

25、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

26、说明: 前 10 条记录

select top 10 * form table1 where 范围

27、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)

28、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)

29、说明: 随机取出 10 条数据

select top 10 * from tablename order by newid()

30、说明: 随机选择记录

select newid()

31、说明: 删除重复记录

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

32、说明: 列出数据库里所有的表名

select name from sysobjects where type='U'

33、说明: 列出表里的所有的

select name from syscolumns where id=object_id('TableName')

34、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

35、说明: 初始化表 table1

TRUNCATE TABLE table1

36、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
```

随机选择数据库记录的方法（使用 Randomize 函数，通过 SQL 语句实现）

对存储在数据库中的数据来说，随机数特性能给出上面的效果，但它们可能太慢了些。你不能要求 ASP“找个随机数”然后打印出来。实际上常见的解决方案是建立如下所示的循环：

```
Randomize
```

```
RNumber = Int(Rnd*499) + 1
```

```
While Not objRec.EOF
```

```
If objRec("ID") = RNumber THEN
```

```
... 这里是执行脚本 ...
```

```
end if
```

```
objRec.MoveNext
```

```
Wend
```

这很容易理解。首先，你取出 1 到 500 范围内的一个随机数（假设 500 就是数据库内记录的总数）。然后，你遍历每一记录来测试 ID 的值、检查其是否匹配 RNumber。满足条件的话就执行由 THEN 关键字开始的那一块代码。假如你的 RNumber 等于 495，那么要循环一遍数据库花的时间可就长了。虽然 500 这个数字看起来大了些，但相比更为稳固的企业解决方案这还是个小型数据库了，后者通常在一个数据库内就包含了成千上万条记录。这时候不就死定了？

采用 SQL，你就可以很快地找出准确的记录并且打开一个只包含该记录的 recordset，如下所示：

```
Randomize
```

```
RNumber = Int(Rnd*499) + 1
```

```
sql = "Select * FROM Customers Where ID = " & RNumber
```

```
set objRec = ObjConn.Execute(SQL)
```

```
Response.WriteRNumber & " = " & objRec("ID") & " " & objRec("c_email")
```

不必写出 RNumber 和 ID，你只需要检查匹配情况即可。只要你对以上代码的工作满意，你自可按需操作“随机”记录。Recordset 没有包含其他内容，因此你很快就能找到你需要的记录这样就大大降低了处理时间。

再谈随机数

现在你下定决心要榨干 Random 函数的最后一滴油，那么你可能会一次取出多条随机记录或者想采用一定随机范围内的记录。把上面的标准 Random 示例扩展一下就可以用 SQL 应对上面两种情况了。

为了取出几条随机选择的记录并存放在同一 recordset 内，你可以存储三个随机数，然后查询数据库获得匹配这些数字的记录：

```
sql = "Select * FROM Customers Where ID = " & RNumber & " or ID = " & RNumber2 & " or ID = " & RNumber3
```

假如你想选出 10 条记录（也许是每次页面装载时的 10 条链接的列表），你可以用 BETWEEN 或者数学等式选出第一条记录和适当数量的递增记录。这一操作可以通过好几种方式来完成，但是 Select 语句只显示一种可能（这里的 ID 是自动生成的号码）：

```
sql = "Select * FROM Customers Where ID BETWEEN " & RNumber & " AND " & RNumber & "+ 9"
```

注意：以上代码的执行目的不是检查数据库内是否有 9 条并发记录。

随机读取若干条记录, 测试过

Access 语法: Select top 10 * From 表名 order BY Rnd(id)

sql server:select top n * from 表名 order by newid()

mysql:select * From 表名 order By rand() Limit n

Access 左连接语法(最近开发要用左连接,Access 帮助什么都没有,网上没有 Access 的 SQL 说明,只有自己测试,现在记下以备后查)

语法 select table1.fid1,table1.fid2,table2.fid2 From table1 left join table2 on table1.fid1,table2.fid1 where ...

使用 SQL 语句 用...代替过长的字符串显示

语法:

SQL 数据库: select case when len(field)>10 then left(field,10)+'...' else field end as news_name,news_id from tablename

Access 数据库: Select iif(len(field)>2,left(field,2)+'...',field) FROM tablename;

Conn.Execute 说明

Execute 方法

该方法用于执行 SQL 语句。根据 SQL 语句执行后是否返回记录集,该方法的使用格式分为以下两种:

1. 执行 SQL 查询语句时,将返回查询得到的记录集。用法为:

Set 对象变量名=连接对象.Execute("SQL 查询语言")

Execute 方法调用后,会自动创建记录集对象,并将查询结果存储在该记录对象中,通过 Set 方法,将记录集赋给指定的对象保存,以后对象变量就代表了该记录集对象。

2. 执行 SQL 的操作语言时,没有记录集的返回。此时用法为:

连接对象.Execute "SQL 操作语言" [, RecordAffected][, Option]

·RecordAffected 为可选项,此出可放置一个变量,SQL 语句执行后,所生效的记录数会自动保存到该变量中。通过访问该变量,就可知道 SQL 语句队多少条记录进行了操作。

·Option 可选项,该参数的取值通常为 adCMDText,它用于告诉 ADO,应该将 Execute 方法之后的第一个字符解释为命令文本。通过指定该参数,可使执行更高效。

·BeginTrans、RollbackTrans、CommitTrans 方法

这三个方法是连接对象提供的用于事务处理的方法。BeginTrans 用于开始一个事物;RollbackTrans 用于回滚事务;CommitTrans 用于提交所有的事务处理结果,即确认事务的处理。

事务处理可以将一组操作视为一个整体,只有全部语句都成功执行后,事务处理才算成功;若其中有一个语句执行失败,则整个处理就算失败,并恢复到处里前的状态。

BeginTrans 和 CommitTrans 用于标记事务的开始和结束,在这两个之间的语句,就是作为事务处理的语句。判断事务处理是否成功,可通过连接对象的 Error 集合来实现,若 Error 集合的成员个数不为 0,则说明有错误发生,事务处理失败。Error 集合中的每一个 Error 对象,代表一个错误信息。

37.一道 SQL 语句面试题,关于 group by

表内容:

2005-05-09 胜

2005-05-09 胜

2005-05-09 负

2005-05-09 负

2005-05-10 胜

2005-05-10 负

2005-05-10 负

如果要生成下列结果, 该如何写 sql 语句?

胜 负

2005-05-09 2 2

2005-05-10 1 2

答: 代码如下:

```
create table #tmp(rq varchar(10),shengfu nchar(1))
insert into #tmp values('2005-05-09','胜')
insert into #tmp values('2005-05-09','胜')
insert into #tmp values('2005-05-09','负')
insert into #tmp values('2005-05-09','负')
insert into #tmp values('2005-05-10','胜')
insert into #tmp values('2005-05-10','负')
insert into #tmp values('2005-05-10','负')
1)select rq, sum(case when shengfu='胜' then 1 else 0 end)'胜',sum(case when shengfu='负' then 1 else 0 end)'负' from #tmp group by rq
2) select N.rq,N.胜,M.负 from (
select rq,胜=count(*) from #tmp where shengfu='胜'group by rq)N inner join
(select rq,负=count(*) from #tmp where shengfu='负'group by rq)M on N.rq=M.rq
3)select a.col001,a.a1 胜,b.b1 负 from
(select col001,count(col001) a1 from tmp1 where col002='胜' group by col001)
a,
(select col001,count(col001) b1 from tmp1 where col002='负' group by col001)
b
where a.col001=b.col001
```

38. 请教一个面试中遇到的 SQL 语句的查询问题

表中有 A B C 三列, 用 SQL 语句实现: 当 A 列大于 B 列时选择 A 列否则选择 B 列, 当 B 列大于 C 列时选择 B 列否则选择 C 列。

示例如下:

```
select (case when a>b then a else b end ),
(case when b>c then b esle c end)
from table_name
```

39. 面试题: 一个日期判断的 sql 语句?

请取出 tb_send 表中日期(SendTime 字段)为当天的所有记录?(SendTime 字段为 datetime 型, 包含日期与时间)

示例如下:

```
select * from tb where datediff(dd,SendTime,getdate())=0
```

40. 有一张表, 里面有 3 个字段: 语文, 数学, 英语。其中有 3 条记录分别表示语文 70 分, 数学 80 分, 英语 58 分, 请用一条 sql 语句查询出这三条记录并按以下条件显示出来(并写出您的思路):

大于或等于 80 表示优秀, 大于或等于 60 表示及格, 小于 60 分表示不及格。

显示格式:

语文	数学	英语
及格	优秀	不及格

示例如下:

```
select
(case when 语文>=80 then '优秀'
```

```
        when 语文>=60 then '及格'
else '不及格') as 语文,
(case when 数学>=80 then '优秀'
      when 数学>=60 then '及格'
else '不及格') as 数学,
(case when 英语>=80 then '优秀'
      when 英语>=60 then '及格'
else '不及格') as 英语,
from table
```

41.在 sqlserver2000 中请用 sql 创建一张用户临时表和系统临时表, 里面包含两个字段 ID 和 IDValues,类型都是 int 型, 并解释下两者的区别?

用户临时表:create table #xx(ID int, IDValues int)

系统临时表:create table ##xx(ID int, IDValues int)

区别:

用户临时表只对创建这个表的用户的 Session 可见,对其他进程是不可见的。
当创建它的进程消失时这个临时表就自动删除。

全局临时表对整个 SQL Server 实例都可见,但是所有访问它的 Session 都消失的时候,它也自动删除。

42.sqlserver2000 是一种大型数据库, 他的存储容量只受存储介质的限制, 请问它是通过什么方式实现这种无限容量机制的。

它的所有数据都存储在数据文件中 (*.dbf), 所以只要文件够大, SQL Server 的存储容量是可以扩大的。

SQL Server 2000 数据库有三种类型的文件:

主要数据文件

主要数据文件是数据库的起点, 指向数据库中文件的其它部分。每个数据库都有一个主要数据文件。主要数据文件的推荐文件扩展名是 .mdf。

次要数据文件

次要数据文件包含除主要数据文件外的所有数据文件。有些数据库可能没有次要数据文件, 而有些数据库则有多个次要数据文件。次要数据文件的推荐文件扩展名是 .ndf。

日志文件

日志文件包含恢复数据库所需的所有日志信息。每个数据库必须至少有一个日志文件, 但可以不止一个。日志文件的推荐文件扩展名是 .ldf。

43.请用一个 sql 语句得出结果

从 table1, table2 中取出如 table3 所列格式数据, 注意提供的数据及结果不准确, 只是作为一个格式向大家请教。

如使用存储过程也可以。

table1

月份 mon 部门 dep 业绩 yj

一月份	01	10
一月份	02	10
一月份	03	5
二月份	02	8
二月份	04	9
三月份	03	8

table2

部门 dep 部门名称 dname

01	国内业务一部
02	国内业务二部
03	国内业务三部
04	国际业务部

table3 (result)

部门 dep	一月份	二月份	三月份
01	10	null	null
02	10	8	null
03	null	5	8
04	null	null	9

1)

```
select a.部门名称 dname,b.业绩 yj as '一月份',c.业绩 yj as '二月份',d.业绩 yj as '三月份'
```

```
from table1 a,table2 b,table2 c,table2 d
```

```
where a.部门 dep = b.部门 dep and b.月份 mon = '一月份' and
```

```
a.部门 dep = c.部门 dep and c.月份 mon = '二月份' and
```

```
a.部门 dep = d.部门 dep and d.月份 mon = '三月份' and
```

2)

```
select a.dep,
```

```
sum(case when b.mon=1 then b.yj else 0 end) as '一月份',
```

```
sum(case when b.mon=2 then b.yj else 0 end) as '二月份',
```

```
sum(case when b.mon=3 then b.yj else 0 end) as '三月份',
```

```
sum(case when b.mon=4 then b.yj else 0 end) as '四月份',
```

```
sum(case when b.mon=5 then b.yj else 0 end) as '五月份',
```

```
sum(case when b.mon=6 then b.yj else 0 end) as '六月份',
```

```
sum(case when b.mon=7 then b.yj else 0 end) as '七月份',
```

```
sum(case when b.mon=8 then b.yj else 0 end) as '八月份',
```

```
sum(case when b.mon=9 then b.yj else 0 end) as '九月份',
```

```
sum(case when b.mon=10 then b.yj else 0 end) as '十月份',
```

```
sum(case when b.mon=11 then b.yj else 0 end) as '十一月份',
```

```
sum(case when b.mon=12 then b.yj else 0 end) as '十二月份',  
from table2 a left join table1 b on a.dep=b.dep
```

44. 华为一道面试题

一个表中的 id 有多个记录, 把所有这个 id 的记录查出来, 并显示共有多少条记录数。

```
-----  
select id, Count (*) from tb group by id having count(*)>1  
select * from(select count(ID) as count from table group by ID)T where T.count>1
```

JavaScript 方面

1、如何校验数字型?

```
var re=/^\d{1,8}$|^\.?\d{1,2}$/;  
var str=document.form1.all(i).value;  
var r=str.match(re);  
if (r==null)  
{  
    sign=-4;  
    break;  
}  
else{  
    document.form1.all(i).value=parseFloat(str);  
}
```

CORBA 方面

1、CORBA 是什么?用途是什么?

答: CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture), 由对象管理组织 (Object Management Group, 缩写为 OMG)标准化。它的组成是接口定义语言(IDL), 语言绑定(binding;也译为联编)和允许应用程序间互操作的协议。 其目的为:
用不同的程序设计语言书写
在不同的进程中运行
为不同的操作系统开发

LINUX 方面

1、LINUX 下线程, GDI 类的解释。

答: LINUX 实现的就是基于核心轻量级进程的"一对一"线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库。