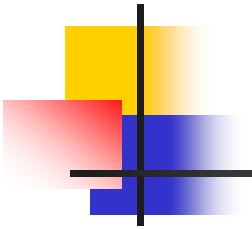


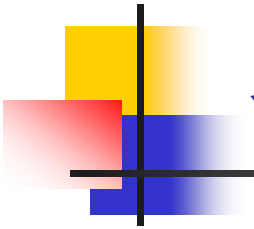
# 数据库习题课

2018年4月3日

# 习题1

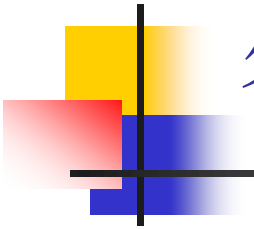
- 数据库如下：
  - Member（memb_no, name, age)
  - Book(isbn, title, authors, publisher)
  - Borrowed(memb_no, isbn, date)

  - 打印借阅了任意由“McGraw-Hill”出版的书的会员名字
  - 打印借阅了所有由“McGraw_Hill”出版的书的会员名



# 习题1

- 对于每个出版商，打印借阅了多于五本由该出版商出版的书的会员名字
- 打印每位会员借阅书籍数量的平均值(若某会员没有借阅任何书籍，那么该会员根本不会出现在borrowed关系中)



# 答案1

- select name from member m, book b, borrowed l where m.memb_no = l.memb_no and l.isbn = b.isbn and b.publisher = 'McGrawHill'
- select distinct m.name from member m

where not exists ((select isbn from book where publisher = 'McGrawHill') except (select isbn from borrowed l where l.memb_no = m.memb_no))

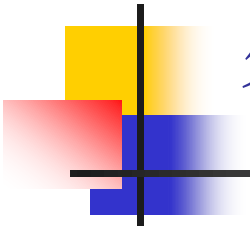
- select publisher, name

from (select publisher, name, count (isbn)

from member m, book b, borrowed l

where m.memb_no = l.memb_no and l.isbn = b.isbn group by publisher, name)

as membpub(publisher, name, count books) where count books > 5



# 答案1

- with memcount as (select count(*) from member) select count(*)/memcount from borrowed

# 习题2

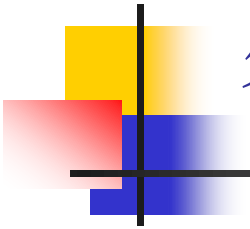
- Employee(employee_name, street, city)
- works(employee_name, company_name, salary)
- Company(company_name, city)
- Managers(employee_name, manager_name)

- 找出数据库中所有不为"First Bank Corporation"工作的雇员
- 找出雇员最多的公司
- 找出平均工资高于“First Bank Corporation”平均工资的公司

# 答案2

- Select employee name from works where company name ≠ 'First Bank Corporation'
- select company name

```
from works
group by company_name
having count (distinct employee name) >= all
        (select count (distinct employee name)
        from works
        group by company name)
```



# 答案2

- select company name

from works

group by company name

having avg (salary) > (select avg (salary)

from works

where company name = ’First Bank Corporation’)

# 习题3

Consider the following expressions, which use the result of a relational algebra operation as the input to another operation. For each expression, explain in words what the expression does.

a. $\sigma_{year \geq 2009}(takes) \bowtie student$

b. $\sigma_{year \geq 2009}(takes \bowtie student)$

c. $\Pi_{ID, name, course\_id}(student \bowtie takes)$

# 答案3

a. For each student who takes at least one course in 2009, display the students information along with the information about what courses the student took. The attributes in the result are:

*ID, name, dept_name, tot_cred, course_id, section_id, semester, year, grade*

b. Same as (a); selection can be done before the join operation.

c. Provide a list of consisting of

*ID, name, course_id*

of all students who took any course in the university.

# 习题4

R(BH, XM, XB, DWH)

S(DWH, DWM)

T(BH, XM, XB, DWH)

有如右所示的关系表R，S，T。写出下面关系代数对应的SQL语句

1） $\sigma_{DWH='100'}(R)$

2） $\prod_{XM,XB}(R)$

3） $\prod_{XM,DWH}(\sigma_{XB='女'}(R))$

4） $R \infty S$

5） $\prod_{XM,XB,DWH}(\sigma_{XB='男'}(R \infty S))$

# 答案4

```
SELECT * FROM R WHERE DWH='100';
SELECT XM,XB FROM R;
SELECT XM,DWH FROM R WHERE XB='女';
SELECT R.*,S.DWM FROM R, S WHERE R.DWH=S.DWH;
SELECT XM,XB,DWH FROM R,S WHERE R.DWH=S.DWH AND XB='男';
```

# 习题5

设有如下关系模式：

student(NO, NAME , SEX ,BIRTHDAY, CLASS)

teacher(NO,NAME,SEX,BIRTHDAY,PROF,DEPART)　　PROF 为职称，DEPART 为系别

course(CNO, CNAME, TNO)

score(NO, CNO, DEGREE)　　DEGREE 为成绩

查询至少有 2 名男生的班号；

查询每个学生的姓名和年龄；

查询和“李军”同性别并同班的所有同学的姓名；

查询最低分大于 70，最高分小于 90 的学生的学号；

# 答案5

```
SELECT CLASS FROM student WHERE SEX= '男'
        GROUP BY CLASS HAVING COUNT(*)>=2;

SELECT NAME, year(date())-year(birthday) as age FROM student;

SELECT name FROM student WHERE sex=(SELECT sex FROM student WHERE name='李军' ) and
    class=(SELECT class FROM student WHERE name='李军');

SELECT no FROM score GROUP BY no HAVING min(degree)>70 and max(degree)<90;
```

# 习题6

设有如下关系：

S（S#,SNAME,AGE,SEX）/*学生（学号，姓名，年龄，性别）*/

C（C#,CNAME,TEACHER）/*课程（课程号，课程名，任课教师）*/

SC（S#,C#,GRADE）/*成绩（学号，课程号，成绩）*/

查询：

（1） 教师“程军”所授课程的课程号和课程名；

（2） “李强”同学不学课程的课程号；

（3） 至少选修了课程号为 k1 和 k5 的学生学号；

（4） 选修课程包含学号为 2 的学生所修课程的学生学号。

# 答案6

(1) $\Pi_{C\#,CNAME}(\sigma_{TEACHER='程军'}(C))$

(2) $\Pi_{C\#}(C)-\Pi_{C\#}((\sigma_{NAME='李强'}(S))\infty SC)$

(3) $\Pi_{S\#,C\#}(SC)\div\Pi_{C\#}(\sigma_{C\#='k1'\cup C\#='k5'}(C))$

(4) $\Pi_{S\#,C\#}(SC)\div\Pi_{C\#}(\sigma_{C\#=2}(SC))$

# 习题7

*member*(*memb_no*, *name*, *dob*)
*books*(*isbn*, *title*, *authors*, *publisher*)
*borrowed*(*memb_no*, *isbn*, *date*)

Write the following queries in relational algebra.

a. Find the names of members who have borrowed any book published by "McGraw-Hill".

b. Find the name of members who have borrowed all books published by "McGraw-Hill".

c. Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill".

d. For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher.

e. Find the average number of books borrowed per member. Take into account that if an member does not borrow any books, then that member does not appear in the *borrowed* relation at all.

# 答案7

a. $t_1 \leftarrow \Pi_{isbn}(\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name}((member \bowtie borrowed) \bowtie t_1))$

b. $t_1 \leftarrow \Pi_{isbn}(\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name,isbn}(member \bowtie borrowed) \div t_1$

c. $t_1 \leftarrow member \bowtie borrowed \bowtie (\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name}\ (\sigma_{countisbn\ >\ 5}\ ((\ memb\_no\ G_{\textbf{count-distinct}(isbn)\ \textbf{as}\ countisbn}(t_1))))$

d. $t_1 \leftarrow member \bowtie borrowed \bowtie books$
$\Pi_{publisher,name}\ (\sigma_{countisbn\ >\ 5}\ ((\ publisher,memb\_no\ G_{\textbf{count-distinct}(isbn)\ \textbf{as}\ countisbn}(t_1)))$

# 习题8

*person* (*driver_id*, *name*, *address*)
*car* (*license*, *model*, *year*)
*accident* (*report_number*, *date*, *location*)
*owns* (*driver_id*, *license*)
*participated* (*report_number*, *license*, *driver_id*, *damage_amount*)

a. Find the number of accidents in which the cars belonging to "John Smith" were involved.

b. Update the damage amount for the car with license number "AABB2000" in the accident with report number "AR2197" to $3000.

# 答案8

```
select    count (*)
from      accident
where     exists
          (select *
           from participated, owns, person
           where owns.driver_id = person.driver_id
                 and person.name = 'John Smith'
                 and owns.license = participated.license
                 and accident.report_number = participated.report_number)
```

```
update participated
set damage_amount = 3000
where report_number = "AR2197" and
      license = "AABB2000")
```

# 习题9

*member*(*memb_no*, *name*, *dob*)
*books*(*isbn*, *title*, *authors*, *publisher*)
*borrowed*(*memb_no*, *isbn*, *date*)

Write the following queries in relational algebra.

a. Find the names of members who have borrowed any book published by "McGraw-Hill".

b. Find the name of members who have borrowed all books published by "McGraw-Hill".

c. Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill".

d. For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher.

# 答案9

a. $t_1 \leftarrow \Pi_{isbn}(\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name}((member \bowtie borrowed) \bowtie t_1))$

b. $t_1 \leftarrow \Pi_{isbn}(\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name,isbn}(member \bowtie borrowed) \div t_1$

c. $t_1 \leftarrow member \bowtie borrowed \bowtie (\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name}\ (\sigma_{countisbn\ >\ 5}\ ((\ memb\_no\ G_{\textbf{count-distinct}(isbn)\ \textbf{as}\ countisbn}(t_1))))$

d. $t_1 \leftarrow member \bowtie borrowed \bowtie books$
$\Pi_{publisher,name}\ (\sigma_{countisbn\ >\ 5}\ ((\ publisher,memb\_no\ G_{\textbf{count-distinct}(isbn)\ \textbf{as}\ countisbn}(t_1)))$

# 习题10

*employee* (*employee_name*, *street*, *city*)
*works* (*employee_name*, *company_name*, *salary*)
*company* (*company_name*, *city*)
*manages* (*employee_name*, *manager_name*)

a. Give all employees of First Bank Corporation a 10 percent raise.

b. Give all managers of First Bank Corporation a 10 percent raise.

c. Delete all tuples in the *works* relation for employees of Small Bank Corporation.

# 答案10

```
update works
set salary = salary * 1.1
where company_name = 'First Bank Corporation'
```

```
update works
set salary = salary * 1.1
where employee_name in (select manager_name
                        from manages)
      and company_name = 'First Bank Corporation'
```

```
delete from works
where company_name = 'Small Bank Corporation'
```