21世纪工程应用计算机技术丛书

C++Builder

应用程序开发实例与技巧

（下册　应用与提高篇）

- C++Builder 数据库编程技术
- C++Builder 网络应用程序开发
- C++Builder 应用程序设计技巧与提高

曹 岩　王海宇　主编

Borland C++Builder

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

C++Builder是Borland公司的应用程序开发工具，它具有面向对象及可视化快速应用程序开发环境的所有特征。全书分为上、下两册，五个部分，共28章。上册“基础篇”介绍了C++Builder集成开发环境(IDE)，C++Builder Windows编程基本技能；下册“应用与提高篇”介绍C++Builder数据库编程技术，C++Builder网络应用程序开发，C++Builder应用程序设计技巧与提高等内容。

本书注重介绍编程技巧，图文并茂，讲解浅显易懂，理论结合实际应用。通过大量实例引导读者进入C++Builder编程世界，使用户在实例学习中提高编程能力，并适时总结大量编程技巧，是学习C++Builder编程的得力助手。使读者从掌握基础内容到精通并熟练使用，最后能应用C++Builder作为开发工具，开发面向实际应用需要的应用软件系统。

本书可供C++Builder编程人员、工程技术人员以及CAD/CAM研究与应用人员阅读，也可作为初学者快速学习和使用C++Builder的教材。

图书在版编目(CIP)数据

C++Builder应用程序开发实例与技巧.下册，应用与提高篇/曹岩，王海宇主编；王海宇等编.—西安：西安交通大学出版社，2005.10
ISBN 7-5605-2084-7

Ⅰ.C... Ⅱ.①曹...②王...③王... Ⅲ.C语言-程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字(2005)第116885号

书　　名	C++Builder应用程序开发实例与技巧(下册 应用与提高篇)
主　　编	曹　岩　王海宇
出版发行	西安交通大学出版社
地　　址	西安市兴庆南路25号(邮编:710049)
电　　话	(029)82668357　82667874(发行部) (029)82668315　82669096(总编办)
网　　址	http://press.xjtu.edu.cn
电子邮件	eibooks@163.com
印　　刷	西安建筑科技大学印刷厂
版　　次	2005年10月第1版　2005年10月第1次印刷
开　　本	787mm×1092mm　1/16
印　　张	27.375
字　　数	1104千字
书　　号	ISBN 7-5605-2084-7/TP·415
定　　价	36.00元

目　录

(下册:应用与提高篇)

第三部分　C++Builder 数据库编程技术

第四部分 C++Builder 网络应用程序开发

第 13 章 网络应用程序开发简介

第 14 章 开发基于 CGI 及 ISAPI 的 Web 应用程序

第 15 章 利用 Web Broker 机制开发 Web 应用程序

第五部分 C++Builder 应用程序设计技巧与提高

附:本书上册“基础篇”简明目录

第一部分　C++Builder 集成开发环境(IDE)

第二部分　C++Builder Windows 程序设计

第三部分

C++Builder
数据库编程技术

第 9 章　C++Builder 数据库程序设计架构

在一个数据库应用程序的编写过程中，最难的并非具体要用到哪些技巧去编写程序代码，更何况 CB 的组件本来都非常好用。最难的是当您拿到一个项目后，怎样去总体的规划它。从数据表结构的划分（如用多少个数据表，用什么样的数据表去存储数据以及这些数据表之间的关系），到整个计算机系统的构建（如项目的投资，需要达到的运行效果，数据处理量的大小，整个网络的结构，选用的数据库系统对数据的处理能力，程序及网络系统的可扩展性），这些才是最关键。数据库程序一般都是一个系统工程，如果前期的工作做得好，会对以后该系统的数据维护、程序编写、程序扩展、系统扩展都带来很大的便利。相反，如果您一开始就选择了一个不合适的结构，即便是程序做出来了，随着用户使用要求的增加，需要加入一些新的功能，或者提升系统的处理能力时，却发现基础的缺陷使得做这些事很困难，甚至不可能，那损失将是很大的。

所幸的是，CB 提供了一个良好的数据库程序架构，VCL 的数据感知组件通过对数据库行为和数据库数据的抽象，使得编写可扩展应用程序变得十分容易。通过隔离数据模块中的数据存取组件，可在数据库应用程序中开发窗体以提供一致的用户接口，通过在对象仓库（Object Repository）中存储已设计好的窗体和数据模块的连接，使其它的开发者可以基于现有的基础开始新工程的设计，而不是从头开始。共享窗体和数据模块也使得开发数据库存取和应用程序界面的联合标准成为可能。您还可以从一级应用程序开始项目的开发，以便减少初期的投资，或是用很少时间做出样板工程，及时跟用户沟通加以改进。随着数据量、用户数以及不同的存取数据的应用程序的个数增加，可以按需要扩展应用程序到多级结构。通过可扩展性规划，在应用程序扩展时，就可以保证编写一级或二级应用程序时的投资和现有代码的再利用，从而轻松实现系统升级。CB 还拥有一些功能扩展性很强的数据库组件，甚至拥有商业决策分析软件开发的组件，采用它们的实时分析方式，您可以轻而易举地加入用户提出的一些新的要求而不用重写代码…… 所有这些功能既全面又强大，让您再也不用为系统的扩展而伤透脑筋。

9.1　Client／Server（客户／服务器）结构

提起数据库应用程序，必然会提到“服务器”、“工作站”和“网络”。在早期的大型计算机的操作环境中，每一位用户都有一套终端机和键盘，用来输入数据或程序，并把它们传送到主机上。而程序的运行，数据的访问预处理，以及最终结果的输出都由远程的主机来完成。在这种集中式处理的操作环境下，主机必须是一台功能超强的计算机，不然就无法应对其沉重的工作

负担。可以想像,这样的系统同时连接用户的数量将是很有限的,而整个系统的处理效率也就直接决定于服务器的处理能力,系统处理能力的扩展也就只有一个办法——升级服务器,费用昂贵且效果不佳。

在现代的网络系统中,客户端的计算机早已不是早期那种没有任何处理能力的终端了,网络的传输能力也比以前有了很大提高,那么能不能把服务器上需要处理的事务分一部分拿到客户端的计算机上来处理呢?答案当然是肯定的。

一个数据库应用程序的功能可以分为几大块,如"用户界面处理程序","用户数据处理程序","数据库访问及查询程序","数据库数据处理程序"等等。除了用户界面处理程序,其它的几块可以根据客户机的处理能力、服务器的处理能力、网络系统的信息传输能力等因素,将其放置到不同的机器上去执行,以发挥硬件系统的最大效能。您也可以将一个庞大的数据库根据数据性质的不同,分割成几个小型的数据库,再分别放置到网络上的几个小型服务器上,以提高数据库本身的处理能力。这样一来,就有可能用一个廉价的网络系统构建出高效能的数据库服务系统了。

实际上,上面所提到的就是所谓的"分布式处理"。分布式处理也就是 Client/Sever(客户/服务器)主从结构的全部内涵。从上面的描述中相信您能很容易的发现主从结构的全部优点,那就是它的"可扩展性"和"灵活性"。它可以允许您经过精心规划,发挥一个网络系统的最大潜力;也可以允许您在系统处理能力不足时扩展服务器的数量及层数,或是调整程序的结构,以满足用户的需求;更重要的是,它可以将各种不同类型的计算机,不同通讯协议的网络,不同的数据库操作系统,以及不同操作系统的计算机,不同的用户界面处理程序连接起来,成为一个整体运行的数据库服务系统。您既可以在客户原有的系统上进行新的开发,并利用其旧有的资源;也可以对现有的系统进行自由的功能扩展。而这一切,都取决于 CB 所提供的灵活、高效、开放的数据库程序开发能力。

由于开发一个数据服务系统是一个"软硬兼施"的工程,所以您必须对系统的很多因素加以考虑来决定数据库应用程序的结构规划。如考虑"客户端"和"服务器端"的计算能力,以及网络的传输能力,以此来决定哪些数据的处理程序放在服务器上,那些放在客户端;由于数据在不同的机器上进行处理及多用户的访问,您还必须考虑到共享数据的"同步"和"锁定"问题,即数据的安全性问题等等。

举一个极端的例子,假入您将所有的数据计算都放在了客户端,而服务器只作为一个文件的共享机器,那么势必要将大量的数据从服务器传输到客户端,客户端机器将数据处理完毕后再将其传回服务器,这样的系统如果同时有很多客户端在线,那么网络上的负担将很沉重,这时您就要考虑网络的传输能力了。此时您就可以将一部分数据处理放在服务器上,以减轻网络的负担。

前面提到,在一个"主从结构"的数据服务系统中可以允许有多台服务器,那么这些服务器之间又是什么样的关系呢?这就牵扯到另外一个关于主从结构系统开发的层级问题。

9.2 关于数据库的分类

设计数据库应用程序,离开数据库系统当然是不行了。数据库应用领域里由于商业利润丰厚,因此吸引了众多的厂商参与竞争,数据库系统软件也是相应的种类繁多。这一方面为人们提供了更多的选择空间,但另一方面也给程序的开发带来不小的困扰。很难用一句话来说明哪一家的数据库系统是最好的,根据您的财力,工程的要求,来选择最适合的数据库操作系统恐怕才是上策,适用即是最好。

数据库操作系统这么多,对没有数据库使用经验的程序员来说,要选择一个还真不是易

事。可总得有个整体的概念，下面就给它们分类。

1. 本地数据库

本地数据库一般是位于本地硬盘或局域网络的一个共享目录上，它们一般都有专门的 API 函数来访问数据库。通常它们是特别为某一系统设计的，当它们被许多用户共享时，需要使用“基于文件的锁定机制”，因此它们有时被称为“基于文件的数据库”。

关于文件的锁定机制，作为程序的开发者应充分注意。像古老一点的 dBASE，它本身不提供这种文件的锁定机制，您需要在程序中自己编写代码来处理文件锁定的问题；而 Paradox 数据库就不用您再大费周折了。因此，除非您是在老的系统上做扩展性开发，一般不要使用 dBASE，在 CB 中，本地数据库程序使用 Paradox 可能就是一个很好的选择，因为它本身就是 Borland 公司自己的产品，所以与 CB 中所有的开发工具兼容性良好，而且安装了 CB 以后就有了它，何必再花冤枉钱去搞其它东西呢？

因为本地数据库与数据库应用程序经常位于同一个系统上，本地数据库一般比远程数据库服务器速度更快。本地数据库的数据表基于文件而建立，所以就数据的存储量而言，本地数据库比远程数据库服务器要少得多。在决定是否使用本地数据库时，必须要考虑有多少数据要存储在数据库里。

使用本地数据库的应用程序被称为一级数据库应用程序，这是因为应用程序与数据库共享一个文件系统。

本地数据库常见的有 Paradox，dBASE，FoxPro 和 Access。

2. 远程数据库

远程数据库操作系统通常是位于一台远程机器上。它们一般使用结构式查询语言（SQL）使客户端能访问数据库，因此它们有时被称为 SQL 服务器（也称为远程数据库管理系统，或 RDBMS）。除了构成 SQL 公共指令外，大部分远程数据库服务器支持自己特有的 SQL。

远程数据库服务器是专门为多个用户同时访问数据库而设计的，与本地数据库使用基于文件的锁定系统不一样，远程数据库服务基于“事务”，因而提供了更为复杂的多用户支持。另外，远程数据库服务器比本地数据库能处理更多数据。有时候，远程数据库服务器的数据甚至可以不存放在一台机器上，而是分布在许多台机器上。

使用远程数据库服务器的应用程序被称为二级数据库应用程序或多级数据库应用程序，因为应用程序与数据库在各自独立的系统（或层次）上运作。

远程数据库操作系统，常见的有 InterBase，Oracle，Sybase，Informix，Microsoft SQL，DB2 等。

这里提一下 InterBase 数据库操作系统。InterBase 本身是 Borland 公司的一个数据库产品，在 CB 产品的光盘中，Borland 为您提供了一个 InterBase ClientSever，让您在还没有决定使用何种数据库产品的时候，就可以在本地计算机上很方便的进行程序的开发，当您在程序开发完成后，再去选择合适的数据库产品。当然，InterBase 本身也是一个很不错的数据库产品，如果您对它的性能满意，可以再向 Borland 购买该产品。选用 InterBase 还有一个好处，那就是和 CB 的兼容性。毕竟是同一家公司的产品，而且 CB 中还包含有很多个专门用于访问 InterBase 数据库的组件，相信会为您的程序开发带来不少的方便。

选择何种数据库由很多不同因素决定，您必须综合考虑这些因素来选用合适的数据库系统。既要避免不必要的功能上的浪费，又要适应以后程序维护和用户使用的需求的变化，以及工程资金的能力，做到物尽所用。

9.3 C++Builder 中的一层结构

“一层结构”的数据库应用程序应该说是最简单的数据库服务系统了。如上文所述，使用本地数据库的应用程序被称为一级数据库应用程序，应用程序与数据库共享一个文件系统。由于本地数据库基于文件的特性，即便它可以被存放在网络上的一台服务器上，但那仅仅是一个共享文件而已，与其放在本地计算机上并没有什么本质上的不同。整个网络系统中并不存在一个 SQL 数据库服务器。这一点初学者很容易搞混，甚至一些介绍数据库编程的书籍上也常常搞错，经常认为只要数据是通过网络访问，就一定是二层或多层结构，而数据库系统或文件存在于本地计算机上，便是一层结构，实在是大错而特错。

一个系统是一层或二层(多层)，只取决于它使用数据库系统的类型，而与数据库文件或系统安装的位置没有什么关系。例如，您可以将一个本地数据库文件放在网络的某台计算机上共享，但它依然是一层结构；当您使用 InterBase 开发数据库应用程序时，为了方便，InterBase 一般都安装在本地计算机上，但这却是一个二层结构的数据库应用程序，因为这时存在一个 InterBase Sever，它本身就具有处理 SQL 命令及数据的能力。

在一级数据库应用程序中，应用程序及数据库共享同一个文件系统，它们使用本地数据库或者以 flat-file 格式存储数据库信息的文件。一级数据库应用程序包括用户接口及数据存取机制(BDE 或者用来装载及存储 flat-file 数据库信息的系统)，用来表示数据表的数据集组件类型依赖于数据是存储于本地数据库(如 Paradox、dBASE、Access 或者 FoxPro)还是存于 flat-file 文件中，图 9-1 说明了这两种可能。

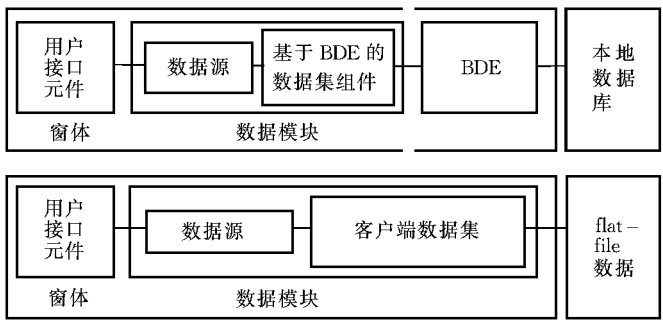

图 9-1 一层数据库应用程序结构

虽然一层数据库应用程序使用的是本地数据库，但这并不意味着它不能使用 SQL 结构化查询语言来编写程序。CB 提供了一个功能强大的 Borland 数据库引擎(BDE)，专门用来处理数据库的应用程序。在大多数情况下，CB 的应用程序不论是访问本地的数据库，还是远程数据库服务器，都必须通过这个 BDE 来进行连接。因此，BDE 在整个程序的开发过程中占有非常重要的地位。通过 BDE，CB 可以很轻易的处理 dBASE、Parodox 等文件数据。当然，您也可以通过 SQL Links 或 ODBC 处理远程数据库服务器的数据。BDE 还提供有一组 API(Application Program Interface)函数，您可以使用动态链接文件使用任何语言去调用它们。如果您在开发程序的过程中不想使用 CB 提供的数据访问组件(Data Access)，这些 API 函数也许就是最好的选择了。

另外，如果您使用了 BDE 开发一层和二层数据库应用程序，在程序发行时一定要将 BDE 和 C++Bulider 的程序文件一并安装在客户端的机器上，这样程序才能正常运行。

9.4　C＋＋Builder 中的二层结构

在二层结构的数据库应用程序的开发上，C＋＋Builder 一共提供了四种数据库应用程序的设计标准，它们分别是 BDE(Borland Database Engine)，ADO(ActivX Data Object)，dbExpress，Interbase Express(直接访问)。这四种数据库设计标准都可以应用在二层主从结构中。从 CB 组件面板的分类中您也可以很清楚的看到这一点。在这四个页面下的组件，有很多组件具有相似的名称，这也说明了它们在数据库应用程序中扮演着相类似的角色。

由图 9－2 可以看到，在二层结构的数据库应用程序中，CB 除了可以使用 BDE 来连接远程数据库服务器外，还可以使用 OLE DB 来连接远程数据库服务器。这就是 ADO 方式的二层主从结构。

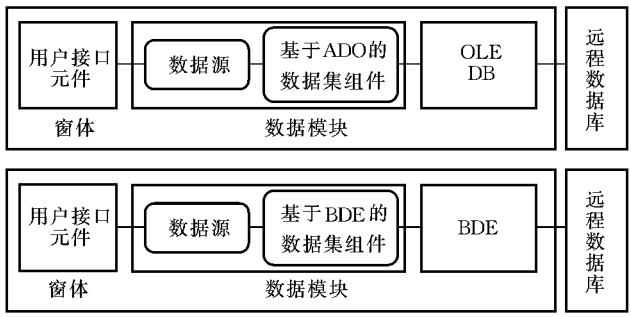

图 9－2　二层数据库应用程序结构

ADO 的全名是 ActiveX Data Object。Microsoft 以 COM 的标准实现 OLE DB，用来取代 ODBC 的角色，作为其数据库访问的中间软件，而 ADO 则是 OLE DB 的最上层，它是用来让 Windows 应用程序调用 OLE DB 的接口(Interface)。其实，ADO 只是 MDAC(Microsoft Data Access Components)的一小部分；MDAC 包含了 Microsoft 所有的数据库访问技术，如 ADO，OLE DB，ODBC，RDS(Remote Data Service)。所以，只要您安装了 MDAC 之后，上面所提到的各种数据库访问技术都会安装进来。为了您的方便，C＋＋Builder 6 会为您自动安装 MDAC2.5。当然您也可以自行安装 MDAC 的最新版本。

由于 ADO 也允许您通过 ODBC 的标准来访问服务器端数据库，所以，只要您有了与服务器端数据库对应的 ODBC 驱动程序，CB 就可以通过 ADO 及 ODBC 服务器端数据库了。BDE 提供的 SQL Links 可以连接六种服务器端数据库，包括 SYBASE，InterBase，ORACLE，MS－SQL，Informix，DB2。如果您还需要连接其它的服务器端数据库，可另外购买 ODBC Drivers 来连接。从控制面板中打开 BDE，在它的“Configration”页面下的“Configration”→“Drivers”→“Native”及 ODBC 节点中可以看到 CB 中已经安装的数据库驱动程序。由此可知，ADO 所支持的服务器端数据库不会少于 BDE。除了服务器端数据库，也就是 SQL 数据库外，MDAC 也提供 OLE DB 驱动程序可以直接连接 Access 及 MS－SQL 数据库。通过 OLE DB 专用的驱动访问 Access 及 MS－SQL 数据库，访问效率当然要比 ODBC 好。

由于 BDE 的访问效率不佳，SQL Link 驱动程序编写又不易(这是对 Borland 而言，而不是您的工作)，因此 C＋＋Builder 提出了另一种数据访问中间标准 dbExpress。Borland 针对各种数据库编写了对应的 dbExpress 本地驱动程序(dbExpress Native Driver)，并封装出一组 dbExpress 组件，让 CB 应用程序可以通过 dbExpress 组件及本地驱动程序去访问服务器端数据库的数据。如图 9－3 所示。

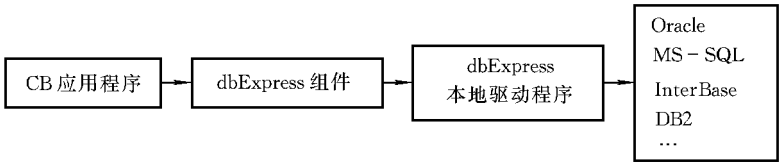

图 9－3 dbExpress 的二层主从结构

dbExpress 组件具有数据访问效率高、平台转移容易的优点，它比 BDE 的访问效率好很多。目前，已经有了 Linux 版本的 dbExpress。唯一遗憾的是 CB6 所提供的 dbExpress 本地驱动目前只支持 MS-SQL，InterBase，Oracle 和 DB2 四种。

CB 还提供一组 InterBase Express 组件让您可以直接、快速地访问 InterBase 数据库。InterBase Express 组件封装了 InterBase 所提供的 API 函数，其访问效率甚至高于 BDE 及 dbExpress 组件。这其中有一个原因应该是 InterBase 是 Borland 自己的产品，因而对于 InterBase数据库更了解，编写的驱动效率当然也就更高了。InterBase Express 组件的缺点就是它专用于 InterBase Server，不适用于其它远程数据库服务器的访问。

9.5 C++Builder 中的三层及多层结构

在两层的主从结构中，整个系统只包括客户端的 CB 应用程序和远程服务器端数据库系统。客户端应用程序包括了实现图形化的用户界面、用户输入数据的验证、送出查询的 SQL 命令给服务器端数据库等工作；而服务器端数据库操作系统则负责接收并响应客户端传来的 SQL 命令，并按照客户端需求访问存放实际数据的原始数据库，最后将执行 SQL 命令的结果数据传回客户端。因此，您所执行的进程实际上被切割成两部分，分别被放置在客户端计算机和服务器端计算机上运行，而这两部分工作量的大小，则可依照您的网络系统的具体特点而灵活划分。

由于在这种结构下，每一个客户端都和服务器端存在一个连接，因此当在线用户急剧增加时，服务器将穷于应付每个客户的请求和连接，最终仍将导致系统整体运行效率降低，尽管您已经将部分工作转移到了客户端的计算机上，但容量仍然是有限的。

前面讲到，二层结构的数据库应用程序发行时必须将 BDE，SQL Link，dbExpress 的一些 DLL 文件（如果有用到的话）安装到客户端的计算机上。虽然安装本身较易实现，但如果您的应用程序版本升级，您就必须到每个客户端计算机上安装更新后的程序，实在是一件烦恼的事情。

还有就是资金的问题。假如您的系统要满足 50 个客户端的访问，由于每个客户都需要一个连接，您是否就得花大价钱去购买 50 个客户访问服务器端数据库系统的用户授权，这可是一笔不小的费用。

二层主从结构的系统中，网上往往只有单一的数据库服务器，一旦死机，是不是整个数据服务系统就瘫痪了呢？即便您有一台备用机器，把它接入系统也得小费周折，无法保证系统的 24 小时连续运作。

针对上述问题，Borland 在 CB 中加入了三层及多层主从结构的数据库应用程序的开发能力，充分发挥了主从结构的分布式处理的特性，使您能够轻而易举的开发出多层分布式处理程序，为用户提供了全面的企业级解决方案。

三层及多层主从结构在二层主从结构的客户端和服务器端加入了一个中间层。如图 9－4中所示。这个中间层叫做“应用程序服务器”。服务些什么呢？基本上是专门负责处理

客户端的连接请求和传送服务器端的传回的数据。它把客户端计算机传来的请求(Request)传给数据库服务器,或将数据库服务器响应后的数据集(DataSet)返回给客户端计算机。这样,数据库服务器就只剩下了和应用程序服务器之间的连接,不用再去应付众多的连接请求,专心去处理数据库文件的访问操作了。如此将大幅度减轻数据库服务器的工作负担,进而提高系统的整体工作效率。

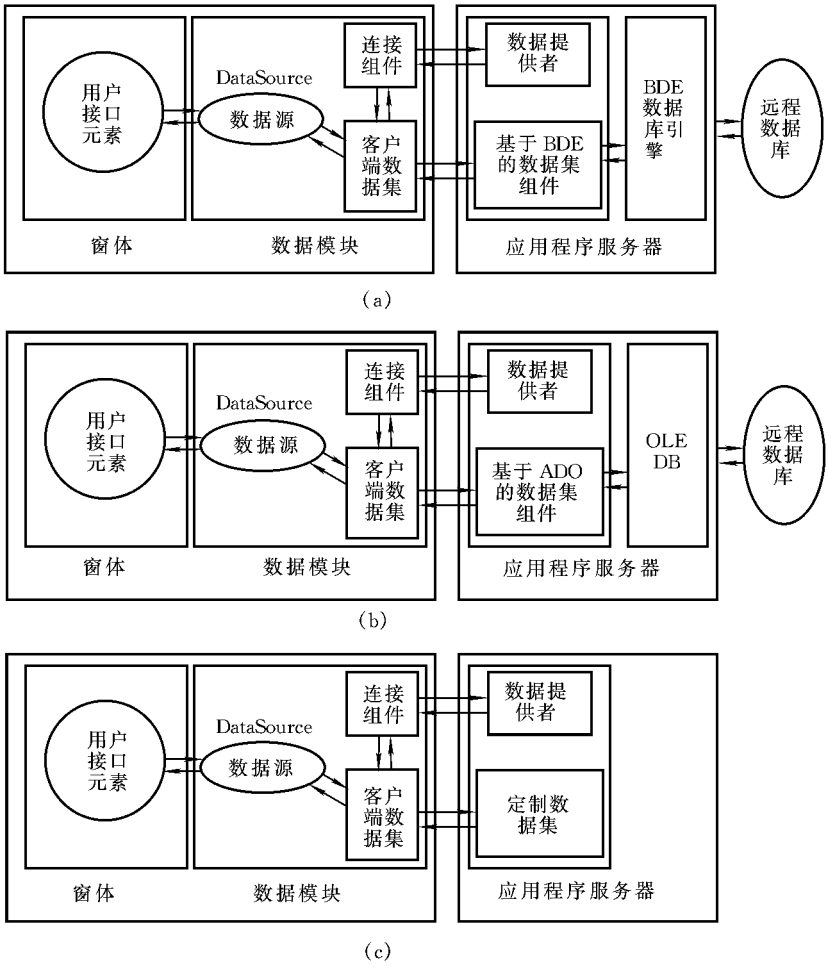

图 9－4　三层数据库应用程序结构

假如一个上千人的公司要召开员工大会,要是让全部员工同时跑到董事会那里去发表意见,那结果可想而知。所以只能选出班组长代表,让他们汇总员工的意见,再根据轻重缓急向董事会提出请求。这样,董事会才能最后制订出策略,再经过班组长执行下来。这里的董事会就相当于数据库服务器,班组长就相当于应用程序服务器,员工就相当于客户端计算机。比喻可能不太恰当,但能说明问题就行了。图 9－5 就是不同连接机制下的三层数据库应用程序的结构示意图。

三层及多层主从结构不仅解决了用户的访问问题,也解决了客户端安装程序和系统维护的困扰。在三层和多层主从结构下,CB 把原来安装在客户端的 DataModule,BDE,SQL Link Driver 都移到了应用程序服务器上,客户端只剩下了一个 User Interface(用户接口)程序和一个 MIDAS.Dll 文件。当您对程序升级时,只需要将修改后的可执行文件通过网络复制到客