

Adaptive Server Anywhere 入门手册

部件号: 32066-01-0802-01 上次修改时间: 2002 年 10 月 版权所有 © 1989-2003 Sybase, Inc. Portions 版权所有 © 2001-2003 iAnywhere Solutions, Inc. 保留所有权利。

事先未经 iAnywhere Solutions, Inc. 的书面许可,本出版物的任何部分不得以任何形式或通过任何方法(电子、机械、手工、光学或其他方法)进行复制、传输和翻译。 iAnywhere Solutions, Inc. 是 Sybase, Inc. 的子公司。

Sybase、SYBASE (徽标)、AccelaTrade、ADA Workbench、Adaptable Windowing Environment、Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, AnswerBase, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Library, APT-Translator, ASEP, Backup Server, BayCam, Bit-Wise, BizTracker、 Certified PowerBuilder Developer、 Certified SYBASE Professional、 Certified SYBASE Professional (徽标)、 ClearConnect, Client Services, Client-Library, CodeBank, Column Design, ComponentPack, Connection Manager, Convoy/DM、Copernicus、CSP、Data Pipeline、Data Workbench、DataArchitect、Database Analyzer、DataExpress、 DataServer、DataWindow、DB-Library、dbQueue、Developers Workbench、Direct Connect Anywhere、DirectConnect、Distribution Director、Dynamo、e-ADK、E-Anywhere、e-Biz Integrator、E-Whatever、EC-GATEWAY、ECMAP、 ECRTP, eFulfillment Accelerator, Electronic Case Management, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server、Enterprise Connect、Enterprise Data Studio、Enterprise Manager、Enterprise SQL Server Managers Enterprise Work Architectures Enterprise Work Designers Enterprise Work Modelers eProcurement Accelerators eremote, Everything Works Better When Everything Works Together, EWA, Financial Fusion, Financial Fusion Server, First Impression, Formula One, Gateway Manager, GeoPoint, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InstaHelp、Intellidex、InternetBuilder、iremote、iScript、Jaguar CTS、jConnect for JDBC、KnowledgeBase、Logical Memory Manager, MainframeConnect, Maintenance Express, MAP, MDI Access Server, MDI Database Gateway, media.splash、MetaWorks、MethodSet、ML Query、MobiCATS、MySupport、Net-Gateway、Net-Library、New Era of Networks、Next Generation Learning、Next Generation Learning Studio、ODEVICE、OASiS、OASiS (徽标)、 ObjectConnect, ObjectCycle, OmniConnect, OmniSOL Access Module, OmniSOL Toolkit, Open Biz, Open Business Interchange, Open Client, Open Client/Server, Open Client/Server Interfaces, Open ClientConnect, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net、 PC Net Library、 Physical Architect、 Pocket PowerBuilder、 PocketBuilder、 Power Through Knowledge、 Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, Powering the New Economy, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, Powersoft Portfolio, Powersoft Professional, PowerStage, PowerStudio, PowerTips, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, Relational Beans, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report Workbench, Report-Execute, Resource Manager, RW-DisplayLib, RW-Library, S Designor, S-Designor, S.W.I.F.T. Message Format Libraries, SAFE, SAFE/PRO, SDF、Secure SQL Server、Secure SQL Toolset、Security Guardian、SKILS、smart.partners、smart.parts, smart.script、 SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/ TPU、SQL Everywhere、SQL Modeler、SQL Remote、SQL Server、SQL Server Manager、SQL Server SNMP SubAgent, SQL Server/CFT, SQL Server/DBM, SQL SMART, SQL Station, SQL Toolset, SQLJ, Stage III Engineering, Startup.Com, STEP, SupportNow, Sybase Central, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Financial Server, Sybase Gateways, Sybase Learning Connection, Sybase MPP, Sybase SQL Desktop、Sybase SQL Lifecycle、Sybase SQL Workgroup、Sybase Synergy Program、Sybase User Workbench、Sybase Virtual Server Architecture、SybaseWare、Syber Financial、SyberAssist、SybMD、SyBooks、System 10、System 11、 System XI(徽标)、SystemTools、 Tabular Data Stream、 The Enterprise Client/Server Company、 The Extensible Software Platform, The Future Is Wide Open, The Learning Connection, The Model For Client/Server Solutions, The Online Information Center, The Power of One, TradeForce, Transact-SQL, Translation Toolkit, Turning Imagination Into Reality, UltraLite, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, WarehouseArchitect, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights、WebViewer、WorkGroup SQL Server、XA-Library、XA-Server 和 XP Server 都是 Sybase, Inc. 或其子公司的 商标。

所有其它商标都是其各自所有者的财产。

目录

	关于本手册	vii
1	Adaptive Server Anywhere 快速入门	1
	第1步: 启动 Adaptive Server Anywhere 数据库服务器	2
	第2步: 启动 Sybase Central	4
	第3步: 启动 Interactive SQL	6
2	数据库和应用程序	9
	关系数据库概念	10
	SQL 和数据库计算	15
	数据库系统的组成部分	18
	各组件是如何协同工作的	20
3	Adaptive Server Anywhere 简介	23
	Adaptive Server Anywhere 简介	24
	Adaptive Server Anywhere 用途	25
	Adaptive Server Anywhere 特点	26
	Adaptive Server Anywhere 数据库服务器	29
	Adaptive Server Anywhere 应用程序	
4	数据库应用程序的体系结构	
	应用程序编程接口	36
	Adaptive Server Anywhere 内在特性	42
5	设计和建立数据库	
	简介	46
	示例数据库	48
	教程:设计和建立简单数据库	53

6	将您的应用程序连接到其数据库65 连接简介		
	创建 ODBC 数据源	67	
7	使用 Interactive SQL		
	Interactive SQL 介绍		
	启动 Interactive SQL		
	使用 Interactive SQL 显示数据	80	
	在 Interactive SQL 中使用 SQL 语句	85	
	配直 Interactive SQL		
	在 Interactive SQL 中使用 SQL 转义语法	100	
8	从数据库表中选择数据	103	
	介绍	104	
	选择完整的表	106	
	从表中选择列	108	
	查询结果排序	111	
	从表中选择行	115	
9	从多个表中选择数据	123	
	介绍	124	
	使用矢量积连接表	126	
	使用 ON 短语限制连接	127	
	使用键连接 (key join) 来连接表		
	使用自然连接 (natural join) 来连接表	131	
	使用外连接来连接表	133	
10	选择集合数据		
	近总数据		
	集合函数简介		
	将集合函数应用于分组数据		
	限制组	140	
11	使用子查询选择数据		
	子查询介绍		
	介绍	145	

单行和多行子查询	148
使用子查询代替连接	150

12

更新数据库	
简介	
将行添加到表中	
修改表中各行	
删除行	
将更改组合到事务中	
完整性检查	

13	系统表	
	系统表	
	SYSCATALOG 视图	
	SYSCOLUMNS 视图	
	其它系统表	
14	Microsoft Visual Basic 快速入门 教程:开发 Visual Basic 应用程序	171 172
	术语表	175

关于本手册

主题	本手册介绍如何使用 Adaptive Server Anywhere 建立简单的数据库和数 据库应用程序。	
读者	本手册供 Adaptive Server Anywhere 的初级用户使用。	
预备知识	本手册假设您熟悉一些基本的编程概念。它还假设您了解 Adaptive Server Anywhere 所在操作系统的应用知识。	
SQL Anywhere Studio 文档	本手册是 SQL Anywhere 文档集的一部分。本节介绍该文档集中的手册 以及如何使用它们。	
SQL Anywhere Studio	SQL Anywhere Studio 文档集由以下手册组成:	
人妇果	• SQL Anywhere Studio 介绍 : 这本书概述 SQL Anywhere Studio 数据库管理和同步技术。它包括的教程向您介绍 SQL Anywhere Studio 的每个组成部分。	
	• SQL Anywhere Studio 的新功能 : 这本书供该软件以前版本的用 户使用。它列出了本产品此版本和以前版本的新功能并介绍了升 级过程。	
	• Adaptive Server Anywhere 入门手册 : 这本书供不熟悉关系数据 库或 Adaptive Server Anywhere 的新用户使用。它提供了使用 Adaptive Server Anywhere 数据库管理系统的快速入门以及设计、 建立和使用数据库的介绍材料。	
	• Adaptive Server Anywhere 数据库管理指南 : 这本书涵盖了运行、管理和配置数据库的相关材料。	
	 Adaptive Server Anywhere SQL 用户指南:这本书介绍如何设 计和创建数据库;如何导入、导出和修改数据;如何检索数据; 以及如何建立存储过程和触发器。 	
	• Adaptive Server Anywhere SQL 参考手册 : 这本书提供 Adaptive Server Anywhere 使用的 SQL 语言的完整参考。它还介绍了	

Adaptive Server Anywhere 系统表和过程。

- Adaptive Server Anywhere 编程指南 : 这本书介绍如何使用 C、 C++和 Java 编程语言建立和部署数据库应用程序。Visual Basic 和 PowerBuilder 等工具的用户可以使用这些工具提供的编程接口。
- Adaptive Server Anywhere 错误消息 : 这本书提供 Adaptive Server Anywhere 错误消息以及诊断信息的完整列表。
- Adaptive Server Anywhere C2 安全性补充说明: Adaptive Server Anywhere 7.0 被美国政府授予 TCSEC (可信计算机系统评估标 准) C2 安全性等级。对于那些希望以相当于 C2 认证环境的方式 运行当前版本的 Adaptive Server Anywhere 的用户,他们可能会对 这本书很感兴趣。这本书未 包括该产品获得认证后新增的安全特 性部分。
- MobiLink 同步技术用户指南: 这本书介绍进行移动计算的 MobiLink 数据同步系统的所有方面,该系统支持在单个 Oracle、 Sybase、Microsoft 或 IBM 数据库和许多 Adaptive Server Anywhere 或 UltraLite 数据库之间共享数据。
- SQL Remote 用户指南:这本书介绍进行移动计算的
 SQL Remote 数据复制系统的所有方面,该系统支持使用间接链接 (例如电子邮件或文件传输)在单个 Adaptive Server Anywhere 或
 Adaptive Server Enterprise 数据库和许多 Adaptive Server Anywhere 数据库之间共享数据。
- UltraLite 用户指南: 这本书介绍如何使用 Adaptive Server Anywhere 数据库的 UltraLite 部署技术为小型设备(如手持管理器)建立数据库应用程序。
- **UltraLite User's Guide for PenRight! MobileBuilder** : 这本书供 PenRight! MobileBuilder 开发工具的用户使用。它介绍如何在 MobileBuilder 编程环境中使用 UltraLite 技术。
- SQL Anywhere Studio 帮助:这本书只联机提供。它包括 Sybase Central、Interactive SQL 和其它图形工具的上下文相关帮助。

除了此文档集之外, SQL Modeler 和 InfoMaker 还包含它们自己的联机 文档。

文档格式 SQL Anywhere Studio 提供以下格式的文档:

• 联机手册 联机手册包括完整的 SQL Anywhere Studio 文档,包括 SQL Anywhere 工具的打印手册和上下文相关帮助。产品的每个维 护版本都将对联机手册进行更新,联机手册是最完整、最新的文 档源。

若要在 Windows 操作系统上访问联机手册,请选择"开始"→"程序"→"Sybase SQL Anywhere 8"→"联机手册"。您可以使用 左窗格中的 HTML 帮助目录、索引和搜索功能并使用右窗格中的 链接和菜单导航联机手册。

若要在 UNIX 操作系统上访问联机手册,请在命令提示符处运行 以下命令:

dbbooks

• **可打印手册** SQL Anywhere 手册是以一系列 PDF 文件的形式提供的,可以使用 Adobe Acrobat Reader 进行查看。

可以在 CD ROM 上 pdf_docs 目录中找到这些 PDF 文件。您可以 在运行安装程序时选择安装这些文件。

- 印刷手册 SQL Anywhere Studio 盒中附带以下手册:
 - SQL Anywhere Studio 介绍。
 - Adaptive Server Anywhere 入门手册。
 - SQL Anywhere Studio 快速参考手册。这本书只有印刷版本。

SQL Anywhere 整套文档集可在 Sybase 销售部门或 Sybase 网上商店 e-Shop 购买, 网址是: http://e-shop.sybase.com/eshop)。

文档约定 本节列出文档中使用的字体和图形约定。

语法约定 以下约定用于 SQL 语法说明:

• 关键字 所有 SQL 关键字像下面示例中单词 ALTER TABLE 那样显示:

ALTER TABLE [owner.]table-name

• **占位符** 必须用适当的标识符或表达式替换的项像下面示例中 owner 和 table-name 那样显示。

ALTER TABLE [owner.]table-name

• **重复项** 重复项列表显示为列表中的一个元素后接省略号(三个 点),就像下面示例中 column-constraint 那样:

ADD column-definition [column-constraint, ...]

允许使用一个或多个列表元素。如果指定了多个元素,则必须用 逗号将它们分开。

• 可选部分 语句的可选部分是用方括号括起来的。

RELEASE SAVEPOINT [savepoint-name]

该方括号表示 savepoint-name 是可选的。不应该键入方括号。

• **选项** 当不必选择或只能选择一个项目列表时,用垂直条隔开项 并且列表括在方括号中。

[ASC | DESC]

例如,您可以选择 ASC 和 DESC 中的一项或者两项都不选。不应 该键入方括号。

 二选一选项 当必须选择两个选项中的一个时,二选一选项括在 花括号中。

[QUOTES { ON | OFF }]

如果选择了 QUOTES 选项,则必须提供 ON 或 OFF 中的一个。不 应该键入方括号和花括号。

一个或多个选项 如果您选择多个选项,请用逗号分隔您的选择。

{ CONNECT, DBA, RESOURCE }

图标 本文档中使用了下列图标:

图标	意义		
	客户端应用程序。		
	数据库服务器,例如 Sybase Adaptive Server Anywhere 或 Adaptive Server Enterprise。		
	UltraLite 应用程序和数据库服务器。在 UltraLite 中, 数据库服务器和应用程序是同一个进程的一部分。		
	数据库。在某些高级别的图中,该图标可能用于表示 数据库和管理该数据库的数据库服务器。		
	复制或同步中间件。这些中间件协助在数据库间共享数据。中间件的示例包括 MobiLink Synchronization Server、SQL Remote Message Agent 以及与 Replication Server 一起使用的 Replication Agent (Log Transfer Manager)。		



Adaptive Server Anywhere 示例数据 库 整个文档中有多个示例使用 Adaptive Server Anywhere 示例数据库。

示例数据库保存在一个名为 asademo.db 的文件中,位于您的 SQL Anywhere 目录中。

示例数据库描述一个小型公司。它包含有关该公司的内部信息(职员、 部门和财务情况),以及产品信息和销售信息(销售订单、客户和合同)。数据库中的所有信息都是虚构的。

下图显示示例数据库中的表以及它们是如何互相关联的。



查找更多信息和提供 我们欢迎您就本文档提出意见、建议或其它反馈信息。

反馈

您可以通过为讨论 SQL Anywhere 技术设置的新闻组提供有关此文档和 软件的反馈。可以在 forums.sybase.com 新闻服务器上找到这些新闻 组。

包括以下新闻组:

- sybase.public.sqlanywhere.general.
- sybase.public.sqlanywhere.linux。
- sybase.public.sqlanywhere.mobilink。
- sybase.public.sqlanywhere.product_futures_discussion。
- sybase.public.sqlanywhere.replication.
- sybase.public.sqlanywhere.ultralite.

新闻组免责声明

iAnywhere Solutions 没有义务为其新闻组提供解决方案、信息或建议;除提供系统操作员监控服务和确保新闻组的运行和可用性外, iAnywhere Solutions 没有义务提供任何其它服务。

如果时间允许, iAnywhere Solutions 技术顾问和其他员工也会提供新闻 组服务。他们是在自愿的基础上提供帮助,并且不能定期提供解决方案 和信息。他们可以提供多少帮助取决于他们的工作量。

^{第1章} Adaptive Server Anywhere 快速入门

关于本章本章介绍如何启动 Adaptive Server Anywhere 数据库服务器以及从
Sybase Central 和 Interactive SQL 连接到示例数据库。

本章供那些相对熟悉数据库并想马上使用该软件的用户使用。

有关数据库特别是 Adaptive Server Anywhere 的介绍性说明,请参阅从 "数据库和应用程序 "第9页开始的几章。

第1步: 启动 Adaptive Server Anywhere 数据库服务器

在本节中,您将启动运行示例数据库的 Adaptive Server Anywhere 数据 库服务器。

Adaptive Server Anywhere 包括两个版本的数据库服务器。个人数据库服务器供单一用户在同一台计算机上使用。网络数据库服务器支持通过网络的客户/服务器通信,供多个用户进行操作。两种数据库服务器在其查询处理和其它内部操作方面是完全等同的:个人数据库服务器在功能上和网络数据库服务器一样强大。

Adaptive Server Anywhere 数据库保存在操作系统文件中。示例数据库保存在 SQL Anywhere 安装目录中名为 asademo.db 的文件中。

✤ 启动运行示例数据库的个人数据库服务器 (Windows):

在 " 开始 " 菜单中, 选择 " 程序 " → "Sybase SQL Anywhere 8" →
 "Adaptive Server Anywhere" → " 个人服务器示例 "。

即会出现数据库服务器窗口。

◆ 启动运行示例数据库的数据库服务器 (命令行):

- 1 打开命令提示。
- 2 将目录更改为 SQL Anywhere 安装目录。

在 Windows 操作系统上,缺省安装目录是 C:\Program Files\Sybase\SQL Anywhere 8。

3 启动运行示例数据库的数据库服务器。

启动数据库服务器的方式取决于您的操作系统以及您是否要从 网络上的其它计算机连接到该数据库。

 如果只想从同一台计算机的 Windows 或 UNIX 操作系统 上进行连接,则输入以下命令启动个人数据库服务器:

dbeng8 asademo.db

• 如果要从网络上其它计算机的 Windows 或 UNIX 操作系 统连接到数据库服务器,则输入以下命令启动网络数据库 服务器:

dbsrv8 asademo.db

• 在 NetWare 上,只提供网络数据库服务器。输入以下命 令:

load dbsrv8.nlm asademo.db

即会出现数据库服务器窗口。

有关启动数据库服务器时可以使用的选项的完整列表,请参阅 "ASA 数据库管理指南 > 数据库服务器 > 数据库服务器 "。

第2步: 启动 Sybase Central

在本节中,您将启动图形数据库管理工具 Sybase Central。 这些说明假设您已经完成了 " 第1步: 启动 Adaptive Server Anywhere 数 据库服务器 " 第 2 页中的操作。

- ✤ 启动 Sybase Central 并连接到示例数据库 (Windows):
 - 1 选择 " 开始 " → " 程序 " → "Sybase SQL Anywhere 8" → "Sybase Central"。
 - 2 连接到示例数据库。
 - 选择"工具"→"连接"。
 - 从插件列表中选择 "Adaptive Server Anywhere 8", 然后单击 "确定 "。
 - 在 "数据库 "选项卡上,输入服务器名称 asademo。该名称 标识您在上一节中启动的数据库服务器。
 - 在 "标识 "选项卡上,输入用户 ID DBA 和口令 SQL。
 - 单击"确定"进行连接。
- ✤ 启动 Sybase Central 并连接到示例数据库(命令行):
 - 1 在命令提示符处,输入以下命令:

scjview

即会出现主 "Sybase Central" 窗口。

- 2 连接到示例数据库。
 - 选择"工具"→"连接"。
 - 从插件列表中选择 "Adaptive Server Anywhere 8", 然后单击 "确定 "。

- 在 " 数据库 " 选项卡上, 输入服务器名称 asademo。该名称 标识您在上一节中启动的数据库服务器。
- 在 "标识 "选项卡上,输入用户 ID DBA 和口令 SQL。
- 单击"确定"进行连接。

您现在可以浏览示例数据库中的表和其它对象。

使用 Sybase Central,您可以执行许多数据库管理任务,包括创建数据 库、备份数据库、创建表和其它数据库对象以及修改数据库表中的数 据。

有关 Sybase Central 的概览,请参阅 "SQL Anywhere Studio 介绍 > 教程: 使用 Sybase Central 管理数据库 "。

第3步: 启动 Interactive SQL

在本节中,您将启动 Interactive SQL 实用程序、连接到示例数据库并输入命令。

◇ 启动 Interactive SQL 并连接到示例数据库 (Windows):

- 1 启动 Interactive SQL:
 - 选择 " 开始 " → " 程序 " → "Sybase SQL Anywhere 8" → "Adaptive Server Anywhere" → "Interactive SQL"。

即会出现 "连接 "对话框。

- 2 连接到示例数据库。
 - 在 "数据库 "选项卡上,输入服务器名称 asademo。该名称 标识您在 "第1步: 启动 Adaptive Server Anywhere 数据库 服务器 "第2页中启动的数据库服务器。
 - 在 "标识 "选项卡上,输入用户 ID DBA 和口令 SQL。
 - 单击"确定"进行连接。

即会出现 "Interactive SQL" 窗口。

3 输入一个命令。

在 "SQL 语句 " 窗格中, 输入 SQL 语句并按 F5 键执行该查询, 如下所示:

SELECT * FROM CUSTOMER

查询的结果集出现在 "结果 "窗格中。

✤ 启动 Interactive SQL 并连接到示例数据库(命令行):

1 在命令提示符处,输入以下命令:

dbisql

即会出现"连接"对话框。

- 2 连接到示例数据库。
 - 在 "数据库 "选项卡上,输入服务器名称 asademo。该名称标识您在 "第1步: 启动 Adaptive Server Anywhere 数据库服务器 "第2页中启动的数据库服务器。
 - 在 "标识 "选项卡上,输入用户 ID DBA 和口令 SQL。
 - 单击"确定"进行连接。

即会出现 "Interactive SQL" 窗口。

3 输入一个命令。

在 "SQL 语句 " 窗格中, 输入如下 SQL 语句并按 F5 键执行该查 询:

SELECT * FROM CUSTOMER

该查询的结果集显示在 "结果 " 窗格中。

您可以在 Interactive SQL 中对数据库输入任何 SQL 语句。有关 SQL 语句的完整列表,请参阅 "ASA SQL 参考手册 > SQL 语句 "。

有关 Interactive SQL 的更多信息,请参阅 " 使用 Interactive SQL" 第 71 页。

^{第2章} 数据库和应用程序

关于本章 本章介绍一些基本的数据库概念。它介绍什么是关系数据库以及关系数 据库的用途。本章还介绍数据库和数据库应用程序是如何协同工作的。

文档集的其它部分假定您熟悉本章中介绍的概念。

关系数据库概念

关系数据库管理系统 (Relational Database-Management System, RDBMS) 是用于存储和检索数据的系统,在该系统中,数据被组织到表中。关系数据库由存储相关数据的表的集合组成。

本节介绍一些在谈论关系数据库时十分重要的术语和概念。

有关阐明本章中描述的各个概念的教程,请参阅 "设计和建立数据库 " 第 45 页。

数据库表

在关系数据库中,所有数据都存储在表中,而表是由行和列组成的。



每个表都具有一列或多列,并且每一列被指派了特定数据类型,例如整 数、字符序列(对于文本)或日期。表中每一行对应于每列都有一个 值。

例如,包含职员信息的表可能如下所示:

emp_ID	emp_Iname	emp_fname	emp_phone
10057	Huong	Zhang	1096
10693	Donaldson	Anne	7821

关系表的特性 关系数据库的表具有某些重要特性:

- 列或行的顺序没有任何意义。
- 每一行为每一列包含一个且只包含一个值,或者包含空值,指示 对于该列没有任何值。
- 给定列的所有值都具有相同的数据类型。

下表列出了某些描述表及其内容的正式和非正式关系数据库术语,同时列出了它们在非关系数据库中的对等术语。本手册使用非正式术语。

非正式关系术语	正式关系术语	非关系术语
表	关系	文件
列	属性	字段
行	元组	记录

在每一表中保留些什 数据库中的每个表都应保存与特定事物 (例如职员、产品或客户) 有 **么?** 关的信息。

> 通过按此方法设计数据库,可以建立一个结构,该结构消除了冗余并消除了因冗余导致的可能的不一致情况。例如,销售和应付帐户部门都可以输入和查询与客户有关的信息。在关系数据库中,与客户有关的信息 只存储一次,并且存储在两个部门都可以访问的表中。

有关数据库设计的更多信息,请参阅 "ASA SQL 用户指南 > 设计数据库 "。

有关如何创建表的说明,请参阅 " 第 3 课: 设计和创建表 " 第 56 页。

表间的关系

关系数据库是一组相关的表。您使用主键和外键描述不同表中各信息间 的关系。**主键**唯一标识表中每一行,**外键**定义不同表中各行间的关系。

通过主键和外键,您可以以高效 (无冗余)的方式使用关系数据库保存信息。

具有主键的表

关系数据库中的每个表都应该具有主键。主键是唯一标识每一行的一列或列集合。表中不能有两个行具有相同的主键值。

示例

在示例数据库中, employee 表存储与职员有关的信息。它具有名为 emp_id 的主键列,保存指派给每一职员的唯一 ID 号。保留 ID 号的单列 是指派主键的常见方式,并且此方式优于名称和其它标识符,因为它们 不一定始终是唯一的。

在示例数据库的 sales_order_items 表中可以看到更复杂的主键。该表保存 与公司订单上各单独项有关的信息,并且具有以下列:

- id 订单号,标识某项所属于的订单。
- line_id 行号,标识任一订单上的每一项。
- **prod_id** 产品 ID,标识所订购的产品。
- quantity 数量,显示订购了多少项。
- ship_date 发货日期,显示发货的时间。

特定销售订单项是由它所属于的订单以及该订单上的行号标识的。这两个数字存储在 id 和 line_id 列中。多个项可以共享一个 id 值 (为多个项 对应于一个订单),或者它们可以共享一个 line_id 号 (不同订单上的所 有第一个项的 line_id 均为 1)。但没有两个项同时共享这两个值,因此 主键由这两列组成。

有关教程示例,请参阅"第4课:标识和创建主键"第57页。

表是由外键关联的

一个表中的信息可以通过外键与其它表中的信息关联。

示例 示例数据库具有一个保存职员信息的表和一个保存部门信息的表。 department 表具有以下列:

- **dept_id** 部门的 ID 号。这是该表的主键。
- **dept_name** 部门的名称。
- **dept_head_id** 部门经理的职员 ID。

为了查找特定职员的部门的名称,不需要将该职员的部门名称放入 *employee* 表中。而是在 *employee* 表包含一列,该列保存与部门列中某一 *dept_id* 值的数字匹配的数字。

employee 表中的 *dept_id* 列被认为是 *department* 表的外键。外键引用包含 相应主键的表中的特定行。

在此示例中, employee 表 (包含关系中的外键)称作**外表**或**引用表**。 department 表 (包含引用的主键)称作**主表**或被引用表。

有关教程示例,请参阅"第5课:设计列属性"第59页。

其它数据库对象

关系数据库保存多组相关的表。以下是构成关系数据库的其它对象:

 索引 索引有助于加快信息查询。从概念上讲,数据库中的索引 与书籍中的索引类似。在书籍中,索引将每一个编入索引的术语 与该词语出现的一页或几页相关联。在数据库中,索引将每一编 入索引的列值与包含该索引值的数据行的物理存储位置相关联。

索引是有助于获得高性能的重要设计元素。您通常必须显式创建 索引,但主键和外键的索引以及唯一列的索引是自动创建的。索 引一旦创建后,索引的使用对用户而言就是不可察觉的。

 视图 视图是计算出的表,即虚拟表。对于客户端应用程序而言, 视图与表类似,但它们不保存数据。而是只要访问视图,就会根据基表计算视图中的信息。

实际保存信息的表有时候称为基表,以将它们和视图区分开来。 视图是由对基表或其它视图的 SQL 查询定义的。

• **存储过程和触发器** 它们是在数据库本身中保存的例程,操作数 据库中的信息。

您可以创建并命名自己的存储过程,以执行特定数据库查询和其 它数据库任务。存储过程可以带有参数。例如,您可以创建一个 存储过程,它返回花费超出指定金额的所有客户的名称。

触发器是只要用户更新、删除或插入数据(根据定义触发器的方式)就自动触发的特殊存储过程。您将触发器与一个表或表内的 列关联。触发器用于自动维护数据库中的业务规则。

- **用户和组**数据库的每一用户都具有用户 ID 和口令。可以为每一 用户设置权限,以便保护机密信息的安全并防止用户进行未授权 的更改。可以将用户指派给组,以便使权限管理更简单。
- Java 对象 可以将 Java 类安装到数据库中。Java 类提供一个强有 力的方法,以生成数据库中的逻辑,以及为存储信息生成用户定 义的数据类型的特殊类。

SQL 和数据库计算

示例

当客户端应用程序想要执行一个数据库任务 (例如使用查询检索信息 或将新行插入表中)时,它使用**结构化查询语言** (Structured Query Language, SQL) 语句执行这样的任务。 SQL 是已由 ANSI 和 ISO 标准化 机构标准化的关系数据库语言。

根据您开发客户端应用程序的方式,可以在以编程语言编写的函数调用 中提供 SQL 语句。某些应用程序开发工具为建立和生成 SQL 语句提供 用户界面。

编程接口将 SQL 语句传递给数据库服务器。数据库服务器接收该语句 并执行它,并且将必需的信息(例如查询结果)返回到应用程序。

客户 / 服务器通信协议在客户端应用程序和数据库服务器之间传送信息,编程接口定义应用程序发送这些信息的方式。无论您使用何种接口以及使用哪一网络协议, SQL 语句都被发送到服务器,并且 SQL 语句的结果被返回到客户端应用程序。

示例 此 SQL 语句从示例数据库的 employee 表中提取所有职员的姓氏:

SELECT emp_lname FROM employee

您可以如上例所示使用 Interactive SQL 将查询发送到数据库服务器,或 者可以将查询装入自己的应用程序。

此 SQL 语句创建称为 food 的表,该表列出食品类型以及职员餐厅内的 库存量:

CREATE TABLE food (
 food_id integer primary key,
 food_type char(20),
 quantity integer
)

有关对 SQL 的说明,请首先自 ""从数据库表中选择数据 " 第 103 页 " 开始,查看各章。

查询

"SQL"中的 "Q" 表示**查询** (Query)。您使用 SELECT 语句从数据库中查询 (或检索)数据。关系系统中的基本查询操作是投影、限制和连接。SELECT 语句执行上述所有操作。

投影和限制 投影是表中各列的子集。**限制**(也称作**选择**)是表中各行的子集(基于某些条件)。

例如,以下 SELECT 语句检索成本超过 \$15 的所有产品的名称和价格:

SELECT name, unit_price FROM product WHERE unit_price > 15

此查询同时使用投影 (SELECT name, unit_price) 和限制 (WHERE unit_price > 15)。

有关更多信息,请参阅 "从表中选择行 " 第 115 页。

连接 连接通过比较每一表中各列的值,将两个或更多表中的各行关联起来。 例如,您可能想要为发运的商品件数超过一打的所有订单项选择订单项 标识号和产品名称:

SELECT sales_order_items.id, product.name
FROM product JOIN sales_order_items
WHERE sales_order_items.quantity > 12

product 表和 sales_order_items 表基于两个表间的外键关系连接在一起。

有关更多信息,请参阅 "从多个表中选择数据 "第 123 页一章。

其它 SQL 语句

除了查询以外,您还可以使用 SQL 进行许多操作。 SQL 包括创建表、视图和其它数据库对象的语句。它还包括修改表的语句(INSERT、 UPDATE 和 DELETE 语句)以及执行在本手册中讨论的其它许多数据 库任务的语句。

系统表

每一 Adaptive Server Anywhere 数据库都包含一组系统表。有一些系统 用来管理数据和系统的特殊表。这些表有时也称作**数据字典**或**系统目** 录。

系统表包含与数据库有关的信息。您永远不可以像更改其它表那样直接 更改系统表。系统表保存与数据库中各表、数据库的用户、每一表中的 各列等有关的信息。这些信息是与各种数据有关的数据,即**元数据**。

数据库系统的组成部分

本节描述数据库应用程序和数据库服务器如何协同工作以管理数据库的。

任何信息系统都包含以下组成部分:

• **数据库** 数据存储在数据库中。在本文档的各图示中,数据库是 用圆柱表示的:



Adaptive Server Anywhere 数据库是一个文件,通常以.db 作为扩展名。 Adaptive Server Anywhere 包括可供您使用的示例数据库: 该示例数据库是位于 Adaptive Server Anywhere 安装目录中的 asademo.db 文件。

有关如何创建数据库的说明,请参阅 "设计和建立数据库 "第45页。

数据库服务器 数据库服务器管理数据库。没有其它任何应用程序直接处理数据库文件;它们全都与数据库服务器通信。

有关启动数据库服务器的说明,请参阅 "将您的应用程序连接到其数据库 "第 65 页。

在本文档的各图示中,数据库服务器如下所示:



Adaptive Server Anywhere 提供两个数据库服务器版本: 个人数据 库服务器和网络数据库服务器。网络数据库服务器除具有个人服 务器的功能外,还支持网络上的客户/服务器通信,而个人数据 库服务器只能接受来自在同一计算机上运行的应用程序的连接。

请求处理引擎在这两种服务器上是等同的。

有关更多信息,请参阅 "Adaptive Server Anywhere 数据库服务器 " 第 29 页。

 编程接口 应用程序使用编程接口与数据库服务器通信。您可以 使用 ODBC、 JDBC、 OLE DB、 Sybase Open Client 或嵌入式 SQL。

许多应用程序开发工具提供自己的编程环境,该环境隐藏基础接口的细节。例如,如果使用 Sybase PowerBuilder 开发应用程序,则永远不必进行 ODBC 函数调用。不过,在这些工具的内部结构中,每一工具都使用这些编程接口之一。

编程接口提供函数调用库,以用于与数据库通信。对于 ODBC 和 JDBC,该库通称为**驱动程序**。该库在 UNIX 操作系统上通常作为 共享库提供,在 PC 操作系统上作为动态链接库 (Dynamic Link Library, DLL) 提供。 JDBC 接口使用 Sybase jConnect 驱动程序, 它是编译的 Java 类的压缩文件。

在本文档的各图示中,编程接口如下所示:



客户端应用程序 客户端应用程序使用编程接口之一与数据库服 务器通信。

如果使用快速应用程序开发 (Rapid Application Development, RAD) 工具 (例如 Sybase PowerBuilder)开发应用程序,您会发现该工 具提供自己的方法来与数据库服务器通信,并且隐藏语言接口的 细节。不过,所有应用程序确实使用某一支持的接口。

在本文档的各图示中,客户端应用程序由以下图标表示:



各组件是如何协同工作的

数据库应用程序可以连接到位于该应用程序自身所在的计算机上的数据 库服务器,或者连接到不同计算机上的网络数据库服务器。此外,使用 Adaptive Server Anywhere,您可以建立分发数据库,数据库在共享数据 的不同计算机上有着实质上的区别。

个人应用程序和嵌入式数据库

您可以使用 Adaptive Server Anywhere 在单个计算机上建立完整的应用 程序和数据库。最简单的完整应用程序或数据库是**独立应用程序**或**个人 应用程序**:这两种应用程序是不与其它数据库连接的独立的应用程序。 在此情况下,数据库服务器和数据库可以由客户端应用程序启动,并且 通常将该数据库称作**嵌入式数据库**:就最终用户而言,该数据库是应用 程序的一部分。

独立应用程序具有以下体系结构,客户端应用程序通过编程接口连接到 在同一计算机上运行的数据库服务器:



Adaptive Server Anywhere 个人数据库服务器通常用于独立的应用程序,不过您也可以在作为网络服务器的那一计算机上使用应用程序。

客户 / 服务器应用程序和多用户数据库

您可以使用 Adaptive Server Anywhere 进行这样一种安装,它具有在不同计算机上运行的多个应用程序,并且通过网络连接到在单独计算机上运行的单个数据库服务器。这就是**客户/服务器**环境,它具有以下体系结构。接口库位于每一客户机上。



在此情况下,数据库服务器是 Adaptive Server Anywhere 网络数据库服务器,支持网络通信。该数据库也称作**多用户数据库**。

要使客户端应用程序在客户 / 服务器环境中运行,不需要对客户端应用 程序进行任何更改,只是要标识它应连接到的服务器。
^{第3章} Adaptive Server Anywhere 简介

关于本章本章介绍 Adaptive Server Anywhere 关系数据库系统。本章还描述了
Adaptive Server Anywhere 的用途和功能。

Adaptive Server Anywhere 简介

前一章 "数据库和应用程序"第9页简要介绍了数据库的一些基本特性。 本章描述 Adaptive Server Anywhere 的一些具体特点并描述其用途。

Adaptive Server Anywhere 提供一系列用来存储和管理数据的工具。您可 以使用这些工具将数据输入数据库中,更改数据库结构,以及查看或变 更数据。

Adaptive Server Anywhere 关系数据库管理系统是 SQL Anywhere Studio 的核心。 Adaptive Server Anywhere 是为要求功能完备的 SQL 数据库的 任务而专门设计的。根据设计,它可以在不同的环境下工作。它充分利用可用内存和 CPU 资源,在具有充足资源的环境中提供极佳的性能。 它还可以在具有有限物理和数据库管理资源的环境下很好地工作,包括移动计算环境、嵌入式数据库使用和工作组服务器。

Adaptive Server Anywhere 用途

Adaptive Server Anywhere 在理想情况下适合于以下角色:

工作组数据库服务器 规模从几个人到几百人的工作组都可以使用 Adaptive Server Anywhere 作为多用户数据库服务器。它提供高性能的工作组数据库,十分适合于(但并不局限于)管理和硬件资源有限的环境。

Adaptive Server Anywhere 可以采用多个 CPU 并使用高达 64 Gb 的 内存。我们的客户在生产中使用具有数十亿字节的 Adaptive Server Anywhere 数据库。

• **嵌入式数据库** 许多应用程序要求数据库 " 后台执行 "。个人信息 管理器, 文档管理系统 — 只与存储信息的应用程序有关。 Adaptive Server Anywhere 可用作这些应用程序的数据库。 UltraLite 部署选项用于具有非常有限资源的嵌入式环境。

嵌入式数据库的一个主要特点就是无需管理即可完全运行。 Adaptive Server Anywhere 在许多要求很高的商业应用程序中都证 实具备这一功能。

 移动计算 膝上型电脑和笔记本电脑现在在许多场所经常使用。
 Adaptive Server Anywhere 可用作这些计算机的 SQL 数据库。使用 MobiLink 同步技术和 SQL Remote 复制, Adaptive Server Anywhere 拓展了在整个企业范围基于事务的计算能力。UltraLite 部署选项和 MobiLink 同步技术为具有有限资源的设备(例如,手 持式计算机)提供完备的数据库功能。

Adaptive Server Anywhere 特点

Adaptive Server Anywhere 是在以下技术特点的基础上建立的:

- 完全 SQL 关系数据库管理系统 Adaptive Server Anywhere 是事务 处理关系数据库管理系统 (Relational Database-Management System, RDBMS),它具备多种功能,包括:完全恢复功能、联机备份、 参照完整性动作、存储过程、触发器、行级并发控制、时间表和 事件、丰富的 SQL 语言以及在完整 SQL RDBMS 中应提供的所有 功能。
- 经济合算的硬件要求 与其它数据库管理系统相比, Adaptive Server Anywhere 要求的内存和磁盘资源更少。
- **易于使用** Adaptive Server Anywhere 可以自我调节并易于管理。 您可以方便地使用 Adaptive Server Anywhere,而无需像其它关系 数据库管理系统那样进行大量的数据库管理工作。
- 独立和网络使用 Adaptive Server Anywhere 可以以独立的方式使用,例如在以数据为中心的应用程序中作为嵌入式数据库,或者在多用户客户/服务器或三层环境中作为网络服务器。作为嵌入式数据库系统,它可以根据需要由应用程序自动启动。
- 高性能 尽管 Adaptive Server Anywhere 在设计上强调简单的管理 和适度的资源要求,但它依然是可伸缩的高性能 DBMS。
 Adaptive Server Anywhere 可在多个 CPU 上运行,具有高级查询优 化程序,并且提供性能监控和调节工具。
- 行业标准接口 Adaptive Server Anywhere 提供本机 ODBC 3.5 驱动 程序,以使 ODBC 应用程序可以保持高性能;它还提供 OLE DB 驱动程序,以用于 ActiveX 数据对象 (ActiveX Data Object, ADO) 编程环境。它与 Sybase jConnect for JDBC 以及 JDBC-ODBC Bridge 一起提供,并且支持嵌入式 SQL 和 Sybase Open Client 接 口。
- **跨平台解决方案** Adaptive Server Anywhere 可以在许多操作系统 上运行,包括 Windows、Novell NetWare、 Sun Solaris 和 Linux。

平台不同,在每一平台上提供的组件也将有所不同。有关信息, 请参阅 " 组件的可用性 " 第 32 页。

Adaptive Server Anywhere 系统要求

Adaptive Server Anywhere 要求以下配置:

- 操作系统和 CPU 您必须具有以下系统之一,才可以运行 Adaptive Server Anywhere 数据库服务器:
 - Windows Windows 95、Windows 98、Windows Me、 Windows NT、Windows 2000 或 Windows XP。对于 Windows NT, 您需要 4.0 版或更高版本。本文档描述如何使 用 Windows NT。在 Windows NT 上运行的 SQL Anywhere Studio 组件也可以运行在其它 Windows 平台上。
 - Windows CE 要获得受支持的芯片和操作系统的列表,请
 参阅 "SQL Anywhere Studio 介绍 > 教程:使用 InfoMaker 创
 建报告 > 操作系统版本 "。
 - Novell NetWare 版本 4.11、 4.2 或 5.x。
 - UNIX,包括 Linux 对于支持的 UNIX 操作系统的详细列 表,请参阅 SQL Anywhere Studio Read Me First for UNIX。
- **PC 硬件** 对于 PC 操作系统,要求 Intel 486 或更高主频的 CPU 或 兼容的 CPU。
- 内存 最少只需 4 Mb 的内存就可以运行 Adaptive Server Anywhere。如果您在数据库中使用 Java,则 Adaptive Server Anywhere 要求 8 Mb 的内存。如果您使用管理工具, Adaptive Server Anywhere 要求至少 32 Mb 的 RAM。除了操作系统所要求 的内存外,您的计算机还必须具有上述充足的内存。

有关管理工具的内存要求的更多信息,请参阅 "SQL Anywhere Studio 介 绍 > 教程:使用 InfoMaker 创建报告 > 管理工具系统要求 "。

有关对于 SQL Anywhere Studio 中各组件的支持的操作系统的更多信息,请参阅 "SQL Anywhere Studio 介绍 > 教程:使用 InfoMaker 创建报告 > 支持的平台和系统要求 "。

网络软件要求

如果您正运行 Adaptive Server Anywhere 网络服务器,则必须安装并运行适当的网络软件。

Adaptive Server Anywhere 网络服务器可用于 Windows、 Novell NetWare、 Linux 和 UNIX 操作系统。

Adaptive Server Anywhere 支持 TCP/IP 网络协议和用于 Novell NetWare 的 SPX 协议。

Adaptive Server Anywhere 数据库服务器

在产品中包括两个版本的 Adaptive Server Anywhere 数据库服务器:

 个人数据库服务器 此类服务器是为单个用户在同一计算机上使 用提供的;例如,作为嵌入式数据库服务器。它还适用于开发工 作。

个人服务器可执行文件的名称如下:

- 在 UNIX 操作系统上, 文件名称是 dbeng8。
- 在 Windows (Windows CE 除外)上,文件名称是 dbeng8.exe。
- 网络数据库服务器 除了具有个人服务器的各项功能外,网络服务器还支持网络上的客户/服务器通信。它是为多用户使用提供的。

网络服务器可执行文件的名称如下:

- 在 UNIX 操作系统上, 文件名称是 dbsrv8。
- 在 Windows (包括 Windows CE)上,文件名称是 dbsrv8.exe。
- 在 Novell NetWare 上,服务器是称作 dbsrv8.nlm 的 NetWare 可装载模块 (NetWare Loadable Module, NLM)。

每一版本中的相同
 请求处理引擎在这两个版本的服务器中是相同的。它们都支持完全相同
 的 SQL 语言,并且都支持完全相同的数据库功能。个人服务器不支持
 网络上的通信、超过 10 个并行连接或使用多于两个 CPU。针对个人服
 务器开发的应用程序在使用上与针对网络服务器开发的应用程序没有什么不同。

Adaptive Server Anywhere 应用程序

本节介绍在 SQL Anywhere Studio 产品中与 Adaptive Server Anywhere 一 起提供的一些数据库应用程序。这些应用程序可帮助您设计、建立和管 理数据库。

Sybase Central

Sybase Central 提供图形用户界面,以用于创建和修改数据库和数据库 对象、检查数据库的结构和管理数据库服务器。

Sybase Central 使您可以执行各种任务,例如: 创建新的数据库、添加 表或向表添加列。

有关使用 Sybase Central 的信息, 请参阅 " 第 2 课: 连接到数据库 " 第 54 页。

Sybase Central	
文件(F) 编辑(E) 视图∨) 工具(T)	帮助(H)
📡 Sybase Central 🔽 🔶 🔿	
Note: Sybase Central	详细信息
🖻 🦻 Adaptive Server Anywhere 8	文件夹
	🖸 Adaptive Server Anywhere 8
□ ◇/////// ⊡ · 【 MobiLink 同步	🥰 MobiLink 同步
_	
	· · · · · · · · · · · · · · · · · · ·

Sybase Central 窗口与 Windows"资源管理器"类似。主窗口被拆分成两个垂直对齐的窗格。左窗格以树状结构显示数据库对象或容器的层次结构图。容器是数据库对象,它可以保留其它数据库对象,包括其它容器。

右窗格显示已在左窗格中选择的容器的内容。例如,若要创建数据库,

在左窗格中单击实用程序文件夹,然后在右窗格中双击创建数据库。

有关 Sybase Central 的说明,请参阅 "SQL Anywhere Studio 介绍 > 教程: 使用 Sybase Central 管理数据库 "。

Interactive SQL

Interactive SQL 是用于键入 SQL 语句并将 SQL 语句发送到数据库的应用程序。通过 Interactive SQL,您可以查询和更改数据库中的数据,还可以修改数据库的结构。

在 Sybase Central 中可以执行的任何任务都可以在 Interactive SQL 中执行,但在 Sybase Central 中执行管理任务会更容易一些。因此,本手册将尽可能使用 Sybase Central。但是,在一个任务十分重要,或者在 Interactive SQL 中实施较为简单时,也包括 Interactive SQL 说明。

有关 Interactive SQL 的说明,请参阅 "使用 Interactive SQL" 第 71 页。

- Interactive SQL 窗
 •
 "SQL 语句"窗格
 这是您键入 SQL 语句以访问和修改数据的窗格。"SQL 语句"窗格上方的标题栏显示当前连接信息。
 - "结果"窗格 这是选项卡式窗格,用来显示查询结果集、来自数据库服务器的消息和与查询执行有关的信息。例如,如果您输入一个查询,询问有多少客户订购了5种或5种以上不同类型的产品,查询出的数目显示在"结果"窗格的"结果"选项卡上,并且在"结果"窗格的"计划"选项卡上显示用于执行语句的查询优化程序的计划。您可以在"结果"选项卡上编辑结果集。

有关编辑结果集的更多信息,请参阅 "在 Interactive SQL 中编辑表值 "第 81 页。

数据库调试程序

除了存储数据外,数据库还可以存储在 SQL 存储过程、触发器和事件 中写入的逻辑。它们还可以包含 Java 类。 为了帮助您在数据库中开发和测试逻辑, Adaptive Server Anywhere 包括了一个调试程序,您可以通过它逐步跟踪数据库中的 SQL 和 Java 逻辑。

有关更多信息,请参阅 "ASA SQL 用户指南 > 调试数据库中的逻辑 "。

实用程序

提供了一组实用程序,可用于执行诸如备份数据库之类的管理任务。实 用程序适合于包括在批处理文件中,以便重复使用。

有关管理实用程序的更多信息,请参阅 "ASA 数据库管理指南 > 数据库 管理实用程序 "。

组件的可用性

您安装哪些组件取决于以下几个方面:正使用的操作系统,在安装软件时所做的选择,以及是接收了 Adaptive Server Anywhere 产品还是将 Adaptive Server Anywhere 作为其它产品的一部分安装。

例如,如果您收到的 Adaptive Server Anywhere 是其它产品的一部分,则不可以同时具有该数据库服务器的两个版本。

不是所有组件都在所有操作系统上提供。例如,对于 NetWare 则没有个 人服务器,只有网络服务器。

SQL Anywhere 程序组

对于 Windows 操作系统,在安装软件后,您将具有 SQL Anywhere 程序 组。通过单击 "开始 "按钮并选择 "程序 " → "Sybase SQL Anywhere 8",您可以访问该程序组。

在 UNIX 系统下安装 SQL Anywhere Studio 则不会提供程序组。

程序组项 程序组包含以下某些或全部项。您所能看到的项取决于安装软件时所做的选择。

• Adaptive Server Anywhere 包含以下项:

- **数据库对象调试程序** 启动 Adaptive Server Anywhere 过程调 试程序。
- **Interactive SQL** 启动 Interactive SQL 实用程序以将 SQL 语 句发送到数据库。
- **网络服务器示例** 启动网络数据库服务器并装入示例数据 库。
- **ODBC 管理器** 启动 **ODBC** 管理器以设置和编辑 **ODBC** 数 据源。
- **个人服务器示例** 启动个人服务器并装入示例数据库。
- MobiLink 程序组 访问 MobiLink 同步程序和示例。
- UltraLite 程序组 访问 UltraLite 程序和示例。
- **iAnywhere 在线资源** 打开有关 iAnywhere Solutions 信息的 Web 页。
- **联机手册** 打开用于 Adaptive Server Anywhere 的联机文档。
- Sybase Central 启动数据库管理工具 Sybase Central。

此外,您可以具有用于 InfoMaker 和 PowerDesigner 的项。

^{第4章} 数据库应用程序的体系结构

关于本章 本章描述数据库应用程序如何与 Adaptive Server Anywhere 数据库服务 器进行通信,并且概要介绍 Adaptive Server Anywhere 的体系结构。

应用程序编程接口

本节将更为详细地介绍应用程序体系结构。在"各组件是如何协同工作的"第 20页中概要介绍了数据库应用程序的体系结构。

Adaptive Server Anywhere 支持多种编程接口, 使您在使用各种类型的应用程序和应用程序开发环境时更具灵活性。

支持的接口和协议

以下图示显示支持的接口以及所使用的接口库。在大多数情况下,接口 库与接口本身具有相同的名称。





与 SQL Anywhere Studio 一起提供的应用程序使用以下若干接口:

通信协议

每一接口库都使用**通信协议**与数据库服务器通信。Adaptive Server Anywhere 支持两种通信协议:**表格式数据流**(Tabular Data Stream, TDS) 和**命令序列**。这两种协议是固有协议,对于大多数用途而言,您使用哪 一协议并无多大关系。您所选择的开发环境将由可用工具控制。

您将看到的主要区别是与数据库的连接。命令序列应用程序和 TDS 应 用程序使用不同的方法来标识数据库和数据库服务器,因此连接对话框 是不同的。

TDS 此协议由 Sybase Adaptive Server Enterprise、Open Client 应用程 序以及 Java 应用程序(这些应用程序通过 TDS 使用 jConnect JDBC 驱 动程序连接)使用。

命令序列 此协议由 Adaptive Server Anywhere、 Adaptive Server IQ 使用,并且由嵌入式 SQL、 ODBC 和 OLE DB 应用程序使用。

ODBC 应用程序



您可以使用多种开发工具和编程语言开发 ODBC 应用程序,如下图中 所示。

例如,在与 SQL Anywhere Studio 一起提供的应用程序中, InfoMaker 和 SQL Modeler 使用 ODBC 连接到数据库。

嵌入式 SQL 应用程序

您可以使用嵌入式 SQL 编程接口开发 C 或 C++ 应用程序。例如,命令 行数据库工具就是以此方式开发的应用程序。

嵌入式 SQL 也是 UltraLite 应用程序的编程接口。



Open Client 应用程序

Open Client 也是 Sybase Adaptive Server Enterprise 支持的接口。您可以 用 C 或 C++ 语言开发 Open Client 应用程序。其它 Sybase 应用程序 (例如 OmniConnect 或 Replication Server)使用 Open Client。



JDBC 应用程序

您可以使用 Java 开发使用 JDBC 的应用程序,以连接到 Adaptive Server Anywhere。与 SQL Anywhere Studio 一起提供的某些应用程序使用 JDBC,这些应用程序包括:存储过程调试程序、Sybase Central 和 Interactive SQL。

Java 和 JDBC 也是用于开发 UltraLite 应用程序的重要编程语言。



OLE DB 应用程序

Adaptive Server Anywhere 包括 OLE DB 驱动程序。您可以直接使用 Microsoft 的 OLE DB 接口开发应用程序,也可以使用 ActiveX 数据对象 (ActiveX Data Objects, ADO) 接口开发应用程序。 ADO 接口包括在 Visual Basic 和其它 Microsoft 编程工具中。

Adaptive Server Anywhere 内在特性

尽管您永远不需要处理数据库服务器的内部结构,但如果对其内部结构 有一定的了解,可以帮助您更好地理解相关进程。

数据库服务器内在特性

Adaptive Server Anywhere 数据库服务器的内部结构允许您高效地处理多个请求。

- 通信层可以处理与客户端应用程序的实际数据交换。此通信层接收来自客户端应用程序的请求,并返回结果。执行这些操作的时机是由客户端和服务器之间的协商控制的,从而确保网络通信量保持最低,而数据在客户端会尽快处于可用状态。
- 分析程序检查发送给数据库服务器的每一 SQL 语句,并将它转换 为可供处理的内部格式。
- 如果请求是查询、更新或删除语句,则可能有多种不同的访问数据的方法,在所用时间上也可能有很大差异。优化程序的工作就是从所有这些可能的方法中进行选择,以找到迅速获取所需数据的最佳途经。
- Java 虚拟机被内置于数据库服务器中,并且 Java VM 处理客户端 应用程序发送的所有 Java 操作或数据库服务器内部调用的所有 Java 操作。
- 数据库服务器的最低级别涉及:读取来自磁盘的数据和将数据写 入磁盘,将数据存入内存中以避免不必要的磁盘访问,以及平衡 不同用户的要求。



数据库内部结构

Adaptive Server Anywhere 数据库中的所有信息通常都存储在单个数据库 文件中,可以从一台计算机复制到另一台计算机。还可以为若干文件生 成数据库,但通常只有非常大的数据库才需要这样做。

除了数据库文件外, Adaptive Server Anywhere 在运行数据库时还使用 另外两个文件。这两个文件是事务日志和临时文件。

- 数据库文件 在内部,数据库文件是由页 (磁盘的固定大小区域)组成的。数据访问层一次读取和写入一页上的数据。许多页保存数据库表中的数据,而其它一些页则保存索引信息、与数据库内数据的分发有关的信息等。
- **事务日志** 事务日志是一个单独的文件,记录在数据库上执行的 所有操作。通常,它具有与数据库文件相同的名称,只是它以后 缀.log 结尾,而不是以.db 结尾。事务日志具有三个重要功能。
 - 记录对数据执行的操作以支持恢复 如果数据库文件损坏, 您可以根据备份以及事务日志重新创建数据库。
 - **提高性能** 通过将信息写入事务日志,数据库服务器可以安 全地处理您的语句,而不必频繁写入数据库文件。

- **支持数据库复制** SQL Remote 和 MobiLink 客户端实用程序 使用此文件复制对有时候连接到网络的便携式计算机上的数 据库的更改。
- 临时文件 在数据库服务器启动时该临时文件打开,在数据库服务器结束时该临时文件关闭。顾名思义,临时文件是在服务器运行时用来保存临时信息的。临时文件不保存需要在两个会话之间保留的信息。

该临时文件存储在您的临时目录中。此目录的位置通常由 TEMP 环境变量标识。

^{第5章} 设计和建立数据库

关于本章本章介绍数据库设计的一些基本原理,并说明如何使用 Sybase Central 创建数据库。这里将使用 Adaptive Server Anywhere 示例数据库来阐述 所涉及的基本原理。

<u>简介</u>

简介

即使是最简单的数据库,也值得花些时间进行设计,例如包含什么样的表、使用哪些键使这些表相关,等等。

就中小型数据库而言,数据库设计虽不是困难任务,但却是非常重要的 一项任务。不正确的数据库设计会导致效率低下,并且可能造成数据库 系统不可靠。数据库应用程序是针对应用数据库的特定部分来建立的, 并依赖于数据库设计,因此不正确的设计可能很难在以后做出修订。

设计大型数据库是非常复杂的任务。有一些正规的方法(如概念数据 建模)将帮助您创建高效的设计。利用 Sybase PowerDesigner 和 DataArchitect 等强大的工具,您就可以应用这些技术来设计和维护大型 数据库设计。

本章并不是要解决大型数据库的设计问题,而是要帮助您确定组合在单个表中的信息的种类,以及分析表间关系并将其分类的方式。

有关数据库设计原则的初步介绍,请参阅 "ASA SQL 用户指南 > 设计数 据库 "。有关更高级的处理,请参阅 Sybase PowerDesigner 文档或专门 讨论数据库设计的手册。

本章假定您已经熟悉数据库表、主键和外键等概念。

有关主键和外键的信息,请参阅 "表间的关系 " 第 11 页。

本章内容

如果想知道	请参阅
在何处查找有关示例数据库的信息	"示例数据库"第48页
如何连接到示例数据库	"连接到示例数据库"第49页
如何查看表	"查看示例数据库中表的列表 " 第 50页
如何查看列	" 查看表列 " 第 51 页

如果想知道	请参阅
如何创建数据库文件	"第1课: 创建数据库文件"第 53页
如何连接到数据库	"第2课:连接到数据库"第54 页
如何创建表	"第3课:设计和创建表"第56 页
如何创建主键	"第4课:标识和创建主键"第 57页
如何设置列属性	"第5课:设计列属性"第59页
如何创建表间关系	"第6课:设计和创建表间关系 "第61页

示例数据库

Adaptive Server Anywhere 附带了一个示例数据库。文档中的许多示例都 使用该示例数据库。

该示例数据库表示一家小型公司,它生产相当有限的几种运动服装。该 数据库包含公司的内部信息(职员、部门和财务数据)以及产品信息 (产品)和销售信息(销售订单、客户和联系信息)。该示例数据库中 的所有信息都是虚构的。

下图显示该示例数据库中的表以及这些表的相互关系。矩形框表示表, 箭头表示外键关系。主键列带有下划线。

我们将使用该数据库来阐述本章所述的概念。



asademo.db

使用 Sybase Central 查看示例数据库的结构

本节介绍如何使用 Sybase Central 查看示例数据库的内容。

连接到示例数据库

为了查看数据库中的表和其它对象,您需要通过 Sybase Central 连接到示例数据库。

✤ 通过 Sybase Central 连接到示例数据库:

- 1 启动 Sybase Central。
 - 从"开始"菜单中,选择"程序"→"Sybase SQL Anywhere 8"
 → "Sybase Central"。

或

• 在命令提示符处输入以下命令:

scjview

- 2 打开"连接"对话框。
 - 单击"工具"→"连接"。
 - 如果系统向您显示插件列表,请从该列表中选择"Adaptive Server Anywhere"。

随即出现 " 连接 " 对话框:

- 3 选择 "ODBC 数据源名称 " 并单击 " 浏览 "。在随即显示的列表 中选择 "ASA 8.0 Sample"。
- 4 单击 " 确定 " 以建立连接。

连接	×
标识 数据库 高级	
• • •	
以下值用于向数据库标识您自己	
用户(U):	
口令(P):	
您可以使用配置文件中存储的缺省连接值。	
● 无◎	
○ ODBC 数据源名(N)	
ASA 8.0 Sample 刘览(回)	
● ODBC 数据源文件(F)	
▼ 浏览(1)	
确定 取消 帮用	љ(H)

查看示例数据库中表的列表

当您连接到示例数据库后,就可以打开左窗格中的文件夹以查看组成数 据库的表和其它对象。

有关通过 Sybase Central 连接到数据库的信息, 请参阅 " 连接到示例数据 库 " 第 49 页。

◆ 查看数据库中表的列表:

1 在 Sybase Central 中打开 Adaptive Server Anywhere。

左窗格上的每个顶级文件夹都是一个**插件**。每个软件组件由一个单独的插件来进行管理。Adaptive Server Anywhere 插件用于 管理 Adaptive Server Anywhere 的各个方面。

2 打开示例数据库。

在 Adaptive Server Anywhere 插件下,将列出您已连接到的数据 库服务器。当示例数据库启动时,该数据库服务器名为 *asademo*。

每个数据库服务器可能会运行一个或多个数据库。在本例中, 示例数据库也名为 asademo。该数据库的旁边还会显示当前连接 的用户名。在本例中,您将作为名为 DBA 的用户建立连接。

3 打开"表"文件夹。

每个数据库都包含一组文件夹,这些文件夹描述不同类型的数 据库对象。在这里,我们只关心数据库中的表。打开 "表 "文 件夹即可列出可用的表。



说明

每个表都有一个**所有者**,所有者在左窗格中的表名旁列出而且在两侧加 有括号。如果您看到了上图中未显示的表,请右击 asademo 数据库,然 后从弹出式菜单中选择 "按所有者过滤对象 "。选择 "DBA",然后清除 列表中的其它用户 ID。单击 "确定 ",将所显示的对象限定为 DBA 所 拥有的对象。

查看表列

Sybase Central 将表显示为容器。每个表都包含列、进表外键和出表外键、索引以及触发器的文件夹。在这里,我们仅考虑示例数据库中 product 表的列。

有关查看表的信息,请参阅 " 查看示例数据库中表的列表 " 第 50 页。

✤ 查看有关 product 表列的信息:

1

在 Sybase Central 中打开 product 表。



2 打开"列"文件夹。

各个列的名称、数据类型以及其它信息出现在右窗格中。

3 查看 color 列的属性。

右击 color 列,然后从弹出式菜单中选择 "属性 "。随即出现 color 属性表。此属性表包含对该列的详细描述。有些属性仅对 于高级用户才有用,本书不介绍这些属性。

您可以使用 Sybase Central 来浏览示例数据库的其它方面,包括每个表中的外键、主键和数据。我们建议您这样做,以便熟悉示例数据库,从 而更好地理解文档其余部分的示例。

教程:设计和建立简单数据库

在设计数据库时,将计划需要存储有关哪些项的信息以及将保存有关每 个项的哪些信息。您还将确定这些项的相互关系。数据库中的项归类为 实体;这些实体之间的链接称作**关系**。

在本教程中,您将设计并建立一个非常简单的数据库,它基于示例数据 库的 product、sales_order_items、sales_order 和 customer 表,但经过了简 化。您所创建的数据库不会在文档的其它部分中使用,但我们强烈建议 您学完本教程,以便熟悉该软件和相应的概念。

第1课: 创建数据库文件

在这一课,您将创建一个用于存放数据库的数据库文件。

有关更多信息,请参阅 "数据库系统的组成部分 "第18页。

概念 数据库文件是用于包含数据库的容器。数据库文件包含所有数据库通用的系统表和其它系统对象,但您必须添加表以及表所包含的数据。

数据库中的表、索引等对象以及它们之间关系的集合组成了数据库**模** 式。模式是不含数据的数据库。本教程介绍如何设计和创建非常简单的 数据库模式。

有关其中一些组件的概念性简介,请参阅 "关系数据库概念 " 第 10 页。

数据库中每个对象 (包括表、列和索引)的名称都是**标识符**。对于哪 些字符可以用作标识符,存在特定的规则。您可以使用任意字母、数字 或符号的组合。但是,如果列名包含字母、数字或下划线之外的字符, 不以字母开头,或者与关键字相同,则必须将列名用双引号引起来。

如果 QUOTED_IDENTIFIER 数据库选项设置为 OFF,则双引号可用于 分隔 SQL 字符串,而不能用于标识符。但不管 QUOTED_IDENTIFIER 如何设置,您都可以用方括号来界定标识符。

有关标识符的更多信息,请参阅 "ASA SQL 参考手册 > SQL 语言元素 > 标识符 "。

1 启动 Sybase Central。

- 2 在左窗格中打开 Adaptive Server Anywhere 容器,然后打开 "实用程序"文件夹。在右窗格中双击"创建数据库"。"创建数据库"向导随即打开。
- 3 阅读引导页上的信息, 然后单击 "下一步 "。
- 4 选择 " 在这台计算机上创建数据库 ", 然后单击 " 下一步 "。
- 5 选择数据库文件的位置和名称:
 - 输入文件名 c:\temp\mysample。如果临时目录不是
 c:\temp,请指定您自己选择的路径。
- 6 单击"完成"以创建数据库。

当创建数据库时,其它选项将变为可用,您可以通过单击 "下 一步 "(而不是 "完成 ")来查看这些选项,但缺省的选项适用 于多数的情况。

"创建数据库 "窗口显示任务的进度。当文件创建完毕时,单击 "确定 "关闭窗口。

第2课: 连接到数据库

在这一课,您将连接到已创建的数据库文件。

有关更多信息,请参阅 "各组件是如何协同工作的 "第 20 页。

练习当创建数据库后,您就可以连接到该数据库以创建表和其它数据库对象。

◆ 连接到数据库:

- 1 启动 Sybase Central。
- 2 选择 " 工具 " → " 连接 ", 打开 " 连接 " 对话框。如果已经装载 多个插件,则可能需要在 " 连接 " 对话框打开之前选择 Adaptive Server Anywhere 插件。
- 3 指定用户 ID 和口令。

在"连接"对话框的第一个选项卡("标识"选项卡)上,输入用户 ID DBA 和口令 SQL。它们是为新数据库创建的值,将 授予您对新数据库的访问权。

在该选项卡底部的配置选项中选择 "无"。

4 指定数据库文件。

单击 "数据库 "选项卡。在 "数据库文件 "字段中输入数据库 文件的完整路径。例如,如果您按照上一课中的建议执行操 作,则应输入以下路径:

c:\temp\mysample.db

5 连接到该数据库。

单击 " 确定 "。 Sybase Central 即会连接到该数据库。

打开左窗格中的数据库服务器容器以查看 mysample 数据库。



第3课:设计和创建表

在这一课,您将在数据库中添加一个表。

概念 数据库中的每个表都应 来说,就是您在标识**实**

数据库中的每个表都应包含有关一个主题的信息。用数据库设计的语言 来说,就是您在标识**实体**。例如,示例数据库在一个表中包含有关职员 的信息,在另一个表中包含有关产品的信息:职员和产品是数据库设计 中的实体。

有关表的简介,请参阅 "数据库表 "第 10 页。

表中的每个列描述您要存储其相关信息的对象的特定特征。例如在示例 数据库中,职员表具有包含以下信息的列:职员 ID 号、名字和姓氏、 地址以及其它与特定职员相关的特定信息。

在数据库图示中,每个表都用矩形框来表示。表的名称位于矩形框的顶部,而列名则在矩形框中列出。

product		
<u>id</u>	<u>integer</u>	
name	char(15)	
description	ch <i>ar</i> (30)	
size	ch <i>ar</i> (18)	
color	char(6)	
quantity	integer	
unit_price	numeria(15,2)	

在示例数据库的 product 表(如上所示)中,每种产品都是一种运动服装。

练习

创建 product 表的简化版本,它仅包含标识符 (id)和 name 列。

♦ 创建 product 表:

- 1 在 Sybase Central 中连接到数据库 (如果您尚未建立连接)。
- 2 打开数据库中的 "表 "文件夹。

首先打开服务器和数据库容器,然后打开 "表 "文件夹。

- 3 双击右窗格中的 "添加表 "。随即出现 "新建表 "对话框。
- 4 为您的表命名。

在该对话框左上角的 "名称 "框中键入 product。

5 创建列。

一个空白列空间出现在表格中。您可以在此空间中创建一个 列。添加一个具有以下属性的列:

- 列名 为该列指定名称 ID。
- **主键** 单击"键"标题,直到出现标有钥匙和字母 P 的图标,指示该列为**主键**。
- 数据类型 为该列指定 integer 数据类型。

您可以忽略其它属性。再添加一个具有以下属性的列:

- **列名** 为该列指定名称 name。该列包含产品名称。
- **数据类型** 为该列指定 char 数据类型以包含字符串, 然后 在大小列中输入最大长度 15。
- 6 若要结束操作,请单击工具栏最右侧的"保存并关闭"图标。

您现在已经创建了数据库中的一个表。该表的数据将包含在数据库文件 中。该表目前是空的。

下面两课将详细讨论列和数据类型。

第4课:标识和创建主键

在这一课,您将进一步学习如何定义表的主键。本课没有相关的练习。 有关更多信息,请参阅"具有主键的表"第12页。 概念

主键是特殊的列,它(或它们)用于唯一地标识表中的行。在 product 表中, id 列唯一地标识各种产品。

product		
<u>id</u>	<u>integer</u>	
name	char(15)	
description	ch <i>ar</i> (30)	
size	ch <i>ar</i> (18)	
color	char(6)	
quantity	integer	
unit_price	numeria(15,2)	

对于 *id* 列,每一行都有一个唯一值,而且每一行中的值仅通过它的 *id* 值与单个产品标识符相关。两种产品可能具有相同的名称或相同的尺 寸,但不可能具有相同的 *id* 号。在该图示中,*id* 列带有下划线,这表示它是主键。

在数据库设计中,专门创建一个列来包含没有其它任何含义的标识符是 一种常见的做法。例如在您的银行、水电费或信用卡贷借对照表上,每 个帐户都有一个唯一的标识符。

使用自动增量主键 通过给主键列赋以缺省值 AUTOINCREMENT,您就可以使主键的输入 变得简单。每次添加一个新行时,该列的值就会自动输入,而且该值比 上次添加的行在该字段中的值大一。

✤ 创建 AUTOINCREMENT 主键:

- 1 右击 product 表, 然后从弹出式菜单中选择 "编辑 "。
- 选择主键列。单击工具栏上的"列属性"图标,打开该列的属 性表。
- 3 选择"值"选项卡。
- 4 选择"缺省值"选项。
- 5 单击 "系统定义的",然后从下拉列表中选择"自动增量"。
- 6 单击 "确定 "以关闭列属性表。
- 7 若要结束操作,请单击"保存并关闭"图标。
第5课:设计列属性

在这一课,您将进一步学习如何选择表列的数据类型以及其它特性。

概念 每个列都有一种与之关联的数据类型。数据类型定义列所包含的信息的 类型。为列选择一种适合该列中数据的数据类型。例如,标识符列通常 具有整数数据类型,而包含名称或地址的列则必须具有字符数据类型。

数据类型分为以下类别:

• Numeric 数据类型 Numeric 数据类型有若干种。有些是确切值 (不受运算中舍入错误的影响),而有些则是近似值。

列的数据类型会影响列的最大大小。例如,如果您指定 SMALLINT,则列可以包含的最大值为 32,767。如果您指定 INTEGER,则最大值为 2,147,483,647。

有关完整列表,请参阅 "ASA SQL 参考手册 > SQL 数据类型 > 数 值数据类型 "。

Character 数据类型 它们用于包含文本字符串,例如名称、地址等。这些数据类型具有一个长度,它指示可以在列中存储的最大字符串长度。

有关列表,请参阅 "ASA SQL 参考手册 > SQL 数据类型 > 字符数 据类型 "。

Binary 数据类型 它们可用于包含可能对应用程序有意义的信息,但以二进制格式编码。

有关列表,请参阅 "ASA SQL 参考手册 > SQL 数据类型 > 二进制数据类型 "。

• Date/time 数据类型 它们包含一天的时间以及日期。

有关列表,请参阅 "ASA SQL 参考手册 > SQL 数据类型 > 日期和时间数据类型 "。

 Long 数据类型 它们有时称作 Blob (二进制大对象)。它们可用 于包含长文本字符串 (在有些数据库中称作备忘录字段)、图像 或其它二进制信息。 有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 数据类型 > LONG BINARY 数据类型 [BINARY]" 和 "ASA SQL 参考手册 > SQL 数据类型 > LONG VARCHAR 数据类型 [字符]"。

另外, Adaptive Server Anywhere 还支持用户定义的数据类型和特殊的 Java 数据类型。本入门手册中暂不讨论这些数据类型。

NULL 和 如果对于该列,每一行都必须包含一个值,则应将该列定义为 NOT NULL。否则,将允许该列包含 NULL (表示缺少值)。缺省设置 是允许 NULL,但如果没有充分的理由来允许 NULL,则应将列显式声 明为 NOT NULL。

有关 NULL 的完整说明,请参阅 "ASA SQL 参考手册 > SQL 语言元素 > NULL 值 "。有关在比较中使用 NULL 的信息,请参阅 "ASA SQL 参考手册 > SQL 语言元素 > 搜索条件 "。

✤ 指定列的数据类型:

- 1 在"表"文件夹中打开该列所在的表。
- 2 打开"列"文件夹。
- 3 双击该列。该列的属性表即会打开。
- 4 单击 "数据类型 "选项卡。
- 5 从第一个下拉框中选择一种基本数据类型。

(练习本课程和上一课介绍了在创建数据库表时需要了解的基本概念。您可以通过在数据库中添加其它一些表来应用这些概念。本章中后面的课程将使用这些表。

使用 Sybase Central 表编辑器在数据库中添加以下表:

- **customer** 添加一个名为 *customer* 的表, 它包含以下列:
 - id 每个客户的标识号。该列具有 integer 数据类型,它是主
 键。使该列成为自动增量键。
 - company_name 公司名称。该列具有 character 数据类型, 其最大长度为 35 个字符。

- sales_order 添加一个名为 sales_order 的表,它包含以下列:
 - id 每份销售订单的标识号。该列具有 integer 数据类型,它
 是主键。使该列成为自动增量键。
 - order_date 下订单的日期。该列具有 date 数据类型。
 - cust_id 下订单的客户的标识号。该列具有 integer 数据类型。
- sales_order_items 添加一个名为 sales_order_items 的表以包含行项 目信息。该表包含以下列:
 - id 行项目所属的销售订单的标识号。该列具有 integer 数据 类型,应该标识为主键列。
 - **line_id** 每个销售订单的标识号。该列具有 **integer** 数据类型,应该标识为**主键**列。
 - prod_id 所订购产品的标识号。该列具有 integer 数据类型。

您现在已经在数据库中创建了四个表。这些表尚未以任何方式相互关 联。在下一课,您将通过定义外键来使这些表相互关联。

第6课:设计和创建表间关系

在这一课,您将学习如何使用外键来设计和创建表间关系。

有关更多信息,请参阅 "表是由外键关联的 "第12页。

概念 虽然每个表都包含有关单个主题的信息,但两个或更多个表可能会包含 相关的信息。例如,职员是部门的成员,或者销售订单是一组产品的销 售订单。数据库中的关系可能表现为表间的外键关系,也可能自身就成 为独立的表。您将在本章中看到这两种情况的示例。

> 通过在数据库中创建关系,可以编写用于管理表中数据的规则或惯例的 代码。当关系置入数据库结构后,将不存在对例外情况的规定。

表间关系分为以下几个类别。

 一对一关系 一个实体中的每个项都对应于另一个实体中的零个 或一个实体。例如,在示例数据库中,一个 职员管理一个 部门。 没有地方可以输入另一个部门经理。重复该部门条目会涉及到重 复部门 ID,由于部门 ID 列是主键,所以这是不可能的。

通常,最好将处于一对一关系的项组合到一个表中。 department 表中有一个 manager 列,而不是有一个单独的表名为 manager。

有关应该单独存放项的情况,请参阅 "ASA SQL 用户指南 > 设计数据库 "。

 多对一关系 多对一关系成为表间的外键关系。在多对一关系中, 一个 实体中的主键成为多个 表中的新外键列。

例如,在您刚创建的数据库中,一个客户可以下多份订单,但 每份订单只能由一个客户发出。若要表示一对多关系, sales_order 表中需要一个**外键列** (cust_id),它映射到 customer 表中的主键列 (id)。通常,为这两个列指定相同的名称将会方便一些。

employ	ee		departmer	nt
emp_id	integer		dept id	integer
manager_id	integer		dept_name	char(40)
emp_fname	char(20)	dent id - dent id	dept_head_id	integer
emp_Iname	char(20)	dept_id = dept_id		
dept_id	integer			
street	char(40)			
city	char(20)			
state	char(4)			
zip_code	char(9)			
phone	char(10)			
status	char(1)			
ss_number	char(11)			
salary	numeric(20,3)			
start_date	date			
termination_date	date			
birth_date	date			
bene_health_ins	char(1)			
bene_life_ins	char(1)			
bene_day_care	char(1)			
sex	char(1)			

sales_order 表的 cust_id 列中的每一个条目都必须与 customer 表的 id 列中的一个条目相匹配。sales_order 表 (包含关系中的外键)称 作外表或引用表。customer 表 (包含被引用的主键)称作主表或 被引用表。

• 多对多关系 多对多关系由一个中间表来表示,而且从该中间表

到每个相关实体存在外键关系。

例如,在示例数据库中,产品和销售订单之间存在多对多关系。 一个销售订单可能包含多种产品,而一种产品可能会出现在多个 销售订单上。



在某些情况下,中间表 (sales_order_items) 包含附加的信息,如已订购产品的件数以及它们的发货日期。在本例中,中间表不包含任何附加信息。

练习 添加用于使数据库中的表相关联的外键。

添加以下外键:

- sales_order_items 中 id 列的外键,引用 sales_order 中的 id 列。该键 将销售订单和销售订单项之间的多对一关系置入数据库。
- sales_order_items 中的 prod_id 列的外键,它引用 product 中的 id 列。 该键将销售订单项和产品之间的多对一关系置入数据库。
- *sales_order* 中的 *cust_id* 列的外键,它引用 *customer* 中的 *id* 列。该键 将销售订单和客户之间的多对一关系置入数据库。

前两个外键一起将销售订单和产品之间的多对多关系置入数据库。

◊ 创建外键:

- 1 单击要为其创建外键的表。
- 2 打开"外键"文件夹。
- 3 双击 "添加外键"以打开"新建外键"向导。
- 4 遵循向导中的指导操作。

至此,您就学完了有关设计和建立关系数据库的入门章节。本书的其余 章节将介绍如何在数据库中添加数据和从数据库中检索数据。这些章节 将使用 Adaptive Server Anywhere 示例数据库,该数据库比您刚创建的 数据库大一些。

^{第6章} 将您的应用程序连接到其数据库

关于本章 本章向您说明如何建立从您的应用程序到其使用的数据库的连接。

连接简介

任何使用数据库的客户端应用程序都必须在可以执行任何工作前建立与 该数据库的连接。客户端应用程序的所有活动都必须借助该连接才可以 进行。例如,您的用户 ID 确定在数据库上要执行操作的权限,并且数 据库服务器具有您的用户 ID,因为它是对建立连接的请求的一部分。

许多客户端应用程序(包括应用程序开发系统)使用**开放式数据库连接**(**Open Database Connectivity, ODBC**)接口访问 Adaptive Server Anywhere。 ODBC 数据源是一组存储在注册表或文件中的连接参数。

您可以使用 ODBC 数据源从以下任何应用程序连接到 Adaptive Server Anywhere 数据库:

- Sybase Central 和 Interactive SQL。
- 所有 Adaptive Server Anywhere 实用程序。
- PowerDesigner 和 InfoMaker。
- 支持 ODBC 的任何应用程序开发环境,例如 Microsoft Visual Basic、Sybase PowerBuilder 和 Borland Delphi。

UNIX 上的 Adaptive Server Anywhere 客户端应用程序可以使用 ODBC 数据源。在 UNIX 上,数据源作为文件存储。

除了 ODBC 之外, Adaptive Server Anywhere 还支持若干编程接口。有 关更多信息,请参阅 "ASA 数据库管理指南 > 连接到数据库 > 连接简介 "。

创建 ODBC 数据源

ODBC 数据源是保存连接参数以供重复使用的简便方法。一旦具有数据 源后,您的连接字符串可以只指定要使用的数据源:

DSN=my data source

Sybase Central 和 Interactive SQL 中的 " 连接 " 对话框具有用于输入 ODBC 数据源名称或 ODBC 数据源文件的字段。

本节介绍如何在 Windows 中创建简单 ODBC 数据源。

有关详细信息,请参阅 "ASA 数据库管理指南 > 连接到数据库 > 使用 "ODBC 管理器 " 配置 ODBC 数据源 "。有关在 UNIX 上设置 ODBC 数据 源的信息,请参阅 "ASA 数据库管理指南 > 连接到数据库 > 在 UNIX 上 使用 ODBC 数据源 "。

✤ 创建简单 ODBC 数据源:

- 1 启动 **ODBC** 管理器:
 - 从 Windows 的 " 开始 " 菜单, 选择 " 程序 " → "Sybase SQL Anywhere 8" → "Adaptive Server Anywhere" → "ODBC 管理 器 "。

或者 从 Sybase Central,选择 " 工具 " → "Adaptive Server Anywhere 8" → "ODBC 管理器 "。

ODBC 数据源管理器随即出现,显示当前安装在您的计算机上的数据源的列表。例如,

2 在 " 用户 DSN" 选项卡上,单击 " 添加 "。

"创建新数据源"向导随即出现。

3 从驱动程序列表中选择 Adaptive Server Anywhere 8.0, 然后单击"完成"。

"Adaptive Server Anywhere ODBC 配置 "对话框随即出现。

该对话框中的许多字段都是可选的。单击窗口右上角的问号,

◎ ODBC 数据源管理器		? X
用户 DSN 系統 DSN 文件 用户数据源(U): 名称 ASA 8.0 Client Sample ASA 8.0 Sample dBASE Files dBase Files - Word Excel Files FoxFro Files - Word MS Access Database peat pfc	DSN 驱动程序 跟踪 连接: <u> 地动程序</u> Adaptive Server Anywhere Adaptive Server Anywhere Microsoft dBase Driver () Microsoft Excel Driver () Microsoft Access Driver Microsoft Access Driver Adaptive Server Anywhere Adaptive Server Anywhere	也 关于 添加 (D) 册除 (E) 配置 (C)
▲ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● <t< td=""><td>和4401792 Server Allywitere</td><td>接的信息。用户 器上。 (A) 帮助</td></t<>	和4401792 Server Allywitere	接的信息。用户 器上。 (A) 帮助

然后单击某一条目字段以找到与该字段有关的更多信息。您可 能只需要使用以下参数:

数据源名("ODBC"选项卡) 键入将在"连接"对话框
 中出现的名称。它可以包含空格,但应是短名称。

例如,在前一章中创建的数据库中,您可以输入 My Sample 作为数据源名称。

• **用户ID("登录"选项卡)**您将用来进行连接的数据 库用户ID。如果省略了该用户ID,系统将在尝试连接时 提示您。

对于新创建的数据库,缺省用户 ID 是 DBA。

• **口令("登录"选项卡)**如果存储在您的计算机上的口 令有安全问题,应该忽略该口令或给该口令加密。如果忽 略该口令,系统将在您尝试连接时进行提示。

DBA 用户 ID 的缺省口令是 SQL。

• 数据库文件("数据库"选项卡) 您可以通过浏览计算 机选择数据库文件。

Adaptive Server Anywhere 8的 ODBC 配置 🛛 ? 🗙
ODBC 登录 数据库 网络 高级
数据源名 (S):
说明@):
隔离级别 (0):
└─ Microsoft 应用程序(SQLStatistics 中的键)创)
□ Delphi 应用程序(H)
□ 抑制读取警告信息 (k)
□ 防止 "驱动程序不支持" 错误 (P)
□ 延迟自动提交直到语句关闭 (U)
- 描述游标行为
○ 永不 (21) ○ 根据需要 ○ 始终 (21)
转换器 (T): <tr
选择转换器 (2)
确定取消

对于在前一章中创建的数据库,您将输入 c:\temp\mysample.db。

4 在您指定所需参数后,单击 "确定 "将创建数据源并关闭该对 话框。 对于数据库连接的完整说明,请参阅 "ASA 数据库管理指南 > 连接到数 据库 "。

有关 ODBC 数据源的更多信息,请参阅 "ASA 数据库管理指南 > 连接到数据库 > 使用 ODBC 数据源 "。

^{第7章} 使用 Interactive SQL

关于本章本章讨论如何运行和使用 Interactive SQL 。 Interactive SQL 是 Adaptive
Server Anywhere 附带的一个实用程序;您可以使用它来执行 SQL 语
句、建立脚本以及显示数据库数据。

Interactive SQL 介绍

Interactive SQL 是用于将 SQL 语句发送到数据库服务器的实用程序。您可以将其用于以下目的:

- 浏览数据库中的信息。
- 检验要包括在应用程序中的 SQL 语句。
- 将数据装载到数据库中并执行管理任务。

另外, Interactive SQL 可以运行**命令文件**或**脚本文件**。例如,您可以建 立针对数据库运行的可重复使用的脚本,然后使用 Interactive SQL 以批 处理的方式执行这些脚本。

关于本章

下表列出了如何运行和使用 Interactive SQL 的最常用的信息。

如果您要了解	那么请参阅
如何启动 Interactive SQL	" 启动 Interactive SQL" 第 74 页
如何连接到数据库	"将您的应用程序连接到其数据 库 " 第 65 页
如何使用 Interactive SQL 工具栏	"使用 Interactive SQL 工具栏 " 第 76 页
如何打开新的 Interactive SQL 窗口	"打开多个窗口"第76页
在何处查找键盘快捷方式的信息	"键盘快捷方式"第77页
如何显示数据	"使用 Interactive SQL 显示数据 "第 80 页
如何在 Interactive SQL 中执行 SQL 命 令	"在 Interactive SQL 中使用 SQL 语句 " 第 85 页
如何设置 Interactive SQL 选项	" 配置 Interactive SQL" 第 91 页

如果您要了解	那么请参阅
在何处查找有关选择数据的更多详细	"从数据库表中选择数据"第
信息	103页
在何处查找有关装载和卸载数据的更	"ASA SQL 用户指南 > 导入和导出
多信息	数据 > 导入和导出介绍 "
如何实现常用任务自动化	"保存、装载和运行命令文件" 第 98 页

启动 Interactive SQL

您可以通过两种方式启动 Interactive SQL: 从 Sybase Central 启动或通过 其自身启动。通过其自身启动 Interactive SQL 的方式取决于您的操作系 统。

如果通过其自身启动 Interactive SQL,则会显示 "连接"对话框,该对话框允许您连接到数据库,就像在 Sybase Central 中所进行的操作一样。

有关连接到数据库的详细信息,请参阅 "ASA 数据库管理指南 > 连接到数据库 "。

◆ 从 Sybase Central 启动 Interactive SQL:

 若要启动 Interactive SQL,请右击 Sybase Central 中的某个数据 库,然后选择 " 文件 " → " 打开 Interactive SQL"。在这种情况 下, Interactive SQL 自动连接到该数据库。

或者

 2 如果启动 Interactive SQL 而不连接到数据库,请选择 " 工具 " → "Adaptive Server Anywhere 8" → " 打开 Interactive SQL"。即 会出现 " 连接 " 对话框。

✤ 启动 Interactive SQL:

 选择 "开始 "→ "程序 "→ "Sybase SQL Anywhere 8" → "Adaptive Server Anywhere" → "Interactive SQL"。即会出现 "连接 "对话框。

或者

2 在命令提示符处键入以下命令:

dbisql

即会出现"连接"对话框。

输入连接到数据库所需的信息,然后单击 "确定 "进行连接。

主窗口说明 Interactive SQL 具有以下窗格:

- SQL 语句 提供一个键入 SQL 语句的位置。
- 结果 显示所执行的命令的结果。例如,如果使用 SQL 语句在数据库中搜索特定的数据,则此窗格中的 "结果 "选项卡会显示符合搜索条件的列和行。如果信息超出窗格的大小,则自动在窗格中显示滚动条。您可以在 "结果 "选项卡上编辑结果集。

有关编辑结果集的更多信息,请参阅 "在 Interactive SQL 中编辑表值 "第 81 页。

" 消息 " 选项卡显示数据库服务器中有关您在 Interactive SQL 中执 行的 SQL 语句的消息。您可以在 " 选项 " 对话框的 " 消息 " 选项 卡中为此选项卡指定选项。

"结果 "窗格中的 "计划 "选项卡和 "UltraLite 计划 "选项卡显示 SQL 语句的查询优化程序的执行计划。可以在 "选项 "对话框的 "计 划 "选项卡上为这些选项卡指定选项。

窗口顶部的标题栏显示连接信息,如下所示:

database-name (userid) on Server-name

例如,如果连接到示例数据库,则标题栏如下所示:

asademo (dba) on asademo

使用 Interactive SQL 工具栏

Interactive SQL 工具栏显示在 Interactive SQL 窗口的顶部。它为您提供 执行常用命令的按钮。



通过此工具栏上的按钮,您可以执行以下操作:

- 调用历史列表中紧接在当前位置之前的已执行的 SQL 语句。
- 查看多达 50 个以前已执行的 SQL 语句的列表。
- 调用历史列表中紧接在当前位置之后的已执行的 SQL 语句。
- 执行当前显示在 "SQL 语句 " 窗格中的 SQL 语句。
- 中断当前 SQL 语句的执行。

作为这些按钮所执行操作的简单提示,您可以将光标保持在每个按钮上 方来查看弹出式说明。

打开多个窗口

可以打开多个 Interactive SQL 窗口。每个窗口对应于一个单独的数据库 连接。可以同时连接到不同服务器上的两个(或多个)不同的数据库, 或者可以打开到单个数据库的并发连接。

✤ 打开新的 Interactive SQL 窗口:

- 1 选择 "窗口 "→ "新建窗口 "。即会出现 "连接 "对话框。
- 在"连接"对话框中,输入连接选项,然后单击"确定"进行 连接。

连接信息(包括数据库名称、用户 ID 和数据库服务器)将出现在 "SQL 语句" 窗格上方的标题栏上。

您还可以使用 "SQL" 菜单中的 " 连接 " 和 " 断开连接 " 命令或通过执行 CONNECT 或 DISCONNECT 语句,连接到数据库或从数据库断开连 接。

键盘快捷方式

Interactive SQL 提供以下键盘快捷方式:

功能键	说明
Alt+F4	退出 Interactive SQL。
Alt+ 左箭头键	显示历史列表中的上一个 SQL 语句。
Alt+右箭头键	显示历史列表中的下一个 SQL 语句。
Ctrl+Break	中断正在被执行的 SQL 语句。
Ctrl+C	将选定的行和列标题复制到 "结果 " 窗格中的剪贴板。 在 "SQL 语句 " 窗格中,将选定的文本复制到剪贴板。
Ctrl+End	移动到当前窗格的底部。
Ctrl+F6	在 Interactive SQL 窗口间循环。
Ctrl+H	显示当前会话过程中所执行的 SQL 语句的历史记录。
Ctrl+Home	移动到当前窗格的顶部。
Ctrl+N	清除 Interactive SQL 窗口中的内容。

功能键	说明
Ctrl+Q	显示查询编辑器。 查询编辑器帮助您建立 SQL 查询。当完成建立查询后, 单击 "确定 "将其输出回 "SQL 语句 " 窗格中。
Ctrl+S	保存 "SQL 语句 " 窗格中的内容。
Esc	清除 "SQL 语句 " 窗格中的内容。
F1	打开 " 帮助 "。
F2	编辑结果集中选定的值。您可以在行中使用 Tab 键在列 之间切换。
F5	执行 "SQL 语句 " 窗格中的所有文本。 您还可以通过单击工具栏上的 " 执行 SQL 语句 " 按钮来 执行此操作。
F7	显示 " 查找表名称 " 对话框。 在此对话框中,您可以查找并选择一个表然后按 Enter 键 将该表名称插入 "SQL 语句 " 窗格中光标所在的位置。 或者,在列表中选中某个表时,再次按 F7 键显示该表中 的列。然后您可以选择一个列并按 Enter 键将该列名称插 入 "SQL 语句 " 窗格中光标所在的位置。
F8	显示 " 查找过程名称 " 对话框。 在此对话框中,您可以查找并选择一个过程然后按 Enter 键将该过程名称插入 "SQL 语句 " 窗格中光标所在的位 置。
F9	执行 "SQL 语句 " 窗格中选中的文本。 如果未选中任何文本,则执行所有语句。
Pgdn	在当前窗格中向下移动。
Pgup	在当前窗格中向上移动。
Shift+F5	刷新计划,不执行在 "SQL 语句 " 窗格中输入的语句。

当 "SQL 语句 " 窗格为激活窗口时, 可以使用以下键盘快捷方式:

功能键	说明
Ctrl+]	将光标移动到匹配括号。可以指定以下类型之一:()、 {}、[]或 <>。
Ctrl+ 退格键	删除光标左边的单词。
Ctrl+Del	删除光标右边的单词。
Ctrl+F3	查找选定文本的下一个出现位置。
Ctrl+G	打开 "转到 "对话框,您可以在该对话框中指定要转到 的行。
Ctrl+L	从 "SQL 语句 " 窗格中删除当前行并将该行放置到剪贴 板上。
Ctrl+Shift+]	将选择扩展到匹配的括号。
Ctrl+Shift+F3	查找选定文本的上一个出现位置。
Ctrl+Shift+L	删除当前行。
Ctrl+Shift+U	将选择内容更改为大写字符。
Ctrl+U	将选择内容更改为小写字符。
Home	将光标移动到当前行的开始位置或当前行的第一个单 词。
Shift+F3	查找选定文本的上一个出现位置。
Shift+Home	将选择扩展到当前行上文本的开始位置。

使用 Interactive SQL 显示数据

Interactive SQL 的一个主要用途是浏览数据库中的信息。本节说明如何 在示例数据库中查询信息。

您可以在 Interactive SQL 中使用 SELECT 语句显示数据库信息。下面的 示例显示要在 "SQL 语句 " 窗格中键入的命令。键入命令之后,必须单 击工具栏上的 " 执行 SQL 语句 " 按钮来执行该命令。

执行完该语句之后,数据(称为结果集)显示在"结果"窗格中的"结果"选项卡上。您可以使用滚动条查看表中位于窗格当前视图之外的区域。

✤ 列出职员表的全部列和行:

- 1 启动 Interactive SQL 并连接到示例数据库。
- 2 在 "SQL 语句 " 窗格中键入以下内容:

SELECT * FROM employee

3 在工具栏上,单击"执行 SQL 语句" 按钮。

emp_id	manager_id	emp_Iname	emp_fname	
102	501	Fran	Whitney	
105	501	Matthew	Cobb	
129	902	Philip	Chin	
148	1293	Julie	Jordan	

有关 Select 语句的更多信息,请参阅 "从数据库表中选择数据 " 第 103 页。

您可以在结果集中添加、删除和更新行。

有关编辑结果集的更多信息,请参阅 "在 Interactive SQL 中编辑表值 "第 81 页。

在 Interactive SQL 中编辑表值

在 Interactive SQL 中执行查询之后,可以编辑结果集来修改数据库。还可以从结果集中选择行,并复制它们以用于其它应用程序。 Interactive SQL 支持编辑、插入和删除行。这些操作与执行 UPDATE、 INSERT 和 DELETE 语句具有相同的结果。

在您可以复制、编辑、插入或删除行之前,您必须在 Interactive SQL 中执行查询,该查询在 "结果 "窗格的 "结果 "选项卡上返回结果集。当您直接编辑结果集时, Interactive SQL 创建并执行一个 SQL 语句,该语句对数据库表执行您的更改。

若要编辑结果集中的某个行或某个值,您必须在要在其中修改值的表或 列上具有必要的权限。例如,如果要删除某行,那么您必须具有对该行 所属的表的 DELETE 权限。

如果您执行以下操作,编辑结果集可能失败:

- 试图编辑您不具有权限的行或列。
- 从具有主键的表中选择列,但没有选择所有主键列。
- 试图编辑 JOIN (例如,结果集中的某个数据来自多个表)的结果 集。
- 输入无效值(例如,在数字列中输入字符串或在不允许空值的列 中输入空值)。

当编辑失败时,将出现一条解释该错误的 Interactive SQL 错误消息,数据库表值保持不变。

对表值进行了更改之后,必须输入 COMMIT 语句使更改成为永久更改。如果要撤消更改,则必须执行 ROLLBACK 语句。

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 语句 > COMMIT 语 句 " 和 "ASA SQL 参考手册 > SQL 语句 > ROLLBACK 语句 "。

在 Interactive SQL 结果集中编辑表值

在 Interactive SQL 中,可以更改数据库表中现有行内的任何或所有值。 您必须在被修改的列上具有 UPDATE 权限。当编辑结果集时,一次只 能对一行中的值进行更改。

✤ 编辑结果集中的行:

- 1 单击您要更改的值。
- 2 右击结果集,然后从弹出式菜单中选择 "编辑"。您还可以按 F2 键来编辑结果集。

一个闪烁的光标会出现在包含该值的表单元格中。

3 输入新值。

您不能在列中输入无效数据类型。例如,您不能将字符串输入 接受 INT 数据类型的列中。

如果您完成了对行中值的编辑,按 Enter 键更新数据库。如果 您要更改该行中的其它值,按 Tab 键或 Shift+Tab 组合键移动到 其它值。

您可以按 Esc 键取消对选定值已进行的更改。

4 执行 COMMIT 语句使您对表的更改成为永久更改。

在结果集中编辑值之后,将用修改后的值更新数据库。不管计算列是不 是结果集的一部分,都将根据修改的值对计算列重新进行计算。但是, 如果结果集中有计算列,并且在计算列中修改了某个值,则用修改后的 值更新数据库。

将行从 Interactive SQL 结果集插入数据库

Interactive SQL 还允许您将新行添加到结果集中。您使用 Tab 键在结果 集的列之间切换以将值添加到该行。当您将值添加到表中之后,字符按 所输入的大小写进行存储。您必须在要添加新行的表上具有 INSERT 权 限。

✤ 将新行插入结果集:

1 右击结果集,然后从弹出式菜单中选择 "添加 "。

即会出现一个新的空行,在该行的第一个值中出现闪烁的光标。

按 Tab 键在行中列之间移动光标。您还可以通过单击选定行中的值插入一个值。

2 输入新值。

不能在列中输入无效数据类型。例如,不能将字符串输入接受 INT 数据类型的列中。

- 3 按 Tab 键移动到下一列。
- 4 重复第2步,直到添加完所有列值。
- 5 按 Enter 键更新数据库。
- 6 执行 COMMIT 语句使您对表的更改成为永久更改。

如果结果集包含计算列并且您没有为该计算列指定值,则当更新数据库 时将计算该值。但是,如果您为计算列指定了一个值,则将用该指定值 更新数据库,而不会为计算列计算一个值。

使用 Interactive SQL 从数据库中删除行

您还可以在 Interactive SQL 中从数据库表中删除行。您必须在要删除行的表上具有 DELETE 权限。您只能选择结果集中连续的行。

✤ 从结果集中删除行:

- 1 选择要删除的行。若要选择行,请执行以下操作:
 - 按住 Shift 键单击行
 - 按住 Shift 键使用向上或向下箭头

如果要删除不连续的行,您必须分别删除每行。

2 右击结果集并从弹出式菜单中选择 " 删除 "。您还可以通过按

Delete 键删除选定的行。

将从数据库表中删除选定的行。

3 执行 COMMIT 语句使您对表的更改成为永久更改。

从 Interactive SQL 结果集中复制行

您可以直接从 Interactive SQL 的结果集中复制行,然后将它们粘贴到其 它应用程序中。复制行还将复制列标题。复制的数据是用逗号分隔的, 这允许其它应用程序 (例如 Microsoft Excel)正确地设置所复制数据的 格式。复制的数据为 ASCII 格式,所有字符串都括在单引号中。只能选 择结果集中连续的行。

✤ 从 Interactive SQL 结果集中复制行:

- 1 选择您要复制的行。若要选择行,请执行以下操作:
 - 按住 Shift 键的同时单击行
 - 按住 Shift 键的同时使用向上或向下箭头
- 2 右击结果集并从弹出式菜单中选择 "复制 "。您还可以通过按 CTRL+C 组合键复制选定的行。

选定的行 (包括它们的列标题)被复制到剪贴板。您可以将它 们粘贴到其它应用程序中。

在 Interactive SQL 中使用 SQL 语句

下面几节介绍您可以在 Interactive SQL 中使用的一些命令。本节介绍在 Interactive SQL 中使用命令的常规任务。

所有 SQL 语句都可以作为命令输入到 Interactive SQL 窗口的顶部窗格中。完成键入后,您需要执行该语句来查看其结果。

- ✤ 若要执行 SQL 语句,请执行以下操作之一:
 - 按 "执行 SQL 语句 " 按钮, 或者 选择 "SQL" → " 执行 " 或 按 F5 键。
- ☆ 清除 "SQL 语句 " 窗格:
 - 选择 " 编辑 " → " 清除 SQL" 或 按 Escape 键。

取消 Interactive SQL 命令

Interactive SQL 工具栏上的 "中断 " 按钮取消命令。

"取消"操作停止当前的处理并提示输入下一个命令。如果当时正在处理命令文件或者"SQL语句"窗格中有多个命令,则提示选择要采取的操作(停止命令文件、继续或退出Interactive SQL)。可以通过Interactive SQL ON_ERROR选项来控制这些操作。

有关 ON_ERROR 选项的更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > ON_ERROR 选项 [ISQL]"。

- **报告的错误** 当检测到中断时,将根据检测到中断的时间报告三种不同的错误之一。
 - 1. 如果中断是由 Interactive SQL 在处理请求时检测到的 (相对于数据 库服务器),则将出现以下消息:

ISQL 命令被用户终止

Interactive SQL 立即停止处理,搁置当前数据库事务。

 如果中断是由数据库服务器在处理数据操作命令(SELECT、 INSERT、DELETE 或 UPDATE)时检测到的,则将出现以下消息:

语句被用户中断。

将撤消当前命令的效应,但事务的剩余部分保持完整。

3. 如果中断是由数据库服务器在处理数据定义命令(CREATE、 DROP、ALTER等)时检测到的,则将出现以下消息:

被用户终止 -- 事务已回退

由于数据定义命令都在命令启动前自动执行 COMMIT,所以 ROLLBACK 的效应是仅仅取消当前的命令。

当数据库服务器正在以批量操作模式运行执行修改数据库的命令 (INSERT、UPDATE和DELETE)时也会出现此消息。在这种情况下,ROLLBACK不仅取消当前的命令,还取消上一次COMMIT 之后执行的所有操作。在某些情况下,数据库服务器可能要花相当 长一段时间执行自动ROLLBACK。

组合多个语句

Interactive SQL 环境允许同时输入多个语句。可以通过在每个语句的结 尾加一个分号 (;) 来实现此目的。命令分隔符是 Interactive SQL 中的一 个可配置的选项,您可以使用 COMMAND_DELIMITER 选项对其进行 更改。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > COMMAND_DELIMITER 选项 [ISQL]"。

◆ 在 "SQL 语句 " 窗格中输入多个语句:

1 在 "SQL 语句 " 窗格中输入以下三个命令。

```
UPDATE employee
SET dept_id = 400,
manager_id = 1576
WHERE emp_id = 467;
UPDATE employee
```

SET dept_id = 400, manager_id = 1576 WHERE emp_id = 195; SELECT * FROM employee WHERE emp_id IN (195, 467);

- 2 在工具栏上,单击 "执行 SQL 语句 "按钮。将执行所有三条语句。执行后,这些命令会保留在 "SQL 语句 "窗格中。如果要 清除此窗格,请按 Esc 键。
- 3 通过输入 ROLLBACK 并执行该语句来回退您的更改。

使用 go 作为替代

如果不用分号,替代方法是在一行的开头单独输入 go。

```
UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 467
go
UPDATE employee
SET dept_id = 400,
    manager_id = 1576
WHERE emp_id = 195
go
SELECT *
FROM employee
WHERE emp_id IN ( 195, 467 )
go
```

提示

您可以按 F9 键只执行 "SQL 语句 " 窗格中的选定文本。

查找表、列和过程

当在 Interactive SQL 中输入命令时,您可以查找存储在当前数据库中的 表、列或过程的名称并将它们插入光标当前所在的位置。

✤ 在数据库中查找表的名称:

- 1 选择 "工具 "→ " 查寻表名 "。
- 2 查找并选择该表。
- 3 单击 " 确定 ", 把表名插入 "SQL 语句 " 窗格中的当前光标位 置。

✤ 在数据库中查找列名称:

- 1 选择 "工具 "→ " 查寻表名 "。
- 2 查找并选择包含该列的表。
- 3 单击"显示列"。
- 4 选择该列并单击 "确定 ",把列名插入 "SQL 语句 " 窗格中的当前光标位置。

◆ 在数据库中查找过程的名称:

- 1 选择"工具"→"查寻过程名"。
- 2 查找并选择该过程。
- 3 单击 "确定 ",把过程名插入 "SQL 语句 " 窗格中的当前光标位 置。

在表和过程查找对话框中,您可以输入所要查找的表或过程的前几个字符。在字段中键入一些内容之后,对话框缩小列表的范围,只包括以您 输入的文本开头的那些项。

您可以使用标准 SQL 通配元字符 % 来表示 "匹配任何内容 "。例如, 若要只列出以 order 结尾的那些表,请在查找字段中键入 %order。若要 列出包含单词 java 的所有项,请键入 %java%。从查找字段中清除搜索 条件将使过滤条件复位以显示所有项。

调用命令

当您执行命令时, Interactive SQL 将其自动保存在一个历史记录列表中,该历史记录列表保持到当前会话结束。Interactive SQL 维护的记录 包含多达 50 个最近的命令。

您可以在 " 命令历史记录 " 对话框中查看整个命令列表。若要访问 " 命 令历史记录 " 对话框,请按 CTRL+H 组合键或单击工具栏上的手册图 标。

🔜 命令历史记录	×
<u>S</u> QL语句(S):	
select * from employee	
select * from department	
	F
	确定 取消

最近的命令显示在列表的顶部。若要调用某个命令,请将其突出显示,然后单击 "确定 "。它将出现在 Interactive SQL 的 "SQL 语句 " 窗格中。

您还可以在不使用 " 命令历史记录 " 对话框的情况下调用命令。使用工 具栏中的箭头在命令中向后或向前滚动, 或者按 ALT+ 右箭头和 ALT+ 左箭头组合键。

您还可以将命令保存在文本文件中,以便在随后的 Interactive SQL 会话中使用它们。

有关更多信息,请参阅 "保存、装载和运行命令文件 "第98页。

记录命令

使用 Interactive SQL 记录功能,您可以在执行命令时记录这些命令。 Interactive SQL 将一直进行记录,直到您停止记录进程或直到您结束当 前会话。记录的命令存储在日志文件中。

✤ 开始记录 Interactive SQL 命令:

- 1 选择 "SQL" → " 开始记录 "。
- 2 在"另存为"对话框中,为日志文件指定一个位置和名称。
- 3 完成后单击 "保存"。

✤ 停止记录 Interactive SQL 命令:

• 选择 "SQL" → " 停止记录 "。

提示

您还可以在 "SQL 语句 " 窗格中键入语句来开始和停止记录。若要开始 记录,请键入并执行 START LOGGING 'c:\filename.sql',其中 c:\filename.sql 是日志文件的路径、名称和扩展名。日志文件必须 具有 .SQL 扩展名。如果路径包含嵌套的空格,您只需要包括单引号即 可。若要停止记录,请键入并执行 STOP LOGGING。

开始记录之后,将记录您尝试执行的所有命令,包括那些未正确执行的 命令。

配置 Interactive SQL

您可以在"选项"对话框中配置 Interactive SQL。此对话框提供命令、 外观、导入 / 导出功能和消息的设置。

可以从界面或使用 SET OPTION 语句设置每个选项。

◆ 访问"选项"对话框:

• 选择 " 工具 " → " 选项 "。

下面几节将介绍 " 选项 " 对话框的各选项卡。

"常规"选项卡

- "选项"对话框的"常规"选项卡具有以下组件。
- 提交选项 允许您选择提交事务的时间。您可以选择在执行每条 语句后和/或当您退出 Interactive SQL 会话时自动提交。还可以通 过在任何适当的时候输入显式 COMMIT 命令手动提交。缺省设置 是在退出时提交。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > AUTO_COMMIT 选项 [ISQL]"。

- **命令文件选项** 允许您配置 Interactive SQL 如何响应错误并指定是 否将命令文件复制到日志中。
 - 出现错误时 缺省设置是 PROMPT, 这表示 Interactive SQL 显示一个消息框询问您是否要继续。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > ON_ERROR 选项 [ISQL]"。

 将命令文件回写到日志 当您开始记录 SQL 语句时,此选项 使从脚本文件(或命令文件)执行的 SQL 语句与交互输入 的 SQL 语句一起复制到日志中。如果禁用此选项,则当您开 始记录时,只将交互输入的 SQL 语句复制到日志中。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > ECHO 选项 [ISQL]"。

"结果"选项卡

"选项"对话框的"结果"选项卡具有以下组件。

• **将空值显示为** 允许您指定在浏览数据时空值在表列中的显示方 式。

缺省设置是 (NULL)。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > NULLS 选项 [ISQL]"。

- **要显示的最大行数** 允许您限制显示的行数。缺省设置是 500。
- **截断长度** 允许您限制 Interactive SQL 的 "结果 " 窗格中每列中显示的字符数。缺省设置是 30。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > TRUNCATION_LENGTH 选项 [ISQL]"。

 显示多个结果集 启用或禁用多个结果集的显示。例如,您可以 在创建包含多个 SELECT 语句的过程时使用此功能。如果启用此 选项,则当调用该过程时,您可以看到每个结果集都在"结果" 窗格中一个单独的选项卡上。

如果您正在使用 jConnect 驱动程序,则选择显示多个结果集在显示任何行之前需要 Interactive SQL 等待完成对整个结果集的检索。对于较大的结果集,这可能导致较长时间的等待。

缺省情况下,此选项关闭。

• **自动重新读取结果** 启用或禁用 Interactive SQL 在您执行完 INSERT、UPDATE 或 DELETE 语句后自动重新生成最新结果集 的功能。例如,如果您正在用显示将受影响的行的"结果"窗格中的"结果"选项卡更新一个表,则此选项使 Interactive SQL 自动刷新"结果"选项卡以反映执行的更改。

缺省情况下,此选项打开。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > AUTO_REFETCH 选项 [ISQL]"。

- 在主控台模式中运行时,您希望看到哪些结果集? 以下选项可以指定当运行 SQL 文件时,将输出哪些结果集。该选项在窗口模式下运行时没有效果,它是针对各个计算机分别设置的。设置了该选项后,如果不单击"设为永久",当您退出 Interactive SQL 时该设置即被放弃。
 - 最后一个 输出文件中最后一个语句的结果集。
 - 全部 为文件中每一个返回结果集的语句输出结果集。
 - 无 不输出任何结果集。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > ISQL_PRINT_RESULT_SET 选项 [ISQL]"。

"导入/导出"选项卡

"选项"对话框的"导入/导出"选项卡具有以下组件。

缺省导出格式 允许您选择缺省的导出文件格式。此格式在"保存"对话框的"文件类型"字段中自动选中,但您仍可以选择其它格式。当 Interactive SQL 在没有指定格式的情况下解释 OUTPUT 语句时也将使用该缺省格式。

缺省设置是 ASCII。

有关导出格式的更多信息,请参阅 "ASA 数据库管理指南>数据 库选项>OUTPUT_FORMAT 选项 [ISQL]"。 缺省导入格式 允许您选择缺省的导入文件格式。此格式在"打开"对话框的"文件类型"字段中自动选中,但您仍可以选择其它格式。当 Interactive SQL 在没有指定格式的情况下解释 INPUT 语句时也将使用该缺省格式。

缺省设置是 ASCII。

有关导入格式的更多信息,请参阅 "ASA 数据库管理指南>数据 库选项>INPUT_FORMAT 选项 [ISQL]"。

 ASCII 选项 允许您指定当以 ASCII 格式导入或导出数据时用于 字段分隔符、引号字符串和转义字符的缺省符号。

缺省设置是将逗号(,)用作字段分隔符、将撇号()用作引号字符 串、将反斜线()用作转义字符。

有关这些选项的更多信息,请参阅 "ASA 数据库管理指南 > 数据 库选项 > Interactive SQL 选项 "。

有关导入和导出的更多信息,请参阅 "ASA SQL 用户指南 > 导入和导出数据 "。

"消息"选项卡

"选项"对话框的"消息"选项卡具有以下组件。

- 测量 SQL 语句的执行时间 启用或禁用 Interactive SQL 度量执行 语句所花时间的功能。当启用此选项 (这是缺省设置)后,时间 显示在 "消息 "窗格中。
- 显示单独的"消息"窗格 允许您指定来自数据库服务器的消息的显示位置。缺省情况下,消息显示在"结果"窗格的"消息"选项卡上。如果您选择此选项,数据库服务器信息将显示在 Interactive SQL窗口中单独的"消息"窗格中。
- "消息"窗格中的缺省行数 允许您指定"消息"窗格的最初高度 (按行数计算)。缺省设置为7行。
"计划"选项卡

"选项"对话框的"计划"选项卡具有以下组件。

- 计划选项 允许您确定给出的有关优化查询方式的详细信息的级别。从以下执行计划选项中进行选择:
 - 图形式计划 执行计划在"计划"选项卡中显示为树型图。
 您可以单击计划图中的节点以查看有关该查询部分的详细信息。此计划是缺省设置。
 - 带统计的图形式计划 执行计划在"计划"选项卡中显示为 树型图,您可以单击节点以查看有关该查询部分的详细信息。还将显示统计信息,指示选定的查询部分所使用的资源。
 - 简要计划 在"结果"窗格"计划"选项卡上用一行文字显示有关执行的基本信息。该行列出访问的表以及是否该行将 被按顺序读取或通过索引访问。
 - **详细计划** 在"计划"选项卡上用多行文字显示有关执行的 详细信息。
- 假设只读游标 如果您希望查询优化程序处理查询的方式就如同 为只读游标执行过该查询一样,请选择该选项。缺省情况下不选 择该选项,表示优化程序应该为"可读可写"游标获取计划。

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 函数 > PLAN 函数 [杂类]"。

假设游标为 您可以根据您指定的游标类型来获取计划。查询优化程序可以假设游标为 "敏感性未定型 "(Asensitive)、 "不敏感"(Insensitive)、 "敏感 "(Sensitive) 或 "由键集驱动 "(Keyset-driven)。缺省为 "敏感性未定型 "。

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 函数 > PLAN 函数 [杂类]"、"ASA 编程指南 > 在应用程序中使用 SQL > 敏感 性未定型游标 "、"ASA 编程指南 > 在应用程序中使用 SQL > 不敏 感游标 "、"ASA 编程指南 > 在应用程序中使用 SQL > 敏感游标 "和"ASA 编程指南 > 在应用程序中使用 SQL > 对值敏感的游标 "。

• 显示 UltraLite 计划 如果想要在 Interactive SQL"结果"窗格中用 一个单独的选项卡显示 UltraLite 计划,请选择此选项。"UltraLite 计划"选项卡使用您指定的计划格式显示 UltraLite 计划优化策 略。对于某些查询, UltraLite 执行计划可能会与为 Adaptive Server Anywhere 选择的计划有所不同。

通过选择 "图形 "、 "简要计划 " 或 " 详细计划 " 类型之一来控制 UltraLite 计划类型。 UltraLite 计划无统计信息可用。此选项是缺 省选定的。

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 函数 > GRAPHICAL_ULPLAN 函数 [杂类]"。

有关执行计划的更多信息,请参阅 "ASA SQL 用户指南 > 查询优化与执行 > 读取访问计划 "。

运行命令文件

本节介绍如何处理由一组命令组成的文件。

将输出写入文件

在 Interactive SQL 中,在执行下一个命令之前,每个命令的数据一直保留在"结果"窗格中的"结果"选项卡上。若要记录您的数据,您可以将每个语句的输出保存到单独的文件中。如果 statement1 和 statement2 是两个 SELECT 语句,那么您可以分别将其输出到 file1 和 file2 中,如下所示:

Statement1; OUTPUT TO file1; statement2; OUTPUT TO file2;

例如,以下命令保存查询的结果:

SELECT * FROM EMPLOYEE; OUTPUT TO "C:\\My Documents\\Employees.txt"

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 语句 > OUTPUT 语 句 [Interactive SQL]"。

执行命令文件

您可以通过以下方式执行命令文件:

• 您可以使用 Interactive SQL READ 命令执行命令文件。

以下语句执行文件 temp.sql:

READ temp.sql

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 语句 > READ 语句 [Interactive SQL]"。

• 您可以将命令文件装载到 "SQL 语句 " 窗格中并在那里直接执行该 命令文件。

通过选择 " 文件 " → " 打开 " 将命令文件装载回 "SQL 语句 " 窗

格。当提示输入文件名时,请输入 temp.sql。

• 您可以提供一个命令文件作为 Interactive SQL 的命令行参数。

保存、装载和运行命令文件

您可以保存当前显示在 "SQL 语句 " 窗格中的命令, 以便将它们用于将 来的 Interactive SQL 会话。在其中保存命令的文件称为命令文件。

命令文件是包含 SQL 语句的文本文件。您可以使用任何编辑器来创建 命令文件。可以将注释行与要执行的 SQL 语句包括在一起。通常,命 令文件还称为**脚本**。

当开始新会话时,您可以将命令文件的内容装载到 "SQL 语句 " 窗格 中,或者您可以直接运行内容。

✤ 将 "SQL 语句 " 窗格中的命令保存到文件中:

- 1 单击"文件"→"另存为"。
- 2 在"保存"对话框中,为该文件指定位置、名称和格式。完成 后单击"保存"。

✤ 将命令从文件装载到 "SQL 语句 " 窗格中:

- 1 选择"文件"→"打开"。
- 2 在"打开"对话框中,查找并选择该文件。完成后单击 "打开"。

◆ 直接运行命令文件:

- 1 选择 "文件 "→ "运行脚本 "。
- 在"打开"对话框中,查找并选择该文件。完成后单击 "打开"。
 - "运行脚本"菜单项等同于 READ 语句。例如,在 "SQL 语句"

窗格中,您还可以通过键入以下内容运行命令文件:

READ 'c:\filename.sql'

其中 c:\filename.sql 是该文件的路径、名称和扩展名。只有 当路径包含空格时才需要使用单引号 (如上所示)。

✤ 以批处理模式运行命令文件:

• 提供命令文件作为 Interactive SQL 的命令行参数。

例如,以下命令对示例数据库运行命令文件 myscript.sql。

dbisql -c "dsn= ASA 8.0 Sample" myscript.sql

在 Interactive SQL 中使用 SQL 转义语法

Interactive SQL 支持 JDBC 转义语法。此转义语法允许您从 Interactive SQL 调用存储过程而不管您正在使用的数据库管理系统。转 义语法的一般格式是

{{ keyword parameters }}

必须 使用双括号。这种成对使用括号的方法是 Interactive SQL 特定的。 连续括号之间一定不能有空格: "{{"是可接受的,但 "{{"是不被接受的。同样,您不能在语句中使用新行字符。因为转义语法不由 Interactive SQL 执行,所以转义语法不能用于存储过程。

您可以使用转义语法访问由 JDBC 驱动程序实现的函数库,这些函数包括数字、字符串、时间、日期和系统函数。

例如,若要以与数据库管理系统无关的方式获得当前用户的名称,您将 键入以下内容:

```
select {{ fn user() }}
```

可用的函数取决于您正在使用的 JDBC 驱动程序。下表列出了 jConnect 和 JDBC-ODBC bridge 支持的函数。

数字函数	字符串函数	系统函数	时间/日期函数
ABS	ASCII	DATABASE	CURDATE
ACOS	CHAR	IFNULL	CURTIME
ASIN	CONCAT	USER	DAYNAME
ATAN	DIFFERENCE	CONVERT	DAYOFMONTH
ATAN2	LCASE		DAYOFWEEK
CEILING	LENGTH		HOUR
COS	REPEAT		MINUTE
COT	RIGHT		MONTH

jConnect 支持的函 数

数字函数	字符串函数	系统函数	时间/日期函数
DEGREES	SOUNDEX		MONTHNAME
EXP	SPACE		NOW
FLOOR	SUBSTRING		QUARTER
LOG	UCASE		SECOND
LOG10			TIMESTAMPADD
PI			TIMESTAMPDIFF
POWER			YEAR
RADIANS			
RAND			
ROUND			
SIGN			
SIN			
SQRT			
TAN			

JDBC-ODBC bridge 支持的函数

数字函数	字符串函数	系统函数	时间/日期函数
ABS	ASCII	IFNULL	CURDATE
ACOS	CHAR	USERNAME	CURTIME
ASIN	CONCAT		DAYNAME
ATAN	DIFFERENCE		DAYOFMONTH
ATAN2	INSERT		DAYOFWEEK
CEILING	LCASE		DAYOFYEAR
COS	LEFT		HOUR

数字函数	字符串函数	系统函数	时间/日期函数
СОТ	LENGTH		MINUTE
DEGREES	LOCATE		MONTH
EXP	LOCATE_2		MONTHNAME
FLOOR	LTRIM		NOW
LOG	REPEAT		QUARTER
LOG10	RIGHT		SECOND
MOD	RTRIM		WEEK
PI	SOUNDEX		YEAR
POWER	SPACE		
RADIANS	SUBSTRING		
RAND	UCASE		
ROUND			
SIGN			
SIN			
SQRT			
TAN			
TRUNCATE			

使用转义语法的语句应该可以用于 Adaptive Server Anywhere、Adaptive Server Enterprise、Oracle、 SQL Server 或从 Interactive SQL 连接到的其 它数据库管理系统。

例如,若要使用 SQL 转义语法获得具有 *sa_db_info* 过程的数据库属性,您将在 Interactive SQL 中的 "SQL 语句 " 窗格中键入以下内容:

{{CALL sa_db_info(1) }}

第8章

从数据库表中选择数据

关于本章 本章介绍从数据库中检索数据的查询。本章说明如何从单个表中检索数 据。

有关从多个相关表中选择数据的信息,请参阅 "从多个表中选择数据 " 第 123 页。此外, "选择集合数据 " 第 135 页说明如何对您的数据进行 分组以及对一个或多个列中的数据执行计算。

介绍

应用程序(客户端)和数据库服务器之间的所有交互操作都是通过将 SQL语句发送到数据库服务器(数据库服务器将信息返回客户端)执 行的。

SELECT 语句从数据库中检索信息以供客户端应用程序使用。SELECT 语句也称为**查询**。信息以结果集的形式传送到客户端。然后,客户端应 用程序可以对结果集进行处理。例如,Interactive SQL 在 "结果 " 窗格 中显示结果集。结果集由一系列的行组成,就像数据库中的表一样。

SELECT 语句在应用程序中可以非常复杂,从多个表中检索非常具体的 信息。本章只使用简单的 SELECT 语句:更加高级的查询将在以后的教 程中进行介绍。

有关 SELECT 语句的完整语法,请参阅 "ASA SQL 参考手册 > SQL 语 句 > SELECT 语句 "。

关于本章

如果您要了解	那么请参阅
如何排序查询结果	" 查询结果排序 " 第 111 页
如何从表中选择列	"从表中选择列"第108页
如何从表中选择行	"从表中选择行"第115页
如何比较查询中的日期	"比较搜索条件中的日期 " 第 116 页
如何使用模式匹配	" 搜索条件中的模式匹配 " 第 118 页
如何通过声音匹配行	"通过发音匹配行"第119页
如何使用复合搜索条件	"使用复合搜索条件"第 119 页
到何处查找有关搜索条件快捷方式的 信息	"复合搜索条件的快捷方式"第 120页

注意 在阅读和练习本章中的示例时,您应该在计算机上运行 Adaptive Server Anywhere 软件。

每个示例都指导您将命令键入 Interactive SQL 并说明您将在计算机屏幕 上看到的结果。如果在阅读教程时不能运行该软件,您仍可以学习 SQL,但是将没有自己进行实践的机会。

这些示例假设您已经启动了 Interactive SQL 并连接到示例数据库。

有关说明,请参阅 "Adaptive Server Anywhere 快速入门 " 第1页。

选择完整的表

最简单的 SELECT 语句检索单个表中的所有数据。此 SELECT 语句的 语法如下:

SELECT * FROM table-name

其中, *table-name* 应该用您正在查询的表的名称进行替换。星号 (*) 是 所有列列表的简写形式。

- ♦ 列出公司销售的所有产品:
 - 在 Interactive SQL 中,在 "SQL 语句" 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT * FROM Product

SELECT 语句检索 product 表的所有行和列,并将它们显示在 "结果 " 窗格的 " 结果 " 选项卡上:

id	name	descrip- tion	size	color	quantity	unit_ price
300	Tee Shirt	Tank Top	Small	White	28	9
301	Tee Shirt	V-neck	Medium	Orange	54	14
302	Tee Shirt	Crew Neck	One size fits all	Black	75	14
400	Baseball Cap	Cotton Cap	One size fits all	Black	112	9

Product 表包含七列。每列都有一个名称,例如 color 或 id。公司 销售的每种产品都有一行,每行在每列中都有单个值。例如, ID 为 301 的产品是 T 恤衫。它是中号 V 型领的,颜色为橙色。

 表名称是不区分大小写的 表名称 Product 以大写字母 P 开头,即 使实际表名称都是小写字母。在其字符串比较中,可以将 Adaptive Server Anywhere 数据库创建为区分大小写的或不区分大 小写的,但在它们作为标识符(例如表名称和列名称)使用时总 是不区分大小写的。

> 有关创建数据库的信息,请参阅 "ASA SQL 用户指南 > 使用数据 库对象 > 创建数据库 " 或 "ASA 数据库管理指南 > 数据库管理实 用程序 > Initialization 实用程序 "。

- SQL 关键字是不区分大小写的 您可以输入 select 或 Select 代替 SELECT。在文档中, SQL 关键字通常使用大写字母。
- 换行符不重要您可以将语句都键入在一行上,或者通过在每行的末尾按ENTER键将它们分开。某些SQL语句(例如SELECT语句)由多个部分组成,这些组成部分称为子句。在许多示例中,为了便于阅读,将每个子句单独放在一行上,因此示例中的语句在文档中通常以下面的形式书写:

SELECT * FROM product

 结果集中行顺序没有意义 无法保证行以什么顺序从数据库返回, 该顺序也没有意义。如果您要以特定的顺序检索行,则必须在查 询中指定顺序。

有关更多信息,请参阅 " 查询结果排序 " 第 111 页。

练习 尝试查询示例数据库中的其它表,例如 employee、 customer、 contact 或 sales_order 表。

从表中选择列

您可以通过紧接在 SELECT 关键字之后列出所需的列来限制 SELECT 语句检索的列。此 SELECT 语句的语法如下:

SELECT column-name-1, column-name-2,... FROM table-name

其中 column-name-1、 column-name-2 和 table-name 应该用您正在查询 所需的列和表的名称讲行替换。

结果集列的列表称为选择列表。它是由逗号分隔的。列表中最后一列之 后没有逗号,如果列表中只有一列,也不使用逗号。以这种方式限制列 通常称为投影。

◆ 列出每种产品的名称、说明和颜色:

. 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT name, description, color FROM product

name	description	color
Tee Shirt	Tank Top	White
Tee Shirt	V-neck	Orange
Tee Shirt	Crew Neck	Black
Baseball Cap	Cotton Cap	Black

对列进行重新排列 列按照您在 SELECT 语句中键入的顺序进行显示。如果您要对列进行重 新排列,只要在语句中更改列名称的顺序即可。例如,若要将 description 列放在左边, 请使用以下语句:

> SELECT description, name, color FROM product

使用计算列

结果集中的列不一定仅为表中的列。它们还可以是从基础数据计算的表 达式。您可以将多个表列组合成单个结果集列,还可以使用多种函数和 运算符控制得到的结果。

♦ 列出每种产品的库存值:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT id, (unit_price * quantity) as value
FROM product

id	value
300	252
301	756
302	1050
400	1008

注意

- 可以赋予列一个别名 缺省情况下,列名称是选择列表中列出的 表达式,但是对于计算列,该表达式十分麻烦,也不能提供足够 的信息。这里在选择列表中将计算列重命名为 value。value 是该列 的别名。
- 可以使用其它运算符 在上面的示例中,使用乘法运算符将多个 列组合在一起。您可以使用其它运算符,包括标准的算数运算符 以及逻辑运算符和字符串运算符。

例如,以下查询列出所有客户的全名:

SELECT id, (fname || ' ' || lname) as "Full name" FROM customer

||运算符连接字符串。在本查询中,列的别名具有空格,因此必须 用双引号括起来。此规则不仅适用于列别名,还适用于数据库中 的表名称和其它标识符。 有关运算符的完整列表,请参阅 "ASA SQL 参考手册 > SQL 语言 元素 > 运算符 "。

• **可以使用函数** 除了组合列之外,您可以使用多种内置函数生成 所需的结果。

例如,以下查询以大写字母列出产品名称:

SELECT id, UCASE(name) FROM product

id	UCASE(product.name)
300	TEE SHIRT
301	TEE SHIRT
302	TEE SHIRT
400	BASEBALL CAP

有关函数的完整列表,请参阅 "ASA SQL 参考手册 > SQL 函数 > 按字母顺序排列的函数列表 "。

查询结果排序

除非另有要求,否则数据库服务器返回表中行的顺序没有任何意义。通 常,以一种更有意义的顺序查看表中的行将十分有用。例如,您可能要 按字母顺序查看产品。

您可以通过 SELECT 语句末尾的 ORDER BY 子句对结果集中的行进行 排序。此 SELECT 语句的语法如下:

SELECT column-name-1, column-name-2,... FROM table-name ORDER BY order-by-column-name

其中 column-name-1、 column-name-2 和 table-name 应该用您正在查询 所需的列和表的名称进行替换,而 order-by-column-name 是该表中的一 个列。像以前一样,您可以使用星号作为表中所有列的简写形式。

◆ 按字母顺序列出产品:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容:

SELECT id, name, description FROM product ORDER BY name

id	name	description
400	Baseball Cap	Cotton Cap
401	Baseball Cap	Wool cap
700	Shorts	Cotton Shorts
600	Sweatshirt	Hooded Sweatshirt

注意• **子句的顺序有意义** ORDER BY 子句必须在 FROM 子句和 SELECT 子句之后。 • **您可以指定升序或降序** 缺省顺序是升序。您可以通过在子句的 末尾添加关键字 DESC 指定降序,如在下面的查询中:

> SELECT id, quantity FROM product ORDER BY quantity DESC

id	quantity
400	112
700	80
302	75
301	54
600	39

 您可以根据多个列进行排序 以下查询先按大小 (按字母顺 序)排序,然后按名称进行排序;

> SELECT id, name, size FROM product ORDER BY size, name

id	name	size
600	Sweatshirt	Large
601	Sweatshirt	Large
700	Shorts	Medium
301	Tee Shirt	Medium

• **ORDER BY 列不一定在选择列表中** 以下查询通过单价对产品进行排序,即使价格不包括在结果集中

SELECT id, name, size FROM product ORDER BY unit price

id	name	size
500	Visor	One size fits all
501	Visor	One size fits all
300	Tee Shirt	Small
400	Baseball Cap	One size fits all

如果您不使用 ORDER BY 子句并且多次执行查询,您可能会得到 不同的结果 这是因为 Adaptive Server Anywhere 可能以不同的顺 序返回相同的结果集。当缺少 ORDER BY 子句时, Adaptive Server Anywhere 以最有效的顺序返回行。这意味着结果集的外观 可能随您上一次访问行的时间和其它因素的不同而有所变化。确 保以特定的顺序返回行的唯一方法是使用 ORDER BY。

使用索引提高 ORDER BY 的性能

有时候, Adaptive Server Anywhere 数据库服务器执行具有 ORDER BY 子句的查询有多种可能的方式。您可以使用索引使数据库服务器更有效 地对表进行搜索。

具有 WHERE 和可以以多种可能方式执行的查询的示例是同时具有 WHERE 子句和ORDER BY 子句的ORDER BY 子句的查询。

查询

SELECT * FROM customer WHERE id > 300 ORDER BY company_name 在本示例中, Adaptive Server Anywhere 必须选择以下两种策略之一进行查询:

- 1. 按照公司名称的顺序检查整个客户表, 查看每行的客户 ID 是否大于 300。
- 2. 使用 id 列上的键只读取 id 大于 300 的公司, 然后需要按照公司名称 对结果进行排序。

如果只有很少的 id 值大于 300,则第二种策略较好,这是因为只对很少的行进行扫描并迅速进行排序。如果大多数 id 值都大于 300,则第一种 策略较好,这是因为不需要进行排序。

解决问题 在 id 和 company_name 上创建一个两列索引就可以解决上面示例中的问题。Adaptive Server Anywhere 可以使用此索引从表中以正确的顺序选择行。但是,请记住,索引占用数据库文件中的空间,保持索引为最新的还需要一些系统开销。不要无限制地创建索引。

有关更多信息,请参阅 "ASA SQL 用户指南 > 监控和提高性能 > 使用索引 "。

从表中选择行

您可以通过将 WHERE 子句添加到查询来限制 SELECT 语句检索的行。 有时候这称为对结果集应用限制。WHERE 子句包含一个指定要返回行 的**搜索条件**。此 SELECT 语句的语法如下:

SELECT column-name-1, column-name-2,... FROM table-name WHERE search-condition

像以前一样,其中的 column-name-1、 column-name-2 和 table-name 应 该用您正在查询所需的列和表的名称进行替换。下面将对 search-condition 进行更详细的介绍。如果您使用 ORDER BY 子句,则它必须 在 WHERE 子句之后。

♦ 列出所有黑色的产品:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容:

```
SELECT *
FROM product
WHERE color = 'black'
```

id	name	description	size	color	quanti ty	unit_ price
302	Tee Shirt	Crew Neck	One size fits all	Black	75	14
400	Baseball Cap	Cotton Cap	One size fits all	Black	112	9
501	Visor	Plastic Visor	One size fits all	Black	28	7

注意

WHERE 子句包含一个选择行的搜索条件。在本例中,该搜索条件是 color = 'black'。其它搜索条件在以下几节中介绍。

有关搜索条件的信息,请参阅 "ASA SQL 参考手册 > SQL 语言元素 > 搜索条件 "。

需要在 black 两边加单引号。它们指示 black 是一个字符串。双引 ٠ 号具有不同的意义。对于列名称和其它标识符,可以使用双引号 使原本无效的字符串有效。

有关字符串的信息,请参阅 "ASA SQL 参考手册 > SQL 语言元素 >字符串"。

- 示例数据库是不区分大小写的,因此不管您搜索 BLACK、black 还是 Black, 都将得到相同的结果。
- 如何排列子句的顺序是有意义的。 SELECT 列表后面是 FROM 子 • 句,再后面是 WHERE 子句,然后是 ORDER BY 子句。以不同的 顺序键入子句将出现语法错误。

尝试使用一些将您在本章中所学内容结合起来的查询。这里有一个查 询,该查询列出所有名为 John 的职员的姓名和出生日期。

> SELECT (emp fname || ' ' || emp lname) as Name, birth date FROM employee WHERE emp fname = 'John' ORDER BY birth date

Name	birth_date
John Letiecq	4/27/1939
John Sheffield	9/25/1955

比较搜索条件中的日期

您可以使用等式以外的运算符选择一组满足搜索条件的行。不等式运算 符(<、>)可用于比较数字、日期、甚至是字符串。

练习

◇ 列出所有生于 1964 年 3 月 13 日以前的职员:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容:

```
SELECT emp_lname, birth_date
FROM employee
WHERE birth_date < 'March 13, 1964'
ORDER BY birth_date DESC</pre>
```

emp_Iname	birth_date
Ahmed	12/12/1963
Dill	7/19/1963
Rebeiro	4/12/1963
Garcia	1/23/1963
Pastor	7/14/1962

注意

- **自动转换为日期** Adaptive Server Anywhere 数据库服务器知道 *birth_date* 列包含日期,自动将字符串 'March 13, 1964' 转换为日期。
- **指定日期的方式** 指定日期有多种方式。以下方式都是 Adaptive Server Anywhere 所接受的:

```
'March 13, 1964'
'1964/03/13'
'1964-03-13'
```

您可以通过设置数据库选项调整对查询中日期的解释。yyyy/mm/ dd 或 yyyy-mm-dd 格式的日期总会被清楚地识别为日期,与 DATE_ORDER 设置无关。

有关控制查询中容许的日期格式的信息,请参阅 "ASA 数据库管理指南>数据库选项>DATE_ORDER 选项 [compatibility]"和 "ASA 数据库管理指南>数据库选项>设置选项 "。

• 其它比较运算符 有关可用比较运算符的完整列表,请参阅 "ASA SQL 参考手册 > SQL 语言元素 > 比较运算符 "。

搜索条件中的模式匹配

模式匹配是一种识别字符数据的通用方式。在 SQL 中, LIKE 关键字用于搜索模式。模式匹配使用通配符字符来匹配不同的字符组合。

✤ 列出姓以 BR 开头的所有职员:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容:

```
SELECT emp_lname, emp_fname
FROM employee
WHERE emp lname LIKE 'br%'
```

emp_Iname	emp_fname
Breault	Robert
Braun	Jane

搜索条件中的 % 表示任何数目的其它字符可以接在字母 BR 之 后。

- ◇ 列出姓符合以下条件的所有职员,姓以 BR 开头,其后没有 字母或有多个字母,然后是一个 T, T 后面没有字母或有多 个字母:
 - 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容:

SELECT emp_lname, emp_fname FROM employee WHERE emp lname LIKE 'BR%T%'

emp_Iname	emp_fname
Breault	Robert

第一个%符号与字符串 eaul 匹配,而第二个%符号与空字符 串(没有字符)匹配。

另一个可以与 LIKE 一起使用的特殊字符是_(下划线)字符,下划线

只与一个字符匹配。例如,模式 'BR_U%' 与所有以 BR 开头并以 U 作为 第四个字母的名称匹配。在 Braun 中, _字符与字母 A 匹配, % 与 N 匹 配。

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 语言元素 > LIKE 条件 "。

通过发音匹配行

使用 SOUNDEX 函数,您可以通过发音匹配行。例如,假设有一个电话留言找姓名发音类似 "Ms. Brown" 的人。公司中哪些职员的姓名发音 与 Brown 类似?

✤ 列出姓发音与 Brown 类似的职员:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容:

```
SELECT emp_lname, emp_fname
FROM employee
WHERE SOUNDEX( emp lname ) = SOUNDEX( 'Brown' )
```

emp_Iname	emp_fname
Braun	Jane

SOUNDEX 使用的算法使该函数主要用于语言为英语的数据库。

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 函数 > SOUNDEX 函数 [字符串]"。

使用复合搜索条件

可以使用 AND 和 OR 逻辑运算符将搜索条件组合起来形成复合搜索条件。有时将该搜索条件的每个单独的部分称为一个**谓词**。

◆ 列出除了名为 Whitney 的职员之外所有生于 1964 年 3 月 13 日之前的职员:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容:

SELECT emp_lname, birth_date
FROM employee
WHERE birth_date < '1964-3-13'
AND emp_lname <> 'whitney'

emp_Iname	birth_date
Cobb	12/4/1960
Jordan	12/13/1951
Breault	5/13/1947
Espinoza	12/14/1939
Dill	7/19/1963
Francis	9/12/1954

复合搜索条件的快捷方式

使用简写形式SQL 具有两个复合搜索条件简写形式。第一个 (BETWEEN) 在您查找一BETWEEN系列值时使用。例如,以下两个查询是等同的:

SELECT emp_lname, birth_date FROM employee WHERE birth date BETWEEN '1963-1-1' AND '1965-3-31'

和

SELECT emp_lname, birth_date
FROM employee
WHERE birth_date >= '1963-1-1'
AND birth_date <= '1965-3-31'</pre>

使用简写形式 IN 第二个简写形式 (IN) 可以在您查找多个值中的一个时使用。以下两个 语句是等同的:

SELECT emp_lname, emp_id
FROM employee
WHERE emp_lname IN ('yeung','bucceri','charlton')

和

SELECT emp_lname, emp_id
FROM employee
WHERE emp_lname = 'yeung'
OR emp_lname = 'bucceri'
OR emp_lname = 'charlton'

第9章

从多个表中选择数据

关于本章 本章介绍在多个表中查看信息的数据库查询。为了完成此操作, SQL 提供了 JOIN 运算符。在查询中将各个表连接在一起有多种不同的方式,本章介绍其中最重要的几种。

介绍

有时,需要从多个相关的表中查看数据。本章说明如何使用 join 以一种 有用且有意义的方式查看数据。

关于本章

如果您要了解	那么请参阅
如何显示表的列表	"显示表的列表"第125页
如何显示两个表中数据的所有组合	"使用矢量积连接表"第126页
如何使连接起作用	"使用 ON 短语限制连接 "第 127 页
如何连接两个具有相同名称的连接外 键的表	"使用键连接 (key join) 来连接 表 " 第 129 页
如何通过具有相同名称的列连接表	"使用自然连接 (natural join) 来 连接表 " 第 131 页

显示表的列表

在 Interactive SQL 中,您可以通过按 F7 键显示表的列表。

查寻表名	×
1. 请键入您要查找的表的前几个字符(Y)	
2. 通过选择表的类型来缩小搜索范围	
▼表(T) □ 系统表(S) □ 视图(V)	
3. 单击所需表,然后单击"确定"或"显示列"(L)	
contact	
customer	
department	
employee	
fin_code	
fin_data	
product	_
sales_order	-
□ 亚示所有者名称(U)	
	T001 0 0
显示列(C) 備定 取消	帮助(日)

突出显示一个表并单击 "显示列 " 来查看该表的列。按 Escape 键可返回 到表列表,再次按该键将返回到 "SQL 语句 " 窗格。如果不按 Escape 键 而是按 Enter 键,则会将突出显示的表或列名称复制到 "SQL 语句 " 窗格中光标当前所在的位置。

按 Escape 键可以离开该列表。

有关示例数据库中表的图示,请参阅 "示例数据库 "第48页。

使用矢量积连接表

示例数据库中有一个表是 sales_order,该表列出了该公司收到的订单。 每份订单都具有一个 sales_rep 列,其中包含负责该订单的销售代表的职员 ID。 sales_order 表中有 648 行, employee 表中有 75 行。

矢量积连接是了解连接的一个简单的起点,但其本身并不是很有用。

- ♦ 列出 employee 和 sales_order 表中的所有数据:
 - 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT * FROM sales_order CROSS JOIN employee

此查询的结果是将 employee 表中的每行与 sales_order 表中的每行 匹配在一起,最终的结果显示在 Interactive SQL"结果 "窗格中 的"结果 "选项卡上。因为 employee 表中有 75 行,而 sales_order 表中有 648 行,所以连接的结果中有 75 (648 = 48,600 行。每行 的前半部分是 sales_order 表中的所有列,而后半部分是 employee 表中的所有列。此连接称为完全矢量积。

下面的章节将介绍如何构造更有选择性的连接。更有选择性的连接可以 看做是对矢量积表应用了限制。

有关更多信息,请参阅 "ASA SQL 用户指南 > 连接:从多个表检索数据 > 交叉连接 "。

使用 ON 短语限制连接

ON 短语对连接中的行应用限制,其方式在很大程度上与 WHERE 子句 对查询中的行应用限制相同。

ON 短语可构建比 CROSS JOIN 构建的连接更为有用的的连接。例如, 您可以对 sales_order 和 employee 表的连接应用 ON 短语以便只检索特定的行,从而使得在检索出来的每行中, sales_order 表中的 sales_rep 与 employee 表中的 sales_rep 相同。这样,每行都包含有关订单和负责该订单的销售代表的信息。

◇ 列出所有销售订单,并列出每份订单的日期和负责职员:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

```
SELECT employee.emp_lname,
            sales_order.id,
            sales_order.order_date
FROM sales_order JOIN employee
            ON sales_order.sales_rep = employee.emp_id
```

emp_Iname	id	order_date
Chin	2008	4/2/2001
Chin	2020	3/4/2001
Chin	2032	7/5/2001
Chin	2044	7/15/2000
Chin	2056	4/15/2001

说明

用表名称作前缀来标识列。使用表名称前缀可以使语句清晰明了,当两个表中存在具有相同名称的列时,必须使用表名称前缀。

此查询的结果仅包含 648 行 (每行对应于 sales_order 表中的一行)。在 矢量积的 48,600 行中,只有其中的 648 行在两个表中具有相同的职员编 号。

结果的排序没有任何意义。您可以添加一个 ORDER BY 子句对查询强 制特定的顺序。

ON 子句包含没有在最终结果集中显示的列。

有关更多信息,请参阅 "ASA SQL 用户指南 > 连接:从多个表检索数据 > 显式连接条件 (ON 短语) "。

使用键连接 (key join) 来连接表

许多常见的连接建立在两个由外键关联的表之间。最常见的连接限制外 键值与主键值相等。

KEY JOIN 运算符根据外键关系连接两个表。换句话说, Adaptive Server Anywhere 生成一个 ON 子句,该子句使一个表的主键列与另一个 表的外键列相等。

上一节中的示例限制 sales_order 表中的外键值等于 employee 表中的主键 值。

SELECT employee.emp_lname, sales_order.id, sales_order.order_date FROM sales_order JOIN employee ON sales_order.sales_rep = employee.emp_id

可以使用 KEY JOIN 更简单地表达该查询:

KEY JOIN 正是键入 ON 子句的快捷方式;这两个查询是相同的。当您 指定了 JOIN 但未指定 CROSS、NATURAL、KEY 或使用 ON 子句时, 缺省使用键连接。

如果您查看职员数据库的图示,表之间的连线表示外键。您可以在图示中两个表通过连线连接的任何地方使用 KEY JOIN 运算符。

有关示例数据库的图示,请参阅 "示例数据库 "第48页。

连接两个以上的表 可以使用连接运算符连接两个或两个以上的表。下面的查询使用四个表 列出每个客户下的订单的总值。它使用 customer、 sales_order、 sales_order_items 和 product 四个表中每对表之间的单一外键关系连接 这四个表。

◆ 列出各个公司以及其订单的总值:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

company_name	value
Bensoul's Boutique	1332
Bush Pro Shop	2940
Sterling & Co.	6804
Ocean Sports	3744

说明

结果集显示的顺序可能会有所不同。结果集中行的顺序没有任何意义。

该示例使用了 SUM 运算符,该运算符是一个集合函数。集合函数与 GROUP BY 子句一起使用,为每个行组返回值。在本例中,将为每个 company_name 计算 sales_order_items.quantity * product.unit_price 的总和 (即为每个产品类型所付的总金额),从而返回每个公司的销售额。

FROM 子句中的圆括号有助于阐明建立连接的顺序。

有关集合函数的更多信息,请参阅 "集合函数简介 "第137页。

有关复杂键连接的更多信息,请参阅 "ASA SQL 用户指南 > 连接:从多 个表检索数据 > 键连接 "。
使用自然连接 (natural join) 来连接表

NATURAL JOIN 运算符根据公共列名称连接两个表。换句话说, Adaptive Server Anywhere 生成一个 ON 子句,该子句使每个表的公共列 相等。

✤ 列出所有职员及他们所在的部门:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT emp_lname, dept_name FROM employee NATURAL JOIN department

emp_Iname	dept_name
Whitney	R & D
Cobb	R & D
Breault	R & D
Shishov	R & D
Driscoll	R & D

Adaptive Server Anywhere 查看这两个表,并确定它们唯一共有的列名称 是 *dept_id*。将在内部生成以下 ON 子句并用于执行连接:

FROM employee JOIN department
 ON employee.dept_id = department.dept_id

NATURAL JOIN 正是键入 ON 子句的快捷方式;这两个查询是相同的。

使用 NATURAL JOIN 时出错

此连接运算符可能会使您不想等同的列相等,从而引起问题。例如,以 下查询会生成不想要的结果:

SELECT * FROM sales_order NATURAL JOIN customer

此查询的结果没有任何行。 Adaptive Server Anywhere 在内部生成以下 ON 子句:

FROM sales_order JOIN customer
 ON sales_order.id = customer.id

sales_order 表中的 *id* 列是*订单* 的 **ID** 编号。*customer* 表中的 *id* 列是*客户* 的 **ID** 编号。没有一个 **ID** 编号匹配。当然,即使找到了匹配的编号,也没有任何意义。

有关更多信息,请参阅 "ASA SQL 用户指南 > 连接:从多个表检索数据 > 自然连接 "。

使用外连接来连接表

在前面的示例中,对于您创建的连接,只有在满足连接条件后才会返回 一些行。这些连接称为**内连接**,它们是缺省连接。有时,您可能想保留 一个表中的所有行。若要实现此目的,您可以使用**外连接**。

您可以指定一个**右外连接**(该连接保留右边的表中的所有行)、左**外连** 接(该连接保留左边的表中的所有行)或**完全外连接**(该连接保留两 个表中的所有行)。

◆ 列出所有客户以及他们所下的所有订单的日期:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT lname, order_date, city
FROM customer KEY LEFT OUTER JOIN sales_order
WHERE customer.state = 'NY'
ORDER BY order date

Iname	order_date	city
Thompson	(空值)	Bancroft
Reiser	1993-01-22	Rockwood
Clarke	1993-01-27	Rockwood
Mentary	1993-01-30	Rockland

该语句包括所有客户,不管他们是否已下订单。如果某个特定的客户未 下任何订单,则结果中对应于订单信息的每一列都包含空值。

有关更多信息,请参阅 "ASA SQL 用户指南 > 连接:从多个表检索数据 > 外连接 "。

^{第10章} 选择集合数据

关于本章 本章介绍如何构造用于获得集合信息的查询。下面是集合信息的示例:

- 一列中所有值的总和。
- 一列中不同条目的数量。
- 一列中各条目的平均值。

汇总数据

有些查询检查表中数据反映行集(而不是单个行)属性的那些方面。 例如,您可能要计算客户为某一订单支付的平均款额,或者需要了解有 多少职员为各个部门工作。对于这些类型的任务,您可以使用**集合**函数 和 GROUP BY 子句。

关于本章

如果想知道	请参阅
如何查看有关整个表的摘要信息	" 集合函数简介 " 第 137 页
如何查看有关表中各个组的摘要信息	"将集合函数应用于分组数据 " 第 138 页
如何查看有关表中有限组集合的摘要 数据	"限制组"第140页

集合函数简介

集合函数为一组行返回一个值。如果没有 GROUP BY 子句,集合函数 将为满足查询其它方面的所有行返回一个值。

◆ 列出公司的职员数:

 在 Interactive SQL 中,将以下语句键入 "SQL 语句" 窗格,然后 按 F5 执行该语句。

SELECT COUNT(*) FROM employee

COUNT(*)

75

结果集仅包含一个列(标题为 COUNT(*))和一个行(包含职员的总人数)。

✤ 列出公司的职员数以及年级最大和最小的职员的出生日期:

• 在 Interactive SQL 中,将以下语句键入 "SQL 语句 " 窗格, 然后 按 F5 执行该语句。

SELECT COUNT(*), MIN(birth_date), MAX(birth_date)
FROM employee

COUNT(*)	MIN(employee.birth_date)	MAX(employee.birth_date)
75	1/2/1936	1/18/1973

COUNT、MIN 和 MAX 函数称作**集合函数**。其中的每个函数都会将信息汇总。集合函数一共有六种:MIN、MAX、COUNT、AVG、SUM和LIST。除 COUNT之外的所有函数都以列的名称作为参数。您可以看到,COUNT 以星号作为其参数。

有关更多信息,请参阅 "ASA SQL 参考手册 > SQL 函数 > 集合函数 "。

将集合函数应用于分组数据

除了提供有关整个表的信息之外,集合函数还可应用于行组。GROUP BY子句将行分组,而集合函数将为每一行组返回一个值。

示例

✤ 列出销售代表和每位销售代表获得的订单数:

 在 Interactive SQL 中,将以下语句键入 "SQL 语句" 窗格,然后 按 F5 执行该语句。

```
SELECT sales_rep, count( * )
FROM sales_order
GROUP BY sales_rep
ORDER BY sales_rep
```

sales_rep	count(*)
129	57
195	50
299	114
467	56

.

GROUP BY 子句指示 Adaptive Server Anywhere 将以其它方式返回的所 有行的集合进行分区。每个分区或组中的所有行在命名的列中都具有相 同的值。每个唯一值或值集仅对应于一个组。在本例中,每个组中的所 有行都具有相同的 sales_rep 值。

诸如 COUNT 等集合函数应用于每个组中的行。因此,此结果集会显示 每个组中行的总数。查询的结果由对应于每个销售代表 ID 号的行组 成。每行都包含销售代表 ID 以及该销售代表获得的销售订单总数。

每当使用 GROUP BY 时,结果表中都有一行对应于在 GROUP BY 子句 中命名的每个列或列集。

有关更多信息,请参阅 "ASA SQL 用户指南 > 对查询结果进行汇总、分 组和排序 > GROUP BY 子句: 将查询结果划分为组 "。

GROUP BY 常见错 GROUP BY 的一个常见错误是试图获取无法正确放入一个组中的信息。 例如,

> -- This query is incorrect SELECT sales rep, emp lname, COUNT(*) FROM sales_order KEY JOIN employee GROUP BY sales rep

将报告以下错误:

误

对选择列表中 'emp_Iname' 的函数或列引用还必须在 GROUP BY 中出现

之所以会报告错误,是因为 Adaptive Server Anywhere 无法确定对于给 定 ID 的职员,每个结果行是否都具有相同的姓氏。

若要解决此问题, 请向 GROUP BY 子句添加以下列。

SELECT sales rep, emp lname, COUNT(*) FROM sales order KEY JOIN employee GROUP BY sales_rep, emp_lname ORDER BY sales rep

如果不适用,则可以使用集合函数来仅选择一个值,如下所示:

SELECT sales_rep, MAX(emp_lname), COUNT(*) FROM sales_order KEY JOIN employee GROUP BY sales rep ORDER BY sales rep

MAX 函数选择每个组的详细信息行中的姓氏最大值 (按字母顺序排在 最后的姓氏)。由于只能有一个不同的最大值,所以该语句有效。在本 例中,相同的姓氏会出现在一个组中的每个详细信息行上。

限制组

您已经知道了如何使用 WHERE 子句来限制结果集中的行。您可以使用 HAVING 子句来限制组中的行。

✤ 列出其订单数超过 55 的所有销售代表:

 在 Interactive SQL 中,将以下语句键入 "SQL 语句" 窗格,然后 按 F5 执行该语句。

SELECT sales_rep, count(*) as orders
FROM sales_order KEY JOIN employee
GROUP BY sales_rep
HAVING count(*) > 55
ORDER BY orders DESC

sales_rep	orders
299	114
129	57
1142	57
467	56

子句顺序

GROUP BY 必须始终出现在 HAVING 子句之前。如果 WHERE 子句和 GROUP BY 子句都存在,则 WHERE 子句必须出现在 GROUP BY 子句 之前。

HAVING 子句和 WHERE 子句可以在一个查询中同时使用。只有在构造 了组之后, HAVING 子句中的条件才会在逻辑上限制结果行。WHERE 子句中的标准在构造组之前在逻辑上取值,从而节约了时间。

有关更多信息,请参阅 "ASA SQL 用户指南 > 对查询结果进行汇总、分 组和排序 > HAVING 子句:选择数据组 "。

组合 WHERE 和 HAVING 子句

有时,您可以使用 WHERE 子句或 HAVING 子句指定同一个行集。在这种情况下,其中一种方法并不比另一种方法更有效。优化程序始终会自动分析您键入的每条语句,并选择执行该语句的有效方法。所使用的语法最好对所需的结果进行了最为清晰的描述。通常,这意味着在前面的子句中消除不需要的行。

示例 若要列出其订单数超过 55 且 ID 大于 1000 的所有销售代表,请键入以下语句。

SELECT sales_rep, count(*)
FROM sales_order KEY JOIN employee
WHERE sales_rep > 1000
GROUP BY sales_rep
HAVING count(*) > 55
ORDER BY sales_rep

以下语句会产生相同的结果。

SELECT sales_rep, count(*)
FROM sales_order KEY JOIN employee
GROUP BY sales_rep
HAVING count(*) > 55 AND sales_rep > 1000
ORDER BY sales_rep

Adaptive Server Anywhere 会检测到这两条语句都描述了相同的结果集, 从而高效地执行各条语句。

第 11 章

使用子查询选择数据

关于本章 本章说明如何使用一个查询的结果作为另一个 SELECT 语句的一部分。 这在建立更复杂和信息量更大的查询时是一种有用的工具。

子查询介绍

关系数据库允许您在多个表中存储相关数据。"从多个表中选择数据" 第 123 页一章简要介绍了从相关表中提取数据的一种方法。另一种方法 涉及到**子查询**,子查询是出现在另一个查询的 WHERE 子句或 HAVING 子句中的查询。子查询使某些查询比连接更易于编写,并且某些查询的 编写必须使用子查询。

关于本章

如果您要了解	那么请参阅
何时使用子查询	"介绍"第145页
如何比较列值和子查询返回的单个值	"单行和多行子查询"第148页
如何使用子查询代替连接	"使用子查询代替连接"第 150 页

介绍

子查询使用一个查询的结果作为另一个查询的一部分。本节说明一种建立查询中可以使用子查询的情况,该查询列出库存不足产品的订单项。

生成该列表涉及两个查询。本节先分别介绍这两个查询,然后说明产生 相同结果的单个查询。

♦ 列出库存少于 20 件的所有产品:

 在 Interactive SQL 中,在 "SQL 语句" 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT id, description, quantity FROM product WHERE quantity < 20

id	description	quantity
401	Wool cap	12

该查询显示只有羊毛帽库存不足。

♦ 列出羊毛帽的所有订单项:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

```
SELECT *
FROM sales_order_items
WHERE prod_id = 401
ORDER BY ship_date DESC
```

id	line_id	prod_id	quantity	ship_date
2082	1	401	48	7/9/2001
2053	1	401	60	6/30/2001
2125	2	401	36	6/28/2001

id	line_id	prod_id	quantity	ship_date
2027	1	401	12	6/17/2001

可以使用子查询将这个确定库存不足项和确定这些项订单的两步过程组合成单个查询。

♦ 列出库存不足项的所有订单项:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下内容,然 后按 F5 键执行该语句。

SELECT *
FROM sales_order_items
WHERE prod_id IN
 (SELECT id
 FROM product
 WHERE quantity < 20)
ORDER BY ship_date DESC</pre>

id	line_id	prod_id	quantity	ship_date
2082	1	401	48	7/9/2001
2053	1	401	60	6/30/2001
2125	2	401	36	6/28/2001
2027	1	401	12	6/17/2001

该语句中的子查询是括在圆括号中的短语:

(SELECT id
 FROM product
 WHERE quantity < 20)</pre>

子查询生成一个 product 表内 id 列中所有值的列表,符合 WHERE 子句 搜索条件。

子查询返回一组行,但只返回单个列。IN关键字将每个值作为集合的 一个成员对待并测试主查询的每个行是否是该集合的成员。

单行和多行子查询

子查询可以返回的行数和列数具有约束。如果您使用 IN、 ANY 或 ALL,则子查询可以返回多个行,但只返回一个列。如果您使用其它运算符,则子查询必须返回单个值。

多行子查询 示例数据库中的两个表与财务结果有关。fin_code 表是一个较小的表, 保存财务数据及其含义的不同代码:

若要列出 fin_data 表中的收入项,请键入以下语句:

SELECT *
FROM fin_data
WHERE fin_data.code IN
 (SELECT fin_code.code
 FROM fin_code
 WHERE type = 'revenue')

year	quarter	code	amount
1999	Q1	r1	1023
1999	Q2	r1	2033
1999	Q3	r1	2998
1999	Q4	r1	3014
2000	Q1	r1	3114

本示例使用限定符清楚地标识每个引用中的 code 列所属的表。在这个特殊示例中,限定符可能已被省略。

另外两个关键字(ANY 和 ALL)可以用作运算符的限定符以允许它们 处理多个行。

以下查询与上面的成功查询相同:

```
SELECT *
FROM fin_data
WHERE fin_data.code = ANY
( SELECT fin_code.code
    FROM fin_code
    WHERE type = 'revenue' )
```

=ANY 条件与 IN 条件相同,但 ANY 还可以和不等式(如 < 或 >)一起使用,从而更灵活地使用子查询。

ALL 关键字与单词 ANY 相似。例如,以下查询列出非收入财务数据:

SELECT *
FROM fin_data
WHERE fin_data.code <> ALL
 (SELECT fin_code.code
 FROM fin_code
 WHERE type = 'revenue')

这与下面使用 NOT IN 的命令等同:

SELECT *
FROM fin_data
WHERE fin_data.code NOT IN
 (SELECT fin_code.code
 FROM fin_code
 WHERE type = 'revenue')

使用子查询的常见错通常,子查询结果集限制为单个列。以下示例没有意义,这是因为误Adaptive Server Anywhere 不知道将 fin_code 中的哪个列与 fin_data.code 列
比较。

-- this query is incorrect
SELECT *
FROM fin_data
WHERE fin_data.code IN
 (SELECT fin_code.code, fin_code.type
 FROM fin_code
 WHERE type = 'revenue')

单行子查询 使用 IN 条件的子查询可以返回一组行,而使用比较运算符的子查询则 只能返回一行。例如,由于子查询返回两行,所以以下命令导致一个错误:

> -- this query is incorrect SELECT * FROM fin_data WHERE fin_data.code = (SELECT fin_code.code FROM fin_code WHERE type = 'revenue')

使用子查询代替连接

假设您需要一个按时间顺序排列的订单和下达这些订单的公司的列表, 需要列出公司名称而不是它们的客户 ID,您可以使用如下连接获得此 结果:

使用连接 若要列出从 2001 年初以后每个订单的订单 ID、日期和公司名称,请键 入以下语句:

```
SELECT sales_order.id,
        sales_order.order_date,
        customer.company_name
FROM sales_order
        KEY JOIN customer
WHERE order_date > '2001/01/01'
ORDER BY order_date
```

id	order_date	company_name
2473	1/4/2001	Peachtree Active Wear
2474	1/4/2001	Sampson & Sons
2106	1/5/2001	Salt & Pepper's
2475	1/5/2001	Cinnamon Rainbow's
2036	1/5/2001	Hermanns

有关连接的更多信息,请参阅 "从多个表中选择数据 " 第 123 页。

使用子查询

以下语句使用子查询代替连接得到相同的结果:

```
SELECT sales_order.id,
    sales_order.order_date,
    ( SELECT company_name FROM customer
    WHERE customer.id = sales_order.cust_id )
FROM sales_order
WHERE order_date > '2001/01/01'
ORDER BY order date
```

即使 sales_order 表不是子查询的一部分,子查询也引用 sales_order 表的 cust_id 列。而 sales_order.cust_id 列在语句的主体部分引用 sales_order 表。 这称为**外部引用**。包含外部引用的任何子查询都称为**相关子查询**。 只要只需要其它表中的一个列时,就可以使用子查询代替连接。(调用 该子查询只返回一列。)在本例中,您只需要 company_name 列,因此可 以将连接更改为子查询。

如果子查询没有任何结果,则此方法称为外连接。教程中以前几节中的 连接的完整名称为内连接。

使用外连接 若要列出 Washington 州的所有客户以及他们最近的订单 ID,请键入以 下内容:

SELECT company_name, state, (SELECT MAX(id) FROM sales_order WHERE sales_order.cust_id = customer.id) FROM customer WHERE state = 'WA'

company_name	state	MAX(sales_order.id)
Custom Designs	WA	2547
It's a Hit!	WA	(空值)

lt's a Hit! 公司没有下任何订单,子查询为此客户返回空值。当使用内连接时,不会列出没有下订单的公司。

您还可以显式指定一个外连接。在本例中,还需要 GROUP BY 子句。

SELECT company_name, state, MAX(sales_order.id) FROM customer KEY LEFT OUTER JOIN sales_order WHERE state = 'WA' GROUP BY company_name, state

^{第12章} 更新数据库

关于本章 本章介绍如何对数据库的内容进行更改。本章介绍了如何添加行、删除 行和更改行的内容,并且介绍了如何使更改成为永久更改或纠正所做更 改。

简介

本章介绍如何对数据库中的数据进行更改。有三个基本操作:

- 插入 您可以将行添加到表中,以包括新数据。
- 删除 您可以删除表中各行以删除数据。
- 更新 您可以修改表中的现有行。

每一操作都是通过执行 SQL 语句执行的。

关于本章

如果您想了解	请查看
如何将行添加到表中	"将行添加到表中"第155页
如何更新表中各行	"修改表中各行"第156页
如何删除表中各行	" 删除行 " 第 157 页
如何执行事务	"将更改组合到事务中 " 第 158 页
Adaptive Server Anywhere 是如何检查 数据中的错误的	"完整性检查"第161页

将行添加到表中

假设公司决定销售新产品,褐色的丙烯酸纤维汗衫。该操作要求您将数据添加到示例数据库的 product 表中。

◆ 将褐色丙烯酸纤维汗衫添加到产品表:

• 在 Interactive SQL 中,在 "SQL 语句 " 窗格中键入以下语句,然 后按 F5 键执行该语句。

INSERT
INTO product (id, name, description, size, color,
quantity, unit_price)
VALUES (800, 'Vest', 'Acrylic Vest', 'Extra
Large', 'Brown', 42, 20.00)

如果您忘记指定这些列之一从而出现错误, Adaptive Server Anywhere 会报告错误。

您还可以在 Interactive SQL 中的结果集将新行添加到数据库表。

有关更多信息,请参阅 "在 Interactive SQL 中编辑表值 "第 81 页。

列中的空值 空值是用于指示某些项未知或不适用的特殊值。允许某些列包含空值, 而其它列则不包含空值。

修改表中各行

在大多数数据库中,您需要更新已存储于数据库中的记录。例如,当职员在部门间调动时,应更改经理 ID,同时也应更改部门 ID。

◆ 在 Interactive SQL 中将职员 #195 调到部门 400:

 在 Interactive SQL 中,在 "SQL 语句" 窗格中键入以下内容,然 后按 F5 键执行该语句。

```
UPDATE employee
SET dept_id = 400, manager_id = 1576
WHERE emp_id = 195
```

该语句同时为职员 Marc Dill (职员 ID 195)执行这两个更新。

因为 Adaptive Server Anywhere 更新满足 WHERE 子句条件的所有行, 所以有时要一次更新多行。例如,如果一组销售部门的职员被调到营销 部门,并且更新了他们的 *dept_id* 列,则以下语句为营销部门的所有职员 将 *manager_id* 设置为 **1576**。

UPDATE employee SET manager_id = 1576 WHERE dept_id = 400

对于已属于营销部门的职员,不进行任何更改。

您还可以根据 Interactive SQL 中的结果集修改行。

有关更多信息,请参阅 "在 Interactive SQL 中编辑表值 "第 81 页。

删除行

有时候您将要从表中删除行。假设 Rodrigo Guevara (职员 ID 249)将 离开公司。以下语句从 *employee* 表中删除 Rodrigo Guevara。

DELETE FROM employee WHERE emp_id = 249

使用 UPDATE 和 DELETE 语句,可以根据需要令搜索条件更为复杂。例如,如果 employee 表正被重新组织,则以下语句将从 employee 表中删除在 1989 年 3 月到 1990 年 3 月间雇佣的所有男性职员。

```
DELETE
FROM employee
WHERE sex = 'm'
AND start_date between '1988-03-03'
AND '1989-03-03'
```

因为您已对数据库做出了不想保留的更改,所以应该按以下所示撤消更改:

ROLLBACK

您还可以根据 Interactive SQL 结果集从数据库表中删除行。

有关更多信息,请参阅 "在 Interactive SQL 中编辑表值 "第 81 页。

将更改组合到事务中

Adaptive Server Anywhere 期望您会将命令组合到事务中。您提交事务以 使对数据库进行的更改成为永久更改。在更改数据时,您所做更改不会 立即成为永久更改。它们而是存储在**事务日志**中,并且在您输入 COMMIT 命令后成为永久更改。

通过了解哪些命令或动作意味着事务的开始或结束,您可以充分利用各 个事务。

开始事务 事务起始于以下事件之一:

- 在与数据库连接后执行的第一个语句。
- 在事务结束后执行的第一个语句。
- 完成事务 以下事件之一将完成任务:
 - 执行 COMMIT 语句使对数据库进行的更改成为永久更改。
 - 执行 ROLLBACK 语句撤消由事务执行的所有更改。
 - 执行具有自动提交副作用的语句:数据库定义命令(例如 ALTER、CREATE、COMMENT和DROP)都具有自动提交副作用。
 - 与数据库断开连接将执行隐式回退。

Interactive SQL 中 Interactive SQL 提供两个选项, 使您可以控制事务结束的时间和方式:

的选项

- 如果您将 AUTO_COMMIT 选项设置为 "ON",则 Interactive SQL 将自动在成功执行每一语句后提交您的结果,并且自动在每一失 败的语句后执行 ROLLBACK。
- COMMIT_ON_EXIT 选项的设置控制退出 Interactive SQL 时对未 提交的更改执行的操作。如果该选项设置为 "ON" (缺省设置), 则 Interactive SQL 执行 COMMIT 语句; 否则, 它会用 ROLLBACK 语句撤消未提交的更改。

如果您使用数据源

缺省情况下,ODBC 以自动提交方式工作。即使您已在 Interactive SQL 中将 AUTO_COMMIT 选项设置为 "OFF",ODBC 的设置也将替换 Interactive SQL 的设置。您可以使用 SQL_ATTR_AUTOCOMMIT 连接 属性更改 ODBC 的设置。ODBC 自动提交与 CHAINED 选项无关。

Adaptive Server Anywhere 也支持 Transact-SQL 命令 (例如 BEGIN TRANSACTION), 以和 Sybase Adaptive Server Enterprise 兼容。

有关更多信息,请参阅 "ASA SQL 用户指南 > Transact-SQL 兼容性 > Transact-SQL 支持概述 "。

使更改成为永久更改

COMMIT 语句使所有更改成为永久更改。

您应该在一组语句都有意义后使用 COMMIT 语句。例如,如果您要将 钱从一个客户的帐户转移到另一个帐户上,则应该将钱添加到一个客户 的帐户上,而从另一个客户的帐户上删除这笔钱,然后再进行提交;因 为在这种情况下,数据库中的钱多于或少于原来的数额都是没有意义 的。

您可以通过将 AUTO_COMMIT 选项设置为 "ON", 指示 Adaptive Server Anywhere 自动提交更改。这是 Interactive SQL 选项。在将 AUTO_COMMIT 设置为 "ON" 后, Adaptive Server Anywhere 必须在您 每次对它执行插入(insert 语句)、更新(update 语句)和删除 (delete 语句)后都更新数据库。这可能会显著降低性能。因此,最好 还是保持将 AUTO_COMMIT 选项设置为缺省值 "OFF"。

小心使用 COMMIT

尝试使用本教程中的示例时,请务必注意:在确定您要对数据库进行永 久更改前,不要使用 COMMIT 语句提交任何更改。

有关 Interactive SQL 选项的更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > Interactive SQL 选项 "。

取消更改

可以取消所做的任何未提交的更改。 SQL 允许您使用 ROLLBACK 语 句撤消最后一次提交后进行的所有更改。

ROLLBACK 语句撤消您最后一次使更改成为永久更改后对数据库所做的所有更改。(请参阅"使更改成为永久更改"第159页)。

有关 Interactive SQL 选项的更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > Interactive SQL 选项 "。

事务和数据恢复

假设系统错误或电源中断突然使您的数据库引擎停止工作。Adaptive Server Anywhere 经过精心设计,可以在这些情况下保护数据库的完整 性。它提供多种恢复数据库的独立方法。例如,它提供您可以在单独的 驱动器上存储的**日志文件**,以便在系统错误对一个驱动器造成损害时, 仍有办法恢复数据。此外,在使用日志文件时, Adaptive Server Anywhere 不必频繁更新您的数据库,从而改善了数据库的性能。

事务处理允许数据库服务器标识数据处于一致状态时的状态。事务处理确保:如果出于任何原因某一事务未成功完成,则撤消或回退整个事务。数据库保持完全不受失败事务的影响。

Adaptive Server Anywhere 的事务处理功能可以确保事务的内容被安全处理,即使在事务处理过程中出现系统错误的情况下也是如此。

有关数据恢复机制的详细说明,请参阅 "ASA 数据库管理指南 > 备份和 数据恢复 "。

完整性检查

Adaptive Server Anywhere 自动检查您的数据以找出某些常见错误。

插入重复数据

例如,假设您尝试创建一个部门,但所提供的 dept_id 值已被使用:

若要实现上述情况,请输入以下命令:

INSERT

INTO department (dept_id, dept_name, dept_head_id)
VALUES (200, 'Eastern Sales', 902)

INSERT 语句将被拒绝,因为它令该表的主键不是唯一的。因为 dept_id 字段是主键,所以不允许出现重复的值。

插入违反关系的值

以下语句将一个新行插入 sales_order 表中,但错误地提供了在职员表中不存在的 sales_rep ID。

INSERT
INTO sales_order (id, cust_id, order_date,
sales_rep)
VALUES (2700, 186, '1995-10-19', 284)

在 employee 表和 sales_order 表之间存在一对多关系,在 sales_order 表的 sales_rep 字段和 employee 表的 emp_id 字段之间具有连接。只有在包含该 连接的主键的表 (employee 表)中输入一个记录后,才能在在包含外键 的表 (sales_order 表)中插入相应记录。

外键employee 表的主键是职员 ID 号。sales_rep 表中的销售代表 ID 号是
employee 表的外键,这意味着,sales_order 表中的每一销售代表号都必
须与 employee 表中某些职员的职员 ID 号匹配。

在您试图为销售代表 284 添加一个订单时,将收到以下错误消息:

表 'sales_order' 中的外键 'ky_so_employee_id' 没有主键值

在 employee 表中没有具有该 ID 号的职员。这禁止您在没有有效销售代 表 ID 的情况下插入订单。此类有效性检查称作参照完整性检查,因为 它维持数据库中各表间的参照完整性。

有关主键和外键的更多信息,请参阅 "表间的关系 "第11页。

有关 DELETE 或 UPDATE 的错误

在执行更新或删除操作时也可能出现外键错误。例如,假设您尝试从部 门表中删除研发部门。*dept_id* 字段作为部门表的主键,构成了一对多关 系中的一方(*employee* 表的 *dept_id* 字段是相应的外键,因此构成了多 方)。在多方上的所有相应记录未全部删除前,不可以删除关系中的一 方上的记录。

DELETE FROM department WHERE dept_id = 100

示例: DELETE 错 将报告一个错误,指示在数据库中有其它记录引用了研发部门,并且未 执行删除操作。

表 'department' 中行的主键被表 'employee' 中的外键 'ky_dept_id' 引用

为了删除研发部门,您需要首先删除该部门中的所有职员:

DELETE FROM employee WHERE dept_id = 100

您现在可以执行研发部门的删除了。

您应该通过输入以下 ROLLBACK 语句取消对该数据库进行的上述更改 (以供将来使用):

ROLLBACK WORK

上一次成功执行 COMMIT WORK 后所进行的所有更改都将被撤消。如 果您尚未执行提交 (COMMIT),则自启动 Interactive SQL 后进行的所有 更改都将被撤消。 **示例: UPDATE 错** 如果您执行了造成数据库不一致的更新操作,将会生成与前面相同的错误 误消息。

例如,以下 UPDATE 语句会导致完整性错误:

UPDATE department SET dept_id = 600 WHERE dept_id = 100

在上述所有示例中,都要在执行每一命令时检查数据库的完整性。任何 会导致数据库不一致的操作都不会执行。

示例:在完成 COMMIT WORK 后检查完整性 可以对数据库进行配置,以便在执行 COMMIT WORK 前不检查完整 性。这在您要更改引用的主键的值时十分重要;例如,在 department 和 employee 表中将研发部门的 ID 从 100 更改为 600。为了进行这些更改, 数据库在这些更改间不得不出现不一致的情况。在这一情况下,您必须 配置数据库以只在提交时进行检查。

有关更多信息,请参阅 "ASA 数据库管理指南 > 数据库选项 > WAIT_FOR_COMMIT 选项 [database]"。

您还能够定义对外键进行自动修正。在上例中,如果已使用 ON UPDATE CASCADE 定义从 *employee* 到 *department* 的外键,则更新部门 ID 将自动更新职员表。

在前面的几个情况中,没有任何方法可以将不一致的数据库作为永久数据库提交。如果更改造成数据库不一致, Adaptive Server Anywhere 还支持其他操作以进行恢复。

有关更多信息,请参阅 "ASA SQL 用户指南 > 确保数据完整性 "一章。

完整性检查

^{第13章} 系统表

关于本章 本章介绍系统表,系统表是每个 Adaptive Server Anywhere 数据库中的 若干个特殊的表。这些系统表说明数据库中的所有表和列。当数据库结 构更改时,数据库服务器自动更新系统表。

系统表

Adaptive Server Anywhere 在**系统表**中存储有关数据库的重要信息。可以 用查看其它表中数据的相同方式查看系统表中的数据,但您不能在系统 表中更新数据。

关于本章

如果您要了解	那么请参阅
到何处查找数据库中所有表的列表	"SYSCATALOG 视图 " 第 167 页
到何处查找有关表中列的信息	"SYSCOLUMNS 视图 " 第 168 页
如何查找有关数据库中表的其它信息	"其它系统表"第169页
SYSCATALOG 视图

SYSCATALOG 是表现系统表 SYSTABLE 更可读形式的视图。它列出数据库中的所有表。您可以以查看数据库中任何其它表中的数据相同的方式查看 SYSCATALOG 视图中的数据。

有关查看数据的更多信息,请参阅 "选择完整的表 "第 106 页。

系统表的所有者 系统表和视图的所有者是特殊用户 ID SYS, 公司表的所有者是 DBA。 另外,有一组视图是特殊用户 ID dbo 所有的,这组视图提供 Sybase Adaptive Server Enterprise 系统目录的模拟;本节中不对这些表进行介 绍。

> 您应该想起该 DBA 是您从 Interactive SQL 连接到数据库时使用的用户 ID。目前,您只是键入了表名称 employee 和 department; SQL 在 SYSCATALOG 中查找具有由 DBA 创建的名称的表。在本示例中,通过键 入 SYS.SYSCATALOG,您指定 SYSCATALOG 是由用户 ID SYS 创建的。请注 意与限定列名称的方式的相似性,例如 employee.emp_id。

系统表中的其它列 此表中的其它列包含其它重要信息。例如,名为 Ncols 的列是每个表中 列的数目,名为 tabletype 的列将该表标识为基表或视图。

SYSCOLUMNS 视图

另一个重要的系统表是一个名为 SYSCOLUMNS 的视图。这是系统表 SYSCOLUMN 的可读形式。它说明数据库内所有表中的所有列。若要查 看表的内容,请键入以下命令,其中 tablename 表示您要列出其列的表 的名称:

```
SELECT *
FROM sys.syscolumns
WHERE tname = tablename
```

例如,

```
SELECT *
FROM sys.syscolumns
WHERE tname = 'employee'
ORDER BY colno
```

此语句列出 employee 表中的所有列。如果您向右查看列,则可以看到从 Coltype 列开始, employee 表中的某些列包含字符信息,而其它列包含整 数和日期信息。

您还可以在 Interactive SQL" 查找表名称 "对话框或 Sybase Central 中查 看列。

☆ 查看表或视图中的列 (Interactive SQL):

- 1 通过按 F7 键调用表的列表。
- 2 选择您要查看其列的表。
- 3 单击"显示列"。

☆ 查看表或视图中的列 (Sybase Central):

- 1 在 Sybase Central 的左窗格中,选择一个数据库,然后打开其 Tables 文件夹。
- 2 选择一个表,然后打开其 Columns 文件夹。
- 3 现在您可以查看并修改列。

其它系统表

数据库中还有一些其它的系统表,本教程未对它们进行说明。您可以通过检查 SYS.SYSCATALOG 找到其名称并在需要时查看这些表。

有关每个系统表的完整说明,请参阅 "ASA SQL 参考手册 > 系统表 "。

^{第 14 章} Microsoft Visual Basic 快速入门

关于本章 本章介绍如何使用 Adaptive Server Anywhere 和 Microsoft Visual Basic 开 发简单的数据库应用程序。

教程:开发 Visual Basic 应用程序

此简要教程基于 Visual Basic 6.0。完整的应用程序可在 Visual Basic 项目 Samples\ASA\VBStarter\ASAStarter.vbp 中找到。

Visual Basic 提供多种数据访问技术。在本教程中,我们将 Microsoft ADO Data Control 和 Adaptive Server Anywhere OLE DB 提供程序结合使用,从 Visual Basic 访问 Adaptive Server Anywhere 示例数据库。

◆ 使用 Visual Basic 开发数据库应用程序:

- 1 启动 Visual Basic,选择一个标准可执行文件工程。
- 2 将 Microsoft ADO Data Control 6.0 添加到您的工具选项板:
 - 在"工程"菜单中,选择"组件"。
 - 从列表中选择 Microsoft ADO Data Control 6.0 组件。

Components	×
Controls Designers Insertable Objects	
Kodak Image Thumbnail Control Metastream 1.0 Type Library Microsoft ActiveMovie Control Microsoft ActiveX Plugin Microsoft ADO Data Control 6.0 (OLEDB) Microsoft Agent Control 2.0 Microsoft Calendar Control 8.0 Microsoft Chart Control 6.0 (OLEDB) Microsoft Comm Control 6.0	
Microsoft Common Dialog Control 6.0 (SP3) Microsoft Data Bound Grid Control 5.0 (SP3) Microsoft Data Bound List Controls 6.0 Microsoft DataGrid Control 6.0 (OLEDB)	Browse
ОК	Cancel Apply

- 单击 " 确定 " 将控件添加到选项板。
- 3 将 ADO Data Control 添加到窗体,如下所示:



4 配置 ADO Data Control:

属性	值
ConnectionString	Provider=ASAPROV;DSN=ASA 8.0 Sample
CursorLocation	2 - asUseServer
CursorType	1 - adOpenKeyset
RecordSource	SELECT * FROM EMPLOYEE

ConnectionString 使用 Adaptive Server Anywhere OLE DB 提供程序 (ASAProv) 连接到 ASA 8.0 Sample 数据源。游标设置利用 Adaptive Server Anywhere 游标,而不使用客户端游标。

- 5 将两个文本框添加到窗体中,如下所示:
- 6 将文本框绑定到 ADO Data Control:
 - 将每个文本框的 DataSource 属性设置为 Adodc1。
 - 将左边文本框的 DataField 属性设置为 emp_fname, 这是保存职员名的列。
 - 将右边文本框的 DataField 属性设置为 emp_lname, 这是保存职员姓的列。

	۰,	F	a	r	m	1																											I	_			×
ł.	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•
		_																		_																	
		Г																11		Г																	
																				L.																	
•	•																		•	L.																	•
•	•	1																	•	1																	•
•	•	ID.	-	_	_	-	T	-	-	-	-	-	-	-	-	_	_	_	_	-	_	_	_	_	_	_	_	_	_	_	T	-	-	-	_	•	•
•	·	н	L	41		4	ι.	۸.	.		_ •	1																			Ш	L.		ь	ı I	•	•
•	•	н	Ľ			٩.	Ŀ	91		JO	С	L .																			Ш				1	•	•
		P	-	-	-	-	۰.																								1	-	-	-	_		
		1			1	1	1	1	1		1	1	1		-			1					1					-		1							
			•									•																								• •	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•

- 7 保存工程。
- 8 运行该示例:
 - 选择"运行"→"启动"运行该应用程序。

应用程序连接到 Adaptive Server Anywhere 示例数据库并 将第一个职员的姓名放在文本框中,如下所示:

🗣 Form1	
Fran	Whitney
Adodc1	♦

• 您可以使用 ADO Data Control 上的按钮滚动浏览结果集的 行。

您现在已经创建了一个使用 Adaptive Server Anywhere 的简单 Visual Basic 应用程序。

有关如何使用 ADO 和 Visual Basic 访问 Adaptive Server Anywhere 数据 库中数据的详细讨论,请参阅白皮书 Accessing Data in Adaptive Server Anywhere Using ADO and Visual Basic, 网址为 http://www.sybase.com/ detail?id=1017429。

术语表

Adaptive Server Anywhere (ASA)	SQL Anywhere Studio 的关系数据库服务器组件。
dbspace	为数据创建更多空间的附加数据库文件。一个数据库可以包含在最多 13 个独立的文件(一个初始文件和 12 个 dbspace)中。每个表及其索 引必须包含在单个数据库文件中。 SQL 命令 CREATE DBSPACE 可将 新文件添加到数据库中。
	另请参阅: "数据库文件 "第 185 页。
DBA 权限	一种权限级别,它允许用户在数据库中执行管理活动。DBA用户在缺省情况下具有DBA权限。
	另请参阅: "数据库管理员 (DBA)" 第 184 页。
FILE	在 SQL Remote 复制中, FILE 是为交换复制消息而使用共享文件的消息系统。对于在没有显式消息传送系统(如 MAPI)的情况下进行的测试和安装,它非常有用。
	另请参阅:"复制"第179页、"MAPI"第176页。
InfoMaker	一种报告和数据维护工具,它用于创建复杂的表格、报告、图形、交叉 表和表,并创建将这些报告用作构件块的应用程序。
Interactive SQL	一种 Adaptive Server Anywhere 应用程序,它用于查询和更改数据库中的数据以及修改数据库的结构。Interactive SQL 为您输入 SQL 语句提供了一个窗格,并且还提供了显示有关如何处理查询的信息以及结果集的窗格。
jConnect	JavaSoft JDBC 标准的 Java 实现。它为 Java 开发人员提供多层和异类环境中的本地数据库访问。
	另请参阅: "JDBC" 第 176 页。
Java 类	Java 代码的主要结构单元。它是组合在一起的过程和变量的集合,这些过程和变量都与一个特定的可识别类别相关。

JAR 文件	Java 存档文件。一种压缩的文件格式,由一个或多个用于 Java 应用程序的包的集合组成。它将安装和运行 Java 程序所必需的全部资源都放在一个压缩文件中。
JDBC	Java 数据库连接。一种 SQL 语言编程接口,它允许 Java 应用程序访问 关系数据。
LTM	日志传送管理器。请参阅"复制代理"第179页。
МАРІ	Microsoft 的消息处理应用程序编程接口。在多种常用的电子邮件系统 (如 Microsoft Mail)中使用的消息系统。
MobiLink	一种基于会话的同步技术,它用于使 UltraLite 和 Adaptive Server Anywhere 数据库与 Sybase 及其它供应商所提供的多种行业标准 SQL 数据库管理系统进行同步。
	另请参阅: "UltraLite" 第 177 页。
MobiLink 用户	MobiLink 用户是在同步系统中标识特定 MobiLink 客户端数据库的名称。当客户端连接到 MobiLink 同步服务器时,它会提供该名称和相关口令(可选)。MobiLink 用户名完全独立于数据库用户名。
NetWare	一种由 Novell 定义的广泛使用的网络操作系统。虽然 NetWare 也可能使用 TCP/IP 协议,但它通常会采用 IPX/SPX 协议。
ODBC	开放式数据库连接。用于数据库管理系统的标准 Windows 接口。 ODBC 是 Adaptive Server Anywhere 所支持的几种接口之一。
ODBC 管理器	一种随 Windows 操作系统提供的 Microsoft 程序,用于设置 ODBC 数据源。
ODBC 数据源	用户要通过 ODBC 访问的数据的规范以及获取该数据时所需的信息。
PowerDesigner	数据库建模应用程序。PowerDesigner为设计数据库或数据仓库提供了结构化的方法。
PowerDynamo	一种 Sybase 产品,用于建立和管理链接到数据库的 Web 应用程序。
PowerJ	一种 Sybase 产品,用于开发 Java 应用程序。
SQL Remote	一种基于消息的复制技术,用于在统一数据库和远程数据库中进行双向 复制。统一数据库必须是 Adaptive Server Anywhere 或 Adaptive Server

	Enterprise。远程数据库必须是 Adaptive Server Anywhere。
SQL	用于与关系数据库进行通信的语言。ANSI 已经为 SQL 指定了标准,最新的标准为 SQL-99(也称作 SQL3)。SQL 非正式地表示结构化查询语言。
SQL 语句	包含用于将指令传递给 DBMS 的 SQL 关键字的字符串。
	另请参阅: "模式 " 第 182 页、 "SQL" 第 177 页、 " 数据库管理系统 (DBMS)" 第 184 页。
Sybase Central	一种数据库管理工具,它在图形用户界面中提供 Adaptive Server Anywhere 数据库设置、属性和实用程序。 Sybase Central 也可用于管理 其它 Sybase 产品,其中包括 MobiLink。
SYS	一种特殊的用户,该用户拥有大多数系统对象。您不能作为 SYS 登录。
UltraLite	用于 Adaptive Server Anywhere 数据库的一种部署技术,面向小型、移动和嵌入式设备。所面向的平台包括手机、传呼机和个人记事本。
Windows	Microsoft Windows 操作系统系列,包括 Windows 95、 Windows 98、 Windows Me、 Windows CE、 Windows NT、 Windows 2000 和 Windows XP。
Windows CE	Microsoft 为移动设备制造的操作系统系列。
包	在 Java 中, 包是相关类集的集合。
标识符	用于引用数据库对象 (如表或列)的字符串。标识符可以包含 A 到 Z、 a 到 z、0 到 9、下划线 (_)、 at 符号 (@)、井号 (#) 或美元符号 (\$) 中的 任意字符。
并发	同时执行两个或更多个独立并且可能正在竞争的进程的情况。Adaptive Server Anywhere 自动使用锁定来隔离事务,并确保每个并发应用程序 看到一致的数据集。
	另请参阅:"事务"第183页、"锁定"第185页、"隔离级别"第180页。
参照完整性	遵守数据一致性(具体是指不同表中主键值和外键值之间的关系)规则的情况。若要实现参照完整性,每个外键中的值必须对应于被引用表中行的主键值。

另请参阅: "主键 " 第 190 页、 " 外键 " 第 186 页、 "Adaptive Server Anywhere (ASA)" 第 175 页。

 插件模块
 在 Sybase Central 中,插件模块是一种访问和管理产品的方式。当您安装相应的产品时,插件通常会自动安装并注册 Sybase Central。通常, 插件在 Sybase Central 主窗口中显示为顶级容器,并且使用产品本身的名称,如 Adaptive Server Anywhere。

另请参阅: "Sybase Central" 第 177 页。

查询 一条或一组 SQL 语句,用于访问和 / 或操作数据库中的数据。

另请参阅: "SQL" 第 177 页。

冲突触发器 在 SQL Remote 复制中,冲突触发器是在应用更新之前监测到更新冲突 时激发的触发器。当 UPDATE 语句 VERIFY 子句中的值未能匹配数据 库中的当前值时,冲突触发器便会激发。

另请参阅: "复制 " 第 179 页、 " 触发器 " 第 178 页。

- **重定向器** 一种 Web 服务器插件,它将路由客户端和 MobiLink 同步服务器之间的 请求和响应。此插件还实现可负载均衡和故障切换机制。
- **抽取** 在 SQL Remote 复制中,抽取是从统一数据库中卸载适当结构和数据的 操作。此信息用于初始化远程数据库。

在 MobiLink 同步中, 抽取是从参考数据库中卸载适当结构和数据的操作。

另请参阅:"复制"第179页。

触发器 一种特殊形式的存储过程,当用户执行修改数据的查询时,该存储过程 会自动执行。

另请参阅: "行级触发器 "第 180 页、"语句级触发器 "第 188 页、"冲 突触发器 "第 178 页、"完整性 "第 186 页。

- **存储过程** 由 SQL 指令序列组成的程序,它存储在数据库中,用于执行特定的任务。
- **代理表** 一种本地表,它所包含的元数据可以像访问本地表一样访问远程数据库 服务器上的表。

另请参阅: "元数据"第188页。 对象树 在 Sybase Central 中,对象树是数据库对象的层次结构。对象树的顶层 显示您的 Sybase Central 版本所支持的全部产品。每种产品展开以后会 显示它自己的对象子树。 另请参阅: "Sybase Central" 第 177 页。 在 SQL Remote 或 MobiLink 中,发布是标识复制数据的数据库对象。 发布 在 MobiLink 中,发布仅存在于客户端。一个发布包括多个项目。对每 个发布做出的更改会定期复制给该发布的所有预订者。 SOL Remote 用 户可以通过预订发布来接收该发布。 MobiLink 用户可以通过创建与发 布的同步预订来同步发布。 另请参阅: "复制 " 第 179 页、 " 项目 " 第 187 页、 " 发布更新 " 第 179 页。 发布更新 在 SOL Remote 复制中,发布更新是对一个数据库中的一个或多个发布 做出的更改列表。发布更新将作为复制消息的一部分定期发送到远程数 据库。 另请参阅:"复制"第179页、"发布"第179页。 发布者 在 SOL Remote 复制中,发布者是数据库中可以与其它复制数据库交换 复制消息的一个用户。 另请参阅:"复制"第179页。 在 Windows 操作系统上,服务是在运行应用程序的用户 ID 未登录时的 服务 应用程序运行方式。 在物理上不相同的数据库之间的数据共享。 Sybase 拥有三项复制技术: 复制 MobiLink、 SQL Remote 和复制服务器。 在复制服务器中,复制代理是用来读取数据库事务日志并将提交的更改 复制代理 发送到复制服务器的程序,它也称作日志传送管理器 (LTM)。 复制服务器 Sybase 的一种基于连接的复制技术,它与 Adaptive Server Anywhere 和 Adaptive Server Enterprise 一起使用。它用于少量数据库之间接近实时的 复制。

复制频率	在 SQL Remote 复制中,复制频率是每个远程用户的一项设置,它确定 发布者的消息代理应该隔多长时间向该远程用户发送一次复制消息。
	另请参阅:"复制"第179页、"远程用户"第189页。
复制消息	在 SQL Remote 或复制服务器中,复制消息是在发布数据库和预订数据 库之间发送的通信消息。消息包含数据、直通语句以及复制系统所需的 信息。
	另请参阅:"直通"第190页、"复制"第179页、"发布更新"第179页。
隔离级别	一个事务中的操作对于其他并发事务中的操作的可见程度。隔离级别有 四级,编号依次为0至3。第3级提供最高级别的隔离。第0级为缺省 设置。
个人服务器	与客户端应用程序在同一台计算机上运行的数据库服务器。个人数据库 服务器通常由一名用户在一台计算机上使用,但它可以支持该用户的多 个并发连接。
工作表	一种内部存储区域,用于在查询优化过程中存储中间结果。
故障切换	在活动服务器、系统或网络出现故障或意外终止时切换到冗余或备用的 服务器、系统或网络。故障切换会自动发生,并且通常置入持续可用的 系统中。
关系数据库管理系统	一种数据库管理系统,它以相关表的形式存储数据。
(RDDIVIS)	另请参阅: "数据库管理系统 (DBMS)" 第 184 页。
规范化	对数据库结构的改进,目的在于按照基于关系数据库理论的规则消除冗 余并改善组织。
归类	字符集和排序顺序的组合,它定义数据库中文本的属性。对于 Adaptive Server Anywhere 数据库,缺省的归类取决于用来运行服务器的操作系统和语言;例如,英语 Windows 系统上的缺省归类为 1252LATIN1。归类(也称作归类序列)用于对字符串进行比较和排序。
行级触发器	每更改一行即执行一次的触发器。
	另请参阅:"触发器"第178页、"语句级触发器"第188页。

回退日志	在每个未提交的事务过程中做出的更改的记录。当收到 ROLLBACK 请 求或者系统出现故障时,未提交的事务会从数据库中回退,将数据库返 回其原先的状态。每个事务都有一个单独的回退日志,当事务完成后, 该日志将被删除。
	另请参阅: "事务 " 第 183 页。
基表	永久的数据表。为了区别于临时表和视图,这种表有时被称作 基表 。
	另请参阅: "临时表 " 第 182 页、 " 视图 " 第 183 页。
集成登录	一种登录功能,它允许将同一个用户 ID 和口令用于操作系统登录、网络登录和数据库连接。
检查点	将数据库的所有更改都保存到数据库文件的时间点。在其它时间,所提 交的更改仅保存到事务日志中。
检查约束	对列或列集强制实施指定条件的一种限制。
	另请参阅: "约束 "第 189 页、 "外键约束 "第 186 页、 "主键约束 "第 190 页、 "唯一约束 "第 186 页。
校验	测试数据库、表或索引是否受到特定类型的文件损坏。
局部临时表	一种临时表,它仅在复合语句期间或连接结束之前存在。当您只需要将数据集装载一次时,局部临时表非常有用。缺省情况下,行会在提交时被删除。
	另请参阅: "临时表 "第 182 页、 "全局临时表 "第 183 页。
角色	在概念数据库建模中,角色是从一个特定角度来描述关系的动词或短语。您可以用两个角色来描述每种关系。例如 "contains" (包含)角色和 "is a member of" (隶属于)角色。
角色名	外键的名称。由于它命名外表和主表之间的关系,因此称作角色名。缺 省情况下,角色名就是表名,除非其它外键已经使用该名称(在这种 情况下,缺省的角色名是表名后接一个三位的唯一数字)。您也可以自 己创建角色名。
	另请参阅: "外键"第186页。
客户 / 服务器	一种软件体系结构,其中的一种应用程序(客户端)从另一种应用程序(服务器)获取信息并向该应用程序发送信息。这两种应用程序通常位于通过网络连接的不同计算机上。

- **连接** 关系系统中的一种基本操作,它通过比较指定列中的值将两个或更多个表中的行链接在一起。
- **连接 ID** 标识客户端应用程序和数据库之间给定连接的唯一编号。您可以使用以下 SQL 语句确定当前的连接 ID:

select connection property('Number')

连接类型 Adaptive Server Anywhere 提供了四种类型的连接:交叉连接、键连接、 自然连接和使用 ON 子句的连接。

另请参阅: "连接"第182页。

- **连接配置** 连接到数据库所需的一组参数,如用户名、口令和服务器名称,它们在存储后即可方便地使用。
- **连接条件** 影响连接结果的限制。您可以通过紧跟在连接语句的后面插入 ON 子句 或 WHERE 子句来指定连接条件。对于自然连接和键连接, Adaptive Server Anywhere 会创建连接条件。

另请参阅: "连接 "第182页、"生成的连接条件"第183页。

临时表 为临时存储数据而创建的表。临时表有两种类型:全局和局部。

另请参阅: "局部临时表 " 第 181 页、 " 全局临时表 " 第 183 页。

- 模式 数据库的结构,其中包括表、列和索引以及它们之间的关系。
- **命令文件** 包含 SQL 语句的文本文件。命令文件可以手动建立,也可以通过数据 库实用程序自动建立。例如,dbunload 实用程序会创建一个命令文件, 该文件包含重新创建给定数据库所必需的 SOL 语句。
- **内连接** 一种连接,在这种连接中,只有在两个表都满足连接条件时行才在结果 集中出现。内连接是缺省设置。

另请参阅: "连接"第182页、"外连接"第186页。

嵌入式 SQLC 程序的编程接口。 Adaptive Server Anywhere 嵌入式 SQL 是 ANSI 和
IBM 标准的实现。

全局临时表	一种临时表,在被显式删除之前,其数据定义对于所有用户都可见。全局临时表允许用户各自打开一个表的相同实例。缺省情况下,行会在提交时被删除,并且行始终会在连接结束被删除。
	另请参阅: "临时表 "第 182 页、 "局部临时表 "第 181 页。
日志文件	Adaptive Server Anywhere 所维护的事务日志。日志文件用于确保数据库 在出现系统或介质故障时可以恢复,提高数据库性能,以及通过 SQL Remote 支持数据复制。
	另请参阅: "事务日志 " 第 183 页、 " 事务日志镜像 " 第 183 页、 " 完全 备份 " 第 186 页。
生成的连接条件	对基于关键字 KEY 或 NATURAL 的连接结果的限制。对于自然连接, 生成的连接条件基于两个表中的公共列名。对于键连接,该条件基于两 个表之间的外键关系。
	另请参阅: "连接 " 第 182 页、 " 连接条件 " 第 182 页。
视图	作为对象存储在数据库中的 SELECT 语句。它使用户能够看到一个或多 个表中的行子集或列子集。每当用户使用特定表或表组合的视图时,都 将利用存储在这些表中的信息重新计算视图。视图对于确保安全非常有 用,它们可用于定制数据库信息的外观,以使得数据访问更加直接。
事务	组成一个逻辑工作单元的 SQL 语句序列。事务要么全部得到处理,要 么根本不做处理。Adaptive Server Anywhere 利用内置的锁定功能支持 事务处理,允许并发事务访问数据库而不会损坏数据。事务以 COMMIT 语句 (使对数据的更改变为永久更改)结束或者以 ROLLBACK 语句 (撤消在事务过程中做出的所有更改)结束。
事务日志	按照更改顺序存储对数据库做出的所有更改的文件。它可提高性能并支 持在数据库文件损坏时恢复数据。为了获取最佳结果,日志文件应该与 数据库文件保留在不同的设备上。
事务日志镜像	同时维护的事务日志文件可选相同副本。每当数据库更改写入事务日志 文件时,它同时也会写入事务日志镜像文件。
	镜像文件应该与事务日志保留在不同的设备上,这样在任意设备出现故 障时,日志的其它副本会确保数据可以安全地恢复。
	另请参阅: "事务日志 " 第 183 页。
授权选项	一种权限级别,它允许用户向其他用户授予权限。

- 数据操作语言 SQL 语句的子集,用于检索和更新数据库的内容。 (DML)
- **数据定义语言** SQL 语句的子集,用于建立数据库结构的模型。DDL 语句可以创建、 (DDL) 修改和删除数据库对象(包括用户)。
- **数据库** 通过主键和外键关联的表的集合。表包含数据库中的信息。表和键一起 定义数据库的结构。数据库管理系统访问这些信息。

另请参阅: "外键"第186页、"主键"第190页、"数据库管理系统(DBMS)"第184页、"关系数据库管理系统(RDBMS)"第180页。

- **数据库对象** 包含或接收信息的数据库组件。表、索引、视图、过程和触发器便是数 据库对象。
- **数据库服务器** 管理对数据库信息的所有访问的计算机程序。Adaptive Server Anywhere 提供两种服务器:网络服务器和个人服务器。
- **数据库管理系统** 用于创建和使用数据库的程序的集合。

另请参阅:"关系数据库管理系统 (RDBMS)" 第 180 页。

- 数据库管理员具有维护数据库所需权限的用户。DBA 通常负责对数据库模式的所有(DBA)更改,并且负责管理用户和用户组。一个用户 ID 为 DBA、口令为 SQL数据库管理员的角色会自动置入数据库。
- **数据库连接** 客户端应用程序和数据库之间的通信渠道。必须具有有效的用户 ID 和 口令才能建立连接。向用户 ID 授予的特权确定在连接过程中可以执行 的操作。
- **数据库名称** 在服务器装载数据库时为数据库指定的名称。缺省的数据库名称是初始 数据库文件的文件名 (不含扩展名)。

另请参阅: "数据库文件"第185页。

数据库所有者 (dbo) 一种特殊的用户,他拥有不归 SYS 所有的系统对象。

另请参阅: "数据库管理员 (DBA)" 第 184 页、 "SYS" 第 177 页。

(DBMS)

数据库文件	数据库包含在一个或多个数据库文件中。其中一个是初始文件,后面的 文件称作 dbspace。每个表(包括其索引)都必须包含在单个数据库文 件中。
	另请参阅: "dbspace" 第 175 页。
数据类型	数据的格式,如 CHAR 或 NUMERIC。在 ANSI SQL 标准中,数据类型 也可以包括对大小、字符集和归类的限制。
	另请参阅: "用户定义数据类型 "第 188 页。
锁定	一种并发控制机制,它在同时执行多个事务的过程中保护数据的完整性。 Adaptive Server Anywhere 会自动应用锁定,以防止两个连接同时更改相同的数据,并防止其它连接读取正在被更改的数据。
	您可以通过设置隔离级别来控制锁定。
	另请参阅: "隔离级别 "第 180 页、"并发 "第 177 页、"完整性 "第 186 页。
索引	一组排序的键和指针,它们与基表中的一个或多个列相关联。表中一个或多个列上的索引可以提高性能。
同步	利用 MobiLink 技术在数据库之间复制数据的过程。
	在 SQL Remote 中,同步专指用初始数据集初始化远程数据库的过程。
	另请参阅: "MobiLink" 第 176 页、 "SQL Remote" 第 176 页。
统一数据库	在数据库复制中,统一数据库是存储数据主副本的数据库。统一数据库 包含所有数据,而远程数据库通常只包含数据子集。一旦出现冲突或不 一致,统一数据库将被认为具有所有数据的主副本。
	另请参阅:"复制"第179页。
外表	包含外键的表。
	另请参阅: "外键"第186页。
外部登录	与远程服务器通信时使用的替代登录名和口令。缺省情况下,Adaptive Server Anywhere 每次代表其客户端连接到远程服务器时都会使用其客 户端的名称和口令。但是,您可以通过创建外部登录来替换这一缺省设 置。外部登录是与远程服务器通信时使用的替代登录名和口令。

外键 在一个表中复制另一个表中主键值的一个或多个列。外键建立表间的关系。

另请参阅:"主键"第190页、"外表"第185页。

外键约束 对列或列集的限制,它指定表中的数据如何与其它表中的数据相关联。 对列集施加外键约束可使这些列成为外键。

另请参阅: "约束 "第 189 页、 "检查约束 "第 181 页、 "主键约束 "第 190 页、 "唯一约束 "第 186 页。

外连接 保留表中所有行的连接。Adaptive Server Anywhere 支持左、右和完全 外连接。左外连接保留表中位于连接运算符左侧的行,当右表中的行不 满足连接条件时,它将返回空值。完全外连接保留两个表中的所有行。

另请参阅: "连接"第182页、"内连接"第182页。

完全备份 整个数据库和事务日志(可选)的备份。完全备份包含数据库中的所有信息,因此会在系统或介质出现故障时提供保护。

另请参阅: " 增量备份 " 第 190 页。

完整性 遵守完整性规则的情况,完整性规则确保数据正确而准确,而且数据库 的关系结构保持不变。

另请参阅: "参照完整性"第177页。

网络服务器 从共享公共网络的计算机接受连接的数据库服务器。

另请参阅: "个人服务器"第 180 页。

唯一约束 对列或列集的限制,它要求所有非空值都互不相同。一个表可以有多个 唯一约束。

另请参阅: "外键约束 "第 186 页、"主键约束 "第 190 页、"约束 "第 189 页。

- **谓词** 一种条件表达式,可以选择性与逻辑运算符 AND 和 OR 组合在一起以 组成 WHERE 或 HAVING 子句中的条件集。在 SQL 中,取值为 UNKNOWN 的谓词被解释为 FALSE。
- **系统表** 由 SYS 或 dbo 拥有的包含元数据的表。数据表也称作数据字典表,是 由数据库服务器创建并维护的。

系统对家 田 S	イS 및 dbo 拥有旳致据库对冢。
----------	--------------------

- **系统视图** 存在于每一个数据库中的一种视图,它以易于理解的格式表示系统表中 包含的信息。
- **相关名** 在查询的 FROM 子句中使用的表或视图的名称 它的初始名称或者在 FROM 子句中定义的别名。
- **项目** 在 SQL Remote 或 MobiLink 中,项目是表示整个表或表中行列子集的 数据库对象。项目在发布中组合在一起。

另请参阅:"复制"第179页、"发布"第179页。

消息类型 在 SQL Remote 复制中,消息类型是指定远程用户如何与统一数据库的 发布者通信的数据库对象。统一数据库可能定义了多种数据类型;这 样,不同的远程用户就可以使用不同的消息系统与该数据库进行通信。

另请参阅: "复制 " 第 179 页、" 统一数据库 " 第 185 页、"MAPI" 第 176 页。

消息系统 在 SQL Remote 复制中,消息系统是用于在统一数据库和远程数据库之间交换信息的协议。 Adaptive Server Anywhere 包含对以下消息系统的支持: FILE、 MAPI、 FTP、 SMTP 和 VIM。

另请参阅:"复制"第179页、"FILE"第175页、"MAPI"第176页。

写文件 用于记录对只读数据库的更改的文件。通常用于压缩的数据库。

另请参阅: "压缩的数据库文件"第187页。

卸载 卸载数据库会将数据库的结构和 / 或数据导出到文本文件(对于结构为 SQL 命令文件,对于数据为 ASCII 逗号分隔文件)。使用 Unload 实用 程序可卸载数据库。

此外,您也可以使用 UNLOAD 语句卸载数据的选定部分。

- 性能统计 反映数据库系统性能的值。例如, CURRREAD 统计表示引擎已发出但 未完成的文件读取次数。
- **压缩的数据库文件** 已经使用实用程序 dbshrink 压缩到较小物理大小的数据库文件。压缩的 数据库是只读的。若要对压缩的数据库文件做出更改,必须使用关联的 写文件。您可以使用实用程序 dbexpand 将压缩的数据库文件展开为正 常的数据库文件。

业务规则 基于现实世界要求的准则。通常,业务规则通过检查约束、用户定义数 据类型以及事务的正确使用来实现。

另请参阅: "约束 "第189页、"用户定义数据类型 "第188页。

用户定义数据类型 用户创建的数据类型,用于指定基本数据类型,它还可以指定缺省值、 检查条件或是否为空。用户定义数据类型也称作用户定义域,它们可应 用于列以强制整个数据库保持一致性。

另请参阅: "数据类型"第185页。

游标 指向结果集的命名链接,用于通过编程接口访问和更新行。在 Adaptive Server Anywhere 中,游标支持在查询结果中向前和向后移动。游标由 两个部分组成:游标结果集 (通常由 SELECT 语句定义)和游标位置。

另请参阅: "游标结果集 "第 188 页、"游标位置 "第 188 页。

游标结果集 与游标关联的查询所得到的行集。

另请参阅: "游标 "第188页、"游标位置"第188页。

游标位置 指向游标结果集中一个行的指针。

另请参阅: " 游标 " 第 188 页、 " 游标结果集 " 第 188 页。

语句级触发器 在整个触发语句完成后执行的触发器。

另请参阅: " 触发器 " 第 178 页、 " 行级触发器 " 第 180 页。

预订 在 SQL Remote 复制中,预订是发布和远程用户之间的链接,它使用户 能够与统一数据库交换该发布的更新。

在 MobiLink 同步中,同步预订是发布和 MobiLink 用户之间的客户端数 据库中的链接,它使发布所描述的数据能够得到同步。

另请参阅: "发布 "第 179 页、" 远程用户 "第 189 页、"MobiLink 用户 "第 176 页。

元数据 有关数据的数据。元数据描述其它数据的性质和内容。

另请参阅: "模式 "第 182 页。

远程 DBA 权限
 在 SQL Remote 中,远程 DBA 权限是消息代理所需的权限级别。在
 MobiLink 中,远程 DBA 权限是 Adaptive Server Anywhere 同步客户端
 (dbmlsync)所需的权限级别。当消息代理或同步客户端作为具有该权
 限的用户建立连接时,它将具有完全的 DBA 访问权。如果不通过消息
 代理或同步客户端来建立连接,该用户 ID 将不具有其它权限。

另请参阅: "DBA 权限 " 第 175 页。

远程权限 在 SQL Remote 复制中,远程权限是与发布数据库交换复制消息的权限。向用户授予远程权限将使该用户成为远程用户。您必须指定消息类型、适当的远程地址和复制频率。概括地说,远程权限也可以指参与SQL Remote 复制的任何用户 (例如,统一发布者和远程发布者)。

另请参阅:"复制"第179页。

远程数据库 在 SQL Remote 复制或 MobiLink 同步中,远程数据库是与统一数据库 交换数据的数据库。远程数据库可能会共享统一数据库中的全部或部分 数据。

另请参阅:"复制"第179页、"统一数据库"第185页。

远程用户 在 SQL Remote 复制中,远程用户是统一数据库中已授予远程权限并且 与复制设置中的一个特定远程数据库关联的数据库用户。若要创建远程 用户,请向普通数据库用户授予远程权限。这样做不仅会确定特定的远 程数据库是否存在,而且还指定用于与该特定远程站点通信的消息类型 和地址。

如果通过从统一数据库中抽取的方式来创建远程数据库,统一数据库中的每一个远程用户都会成为一个特定远程数据库中数据的发布者。

另请参阅: "SQL Remote" 第 176 页、 " 统一数据库 " 第 185 页、 " 发布者 " 第 179 页。

约束 对特定数据库对象(如表或列)中包含的值的限制。例如,列可以具有唯一性约束,该约束要求该列中的所有值互不相同。表可以具有外键约束,该约束指定该表中的信息如何与其它表中的数据相关联。

另请参阅: "检查约束"第 181 页、"外键约束"第 186 页、"主键约束" 第 190 页、 "唯一约束"第 186 页。 **增量备份** 仅包含事务日志的备份,通常在两次完全备份之间使用。

另请参阅: "事务日志 "第 183 页。

- **争用**为获取资源而竞争的行为。例如,就数据库而言,两个或更多个试图编 辑数据库中同一行的用户就是在争用编辑该行的权利。
- **直通** 在 SQL Remote 复制中,直通 (Passthrough) 是统一数据库的发布者可直 接用 SQL 语句更改远程数据库的一种模式。直通是为特定远程数据库 设置的。在常规直通模式下,在统一数据库做出的所有数据库更改都会 直通到选定的远程数据库。在纯直通模式下,更改在远程数据库中发 生,而不会在统一数据库中发生。
- **主表** 包含外键关系中的主键的表。

主键 其值唯一标识表中各行的一个列或多个列。

另请参阅 " 外键 " 第 186 页。

主键约束 对主键列的唯一性约束。一个表只能有一个主键约束。

另请参阅: "约束 "第 189 页、 "检查约束 "第 181 页、 "外键约束 "第 186 页、 "唯一约束 "第 186 页、 "完整性 "第 186 页。

子查询 嵌套在 SELECT、 INSERT、 UPDATE 或 DELETE 语句或者其它子查询 中的 SELECT 语句。

子查询有两种类型:相关和嵌套。

索引

字母

Adaptive Server Anywhere 编程接口 36 简介 24 快速入门1 内部 42 特点 26 系统要求 27 应用程序 30 用途 25 Adaptive Server Anywhere 简介 23 ADO 开发工具 41 数据控制 172 ALTER 语句 自动提交 158 API Adaptive Server Anywhere 36 ADO 41 JDBC 40 ODBC 38 OLE DB 41 **Open Client 39** 嵌入式 SQL 39 ASA 术语定义 175 asademo.db 文件 关于 xii, 48 AUTO COMMIT 选项 在 Interactive SQL 中组合更改 158 BETWEEN 条件 WHERE 子句 120

BLOB 关于 59 **COMMENT** 语句 自动提交 158 **COMMIT**语句 关于 159 事务 158 **COUNT** 函数 应用于分组数据 138 **DBA** 权限 术语定义 175 dbeng8 限制 29 dbisql 实用程序 关于 71 DBMS 术语定义 184 dbspace 术语定义 175 DDL 术语定义 184 DELETE 语句 错误 162 关于 157 示例 162 DML 术语定义 184 **DROP**语句 自动提交 158 FILE 术语定义 175 go 语句分隔符 87

GROUP BY 子句
错误 138
集合函数 138
HAVING 子句
GROUP BY 子句和 140
WHERE 子句和 141
HPC
Windows CE 支持 27
HPC Pro
Windows CE 文持 27
IN 余件 120 InfoMaker
术语定义 175
INSERT 语句
简介 155
示例 161
Interactive SQL
JDBC 转义语法 100
SQL 语句窗格 106
保存命令 98, 106
报告的错误 85
编辑表值 81, 82
插入行 82
查找表名称 87
查找过程名称 87
查找列名称 87
错误处理 91
打开多个窗口 76
调用命令 89
多个结果集 92
复制行 84
概述 72
更新计算列 82
工具栏说明 76
功能键 77
关于 71
记录命令 90, 91

简介 31
键盘快捷方式 77
将更改组合到事务中 158
截断列 92
结果窗格选项 92
快速入门 6
列长度 92
命令概述 85
命令文件 97
配置 91
批处理操作 99
启动 74
取得命令 106
取消命令 85
删除行 83
设置首选项 91
术语定义 175
停止命令 85
显示表的列表 125
显示数据 80
显示一组表 87
显示一组过程 87
现有效果 158
消息窗格选项 94
选项对话框 91
运行脚本 98
执行命令 85,106
中断命令 85
主窗口说明 74
装载命令 98, 106
组合多个语句 86
ISQL 1
JAR 文件
不倍定义 176 L · · · · · · · · · · · · · · · · · · ·
Java 调试性户 简介 21
间开 51 Java 类

术语定义 175 jConnect 术语定义 175 **JDBC** Interactive SQL 中的转义语法 100 开发工具 40 术语定义 176 JDBC 转义语法 在 Interactive SQL 中使用 100 **LIKE**条件 介绍 118 LTM 术语定义 179 MAPI 术语定义 176 Microsoft Visual Basic 快速入门 171 MIPS 芯片 Windows CE 支持 27 MobiL ink 术语定义 176 MobiLink 用户 术语定义 176 NetWare 术语定义 176 NULL 列中允许 60 ODBC 开发工具 38 术语定义 176 数据源简介 67 ODBC 管理器 术语定义 176 ODBC 数据源 术语定义 176 OLE DB 开发工具 41 OLE DB 和 ADO 编程接口 171 **ON** 短语 介绍 127

Open Client 开发工具 39 **ORDER BY**子句 使用索引提高性能 113 示例 111 需要该子句以确保行始终以相同的顺序显示 113 PowerDesigner 术语定义 176 PowerDynamo 术语定义 176 **PowerJ** 术语定义 176 PPC Windows CE 支持 27 **RDBMS** 术语定义 180 已定义 10 ROLLBACK 语句 关于 160 简介 157 事务 158 SELECT 语句 Interactive SQL 80 介绍 104 子查询 145 SOUNDEX 函数 关于 119 SOL 关于 15 开发查询 72 术语定义 177 SQL Anywhere Studio 文档 vii SOL Remote 术语定义 176 SQL 和数据库计算 15 **SQL**语句 术语定义 177 Sybase Central

简介 30 快速入门4 术语定义 177 SYS 术语定义 177 SYSCATALOG 视图 关于 167 SYSCOLUMNS 视图 关于 168 TDS 通信协议 图示 36 UltraLite 术语定义 177 **UPDATE** 语句 错误 163 简介 156 示例 163 Visual Basic 快速入门 172 WHERE 子句 BETWEEN 条件 120 HAVING 子句和 141 模式匹配 118 日期比较介绍 116 删除行 157 示例 115 修改表中各行 156 Windows 术语定义 177 支持的操作系统 27 Windows CE 术语定义 177 支持的版本 27 支持的设备 27

术语定义 177 保存 Interactive SQL 中的命令 98 事务结果 158 比较 关于 115 介绍 116 使用子查询 148 编程接口 Adaptive Server Anywhere 36 ADO 41 JDBC 40 ODBC 38 OLE DB 41 Open Client 39 嵌入式 SQL 39 在 Adaptive Server Anywhere 中支持的 36 编辑 Interactive SQL 中的表值 81,82 标识符 术语定义 177 表 关于 10 设计 56 特性 10 外键 12 在 Interactive SQL 中查找 87 表值 在 Interactive SQL 中编辑 81,82 别名 用于列 109 并发 术语定义 177 不等式 测试 116

B

包

С

参数 函数 137 参照完整性 术语定义 177 操作系统 支持的 27 插件模块 术语定义 178 插入 在 Interactive SOL 中将行插入表中 82 查询 Interactive SQL 85 SELECT 语句 104 术语定义 178 已定义 16 查询优化 Adaptive Server Anywhere 特点 26 查找 Interactive SOL 中的表 87 Interactive SQL 中的过程 87 Interactive SQL 中的列 87 查找表名称对话框 显示表的列表 125 程序组 Adaptive Server Anywhere 32 冲突触发器 术语定义 178 抽取 术语定义 178 触发器 术语定义 178 创建 简单 ODBC 数据源 67 数据库 53 创建数据库向导 使用 53 从表中选择行 115 从多个表中选择数据 123

从数据库表中选择数据 103 存储过程 术语定义 178 错误 Interactive SQL 85 错误处理 Interactive SQL 91

D

打开多个 Interactive SQL 窗口 76 代理表 术语定义 178 导出数据 Interactive SQL 选项 93 导入数据 Interactive SQL 选项 93 调试 简介 31 调试程序 [dbprdbg] 实用程序 简介 31 调用 Interactive SOL 中的命令 89 对象树 术语定义 179 多对多关系 已定义 62 多个结果集 在 Interactive SQL 中查看 92

Е

二进制大对象 关于 59

F

发布

术语定义 179 发布更新 术语定义 179 发布者 术语定义 179 反馈 提供 xii 文档 xii 分组数据 137 服务 术语定义 179 复合搜索条件 使用 119 复制 Interactive SOL 中的行 84 术语定义 179 复制服务器 术语定义 179 复制频率 术语定义 180 复制消息 术语定义 180

G

概念数据库模型 定义 53 隔离级别 术语定义 180 个人服务器 关于 29 平台支持 29 术语定义 180 限制 29 更新 Interactive SQL 中的值 82 数据 153 更新数据库 153 工具栏 Interactive SQL 76 工作表 术语定义 180 工作组计算 要求 25 功能键 Interactive SQL 77 故障切换 术语定义 180 关系 多对多 62 关于 61 基数 61 实体 11 一对多 62 一对一 61 关系数据库 概念 10 关于 11 术语 11 关系数据库管理系统 已定义 10 规范化 术语定义 180 归类 术语定义 180 过程 在 Interactive SQL 中查找 87

Η

函数 SOUNDEX 函数 119 恢复 Adaptive Server Anywhere 特点 26 回退 事务 158 回退日志 术语定义 181 汇总数据 136

J

基表 13 术语定义 181 基数 关系和 61 集成登录 术语定义 181 集合函数 简介 137 应用于分组数据 138 技术支持 新闻组 xii 计划 Interactive SQL 选项 95 在 Interactive SOL 中的外观 95 在 Interactive SQL 中设置游标类型 95 计算列 在 Interactive SQL 中更新 82 在 Interactive SOL 中添加到新行 83 在 Interactive SQL 中重新计算 82 记录 Interactive SQL 中的命令 90 检查 数据完整性 161 检查点 术语定义 181 检查约束 术语定义 181 检索 Interactive SQL 中的命令 89 键 关于 11 外 11

主 11 键连接 介绍 129 键盘快捷方式 Interactive SOL 77 将更改组合到事务中 158 将您的应用程序连接到其数据库 65 脚本 在 Interactive SQL 中运行 98 角色 术语定义 181 角色名 术语定义 181 教程 开发 Visual Basic 应用程序 172 设计数据库 53 截断 Interactive SQL 中的列 92 结果集 插入行 82 多次执行查询 111 复制行 84 删除行 83 疑难解答 113 在 Interactive SQL 中编辑表值 81,82 结束事务 158 局部临时表 术语定义 181

K

开发 SQL 语句 72 开始 事务 158 开始菜单 Adaptive Server Anywhere 32 可用性 Adaptive Server Anywhere 组件 32 客户 / 服务器 术语定义 181 空 在列中允许 155 空值 在 Interactive SQL 中的外观 92 口令 连接到新数据库 54 快速入门 Adaptive Server Anywhere 1 Interactive SQL 6 Sybase Central 4 开发 Visual Basic 应用程序 172 数据库服务器 2

L

连接 或子查询 150 简介 66 介绍 124 术语定义 182 连接 ID 术语定义 182 连接类型 术语定义 182 连接配置 术语定义 182 连接条件 术语定义 182 歽 别名 109 从表中选择 108 关于 10 计算 109 排序 108

数据类型 59

允许 NULL 60 在 Interactive SQL 中查找 87 在 Interactive SQL 中截断 92 临时表 术语定义 182 临时文件 简介 43

\mathbf{M}

命令 Interactive SQL 85 上一个 89 在 Interactive SQL 中保存 98, 106 在 Interactive SQL 中编辑 89 在 Interactive SQL 中调用 89 在 Interactive SOL 中记录 90 在 Interactive SQL 中取得 106 在 Interactive SQL 中取消 85 在 Interactive SOL 中停止 85 在 Interactive SQL 中执行 85, 106 在 Interactive SOL 中中断 85 在 Interactive SQL 中装载 98, 106 命令历史记录窗口 在 Interactive SQL 中调用命令 89 命令文件 SQL 语句窗格 86,97 概述 86,97 建立 86,97 术语定义 182 命令行实用程序 简介 32 命令序列通信协议 关于 37 图示 36 模式 术语定义 182

已定义 53 模式匹配 介绍 118

Ν

内部 Adaptive Server Anywhere 42 数据库服务器 42 内连接 术语定义 182

P

排序 查询结果 111 配置 Interactive SQL 91 批处理操作 Interactive SQL 99 平台 支持的 27

Q

启动 Interactive SQL 74 嵌入式 SQL 开发工具 39 术语定义 182 嵌入式数据库 要求 25 清除 SQL 语句窗格 85 区分大小写 SQL 106 表名称 106 取消 Interactive SQL 命令 85 全局临时表 术语定义 183

R

日期 复合 119 搜索条件 120 搜索条件介绍 116 组合 119 日志文件 术语定义 183

S

删除 从表中删除行 83 使用 Interactive SQL 删除行 83 设计 数据库 45,53 设计和建立数据库 45 生成的连接条件 术语定义 183 实用程序 简介 32 矢量积 介绍 126 使用 Interactive SQL 71 使用 Interactive SQL 显示数据 80 使用子查询选择数据 143 示例数据库 关于 48 关于 asademo.db xii 事务 开始 158 术语定义 183 数据恢复 160 完成 158

组合更改 158 事务处理 Adaptive Server Anywhere 特点 26 数据恢复 160 事务日志 简介 43 术语定义 183 事务日志镜像 术语定义 183 视图 SYSCATALOG 167 SYSCOLUMNS 168 术语定义 183 首选项 Interactive SQL 91 授权选项 术语定义 183 输入 Interactive SQL 命令 85 在 Interactive SQL 中输入多个语句 86 属性 表 11 数据定义语句 自动提交 158 数据恢复 事务 160 数据库 **SQL** 15 查询 16 创建 53 对象 13 服务器 18 关系 10 客户端应用程序 18 设计 45 设计概念 53 术语定义 184 文件 43

系统表 17 语言接口 18 组件 18 数据库大小 数十亿字节数据库 25 数据库对象 关于 13 术语定义 184 数据库服务器 个人服务器和网络服务器之间的差别 29 快速入门2 连接到 65 内部 42 术语定义 184 运行 65 数据库管理员 术语定义 184 数据库和应用程序9 数据库连接 术语定义 184 数据库名称 术语定义 184 数据库所有者 术语定义 184 数据库文件 简介 43 术语定义 185 数据库应用程序 体系结构 35 数据库应用程序的体系结构 35 数据类型 关于 10 术语定义 185 选择 59 数据源 简介 67 搜索条件 GROUP BY 子句 140

介绍 115 快捷方式 120 模式匹配 118 日期比较 116 子查询 145 索引 介绍 113 术语定义 185 锁定 术语定义 185

Т

添加 行 155 在 Interactive SQL 中添加新行 82 条件 GROUP BY 子句 140 模式匹配 118 搜索 115, 120 通配符 模式匹配 118 通信协议 Adaptive Server Anywhere 37 同步 术语定义 185 统一数据库 术语定义 185 投影 已定义 16 图标 用于手册 x

W

外表 术语定义 185 外部登录

术语定义 185 外部引用 定义的 150 外键 插入 161 关于 12 术语定义 186 已定义 11 外键创建向导 使用 63 外键约束 术语定义 186 外连接 介绍 133 术语定义 186 完成事务 158 完全备份 术语定义 186 完整性 检查 161 术语定义 186 网络服务器 关于 29 平台支持 29 术语定义 186 网络软件要求 28 唯一约束 术语定义 186 位图 存储为 Blob 59 谓词 介绍 119 术语定义 186 文档 SQL Anywhere Studio vii 约定 ix

Х

系统表 SYSTABLE 169 介绍 166 术语定义 186 已定义 17 系统错误 事务 160 系统对象 术语定义 187 系统视图 SYSCATALOG 167 SYSCOLUMNS 168 术语定义 187 系统要求 Adaptive Server Anywhere 27 限制 已定义 16 相关名 术语定义 187 相关子查询 定义的 150 项目 术语定义 187 向导 创建数据库 53 外键创建 63 消息类型 术语定义 187 消息系统 术语定义 187 校验 术语定义 181 写文件 术语定义 187 卸载 术语定义 187 新闻组

技术支持 xii 行 从表中选择 115 关于 10 使用 Interactive SOL 删除 83 使用 Interactive SQL 添加 82 添加 155 在 Interactive SQL 中编辑值 82 在 Interactive SQL 中插入 82 在 Interactive SQL 中复制 84 行级触发器 术语定义 180 行级锁定 Adaptive Server Anywhere 特点 26 性能监控 Adaptive Server Anywhere 特点 26 性能统计 术语定义 187 选项 Interactive SQL 91 选项对话框 配置 Interactive SQL 91 选择 已定义 16 选择集合数据 135 选择列表 计算列 109 列名称 109

Y

压缩的数据库文件 术语定义 187 要求 Adaptive Server Anywhere 27 业务规则 术语定义 188 一对一关系
定义 61,62 移动计算 要求 25 疑难解答 GROUP BY 子句 138 结果集可能会更改 113 子查询 149 自然连接 131 硬件要求 Adaptive Server Anywhere 特点 26 用户ID 新数据库 54 用户定义数据类型 术语定义 188 游标 术语定义 188 在 Interactive SQL 中指定计划类型 95 游标结果集 术语定义 188 游标位置 术语定义 188 语句级触发器 术语定义 188 预订 术语定义 188 元数据 术语定义 188 系统表 17 元组 11 远程 DBA 权限 术语定义 189 远程权限 术语定义 189 远程数据库 术语定义 189 远程用户 术语定义 189 约定 文档 ix

约束 术语定义 189 运行 Interactive SQL 命令 85 SQL 脚本 97, 98 命令文件 97

Ζ

在 Interactive SQL 中使用 SQL 语句 85 增量备份 术语定义 190 争用 术语定义 190 支持 新闻组 xii 支持的平台 Adaptive Server Anywhere 27 首通 术语定义 190 执行 Interactive SQL 中的命令 85 多次执行查询 113 佰 在 Interactive SQL 中编辑 82 重定向器 术语定义 178 主表 术语定义 190 主键 11 术语定义 190 主键约束 术语定义 190 转义语法 Interactive SQL 100 装载 Interactive SQL 中的命令 98 子查询 比较 148

或连接 150 介绍 145 术语定义 190 相关子查询 150 疑难解答 149 自动提交 ALTER 语句 158 COMMENT 语句 158 DROP 语句 158 数据定义语句 158 自然连接 错误 131 介绍 131 字母顺序 ORDER BY 子句 111 组合 在 Interactive SQL 中组合多个语句 86 组件 可用性 32