```
-查询用友版本号
use ufsystem
go
select * from UA_Version
go
------------------------------------------------------------------
--查看系统用户信息表
use ufsystem
select cUser_Id as 操作员编码,
cUser_Name as 操作员名称,
nState as 是否停用 ,
iAdmin as 是否帐套主管理,
cDept as 所属部门,
cBelongGrp as 所在组,
nState as 是否停用
from UA_User
--查看具有帐套主管身份的操作员
select cUser_Id as 操作员编码,
cUser_Name as 操作员名称
from UA_User where iAdmin=1;
--查看被停用的操作员
select cUser_Id as 操作员编码,
cUser_Name as 操作员名称
from UA_User where nState=1;
--帐套主子表相关信息
use ufsystem
--帐套主表
select
 cAcc_Id as 账套号,
 cAcc_Name as 账套名称,
 cAcc_Path as 账套路径,
 iYear as 启用会计期年,
 iMonth as 启用会计期月,
 cAcc_Master as 账套主管,
 cCurCode as 本币代码,
 cCurName as 本币名称,
 cUnitName as 单位名称,
 cUnitAbbre as 单位简称,
 cUnitAddr as 单位地址,
```

```
 cUnitZap as 邮政编码,
 cUnitTel as 联系电话,
 cUnitFax as 传真,
 cUnitEMail as 电子邮件,
 cUnitTaxNo as 税号,
 cUnitLP as 法人,
 cEntType as 企业类型,
 cTradeKind as 行业类型,
 cIsCompanyVer as 是否集团版,
 cDomain as 域名,
 cDescription as 备注,
 cOrgCode as 机构编码,
 iSysID as 账套内部标识
from ua_account
--帐套子表
select cAcc_Id as 账套号,
iYear as 账套年度,
 cSub_Id as 模块标识,
 bIsDelete as 是否删除,
 bClosing as 是否关闭,
 iModiPeri as 会计期间,
 dSubSysUsed as 启用会计日期,
 cUser_Id as 操作员,
 dSubOriDate as 启用自然日期
from ua_account_sub
--当客户的数据在其它机器上做的升级然后拷回到原机器
/*拷回的数据，通过‘系统管理’在原机器上引入后，并不会在
ufsystem数据库中的ua_account_sub这个帐套子表中回写上一年度的bClosing字
段来关闭上一年度
*/
--比如002帐套结转后年度为2010,则用于关闭上一(2009)年度的sql如下：
select * from ua_account_sub where cAcc_Id='002' and iYear=2008
update ua_account_sub set bclosing=0
where cAcc_Id='002' and iYear=2008
----------------------------------------------------------------
--清除异常任务及单据锁定
use ufsystem
delete from ua_task
delete from ua_tasklog
```

```
go
delete from ufsystem..ua_task
delete from ufsystem..ua_tasklog
go
Select *
From ua_task
Where(cacc_id='***') --注：（***为账套号）
--科目锁定的解决
/*XX科目已经被用户[XX]锁定”
或“科目(xxxxxx)正在被机器(xxxx)上的用户(xxx)进行(xxxx)操作锁定,请稍候再试
”。
*/
use UFDATA_002_2008
select ccode as 科目编码,
 cauth as 功能名称,
 cuser as 用户名,
 cmachine as 机器名
from GL_mccontrol
delete from GL_mccontrol
-----------------------------------------------------------------
--如何取得一个数据表的所有列名
/*
方法如下：先从SYSTEMOBJECT系统表中取得数据表的SYSTEMID,然后再SYSCOLUMN表中取得该数据表的所有列名。
SQL语句如下：*/
/*(方法一*/
select * from ufsystem..ua_account
select * from syscolumns where id=object_id('ua_account')
declare @objid int,@objname char(40)
set @objname = 'ua_account'
select @objid = id from sysobjects where id = object_id(@objname)
select 'Column_name' = name from syscolumns where id = @objid order by colid
/*(方法二（邹建）*/
---跟踪程序的运行就可以了.
/*

 开始--程序--MS SQLSERVER
 --事件探察器(SQL Profiler)
```

--文件
--新建--跟踪...
--设置要跟踪的服务器的信息(连接服务器)--确定
--设置跟踪的项目...
--然后数据库的调用情况就会显示出来

在跟踪项目设置中,如果不熟悉的话,一般用默认设置
筛选项目有几个可以注意一下:

1.DatabaseName
同于你要监测的数据库名(不过这个好像不起作用,我的电脑上设置无效)
2.DatabaseID 同于你要检测的数据库的dbid,可以用 select
db_id(N'你要监测的库名')得到dbid
3.ObjectName 同于你要监测的对象名,例如表名,视图名等
4.ObjectID 同于你要监测的对象的id,可以用 select
object_id(N'你要监测的对象名')得到id
5.Error 同于错误,如果经常出现某个编号的错误,则针对此错误号
6.Seccess 同于0,失败,1,成功,如果是排错,就过滤掉成功的处理
*/
/*方法三：*/
--如果直接查询,可以参考我的这段代码:

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[p_search]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[p_search]
GO

/*--搜索某个字符串在那个表的那个字段中

--邹建 2004.10(引用请保留此信息)--*/

/*--调用示例
use pubs
exec p_search N'l'
--*/
create proc p_search
@str Nvarchar(1000) --要搜索的字符串
as
if @str is null return
```

```
 declare @s Nvarchar(4000)
 create table #t(表名 sysname,字段名 sysname)

 declare tb cursor local for
 select s='if exists(select 1 from ['+replace(b.name,']',']]')+'] where
['+a.name+'] like N''%'+@str+'%'')
 print ''所在的表及字段: ['+b.name+'].['+a.name+']'''
 from syscolumns a join sysobjects b on a.id=b.id
 where b.xtype='U' and a.status>=0
 and a.xtype in(175,239,99,35,231,167)
 open tb
 fetch next from tb into @s
 while @@fetch_status=0
 begin
 exec(@s)
 fetch next from tb into @s
 end
 close tb
 deallocate tb
 go
-------------------------------------------------------------------
-- 通过SQL语句来更改用户的密码
/*修改别人的,需要sysadmin role */
EXEC sp_password NULL, 'newpassword', 'User'
/*如果帐号为SA执行*/
EXEC sp_password NULL, 'newpassword', sa
-------------------------------------------------------------------
-- 通怎么判断出一个表的哪些字段不允许为空？
select COLUMN_NAME from INFORMATION_SCHEMA.COLUMNS where
IS_NULLABLE='NO' and TABLE_NAME='ua_account'
-------------------------------------------------------------------
-- 如何在数据库里找到含有相同字段的表？
-- a. 查已知列名的情况
SELECT b.name as TableName,a.name as columnname
From syscolumns a INNER JOIN sysobjects b
ON a.id=b.id
AND b.type='U'
AND a.name='cacc_id' --本例如：cacc_id列
```

```
-- b. 未知列名查所有在不同表出现过的列名
Select o.name As tablename,s1.name As columnname
From syscolumns s1, sysobjects o
Where s1.id = o.id
 And o.type = 'U'
 And Exists (
 Select 1 From syscolumns s2
 Where s1.name = s2.name
 And s1.id <> s2.id
 )
---------------------------------------------------------------
-- 查询第xxx行数据
-- 假设id是主键：
select *
from (select top xxx * from yourtable) aa
where not exists(select 1 from (select top xxx-1 * from yourtable) bb where
aa.id=bb.id)
-- 如果使用游标也是可以的
fetch absolute [number] from [cursor_name]
-- 行数为绝对行数
---------------------------------------------------------------
-- SQL Server日期计算
/*a. 一个月的第一天*/
SELECT DATEADD(mm, DATEDIFF(mm,0,getdate()), 0)
/*b. 本周的星期一*/
SELECT DATEADD(wk, DATEDIFF(wk,0,getdate()), 0)
/*c. 一年的第一天*/
SELECT DATEADD(yy, DATEDIFF(yy,0,getdate()), 0)
/*d. 季度的第一天*/
SELECT DATEADD(qq, DATEDIFF(qq,0,getdate()), 0)
/*e. 上个月的最后一天 */
SELECT dateadd(ms,-3,DATEADD(mm, DATEDIFF(mm,0,getdate()), 0))
/*f. 去年的最后一天*/
SELECT dateadd(ms,-3,DATEADD(yy, DATEDIFF(yy,0,getdate()), 0))
/*g. 本月的最后一天*/
SELECT dateadd(ms,-3,DATEADD(mm, DATEDIFF(m,0,getdate())+1, 0))
/*h. 本月的第一个星期一*/
select DATEADD(wk, DATEDIFF(wk,0,
 dateadd(dd,6-datepart(day,getdate()),getdate())
```

), 0)
/*i. 本年的最后一天*/
SELECT dateadd(ms,-3,DATEADD(yy, DATEDIFF(yy,0,getdate())+1, 0))
1.显示本月第一天
SELECT DATEADD(mm,DATEDIFF(mm,0,getdate()),0)
select convert(datetime,convert(varchar(8),getdate(),120)+'01',120)
2.显示本月最后一天
select dateadd(day,-1,convert(datetime,convert(varchar(8),dateadd(month,1,getdate()),120)+'01',120))
SELECT dateadd(ms,-3,DATEADD(mm,DATEDIFF(m,0,getdate())+1,0))
3.上个月的最后一天
SELECT dateadd(ms,-3,DATEADD(mm,DATEDIFF(mm,0,getdate()),0))
4.本月的第一个星期一
select DATEADD(wk,DATEDIFF(wk,0,
dateadd(dd,6-datepart(day,getdate()),getdate())),0)
5.本年的第一天
SELECT DATEADD(yy,DATEDIFF(yy,0,getdate()),0)
6.本年的最后一天
SELECT dateadd(ms,-3,DATEADD(yy,DATEDIFF(yy,0,getdate())+1,0))
7.去年的最后一天
SELECT dateadd(ms,-3,DATEADD(yy,DATEDIFF(yy,0,getdate()),0))
8.本季度的第一天
SELECT DATEADD(qq,DATEDIFF(qq,0,getdate()),0)
9.本周的星期一
SELECT DATEADD(wk,DATEDIFF(wk,0,getdate()),0)
10.查询本月的记录
select * from tableName where DATEPART(mm, theDate) = DATEPART(mm, GETDATE()) and DATEPART(yy, theDate) = DATEPART(yy, GETDATE())
11.查询本周的记录
select * from tableName where DATEPART(wk, theDate) = DATEPART(wk, GETDATE()) and DATEPART(yy, theDate) = DATEPART(yy, GETDATE())
12.查询本季的记录
select * from tableName where DATEPART(qq, theDate) = DATEPART(qq, GETDATE()) and DATEPART(yy, theDate) = DATEPART(yy, GETDATE())
其中:GETDATE()是获得系统时间的函数。
13.获取当月总天数:
select DATEDIFF(dd,getdate(),DATEADD(mm, 1, getdate()))
select datediff(day,
dateadd(mm, datediff(mm,'',getdate()), ''),

```
dateadd(mm, datediff(mm,'',getdate()), '1900-02-01'))
14.获取当前为星期几
DATENAME(weekday, getdate())
----------------------------------------------------------------
/*查询数据库的所有用户表*/
use ufdata_002_2008
select name from sysobjects where type='U'
----------------------------------------------------------------
--查看数据库中所有的触发器
use ufdata_002_2008
go
select * from sysobjects where xtype='TR'
----------------------------------------------------------------
--查询特定数据库中某一不知归属表的触发器
/*查询某一个触发器TR_Ap_CloseBills所归属的表*/
use ufdata_002_2008
declare @parent_obj_id int --定义父对象id变量
--先找出父对象（所在表）的id(在触发器不重复归属于多个表的情况下)
select @parent_obj_id=parent_obj
from sysobjects where name='TR_Ap_CloseBills'
and xtype='TR'
print '所在父对象（表）的ID是：'+str(@parent_obj_id)
--接下来找出父对象（表）的名称
select name as 触发器所在表为
from sysobjects where type='U' and id=@parent_obj_id
----------------------------------------------------------------
--查看触发器内容
use ufdata_002_2008
go
exec sp_helptext 'TR_Ap_CloseBills'
----------------------------------------------------------------
--用于查看触发器的属性（参数指定触发器所在的表）
use ufdata_002_2008
go
exec sp_helptrigger Ap_CloseBills
----------------------------------------------------------------
--创建触发器
/*
（1）创建一个简单的触发器
```

触发器是一种特殊的存储过程，类似于事件函数，SQL Server? 允许为
INSERT、UPDATE、DELETE
创建触发器，即当在表中插入、更新、删除记录时，触发一个或一系列 T-
SQL语句。
触发器可以在查询分析器里创建，也可以在表名上点右键->“所有任务”->“管理
触发器”来创建，不过都是要写 T-SQL
语句的，只是在查询分析器里要先确定当前操作的数据库。
创建触发器用 CREATE TRIGGER
格式如下：

CREATE TRIGGER 触发器名称
ON 表名
FOR INSERT、UPDATE 或 DELETE
AS
T-SQL 语句

注意：触发器名称是不加引号的。
*/
--如下是联机丛书上的一个示例，当在 titles 表上更改记录时，发送邮件通知
MaryM。
CREATE TRIGGER reminder
ON titles
FOR INSERT, UPDATE, DELETE
AS
EXEC master..xp_sendmail 'MaryM',
'Don''t forget to print a report for the distributors.'
/*
（2）删除触发器
用查询分析器删除
在查询分析器中使用 drop trigger 触发器名称 来删除触发器。
也可以同时删除多个触发器：drop trigger 触发器名称,触发器名称...
注意：触发器名称是不加引号的。在删除触发器之前可以先看一下触发器是否存在
：
格式如下：
---
if Exists(
select name from sysobjects
where name=触发器名称 and xtype='TR'
)

用企业管理器删除
在企业管理器中，在表上点右键->“所有任务”->“管理触发器”，选中所要删除的触发器，然后点击“删除”。
*/
/*
（3）重命名触发器
用查询分析器重命名
exec sp_rename 原名称, 新名称
sp_rename 是 SQL Server? 自带的一个存储过程，用于更改当前数据库中用户创建的对象的名称，如表名、列表、索引名等。
用企业管理器重命名
在表上点右键->“所有任务”->“管理触发器”，选中所要重命名的触发器，修改触发器语句中的触发器名称，点击“确定”。
*/
/*（4）更多功能
①INSTEAD OF 子句
执行触发器语句，但不执行触发触发器的 SQL
语句，比如试图删除一条记录时，将执行触发器指定的语句，此时不再执行 delete 语句。例：

```
create trigger f
on tbl
instead of delete
as
 insert into Logs...
```
---
②IF UPDATE(列名) 子句
检查是否更新了某一列，用于 insert 或 update，不能用于 delete。例：

```
create trigger f
on tbl
for update
as
 if update(status) or update(title)
 sql_statement --更新了 status 或 title 列
```
---
③inserted、deleted（两个虚拟表的使用）
这是两个虚拟表，inserted 保存的是 insert 或 update

之后所影响的记录形成的表，deleted 保存的是 delete 或 update
之前所影响的记录形成的表。例：

```
create trigger tbl_delete
on tbl
for delete
as
 declare @title varchar(200)
 select @title=title from deleted
 insert into Logs(logContent) values('删除了 title 为：' + title + '的记录')
```

说明：如果向 inserted 或 deleted 虚拟表中取字段类型为 text、image
的字段值时，所取得的值将会是 null。
*/
/*
（5）递归、嵌套触发器
递归分两种，间接递归和直接递归。我们举例解释如下，假如有表1、表2名称分别为 T1、T2，在 T1、T2 上分别有触发器 G1、G2。
? 间接递归：对 T1 操作从而触发 G1，G1 对 T2 操作从而触发 G2，G2 对 T1 操作从而再次触发 G1...
? 直接递归：对 T1 操作从而触发 G1，G1 对 T1 操作从而再次触发 G1...

嵌套触发器
类似于间接递归，间接递归必然要形成一个环，而嵌套触发器不一定要形成一个环，它可以 T1->T2->T3...这样一直触发下去，最多允许嵌套 32 层。

设置直接递归
默认情况下是禁止直接递归的，要设置为允许有两种方法：
? T-SQL：exec sp_dboption 'dbName', 'recursive triggers', true
? EM：数据库上点右键->属性->选项。

设置间接递归、嵌套
默认情况下是允许间接递归、嵌套的，要设置为禁止有两种方法：
? T-SQL：exec sp_configure 'nested triggers', 0 --第二个参数为 1 则为允许
? EM：注册上点右键->属性->服务器设置。
*/
/*（6）触发器回滚
我们看到许多注册系统在注册后都不能更改用户名，但这多半是由应用程序决定的，如果直接打开数据库表进行更改，同样可以更改其用户名，在触发器中利用回滚

就可以巧妙地实现无法更改用户名。
语句如下：
---
use 数据库名
go
create trigger tr
on 表名
for update
as
 if update(userName)
 rollback tran

关键在最后两句，其解释为：如果更新了 userName 列，就回滚事务。
*/
/*（7）禁用、启用触发器*/
--禁用：
alter table 表名 disable trigger 触发器名称
--启用：
alter table 表名 enable trigger 触发器名称
/*如果有多个触发器，则各个触发器名称之间用英文逗号隔开。
如果把“触发器名称”换成“ALL”，则表示禁用或启用该表的全部触发器。