

sql语句学习:Sql 语句学习指南

疯狂代码 <http://www.crazycoder.cn/> j:<http://www.crazycoder.cn/DataBase/Article69022.html>

1.在查询结果中显示列名:

a.用as关键字:select name as '姓名' from students order by age

b.直接表示:select name '姓名' from students order by age

2.精确查找:

a.用in限定范围:select * from students where native in ('湖南','四川')

b.between...and:select * from students where age between 20 and 30

c. "=" :select * from students where name = '李山'

d.like:select * from students where name like '李%' (注意查询条件中有 "%" 则介绍说明是部分匹配而且还有先后信息在里面即查找以 "李" 开头匹配项所以若查询有 "李" 所有对象应该命令:'%李%';若是第 2 个字为李则应为'_李%'或'_李'或'_李_')

e.匹配检查符:select * from courses where cno like '[AC]%' (表示或关系和"in(...)"类似而且"%"可以表示范围如:select * from courses where cno like '[A-C]%')

3.对于时间类型变量处理

a.smalldatetime:直接按照串处理方式进行处理例如:

select * from students where birth >= '1980-1-1' and birth <= '1980-12-31'

4.集

a.count求和如:select count(*) from students (求学生总人数)

b.avg(列)求平均如:select avg(mark) from grades where cno='B2'

c.max(列)和min(列)求最大和最小

5.分组group

常用于统计时如分组查总数:

select gender,count(sno)

from students

group by gender

(查看男女学生各有多少)

注意:从哪种角度分组就从哪列"group by"

对于多重分组只需将分组规则罗列比如查询各届各专业男女同学人数那么分组规则有:届别(grade)、专业(mno)和性别(gender)所以有"group by grade, mno, gender"

select grade, mno, gender, count(*)

from students

group by grade, mno, gender

通常group还和having联用比如查询1门课以上不及格学生则按学号(sno)分类有:

select sno,count(*) from grades

where mark<60

group by sno

having count(*)>1

6.UNION联合

合并查询结果如:

```
SELECT * FROM students
```

```
WHERE name like '张%'
```

```
UNION [ALL]
```

```
SELECT * FROM students
```

```
WHERE name like '李%'
```

7.多表查询

a.内连接

```
select g.sno,s.name,c.coursename
```

```
from grades g JOIN students s _disibledevent=> JOIN courses c _disibledevent=> (注意可
```

以引用别名)

b.外连接

b1.左连接

```
select courses.cno,max(coursename),count(sno)
```

```
from courses LEFT JOIN grades _disibledevent=> group by courses.cno
```

左连接特点:显示全部左边表中所有项目即使其中有些项中数据未填写完全

左外连接返回那些存在于左表而右表中却没有行再加上内连接行

b2.右连接

和左连接类似

b3.全连接

```
select sno,name,major
```

```
from students FULL JOIN majors _disibledevent=> 两边表中内容全部显示
```

c.自身连接

```
select c1.cno,c1.coursename,c1.pno,c2.coursename
```

```
from courses c1,courses c2 where c1.pno=c2.cno
```

采用别名解决问题

d.交叉连接

```
select lastname+firstname from lastname CROSS JOIN firstanme
```

相当于做笛卡儿积

8.嵌套查询

a.用关键字IN,如查询李山同乡:

```
select * from students
```

```
where native in (select native from students where name=' 李山')
```

b.使用关键字EXIST,比如下面两句是等价:

```
select * from students
where sno in (select sno from grades where cno='B2')
select * from students where exists
(select * from grades where
grades.sno=students.sno AND cno='B2')
```

9.有关排序order

- a.对于排序order有两种思路方法:asc升序和desc降序
- b.对于排序order,可以按照查询条件中某项排列而且这项可用数字表示如:

```
select sno,count(*) ,avg(mark) from grades
group by sno
having avg(mark)>85
order by 3
```

10.其他

- a.对于有空格识别名称应该用""括住
- b.对于某列中没有数据特定查询可以用null判断如select sno,courseno from grades where mark IS

NULL

- c.注意区分在嵌套查询中使用any和all区别any相当于逻辑运算 “||” 而all则相当于逻辑运算 “&&”
- d.注意在做否定意义查询是小心进入陷阱:

如没有选修 ‘B2’课程学生 :

```
select students.*
from students, grades
where students.sno=grades.sno
AND grades.cno <> 'B2'
```

上面查询方式是正确方式见下方:

```
select * from students
where not exists (select * from grades
where grades.sno=students.sno AND cno='B2')
```

11.有关有难度多重嵌套查询解决思想:

如选修了全部课程学生:

```
select *
from students
where not exists ( select *
from courses
where NOT EXISTS
(select *
from grades
```

```
where sno=students.sno
```

```
AND cno=courses.cno))
```

2009-9-6 22:59:07

疯狂代码 <http://www.crazycoder.cn/>