sql 面试大全

下列语句部分是 Mssql 语句，不可以在 access 中使用。

SQL 分类：
DDL—数据定义语言(CREATE，ALTER，DROP，DECLARE)
DML—数据操纵语言(SELECT，DELETE，UPDATE，INSERT)
DCL—数据控制语言(GRANT，REVOKE，COMMIT，ROLLBACK) 首先,简要介绍基础语句：
1、说明：创建数据库
CREATE DATABASE database-name

2、说明：删除数据库
drop database dbname

3、说明：备份 sql server

--- 创建 备份数据的 device

USE master
EXEC sp_addumpdevice 'disk', 'testBack', 'c:mssql7backupMyNwind_1.dat'

--- 开始 备份

BACKUP DATABASE pubs TO testBack

4、说明：创建新表
create table tabname(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 tabname

6、说明：增加一个列

Alter table tabname 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 tabname add primary key(col) 说明：删除主键： Alter table tabname drop primary key(col)

8、说明：创建索引：create [unique] index idxname on tabname(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) s(1,2)删除：delete from table1 where 范围更新：update table1 set field1=1 where 范围查找：select * from table1 where field1 like ’%1%’---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 sum from table1 平均：select avg(field1) as avg from table1 最大：select max(field1) as max from table1 最小：select min(field1) as min 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 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。

不错的 sql 语句
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 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

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
Delete from tablename where id not in (select max(id) from tablename group by col1,col2,...)

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

21、说明：列出表里的所有的
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

24、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

sql 技巧

如何删除一个表中重复的记录？
create table a_dist(id int,name varchar(20))

insert into a_dist values(1,'abc')
insert into a_dist values(1,'abc')
insert into a_dist values(1,'abc')
insert into a_dist values(1,'abc')

exec up_distinct 'a_dist','id'

select * from a_dist

create procedure up_distinct(@t_name varchar(30),@f_key varchar(30))
--f_key 表示是分组字段，即主键字段
as
begin
declare @max integer,@id varchar(30) ,@sql varchar(7999) ,@type integer
select @sql = 'declare cur_rows cursor for select '+@f_key+' ,count(*) from ' +@t_name +' group by ' +@f_key +' having count(*) > 1'
exec(@sql)

```
open cur_rows
fetch cur_rows into @id,@max
while @@fetch_status=0
begin
select @max = @max -1
set rowcount @max
select @type = xtype from syscolumns where id=object_id(@t_name) and name=@f_key
if @type=56
select @sql = 'delete from '+@t_name+' where ' + @f_key+' = '+ @id
if @type=167
select @sql = 'delete from '+@t_name+' where ' + @f_key+' = '+""+ @id +""
exec(@sql)
fetch cur_rows into @id,@max
end
close cur_rows
deallocate cur_rows
set rowcount 0
end
```

```
select * from systypes
select * from syscolumns where id = object_id('a_dist')
```

查询数据的最大排序问题（只能用一条语句写）

```
CREATE TABLE hard (qu char (11) ,co char (11) ,je numeric(3, 0))
```

```
insert into hard values ('A','1',3)
insert into hard values ('A','2',4)
insert into hard values ('A','4',2)
insert into hard values ('A','6',9)
insert into hard values ('B','1',4)
insert into hard values ('B','2',5)
insert into hard values ('B','3',6)
insert into hard values ('C','3',4)
insert into hard values ('C','6',7)
insert into hard values ('C','2',3)
```

要求查询出来的结果如下：

qu co je
---------- ---------- -----
A 6 9
A 2 4
B 3 6
B 2 5
C 6 7
C 3 4

就是要按 qu 分组，每组中取 je 最大的前 2 位！！
而且只能用一句 sql 语句！！！
select * from hard a where je in (select top 2 je from hard b where a.qu=b.qu order by je)

求删除重复记录的 sql 语句？
怎样把具有相同字段的纪录删除，只留下一条。
例如，表 test 里有 id,name 字段

如果有 name 相同的记录 只留下一条，其余的删除。

name 的内容不定，相同的记录数不定。
有没有这样的 sql 语句？
==========
A:一个完整的解决方案：

将重复的记录记入 temp1 表:
select [标志字段 id],count(*) into temp1 from [表名]
group by [标志字段 id]
having count(*)>1

2、将不重复的记录记入 temp1 表:
insert temp1 select [标志字段 id],count(*) from [表名] group by [标志字段 id] having count(*)=1

3、作一个包含所有不重复记录的表：
select * into temp2 from [表名] where 标志字段 id in(select 标志字段 id from temp1)

4、删除重复表:
delete [表名]

5、恢复表：
insert [表名] select * from temp2

6、删除临时表:
drop table temp1
drop table temp2
==========
B:
create table a_dist(id int,name varchar(20))

insert into a_dist values(1,'abc')
insert into a_dist values(1,'abc')
insert into a_dist values(1,'abc')
insert into a_dist values(1,'abc')

exec up_distinct 'a_dist','id'

select * from a_dist

create procedure up_distinct(@t_name varchar(30),@f_key varchar(30))

```
--f_key 表示是分组字段，即主键字段
as
begin
declare @max integer,@id varchar(30) ,@sql varchar(7999) ,@type integer
select @sql = 'declare cur_rows cursor for select '+@f_key+' ,count(*) from ' +@t_name +' group
by ' +@f_key +' having count(*) > 1'
exec(@sql)
open cur_rows
fetch cur_rows into @id,@max
while @@fetch_status=0
begin
select @max = @max -1
set rowcount @max
select @type = xtype from syscolumns where id=object_id(@t_name) and name=@f_key
if @type=56
select @sql = 'delete from '+@t_name+' where ' + @f_key+' = '+ @id
if @type=167
select @sql = 'delete from '+@t_name+' where ' + @f_key+' = '+""+ @id +""
exec(@sql)
fetch cur_rows into @id,@max
end
close cur_rows
deallocate cur_rows
set rowcount 0
end
```

```
select * from systypes
select * from syscolumns where id = object_id('a_dist')
```

行列转换--普通

假设有张学生成绩表(CJ)如下

Name Subject Result

张三 语文 80

张三 数学 90

张三 物理 85

李四 语文 85

李四 数学 92

李四 物理 82

想变成

姓名 语文 数学 物理

张三 80 90 85
李四 85 92 82

```
declare @sql varchar(4000)
set @sql = 'select Name'
select @sql = @sql + ',sum(case Subject when '''+Subject+''' then Result end) ['+Subject+']'
from (select distinct Subject from CJ) as a
select @sql = @sql+' from test group by name'
exec(@sql)
```

行列转换--合并

有表 A,
id pid
1 1
1 2
1 3
2 1
2 2
3 1
如何化成表 B:
id pid
1 1,2,3
2 1,2
3 1

创建一个合并的函数

```
create function fmerg(@id int)
returns varchar(8000)
as
begin
declare @str varchar(8000)
set @str=''
select @str=@str+','+cast(pid as varchar) from 表 A where id=@id
set @str=right(@str,len(@str)-1)
return(@str)
End
go
```

```
--调用自定义函数得到结果
select distinct id,dbo.fmerg(id) from 表 A
```

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：

```
declare @objid int,@objname char(40)
set @objname = 'tablename'
select @objid = id from sysobjects where id = object_id(@objname)
select 'Column_name' = name from syscolumns where id = @objid order by colid
```

或

```
SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME ='users'
```

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

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=tablename
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='你的字段名字'
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))。

获取表结构[把 'sysobjects' 替换 成 'tablename' 即可]

```
SELECT CASE IsNull(I.name, '')
When '' Then ''
Else '*'
End as IsPK,
Object_Name(A.id) as t_name,
A.name as c_name,
IsNull(SubString(M.text, 1, 254), '') as pbc_init,
T.name as F_DataType,
CASE IsNull(TYPEPROPERTY(T.name, 'Scale'), '')
WHEN '' Then Cast(A.prec as varchar)
ELSE Cast(A.prec as varchar) + ',' + Cast(A.scale as varchar)
END as F_Scale,
```

```
A.isnullable as F_isNullAble
FROM Syscolumns as A
JOIN Systypes as T
ON (A.xType = T.xUserType AND A.Id = Object_id('sysobjects') )
LEFT JOIN ( SysIndexes as I
JOIN Syscolumns as A1
ON ( I.id = A1.id and A1.id = object_id('sysobjects') and (I.status & 0x800) = 0x800 AND
A1.colid <= I.keycnt) )
ON ( A.id = I.id AND A.name = index_col('sysobjects', I.indid, A1.colid) )
LEFT JOIN SysComments as M
ON ( M.id = A.cdefault and ObjectProperty(A.cdefault, 'IsConstraint') = 1 )
ORDER BY A.Colid ASC
```

提取数据库内所有表的字段详细说明的 SQL 语句

```
SELECT
(case when a.colorder=1 then d.name else '' end) N'表名',
a.colorder N'字段序号',
a.name N'字段名',
(case when COLUMNPROPERTY( a.id,a.name,'IsIdentity')=1 then '√'else ''
end) N'标识',
(case when (SELECT count(*)
FROM sysobjects
WHERE (name in
(SELECT name
FROM sysindexes
WHERE (id = a.id) AND (indid in
(SELECT indid
FROM sysindexkeys
WHERE (id = a.id) AND (colid in
(SELECT colid
FROM syscolumns
WHERE (id = a.id) AND (name = a.name)))))) AND
(xtype = 'PK'))>0 then '√' else '' end) N'主键',
b.name N'类型',
a.length N'占用字节数',
COLUMNPROPERTY(a.id,a.name,'PRECISION') as N'长度',
isnull(COLUMNPROPERTY(a.id,a.name,'Scale'),0) as N'小数位数',
```

```
(case when a.isnullable=1 then '√'else '' end) N'允许空',
isnull(e.text,'') N'默认值',
isnull(g.[value],'') AS N'字段说明'
FROM syscolumns a
left join systypes b
on a.xtype=b.xusertype
inner join sysobjects d
on a.id=d.id and d.xtype='U' and d.name<>'dtproperties'
left join syscomments e
on a.cdefault=e.id
left join sysproperties g
on a.id=g.id AND a.colid = g.smallid
order by object_name(a.id),a.colorder
```

快速获取表 test 的记录总数[对大容量表非常有效]

快速获取表 test 的记录总数:

```
select rows from sysindexes where id = object_id('test') and indid in (0,1)
```

update 2 set KHXH=(ID+1)\2 2 行递增编号

```
update [23] set id1 = 'No.'+right('00000000'+id,6) where id not like 'No%' //递增
update [23] set id1= 'No.'+right('00000000'+replace(id1,'No.',''),6) //补位递增
delete from [1] where (id%2)=1
```

奇数

替换表名字段

```
update [1] set domurl = replace(domurl,'Upload/Imgswf/','Upload/Photo/') where domurl like
'%Upload/Imgswf/%'
```

截位

```
SELECT LEFT(表名, 5)
```

截位

```
SELECT LEFT(表名, 5)
```

（MS SQL Server）SQL 语句导入导出大全

```
/*******  导出到 excel
EXEC master..xp_cmdshell 'bcp SettleDB.dbo.shanghu out c:\temp1.xls -c -q
-S"GNETDATA/GNETDATA" -U"sa" -P""'
```

/********** 导入 Excel
SELECT *
FROM OpenDataSource('Microsoft.Jet.OLEDB.4.0',
  'Data Source="c:\test.xls";User ID=Admin;Password=;Extended properties=Excel
5.0')...xactions

SELECT cast(cast(科目编号 as numeric(10,2)) as nvarchar(255))+' ' 转换后的别名
FROM OpenDataSource('Microsoft.Jet.OLEDB.4.0',
  'Data Source="c:\test.xls";User ID=Admin;Password=;Extended properties=Excel
5.0')...xactions

select * from OPENROWSET('MICROSOFT.JET.OLEDB.4.0','Excel
5.0;HDR=YES;DATABASE=c:\Book1.xls',Sheet1$)

HDR=YES;Excel 第一行当成标题行

HDR=NO;第一行不当成标题行

/** 导入文本文件
EXEC master..xp_cmdshell 'bcp "dbname..tablename" in c:\DT.txt -c -Sservername -Usa
-Ppassword'

/** 导出文本文件
EXEC master..xp_cmdshell 'bcp "dbname..tablename" out c:\DT.txt -c -Sservername -Usa
-Ppassword'

或

```
EXEC master..xp_cmdshell 'bcp "Select * from dbname..tablename" queryout c:\DT.txt -c
-Sservername -Usa -Ppassword'
```

导出到 TXT 文本，用逗号分开

```
exec master..xp_cmdshell 'bcp "库名..表名" out "d:\tt.txt" -c -t ,-U sa -P password'
```

```
BULK INSERT 库名..表名
FROM 'c:\test.txt'
WITH (
    FIELDTERMINATOR = ';',
    ROWTERMINATOR = '\n'
)
```

```
--/* dBase IV 文件
select * from
OPENROWSET('MICROSOFT.JET.OLEDB.4.0'
,'dBase IV;HDR=NO;IMEX=2;DATABASE=C:\','select * from [客户资料 4.dbf]')
--*/
```

```
--/* dBase III 文件
select * from
OPENROWSET('MICROSOFT.JET.OLEDB.4.0'
,'dBase III;HDR=NO;IMEX=2;DATABASE=C:\','select * from [客户资料 3.dbf]')
--*/
```

```
--/* FoxPro 数据库
select * from openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;SourceType=DBF;SourceDB=c:\',
'select * from [aa.DBF]')
```

--*/

/**************导入 DBF 文件*****************/
select * from openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data;
SourceType=DBF',
'select * from customer where country != "USA" order by country')
go
/***************** 导出到 DBF ***************/
如果要导出数据到已经生成结构(即现存的)FOXPRO 表中,可以直接用下面的 SQL 语句

insert into openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;SourceType=DBF;SourceDB=c:\',
'select * from [aa.DBF]')
select * from 表

说明:
SourceDB=c:\ 指定 foxpro 表所在的文件夹
aa.DBF 指定 foxpro 表的文件名.

/**************导出到 Access*******************/
insert into openrowset('Microsoft.Jet.OLEDB.4.0',
    'x:\A.mdb';'admin';",A 表) select * from 数据库名..B 表

/**************导入 Access*******************/
insert into B 表 selet * from openrowset('Microsoft.Jet.OLEDB.4.0',
    'x:\A.mdb';'admin';",A 表)

********** 导入 xml 文件

```
DECLARE @idoc int
DECLARE @doc varchar(1000)
--sample XML document
SET @doc ='
<root>
   <Customer cid= "C1" name="Janine" city="Issaquah">
      <Order oid="O1" date="1/20/1996" amount="3.5" />
      <Order oid="O2" date="4/30/1997" amount="13.4">Customer was very satisfied
      </Order>
    </Customer>
    <Customer cid="C2" name="Ursula" city="Oelde" >
      <Order oid="O3" date="7/14/1999" amount="100" note="Wrap it blue
           white red">
          <Urgency>Important</Urgency>
          Happy Customer.
      </Order>
      <Order oid="O4" date="1/20/1996" amount="10000"/>
    </Customer>
</root>
'
-- Create an internal representation of the XML document.
EXEC sp_xml_preparedocument @idoc OUTPUT, @doc



-- Execute a SELECT statement using OPENXML rowset provider.
SELECT *
FROM OPENXML (@idoc, '/root/Customer/Order', 1)
      WITH (oid       char(5),
           amount    float,
           comment ntext 'text()')
EXEC sp_xml_removedocument @idoc




/********************导整个数据库**********************************************/
```

用 bcp 实现的存储过程

```
/*
实现数据导入/导出的存储过程
        根据不同的参数,可以实现导入/导出整个数据库/单个表
调用示例:
--导出调用示例
----导出单个表
exec file2table 'zj','','','xzkh_sa..地区资料','c:\zj.txt',1
----导出整个数据库
exec file2table 'zj','','','xzkh_sa','C:\docman',1


--导入调用示例
----导入单个表
exec file2table 'zj','','','xzkh_sa..地区资料','c:\zj.txt',0
----导入整个数据库
exec file2table 'zj','','','xzkh_sa','C:\docman',0


*/
if exists(select 1 from sysobjects where name='File2Table' and objectproperty(id,'IsProcedure')=1)
drop procedure File2Table
go
create procedure File2Table
@servername varchar(200)    --服务器名
,@username varchar(200)     --用户名,如果用 NT 验证方式,则为空''
,@password varchar(200)     --密码
,@tbname varchar(500)     --数据库.dbo.表名,如果不指定:.dbo.表名,则导出数据库的所有用户表
,@filename varchar(1000)    --导入/导出路径/文件名,如果@tbname 参数指明是导出整个数据库,则这个参数是文件存放路径,文件名自动用表名.txt
,@isout bit        --1 为导出,0 为导入
as
declare @sql varchar(8000)
```

```
if @tbname like '%.%.%' --如果指定了表名,则直接导出单个表
begin
set @sql='bcp '+@tbname
   +case when @isout=1 then ' out ' else ' in ' end
   +' "'+@filename+'" /w'
   +' /S '+@servername
   +case when isnull(@username,'')='' then '' else ' /U '+@username end
   +' /P '+isnull(@password,'')
exec master..xp_cmdshell @sql
end
else
begin --导出整个数据库,定义游标,取出所有的用户表
declare @m_tbname varchar(250)
if right(@filename,1)<>'\' set @filename=@filename+'\'



set @m_tbname='declare #tb cursor for select name from '+@tbname+'..sysobjects where
xtype="U"'
exec(@m_tbname)
open #tb
fetch next from #tb into @m_tbname
while @@fetch_status=0
begin
   set @sql='bcp '+@tbname+'..'+@m_tbname
    +case when @isout=1 then ' out ' else ' in ' end
    +' "'+@filename+@m_tbname+'.txt " /w'
    +' /S '+@servername
    +case when isnull(@username,'')='' then '' else ' /U '+@username end
    +' /P '+isnull(@password,'')
   exec master..xp_cmdshell @sql
   fetch next from #tb into @m_tbname
end
close #tb
deallocate #tb
end
go
```

/**********Excel 导到 Txt**********/

想用

select * into opendatasource(...) from opendatasource(...)
实现将一个 Excel 文件内容导入到一个文本文件

假设 Excel 中有两列，第一列为姓名，第二列为很行帐号(16 位)
且银行帐号导出到文本文件后分两部分，前 8 位和后 8 位分开。

如果要用你上面的语句插入的话,文本文件必须存在,而且有一行:姓名,银行账号 1,银行账号 2
然后就可以用下面的语句进行插入
注意文件名和目录根据你的实际情况进行修改.

```
insert into
opendatasource('MICROSOFT.JET.OLEDB.4.0'
,'Text;HDR=Yes;DATABASE=C:\'
)...[aa#txt]
--,aa#txt)
--*/
select 姓名,银行账号 1=left(银行账号,8),银行账号 2=right(银行账号,8)
from
opendatasource('MICROSOFT.JET.OLEDB.4.0'
,'Excel 5.0;HDR=YES;IMEX=2;DATABASE=c:\a.xls'
--,Sheet1$)
)...[Sheet1$]
```

如果你想直接插入并生成文本文件,就要用 bcp

```
declare @sql varchar(8000),@tbname varchar(50)

--首先将 excel 表内容导入到一个全局临时表
select @tbname='[##temp'+cast(newid() as varchar(40))+']'
,@sql='select 姓名,银行账号 1=left(银行账号,8),银行账号 2=right(银行账号,8)
```

```
into '+@tbname+' from
opendatasource("MICROSOFT.JET.OLEDB.4.0"
,"Excel 5.0;HDR=YES;IMEX=2;DATABASE=c:\a.xls"
)...[Sheet1$]'
exec(@sql)



--然后用 bcp 从全局临时表导出到文本文件
set @sql='bcp "'+@tbname+'" out "c:\aa.txt" /S"(local)" /P"" /c'
exec master..xp_cmdshell @sql



--删除临时表
exec('drop table '+@tbname)
```

用 bcp 将文件导入导出到数据库的存储过程:

```
/*--bcp-二进制文件的导入导出



支持 image,text,ntext 字段的导入/导出
image 适合于二进制文件;text,ntext 适合于文本数据文件



注意:导入时,将覆盖满足条件的所有行
  导出时,将把所有满足条件的行也出到指定文件中



此存储过程仅用 bcp 实现
邹建 2003.08----------------*/
```

```
/*--调用示例
--数据导出
exec p_binaryIO 'zj','','','acc_演示数据..tb','img','c:\zj1.dat'



--数据导出
exec p_binaryIO 'zj','','','acc_演示数据..tb','img','c:\zj1.dat','',0
--*/
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[p_binaryIO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[p_binaryIO]
GO



Create proc p_binaryIO
@servename varchar (30),--服务器名称
@username varchar (30), --用户名
@password varchar (30), --密码
@tbname varchar (500),  --数据库..表名
@fdname varchar (30),    --字段名
@fname varchar (1000), --目录+文件名,处理过程中要使用/覆盖:@filename+.bak
@tj varchar (1000)='',    --处理条件.对于数据导入,如果条件中包含@fdname,请指定表名前缀
@isout bit=1     --1 导出((默认),0 导入
AS
declare @fname_in varchar(1000) --bcp 处理应答文件名
,@fsize varchar(20)     --要处理的文件的大小
,@m_tbname varchar(50)    --临时表名
,@sql varchar(8000)



--则取得导入文件的大小
if @isout=1
set @fsize='0'
else
begin
create table #tb(可选名 varchar(20),大小 int
   ,创建日期 varchar(10),创建时间 varchar(20)
   ,上次写操作日期 varchar(10),上次写操作时间 varchar(20)
   ,上次访问日期 varchar(10),上次访问时间 varchar(20),特性 int)
```

```
insert into #tb
exec master..xp_getfiledetails @fname
select @fsize=大小 from #tb
drop table #tb
if @fsize is null
begin
   print '文件未找到'
   return
end



end



--生成数据处理应答文件
set @m_tbname='[##temp'+cast(newid() as varchar(40))+']'
set @sql='select * into '+@m_tbname+' from(
select null as 类型
union all select 0 as 前缀
union all select '+@fsize+' as 长度
union all select null as 结束
union all select null as 格式
) a'
exec(@sql)
select @fname_in=@fname+'_temp'
,@sql='bcp "'+@m_tbname+'" out "'+@fname_in
+'" /S"'+@servename
+case when isnull(@username,")=" then "
   else '" /U"'+@username end
+'" /P"'+isnull(@password,")+'" /c'
exec master..xp_cmdshell @sql
--删除临时表
set @sql='drop table '+@m_tbname
exec(@sql)



if @isout=1
begin
set @sql='bcp "select top 1 '+@fdname+' from '
   +@tbname+case isnull(@tj,") when " then "
    else ' where '+@tj end
```

```
    +'" queryout "'+@fname
    +'" /S"'+@servename
    +case when isnull(@username,")=" then "
     else '" /U"'+@username end
    +'" /P"'+isnull(@password,")
    +'" /i"'+@fname_in+'""
exec master..xp_cmdshell @sql
end
else
begin
--为数据导入准备临时表
set @sql='select top 0 '+@fdname+' into '
    +@m_tbname+' from ' +@tbname
exec(@sql)



--将数据导入到临时表
set @sql='bcp "'+@m_tbname+'" in "'+@fname
    +'" /S"'+@servename
    +case when isnull(@username,")=" then "
     else '" /U"'+@username end
    +'" /P"'+isnull(@password,")
    +'" /i"'+@fname_in+'""
exec master..xp_cmdshell @sql



--将数据导入到正式表中
set @sql='update '+@tbname
    +' set '+@fdname+'=b.'+@fdname
    +' from '+@tbname+' a,'
    +@m_tbname+' b'
    +case isnull(@tj,") when " then "
     else ' where '+@tj end
exec(@sql)



--删除数据处理临时表
set @sql='drop table '+@m_tbname
end
```

--删除数据处理应答文件
set @sql='del '+@fname_in
exec master..xp_cmdshell @sql

go

/** 导入文本文件
EXEC master..xp_cmdshell 'bcp "dbname..tablename" in c:\DT.txt -c -Sservername -Usa -Ppassword'

改为如下，不需引号
EXEC master..xp_cmdshell 'bcp dbname..tablename in c:\DT.txt -c -Sservername -Usa -Ppassword'

/** 导出文本文件
EXEC master..xp_cmdshell 'bcp "dbname..tablename" out c:\DT.txt -c -Sservername -Usa -Ppassword'
此句需加引号

sql 常见面试题

sql 理论题

1．触发器的作用？

答：触发器是一中特殊的存储过程，主要是通过事件来触发而被执行的。它可以强化约束，来维护数据的完整性和一致性，可以跟踪数据库内的操作从而不允许未经许可的更新和变化。可以联级运算。如，某表上的触发器上包含对另一个表的数据操作，而该操作又会导致该表触发器被触发。

2。**什么是存储过程？用什么来调用？**

答：存储过程是一个预编译的 SQL 语句，优点是允许模块化的设计，就是说只需创建一次，以后在该程序中就可以调用多次。如果某次操作需要执行多次 SQL，使用存储过程比单纯 SQL 语句执行要快。可以用一个命令对象来调用存储过程。

3。索引的作用？和它的优点缺点是什么？
答：索引就一种特殊的查询表，数据库的搜索引擎可以利用它加速对数据的检索。它很类似与现实生活中书的目录，不需要查询整本书内容就可以找到想要的数据。索引可以是唯一的，创建索引允许指定单个列或者是多个列。缺点是它减慢了数据录入的速度，同时也增加了数据库的尺寸大小。

3。什么是内存泄漏？
答：一般我们所说的内存泄漏指的是堆内存的泄漏。堆内存是程序从堆中为其分配的，大小任意的，使用完后要显示释放内存。当应用程序用关键字 new 等创建对象时，就从堆中为它分配一块内存，使用完后程序调用 free 或者 delete 释放该内存，否则就说该内存就不能被使用，我们就说该内存被泄漏了。

4。**维护数据库的完整性和一致性，你喜欢用触发器还是自写业务逻辑？为什么？**

答：我是这样做的，尽可能使用约束，如 check,主键，外键，非空字段等来约束，这样做效率最高，也最方便。其次是使用触发器，这种方法可以保证，无论什么业务系统访问数据库都可以保证数据的完整新和一致性。最后考虑的是自写业务逻辑，但这样做麻烦，编程复杂，效率低下。

5。**什么是事务？什么是锁？**
答：事务就是被绑定在一起作为一个逻辑工作单元的 SQL 语句分组，如果任何一个语句操作失败那么整个操作就被失败，以后操作就会回滚到操作前状态，或者是上有个节点。为了确保要么执行，要么不执行，就可以使用事务。要将有组语句作为事务考虑，就需要通过 ACID 测试，即原子性，一致性，隔离性和持久性。

锁：在所以的 DBMS 中，锁是实现事务的关键，锁可以保证事务的完整性和并发性。与现实生活中锁一样，它可以使某些数据的拥有者，在某段时间内不能使用某些数据或数据结构。当然锁还分级别的。

6。**什么叫视图？游标是什么**？

答：**视图**是一种虚拟的表，具有和物理表相同的功能。可以对视图进行增，改，查，操作，试图通常是有一个表或者多个表的行或列的子集。对视图的修改不影响基本表。它使得我们获取数据更容易，相比多表查询。

游标：是对查询出来的结果集作为一个单元来有效的处理。游标可以定在该单元中的特定行，从结果集的当前行检索一行或多行。可以对结果集当前行做修改。一般不使用游标，但是需要逐条处理数据的时候，游标显得十分重要。

7。为管理业务培训信息，建立 3 个表：

S(S#,SN,SD,SA)S#,SN,SD,SA 分别代表学号，学员姓名，所属单位，学员年龄

C(C#,CN)C#,CN 分别代表课程编号，课程名称

SC(S#,C#,G) S#,C#,G 分别代表学号，所选的课程编号，学习成绩

（1）使用标准 SQL 嵌套语句查询选修课程名称为’税收基础’的学员学号和姓名?

答案：select s# ,sn from s where S# in(select S# from c,sc where c.c#=sc.c# and cn=’税收基础’)

(2) 使用标准 SQL 嵌套语句查询选修课程编号为’C2’的学员姓名和所属单位?

答：select sn,sd from s,sc where s.s#=sc.s# and sc.c#=’c2’

(3) 使用标准 SQL 嵌套语句查询不选修课程编号为’C5’的学员姓名和所属单位?

答：select sn,sd from s where s# not in(select s# from sc where c#=’c5’)

(4)查询选修了课程的学员人数

答：select 学员人数=count(distinct s#) from sc

(5) 查询选修课程超过 5 门的学员学号和所属单位?

答：select sn,sd from s where s# in(select s# from sc group by s# having count(distinct c#)>5)

是查询 A(ID,Name)表中第 31 至 40 条记录，ID 作为主键可能是不是连续增长的列，完整的查询语句如下：

```
select top 10 * from A where ID >(select max(ID) from (select top 30 ID from A order by A

) T) order by A
```

要求是查询表 A 中存在 ID 重复三次以上的记录,完整的查询语句如下：

```
select * from(select count(ID) as count from table group by ID)T where T.count>3
```

```
create table testtable1
(
 id int IDENTITY,
 department varchar(12)
)

select * from testtable1
insert into testtable1 values('设计')
insert into testtable1 values('市场')
insert into testtable1 values('售后')
/*
结果
id department
1   设计
2   市场
3   售后
*/
create table testtable2
(
 id int IDENTITY,
 dptID int,
 name varchar(12)
```

```
)
insert into testtable2 values(1,'张三')
insert into testtable2 values(1,'李四')
insert into testtable2 values(2,'王五')
insert into testtable2 values(3,'彭六')
insert into testtable2 values(4,'陈七')
/*
用一条 SQL 语句，怎么显示如下结果
id dptID department name
1   1     设计       张三
2   1     设计       李四
3   2     市场       王五
4   3     售后       彭六
5   4     黑人       陈七
*/
```

答案是：

```
SELECT testtable2.* , ISNULL(department,'黑人')
FROM testtable1 right join testtable2 on testtable2.dptID = testtable1.ID
```

在面试应聘的 SQL Server 数据库开发人员时，我运用了一套标准的基准技术问题。下面这些问题是我觉得能够真正有助于淘汰不合格应聘者的问题。它们按照从易到难的顺序排列。当你问到关于主键和外键的问题时，后面的问题都十分有难度，因为答案可能会更难解释和说明，尤其是在面试的情形下。

你能向我简要叙述一下 SQL Server 2000 中使用的一些数据库对象吗?

你希望听到的答案包括这样一些对象:表格、视图、用户定义的函数，以及存储过程;如果他们还能够提到像触发器这样的对象就更好了。如果应聘者不能回答这个基本的问题，那么这不是一个好兆头。

NULL 是什么意思?

NULL(空)这个值是数据库世界里一个非常难缠的东西，所以有不少应聘者会在这个问题上跌跟头您也不要觉得意外。NULL 这个值表示 UNKNOWN(未知):它不表示“”(空字符串)。假设您的 SQL Server 数据库里有 ANSI_NULLS，当然在默认情况下会有，对 NULL 这个值的任何比较都会生产一个 NULL 值。您不能把任何值与一个 UNKNOWN 值进行比较，并在逻辑上希望获得一个答案。您必须使用 IS NULL 操作符。

什么是索引?SQL Server 2000 里有什么类型的索引?

任何有经验的数据库开发人员都应该能够很轻易地回答这个问题。一些经验不太多的开发人员能够回答这个问题，但是有些地方会说不清楚。简单地说，索引是一个数据结构，用来快速访问数据库表格或者视图里的数据。在 SQL Server 里，它们有两种形式:聚集索引和非聚集索引。聚集索引在索引的叶级保存数据。这意味着不论聚集索引里有表格的哪个(或哪些)字段，这些字段都会按顺序被保存在表格。由于存在这种排序，所以每个表格只会有一个聚集索引。非聚集索引在索引的叶级有一个行标识符。这个行标识符是一个指向磁盘上数据的指针。它允许每个表格有多个非聚集索引。

什么是主键?什么是外键?

主键是表格里的(一个或多个)字段，只用来定义表格里的行;主键里的值总是唯一的。外键是一个用来建立两个表格之间关系的约束。这种关系一般都涉及一个表格里的主键字段与另外一个表格(尽管可能是同一个表格)里的一系列相连的字段。那么这些相连的字段就是外键。

什么是触发器?SQL Server 2000 有什么不同类型的触发器?

让未来的数据库开发人员知道可用的触发器类型以及如何实现它们是非常有益的。触发器是一种专用类型的存储过程，它被捆绑到 SQL Server 2000 的表格或者视图上。在 SQL Server

2000 里，有 INSTEAD-OF 和 AFTER 两种触发器。INSTEAD-OF 触发器是替代数据操控语言(Data Manipulation

Language，DML)语句对表格执行语句的存储过程。例如，如果我有一个用于 TableA 的 INSTEAD-OF-UPDATE

触发器，同时对这个表格执行一个更新语句，那么 INSTEAD-OF-UPDATE 触发器里的代码会执行，而不是我执行的更新语句则不会执行操作。

AFTER 触发器要在 DML 语句在数据库里使用之后才执行。这些类型的触发器对于监视发生在数据库表格里的数据变化十分好用。

您如何确一个带有名为 Fld1 字段的 TableB 表格里只具有 Fld1 字段里的那些值，而这些值同时在名为 TableA 的表格的 Fld1 字段里?

这个与关系相关的问题有两个可能的答案。第一个答案(而且是您希望听到的答案)是使用外键限制。外键限制用来维护引用的完整性。它被用来确保表格里的字段只保存有已经在不同的(或者相同的)表格里的另一个字段里定义了的值。这个字段就是候选键(通常是另外一个表格的主键)。

另外一种答案是触发器。触发器可以被用来保证以另外一种方式实现与限制相同的作用，但是它非常难设置与维护，而且性能一般都很糟糕。由于这个原因，微软建议开发人员使用外键限制而不是触发器来维护引用的完整性。

对一个投入使用的在线事务处理表格有过多索引需要有什么样的性能考虑?

你正在寻找进行与数据操控有关的应聘人员。对一个表格的索引越多，数据库引擎用来更新、插入或者删除数据所需要的时间就越多，因为在数据操控发生的时候索引也必须要维护。

你可以用什么来确保表格里的字段只接受特定范围里的值?

这个问题可以用多种方式来回答，但是只有一个答案是“好”答案。您希望听到的回答是 Check 限制，它在数据库表格里被定义，用来限制输入该列的值。

触发器也可以被用来限制数据库表格里的字段能够接受的值，但是这种办法要求触发器在表格里被定义，这可能会在某些情况下影响到性能。因此，微软建议使用 Check 限制而不是其他的方式来限制域的完整性。

如果应聘者能够正确地回答这个问题，那么他的机会就非常大了，因为这表明他们具有使用存储过程的经验。

返回参数总是由存储过程返回，它用来表示存储过程是成功还是失败。返回参数总是 INT 数据类型。

OUTPUT 参数明确要求由开发人员来指定，它可以返回其他类型的数据，例如字符型和数值型的值。(可以用作输出参数的数据类型是有一些限制的。)您可以在一个存储过程里使用多个 OUTPUT 参数，而您只能够使用一个返回参数。

什么是相关子查询?如何使用这些查询?

经验更加丰富的开发人员将能够准确地描述这种类型的查询。相关子查询是一种包含子查询的特殊类型的查询。查询里包含的子查询会真正请求外部查询的值，从而形成一个类似于循环的状况。

什么是 SQL 注入式攻击?

所谓 SQL 注入式攻击，就是攻击者把 SQL 命令插入到 Web 表单的输入域或页面请求的查询字符串，欺骗服务器执行恶意的 SQL 命令。在某些表单中，用户输入的内容直接用来构造（或者影响）动态 SQL 命令，或作为存储过程的输入参数，这类表单特别容易受到 SQL 注入式攻击。常见的 SQL 注入式攻击过程类如：

⑴ 某个 ASP.NET Web 应用有一个登录页面，这个登录页面控制着用户是否有权访问应用，它要求用户输入一个名称和密码。

⑵ 登录页面中输入的内容将直接用来构造动态的 SQL 命令，或者直接用作存储过程的参数。下面是 ASP.NET 应用构造查询的一个例子：

```
System.Text.StringBuilder query = new System.Text.StringBuilder(
    "SELECT * from Users WHERE login = '")
    .Append(txtLogin.Text).Append("' AND password='")
    .Append(txtPassword.Text).Append("'");
```

⑶ 攻击者在用户名字和密码输入框中输入"'或'1'='1"之类的内容。

⑷ 用户输入的内容提交给服务器之后，服务器运行上面的 ASP.NET 代码构造出查询用户的 SQL 命令，但由于攻击者输入的内容非常特殊，所以最后得到的 SQL 命令变成：SELECT * from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'。

⑸ 服务器执行查询或存储过程，将用户输入的身份信息和服务器中保存的身份信息进行对比。

⑹ 由于 SQL 命令实际上已被注入式攻击修改，已经不能真正验证用户身份，所以系统会错误地授权给攻击者。

如果攻击者知道应用会将表单中输入的内容直接用于验证身份的查询，他就会尝试输入某些特殊的 SQL 字符串篡改查询改变其原来的功能，欺骗系统授予访问权限。

系统环境不同，攻击者可能造成的损害也不同，这主要由应用访问数据库的安全权限决定。如果用户的帐户具有管理员或其他比较高级的权限，攻击者就可能对数据库的表执行各种他想要做的操作，包括添加、删除或更新数据，甚至可能直接删除表

如何防范 SQL 注入式攻击？

好在要防止 ASP.NET 应用被 SQL 注入式攻击闯入并不是一件特别困难的事情，只要在利用表单输入的内容构造 SQL 命令之前，把所有输入内容过滤一番就可以了。过滤输入内容可以按多种方式进行。

⑴ 对于动态构造 SQL 查询的场合，可以使用下面的技术：

第一：替换单引号，即把所有单独出现的单引号改成两个单引号，防止攻击者修改 SQL 命令的含义。再来看前面的例子，"SELECT * from Users WHERE login = ''' or ''1''=''1' AND password = ''' or ''1''=''1'"显然会得到与"SELECT * from Users WHERE login = '' or '1'='1' AND password = '' or '1'='1'"不同的结果。

第二：删除用户输入内容中的所有连字符，防止攻击者构造出类如"SELECT * from Users WHERE login = 'mas' -- AND password ='"之类的查询，因为这类查询的后半部分已经被注释掉，不再有效，攻击者只要知道一个合法的用户登录名称，根本不需要知道用户的密码就可以顺利获得访问权限。

第三：对于用来执行查询的数据库帐户，限制其权限。用不同的用户帐户执行查询、插入、更新、删除操作。由于隔离了不同帐户可执行的操作，因而也就防止了原本用于执行 SELECT 命令的地方却被用于执行 INSERT、UPDATE 或 DELETE 命令。

⑵ 用存储过程来执行所有的查询。SQL 参数的传递方式将防止攻击者利用单引号和连字符实施攻击。此外，它还使得数据库权限可以限制到只允许特定的存储过程执行，所有的用户输入必须遵从被调用的存储过程的安全上下文，这样就很难再发生注入式攻击了。
⑶ 限制表单或查询字符串输入的长度。如果用户的登录名字最多只有 10 个字符，那么不要认可表单中输入的 10 个以上的字符，这将大大增加攻击者在 SQL 命令中插入有害代码的难度。
⑷ 检查用户输入的合法性，确信输入的内容只包含合法的数据。数据检查应当在客户端和服务器端都执行——之所以要执行服务器端验证，是为了弥补客户端验证机制脆弱的安全性。在客户端，攻击者完全有可能获得网页的源代码，修改验证合法性的脚本（或者直接删除脚本），然后将非法内容通过修改后的表单提交给服务器。因此，要保证验证操作确实已经执行，唯一的办法就是在服务器端也执行验证。你可以使用许多内建的验证对象，例如 RegularExpressionValidator，它们能够自动生成验证用的客户端脚本，当然你也可以插入服务器端的方法调用。如果找不到现成的验证对象，你可以通过 CustomValidator 自己创建一个。
⑸ 将用户登录名称、密码等数据加密保存。加密用户输入的数据，然后再将它与数据库中保存的数据比较，这相当于对用户输入的数据进行了"消毒"处理，用户输入的数据不再对数据库有任何特殊的意义，从而也就防止了攻击者注入 SQL 命令。
System.Web.Security.FormsAuthentication 类有一个 HashPasswordForStoringInConfigFile，非常适合于对输入数据进行消毒处理。
⑹ 检查提取数据的查询所返回的记录数量。如果程序只要求返回一个记录，但实际返回的记录却超过一行，那就当作出错处理

Sql 常见题目

为管理岗位业务培训信息，建立 3 个表:
S (S#,SN,SD,SA) S#,SN,SD,SA 分别代表学号、学员姓名、所属单位、学员年龄
C (C#,CN) C#,CN 分别代表课程编号、课程名称
SC (S#,C#,G) S#,C#,G 分别代表学号、所选修的课程编号、学习成绩
1. 使用标准 SQL 嵌套语句查询选修课程名称为’税收基础’的学员学号和姓名
--实现代码:
Select SN,SD FROM S Where [S#] IN(Select [S#] FROM C,SC Where C.[C#]=SC.[C#] AND CN=N'税收基础')
2. 使用标准 SQL 嵌套语句查询选修课程编号为’C2’的学员姓名和所属单位
--实现代码:
Select S.SN,S.SD FROM S,SC Where S.[S#]=SC.[S#] AND SC.[C#]='C2'
3. 使用标准 SQL 嵌套语句查询不选修课程编号为’C5’的学员姓名和所属单位
--实现代码:
Select SN,SD FROM S Where [S#] NOT IN(Select [S#] FROM SC Where [C#]='C5')
4. 使用标准 SQL 嵌套语句查询选修全部课程的学员姓名和所属单位
--实现代码:

Select SN,SD FROM S Where [S#] IN(Select [S#] FROM SC RIGHT JOIN C ON SC.[C#]=C.[C#] GROUP BY [S#] HAVING COUNT(*)=COUNT([S#]))

5. 查询选修了课程的学员人数

--实现代码:

Select 学员人数=COUNT(DISTINCT [S#]) FROM SC

6. 查询选修课程超过 5 门的学员学号和所属单位

--实现代码:

Select SN,SD FROM S Where [S#] IN(Select [S#] FROM SC GROUP BY [S#] HAVING COUNT(DISTINCT [C#])>5)

题目 2:

问题描述:

S (SNO,SNAME) 学生关系。SNO 为学号，SNAME 为姓名

C (CNO,CNAME,CTEACHER) 课程关系。CNO 为课程号，CNAME 为课程名，CTEACHER 为任课教师

SC(SNO,CNO,SCGRADE) 选课关系。SCGRADE 为成绩

1. 找出没有选修过“李明”老师讲授课程的所有学生姓名

--实现代码:

Select SNAME FROM S Where NOT EXISTS(Select * FROM SC,C Where SC.CNO=C.CNO AND CNAME='李明' AND SC.SNO=S.SNO)

2. 列出有二门以上(含两门)不及格课程的学生姓名及其平均成绩

--实现代码:

Select S.SNO,S.SNAME,AVG_SCGRADE=AVG(SC.SCGRADE) FROM S,SC,(Select SNO FROM SC Where SCGRADE<60 GROUP BY SNO HAVING COUNT(DISTINCT CNO)>=2)A Where S.SNO=A.SNO AND SC.SNO=A.SNO GROUP BY S.SNO,S.SNAME

3. 列出既学过“1”号课程，又学过“2”号课程的所有学生姓名

--实现代码:

Select S.SNO,S.SNAME FROM S,(Select SC.SNO FROM SC,C Where SC.CNO=C.CNO AND C.CNAME IN('1','2') GROUP BY SNO HAVING COUNT(DISTINCT CNO)=2)SC Where S.SNO=SC.SNO

4. 列出“1”号课成绩比“2”号同学该门课成绩高的所有学生的学号

--实现代码:

Select S.SNO,S.SNAME FROM S,(Select SC1.SNO FROM SC SC1,C C1,SC SC2,C C2 Where SC1.CNO=C1.CNO AND C1.NAME='1' AND SC2.CNO=C2.CNO AND C2.NAME='2' AND SC1.SCGRADE>SC2.SCGRADE)SC Where S.SNO=SC.SNO

5. 列出“1”号课成绩比“2”号课成绩高的所有学生的学号及其“1”号课和“2”号课的成绩

--实现代码:

Select S.SNO,S.SNAME,SC.[1 号课成绩],SC.[2 号课成绩] FROM S,(Select SC1.SNO,[1 号课成绩]=SC1.SCGRADE,[2 号课成绩]=SC2.SCGRADE FROM SC SC1,C C1,SC SC2,C C2 Where SC1.CNO=C1.CNO AND C1.NAME='1' AND SC2.CNO=C2.CNO AND C2.NAME='2' AND SC1.SCGRADE>SC2.SCGRADE)SC Where S.SNO=SC.SNO

求其中同一个号码的两次通话之间间隔大于 10 秒的通话记录 ID
例如：6，7，8，9，10 条记录均符合

ID 主叫号码 被叫号码　通话起始时间　　通话结束时间　　通话时长

1 98290000 0215466546656 2007-02-01 09:49:53.000 2007-02-01 09:50:16.000 23
2 98290000 021546654666 2007-02-01 09:50:29.000 2007-02-01 09:50:41.000 12
3 98290000 021546654666 2007-02-01 09:50:58.000 2007-02-01 09:51:12.000 14
4 68290900 0755133329866 2007-02-01 10:04:31.000 2007-02-01 10:07:13.000 162
5 78290000 0755255708638 2007-02-01 10:48:26.000 2007-02-01 10:49:23.000 57
6 78290000 0755821119109 2007-02-01 10:49:39.000 2007-02-01 10:52:55.000 196
7 78290000 035730928370 2007-02-01 11:30:45.000 2007-02-01 11:31:58.000 73
8 78290000 0871138889904 2007-02-01 11:33:47.000 2007-02-01 11:35:00.000 73
9 68290000 035730928379 2007-02-01 11:52:20.000 2007-02-01 11:54:56.000 156
10 68290000 0298521811199 2007-02-01 12:44:45.000 2007-02-01 12:45:04.000 19

答案：

```
SELECT DISTINCT a.* FROM dbo.hc a left join dbo.hc b
ON a.主叫号码=b.主叫号码
WHERE a.id<>b.id AND (DATEDIFF(second,a.通话起始时间,b.通话结束时间)>10 AND
DATEDIFF(second,b.通话起始时间,a.通话结束时间)>10)
```

Sql Server 关于按周统计的问题

统计 Sql Server 里一个销售明细表里某个时间段的销售额，而且要按周进行比较，以下是该语句的写法：

```
select sum(销售金额), datename(week, 销售日期-1) from sales where 销售日期 betwee
begindate and enddate group by datename(week, 销售日期-1)
```

注意：这里之所以要把销售日期-1 是因为 sql server 默认的一周的第一天是星期天，而我们习惯的统计是以星期一到星期天计算的，所以减一。