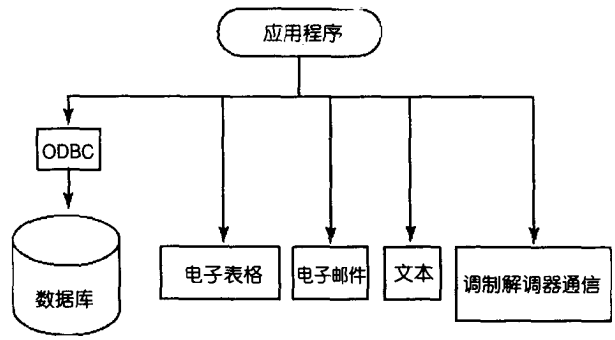


图 1-3 对于使用传统数据库访问技术的每一个数据源来说，数据访问具有不同的特点

此协议的最终结果是在所有的数据源之间有了一个任何程序可访问的层，图 1-4 中用图显示这个层。

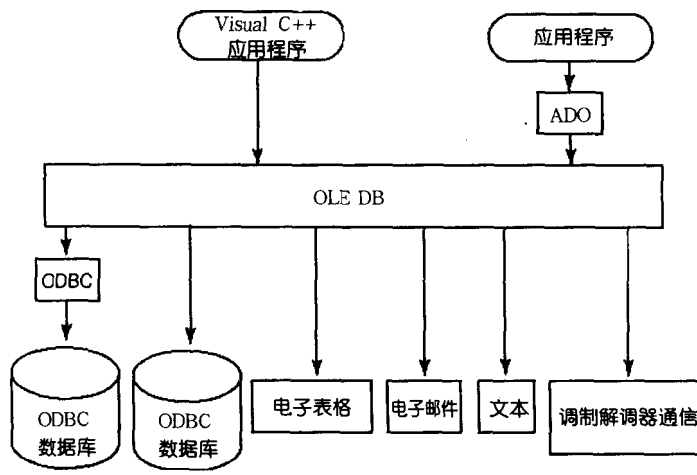


图 1-4 UDA 使用 OLE DB技术访问任何数据源

下面的章节描述了 UDA组件并讨论了这些组件的特性。

对象链接与嵌入数据库(OLE DB)

OLE DB是 Microsoft公司的数据库连接的最新标准。OLE DB 由一组 COM(组件对象模型）接口组成。这些接口提供集中数据库管理服务。设计 OLE DB 用来访问几种不同的数据源。为了访问数据库，许多数据库销售商（如 Oracle 公司、Informix 公司、Microsoft 公司)不仅提供了 ODBC驱动程序,而且提供了 OLE DB驱动程序或者 OLE DB 供应程序。

OLE DB 的两个基本结构是 OLE DB供应程序和 OLE DB 用户程序。编写供应程序提

供机会访问某个特殊的数据源，该数据源常常就是一个数据库。但也可能是提供任何其他的数据源,例如 调制解调器传输、电子表格、文档或文本文件。你可以在供应程序中获取数据流,然后可以在 OLE DB 用户程序中使用该数据流。如果你的供应程序启动了这个功能，像使用数据库表一样，可以前后滚动甚至更新。图 1-5 显示了 OLE DB的基本结构。

如图 1-5 所示，应用程序得到的每一个数据源具有单一的 OLE DB 供应程序,该供应程序用于访问数据源。几个不同的用户程序可以使用该 OLE DB供应程序。这些用户程序都是相同的，或者编写专门的用户程序来利用供应程序所启动的特殊过程。这样一来，由于 OLE DB 允许简化的数据访问每一个数据源，OLE DB 不仅作用大,而且非常灵活,因为可以编写用户程序，这些用户程序可以利用供应程序各方面的优点。

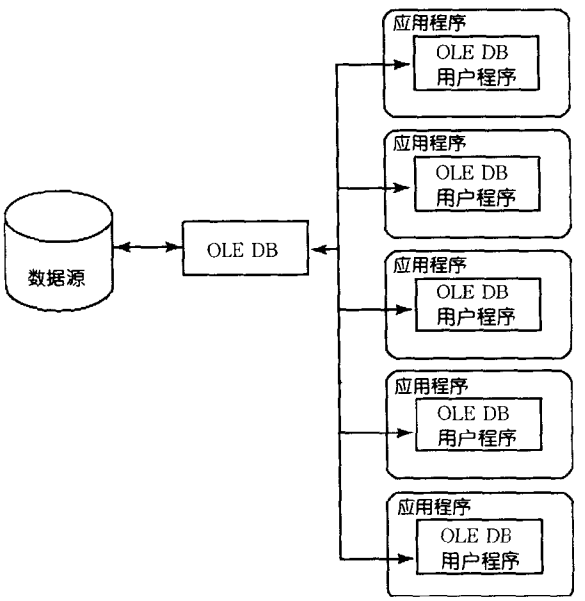


图 1-5 OLE DB依靠 OLE DB供应程序和 OLE DB 用户程序之间的交互作用



本书的第 3 部分专门研究 OLE DB,并讨论 OLE DB 提供的许多功能。

ActiveX 数据对象 (ADO)

如前面的图 1-4所示,ADO是一个能快捷、便利地访问 OLE DB的类层。实际上,ADO是类属程序 OLE DB用户程序,设计用来访问任何 OLE DB 供应程序。ADO 设计上类似于 ODBC、DAO 和 RDO编码,因此那些想转向 OLE DB 的开发者很容易做到这一点。ADO还可以复制早期数据库技术所支持的接口。通过使棘手的任务容易完成，以及增强性能，

ADO 也已改进了以前的接口。ADO的主要目标是使得 OLE DB 供应程序提供的普通运算更容易。因为 ADO使用 OLE DB ,所以你会注意到,对象、接口以及这两种技术的所有性能之间有许多相似之处。在不同的 OLE DB 供应程序之间,ADO 也隐藏着许多不兼容性,弥补了供应程序中的遗漏功能。

ADO跨越平台使用,使任何语言的开发者(包括 Visual J + +)无缝地访问数据。所有的 Visual Studio工具都能用 ADO访问数据。ADO 大大减少了编写复杂客户 /服务器和 Active X代码所需要的时间。本章后几节讨论 ADO 和 Visual J++ 的可视数据库工具。

本书后几章探讨使你能用 ADO进行访问。



第 18 章将讨论 ADO 的开发。

开放数据库互连 (ODBC)

ODBC是最开放性的、当前用于相关的数据连接的最成功的标准。多数数据库除了包括本机驱动程序外还包括访问数据库的 ODBC 驱动程序。有些数据库甚至放弃了本机驱动程序,只使用 ODBC 驱动程序。图 1-6 显示了如何构造一个 ODBC。

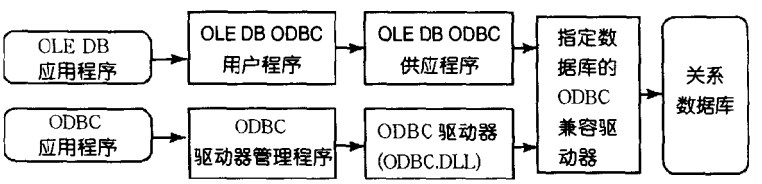


图 1-6 设计 ODBC 连接各种各样的关系数据库

在图 1-6 中,可以看出,数据库销售商所要做的一切就是编写指定数据库的 ODBC驱动程序。然后用这个 ODBC驱动程序访问该数据库,并且给 ODBC 和 OLE DB 应用程序传送数据库数据。

ODBC数据库也能容易地从不同的数据库转移。如果改变数据库引擎,你只需要使 ODBC 接口指向新数据库,而不用以新的专用数据库调用取代应用程序中旧的专用的程序调用。

OLE DB不使用 ODBC Driver Manager,为了高效而使用它自己的 ODBC 供应程序,使用它来实现快速 ODBC数据访问。从这种意义上来说,你仍然可以使用 OLE DB 访问 ODBC 数据库,而不失去 ODBC 多数据库支持的优势。

OLE DB 与 ODBC

OLE DB 没有得到所有的数据库销售商的广泛支持，至今还没有被 Visual C ++ 市场所接受。另一方面， ODBC 在 Visual C++ 中比任何数据库平台使用的更广泛。然而， ODBC 和 OLE DB 似乎有点重复。据谣传当 OLE DB 的功能完整时， Microsoft 公司打算使 ODBC 和图象不一致。

我们也许会理解为什么新的 OLE DB 标准能够流行。该标准来源于 Windows 3.0 和 3.0 以上的版本。但是，许多工具仍然需要 ODBC 起作用。Java 的 JDBC(Java 数据库连接)、PowerBuilder、CASE(计算机辅助软件工程)工具甚至 Visual Studio 的某些方面（例如数据库工程)仍然需要把 ODBC 作为前端。但是，如果我必须选择现实的技术用于开发新的应用程序的话，我会选择 OLE DB 而不是 ODBC。OLE DB 不久就会享有 ODBC 当前所享有的同一支持级别。你会不愿意使用将要过时的技术编写应用程序。



本书第 2 部分探讨 ODBC 能提供的许多函数性。

应用程序向导

Visual C++ 使用应用程序向导快速开发 Visual C++ 数据库应用程序，或者给现有的应用程序添加数据库的函数性。下面是本书要详细讨论的向导，说明向导与数据库开发之间的关系。

- 利用 OLE DB、ODBC 或者 DAO，MFC AppWizard(微软基础类库应用程序向导)用于开发 MFC 程序。
- 利用微软公司的 Active 模板库(ATL)，ATL AppWizard（应用程序向导）用于开发速度快、小型、高效的程序。当编写 OLE DB 应用程序时，使用模板使开发更容易。
- ATL 对象向导使你能够快速给现行的 ATL 或 MFC 应用程序添加 ATL 对象。例如：OLE DB 供应程序或用户程序或者 ATL 对话框。

在继续读本书的同时，你会发现这些向导更容易实现数据库的开发过程。

小结

Visual C++ 为容易的数据库开发提供了各种各样的工具。应用程序向导、数据库工程以及 MFC 工具使 VC++ 成为客户／服务器开发的强有力的工具。概括如下：

- 客户／服务器技术使某种处理和存储发生在客户和服务器计算机上。数据库开发利用客户／服务器结构。
- SQL （结构查询语言）是一种多数数据库使用的标准语言。因为 SQL 启动关系命令，而传统数据库的定向处理语言不能启动。所以 SQL 是数据库应用程序开发的极大的辅助工具。
- 应用程序设计可能非常复杂。本章包括在开发数据库程序之前你要提出的一些问题。
- UDA（通用数据访问）是对于所有数据源只具有一个接口的 Microsoft 视觉。
- UDA 由 OLE DB、ADO 和 ODBC 组成。
- OLE DB 是 Microsoft 数据库开发的选择工具。OLE DB 利用数据供应程序和数据用户程序启动访问任何数据源的灵活方法。
- Microsoft Visual Studio （微软可视化工作室）包括视觉数据库开发工具。该工具使开发人员能够在窗口内绘制 OLE DB、ODBC 和 DAO 函数性。
- DAO 和 RDO 还可用于连接一个数据库。但是，应该利用 ODBC 或者 OLE DB 进行 Visual C++ 开发。
- 包括容易实现数据库开发的 Visual C++ 向导程序。全书贯穿和讨论用这些向导程序进行数据库应用程序开发。

第 2 章

了解 SQL

＊＊

本章包括

◆掌握 SQL 的基本原理

◆检查 SQL SELECT 语句

◆执行 INSERT、UPDATE 和 DELETE 语句

＊＊

SQL(结构化查询语言)是大多数数据库使用的一种语言。使用 SQL ,你就可以使用任何数据库源和数据。本章探究如何使用 SQL 以及在 C++程序中使用 SQL 的方法。



在每一个 SQL 语句的末尾,有些数据库接口要求使用分号(;),而有些不要求使用。在 Visual C++ 内编程,不能使用。本章没有给 SQL 实例添加分号，但是如果你使用的检测 SQL 数据库接口要求分号，你就应该使用它。

使用 SELECT 语句

SELECT语句用于从数据库中选取信息，最简单的 SELECT语句的格式如下：

```
SELECT column1 , column2 , ...
FROM table1 , table2
```

例如 如果有一个如图 2-1 所示格式的 Microsoft Access 数据库，可以使用如下 SQL 语句从 Student(学生)表中选择全部行(或记录)：

```
SELECT StudentID,
    FirstName,
    MidName,
```

需要全本请访问 www.ertongbook.com