

CSDN

博客 (http://blog.csdn.net/ref=toolbar)
学院 (http://edu.csdn.net/ref=toolbar)
下载 (http://download.csdn.net/ref=toolbar)
GitChat (http://gitbook.cn/?ref=csdn) ...

5

写博客

发Chat

登录 (https://passport.csdn.net/account/login?ref=toolbar) 注册 (https://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)
/postedit?ref=toolbar
/activity?utm_source=csdnblog1

SQL练习题完整（做完你就是高手）

转载 2013年12月18日 15:51:15 23745

会飞的鱼online (http://b...

+关注

(http://blog.csdn.net/suheonline)

码云

0

原创

52

粉丝

10

喜欢

1

(https://g/suheonli

SQL 练习题答案

一、补充作业一、

设有三个关系：

S (SNO, SNAME, AGE, SEX, Sdept)

SC (SNO, CNO, GRADE)

C (CNO, CNAME, TEACHER)

试用关系代数表达式表示下列查询：

1、查询学号为S3学生所学课程的课程名与任课教师名。

2、查询至少选修LIU老师所教课程中一门课的女生姓名。

3、查询WANG同学不学的课程的课程号。

4、查询至少选修两门课程的学生学号。

5、查询选修课程中包含LIU老师所教全部课程的学生学号。

补充作业二、

三个关系同上，试用SQL语言表示下列查询：

1、 查询一门课程都及格的学生的学号

方法1：

提示：根据学号分组，就得到每个学生所有的课程成绩，在某个学生这一组成绩里，如果他所有的课程成绩都大于60分则输出该组学生的学号

Select sno from sc group by sno having (min(grade)>=60)

2、查询既有课程大于90分又有课程不及格的学生的学号

自身连接：

Select sno from sc where grade >90 and sno in (select sno from sc where grade<60)

3、查询平均分不及格的课程号和平均成绩

Select cno , avg(GRADE) from sc group by cno having avg(grade)<60

加入查询平均分更精确的课程号和课程成绩和课程60万程序员共同成长！

他的最新文章

更多文章 (http://blog.csdn.net/suheonline)

ASPGridViewExporter excel导出 (http://blog.csdn.net/suheonline/article/details/44781603)

asp.net 生成 excel 解决迅雷下载 (http://blog.csdn.net/suheonline/article/details/43191249)

UpdatePanel GridView 无刷新操作 (http://blog.csdn.net/suheonline/article/details/42916667)

SQL 检测是否存在相同的行。 (http://blog.csdn.net/suheonline/article/details/42707501)

win2003 安装.net4.0 同时部署asp.net4.0 (http://blog.csdn.net/suheonline/article/details/42707175)

他的热门文章

SQL练习题完整（做完你就是高手） (http://blog.csdn.net/suheonline/article/details/17396197)

23458 登录 注册

1 of 33

1/3/18, 23:17

X

8、查询有30名以上学生选修的课程号。

Select cno from sc group by cno having count(*)>30

9、查询至今没有考试不及格的学生学号。

Select distinct sno from sc where sno not in

(Select sno from sc where grade<60)

或： Select sno from sc group by sno having min(grade)>=60

10、查询所有考试成绩的平均分相同的学生学号分组

二、

1、找出选修课程号为C2的学生学号与成绩。

Select sno,grade from sc where cno='C2'

2、找出选修课程号为C4的学生学号与姓名。

Select s.sno,sname from s,sc where s.sno=sc.sno and cno='C4'

注意本题也可以用嵌套做

思考本题改为“找出选修课程号为C4的学生学号、姓名与成绩”后还能用嵌套做吗？

3、找出选修课程名为 Maths 的学生学号与姓名。

Select s.sno,sname from s,sc,c

where s.sno=sc.sno and c.cno=sc.cno and cname='Maths'

注意本题也可以用嵌套做

4、找出选修课程号为C2或C4的学生学号。

Select distinct sno from sc where cno in ('C2','C4')

或： Select distinct sno from sc where cno='C2' or cno='C4'

5、找出选修课程号为C2和C4的学生学号。

Select sno from sc where cno='C2' and sno in

(Select sno from sc where cno='C4')

注意本题也可以用连接做

思考： Select distinct sno from sc where cno='C2' and cno='C4' 正确吗？

6、找出不学C2课程的学生姓名和年龄。

Select sname,sage from s where sno not in

(Select sno from sc where cno='C2')

或： Select sname,sage from s where not exists

(Select * from sc where sno=s.sno and cno='C2')

7、找出选修了数据库课程的所有学生姓名。(同3)

Select sname from s,sc,c

where s.sno=sc.sno and c.cno=sc.cno and cname='数据库'

8、找出数据库课程不及格的女生姓名。

连接： Select sname from s,sc,c

where s.sno=sc.sno and c.cno=sc.cno and cname='数据库'

and grade<60 and ssex='女'

嵌套： Select sname from s where ssex='女' and sno in

(Select sno from sc where grade<60 and cno in

(Select cno from c where cname='数据库')

)

9、找出各门课程的平均成绩，输出课程名和平均成绩。

Select cname,avg(grade) from sc,c

where c.cno=sc.cno group by c.cno,cname

思考本题也可以用嵌套做吗？

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

10、找出各个学生的平均成绩，输出学生姓名和平均成绩。

```
Select sname, avg(grade) from s, sc
where s.sno=sc.sno group by s.sno, sname
```

思考本题也可以用嵌套做吗?

11、找出至少有30个学生选修的课程名。

```
Select cname from c where cno in
( Select cno from sc group by cno having count(*) >= 30 )
```

注意本题也可以用连接做

12、找出选修了不少于3门课程的学生姓名。

```
Select sname from s where sno in
( Select sno from sc group by sno having count(*) >= 3 )
```

注意本题也可以用连接做

13、找出各门课程的成绩均不低于90分的学生姓名。

```
Select sname from s, sc where s.sno=sc.sno
group by s.sno, sname having min(grade) >= 90
```

方法二:

```
Select sname from s where sno not in
( Select sno from sc where grade < 90 )
```

只要有一门不小于90分就会输出该学生学号

14、找出数据库课程成绩不低于该门课程平均分的学生姓名。

```
Select sname from s, sc, c
where s.sno=sc.sno and sc.cno=c.cno and cname='数据库' and grade >
( Select avg(grade) from sc, c
where sc.cno=c.cno and cname='数据库'
)
```

15、找出各个系科男女学生的平均年龄和人数。

```
Select sdept, ssex, avg(sage), count(*) from s group by sdept, ssex
```

16、找出计算机系(JSJ)课程平均分最高的学生学号和姓名。

```
Select s.sno, sname from s, sc where s.sno=sc.sno and sdept='JSJ'
group by s.sno, sname
having avg(grade) >= ALL
( Select avg(grade) from s, sc
where s.sno=sc.sno and sdept='JSJ'
group by s.sno
)
```

17、(补充)查询每门课程的及格率。

本题可以分三步做:

第1步: 得到每门课的选修人数

```
create view v_all(cno, cnt)
as select cno, count(*) from sc group by cno
```

第2步: 得到每门课及格人数

```
create view v_pass(cno, cnt_pass)
as select cno, count(*) from sc where grade >= 60 group by cno
```

第3步: 每门课的及格人数/每门课选修人数

```
select v_all.cno, cnt_pass*100/cnt from v_all, v_pass
where v_all.cno = v_pass.cno
```

18、查询平均分不及格的学生学号，姓名，平均分。

```
Select s.sno, sname, avg(grade) from student, sc
where student.sno=sc.sno
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长。

登录

注册

×

```
group by sc.sno,sname
having avg(grade)<60
```

思考本题也可以用嵌套做吗?

19、查询平均分不及格的学生人数。

```
Select count(*) from student
where sno in
( select sno from sc group by sno having avg(grade)<60 )
```

下面是一个典型的错误

```
Select count(*) from sc group by sno having avg(grade)<60
```

这是每个学生有几门不及格的数目

三、

1、查询工资在1000到3000元之间的男性业务员的姓名和办公室编号。

```
Select Yname, Ono from YWY
where Salary between 1000 and 3000 and Ysex='男'
```

2、查询各个办公室的业务员人数，输出办公室编号和对应的人数。

```
Select Ono, count(*) from YWY group by Ono
```

3、查询每个客户在2002年5月购买的总金额，输出客户号和相应的总金额。

```
Select Kno, sum(Fmoney) from FP
where Fdate between '2002.5.1' and '2002.5.31'
group by Kno
```

4、查询2002年5月购买次数超过5次的所有客户号，且按客户号升序排序。

```
Select Kno from FP
where Fdate between '2002.5.1' and '2002.5.31'
group by Kno
having count(*) > 5
order by Kno ASC
```

5、查询各办公室男性和女性业务员的平均工资。

```
Select Ono, Ysex, avg(Salary) from YWY group by Ono, Ysex
```

6、查询2002年5月曾经在王海亮业务员手中购买过商品的客户号、客户姓名和联系电话。

```
Select Kno, Kname, Phone from KH where Kno in
( Select Kno from FP
where Fdate between '2002.5.1' and '2002.5.31' and Yno in
( Select Yno from YWY where Yname='王海亮' )
)
```

注意本题也可以用连接做

7、查询所有工资比1538号业务员高的业务员的编号、姓名和工资。

```
Select Yno, Yname, Salary from YWY where Salary >
( Select Salary from YWY where Yno='1538' )
```

8、查询所有与1538号业务员在同一个办公室的其他业务员的编号和姓名。

```
Select Yno, Yname from YWY where Yno != '1538' and Ono in
( Select Ono from YWY where Yno='1538' )
```

9、查询销售总金额最高的业务员的编号。

```
Select Yno from FP group by Yno having sum(Fmoney) >= ALL
( Select sum(Fmoney) from FP group by Yno )
```

10、查询所有业务员的编号、姓名、工资以及工资比他高的其他业务员的平均工资。

利用自连接

```
Select Y1.Yno, Y1.Yname, Y1.Salary, avg(Y2.Salary)
from YWY Y1, YWY Y2
where Y1.Salary < Y2.Salary
group by Y1.Yno
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

四、

登录

注册

×

- 1、找出每个班级的班级代码、学生人数、平均成绩。

```
SelectBJDM,count(*),avg(CJ) from SC group by BJDM
```

- 2、找出每个学生的班级代码、学生姓名、考试科目数、总成绩。

```
SelectBJDM,XSXM,count(*),sum(CJ) from SC
group byBJDM,BNXH,XSXM
```

- 3、输出一张表格，每位学生对应一条记录，包括字段：

班级代码、学生姓名、语文成绩、数学成绩、外语成绩。

```
SelectSC1.BJDM,SC1.XSXM,SC1.CJ,SC2.CJ,SC3.CJ
from SC SC1, SC SC2, SC SC3
whereSC1.BJDM=SC2.BJDM and SC1.BNXH=SC2.BNXH and
SC2.BJDM=SC3.BJDM and SC2.BNXH=SC3.BNXH and
SC1.KM='语文' and SC2.KM='数学' and SC3.KM='外语'
```

- 4、输出一张表格，有成绩低于60分的每位学生对应一条记录，包括字段：

班级代码、学生姓名、最低成绩。

```
SelectBJDM,XSXM,min(CJ) from SC
where CJ<60 group by BJDM,BNXH,XSXM
或： SelectBJDM,XSXM,min(CJ) from SC
group byBJDM,BNXH,XSXM
havingmin(CJ)<60
```

- 5、输出一张表格，有成绩低于60分的每位学生对应一条记录，包括字段：

班级代码、学生姓名、最高成绩、平均成绩。

```
SelectBJDM,XSXM,max(CJ) from SC
group byBJDM,BNXH,XSXM
havingmin(CJ)<60
```

请思考下列做法是否正确：

```
SelectBJDM,XSXM,max(CJ),avg(CJ) from SC
where CJ<60 group byBJDM,BNXH,XSXM
```

- 6、输出一张表格，所有成绩都不低于60分的每位学生对应一条记录，包括字段：

班级代码、学生姓名、平均成绩。

```
SelectBJDM,XSXM,avg(CJ) from SC
group by BJDM,BNXH,XSXM
havingmin(CJ)>=60
```

- 7、输出一张表格，每一位学生对应一条记录，包括字段：

班级代码、学生姓名、去掉一个最低分后的平均成绩。

```
SelectBJDM,XSXM,(sum(CJ)-min(CJ))/(count(*)-1) from SC
group byBJDM,BNXH,XSXM
```

- 8、输出一张表格，每门科目对应一条记录，包括字段：

科目、去掉一个最低分后的平均成绩。

```
Select KM,(sum(CJ)-min(CJ))/(count(*)-1)from SC
group by KM
```

实验指导中“八 SQL查询语句”的答案

- 1、查询年龄在19至21岁之间的女生的学号，姓名，年龄，按年龄从大到小排列。

```
Selectsno,sname,sage from student
where sagebetween 19 and 21 and ssex='女'
order by sagedesc
```

- 2、查询姓名中有“明”字的学生人数。

```
Select count(*)from student
where sname like "%明%"
```

- 3、查询1001课程没有成绩的学生的学号。

```
Selectsno from sc where cno='1001' and grade is null
```

- 4、查询JSJ、SX、WL系的学生学号，姓名，结果按系及学号排列。

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



```
Select sno,sname,sdept from student
where sdept in( 'JSJ', 'SX', 'WL' )
order by sdept,sno
```

5、计算每一门课的总分、平均分，最高分、最低分。

```
Select cno,sum(grade),avg(grade),max(grade),min(grade)
from sc
group by cno
```

6、查询平均分大于90分的男学生学号及平均分。

连接：

```
select sc.sno,avg(grade) from student,sc
where student.sno=sc.sno and ssex='男'
group by sc.sno
having avg(grade)>90
```

嵌套：

```
select sno,avg(grade) from sc
where sno in (select sno from student where ssex='男')
group by sno
having avg(grade)>90
```

7、查询选修课程超过2门的学生姓名。

```
select sname from student,sc
where student.sno=sc.sno
group by sc.sno,sname
having count(*)>2
```

本题也可以用嵌套做

8、查询 JSJ 系的学生选修的课程号。

```
Select distinct cno from student,sc
where student.sno=sc.sno and sdept='JSJ'
```

本题也可以用嵌套做

9、查询选修1002课程的学生姓名(用连接和嵌套2种方法)

连接：Select sname from student,sc

```
where student.sno=sc.sno and cno='1002'
```

嵌套：Select sname from student where sno in
(select sno from sc where cno='1002')

10、查询学生姓名以及他选修课程的课程号及成绩。

```
Select sname,cno,grade from student,sc
where student.sno=sc.sno
```

思考本题也可以用嵌套做吗？

11、查询选修“数据库原理”课且成绩 80 以上的学生姓名(用连接和嵌套2种方法)

连接：Select sname from student,sc,course

```
where student.sno=sc.sno and sc.cno=course.cno and
cname='数据库原理' and grade>80
```

嵌套：Select sname from student where sno in
(select sno from sc where grade>80 and cno in
(select cno from course where cname='数据库原理'))

14、查询没有选修1002课程的学生姓名。

```
Select sname from student
where sno not in (select sno from sc where cno='1002')
```

或：select sname from student
where not exists

```
(select * from sc where cno='1002' and sno=student.sno)
```

思考本题也可以用一般的连接做吗？

15、查询平均分最高的学生学号及平均分

```
Select sno,avg(grade)
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

```

from sc
group by sno
having avg(grade) >= ALL ( Select avg(grade)
                        from sc
                        group by sno
                      )

```

16、查询每门课程成绩都高于该门课程平均分的学生学号。
 可以先计算每门课程平均分
 create view c_avg(cno, avg_grade)
 as select cno, avg(grade) from sc group by cno
 再查询
 Select distinct sno from sc
 where sno not in (Select sno from sc, c_avg
 where sc.cno = c_avg.cno and grade < avg_grade
)

```

=====
SELECT DISTINCT Sno
FROM SC SC1
WHERE SC1.Sno NOT IN
( SELECT SC2.Sno
  FROM SC SC2
  WHERE SC2.Grade <=
    ( SELECT AVG(SC3.Grade)
      FROM SC SC3
      WHERE SC3.Cno = SC2.Cno
    )
)

```

或:
 SELECT DISTINCT Sno
 FROM SC SC1
 WHERE NOT EXISTS
 (SELECT *
 FROM SC SC2
 WHERE SC2.Sno = SC1.Sno AND SC2.Grade <=
 (SELECT AVG(SC3.Grade)
 FROM SC SC3
 WHERE SC3.Cno = SC2.Cno
)
)

(3) 检索至少选修LIU老师所授课程中一门课程的女学生姓名。

```

SELECT SNAME
FROM S
WHERE SEX = 'F' AND S# IN
( SELECT S#
  FROM SC
  WHERE C# IN
    ( SELECT C#
      FROM C
      WHERE TEACHER = 'LIU' )
)

```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！
 NOTICE: 有多种写法，比如联接查询写法:
 SELECT SNAME

[登录](#)
[注册](#)



```
FROM S,SC,C
WHERE SEX= 'F' AND SC.S#=S.S#
AND SC.C#=C.C#
AND TEACHER='LIU'
```

但上一种写法更好一些。



5



(4) 检索WANG同学不学的课程的课程号。

```
SELECT C#
FROM C
WHERE C# NOT IN
(SELECT C#
FROM SC
WHERE S# IN
(SELECT S#
FROM S
WHERE SNAME='WANG'))
```

(5) 检索至少选修两门课程的学生学号。

```
SELECT DISTINCT X.SNO
FROM SC X,SC Y
WHERE X.SNO=Y.SNO AND X.CNO<>Y.CNO
```

Notice:对表SC进行自连接, X, Y是SC的两个别名。

(6) 检索全部学生都选修的课程的课程号与课程名。

```
SELECT C#,CNAME
FROM C
WHERE NOT EXISTS
(SELECT *
FROM S
WHERE S# NOT IN
(SELECT *
FROM SC
WHERE SC.C#=C.C#))
```

要从语义上分解: (1) 选择课程的课程号与课程名, 不存在不选这门课的同学。

其中, “不选这门课的同学” 可以表示为:

SELECT *	或者 SELECT *
FROM S	FROM S
WHERE S# NOT IN	WHERE NOT EXISTS
(SELECT *	(SELECT *
FROM SC	FROM SC
WHERE SC.C#=C.C#)	WHERE S.S#=C.S# AND
	SC.C#=C.C#)

(7) 检索选修课程包含LIU老师所授课的学生学号。

```
SELECT DISTINCT S#
FROM SC
WHERE C# IN
(SELECT C#
FROM C
WHERE TEACHER='LIU'))
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！
 2.3. 设有两个基本表R(A, B, C)和S(D, E, F)，试用SQL查询语句表达下列关系代数表达式:

登录

注册



(1) $\pi_A(R)(2)\sigma_{B='17'}(R)(3)R \times S(4))\pi_{A,F}(\sigma_{C=D}(R \times S))$
 (1)SELECT A FROM R
 (2)SELECT * FROM R WHERE B='17'
 (3)SELECT A,B,C,D,E,F FROM R,S
 (4)SELECT A,F FROM R,S WHERE R.C=S.D



5



3.43.4 设有两个基本表R (A, B, C) 和S (A, B, C) 试用SQL查询语句表达下列关系代数表达式:

(1) $R \cup S$ (2) $R \cap S$ (3) $R - S$ (4) $\pi_{A,B}(R)\pi_{B,C}(S)$

(1)SELECT A,B,C
FROM R
UNION
SELECT A,B,C
FROM S

(2)SELECT A,B,C
FROM R
INTERSECT
SELECT A,B,C
FROM S

(3)SELECT A,B,C
FROM R
WHERE NOT EXISTS
(SELECT A,B,C
FROM S
WHERE R.A=S.A AND R.B=S.B AND R.C=S.C)

(4)SELECT R.A,R.B,S.C
FROM R,S
WHERE R.B=S.B

3.5 试叙述SQL语言的关系代数特点和元组演算特点。
(P61-62)

3.6 试用SQL查询语句表达下列对教学数据库中三个基本表S、SC、C的查询:

(1) 统计有学生选修的课程门数。
SELECT COUNT(DISTINCT C#) FROM SC

(2) 求选修C4课程的学生们的平均年龄。
SELECT AVG(AGE)
FROM S
WHERE S# IN
(SELECT S#
FROM SC
WHERE C#='C4')

或者,

SELECT AVG(AGE)
FROM S, SC
WHERE S.S#=SC.S# AND C#='004'

(3) 求出老师所授课程的每门课程的学生平均成绩。
SELECT CNAME,AVG(GRADE)

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

[登录](#)
[注册](#)


```
FROM SC,C
WHERE SC.C#=C.C# AND TEACHER='LIU'
GROUP BY C#
```

(4) 统计每门课程的学生选修人数（超过10人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列。

```
SELECT DISTINCT C#,COUNT(S#)
FROM SC
GROUP BY C#
HAVING COUNT(S#)>10
ORDER BY 2 DESC, C# ASC
```

(5) 检索学号比WANG同学大，而年龄比他小的学生姓名。

```
SELECT X.SNAME
FROM S AS X, S AS Y
WHERE Y.SNAME='WANG' AND X.S#>Y.S# AND X.AGE<Y.AGE
```

(6) 检索姓名以WANG打头的所有学生的姓名和年龄。

```
SELECT SNAME,AGE
FROM S
WHERE SNAME LIKE 'WANG%'
```

(7) 在SC中检索成绩为空值的学生学号和课程号。

```
SELECT S#,C#
FROM SC
WHERE GRADE IS NULL
```

(8) 求年龄大于女同学平均年龄的男学生姓名和年龄。

```
SELECT SNAME,AGE
FROM S AS X
WHERE X.SEX='男' AND X.AGE>(SELECT AVG(AGE)FROM S AS Y WHERE Y.SEX='女')
```

(9) 求年龄大于所有女同学年龄的男学生姓名和年龄。

```
SELECT SNAME,AGE
FROM S AS X
WHERE X.SEX='男' AND X.AGE>ALL (SELECT AGE FROM S AS Y WHERE Y.SEX='女')
```

除法运算

3.7 试用SQL更新语句表达对教学数据库中三个基本表S、SC、C的各个更新操作：

(1) 往基本表S中插入一个学生元组（‘S9’，‘WU’，18）。

```
INSERT INTO S(S#,SNAME,AGE) VALUES('59','WU',18)
```

(2) 在基本表S中检索每一门课程成绩都大于等于80分的学生学号、姓名和性别，并把检索到的值送往另一个已存在的基本表STUDENT（S#，SNAME，SEX）。

```
INSERT INTO STUDENT(S#,SNAME,SEX)
SELECT S#,SNAME,SEX
FROM S WHERE NOT EXISTS
(SELECT * FROM SC WHERE
GRADE<80 AND S.S#=SC.S#)
```

(3) 在基本表SC中删除尚无成绩的选课元组。

```
DELETE FROM SC
WHERE GRADE IS NULL
```

(4) 把WANG同学的学习选课和成绩全部删去。

```
DELETE FROM SC
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

```
WHERE S# IN
  (SELECT S#
   FROM S
   WHERE SNAME='WANG')
```

(5) 把选修MATHS课不及格的成绩全改为空值。

```
UPDATE SC
SET GRADE=NULL
WHERE GRADE<60 AND C# IN
  (SELECT C#
   FROM C
   WHERE CNAME='MATHS')
```

(6) 把低于总平均成绩的女同学成绩提高5%。

```
UPDATE SC
SET GRADE=GRADE*1.05
WHERE GRADE<(SELECT AVG(GRADE) FROM SC) AND S# IN (SELECT S# FROM
SWHERE SEX='F')
```

(7) 在基本表SC中修改C4课程的成绩，若成绩小于等于75分时提高5%，若成绩大于75分时提高4%（用两个UPDATE语句实现）。

```
UPDATE SC
SET GRADE=GRADE*1.05
WHERE C#='C4' AND GRADE<=75
UPDATE SC
SET GRADE=GRADE*1.04
WHERE C#='C4' AND GRADE>75
```

3.8 在第1章例1.4中提到“仓库管理”关系模型有五个关系模式：

零件 PART (P#, PNAME, COLOR, WEIGHT)

项目 PROJECT (J#, JNAME, DATE)

供应商 SUPPLIER (S#, SNAME, SADDR)

供应 P_P (J#, P#, TOTOAL)

采购 P_S (P#, S#, QUANTITY)

(1) 试用SQLDDL语句定义上述五个基本表，并说明主键和外键。

```
CREATE TABLE PART
(P# CHAR(4) NOT NULL,PNAME CHAR(12) NOT NULL,
COLOR CHAR(10),WEIGHT REAL,
PRIMARY KEY(P#))
```

```
CREATE TABLE PROJECT
(J# CHAR(4) NOT NULL,JNAME CHAR(12) NOT NULL,
DATE DATE,
PRIMARY KEY(J#))
```

```
CREATE TABLE SUPLIER
(S# CHAR(4) NOT NULL,SNAME CHAR(12),SADDR VARCHAR(20),
PRIMARY KEY(S#))
```

```
CREATE TABLE P_P
(J# CHAR(4),P# CHAR(4),TOTAL INTEGER,
PRIMARY KEY(J#,P#),
FOREIGN KEY(J#) REFERENCE PROJECT(J#),
FOREIGN KEY(P#) REFERENCE PART(P#))
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



```
CREATE TABLE P_S
(P# CHAR(4),S# CHAR(4),QUANTITY INTEGER,
PRIMARY KEY(P#,S#),
FOREIGN KEY(P#) REFERENCE PART(P#),
FOREIGN KEY(S#) REFERENCE SUPPLIER(S#))
```



5



(2) 试将PROGECT、P__P、PART三个基本表的自然联接定义为一个视图VIEW1，PART、P__S、SUPPLIER 三个基本表的自然联接定义为一个视图VIEW2。

```
CREATE VIEW VIEW1(J#,JNAME,DATE,P#,PNAME,COLOR,WEIGHT,TOTAL)
AS SELECT
PROJECT.J#,JNAME,DATE,PART.P#,PNAME,COLOR,WEIGHT,TOTAL
FROM PROJECT,PART,P_P
WHERE PART.P#=P_P.P# AND P_P.J#=PROJECT.J#
```

```
CREATE VIEW VIEW2(P#,PNAME,COLOR,WEIGHT,S#,SNAME,SADDR,QUANTITY)
AS SELECT
PART.P#,PNAME,COLOR,WEIGHT,SUPPLIER.S#,SNAME,SADDR,QUANTITY
FROM PART,P_S,SUPPLIER
WHERE PART.P#=P_S.P# AND P_S.S#=SUPPLIER.S#
```

(3) 试在上述两个视图的基础上进行数据查询：

1) 检索上海的供应商所供应的零件的编号和名字。

```
SELECT P#,PNAME FROM VIEW2 WHERE SADDR='SHANGHAI'
```

2) 检索项目J4所用零件的供应商编号和名字。

```
SELECT S#,SNAME FROM VIEW2 WHERE P# IN(SELECT P# FROM VIEW1 WHERE
J#='J4')
```

3.9 对于教学数据库中基本表SC，已建立下列视图：

```
CREATE VIEWS__GRADE (S#, C__NUM, AVG__GRADE)
AS SELECT S#, COUNT(C#), AVG (GRADE)
FROM SC
GROUP BY S#
```

试判断下列查询和更新是否允许执行。若允许，写出转换到基本表SC上的相应操作。

(1) SELECT*
FROM S__GRADE
允许
SELECT S#,COUNT(C#),AVG(GRADE)FROM SC GROUP BY S#

(2) SELECT S#, C__NUM
FROM S__GRADE
WHERE AVG__GRADE>80
允许
SELECT S#,COUNT(C#) FROM SC WHERE AVG(GRADE)>80

(3) SELECT S#, AVG__GRADE
FROM S__GRADE
WHERE C__NUM>(SELECT C__NUM
FROM S__GRADE
WHERE S# = 'S4')

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！
SELECT S#,AVG(GRADE)

[登录](#)
[注册](#)


```
FROM SC AS X
WHERE COUNT(X.C#)>(SELECTCOUNT(Y.C#) FROM SC AS Y WHERE Y.S#='S4')
GROUP BY S#
```

(4) UPDATES__GRADE
SETC__NUM=C__NUM+1
WHERE S# = 'S4'
不允许

(5) DELETEFROMS__GRADE
WHERE C__NUM>4
不允许

3.10 预处理方式对于嵌入式SQL的实现有什么重要意义？

预处理方式是先用预处理程序对源程序进行扫描，识别出SQL语句，并处理成宿主语言的函数调用形式；然后再用宿主语言的编译程序把源程序编译成目标程序。这样，不用扩充宿主语言的编译程序，就能处理SQL语句。

3.11 在宿主语言的程序中使用SQL语句有哪些规定？

在宿主语言的程序中使用SQL语句有以下规定：

- (1) 在程序中要区分SQL语句与宿主语言语句
- (2) 允许嵌入的SQL语句引用宿主语言的程序变量（称为共享变量），但有两项规定：
 - 1) 引用时，这些变量前必须加“：”作为前缀标识，以示与数据库中变量有区别。
 - 2) 这些变量由宿主语言的程序定义，并用SQL的DECLARE语句说明。
- (3) SQL的集合处理方式与宿主语言单记录处理方式之间要协调。需要采用游标机制，把集合操作转换成单记录处理方式。

3.12 SQL的集合处理方式与宿主语言单记录处理方式之间如何协调？

由于SQL语句处理的是记录集合，而宿主语言语句一次只能处理一个记录，因此需要用游标(cursor)机制，把集合操作转换成单记录处理方式。

2.13 嵌入式SQL语句何时不必涉及到游标？何时必须涉及到游标？

- (1) INSERT、DELETE、UPDATE语句，查询结果肯定是单元组时的SELECT语句，都可直接嵌入在主程序中使用，不必涉及到游标。
- (2) 当SELECT语句查询结果是多个元组时，此时宿主语言程序无法使用，一定要用游标机制把多个元组一次一个地传送给宿主语言处理。

(电商)数据库原理及应用_模拟考核试题及参考答案

一、单选(每空1分,共10分)

1. 在Access数据库中，数据保存在（ ）对象中。
A. 窗体 B. 查询 C. 报表 D. 表
2. 如果某一字段数据类型为文本型，字段大小为8，该字段中最多可输入（ ）个汉字。
A. 8 B. 4 C. 16 D. 32
3. 文本型字段最多可以存放（ ）个字符。
A. 250 B. 252 C. 254 D. 255
4. Access用户操作界面由（ ）部分组成。
A. 4 B. 5 C. 3 D. 6
5. 下列（ ）图标是Access中表对象的标志。
A. B. C. D.
6. 在设计Access数据表时，“索引”属性有（ ）取值。
A. 1 B. 2 C. 3 D. 4
7. Access中包含有（ ）种数据类型。

加入CSDN，享受更精准的内容推荐，与300万程序员共同成长！

登录

注册

×

A. 9 B. 10 C. 7 D. 8

8. 在一个宏中要打开一个报表,应该使用的操作是()。

A. OpenForm B. OpenReport C. OpenTable D. OpenQuery

9. 可以通过Internet进行数据发布的对象是()。

A. 窗体 B. 报表 C. 查询 D. 数据访问页

10. 模块窗口由()个部分组成。

A. 2 B. 3 C. 4 D. 5



5



二、填空(每空1分,共20分)

1. 在人工管理和文件管理阶段,程序设计__依赖于__数据表示。

2. 在文件系统中,存取数据的基本单位为__记录__,在数据库系统中,存取数据的基本单位为__数据项__。

3. 若实体A和B是多对多的联系,实体B和C是1对1的联系,则实体A和C是__多__对__多__的联系。

4. 在一个关系中不允许出现重复的__元组__,也不允许出现具有相同名字的__属性__。

5. 数据库系统中的四类用户分别为__数据库管理员、数据库设计员、应用程序员、终端用户__。

6. 在存取数据库的数据的过程中,使用了两个数据缓冲区,分别为__系统__缓冲区和__用户__缓冲区。

7. 学生关系中的班级号属性与班级关系中的班级号主码属性相对应,则__班级号__为学生关系中的__外码__。

8. 设一个关系A具有a1个属性和a2个元组,关系B具有b1个属性和b2个元组,则关系A'B具有__a1+b1__个属性和__a2*b2__个元组。

9. 设一个学生关系为S(学生号,姓名),课程关系为C(课程号,课程名),选课关系为X(学生号,课程号,成绩),求出所有选课的学生信息的运算表达式为__ $\sigma_{\text{学生号}(X)}$ __与__S__的自然连接。

10. 在一个关系R中,若存在 $X \rightarrow Y$ 和 $X \rightarrow Z$,则存在__ $X \rightarrow (Y,Z)$ __,称此为函数依赖的合并性规则。

三、填空(每空1分,共20分)

1. 若一个关系的任何非主属性都不部分依赖依赖于任何候选码,则称该关系达到__第二__范式。

2. 在SQL中,列级完整性约束分为__6__种情况,表级完整性约束分为__4__种情况。

3. 在SQL中,每个视图中的列可以来自不同的__表__,它是在原有表的基础上__建立__的逻辑意义上的新关系。

4. 在SQL的查询语句中,group by选项实现__分组统计__功能,order by选项实现对结果表的__排序__功能。

5. 对于较复杂的系统,概念设计阶段的主要任务是:首先根据系统的各个局部应用画出各自对应的__局部ER图__,然后再进行综合和整体设计,画出__整体ER图__。

6. 机器实现阶段的目标是在计算机系统中得到一个满足__设计要求、功能完善、操作方便__的数据库应用系统。

7. Access的用户操作界面由__标题栏、菜单栏、工作区、工具栏、状态栏等五个部分组成。

8. Access“表”结构设计窗口中上半部分的“表设计器”是由__字段名称、数据类型、说明__等三列组成。

9. Access中的窗体由__页眉、主体__和页脚等三个部分组成。

四、填空(每空1分,共20分)

1. 设一个关系为R(A,B,C,D,E),它的最小函数依赖集为 $FD=\{A \rightarrow B, A \rightarrow C, (C,D) \rightarrow E\}$,则该关系的候选码为__(A,D)__,候选码函数决定E是__伪传递__性。

2. 设一个关系为R(A,B,C,D,E),它的最小函数依赖集为 $FD=\{A \rightarrow B, A \rightarrow C, (A,D) \rightarrow E\}$,该关系只满足__第一__范式,若要规范化为高一级的范式,则将得到__2__个关系。

3. 在实际的数据库管理系统中,对数据库的操作方式有__命令交互、程序执行、窗口界面__等三种。

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

登录

注册



4. 在SQL中,主码约束的关键字为____ primary key_____,外码约束的关键字为____ foreignkey _____。

5. 基本表属于全局模式中的表,它是____实表____,而视图则属于局部模式中的表,它是____虚表____。

6. 在SQL新版的查询语句中,select选项实现投影运算,from选项实现____连接____运算,where选项实现____选择____运算。

7. 数据字典是对系统工作流程中____数据____和____处理____的描述。

8. 关系数据库系统中的全局模式由若干个基本表所组成,表与表之间的联系是通过定义的____主码____和____外码____实现的。

9. 在旅店管理中,涉及到的基本表有四个,它们分别为客房表、住宿表、____旅客登记表、消费卡表____。

10. 在设置或撤消数据库密码的过程中,密码对于字母__大小写__是敏感的。

五、填空(每空1分,共10分)

1. Access的用户操作界面由_标题栏、菜单栏、工作区_、工具栏、状态栏等五个部分组成。

2. Access“表”结构设计窗口中上半部分的“表设计器”是由__字段名称、数据类型_和说明等三列组成。

3. Access中的窗体由__页眉、主体、页脚__等三个部分组成。

4. 在Access中模块分为_类模块_和__标准模块_两种类型。

六、根据主教材第四章所给的商品库和教学库,或者按照下列所给的每条SQL查询语句写出相应的功能,或者按照下列所给的每种功能写出相应的SQL查询语句。(每小题4分,共20分)

在名称为商品库的数据库中包含有商品表1和商品表2,它们的定义分别为:

商品表1(商品代号 char(8),分类名 char(8),单价 float,数量 int)

商品表2(商品代号 char(8),产地 char(6),品牌 char(6),)

在名称为教学库的数据库中包含有学生、课程和选课三个表,它们的定义分别为:

学生(学生号 char(7),姓名 char(6),性别 char(2),出生日期 datetime,

专业 char(10),年级 int)

课程(课程号 char(4),课程名 char(10),课程学分 int

选课(学生号 char(7),课程号 char(4),成绩 int)

1. select distinct 产地

from 商品表2

功能:从商品库中查询出所有商品的不同产地。

2. select *

from 学生

where 学生号 in (select 学生号

from 选课

group by 学生号 having count(*)=1

)

功能:从教学库中查询出只选修了一门课程的全部学生。

3. select *

from 学生

where 学生号 in (select 学生号

from 选课

group by 学生号 having count(*)<=2

) or not exists (select *

from 选课

where 学生.学生号=选课.学生号

)

功能:从教学库中查询出最多选修了2门课程(含未选任何课程)的全部学生。

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

4. 从商品库中查询出每类(即分类名相同)商品的最高单价。

[登录](#)

[注册](#)

✕


```
select 分类名,max(单价) as 最高单价
from 商品表1
group by 分类名
```

5. 从教学库中查询出至少选修了姓名为@m1学生所选课程中一门课的全部学生。

```
select distinct 学生.*
from 学生,选课
where 学生.学生号=选课.学生号 and 课程号=any(select 课程号
from 学生,选课
where 学生.学生号=选课.学生号 and 姓名=@m1
)
```

1. Student(S#,Sname,Sage,Ssex) 学生表

2. Course(C#,Cname,T#) 课程表

3. SC(S#,C#,score) 成绩表

4. Teacher(T#,Tname) 教师表

5.

6. 问题:

7. 1、查询“001”课程比“002”课程成绩高的所有学生的学号;

8. select a.S# from (select s#,score from SC where C#='001') a,(select s#,score

9. from SC where C#='002') b

10. where a.score>b.score and a.s#=b.s#;

11. 2、查询平均成绩大于60分的同学的学号和平均成绩;

12. select S#,avg(score)

13. from sc

14. group by S# having avg(score)>60;

15. 3、查询所有同学的学号、姓名、选课数、总成绩;

16. select Student.S#,Student.Sname,count(SC.C#),sum(score)

17. from Student left Outer join SC on Student.S#=SC.S#

18. group by Student.S#,Sname

19. 4、查询姓“李”的老师的个数;

20. select count(distinct(Tname))

21. from Teacher

22. where Tname like '李%';

23. 5、查询没学过“叶平”老师课的同学的学号、姓名;

24. select Student.S#,Student.Sname

25. from Student

26.

where S# not in (select distinct(SC.S#) from SC,Course,Teacher where SC.C#=Course.C# and Teacher.T#=Course.T# and Teacher.Tname='叶平');

27. 6、查询学过“001”并且也学过编号“002”课程的同学的学号、姓名;

28.

select Student.S#,Student.Sname from Student,SC where Student.S#=SC.S# and SC.C#='001'and exists(Select * from SC as SC_2 where SC_2.S#=SC.S# and SC_2.C#='002');

29. 7、查询学过“叶平”老师所教的所有课的同学的学号、姓名;

30. select S#,Sname

31. from Student

32.

where S# in (select S# from SC ,Course ,Teacher where SC.C#=Course.C# and Teacher.T#=Course.T# and Teacher.Tname='叶平' group by S# having count(SC.C#)=(select count(C#) from Course,Teacher where Teacher.T#=Course.T# and Tname='叶平'));

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

33. 8、查询课程编号“002”的成绩比课程编号“001”课程低的所有同学的学号、姓名；

34.
 Select S#,Sname from (select Student.S#,Student.Sname,score ,(select score from SC SC_2 where SC_2.S#=Student.S# and SC_2.C#='002') score2
 35. from Student,SC where Student.S#=SC.S# and C#='001') S_2 where score2 <score;

36. 9、查询所有课程成绩小于60分的同学的学号、姓名；

37. select S#,Sname
 38. from Student
 39.
 40. where S# not in (select Student.S# from Student,SC where S.S#=SC.S# and score>60);

41. 10、查询没有学全所有课的同学的学号、姓名；

41. select Student.S#,Student.Sname
 42. from Student,SC
 43.
 44. where Student.S#=SC.S# group by Student.S#,Student.Sname having count(C#) <(select count(C#) from Course);

45. 11、查询至少有一门课与学号为“1001”的同学所学相同的同学的学号和姓名；

45. select S#,Sname from Student,SC where Student.S#=SC.S# and C# in (select C# from SC where S#='1001');

46. 12、查询至少学过学号为“001”同学所有一门课的其他同学学号和姓名；

47. select distinct SC.S#,Sname
 48. from Student,SC
 49. where Student.S#=SC.S# and C# in (select C# from SC where S#='001');

50. 13、把“SC”表中“叶平”老师教的课的成绩都更改为此课程的平均成绩；

51. update SC set score=(select avg(SC_2.score)
 52. from SC SC_2
 53. where SC_2.C#=SC.C#) from Course,Teacher where Course.C#=SC.C# and Course.T#=Teacher.T# and Teacher.Tname='叶平');

54. 14、查询和“1002”号的同学学习的课程完全相同的其他同学学号和姓名；

55. select S# from SC where C# in (select C# from SC where S#='1002')
 56. group by S# having count(*)=(select count(*) from SC where S#='1002');

57. 15、删除学习“叶平”老师课的SC表记录；

58. Delect SC
 59. from course ,Teacher
 60. where Course.C#=SC.C# and Course.T#= Teacher.T# and Tname='叶平';

61. 16、向SC表中插入一些记录，这些记录要求符合以下条件：没有上过编号“003”课程的同学学号、2、
 62. 号课的平均成绩；

63. Insert SC select S#,'002',(Select avg(score)
 64. from SC where C#='002') from Student where S# not in (Select S# from SC where C#='002');

65. 17、按平均成绩从高到低显示所有学生的“数据库”、“企业管理”、“英语”三门的课程成绩，按如下形式显示： 学生ID,,数据库,企业管理,英语,有效课程数,有效平均分

66. SELECT S# as 学生ID
 67. ,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='004') AS 数据库
 68. ,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='001') AS 企业管理
 69. ,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='006') AS 英语
 70. ,COUNT(*) AS 有效课程数, AVG(t.score) AS 平均成绩
 71. FROM SC AS t
 72. GROUP BY S#
 73. ORDER BY avg(t.score)

74. 18、查询各科成绩最高和最低的分：以如下形式显示： 课程ID，最高分，最低分

75. SELECT L.C# AS 课程ID,L.score AS 最高分,R.score AS 最低分
 76. FROM SC L ,SC AS R
 77. WHERE L.C# = R.C# and
 78. L.score = (SELECT MAX(IL.score)
 79. FROM SC AS IL,Student AS IM
 80. WHERE L.C# = IL.C# and IM.S#=IL.S#
 81. GROUP BY IL.C#)
 82. AND
 83. R.Score = (SELECT MIN(IR.score)
 84. FROM SC AS IR
 85. WHERE R.C# = IR.C#
 86. GROUP BY IR.C#
 87.);
 88. 19、按各科平均成绩从低到高和及格率的百分数从高到低顺序
 89. SELECT t.C# AS 课程号,max(course.Cname)AS 课程名,isnull(AVG(score),0) AS 平均成
 90. 绩
 91. ,100 * SUM(CASE WHEN isnull(score,0)>=60 THEN 1 ELSE 0 END)/COUNT(*) AS 及
 92. 格百分数
 93. FROM SC T,Course
 94. where t.C#=course.C#
 95. GROUP BY t.C#
 96. ORDER BY 100 * SUM(CASE WHEN isnull(score,0)>=60 THEN 1 ELSE 0 END)/COUNT(*) DE
 97. SC
 98. 20、查询如下课程平均成绩和及格率的百分数(用"1行"显示): 企业管理 (001) , 马克思
 99. (002) , OO&UML (003) , 数据库 (004)
 100. SELECT SUM(CASE WHEN C# = '001' THEN score ELSE 0 END)/SUM(CASE C# WHEN '001' T
 101. HEN 1 ELSE 0 END) AS 企业管理平均分
 102. ,100 * SUM(CASE WHEN C# = '001' AND score >= 60 THEN 1 ELSE 0 END)/SUM(CASE WH
 103. EN C# = '001' THEN 1 ELSE 0 END) AS 企业管理及格百分数
 104. ,SUM(CASE WHEN C# = '002' THEN score ELSE 0 END)/SUM(CASE C# WHEN '002' THEN
 105. 1 ELSE 0 END) AS 马克思平均分
 106. ,100 * SUM(CASE WHEN C# = '002' AND score >= 60 THEN 1 ELSE 0 END)/SUM(CASE WH
 107. EN C# = '002' THEN 1 ELSE 0 END) AS 马克思及格百分数
 108. ,SUM(CASE WHEN C# = '003' THEN score ELSE 0 END)/SUM(CASE C# WHEN '00
 109. 3' THEN 1 ELSE 0 END) AS UML平均分
 110. ,100 * SUM(CASE WHEN C# = '003' AND score >= 60 THEN 1 ELSE 0 END)/SUM(
 111. CASE WHEN C# = '003' THEN 1 ELSE 0 END) AS UML及格百分数
 112. ,SUM(CASE WHEN C# = '004' THEN score ELSE 0 END)/SUM(CASE C# WHEN '00
 113. 4' THEN 1 ELSE 0 END) AS 数据库平均分
 114. ,100 * SUM(CASE WHEN C# = '004' AND score >= 60 THEN 1 ELSE 0 END)/SUM(
 115. CASE WHEN C# = '004' THEN 1 ELSE 0 END) AS 数据库及格百分数
 116. FROM SC
 117. 21、查询不同老师所教不同课程平均分从高到低显示
 118. SELECT max(Z.T#) AS 教师ID,MAX(Z.Tname) AS 教师姓名,C.C# AS 课程
 119. I D,MAX(C.Cname) AS 课程名称,AVG(Score) AS 平均成绩
 120. FROM SC AS T,Course AS C,Teacher AS Z
 121. where T.C#=C.C# and C.T#=Z.T#
 122. GROUP BY C.C#
 123. ORDER BY AVG(Score) DESC
 124. 22、查询如下课程成绩第3名到第6名的学生成绩单: 企业管理 (001) , 马克思

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



(002) , UML (003) , 数据库 (004)

112. [学生ID],[学生姓名],企业管理,马克思,UML,数据库,平均成绩

113. SELECT DISTINCT top 3

114. SC.S# As 学生学号,

115. Student.Sname AS 学生姓名 ,

116. T1.score AS 企业管理,

117. T2.score AS 马克思,

118. T3.score AS UML,

119. T4.score AS 数据库,

120. ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) + ISNULL(T4.score,0) as 总分

121. FROM Student,SC LEFT JOIN SC AS T1

122. ON SC.S# = T1.S# AND T1.C# = '001'

123. LEFT JOIN SC AS T2

124. ON SC.S# = T2.S# AND T2.C# = '002'

125. LEFT JOIN SC AS T3

126. ON SC.S# = T3.S# AND T3.C# = '003'

127. LEFT JOIN SC AS T4

128. ON SC.S# = T4.S# AND T4.C# = '004'

129. WHERE student.S#=SC.S# and

130. ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) + ISNULL(T4.score,0)

131. NOT IN

132. (SELECT

133. DISTINCT

134. TOP 15 WITH TIES

135. ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) + ISNULL(T4.score,0)

136. FROM sc

137. LEFT JOIN sc AS T1

138. ON sc.S# = T1.S# AND T1.C# = 'k1'

139. LEFT JOIN sc AS T2

140. ON sc.S# = T2.S# AND T2.C# = 'k2'

141. LEFT JOIN sc AS T3

142. ON sc.S# = T3.S# AND T3.C# = 'k3'

143. LEFT JOIN sc AS T4

144. ON sc.S# = T4.S# AND T4.C# = 'k4'

145. ORDER BY ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) + ISNULL(T4.score,0) DESC);

146.

147. 23、统计列印各科成绩,各分数段人数:课程ID,课程名称,[100-85],[85-70],[70-60],[<60]

148. SELECT SC.C# as 课程ID, Cname as 课程名称

149. ,SUM(CASE WHEN score BETWEEN 85 AND 100 THEN 1 ELSE 0 END) AS [100 - 85]

150. ,SUM(CASE WHEN score BETWEEN 70 AND 85 THEN 1 ELSE 0 END) AS [85 - 70]

151. ,SUM(CASE WHEN score BETWEEN 60 AND 70 THEN 1 ELSE 0 END) AS [70 - 60]

152. ,SUM(CASE WHEN score < 60 THEN 1 ELSE 0 END) AS [60 -]

153. FROM SC, Course

154. where SC.C#=Course.C#

155. GROUP BY SC.C#,Cname;

156.

157. 24、查询学生平均成绩及其名次

158. SELECT 1/(SELECT COUNT(distinct 平均成绩)

159. FROM (SELECT S#,AVG(score) AS 平均成绩

加入CSDN, 享受更精准的内容推荐, 与500万程序员共同成长!

登录

注册

×

160. FROM SC
 161. GROUP BY S#
 162.) AS T1
 163. WHERE 平均成绩 > T2.平均成绩) as 名次,
 164. S# as 学生学号,平均成绩
 165. FROM (SELECT S#,AVG(score) 平均成绩
 166. FROM SC
 167. GROUP BY S#
 168.) AS T2
 169. ORDER BY 平均成绩 desc;
 170.
 171. 25、查询各科成绩前三名的记录:(不考虑成绩并列情况)
 172. SELECT t1.S# as 学生ID,t1.C# as 课程ID,Score as 分数
 173. FROM SC t1
 174. WHERE score IN (SELECT TOP 3 score
 175. FROM SC
 176. WHERE t1.C#= C#
 177. ORDER BY score DESC
 178.)
 179. ORDER BY t1.C#;
 180. 26、查询每门课程被选修的学生数
 181. select c#,count(S#) from sc group by C#;
 182. 27、查询出只选修了一门课程的全部学生的学号和姓名
 183. select SC.S#,Student.Sname,count(C#) AS 选课数
 184. from SC ,Student
 185. where SC.S#=Student.S# group by SC.S# ,Student.Sname having count(C#)=1;
 186. 28、查询男生、女生人数
 187. Select count(Ssex) as 男生人数 from Student group by Ssex having Ssex='男';
 188. Select count(Ssex) as 女生人数 from Student group by Ssex having Ssex='女';
 189. 29、查询姓“张”的学生名单
 190. SELECT Sname FROM Student WHERE Sname like '张%';
 191. 30、查询同名同性学生名单，并统计同名人数
 192. select Sname,count(*) from Student group by Sname having count(*)>1;;
 193. 31、1981年出生的学生名单(注：Student表中Sage列的类型是datetime)
 194. select Sname, CONVERT(char(11),DATEPART(year,Sage)) as age
 195. from student
 196. where CONVERT(char(11),DATEPART(year,Sage))='1981';
 197. 32、查询每门课程的平均成绩，结果按平均成绩升序排列，平均成绩相同时，按课程号降序排列
 198. Select C#,Avg(score) from SC group by C# order by Avg(score),C# DESC ;
 199. 33、查询平均成绩大于85的所有学生的学号、姓名和平均成绩
 200. select Sname,SC.S# ,avg(score)
 201. from Student,SC
 202. where Student.S#=SC.S# group by SC.S#,Sname having avg(score)>85;
 203. 34、查询课程名称为“数据库”，且分数低于60的学生姓名和分数
 204. Select Sname,isnull(score,0)
 205. from Student,SC,Course
 206. where SC.S#=Student.S# and SC.C#=Course.C# and Course.Cname='数据库'
 'and score <60;
 207. 35、查询所有学生的选课情况;
 208. SELECT SC.S#,SC.C#,Sname,Cname
 209. FROM SC,Student,Course
 210. where SC.S#=Student.S# and SC.C#=Course.C# ;
 211. 36、查询任何一门课程成绩在70分以上的姓名、课程名称和分数;
 212. SELECT distinct student.S#,student.Sname,SC.C#,SC.score

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



213. FROM student,Sc
 214. WHERE SC.score>=70 AND SC.S#=student.S#;
 215. 37、查询不及格的课程,并按课程号从大到小排列
 216. select c# from sc where score <60 order by C# ;
 217. 38、查询课程编号为003且课程成绩在80分以上的学生的学号和姓名;
 218. select SC.S#,Student.Sname from SC,Student where SC.S#=Student.S# and Score>80 and C#='003';
 219. 39、求选了课程的学生人数
 220. select count(*) from sc;
 221. 40、查询选修“叶平”老师所授课程的学生中,成绩最高的学生姓名及其成绩
 222. select Student.Sname,score
 223. from Student,SC,Course C,Teacher
 224. where Student.S#=SC.S# and SC.C#=C.C# and C.T#=Teacher.T# and Teacher.Tname='叶平' and SC.score=(select max(score)from SC where C#=C.C#);
 225. 41、查询各个课程及相应的选修人数
 226. select count(*) from sc group by C#;
 227. 42、查询不同课程成绩相同的学生的学号、课程号、学生成绩
 228. select distinct A.S#,B.score from SC A ,SC B where A.Score=B.Score and A.C# <>B.C#;
 229. **43、查询每门功课成绩最好的前两名**
 230. SELECT t1.S# as 学生ID,t1.C# as 课程ID,Score as 分数
 231. FROM SC t1
 232. WHERE score IN (SELECT TOP 2 score
 233. FROM SC
 234. WHERE t1.C#= C#
 235. ORDER BY score DESC
 236.)
 237. ORDER BY t1.C#;
 238. 44、统计每门课程的学生选修人数(超过10人的课程才统计)。要求输出课程号和选修人数,查询结果按人数降序排列,查询结果按人数降序排列,若人数相同,按课程号升序排列
 239. select C# as 课程号,count(*) as 人数
 240. from sc
 241. group by C#
 242. order by count(*) desc,c#
 243. 45、检索至少选修两门课程的学生学号
 244. select S#
 245. from sc
 246. group by s#
 247. having count(*) > = 2
 248. 46、查询全部学生都选修的课程的课程号和课程名
 249. select C#,Cname
 250. from Course
 251. where C# in (select c# from sc group by c#)
 252. 47、查询没学过“叶平”老师讲授的任一课程的学生姓名
 253. select Sname from Student where S# not in (select S# from Course,Teacher,SC where Course.T#=Teacher.T# and SC.C#=course.C# and Tname='叶平');
 254. 48、查询两门以上不及格课程的同学的学号及其平均成绩
 255. select S#,avg(isnull(score,0)) from SC where S# in (select S# from SC where score <60 group by S# having count(*)>2)group by S#;
 256. 49、检索“004”课程分数小于60,按分数降序排列的同学学号
 257. select S# from SC where C#='004'and score <60 order by score desc;
 258. 50、删除“002”同学的“001”课程的成绩
 259. delete from SC where S#='001'and C#='001';
 260.

加入CSDN, 享受更精准的内容推荐, 与500万程序员共同成长!

登录

注册

×



5



模拟考核试题参考答案



一、单选(每空1分,共10分)

1. D 2. B 3. D 4. B 5. C 6. C 7. B 8. B 9. D 10. B



二、填空(每空1分,共20分)



1. 依赖于
2. 记录、数据项
3. 多、多
4. 元组、属性
5. 数据库管理员、数据库设计员、应用程序员、终端用户(次序无先后)
6. 系统、用户
7. 班级号、外码
8. $a1+b1$ 、 $a2 \cdot b2$
9. \bar{O} 学生号(X)、S (次序无先后)
10. $X \rightarrow (Y, Z)$



三、填空(每空1分,共20分)

1. 第二
2. 6、4
3. 表、建立
4. 分组统计、排序
5. 局部ER图、整体ER图
6. 设计要求、功能完善、操作方便
7. 标题栏、菜单栏、工作区(次序无先后)
8. 字段名称、数据类型、说明(次序无先后)
9. 页眉、主体(次序无先后)

四、填空(每空1分,共20分)

1. (A,D)、伪传递
2. 第一、2
3. 命令交互、程序执行、窗口界面(次序无先后)
4. primary key、foreign key
5. 实表、虚表
6. 连接、选择
7. 数据、处理(次序无先后)
8. 主码、外码(次序无先后)
9. 旅客登记表、消费卡表(次序无先后)
10. 大小写

五、填空(每空1分,共10分)

1. 标题栏、菜单栏、工作区(次序无先后)
2. 字段名称、数据类型(次序无先后)
3. 页眉、主体、页脚(次序无先后)
4. 类模块、标准模块(次序无先后)

六、根据主教材第四章所给的商品库和教学库,或者按照下列所给的每条SQL查询语句写出相应的功能,或者按照下列所给的每种功能写出相应的SQL查询语句。(每小题4分,共20分)

1. 从商品库中查询出所有商品的不同产地。
2. 从教学库中查询出只选修了一门课程的全部学生。
3. 从教学库中查询出最多选修了2门课程(含未选任何课程)的全部学生。
4. select 分类名,max(单价) as 最高单价

from 商品表1

group by 分类名

5. select distinct 学生*

from 学生,选课

加入CSDN,享受更精准的内容推荐,与500万程序员共同成长!

[登录](#)[注册](#)

```

where 学生.学生号=选课.学生号 and 课程号=any(select 课程号
from 学生,选课
where 学生.学生号=选课.学生号 and 姓名=@m1
)

```

3.1 名词解释



5



- (1)SQL模式：SQL模式是表和授权的静态定义。一个SQL模式定义为基本表的集合。一个由模式名和模式拥有者的用户名或账号来确定,并包含模式中每一个元素(基本表、视图、索引等)的定义。
- (2)SQL数据库：SQL(Structured Query Language)，即‘结构式查询语言’，采用英语单词表示和结构式的语法规则。一个SQL数据库是表的汇集，它用一个或多个SQL模式定义。
- (3)基本表：在SQL中，把传统的关系模型中的关系模式称为基本表(Base Table)。基本表是实际存储在数据库中的表，对应一个关系。
- (4)存储文件：在SQL中，把传统的关系模型中的存储模式称为存储文件(Stored File)。每个存储文件与外部存储器上一个物理文件对应。
- (5)视图：在SQL中，把传统的关系模型中的子模式称为视图(View)，视图是从若干基本表和（或）其他视图构造出来的表。
- (6)行：在SQL中，把传统的关系模型中的元组称为行(row)。
- (7)列：在SQL中，把传统的关系模型中的属性称为列(coloumn)。
- (8)实表：基本表被称为“实表”，它是实际存放在数据库中的表。
- (9)虚表：视图被称为“虚表”，创建一个视图时，只把视图的定义存储在数据词典中，而不存储视图所对应的数据。
- (10)相关子查询：在嵌套查询中出现的符合以下特征的子查询：子查询中查询条件依赖于外层查询中的某个值，所以子查询的处理不只一次，要反复求值，以供外层查询使用。
- (11)联接查询：查询时先对表进行笛卡尔积操作，然后再做等值联接、选择、投影等操作。联接查询的效率比嵌套查询低。
- (12)交互式SQL：在终端交互方式下使用的SQL语言称为交互式SQL。
- (13)嵌入式SQL：嵌入在高级语言的程序中使用的SQL语言称为嵌入式SQL。
- (14)共享变量：SQL和宿主语言的接口。共享变量有宿主语言程序定义，再用SQL的DECLARE语句说明，SQL语句就可引用这些变量传递数据库信息。
- (15)游标：游标是与某一查询结果相联系的符号名，用于把集合操作转换成单记录处理方式。
- (16)卷游标：为了克服游标在推进时不能返回的不便，SQL2提供了卷游标技术。卷游标在推进时不但能沿查询结果中元组顺序从头到尾一行行推进，也能一行行返回。

3.2 对于教学数据库的三个基本表

学生S(S#,SNAME,AGE,SEX)

学习 SC(S#,C#,GRADE)

课程C(C#,CNAME,TEACHER)

试用SQL的查询语句表达下列查询：

- (1) 检索LIU老师所授课程的课程号和课程名。

```

SELECT C#, CNAME
FROM C
WHERE TEACHER= 'LIU'

```

- (2) 检索年龄大于23岁的男学生的学号和姓名。

```

SELECT S#, SNAME
FROM S
WHERE (AGE>23) AND (SEX= 'M' )

```

- (3) 检索至少选修LIU老师所授课程中一门课程的女学生姓名。

```

SELECT SNAME
FROM S

```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！
WHERE SEX= 'F' AND S# IN

[登录](#)
[注册](#)



```
(SELECT S#
FROM SC
WHERE C# IN
  (SELECT C#
   FROM C
   WHERE TEACHER= 'LIU' )
```



5



NOTICE: 有多种写法, 比如联接查询写法:

```
SELECT SNAME
FROM S,SC,C
WHERE SEX= 'F' AND SC.S#=S.S#
AND SC.C#=C.C#
AND TEACHER='LIU'
```

但上一种写法更好一些。



(4) 检索WANG同学不学的课程的课程号。

```
SELECT C#
FROM C
WHERE C# NOT IN
  (SELECT C#
   FROM SC
   WHERE S# IN
    (SELECT S#
     FROM S
     WHERE SNAME='WANG'))
```

(5) 检索至少选修两门课程的学生学号。

```
SELECT DISTINCT X.SNO
FROM SC X,SC Y
WHERE X.SNO=Y.SNO AND X.CNO<>Y.CNO
```

Notice:对表SC进行自连接, X, Y是SC的两个别名。

(6) 检索全部学生都选修的课程的课程号与课程名。

```
SELECT C#,CNAME
FROM C
WHERE NOT EXISTS
  (SELECT *
   FROM S
   WHERE S# NOT IN
    (SELECT *
     FROM SC
     WHERE SC.C#=C.C#))
```

要从语义上分解: (1) 选择课程的课程号与课程名, 不存在不选这门课的同学。

其中, “不选这门课的同学” 可以表示为:

SELECT *	或者 SELECT *
FROM S	FROM S
WHERE S# NOT IN	WHERE NOT EXISTS
(SELECT *	(SELECT *
FROM SC	FROM SC
WHERE SC.C#=C.C#)	WHERE S.S#=C.S# AND
	SC.C#=C.C#)

```

SELECT DISTINCT S#
FROM SC
WHERE C# IN
  (SELECT C#
   FROM C
   WHERE TEACHER='LIU'))

```



5

3.3 设有两个基本表R (A, B, C) 和S (D, E, F) , 试用SQL查询语句表达下列关系代数表达式:



(1) $\pi_A(R)$ (2) $\sigma_{B='17'}(R)$ (3) $R \times S$ (4) $\pi_{A,F}(\sigma_{C=D}(R \times S))$

```

(1) SELECT A FROM R
(2) SELECT * FROM R WHERE B='17'
(3) SELECT A,B,C,D,E,F FROM R,S
(4) SELECT A,F FROM R,S WHERE R.C=S.D

```

3.4.3.4 设有两个基本表R (A, B, C) 和S (A, B, C) 试用SQL查询语句表达下列关系代数表达式:

(1) $R \cup S$ (2) $R \cap S$ (3) $R - S$ (4) $\pi_{A,B}(R) \pi_{B,C}(S)$

```

(1) SELECT A,B,C
   FROM R
   UNION
   SELECT A,B,C
   FROM S

```

```

(2) SELECT A,B,C
   FROM R
   INTERSECT
   SELECT A,B,C
   FROM S

```

```

(3) SELECT A,B,C
   FROM R
   WHERE NOT EXISTS
     (SELECT A,B,C
      FROM S
      WHERE R.A=S.A AND R.B=S.B AND R.C=S.C)

```

```

(4) SELECT R.A,R.B,S.C
   FROM R,S
   WHERE R.B=S.B

```

3.5 试叙述SQL语言的关系代数特点和元组演算特点。
(P61-62)

3.6 试用SQL查询语句表达下列对教学数据库中三个基本表S、SC、C的查询:

(1) 统计有学生选修的课程门数。
SELECT COUNT(DISTINCT C#) FROM SC

(2) 求选修C4课程的学生的平均年龄。
SELECT AVG(AGE)
FROM S

```

WHERE S# IN
(SELECT S#

```

加入CSDN, 享受更精准的内容推荐, 与500万程序员共同成长!

[登录](#)
[注册](#)


```
FROM SC
WHERE C#='C4')
```

或者,

```
SELECT AVG(AGE)
FROM S, SC
WHERE S.S#=SC.S# AND C#='004'
```



5



- (3) 求LIU老师所授课程的每门课程的学生平均成绩。

```
SELECT CNAME,AVG(GRADE)
FROM SC,C
WHERE SC.C#=C.C# ANDTEACHER='LIU'
GROUP BY C#
```



- (4) 统计每门课程的学生选修人数（超过10人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列。



```
SELECT DISTINCT C#,COUNT(S#)
FROM SC
GROUP BY C#
HAVING COUNT(S#)>10
ORDER BY 2 DESC, C# ASC
```



- (5) 检索学号比WANG同学大，而年龄比他小的学生姓名。

```
SELECT X.SNAME
FROM S AS X, S AS Y
WHERE Y.SNAME='WANG' AND X.S#>Y.S# AND X.AGE<Y.AGE
```

- (6) 检索姓名以WANG打头的所有学生的姓名和年龄。

```
SELECT SNAME,AGE
FROM S
WHERE SNAME LIKE 'WANG%'
```

- (7) 在SC中检索成绩为空值的学生学号和课程号。

```
SELECT S#,C#
FROM SC
WHERE GRADE IS NULL
```

- (8) 求年龄大于女同学平均年龄的男学生姓名和年龄。

```
SELECT SNAME,AGE
FROM S AS X
WHERE X.SEX='男' AND X.AGE>(SELECT AVG(AGE)FROM S AS Y WHERE Y.SEX='女')
```

- (9) 求年龄大于所有女同学年龄的男学生姓名和年龄。

```
SELECT SNAME,AGE
FROM S AS X
WHERE X.SEX='男' AND X.AGE>ALL (SELECT AGE FROM S AS Y WHERE Y.SEX='女')
```

3.7 试用SQL更新语句表达对教学数据库中三个基本表S、SC、C的各个更新操作：

- (1) 往基本表S中插入一个学生元组（‘S9’，‘WU’，18）。

```
INSERT INTO S(S#,SNAME,AGE) VALUES('59','WU',18)
```

- (2) 在基本表S中检索每一门课程成绩都大于等于80分的学生学号、姓名和性别，并把检索到的值送往另一个已存在的基本表STUDENT（S#，SANME，SEX）。

```
INSERT INTO STUDENT(S#,SNAME,SEX)
```

```
SELECT S#, SNAME,SEX
FROM S WHERE NOT EXISTS
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

[登录](#)
[注册](#)


```
(SELECT * FROM SC WHERE
  GRADE<80 AND S.S#=SC.S#)
```

- (3) 在基本表SC中删除尚无成绩的选课元组。

```
DELETE FROM SC
  WHERE GRADE IS NULL
```



5



- (4) 把WANG同学的学习选课和成绩全部删去。

```
DELETE FROM SC
  WHERE S# IN
    (SELECT S#
     FROM S
     WHERE SNAME='WANG')
```

- (5) 把选修MATHS课不及格的成绩全改为空值。

```
UPDATE SC
  SET GRADE=NULL
  WHERE GRADE<60 AND C# IN
    (SELECT C#
     FROM C
     WHERE CNAME='MATHS')
```

- (6) 把低于总平均成绩的女同学成绩提高5%。

```
UPDATE SC
  SET GRADE=GRADE*1.05
  WHERE GRADE<(SELECT AVG(GRADE) FROM SC) AND S# IN (SELECT S# FROM
  SWHERE SEX='F')
```

- (7) 在基本表SC中修改C4课程的成绩，若成绩小于等于75分时提高5%，若成绩大于75分时提高4%（用两个UPDATE语句实现）。

```
UPDATE SC
  SET GRADE=GRADE*1.05
  WHERE C#='C4' AND GRADE<=75
UPDATE SC
  SET GRADE=GRADE*1.04
  WHERE C#='C4' AND GRADE>75
```

3.8 在第1章例1.4中提到“仓库管理”关系模型有五个关系模式：

零件 PART (P#, PNAME, COLOR, WEIGHT)

项目 PROJECT (J#, JNAME, DATE)

供应商 SUPPLIER (S#, SNAME, SADDR)

供应 P__P (J#, P#, TOTOAL)

采购 P__S (P#, S#, QUANTITY)

- (1) 试用SQLDDL语句定义上述五个基本表，并说明主键和外键。

```
CREATE TABLE PART
(P# CHAR(4) NOT NULL,PNAME CHAR(12) NOT NULL,
COLOR CHAR(10),WEIGHT REAL,
PRIMARY KEY(P#))
```

```
CREATE TABLE PROJECT
(J# CHAR(4) NOT NULL,JNAME CHAR(12) NOT NULL,
DATE DATE,
PRIMARY KEY(J#))
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

[登录](#)
[注册](#)


```
CREATE TABLE SUPLIER
(S# CHAR(4) NOT NULL,SNAME CHAR(12),SADDR VARCHAR(20),
PRIMARY KEY(S#))
```

```
CREATE TABLE P_P
(J# CHAR(4),P# CHAR(4),TOTAL INTEGER,
PRIMARY KEY(J#,P#),
FOREIGN KEY(J#) REFERENCE PROJECT(J#),
FOREIGN KEY(P#) REFERENCE PART(P#))
```

```
CREATE TABLE P_S
(P# CHAR(4),S# CHAR(4),QUANTITY INTEGER,
PRIMARY KEY(P#,S#),
FOREIGN KEY(P#) REFERENCE PART(P#),
FOREIGN KEY(S#) REFERENCE SUPLIER(S#))
```

(2) 试将PROGECT、P__P、PART三个基本表的自然联接定义为一个视图VIEW1，PART、P__S、SUPPLIER 三个基本表的自然联接定义为一个视图VIEW2。

```
CREATE VIEW VIEW1(J#,JNAME,DATE,P#,PNAME,COLOR,WEIGHT,TOTAL)
AS SELECT
PROJECT.J#,JNAME,DATE,PART.P#,PNAME,COLOR,WEIGHT,TOTAL
FROM PROJECT,PART,P_P
WHERE PART.P#=P_P.P# AND P_P.J#=PROJECT.J#
```

```
CREATE VIEW VIEW2(P#,PNAME,COLOR,WEIGHT,S#,SNAME,SADDR,QUANTITY)
AS SELECT
PART.P#,PNAME,COLOR,WEIGHT,SUPPLIER.S#,SNAME,SADDR,QUANTITY
FROM PART,P_S,SUPPLIER
WHERE PART.P#=P_S.P# AND P_S.S#=SUPPLIER.S#
```

(3) 试在上述两个视图的基础上进行数据查询：

1) 检索上海的供应商所供应的零件的编号和名字。

```
SELECT P#,PNAME FROM VIEW2 WHERE SADDR='SHANGHAI'
```

2) 检索项目J4所用零件的供应商编号和名字。

```
SELECT S#,SNAME FROM VIEW2 WHERE P# IN(SELECT P# FROM VIEW1 WHERE
J#='J4')
```

3.9 对于教学数据库中基本表SC，已建立下列视图：

```
CREATE VIEWS__GRADE (S# , C__NUM , AVG__GRADE)
AS SELECT S# , COUNT(C#) , AVG (GRADE)
FROM SC
GROUP BY S#
```

试判断下列查询和更新是否允许执行。若允许，写出转换到基本表SC上的相应操作。

(1) SELECT*

```
FROM S__GRADE
允许
SELECT S#,COUNT(C#),AVG(GRADE)FROM SC GROUP BY S#
```

(2) SELECT S# , C__NUM

```
FROM S__GRADE
WHERE AVG__GRADE > 80
允许
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

×

```
SELECT S#,COUNT(C#) FROM SC WHEREAVG(GRADE)>80
```

```
(3) SELECTS# , AVG__GRADE
FROMS__GRADE
WHEREC__NUM>(SELECTC__NUM
FROMS__GRADE
WHERE S# = 'S4' )
允许
5 SELECT S#,AVG(GRADE)
FROM SC AS X
WHERE COUNT(X.C#)>(SELECTCOUNT(Y.C#) FROM SC AS Y WHERE Y.S#='S4')
GROUP BY S#
```

```
(4) UPDATES__GRADE
SETC__NUM=C__NUM+1
WHERE S# = 'S4'
不允许
```

```
(5) DELETEDFROMS__GRADE
WHEREC__NUM>4
不允许
```

3.10 预处理方式对于嵌入式SQL的实现有什么重要意义？

预处理方式是先用预处理程序对源程序进行扫描，识别出SQL语句，并处理成宿主语言的函数调用形式；然后再用宿主语言的编译程序把源程序编译成目标程序。这样，不用扩充宿主语言的编译程序，就能处理SQL语句。

3.11 在宿主语言的程序中使用SQL语句有哪些规定？

在宿主语言的程序中使用SQL语句有以下规定：

- (1) 在程序中要区分SQL语句与宿主语言语句
- (2) 允许嵌入的SQL语句引用宿主语言的程序变量（称为共享变量），但有两项规定：
 - 1) 引用时，这些变量前必须加“：”作为前缀标识，以示与数据库中变量有区别。
 - 2) 这些变量由宿主语言的程序定义，并用SQL的DECLARE语句说明。
- (3) SQL的集合处理方式与宿主语言单记录处理方式之间要协调。需要采用游标机制，把集合操作转换成单记录处理方式。

3.12SQL的集合处理方式与宿主语言单记录处理方式之间如何协调？

由于SQL语句处理的是记录集合，而宿主语言语句一次只能处理一个记录，因此需要用游标(cursor)机制，把集合操作转换成单记录处理方式。

2.13 嵌入式SQL语句何时不必涉及到游标？何时必须涉及到游标？

(1)INSERT、DELETE、UPDATE语句，查询结果肯定是单元组时的SELECT语句，都可直接嵌入在主程序中使用，不必涉及到游标。

(2)当SELECT语句查询结果是多个元组时，此时宿主语言程序无法使用，一定要用游标机制把多个元组一次一个地传送给宿主语言处理。



经典SQL练习题

qaz13177_58_

2010年05月10日 17:04

125586

今天在网上找了几道经典的SQL练习题做了一下，虽然都不难，但是对打基础是很有好处的，在明白的基础上可以进一步做分析，来研究一下各种解法的优劣，甚至进行简单的优化。。现在将题目和答案分享一下。我使用的是...

(http://blog.csdn.net/qaz13177_58_/article/details/5575711)

SQL查询语句练习题27道

friendan

2012年10月15日 13:26

107080

练习环境为：XP+SQL2000数据库练习使用的数据库为：学生管理数据库数据库下载地址为: http://download.csdn.net/download/friendan/4648150 ...

(http://blog.csdn.net/friendan/article/details/8072668)

一位超过10年工作经验的程序员总结！

广告

这做了10年web开发的程序员的总结分析给你，也可能是我们正在经历的人生。他的这22条总结希望可以帮到你少走弯路...

http://www.baidu.com/cb.php?c=lgF_pyfqnHmknj0dP1f0IZ0qnfK9ujYzP1ndPWb10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1dWmymLrAmsPhwWPWFbPy7B0AwY5HDdnHf4rHcdrHm0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEIAqspynElvNBnHqdlAdxTvqdThP-5yF_UvTkn0KzujYznfKBuHYs0ZKz5H00ly-b5HDdP1f1PWD0Uv-b5HDzrH63nHf0mv-b5HTzPWb1n6KElv3qn0KsXHYznjm0mLFW5H61n1c)

【SQL练习】经典SQL练习题

mrbcy

2017年04月03日 18:12

4367

准备数据建表语句CREATE TABLE students (sno VARCHAR(3) NOT NULL, sname VARCHAR(4) NOT NULL, ssex VARCHAR(2)) ...

(http://blog.csdn.net/mrbcy/article/details/68965271)

Sql经典题库（附答案）

2012年02月17日 10:28

59KB

下载

DOC

SQL练习50题整理

mitsuki_blog

2016年07月21日 12:29

3544

CREATE DATABASE 50q; USE 50q; CREATE TABLE STUDENT (SNO VARCHAR(3) NOT NULL, SNAME VA...

(http://blog.csdn.net/mitsuki_blog/article/details/51981743)

4

工业触摸显示器

530.00/台

10/12/15/17/19寸触控工业显示器电容触摸工

5

福利厂直销 串口dvd刻录机台式电脑内置光驱

36.00/台

6

6+2滑轨式显卡延长线 557电源线 显卡6P转

2.80/条

sql老师学生经典模式(50题)-练习

2015年10月28日 13:02

5KB

下载

??

Sql server练习50题

2012年10月28日 10:23

327KB

下载

RAR

SQL练习题整理

suheonline

2013年12月18日 15:48

3384

有2张表，为父子关系表： 1 Employee 表 id Name Age 1 张三 31 2 李四 28 3 王五 40 ... 2 Wages 表 -----...

(http://blog.csdn.net/suheonline/article/details/17395821)

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录注册

31 of 331/3/18, 23:17

SQL基础练习题

xu200yang

2017年04月06日 14:19

📖 583

不得不说SQL是数据分析师入门的基础，想开始从事数据分析工作的同学最好将SQL练一练再找工作。大概看了一个月的MYSQL从入门到精通，无奈纸上谈兵，没有实战经验，于是从网上找了一些练习题，整理成下文。...

(http://blog.csdn.net/xu200yang/article/details/69389165)

很全的sql语句练习题（+sql脚本文件）

qq_22075041

2017年04月02日 17:42

📖 3460

欢迎使用Markdown编辑器写博客本Markdown编辑器使用StackEdit修改而来，用它写博客，将会带来全新的体验哦： Markdown和扩展Markdown简洁的语法代码块高亮图片链接和...

(http://blog.csdn.net/qq_22075041/article/details/68953187)

超经典SQL练习题，做完这些你的SQL就过关了

flycat296

2017年03月19日 10:19

📖 3116

SQL 联系 题 使用方法：我用的数据库是 Ms SQL Server 2008 ,练习时应当自己建数据，自己先思考，切勿急躁翻答案！ 否则效果减半，做完这些，恭喜你，你的 SQL 就算过关了。...

(http://blog.csdn.net/flycat296/article/details/63681089)

SQL数据库经典练习题

DH2442897094

2017年09月29日 12:22

📖 437

SQL数据库面试题以及答案（50例题） Student(S#,Sname,Sage,Ssex)学生表 S#： 学号 Sname： 学生姓名 Sage： 学生年龄 Ssex： 学生性别 Course(C#,Cn...

(http://blog.csdn.net/DH2442897094/article/details/78132667)

SQL完整的练习题，对初学者和高手都有用。

2010年04月08日 15:57

15KB

📄 下载

sql超级经典（练习题+答案）

2012年08月08日 15:16

372KB

📄 下载

sql常见面试题（我觉得太好了）

u012467492

2015年07月07日 15:29

📖 82547

Sql常见面试题 受用了 1. 用一条SQL 语句 查询出每门课都大于80 分的学生姓名 name kecheng fenshu 张三 语文 81 张三 ...

(http://blog.csdn.net/u012467492/article/details/46790205)

SQL练习题（含答案）

2008年10月25日 16:28

76KB

📄 下载

SQL Server 练习题3

zjxc

2005年05月31日 20:27

📖 11799

题目3 问题描述： 本题用到下面三个关系表： CARD 借书卡。 CNO 卡号， NAME 姓名， CLASS 班级BOOKS 图书。 BN O 书号， BNAME 书名,AUTHOR...

(http://blog.csdn.net/zjxc/article/details/385114)

SQL50道练习题（附答案）

2014年04月24日 12:18

78KB

📄 下载

SQL测试试题及答案

2011年12月22日 22:52 303KB [下载](#)



SQL语句强化练习题及答案

jiangsanfeng1111 2016年04月18日 11:16 1399

SQL语句强化练习题及答案一、简单查询 1、列出全部学生的信息。SELECT * FROM 学生 2、列出软件专业全部学生的学号及姓名。SELECT 学号,姓名 FROM 学生...
 5 (http://blog.csdn.net/jiangsanfeng1111/article/details/51178165)

