**数据库常用操作语句**

基础

1.插入指定行数据

Insert into 表名(列名) values(值)

**1、说明：创建数据库**  
**CREATE DATABASE database-name**   
**2、说明：删除数据库  
drop database dbname**  
**4、说明：创建新表  
create table tablename(col1 type1 [not null] [primary key],col2 type2 [not null],..)**

**根据已有的表创建新表：**   
**A：create table tab\_new like tab\_old (使用旧表创建新表)**  
**B：create table tab\_new as select col1,col2… from tab\_old definition only**  
**5、说明：删除新表  
drop table tablename**   
**6、说明：增加一个列  
Alter table tablename add column col type**  
**注**：列增加后将不能删除。DB2中列加上后数据类型也不能改变，唯一能改变的是增加varchar类型的长度。  
**7、说明：添加主键**： **Alter table tablename add primary key(col)**   
说明：删除主键： Alter table tablename drop primary key(col)   
**8、说明：创建索引**：**create [unique] index idxname on tablename(col….)**   
删除索引：**drop index idxname**  
注：索引是不可更改的，想更改必须删除重新建。  
**9、说明：创建视图**：create view viewname as select statement   
**删除视图**：drop view viewname  
**10、说明：几个简单的基本的sql语句  
选择：**select \* from table1 where 范围  
**插入：**insert into table1(field1,field2) values(value1,value2)  
**删除：**delete from table1 where 范围 **更新**：update table1 set field1=value1 where 范围  
**查找**：select \* from table1 where field1 like ’%value1%’ ---like的语法很精妙，查资料!  
**排序**：select \* from table1 order by field1,field2 [desc]  
**总数**：select count as totalcount from table1  
**求和**：select sum(field1) as sumvalue from table1  
**平均**：select avg(field1) as avgvalue from table1  
**最大**：select max(field1) as maxvalue from table1  
**最小**：select min(field1) as minvalue from table1  
**11、说明：几个高级查询运算词  
A：** UNION 运算符   
UNION 运算符通过组合其他两个结果表（例如 TABLE1 和 TABLE2）并消去表中任何重复行而派生出一个结果表。当 ALL 随 UNION一起使用时（即 UNION ALL），不消除重复行。两种情况下，派生表的每一行不是来自 TABLE1 就是来自 TABLE2。   
**B： EXCEPT 运算符**   
EXCEPT 运算符通过包括所有在 TABLE1 中但不在 TABLE2 中的行并消除所有重复行而派生出一个结果表。当 ALL 随 EXCEPT 一起使用时 (EXCEPT ALL)，不消除重复行。   
**C： INTERSECT** 运算符  
INTERSECT 运算符通过只包括 TABLE1 和 TABLE2 中都有的行并消除所有重复行而派生出一个结果表。当 **ALL** 随 INTERSECT 一起使用时 (INTERSECT ALL)，不消除重复行。   
**注：**使用运算词的几个查询结果行必须是一致的。   
**12、说明：使用外连接**   
A、**left （outer） join**：   
左外连接（左连接）：结果集几包括连接表的匹配行，也包括左连接表的所有行。   
SQL: select a.a, a.b, a.c, b.c, b.d, b.f from a LEFT OUT JOIN b ON a.a = b.c  
**B：right （outer） join:**   
右外连接(右连接)：结果集既包括连接表的匹配连接行，也包括右连接表的所有行。   
**C：full/cross （outer） join**：   
全外连接：不仅包括符号连接表的匹配行，还包括两个连接表中的所有记录。  
**12、分组:Group by:** 一张表，一旦分组 完成后，查询后只能得到组相关的信息。  
**组相关的信息：**（统计信息） count,sum,max,min,avg **分组的标准)  
在SQLServer中分组时：不能以text,ntext,image类型的字段作为分组依据  
在selecte统计函数中的字段，不能和普通的字段放在一起；**

**13、对数据库进行操作：  
分离数据库**： sp\_detach\_db; 附加数据库：**sp\_attach\_db** 后接表明，附加需要完整的路径名  
**14.如何修改数据库的名称:**  
sp\_renamedb 'old\_name', 'new\_name'

**二、提升**

**1、说明：复制表(只复制结构,源表名：a 新表名：b) (Access可用)  
法一：**select \* into b from a where 1<>1（仅用于SQlServer）  
**法二：**select top 0 \* into b from a  
**2、说明：拷贝表(拷贝数据,源表名：a 目标表名：b) (Access可用)**insert into b(a, b, c) select d,e,f from b;

**3、说明：跨数据库之间表的拷贝(具体数据使用绝对路径) (Access可用)**insert into b(a, b, c) select d,e,f from b in ‘具体数据库’ where 条件  
例子：..from b in '"&Server.MapPath(".")&"\data.mdb" &"' where..

**4、说明：子查询(表名1：a 表名2：b)**select a,b,c from a where a IN (select d from b ) 或者: select a,b,c from a where a IN (1,2,3)

**5、说明：显示文章、提交人和最后回复时间**select a.title,a.username,b.adddate from table a,(select max(adddate) adddate from table where table.title=a.title) b

**6、说明：外连接查询(表名1：a 表名2：b)**select a.a, a.b, a.c, b.c, b.d, b.f from a LEFT OUT JOIN b ON a.a = b.c

**7、说明：在线视图查询(表名1：a )**select \* from (SELECT a,b,c FROM a) T where t.a > 1;

**8、说明：between的用法,between限制查询数据范围时包括了边界值,not between不包括**select \* from table1 where time between time1 and time2  
select a,b,c, from table1 where a not between 数值1 and 数值2

**9、说明：in 的使用方法**select \* from table1 where a [not] in (‘值1’,’值2’,’值4’,’值6’)

**10、说明：两张关联表，删除主表中已经在副表中没有的信息**   
delete from table1 where not exists ( select \* from table2 where table1.field1=table2.field1 )

**11、说明：四表联查问题：**select \* from a left inner join b on a.a=b.b right inner join c on a.a=c.c inner join d on a.a=d.d where .....

**12、说明：日程安排提前五分钟提醒**   
SQL: select \* from 日程安排 where datediff('minute',f开始时间,getdate())>5

**13、说明：一条sql** **语句搞定数据库分页**select top 10 b.\* from (select top 20 主键字段,排序字段 from 表名 order by 排序字段 desc) a,表名 b where b.主键字段 = a.主键字段 order by a.排序字段  
**具体实现：**  
关于数据库分页：

declare @start int,@end int

@sql nvarchar(600)

set @sql=’select top’+str(@end-@start+1)+’+from T where rid not in(select top’+str(@str-1)+’Rid from T where Rid>-1)’

exec sp\_executesql @sql

**注意：在top后不能直接跟一个变量，所以在实际应用中只有这样的进行特殊的处理。Rid为一个标识列，如果top后还有具体的字段，这样做是非常有好处的。因为这样可以避免** top**的字段如果是逻辑索引的，查询的结果后实际表中的不一致（逻辑索引中的数据有可能和数据表中的不一致，而查询时如果处在索引则首先查询索引）**

**14、说明：前10条记录**select top 10 \* form table1 where 范围

**15、说明：选择在每一组b值相同的数据中对应的a最大的记录的所有信息(类似这样的用法可以用于论坛每月排行榜,每月热销产品分析,按科目成绩排名,等等.)**select a,b,c from tablename ta where a=(select max(a) from tablename tb where tb.b=ta.b)

**16、说明：包括所有在** TableA 中但不在 TableB和TableC 中的行并消除所有重复行而派生出一个结果表  
(select a from tableA ) except (select a from tableB) except (select a from tableC)

**17、说明：随机取出10条数据**select top 10 \* from tablename order by **newid()**

**18、说明：随机选择记录**select newid()

**19、说明：删除重复记录  
1),**delete from tablename where id not in (select max(id) from tablename group by col1,col2,...)  
**2)**,select distinct \* into temp from tablename  
delete from tablename  
insert into tablename select \* from temp  
**评价： 这种操作牵连大量的数据的移动，这种做法不适合大容量但数据操作  
3),例如：在一个外部表中导入数据，由于某些原因第一次只导入了一部分，但很难判断具体位置，这样只有在下一次全部导入，这样也就产生好多重复的字段，怎样删除重复字段**

alter table tablename  
--添加一个自增列  
add column\_b int identity(1,1)  
delete from tablename where column\_b not in(  
select max(column\_b) from tablename group by column1,column2,...)  
alter table tablename drop column column\_b

**20、说明：列出数据库里所有的表名**select name from sysobjects where type='U' // U代表用户

**21、说明：列出表里的所有的列名**select name from syscolumns where id=object\_id('TableName')

**22、说明：列示type、vender、pcs字段，以type字段排列，case可以方便地实现多重选择，类似select 中的case。**select type,sum(case vender when 'A' then pcs else 0 end),sum(case vender when 'C' then pcs else 0 end),sum(case vender when 'B' then pcs else 0 end) FROM tablename group by type  
**显示结果：  
type vender pcs**电脑 A 1  
电脑 A 1  
光盘 B 2  
光盘 A 2  
手机 B 3  
手机 C 3

**23、说明：初始化表table1**

TRUNCATE TABLE table1

1. **说明：选择从10到15的记录**select top 5 \* from (select top 15 \* from table order by id asc) table\_别名 order by id desc

**三、技巧**

**1、1=1，1=2的使用，在SQL语句组合时用的较多**

**“where 1=1”** 是表示选择全部    “where 1=2”全部不选，  
如：  
if @strWhere !=''   
begin  
set @strSQL = 'select count(\*) as Total from [' + @tblName + '] where ' + @strWhere   
end  
else   
begin  
set @strSQL = 'select count(\*) as Total from [' + @tblName + ']'   
end

**我们可以直接写成**

错误！未找到目录项。 **set @strSQL = 'select count(\*) as Total from [' + @tblName + '] where 1=1** 安定 '+ @strWhere 2、收缩数据库  
--重建索引  
DBCC REINDEX  
DBCC INDEXDEFRAG  
--收缩数据和日志  
DBCC SHRINKDB  
DBCC SHRINKFILE

**3、压缩数据库**dbcc shrinkdatabase(dbname)

**4、转移数据库给新用户以已存在用户权限**exec sp\_change\_users\_login 'update\_one','newname','oldname'  
go

**5、检查备份集**RESTORE VERIFYONLY from disk='E:\dvbbs.bak'

**6、修复数据库**ALTER DATABASE [dvbbs] SET SINGLE\_USER  
GO  
DBCC CHECKDB('dvbbs',repair\_allow\_data\_loss) WITH TABLOCK  
GO  
ALTER DATABASE [dvbbs] SET MULTI\_USER  
GO

**7、日志清除**SET NOCOUNT ON  
DECLARE @LogicalFileName sysname,  
@MaxMinutes INT,  
@NewSize INT

USE tablename -- 要操作的数据库名  
SELECT  @LogicalFileName = 'tablename\_log', -- 日志文件名  
@MaxMinutes = 10, -- Limit on time allowed to wrap log.  
@NewSize = 1  -- 你想设定的日志文件的大小(M)

Setup / initialize  
DECLARE @OriginalSize int  
SELECT @OriginalSize = size   
FROM sysfiles  
WHERE name = @LogicalFileName  
SELECT 'Original Size of ' + db\_name() + ' LOG is ' +   
CONVERT(VARCHAR(30),@OriginalSize) + ' 8K pages or ' +   
CONVERT(VARCHAR(30),(@OriginalSize\*8/1024)) + 'MB'  
FROM sysfiles  
WHERE name = @LogicalFileName  
CREATE TABLE DummyTrans  
(DummyColumn char (8000) not null)

DECLARE @Counter    INT,  
@StartTime DATETIME,  
@TruncLog   VARCHAR(255)  
SELECT @StartTime = GETDATE(),  
@TruncLog = 'BACKUP LOG ' + db\_name() + ' WITH TRUNCATE\_ONLY'

DBCC SHRINKFILE (@LogicalFileName, @NewSize)  
EXEC (@TruncLog)  
-- Wrap the log if necessary.  
WHILE @MaxMinutes > DATEDIFF (mi, @StartTime, GETDATE()) -- time has not expired  
AND @OriginalSize = (SELECT size FROM sysfiles WHERE name = @LogicalFileName)    
AND (@OriginalSize \* 8 /1024) > @NewSize    
BEGIN -- Outer loop.  
SELECT @Counter = 0  
WHILE   ((@Counter < @OriginalSize / 16) AND (@Counter < 50000))  
BEGIN -- update  
INSERT DummyTrans VALUES ('Fill Log') DELETE DummyTrans  
SELECT @Counter = @Counter + 1  
END  
EXEC (@TruncLog)    
END  
SELECT 'Final Size of ' + db\_name() + ' LOG is ' +  
CONVERT(VARCHAR(30),size) + ' 8K pages or ' +   
CONVERT(VARCHAR(30),(size\*8/1024)) + 'MB'  
FROM sysfiles   
WHERE name = @LogicalFileName  
DROP TABLE DummyTrans  
SET NOCOUNT OFF

**8、说明：更改某个表**exec sp\_changeobjectowner 'tablename','dbo'

**9、存储更改全部表**

CREATE PROCEDURE dbo.User\_ChangeObjectOwnerBatch  
@OldOwner as NVARCHAR(128),  
@NewOwner as NVARCHAR(128)  
AS

DECLARE @Name    as NVARCHAR(128)  
DECLARE @Owner   as NVARCHAR(128)  
DECLARE @OwnerName   as NVARCHAR(128)

DECLARE curObject CURSOR FOR   
select 'Name'    = name,  
   'Owner'    = user\_name(uid)  
from sysobjects  
where user\_name(uid)=@OldOwner  
order by name

OPEN   curObject  
FETCH NEXT FROM curObject INTO @Name, @Owner  
WHILE(@@FETCH\_STATUS=0)  
BEGIN       
if @Owner=@OldOwner   
begin  
   set @OwnerName = @OldOwner + '.' + rtrim(@Name)  
   exec sp\_changeobjectowner @OwnerName, @NewOwner  
end  
-- select @name,@NewOwner,@OldOwner

FETCH NEXT FROM curObject INTO @Name, @Owner  
END

close curObject  
deallocate curObject  
GO

**10、SQL SERVER中直接循环写入数据**declare @i int  
set @i=1  
while @i<30  
begin  
insert into test (userid) values(@i)  
set @i=@i+1  
end  
**案例：  
有如下表，要求就裱中所有沒有及格的成績，在每次增長0.1的基礎上，使他們剛好及格:**

Name score

Zhangshan 80

Lishi  59

Wangwu  50

Songquan 69

**while((select** **min(score) from tb\_table)<60)**

**begin**

**update tb\_table set score** **=score\*1.01**

**where score<60**

**if (select** **min(score) from tb\_table)>60**

**break**

**else**

**continue**

**end**

**数据开发-经典**

**1.按姓氏笔画排序:**  
Select \* From TableName Order By CustomerName Collate Chinese\_PRC\_Stroke\_ci\_as //从少到多

**2.数据库加密:**select encrypt('原始密码')  
select pwdencrypt('原始密码')  
select pwdcompare('原始密码','加密后密码') = 1--相同；否则不相同 encrypt('原始密码')  
select pwdencrypt('原始密码')  
select pwdcompare('原始密码','加密后密码') = 1--相同；否则不相同

**3.取回表中字段:**declare @list varchar(1000),  
@sql nvarchar(1000)   
select @list=@list+','+b.name from sysobjects a,syscolumns b where a.id=b.id and a.name='表A'  
set @sql='select '+right(@list,len(@list)-1)+' from 表A'   
exec (@sql)

**4.查看硬盘分区:  
EXEC master..xp\_fixeddrives**

**5.比较A,B表是否相等:**if (select checksum\_agg(binary\_checksum(\*)) from A)  
     =  
    (select checksum\_agg(binary\_checksum(\*)) from B)  
print '相等'  
else  
print '不相等'

**6.杀掉所有的事件探察器进程:**DECLARE hcforeach CURSOR GLOBAL FOR SELECT 'kill '+RTRIM(spid) FROM master.dbo.sysprocesses  
WHERE program\_name IN('SQL profiler',N'SQL 事件探查器')  
EXEC sp\_msforeach\_worker '?'

**7.记录搜索:  
开头到N条记录**Select Top N \* From 表  
-------------------------------  
**N到M条记录(要有主索引ID)**Select Top M-N \* From 表 Where ID in (Select Top M ID From 表) Order by ID   Desc  
----------------------------------  
**N到结尾记录**Select Top N \* From 表 Order by ID Desc  
**案例  
例如1：一张表有一万多条记录，表的第一个字段 RecID 是自增长字段， 写一个SQL语句， 找出表的第31到第40个记录。**

select top 10 recid from A where recid not in(select top 30 recid from A)

分析：如果这样写会产生某些问题，如果recid在表中存在逻辑索引。

select top 10 recid from A where……是从索引中查找，而后面的select top 30 recid from A则在数据表中查找，这样由于索引中的顺序有可能和数据表中的不一致，这样就导致查询到的不是本来的欲得到的数据。

**解决方案**

**1，** **用order by** select top 30 recid from A order by ricid **如果该字段不是自增长，就会出现问题**

**2，** **在那个子查询中也加条件：**select top 30 recid from A where recid>-1

**例2：查询表中的最后以条记录，并不知道这个表共有多少数据,以及表结构。  
set** @s = 'select top 1 \* from T where pid not in (select top ' + str(@count-1) + ' pid from  T)'

**print** @s exec sp\_executesql @s

**9：获取当前数据库中的所有用户表**  
select Name from sysobjects where xtype='u' and status>=0

**10：获取某一个表的所有字段**select name from syscolumns where id=object\_id('表名')

select name from syscolumns where id in (select id from sysobjects where type = 'u' and name = '表名')

两种方式的效果相同

**11：查看与某一个表相关的视图、存储过程、函数**select a.\* from sysobjects a, syscomments b where a.id = b.id and b.text like '%表名%'

**12：查看当前数据库中所有存储过程**select name as 存储过程名称 from sysobjects where xtype='P'

**13：查询用户创建的所有数据库**  
select \* from master..sysdatabases D where sid not in(select sid from master..syslogins where name='sa')  
或者  
select dbid, name AS DB\_NAME from master..sysdatabases where sid <> 0x01

**14：查询某一个表的字段和数据类型**select column\_name,data\_type from information\_schema.columns  
where table\_name = '表名'

**15：不同服务器数据库之间的数据操作**

**--创建链接服务器**

exec sp\_addlinkedserver 'ITSV ', ' ', 'SQLOLEDB ', '远程服务器名或ip地址 '

exec sp\_addlinkedsrvlogin 'ITSV ', 'false ',null, '用户名 ', '密码 '

--查询示例

select \* from ITSV.数据库名.dbo.表名

--导入示例

select \* into 表 from ITSV.数据库名.dbo.表名

**--以后不再使用时删除链接服务器**

exec sp\_dropserver 'ITSV ', 'droplogins '

**--连接远程/局域网数据**(openrowset/openquery/opendatasource)

--1、openrowset

--查询示例

select \* from openrowset( 'SQLOLEDB ', 'sql服务器名 '; '用户名 '; '密码 ',数据库名.dbo.表名)

--生成本地表

select \* into 表 from openrowset( 'SQLOLEDB ', 'sql服务器名 '; '用户名 '; '密码 ',数据库名.dbo.表名)

--把本地表导入远程表

insert openrowset( 'SQLOLEDB ', 'sql服务器名 '; '用户名 '; '密码 ',数据库名.dbo.表名)

select \*from 本地表

--更新本地表

update b

set b.列A=a.列A

from openrowset( 'SQLOLEDB ', 'sql服务器名 '; '用户名 '; '密码 ',数据库名.dbo.表名)as a inner join 本地表 b

on a.column1=b.column1

--openquery用法需要创建一个连接

--首先创建一个连接创建链接服务器

exec sp\_addlinkedserver 'ITSV ', ' ', 'SQLOLEDB ', '远程服务器名或ip地址 '

--查询

select \*

FROM openquery(ITSV, 'SELECT \* FROM 数据库.dbo.表名 ')

--把本地表导入远程表

insert openquery(ITSV, 'SELECT \* FROM 数据库.dbo.表名 ')

select \* from 本地表

--更新本地表

update b

set b.列B=a.列B

FROM openquery(ITSV, 'SELECT \* FROM 数据库.dbo.表名 ') as a

inner join 本地表 b on a.列A=b.列A

--3、opendatasource/openrowset

SELECT \*

FROM opendatasource( 'SQLOLEDB ', 'Data Source=ip/ServerName;User ID=登陆名;Password=密码 ' ).test.dbo.roy\_ta

--把本地表导入远程表

insert opendatasource( 'SQLOLEDB ', 'Data Source=ip/ServerName;User ID=登陆名;Password=密码 ').数据库.dbo.表名

select \* from 本地表

**SQL Server基本函数**

SQL Server基本函数

**1.字符串函数** 长度与分析用

**1,datalength**(Char\_expr) 返回字符串包含字符数,但不包含后面的空格  
**2,substring**(expression,start,length) 取子串，字符串的下标是从“1”，start为起始位置，length为字符串长度，实际应用中以len(expression)取得其长度  
**3,right**(char\_expr,int\_expr) 返回字符串右边第int\_expr个字符，还用**left**于之相反  
**4,isnull(** check**\_**expression **,** replacement\_value **)**如果check\_expression為空，則返回replacement\_value的值，不為空，就返回check\_expression字符操作类

**5,Sp\_addtype** 自定義數據類型  
例如：EXEC sp\_addtype birthday, datetime, 'NULL'

**6,set nocount {on|off}**

使返回的结果中不包含有关受 Transact-SQL 语句影响的行数的信息。如果存储过程中包含的一些语句并不返回许多实际的数据，则该设置由于大量减少了网络流量，因此可显著提高性能。SET NOCOUNT 设置是在执行或运行时设置，而不是在分析时设置。

SET NOCOUNT 为 ON 时，不返回计数（表示受 Transact-SQL 语句影响的行数）。  
SET NOCOUNT 为 OFF 时，返回计数

**常识**

**在SQL查询中：from**后最多可以跟多少张表或视图：**256**

**在SQL语句中出现 Order by,查询时，先排序，后取**

**在SQL中，一个字段的最大容量是8000，而对于nvarchar(4000),由于nvarchar是Unicode码。**

**SQLServer2000同步复制技术实现步骤**