# Sybase数据库简介

## Sybase数据库的特点

### （1）它是基于客户/服务器体系结构的数据库

一般的关系数据库都是基于主/从式的模型。在主/从式的结构中，所有的应用都运行在一台机器上。用户只是通过终端发命令或简单地查看应用运行的结果。

而在客户/服务器结构中，应用被分在了多台机器上运行。一台机器是另一个系统的客户，或是另外一些机器的服务器。这些机器通过局域网或广域网联接起来。

客户/服务器模型的好处是：

● 它支持共享资源且在多台设备间平衡负载。

● 允许容纳多个主机的环境，充分利用了企业已有的各种系统。

### （2）它是真正开放的数据库

由于采用了客户/服务器结构，应用被分在了多台机器上运行。更进一步，运行在客户端的应用不必是Sybase公司的产品。对于一般的关系数据库，为了让其它语言编写的应用能够访问数据库，提供了预编译。Sybase数据库，不只是简单地提供了预编译，而且公开了应用程序接口DB-LIB，鼓励第三方编写DB-LIB接口。由于开放的客户DB-LIB允许在不同的平台使用完全相同的调用，因而使得访问DB-LIB的应用程序很容易从一个平台向另一个平台移植。

### （3）它是一种高性能的数据库

Sybase真正吸引人的地方还是它的高性能。体现在以下几方面：

#### 可编程数据库

通过提供存储过程，创建了一个可编程数据库。存储过程允许用户编写自己的数据库子例程。这些子例程是经过预编译的，因此不必为每次调用都进行编译、优化、生成查询规划，因而查询速度要快得多。

#### 事件驱动的触发器

触发器是一种特殊的存储过程。通过触发器可以启动另一个存储过程，从而确保数据库的完整性。

#### 多线索化

Sybase数据库的体系结构的另一个创新之处就是多线索化。一般的数据库都依靠操作系统来管理与数据库的连接。当有多个用户连接时，系统的性能会大幅度下降。Sybase数据库不让操作系统来管理进程，把与数据库的连接当作自己的一部分来管理。此外，Sybase的数据库引擎还代替操作系统来管理一部分硬件资源，如端口、内存、硬盘，绕过了操作系统这一环节，提高了性能。

## Sybase数据库的组成

Sybase数据库主要由三部分组成：

（1）进行数据库管理和维护的一个联机的关系数据库管理系统Sybase SQL Server，Sybase SQL Server是个可编程的数据库管理系统，它是整个Sybase产品的核心软件，起着数据管理、高速缓冲管理、事务管理的作用。

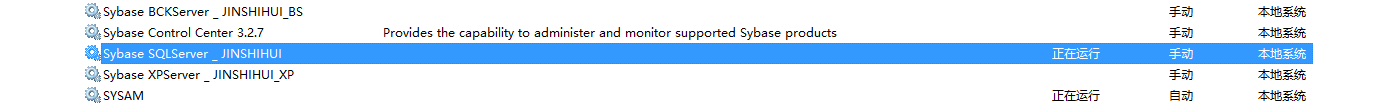
(2) 支持数据库应用系统的建立与开发的一组前端工具Sybase SQL Toolset，ISQL是与SQL Server进行交互的一种SQL句法分析器。ISQL接收用户发出的SQL语言，将其发送给SQL Server，并将结果以形式化的方式显示在用户的标准输出上。DWB是数据工作台，是Sybase SQL Toolset的一个主要组成部分，它的作用在于使用户能够设置和管理SQL Server上的数据库，并且为用户提供一种对数据库的信息执行添加、更新和检索等操作的简便方法。在DWB中能完成ISQL的所有功能，且由于DWB是基于窗口和菜单的，因此操作比ISQL简单，是一种方便实用的数据库管理工具。APT是Sybase客户软件部分的主要产品之一，也是从事实际应用开发的主要环境。APT工作台是用于建立应用程序的工具集，可以创建从非常简单到非常复杂的应用程序，它主要用于开发基于表格（Form）的应用。其用户界面采用窗口和菜单驱动方式，通过一系列的选择完成表格（Form）、菜单和处理的开发。

(3) 可把异构环境下其它厂商的应用软件和任何类型的数据连接在一起的接口Sybase Open Client/Open Server。通过Open Client的DB-LIB库，应用程序可以访问SQL Server。而通过Open Server的SERVER-LIB，应用程序可以访问其它的数据库管理系统。

# sybase数据库服务

需要启动如下服务才能连接sybase数据库

Sybase SQLServer \_JINSHIHUI



# Sybase基础知识

Sybase SQL Server是一个多库数据库系统，这些数据库包括系统数据库和用户数据库，而不论是系统数据库还是用户数据库，都建立在数据库设备上。

## 数据库设备

所有的数据库都创建在数据库设备上。所谓数据库设备，不是指一个可识别的物理设备，而是指用于存储数据库和数据库对象的磁盘原始分区或操作系统文件。增加一个新的数据库设备时，必须对这些设备“初始化”。初始化的过程就是将物理磁盘、磁盘分区或操作系统文件变为SYBASE数据库可以识别的设备。初始化数据库设备使用DISK INIT命令：

DISK INIT

NAME=设备名，

PHYSNAME=物理设备名，

VDEVNO=虚拟设备号，

SIZE=块数

其中，NAME指数据库设备名，此名将用于CREATE DATABASE和ALTER DATABASE命令。PHYSNAME是原始磁盘分区或操作系统文件名。VDEVNO是数据库设备的标识号，在SQL Server中，它必须是唯一的。SIZE的单位是2K的块，对于新创建的数据库，最小的SIZE是model数据库的尺寸，即1024个2K的块（2M）。

例如：/\*将/dev目录下400M的物理设备初始化为SYBASE的数据库设备tele114\_log01\*/

1>disk init

2>name=“tele114\_log01”,

3>physname=“/dev/rtelelog”

4>vdevno=4

5>size=204800 /\*2Kbyte\*204800=400Mbyte\*/

6>go

## 段

数据库设备从逻辑上被划分为数据库段以允许将某一特定的对象放置在指定的段上（创建对象时指定），一数据库设备可拥有多达192个段，一段可使用255个逻辑设备上的存储空间。当用户创建一个数据库时，SQL SERVER 会自动在该数据库中创建三个段： SYSTEM、LOGSEGMENT 、DEFAULT，这三个段分别用来存储数据库的系统表、事务日志和其他数据库对象。

在数据库中创建段的步骤是：

先用Disk init 初始化物理设备

通过使用Create database 和alter database 的on 子句，使数据库设备对数据库可用，于是新设备自动增加到数据库的default 和system段。

一旦数据库设备存在并对数据库可用，使用存储过程Sp\_addsegment 定义数据库的段。语法如下：

Sp\_addsegment 段名，数据库名，设备名

## 系统数据库

安装Sybase数据库时自动生成下列系统数据库：

● 主数据库master；

● 模型数据库model；

● 系统过程数据库sybsystemprocs；

● 临时数据库tempdb

也可选择下列数据库：

● 安全审核数据库sybsecurity；

● 示例数据库pubs2；

● 命令语法数据库sybsyntax。

### （1） master数据库

包含许多系统表和系统过程，从总体上控制用户数据库和SQL Server的操作，构成了SYBASE系统的数据字典。MASTER数据库主要记录信息为：

● 登录帐号（SQL服务器用户名）；syslogins,sysremolelogins

● 正在执行的过程；sysprocesses

● 可修改的环境变量；sysconfigures

● 系统错误信息；sysmessages

● SQL服务器上每个数据库的信息；sysdatabases

● 每个数据库占用的存储空间；sysusages

● 系统上安装的磁带及磁带信息；sysdevices

● 活动锁信息；syslocks

### （2） model数据库

提供了新用户数据库的初型。当我们每次用CREATE DATABASE命令 时，SQL Server都产生一个model数据库的拷贝，然后把它扩展到命令要求的尺寸。如果我们修改了model数据库，那么以后新创建的数据库都将随它而改变。下面是几个通常用到的对model数据库的改变：

● 增加用户数据库类型、规则或缺省；

● 可存取SQL Server上所有数据库的用户用sp\_adduser增加到model数据库上；

● 缺省权限（特别是guest帐号）可在model中建立；

● 诸如select into/bulkcopy那样的数据库选项可在model中设置。这些设置反映到所有新创建的数据库中。它们在model中最原始的值是关（off）。

在一般情况下，大多数用户无权修改model数据库，又没被授权读取。因为model中所有内容已拷贝到新数据库中，授权读model没什么意义。

use model

go

sp\_help

go

Name Owner Object\_type

--------------- --------------- ----------------------

sysalternates dbo system table

syscolumns dbo system table

syscomments dbo system table

sysconstraints dbo system table

sysdepends dbo system table

sysgams dbo system table

sysindexes dbo system table

syskeys dbo system table

syslogs dbo system table

sysobjects dbo system table

sysprocedures dbo system table

sysprotects dbo system table

sysreferences dbo system table

sysroles dbo system table

syssegments dbo system table

systhresholds dbo system table

systypes dbo system table

sysusermessages dbo system table

sysusers dbo system table

### （3） sybsystemprocs数据库

数据库中存储系统过程，当用户执行一存储过程（以sp\_开始）时，SQL 服务器首先在用户当前数据库中搜索指定的存储过程，如果没有相应过程，则到系统过程数据库中寻找，若没有，再到MASTER数据库中寻找。

### （4）临时数据库tempdb

该数据库为临时表和临时工作提供了一个存储区域。临时数据库是SQL Server上所有数据库共享的工作空间。由于这些表都是临时的，所以每当用户退出或系统失败，这些表都将消失。

### （5）安全审核数据库sybsecurity

数据库中保存系统的安全审核信息，它可跟踪记录每个用户的操作情况，为维护系统安全提供控制手段。

### （6）示例数据库pubs2

为用户提供一个学习SYBASE的数据库模型。

### （7）命令语法数据库sybsyntax

库中保存SYBASE数据库的命令语法及支持的语言库。

## 用户数据库

用户数据库是指用Create database 命令创建的数据库。所有新的用户数据库都要在master数据库中创建，也就是说不能存取master数据库的用户是无权创建新的数据库的。SA可以将创建数据库的权限授予其他用户。新建数据库中存在一些系统表，在sysusers表中至少有一条记录，即该数据库的创建者。数据库创建时，创建者既为该数据库的 owner, 当然创建者可以将这一地位或这一所有权用系统过程授予别的用户。

## 数据类型

|  |  |  |
| --- | --- | --- |
| 字段类型 | 描述 | 字段长度及其缺省值 |
| char(size) | 用于保存定长字节的字符串数据 | 每行定长（不足部分补为空格），最大长度为255字节，缺省值为每行1字节 |
| varchar(size) | 用于保存变长的字符串数据，其中最大字节长度由size指定 | 每行长度可变，最大长度为255字节，数据长度缺省为1 |
| Nchar(Size) | 多字节字符串。 | 每行定长（不足部分补为空格）。 长度(Size）是指字符的个数，最大字节数为Size\*A@@Nchar Size。 |
| Nvarchar(Size) | 可变长，多字节字符串。 | 每行变长。长度(Size）是指字符的个数，最大字节数为Size\*A@@Nchar Size。 |
| Timestamp | 用于改变管理的二进制值。 | 字节数为16 |
| Datetime | 保存定长的日期时间数据，范围为1753年1月1日到公元9999年12月21日，精度为3微秒间隔。 | 每行固定为8字节长度 |
| Numeric(P,S) | 保存精确数值数据类型，范围为-10E-38到+10E-38,P是精度,总小数位数(1~38),S是小数位(小数位小于等于P) | 最大长度为2~17字节 |

## 数据库对象（重点）

### （1）表(Tables)

在一个关系数据库中，表是最重要的数据库对象，一切数据都存放在表中。其它数据库对象都是为了用户很好地操作表中的数据。表都以一张二维表的形式存在。其中每一列称之为一个字段，或一个域；而每一行称之为一个记录，也就是一项数据。

有一类表，它们的名字都是以sys开头的，这些表称为系统表，系统表记录着有关SQL Server的信息。在master数据库中的表大部分为系统表，这些表是随着master数据库的创建而建立的。另外每个用户数据库在创建时都有一个系统表的子集。

例如，只有在master数据库中才有的系统表有：sysconfigures、sysdatabases、sysdevices、syslogins等；在用户数据库和系统数据库中都有的系统表有：sysalternates、syscolumns、sysindexs、syslogs、sysobjects、sysusers等。

### （2）视图(Views)

视图是查看一张或几张表中数据的一种方式。通过将一张或几张表中的一部分数据进行组合得到视图。视图看上去与表非常相象，但与表还是有着本质的区别。通过视图看到的数据实际上都是存放在表中的，在数据库中仅存在视图的定义。

使用视图的好处是：

1、操作方便

2、安全

### （3）索引(Index)

索引是对字段生成的，用于加快数据的检索。

在Sybase数据库中，索引是由系统自动管理的，也就是说Sybase的索引操作对用户是透明的。表的索引一旦建立，系统会自动对其进行更新维护以使它与相应表对应一致，操作时用户无须指定索引，系统会自动确定是否使用索引进行操作。

Sybase支持三种不同类型的索引，一是复合索引，即包含几个列的索引；二是唯一性索引，即任意两行中的索引列不能有相同的取值；三是聚集索引（Clustered indexes）和非聚集索引(Nonclustered indexs)，聚集索引使SQL服务器不断重排表行的物理顺序以使其与逻辑索引顺序相一致，非聚集索引则不需要表行的物理顺序与逻辑顺序一致。

每个表最多只能建立一个聚集索引，非聚集索引则可以建立多个，与非聚集索引相比，聚集索引的检索速度要快，应当在建立任何非聚集索引以前建立聚集索引。

### （4）存储过程(Stored procedures)

存储过程是用T-SQL语言编写成的SQL子例程，它存储于SQL服务器上供用户调用执行。与一般的SQL语句和批处理语句不同的是，存储过程是经过预编译的。当首次运行一个存储过程时，SQL Server的查询处理器将对其分析，并产生最终的执行方案。由于查询处理的大部分工作已经完成，所以以后执行存储过程时速度将会很快。执行存储过程时可带参数并可调用其他存储过程，执行完毕后返回信息以指示是否成功完成相应操作。存储过程有两种：一种是SQL服务器安装时自动建立的系统存储过程（系统过程），另一种是用户自己创建的存储过程。

系统过程是用于系统管理，并且为用户提供了从事数据库管理的一种途径。这些系统过程都是以sp\_开头的，它们都放在master数据库中且隶属于sa(系统管理员) ，也有很多可以在任一个数据库中运行的系统过程。

#### 常见的系统存储过程

|  |  |
| --- | --- |
| Sp\_addgroup | 在当前数据库中建立一个数据库用户组 |
| Sp\_addlogin | 建立一个SQL服务器用户 |
| Sp\_adduser | 在当前数据库中增加一个用户 |
| sp\_changegroup | 改变数据库用户组 |
| Sp\_dboption | 查询或改变数据库系统设置 |
| Sp\_dropdevice | 删除设备 |
| Sp\_dropgroup | 删除组 |
| Sp\_droplogin | 删除帐号 |
| Sp\_help | 查询数据库对象及所有数据库信息 |
| Sp\_helpdb | 查询数据库信息 |
| Sp\_helpdevice | 查询设备信息 |
| Sp\_helpgroup | 查询组信息 |
| Sp\_helpindex | 查询给定表信息 |
| Sp\_helpuser | 查询用户信息 |
| Sp\_lock | 查询当前加锁信息 |
| Sp\_monitor | 查询SQL服务器统计信息 |
| Sp\_password | 改变登录帐号口令 |
| Sp\_spaceused | 查询表中的行数、数据页数及空间大小 |
| Sp\_who | 查询当前用户及过程信息 |
| Sp\_syntax | 查询操作语法 |
| Sp\_configure | 查询配置系统参数 |

### （5）触发器(Triggers)

触发器是一种特殊的存储过程，用来维护不同表中的相关数据的一致性。当在一张表中插入、删除和修改数据时，触发器就会触发另一个存储过程，从而保持数据的一致性。

### （6）缺省与规则(Defaults and rules)

缺省是在数据录入时，若用户没有输入数据，SQL Server自动输入的值。规则可以理解为对数据库、某一列、某用户数据类型的限制。

# 数据库对象（重点）

## 数据库

### 打开数据库

Use tele114

### 创建数据库

拥有创建数据库权利的用户可以创建自己的数据库。

CREATE DATABASE test

### 删除数据库

Drop database test

### 修改数据库

Alter database tele114 on tele114\_run02=100, tele114\_idx02=100

## 表

### 建表

create table spec\_code

(

No tinyint not null, /\*特编号(0--99)\*/

Name varchar(50) null, /\*名称\*/

Addr varchar(50) null, /\*地址\*/

Tel varchar(8) null, /\*电话号码\*/

StaffNo varchar(4) not null /\*录入员工号\*/

)

on segrun

/\*将表放在segrun段上\*/

### 修改表

#### 在表中增加新的列

(用这个命令增加的列必须允许null值)

Alter table friends\_etc add country varchar(20 ) null

#### 指定默认值

create table ny\_titles

(title\_id char(6t),

Title varchar(80),

Price money default null,

Total\_sales int default (0)

)

#### 指定Unique 和primary key 约束

Primary key 不允许空值，常用来产生唯一的聚集索引，unique 允许空值，常用来产生唯一的非聚集索引。

Create table my\_sales

(stor\_id char(4),

Ord\_num varchar(20),

Date datetime,

Unique clustered(stor\_id,ord\_num)

)

#### 将表中的某个字段设置为主键

ALTER TABLE 表名 ADD CONSTRAINT 约束名 PRIMARY KEY(列名)

示例：

ALTER TABLE [tbl\_weblead\_input] ADD CONSTRAINT pk\_txt\_sequence PRIMARY KEY(txt\_sequence)

### 如何锁表

select \* from tablename holdlock

### Sybase中的锁及死锁问题

一般来说，数据库都会有两种锁：内存锁和对象锁。Oracle中有latch和lock，sybase中有spinlock和lock。内存锁实际上就是数据库系统将自己管理的内存区按单元加锁，以防止一个任务在使用时被另一个任务修改。用完这个内存单元后，内存锁被立即释放。我们只会论及sybase数据库的对象锁lock。

#### 锁类型

分为排他锁、共享锁和更新锁

### 删除表

drop tabel 表名

## 索引

### 建立索引

create unique clustered index code1thidx1 on code\_1th(Code1,Name,UnitNo1,Used)

on segrun

Unique: 建立唯一性索引

Clustered : 建立聚集索引，使得数据行的逻辑顺序与物理顺序一致，查询速度较快，每个表仅允许建立一个唯一性聚集索引。

On segrun:将索引放在segrun上

### 删除索引

drop index 表名.索引名

## 默认

当用户没有显示地为某一列赋值时，SQL SERVER 为该列插入的值

创建默认值CA，把默认值CA用到friends\_etc表中的State列上

Creat default statedflt

Go

Sp\_bindefault statedflt, "frends\_etc.state"

go

### 断离默认

Sp\_unbindefault "frends\_etc.state"

### 删除默认

Drop default statedflt

## 规则

可以理解为对数据库、某一列、某用户数据类型的限制.规则中的任何变量名必须冠以@，不能直接引用列或其他数据库对象。

创建规则并把他关联到用户定义的数据类型P#上

Create rule phonerule as @phone like 415[0-9][0-9][0-9][0-9]

Sp\_bindrule phonerule ,"p#"

### 断离规则

Sp\_unbindrule "p#"

### 删除规则

Drop rule phonerule

## 视图

视图是从一个或多个基本表或其他视图上导出的表,可以在视图上进行增、删、改

### 创建视图

创建视图的句法为：

create view view\_name

(column\_1,...,column\_n)

as select\_statements

例：在authors表和publishers表上定义一个视图如下：

Create view cities (authorname, acity, publishername, pcity)

As select au\_lname,authors.city, pub\_name, publishers.city

From authors, publishers

### 删除视图

通过使用Drop view 命令来撤消一个视图

Drop view view\_name

## 存储过程（重点）

### 创建存储过程

CREATE PROCEDURE procedure\_name [(para1[,para2, ...])]

As sql\_statements

存储过程的参数名必须以@开头，包括@最长为30个字符。

### 执行存储过程

EXEC[UTE] procedure\_name [(para1[,para2, ...])]

示例：

declare @rOutput CHAR(1)

EXEC db\_cus..po\_ec\_VerifyVoucherCust 'S1484767G', @rOutput OUTPUT

select @rOutput

说明：不能加圆括号，否则会报错的。

存储过程能够返回一个整数，用来指示存储过程执行是完全错误还是错误。SQL SERVER 用0指示成功的返回，-1~-99指示不同的错误。当前所用的0及-1~-14的含义如表所示。

|  |  |
| --- | --- |
| 值 | 含义 |
| 0 | 过程执行无错 |
| -1 | 丢失对象 |
| -2 | 数据类型错误 |
| -3 | 进程被选作死锁的牺牲品 |
| -4 | 权限错误 |
| -5 | 语法错误 |
| -6 | 各种用户错误 |
| -7 | 资源错误：如空间错误 |
| -8 | 非严重的内部错误 |
| -9 | 触及系统限制 |
| -10 | 严重的内部不一致 |
| -11 | 严重的内部不一致 |
| -12 | 表或索引坏了 |
| -13 | 数据库坏了 |
| -14 | 硬件错误 |

### 更改存储过程名

Sp\_rename objname newname

### 删除存储过程

DROP PROCEDURE procedure\_name

## 触发器

### 创建触发器

CREATE TRIGGER [owner.]trigger\_name

ON [owner.]table\_name

{FOR INSERT,UPDATE,DELETE}

AS

SQL\_statements

或使用IF UPDATE 子句：

CREATE TRIGGER [owner.]trigger\_name

ON [owner.]table\_name

{FOR INSERT,UPDATE}

AS

[IF UPDATE (column\_name)

[{AND|OR} UPDATE (column\_name)]...}

SQL\_statments

[IF UPDATE(column\_name)

[{AND|OR} UPDATE column\_name)]...

SQL\_statements]...

### 删除触发器

DROP TRIGGER [owner.]trigger\_name

# SQL 查询语言（重点）

SQL不仅包括查询数据的有关命令，还包括创建数据库及其对象增、删、改数据等功能。分别定义为数据查询语言，数据定义语言及数据操作语言。这里先介绍数据查询语言。其基本句法为：

Select select\_list from table\_list where search\_conditions

## 简单查询

### 选择若干列

Select expression [,expression]... From table\_list

### 选择若干行

查出电话号码以415到头的记录

Select \* from code\_1th where tel like '415%'

### 查询中消除重复的行

Select distinct tel from code\_1th

对text 和 char 可用 like ,其中可用通配符‘%’及‘-’，分别代表多个字符和单个字符。

### 其他常用查询条件有：(出text类型除外)

|  |  |
| --- | --- |
| 大小比较 | =、>、<、>=、<=、!=、!>、!< |
| 范围确定 | between exp1 and exp2 not between exp1 and exp2 |
| 列表或集合 | in (exp1[,exp2,[...]]) not in (...) |
| 谓词 | like |
| 多重条件 | and or not |

## 连接查询

### 等值连接和不等值连接

通过‘=’来比较两个表之间的数据时，称为等值连接；而通过其他比较符时，称为不等值连接

等值连接：

Select \* from publishers, authors where publishers.city=authors.city

不等值连接：

### 自然连接

在连接的目标列中相同名的列只保留一个

Select publishers.pub\_id publishers.pub\_name, publishers.state, authors.\*

From publishers, authors where publishers.city=authors.city

## 子查询

### 表达式子查询

Select au\_lname, au\_fname from authors where city=

(select city from publishers where pub\_name="abcde")

可以使用一切大小比较操作符；在操作符和子查询之间可以使用All 或any。

### 定谓词子查询

Select pub\_name from publishers from publishers

Where pub\_id in

(select pub\_id from titles where type='abcde')

## 相关查询

相关查询即嵌套查询依赖于外部父查询的值，嵌套查询要重复执行若干次。

Select distinct t1.type from titles t1

Where t1.type in ( select t2.type from titles t2 where t1.pub\_id!=t2.pub\_id)

## 函数、分组与排序

### 对查询结果进行聚集处理

聚集函数： Sum([all|distinct] expression),avg([all|distinct] exoression) ,

Count([all|distinct]expression), count(\*), max(expression), min(expression)

Select count(\*) from titles

### 用Group by 和 having 子句对查询结果分组

Select type ,avg(advance), sum(total\_sales) from titles group by type

Select type from titles group by type having count(\*) >1

Having 类似于where , 但where 不能用聚集函数。

### 用Order by 对查询结果进行排序

Select type ,avg(price) from titles group by type order by avg(price)

### Compute 子句

完成基于每一组中的值的聚集运算，聚集值作为一个新行出现在查询结果中。

Select type ,price advance from titles order by type compute sum(price), sum(advance) by type

# 常用sql语句

## 查看数据库的版本

select @@version

## 查看数据库启动的参数

sp\_configure

## 查看数据库的信息

select \* from master..sysprocesses

查看数据库服务器名和实例名

print 'Server Name...............：' + convert(varchar(30),@@SERVERNAME)

print 'Instance..................：' + convert(varchar(30),@@SERVICENAME)

## 查看数据库的用户

sp\_helpuser

## 查看所有数据库名称及大小

sp\_helpdb

## 创建数据库

create database 数据库名

## 重命名数据库

sp\_renamedb 'old\_dbname', 'new\_dbname'

## 查看某数据库下对象及用户权限

sp\_helprotect

## 查看远端数据库用户登录信息

sp\_helpremotelogin

## 查看数据库下某个数据对象的大小

sp\_spaceused @objname（对象名）

还可以用sp\_toptables过程看最大的N(默认为50)个表

示例：

sp\_spaceused 'tbl\_mainsite\_familyfest'

## 查看数据库下某个数据对象的索引信息

sp\_helpindex @objname

clustered索引是把记录按物理顺序排列的，索引占的空间比较少。对键值DML操作十分频繁的表我建议用非clustered索引和约束，fillfactor参数都用默认值。

## 查看数据库下某个数据对象的的约束信息

sp\_helpconstraint @objname

## 查看数据库里所有的存储过程（重点）

use @database\_name

sp\_stored\_procedures

## 查看存储过程和函数的源代码（重点）

sp\_helptext '@procedure\_name'

## 执行存储过程（重点）

exec [procedure\_name] [参数]

示例：

exec db\_ecm..sp\_verifyBirthInfo 'GMAINSED KMED','S7483841C','12/02/1974'

## 查看包含某个字符串@str的数据对象名称

select distinct object\_name(id) from syscomments where text like '%@str%'

创建加密的存储过程或函数在AS前面加WITH ENCRYPTION参数

解密加密过的存储过程和函数可以用sp\_decrypt过程

## 查看数据库里用户和进程的信息

sp\_who

## 查看数据库里锁的情况

sp\_lock

进程号1--50是SQL Server系统内部用的,进程号大于50的才是用户的连接进程.

spid是进程编号

查看进程正在执行的SQL语句

dbcc inputbuffer()

推荐大家用经过改进后的sp\_who3过程可以直接看到进程运行的SQL语句

sp\_who3

检查死锁用sp\_who\_lock过程

sp\_who\_lock

## 查看和收缩数据库日志文件的方法

查看所有数据库日志文件大小

dbcc sqlperf(logspace)

如果某些日志文件较大，收缩简单恢复模式数据库日志，收缩后@database\_name\_log的大小单位为M

backup log @database\_name with no\_log

dbcc shrinkfile (@database\_name\_log, 5)

## 设置显示行的总数

set rowcount 1000

## 分析SQL 语句的方法

set statistics time {on | off}

set statistics io {on | off}

### 图形方式显示查询执行计划

在查询分析器->查询->显示估计的评估计划(D)-Ctrl-L 或者点击工具栏里的图形

### 文本方式显示查询执行计划

set showplan\_all {on | off}

set showplan\_text { on | off }

set statistics profile { on | off }

# 复制表结构和数据到新的表

select \* into tbl\_weblead\_input\_20180814 from tbl\_weblead\_input

说明：红色的是新表，蓝色的是原有的表

# 批量向表中插入数据

1、创建存储过程

2、执行存储过程

示例：

--删除存储过程

IF EXISTS (SELECT id FROM sysobjects WHERE type='P' AND name='INSERT\_USERS')

DROP PROCEDURE INSERT\_USERS

--创建存储过程

CREATE PROCEDURE INSERT\_USERS

@number INTEGER

as

DECLARE

@id INTEGER, --循环变量

@p1\_nric VARCHAR(16), --p1\_nric

@l\_UUID VARCHAR(32)

begin

SELECT @id = 0

while @id<@number

begin

SELECT @l\_UUID = NEWID()

select @p1\_nric= @l\_UUID

INSERT INTO dbo.tbl\_mainsite\_familyfest (id, FormID, p1\_name, p1\_nric, p1\_mobileNo, p1\_email, p1\_relationship, p1\_isAIA\_user, p2\_name, p2\_nric, p2\_mobileNo, p2\_email, p2\_relationship, p2\_isAIA\_user, childrenNo\_Four, childrenNo\_Twelve, hear\_method, others, uniqueURL, created\_time, created\_by, updated\_time, updated\_by, txt\_buffer\_val1, txt\_buffer\_val2, txt\_buffer\_val3, txt\_buffer\_val4, txt\_buffer\_val5)

VALUES (@l\_UUID,'aiaff2431794358','APD Test 2',@p1\_nric,'+658123456','iehowe@apdgroup.com','father','N','Wife', 'S123456T','','','mother','N','1','1','Website','Others - Please specify','dyRWgGz-aiaff2431794358-APD-Test-2',getdate(),'system',getdate(),'system','','','','','')

SELECT @id = @id + 1

END

END

执行存储过程：

DECLARE @number INTEGER

EXEC INSERT\_USERS 100000 --插入数据量，可修改

GO

# 更新数据记录

## 为表增加一个字段

语法：

alter table 表名 add 列名 数据类型 null

示例：

ALTER TABLE tbl\_mainsite\_familyfest ADD check\_flag VARCHAR(2) NULL

## 修改表中某个字段的名称

语法：

sp\_rename “表名.列名”,新的列名

示例：

sp\_rename "tbl\_mainsite\_familyfest.check\_flag",mkt\_consent\_agreement\_flag

## 修改表中某个字段的数据类型

语法：

alter table 表名 modify 列名 新的数据类型

示例：

ALTER TABLE tbl\_mainsite\_familyfest modify mkt\_consent\_agreement\_flag VARCHAR(10)

说明：这个语法适用于修改字段的数据类型长度。

## 将表中某个字段设置为主键

alter table 表名 add constraint 约束名 primary key(列名)

ALTER TABLE tbl\_weblead\_input ADD CONSTRAINT pk\_txt\_sequence PRIMARY KEY(txt\_sequence)

## 删除表中的主键

ALTER TABLE 表名 DROP CONSTRAINT 主键约束名

示例：

ALTER TABLE tbl\_mainsite\_familyfest DROP CONSTRAINT tbl\_mainsi\_9794559551

## 给表设置联合主键

CREATE TABLE test\_jin (

ID int not null,

NAME varchar(50) not null,

location varchar(200) null

)

alter table test\_jin add constraint pk\_test\_jin primary key (ID,NAME)

或者

create unique clustered index pk\_test\_jin on test(ID,NAME)

go

# 统计函数

## AVG(字段名)

得出一个表格栏平均值

## COUNT(\*|字段名)

对数据行数的统计或对某一栏有值的数据行数统计

## MAX(字段名)

取得一个表格栏最大的值

## MIN(字段名)

取得一个表格栏最小的值

## SUM(字段名)

把数据栏的值相加

引用以上函数的方法：

sql="select sum(字段名) as 别名 from 数据表 where 条件表达式"

set rs=conn.excute(sql)

用 rs("别名") 获取统的计值，其它函数运用同上。

# 常用的函数（重点）

## 字符串函数

### datalength(char\_expr)

在char\_expr中返回字符的长度值，忽略尾空

### substring(expression,start,length)

返回部分字符串

### right(char\_expr,int\_expr)

返回char\_expr右边的int\_expr字符

### upper(char\_expr)

把char\_expr转换成大写形式

### lower(char\_expr)

把char\_expr转换成小写形式

### space(int\_expr)

生成有int\_expr个空格的字符串

### replicate(char\_expr,int\_expr)

重复char\_expr,int\_expr次

### stuff(expr1,start,length,expr2)

用expr2代替epxr1中start起始长为length的字符串

### reverse(char\_expr)

反写char\_expr中的文本

### ltrim(char\_expr)

删除头空

### rtrim(char\_expr)

删除尾空

### ascii(char\_expr)

返回char\_expr中第一个字符的ASCII值

### char(int\_expr)

把ASCII码转换为字符

### soundex(char\_expr)

返回char\_expr的soundex值

### difference(char\_expr1,char\_expr2)

返回表达式soundex值之差

### charindex(char\_expr,expression)

返回指定char\_expr的开始位置,否则为0

### patindex("%pattern%",expression)

返回指定样式的开始位置,否则为0

### escape子句

用某个转义字符可在搜索字符串时将通配符作为文字来包含。ANSI-89 SQL标准定义了escape子句指定某个转义字符，缺省情况下，[]来转义某个通配符，例：

select \* from test\_tab where description like "%20[%]%"

语法：

like char\_expression escape escape\_character

例

select \* from test\_tab

where description like "%20#%%" escape "#"

### + 可用于串接字符

select au\_laname+","+au\_fname from authors

## 数学函数

### abs(numeric\_expr)

返回指定值的绝对值

### ceiling(numeric\_expr)

返回大于或等于指定值的最小整数

### exp(float\_expr)

给出指定值的指数值

### floor(numeric\_expr)

返回小于或等于指定值的最大整数

### pi()

返回常数3.1415926

### power(numeric\_expr,power)

返回numeric\_expr的值给power的幂

### rand([int\_expr])

返回0-1之间的随机浮点数，可指定基值

### round(numeric\_expr,int\_expr)

把数值表达式圆整到int\_expr指定的精度

### sign(int\_expr)

返回正+1，零0或负-1

### sqrt(float\_expr)

返回指定值的平方根

## 日期函数

### getdate()

返回当前的系统日期和时间

### datename(datepart,date\_expr)

以字符串形式返回date\_expr指定部分的值,转换成合适的名字

### datepart(datepart,date\_expr)

作为整数返回date\_expr值的指定部分

### datediff(datepart,date\_expr1,date\_expr2)

返回date\_expr2-date\_expr1,通过指定的datepart度量

### dateadd(datepart,number,date\_expr)

返回日期,通过在date\_expr上增加指定number的日期部件而产生的

### datepart说明

日期部件 缩写 值范围

年 yy 1753-9999

季度 qq 1-4

月 mm 1-12

每年中的天 dy 1-366

天 dd 1-31

星期 wk 1-54

星期天 dw 1-7(1=sunday)

小时 hh 0-23

分钟 mi 0-59

秒 ss 0-59

毫秒 ms 0-999

例： select invoice\_no,

datediff(dd,date\_shipped,getdate())

from invoices

where balance\_due>0

## 转换函数

### convert(datatype [(length)],expression)

此函数把值从一种类型改变成另一种类型

select "Advance="+convert(char(12),advance) from titles

### 日期转换

convert(datetype[(length)],expression,format)

format指定将日期转换为什么格式，有以下值：

没有世纪 有世纪 转换字符串中日期格式

0 or 100 mon dd yyy hh:miAM(or PM)

1 101 mm/dd/yyyy

2 102 yyyy.mm.dd

3 103 dd/mm/yyyy

4 104 dd.mm.yy

5 105 dd-mm-yy

6 106 dd mon yy

7 107 mon dd,yy

8 108 hh:mi:ss

9 or 109 mon dd,yyyy hh:mi:ss:mmmAM(or PM)

10 110 mm-dd-yyyy

11 111 yyyy/mm/dd

12 112 yyyymmdd

例: convert(char(11),dt\_registdate ,111) + convert(char(5),dt\_registdate ,108)

显示结果: 2006/10/13 20:26

## 系统函数

### host\_id()

客户进程的当前主机进程ID号

### host\_name()

客户进程的当前主计算机名

suser\_id(["login\_name"])

用户的SQL Server ID号

suser\_name([server\_user\_id])

用户的SQL Server登录名

user\_id(["name\_in\_db"])

用户在数据库中的ID号

user\_name([user\_id])

用户在数据库中的名字

user

用户在数据库中的名字

show\_role()

用户的当前活动角色

数据库和对象信息

db\_id(["db\_name"]) 数据库ID号

db\_name([db\_id]) 数据库名

object\_id("objname") 数据库对象ID号

object\_name(obj\_id]) 数据库对象号

col\_name(obj\_id,col\_id) 对象的栏名

col\_length("objname","colname") 栏的长度

index\_col("objname",index\_id,key#) 已索引的栏名

valid\_name(char\_expr) 若char\_expr不是有效标识符，则返回0

数据函数

datalength(expression) 按字节返回expression的长度

tsequal(timestamp1,timestamp2) 比较时戳值，若时戳值不匹配，则返回出错消息

isnull()

isnull函数用指定的值代替查询栏或合计中的空值

例：

select avg(isnull(total\_order,$0))

from invoices

# 如何查看服务器端、客户端字符集

## （1）查看服务器端字符集

sp\_helpsort

go

## （2）查看客户端字符集

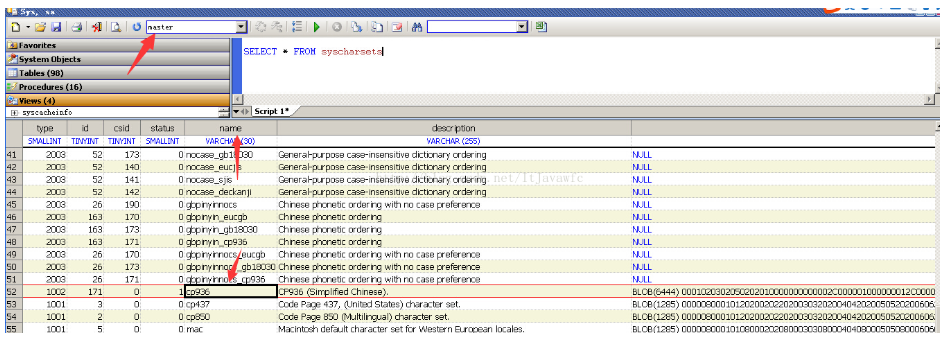
select @@client\_csname

go

Sybase 中文乱码是由于默认的编码集为ISO-1 ，在Sybase Central v4.3中中文显示为乱码，将字符集改成CP936后，中文乱码解决。

# 更改字符集方法

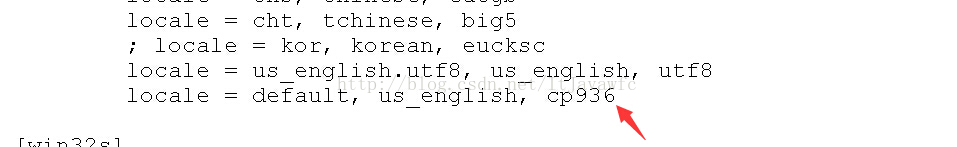
（1）打开Sybase Central 连接上数据库，打开master数据库的syscharsets表，查看name列里面有没有CP936数据，如果没有，进入步骤二；如果有，操作步骤三。



（2）运行 --> CMD 进入DOS窗口命令提示，进入目录 D:\sybase\charsets\cp936 （这是我的Sybase安装目录），输入命令 charset -Usa -P -S binary.srt cp936



（3）按步骤一，打开master数据库的syscharsets表，查看name列CP936对应的id应该是171，在左边数据库服务器名上点(我的服务器名为GZLCL)“右键-->属性  --> 服务器配置 ”在“显示配置参数”中选择 “Langguages” ，“default character set id"填写171 ，“确定” 然后重新启动服务。  
或者isql命令行：sp\_configure "default character set id",171

（4）配置locales.dat文件：文件路径为D:\sybase\locales\locales.dat  用记事本打开 找到 “[NT]”  将其默认字符集改为locale = default, us\_english, cp936   
 

这样在Sybase Central 里面中文乱码就可以解决了.

# 批处理和流程控制语言

## 批处理

SYBASE系统允许用户将若干条相关的语句组成一个整体，交给系统同时处理。一批语句的结束以go语句为标志。严格地讲单条SQL语句也是一个批处理。

例：Select count(\*) from titles

Select count(\*) from authors

Go

也可以将一组SQL命令放在一个文件中，再将这个文件交给sql执行。

## 流程控制语言

在批处理中，我们可以加入流程控制语句从而达到编程的目的。在T-SQL语言中，流控制语句有：

if…else…

begin…End…

while和break…Continue

declare和局部变量以及全局变量

Goto

Return

例：

if exists(select \* from zyl where xm="李军")

print "ok"

else

print "no this one"

### 使用批处理有以下规则

1、CREATE PROCEDURE、CREATE RULE、 CREATE DEFAULT 、CREATE TRIGGER 以及CREATE VIEWE 必须单独交给SQL SERVER， 他们不能和别的SQL语句一起作批处理（CREATE DATABASE、CREATE TABLE 和CREATE INDEX 能和其他语句组成批处理）。

2、USE命令必须在引用这个数据库的对象的语句的前一组批处理中提交。

3、用SET语句做的任何选择设置在批处理尾处发生作用。可以把SET语句和查询语句放入一组批处理中，但SET选择项不能在这组批处理中使用。

## 系统的启动和关闭

### 启动SQL Server

应用SQL Server安装程序（sybinit）创建SQL Server或Backup Server, 默认文件名为RUN\_SYBASE 和RUN\_SYB\_BACKUP，放在￥SYBASE/INSTALL目录下。

Startserver 用来启动一SQL Server 或Backup Server:

Startserver -f RUN\_SYBASE

Startserver -f RUN\_SYB\_BAKUP

可以通过showserver查看SQL ServerBackup Server是否已经启动，在$SYASE/INSTALL下运行showserver 查看SQL Server 是否启动

UID PID PPID C STIME TTY TIME CMD

（Backup Server已启动的信息）

sybase 831 830 0 17:23:50 ttyp1 00:00:00 /sybase/bin/backupserver 朣 SYB\_BACKUP -I/sybase/interfaces -M/sybase/bin/sybmul

（SQL Server已启动的信息）

sybase 674 1 0 14:29:11 ttyp0 00:00:45 /sybase/bin/dataserver -d/sybase/master.dat 朹SYBASE -e/sybase/install/errorlog

### SQL Server的启动过程

master数据库启动-----------鄐ybsystemproces数据库启动-----------郷清空tempdb---------郷启动其它数据库-------郷事务的回滚

### 启动SQL Server的单用户模式

如果要恢复一已被损坏的master数据库，则必须要启动SQL Server的单用户模式。单用户模式下仅允许一个系统管理员向SQL Server注册。

Startserver -f RUN\_SYBASE -m

### 关闭服务器

关闭SQL Server可以在isql中使用T-SQL命令：

关闭SQL备服务器

shutdown SYB\_BACKUP

关闭SQL主服务器

shutdown [with nowait]

### Sybase数据库目录说明

/usr/sybase =.. 主目录

.. /install 安装程序

../bin 执行程序

../include 库函数文件

../lib 宿主语言库

../doc 联机文件

../scripts SQL 动态描述文件

../help 帮助信息

../locales 语言限制文件

../termdef 终端定义文件

../charsets 字符集文件

# SQL Server的用户及权限

## 查询用户

sp\_helpuser

select \* from db\_cus..sysusers

Sybase中的用户分为两种：SQL服务器用户（登录帐号）和数据库用户。

安装完SQL服务器后，系统自动建立一个SQL服务器用户sa，口令为空，即系统管理员，他对整个系统有操作权，其他用户均由系统管理员建立。

在SQL Server中有三种特殊的用户：系统管理员、用户数据库所有者（建立相应数据库的数据库用户）DBO、一般用户。系统管理员对整个系统有操作权；用户数据库所有者对他所建立的数据库具有全部操作权利；一般用户对给定的数据库只有被授权的操作权限。

数据库用户一般可分为用户组，任一数据库在建立后即被赋予一个用户组public。

## 建立SQL服务器用户名（登录帐号）

作为一个用户，为了使用SQL Server，他必须在SQL Server上有一个SQL服务器用户名（登录帐号）。这个帐号是系统管理员通过sp\_addlogin来增加的。

sp\_addlogin 登录名，口令，库名，语言，全名

例：建立用户zhang, 口令为zhangling(最低六位)，全名为Zhang ling

sp\_addlogin zhang,zhangling,null,null,"Zhang ling"

go

select \* from syslogins

go

## 增加数据库用户名

同样，为了使用SQL Server上的数据库，他必须在数据库中有一个用户名，这个用户名是数据库所有者通过sp\_adduser来增加的。数据库用户名不必和用户帐户一样，而且在不同的数据库中用户名可以不同。多个用户可以有相同的SQL Server帐户。同样，多个SQL Server帐户可以有相同的数据库名。

sp\_adduser 登录名，别名，组名

登录名为用户的SQL服务器登录名；别名为用户在数据库中的用户名；组名为用户所属的数据库用户组。

例：用户zhang增加为tele114的用户，别名为zhang1,属于用户组china

1>sp\_addgroup china

2>go

1>sp\_adduser zhang,zhang1,china

2>go

sp\_helpuser

go

\*例：删除别名zhang1,用户组china,登录名zhang

use tele114

go

sp\_dropuser zhang1

go

sp\_helpuser

go

1>sp\_dropgroup china

2>go

sp\_helpgroup

go

use master

go

sp\_droplogin zhang

go

select \* from syslogins

go

数据库操作授权

grant 命令序列 to 用户名

A. 系统管理员可以授予其他用户CREATE DATABASE的权限，使其他用户可以成为数据库所有者。

B. 数据库所有者在他所拥有的数据库中，可以授予其他用户的权限有：

CREATE TABLE ------------------- 建表

CREATE DEFAULT ------------------ 建缺省

CREATE RULE ------------------- 建规则

CREATE PROCedure ------------------ 建存储过程

CREATE VIEW ------------------- 建视图

DUMP DATABASE ------------------- 备份数据库

DUMP TRANsaction ------------------ 备份日志

C. 数据库对象所有者可以授予其他用户的操作权限有：

SELECT

UPDATE

INSERT

EXECute

DELETE

REFERENCE

例：授予zhang1在数据库tele114上建表，建视图，建存储过程;对表students有select,reference权；对name,native字段有update权。

use tele114

go

grant creat table,create procedure ,create view to zhang1

go

grant select,reference on students to zhang1 with grant option

go

grant update on students(name,native) to zhang1

go

sp\_helprotect

go

# 数据库备份与恢复

SQL Server提供了两种不同类型的恢复：一类是系统自动完成的恢复；另一类是人工完成的恢复。

## 系统软件故障恢复

系统自动恢复是在系统软件出现故障的情况下通过重新启动SQL Server进行恢复。

自动恢复从master数据库开始，然后是model数据库，再是temporary数据库，最后是用户数据库。在每个数据库上，恢复机制观察事务日志。如果日志写入比数据库更近的数据，则向前恢复数据库。如果事务尚未完成，则要回退它们。

所谓事务日志就是一张系统表，syslogs。在这张表中记录着对数据库的更新请求，例如插入、修改和删除。查询操作不记入事务日志中。

介质故障的恢复

一旦介质发生故障，只有通过以前备份的数据库和事务日志才能恢复数据库，为此我们需要经常备份数据库和事务日志。备份数据库和事务日志我们使用dump命令，而恢复数据库和事务日志我们使用load命令。

与创建数据库需要一个数据库设备类似，我们备份数据库和事务日志也需要一个卸出设备。增加卸出设备可以通过系统过程sp\_addumpdevice来实现。卸出设备可以是磁盘、磁带或操作系统文件。

sp\_addumpdevice "type",device\_name,physicalname

例：

sp\_addumpdevice "disk", "db114", "/usr/sybase/zyl114.bak",1

（1） 备份数据库和事务日志

备份数据库的语法为：

dump database database\_name to dump\_device

例如：

dump database tele114 to db114

备份事务日志的语法为：

dump transaction database\_name to dump\_device

[with truncate\_only | with no\_log | with no\_truncate]

前两项在日志中删除提交的事务，后一项不删除。

在备份数据库前的所有事务日志备份都是没有意义的。我们一般不经常备份数据库，但我们要经常备份事务日志。这是因为备份事务日志比备份数据库需要更少的时间和存储空间。

（2）恢复用户数据库

要恢复已被破坏的数据库，可以按如下步骤进行：

用带no\_truncate选项的dump命令卸出用户数据库事务日志。

用和被破坏的数据库的同样的结构建立一个新的数据库。

用load命令重新装入数据库和事务日志。

装入数据库的语法为：

load database database\_name

from file\_name

装入事务日志的语法为：

load transaction database\_name

from file\_name

例：数据库备份

dump db\_name to file\_name

go

数据库恢复

load db\_name from file\_name

go

日志备份

dump transaction db\_name to file\_name

go

日志恢复

load transaction db\_name from file\_name

go

日志删除

dump transaction db\_name with truncate\_only

Go

（3）master数据库的恢复

A、保存系统表的拷贝：sysdatabases、sysdevices、sysusages、sysloginroles和Syslogins。

B、转储主设备上的用户数据库。

C、关闭SQL Server并使用buildmaster建立新的master数据库或主设备。

Buildmaster -m ...仅重建master数据库

Buildmaster ...重建主设备

D、已单用户模式启动SQL Server( Master-Recover )

Startserver -f RUN\_服务器名 -m

E、若master数据库用alter database 命令扩展过，严格重建它在sysusages中的分配。

F、若Backup Server 的网络名（接口文件中的名字）不是SYB\_BACKUP,在sysservers中改动网络名。

Update sysservers set srvnetname="PRODUCTION\_BSRV" where srvname="SYB\_BACKUP"

G、使用showserver查看，检查确保Backup Server正运行，若有必要，重启Bakup Server

H、装载master的备份。

Load database master from 'master.bak'

Load database执行成功后，SQL Server 自动关闭。

I、若自master的上次转储以来增加过数据库设备，执行disk reinit命令以重建sysdevices。

J、若执行了disk reinit，或自上次转储以来使用过create database或alter database，执行disk refit 重建sysusages和sysdatabases表，执行完毕后SQL Server自动关闭。

K、检查一致性：比较sysusages和sysdatabases的备份和新的联机版本，对每个数据库执行dbcc checkalloc, 并检查每个数据库中重要的表；若发现错误重新执行disk reinit和disk refit。

L、若恢复了整个主设备，则恢复model数据库。

Cd $SYBASE/scripts

Setenv DSQUERY 服务器名

Isql -Usa -P口令 -S服务器名 <installmodel

M、重新装载所有受影响的用户数据库。

N、恢复服务器用户ID。

O、以多用户模式重新启动SQL Server。

（4）恢复model 数据库

Buildmaster -d/设备名 -x

（5）恢复sybsystemproces数据库

A、删除sybsystemproces数据库（若物理磁盘损坏，删除设备，使用Disk init 重建设备）

B、重建数据库

Create database sybsystemporces on 设备名=数据库大小

C、恢复数据库

Cd $SYBASE/scripts

Setenv DSQUERY 服务器名

Isql -Usa -P口令 -S服务器名 <installmaster

# 几个实用程序

## bcp

以用户指定的形式将数据库表拷贝到操作系统文件中，或从操作系统文件中拷贝一个数据库表。提供了在数据库表和操作系统文件之间传递数据的一个方便、快速的方法。

bcp [[database-name.]owner.]table\_name {in|out} datafile [各种开关]

其中，table\_name是想要拷贝的数据库表名；{in/out}为拷贝方向，in表示从文件到数据库的拷贝，out表示从数据库到文件的拷贝；datafile是操作系统文件的全路径名。

常用的开关有：

-f formatfile----存有上次使用bcp处理同一表时得到的应答文件的全路径名，格式文件名的扩展名为.fmt。使用格式文件，省去了再重复回答前面的问题。

-U username----允许用户指定一个注册名。

-P password----允许用户指定当前的SQL Server口令。如果未使用-P选项，bcp将提示输入一个口令。

-S Server----允许用户指定与其相连的SQL Server的名字。

例：在客户端备份tele114数据库中code\_1th表,按照code\_1th.fmt的格式

bcp tele114..code\_1th out code\_1th.txt -杅code\_1th.fmt 朥sa 朠

## isql

与SQL Server交互的SQL句法分析器。

isql [各种开关]

常用的开关有：

-U username----允许用户指定一个注册名。

-P password----允许用户指定当前的SQL Server的口令。

-S Server----允许用户指定与其连接的SQL Server的名称。

-i输入文件

-o输出文件

终止一条命令：go

清除查询缓冲：reset

调用编辑器：vi

执行一条操作系统命令：！！命令

退出isql：quit或exit

## showserver

显示当前在本机上正在运行的SQL Server。

showserver

## startserver

启动SQL Server。

startserver [各种开关]

常用的开关有：

-f runserverfile----指定一个运行服务器文件名，每次重启SQL Server时，以这个文件为参考。在我们的实际机器上，这个文件名为RUN\_SYBASE或RUN\_SYB\_BACKUP。

-m ----以单用户的方式启动SQL Server。这个方式用于恢复master数据库。

## syman

显示联机的库例程，系统过程及实用程序手册。

syman [keyword]

syman实用程序的联机文本文件存放在$SYBASE下的doc子目录中。

## dbcc 数据库一致性检查

Dbcc 用来检查一个数据库的逻辑一致性和物理一致性。通常在下面的情况下使用dbcc:

常规数据库维护

确定当一个系统错误后可能损坏的范围

在备份数据库之前

怀疑数据库受到损坏。

dbcc有两个主要功能：可检查分配表结构（命令checkalloc、tablealloc和Indexalloc);可检查页链接和页级、行级上的数据指针。

只有表属主可以用checktable、fix-text和reindex选项执行dbcc命令。只有数据库属主可以使用checkdb、checkalloc、checkcatalog、indexalloc和tablealloc选项。只有系统管理员可以使用dbrepair选项

dbcc命令的完整语法格式如下：

Dbcc

{checktable({表明|表标识}[，Skip\_ncindex])|

Checkdb [(数据库名[，skip\_ncindex])]|

Checkalloc [(数据库名[，fix|nofix])]|

Talbealloc （{表明|表标识}[，{full|optimized|fast|null}[,fix|nofix]]))|

Indexalloc （{表明|表标识}，索引标识[，{full|optimized|fast|null}[,fix|nofix]]))|

Checkcatalog [（数据库名）]

Dbrepair (数据库名，dropdb）|

Reindex ({表明|表标识})|

fix\_text({表明|表标识})}

# Sybase客户端软件

## 安装

执行setup.exe，一路继续安装完成，没有特殊的选择。

## 运行

Sybase Client应用软件包常用工具包含三个部分：Dsedit、SQL Advantage、Sybase Central。现假设要连接一台Sybase Server，可以遵从下述简要操作顺序。

1）运行Dsedit，选择OK，进入InterfacesDriver窗口模式，在Server子窗口中点击鼠标右键，在弹出菜单中选择Add，输入服务器名，如INET（只是个标识而已），在Attributes子窗口中点击鼠标右键，在弹出菜单中选择Modify Attributes，在对话框中执行Add操作，协议类型为缺省值TCP，Network Address的格式为“SybaseServerIP,SybaseServer端口号”，如“10.108.114.13,4100”，其中端口号可以在SybaseServer上查到。

具体信息在$SYBASE/interfaces文件中，在文件中有如下信息：

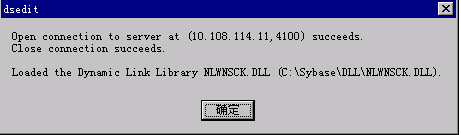
inet2

master tli tcp /dev/tcp \x000210040a6c720d0000000000000000

query tli tcp /dev/tcp \x000210040a6c720d0000000000000000

其中inet2为SybaseServer的名称，0X1004=4100为SybaseServer的TCP端口。

设置完服务器信息后，可以执行Ping Server操作，如果显示结果为：



则表示客户端已经正确的设置好了服务器信息。

2）运行SQL Advantage，进入此工具时先要登录相应的服务器，可以在菜单中选择server\connet操作，在对话框中选择相应的SybaseServer，Server列表必须在Dsedit工具中定义，利用此工具可以方便地操作Server数据库，此工具与Microsoft SQL Server 7.0的Isql工具使用方法相同。

3）运行Sybase Central，点击相应的服务器名称，则弹出登录对话框，输入用户及口令，可以进行数据库设备、用户、数据库等方面的管理，其中master、model、sybsystemprocs、tempdb为系统数据库，其余为应用系统数据库。

# 附录

附录1：只有在master数据库中才能有的系统表

|  |  |
| --- | --- |
| 系统表名 | 表中的内容 |
| Sysconfigures | 一行记录了用户可设置的配置参数 |
| Syscurconfigs | 有关SQL Server当前正使用的配置参数情况 |
| Sysdatabases | 一行记录了SQL Server中的一个数据库 |
| Sysdevices | 一行记录了数据库的每一磁带转储设备，磁盘转储设备，数据库设备和磁盘分区 |
| 设备和磁盘分区 |  |
| Syslocks | 有关动态锁的情况 |
| Syslogins | 一行记录了每一有效的SQL Server的用户帐号 |
| Sysmessages | 一行记录了每一系统错误或警告 |
| Sysprocesses | 有关Server进程的情况 |
| Sysremotelogins | 一行记录了一个远程用户 |
| Sysservers | 一行记录了一个远程SQL Server |

## 附录2：Sysusages 一行记录了分配给一个数据库的每一个磁盘分析

|  |  |
| --- | --- |
| 系统表名 | 表中的内容 |
| Sysalternates | 一行记录了SQL Server用户在当前数据库中的标识 |
| syscoumns | 一行记录了一个表或视图的每一列，一个存储过程的每一参数 |
| syscomments | 一行或多行记录了每一视图、规则、缺省值、触发器和存储过程 |
| sysdepends | 一行记录了由一个过程、视图或触发器所参照的每一过程，视图或表 |
| Syskeys | 一行记录了每一主码，外码或公用码 |
| Syslogs |  |
| sysobjects | 一行记录了每一表、视图、存储过程、日志、规则、缺省值、触发器和临时表（仅在tempdb中） |
| sysprocedures | 一行记录了每一视图、规则、缺省、触发器和过程 |
| syprotects | 记录用户的权限信息 |
| sysegments | 一行记录了每一片段（命名的的磁盘分片汇集） |
| systypes | 一行记录了每一由系统提供的和用户定义的数据类型 |
| sysusers | 一行记录了一个数据库的合法用户，或用户组 |

## 附录3： 由SQL Server提供的系统过程功能和执行权

|  |  |  |
| --- | --- | --- |
| 过程名 | 功能 | 执行权限 |
| sp\_addalias | 把数据库中的一个用户映象到另一个用户 | 数据库所有者 |
| sp\_addgroup | 向数据库增加分组，在授权和撤权命令中分组被用作聚合名字 | 数据库所有者 |
| sp\_addlanguage | 为服务器定义可替换语言，并把它增加到master，dbo,syslanguages中。 | 系统管理员 |
| sp\_addlogin | 通过向master.dbo,syslogins中增加表项授权新的SQL Server用户 | 系统管理员 |
| sp\_addremotelogin | 通过向master，bo,sysremotelogins增添表项授权新的远程服务器用户 | 系统管理员 |
| sp\_addsegment | 在指定的数据库设备上创建一个段 | 数据库所有者系统管理员 |
| sp\_addserver | 定义远程服务器，或定义本地服务器名字 | 数据库所有者系统管理员 |
| sp\_addtype | 创建用户定义的数据类型 | Public |
| sp\_addumpdervce | 向SQL Server增加转储设备 | 系统管理员 |
| sp\_adduser | 向当前数据库添加新的用户 | 数据库所有者 |
| sp\_bindefault | 为一列或用户下义的数据类型定义缺省值 | 目标（对象）所有者 |
| sp\_bindrule | 向一列或用户定义的数据类型赋一规则 | 目标汽有者 |
| sp\_changedbowner | 改变数据库的所有者 | 数据库所有者 |
| sp\_changegroup | 改变用户所属的分组 | 数据库所有者 |
| sp\_checknames | 检查当前数据库在7位ASCII码集之外的字符的名字 | public |
| sp\_commonkey | 在两个表或视图之间定义一公共键－－经常被连接的列 | 两表（或视图）的所有者 |
| sp\_configure | 显示或改变配置变量 | 没有参数或只有第一个参数的执行权属于public，否则属于系统管理员 |
| sp\_dboption | 显示或改变数据库选项 | 不带参数属于public，带参数时属于数据库所有者 |
| sp\_defaultdb | 改变一用户的缺省数据库 | 系统管理员（但每个用户都可以改变他自己的缺省语言） |
| sp\_depends | 显示数据库目标相关性的信息 | public |
| sp\_diskdefault | 设备数据库设备的状态为defaulton或defaultoff（表明一数据设备是否能用于数据库的岂省存储） | 系统管理员 |
| sp\_dropalias | 删除用sp\_addalias建立的用户名别名标识 | 数据库所有者 |
| sp\_dropdevice | 删除一SQL Server数据库设备或转储设备 | 系统管理员 |
| sp\_dropgroup | 从数据库中删除一分组 | 数据库所有者 |
| sp\_droplanguage | 从服务器中删除可替换语言，并从master.dbo.syslanguages中删掉相应的行 | 系统管理员 |
| sp\_droplogin | 通过删掉用户在master,dbo,syslogins中的表项删除SQL Server用户 | 系统管理员 |
| sp\_dropremotelogin | 删除远程用户注册 | 系统管理员 |
| sp\_dropsegment | 从数据库中删除一个段或从特定数据库设备上卸掉一个段的映象 | 数据库所有者 |
| sp\_dropserver | 删除远程服务器 | 系统管理员 |
| sp\_droptype | 删除用户定义的数据类型 | 数据库所有者数据类型所有者 |
| sp\_dropuser | 从当前是数据库中删除一用户 | 数据库属主 |
| sp\_extendsegment | 把段的区域扩充到另一个数据库设备或在当前数据库设备上进行扩充 | 数据库属主 |
| sp\_foreignkey | 在当前数据库的表或视图上定义一外部键 | 表的属主 |
| sp\_help | 报告目标或SQL Server提供的或用户定义的数据库类型的信息 | public |
| sp\_helpdb | 报告－－指定数据库或所有数据库的信息 | public |
| sp\_helpdevice | 报告－－指定设备或所有SQL Server数据库设备转储设备的信息 | public |
| sp\_helpgroup | 报告－－特定组或当前数据库的所有组的信息 | public |
| sp\_helpinde | 报告－－表中的索引信息 | public |
| sp\_helpjoins | 列出可连接两个表或视图的列 | public |
| sp\_helpkey | 报告某个特定表或视图的一主键，外部键或公共键或者当前数据库中所有键的信息 | public |
| sp\_helplanguage | 报告一特替换语言或所有语言的信息 | public |
| sp\_helplog | 报告包括日志第一页的设备名 | public |
| sp\_helpremotelogin | 报告一特定远程服务器或所有远程服务器的注册信息 | public |
| sp\_helprotect | 报告一数据库对象或用户权限，且由用户选择对象 | public |
| sp\_helpsegment | 报告当前数据库中一特定段或所有段的信息 | public |
| sp\_helpserver | 报告一特定远程服务器或所有远程服务器的信息 | public |
| sp\_helpsort | 显示SQL Server的缺省排列顺序以及字符集 | public |
| sp\_helptext | 打印一系统过程、触发器、视图、缺省或规则的定义文本 | public |
| sp\_helpuser | 报告当前数据库中一个用户或所有用户的信息 | public |
| sp\_lock | 报告当前上锁的进程的信息 | public |
| sp\_logdevice | 将系统表syslogs（其中包括事务日志）放在一个分设的数据库设备中 | 数据库属主 |
| sp\_monitor | 显示SQL Server的统计数字 | 系统管理员 |
| sp\_password | 增加或改变SQL Server注册的口令 | 对自己的口令，权限属于public，系统管理员有权修改所有用户口令 |
| sp\_placeobject | 为一特定中某个表或索引放置将来的空间分配 | 表属主，数据库属主系统管理员 |
| sp\_primarykey | 定义一表或视图上的主键（主码） | 表的所有者 |
| sp\_recompile | 使生个使用被指定表的存储过程、触发器在下次运行时重新编译 | public |
| sp\_remoteoption | 显示或改变远程注册选项 | 显示选项时属 public改变选项时属系统管理员 |
| sp\_rename | 改变当前数据库中一用户建立目标的名称 | 特定目标，权限属目标属主对数据库所有对象，权限属数据库所有者& |
| sp\_renamedb | 改变数据库名 | 系统管理员 |
| sp\_serveroption | 显示或改变服务器选项 | 显示选项，权限，权限属 public改变选项，权限属系统管理员 |
| sp\_setlangalian | 分配或改变一替换语言的别名 | 系统管理员 |
| sp\_spaceused | 显示行数、数据页数以及当前数据库中由某个目标或所有目标所占用的空间 | public |
| sp\_unbindefault | 从一列或用户定义的数据类型中解除一规则 | 目标属主（所有者） |
| sp\_who | 报告特定用户或SQL Server所有当前用户和进程的信息 | public |