Still More on SQL

Module 8

Prof. Alex Brodsky

Database Systems

Database Management - GMU. Prof. Alex Brodsky. Module 8

1

Example Instances

R1

sid	bid	day
22	101	10/10/96
58	103	11/12/96

❖ We will use these instances of the Sailors and Reserves relations in our examples.

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

❖ If the key for the Reserves relation contained only the attributes *sid* and *bid,* how would the semantics differ?

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

Database Management - GMU. Prof. Alex Brodsky. Module 6

2

Queries With GROUP BY and HAVING

SELECT   [DISTINCT] *target-list*
FROM   *relation-list*
WHERE   *qualification*
GROUP BY   *grouping-list*
HAVING   *group-qualification*

❖ The *target-list* contains (i) attribute names (ii) terms with aggregate operations (e.g., MIN (*S.age*)).

  – The <u>attribute list (i)</u> must be a subset of *grouping-list*. Intuitively, each answer tuple corresponds to a *group,* and these attributes must have a single value per group. (A *group* is a set of tuples that have the same value for all attributes in *grouping-list*.)

Database Management - GMU. Prof. Alex Brodsky. Module 8

3

Conceptual Evaluation

❖ The cross-product of *relation-list* is computed, tuples that fail *qualification* are discarded, `*unnecessary'* fields are deleted, and the remaining tuples are partitioned into groups by the value of attributes in *grouping-list*.

❖ The *group-qualification* is then applied to eliminate some groups. Expressions in *group-qualification* must have a *single value per group*!

- – In effect, an attribute in *group-qualification* that is not an argument of an aggregate op also appears in *grouping-list*. (SQL does not exploit primary key semantics here!)

❖ One answer tuple is generated per qualifying group.

Database Management - GMU. Prof. Alex Brodsky. Module 8

4

For each red boat, find the number of reservations for this boat

SELECT B.bid, COUNT (*) AS scount
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid AND B.color='red'
GROUP BY B.bid

❖ Grouping over a join of three relations.
❖ What do we get if we remove *B.color='red'* from the WHERE clause and add a HAVING clause with this condition?
❖ What if we drop Sailors and the condition involving S.sid?

Database Management - GMU. Prof. Alex Brodsky. Module 8

5

Find those ratings for which the average age is the minimum over all ratings

❖ Aggregate operations cannot be nested! WRONG:

SELECT S.rating
FROM Sailors S
WHERE S.age = (SELECT MIN (AVG (S2.age)) FROM Sailors S2)

❖ Correct solution (in SQL/92):

SELECT Temp.rating, Temp.avgage
FROM (SELECT S.rating, AVG (S.age) AS avgage
            FROM Sailors S
            GROUP BY S.rating) AS Temp
WHERE Temp.avgage = (SELECT MIN (Temp.avgage)
                                      FROM Temp)

Database Management - GMU. Prof. Alex Brodsky. Module 8

6

Continue from previous

However, this should work on Oracle 8:

SELECT S.rating
FROM Sailors S
Group by S.rating
Having AVG(S.age) = (SELECT MIN (AVG (S2.age))
                     FROM Sailors S2
                     Group by rating);

Database Management - GMU. Prof. Alex Brodsky. Module 8

7

Conclusion so far

❖ Post processing on the result of queries is supported.

❖ Aggregation is the most complex "post processing"

– "Group by" clause partition the results into groups

– "Having" clause puts condition on groups (just like Where clause on tuples).

Database Management - GMU. Prof. Alex Brodsky. Module 8

8

Null Values

❖ Field values in a tuple are sometimes *unknown* (e.g., a rating has not been assigned) or *inapplicable* (e.g., no spouse's name).

❖ SQL provides a special value *null* for such situations.

Database Management - GMU. Prof. Alex Brodsky. Module 8

9

Deal with the null value

❖ Special operators needed to check if value is/is not *null*.

- "is null" always true or false (never unknown)

- "is not null"

❖ Is *rating>8* true or false when *rating* is equal to *null*?

- Actually, it's unknown.

- Three-valued logic

Database Management - GMU. Prof. Alex Brodsky. Module 8

10

Three valued logic

AND	False	True	Unknown
False	False	False	False
True	False	True	Unknown
Unknown	False	Unknown	Unknown

OR	False	True	Unknown
False	False	True	Unknown
True	True	True	True
Unknown	Unknown	True	Unknown

	NOT
False	True
True	False
Unknown	Unknown

Database Management - GMU. Prof. Alex Brodsky. Module 8

11

Other issues with the null value

❖ WHERE and Having clause eliminates rows that don't evaluate to true (i.e., rows evaluate to false or unknown.

❖ Aggregate functions ignore nulls (except count(*))

❖ Distinct treats all nulls as the same

Database Management - GMU. Prof. Alex Brodsky. Module 8

12

Outer joins

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

(left outer-join)

sid	bid	day
22	101	10/10/96
58	103	11/12/96

Assignment Project Exam Help

https://powcoder.com

=

Add WeChat powcoder

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
31	lubber	8	55.5	Null	Null
58	rusty	10	35.0	103	11/12/96

Database Management - GMU. Prof. Alex Brodsky. Module 8

13

In Oracle

Select *

From Sailor S, Reserve R

Where S.sid = R.sid (+);

How about:

Select S.sid, count(R.bid)

From Sailor S, Reserve R

Where S.sid = R.sid (+)

Group by S.sid;

OR

Select S.sid, count(*)

From Sailor S, Reserve R

Where S.sid = R.sid (+)

Group by S.sid;

Database Management - GMU. Prof. Alex Brodsky. Module 8

14

More outer joins

❖ Left outer join

  + sign on the right in Oracle:
  Select * from R, S where R.id=S.id(+)

❖ Right outer join

  + sign on the left in Oracle:
  Select * from R, S where R.id(+)=S.id

❖ Full outer join

  – not implemented in Oracle

Database Management - GMU. Prof. Alex Brodsky. Module 8

15

More on value functions

❖ Values can be transformed before aggregated:

Select sum(S.A/2) from S;

❖ An interesting decode function (Oracle specific):

decode(value, if1, then1, if2, then2, ..., else):

Select sum(decode(major, 'INFS', 1, 0)) as No_IS_Stu,
        sum(decode(major, 'INFS', 0, 1)) as Non_NonIS_Stu

From student ;

❖ Calculating GPA from letter grades (HW4)?

Database Management - GMU. Prof. Alex Brodsky. Module 8

16

Examples

Department (<u>D-code</u>, D-Name, Chair-SSn)
Course (<u>D-code, C-no</u>, Title, Units)
Prereq (<u>D-code, C-no, P-code, P-no</u>)
Class (<u>Class-no</u>, D-code, C-no, Instructor-SSn)
Faculty (<u>Ssn</u>, F-Name, D-Code, Rank)
Student (<u>Ssn</u>, S-Name, Major, Status)
Enrollment (<u>Class-no, Student-Ssn</u>)
Transcript (<u>Student-Ssn, D-Code, C-no,</u> Grade)

Database Management - GMU. Prof. Alex Brodsky. Module 8

17

Query 1

List the classes (class_no) taken by students

whose names start with 'T'.

select distinct class_no
from enrollment e, student s
where e.student_ssn = s.ssn and s.s_name like 'T%';

Database Management - GMU. Prof. Alex Brodsky. Module 8

18

Query 2

List the students (SSN) who are currently taking exactly one class.

select distinct student_ssn

from enrollment

group by student_ssn

having 1=count(*);

Database Management - GMU. Prof. Alex Brodsky. Module 8

19

Query 3

Give the percentage of the students (among all students) who are currently taking courses offered by ISE (D_code='ISE')

select count(distinct e.student_ssn)/count(distinct s.ssn) as Percent
from enrollment e, class c, student s
where e.class_no=c.class_no and d_code='ISE';

Database Management - GMU. Prof. Alex Brodsky. Module 8

20

Query 4

List the faculty members (F_Name) who teach 2 or more classes. List these faculty members by the number of classes they teach.

select f.f_name
from faculty f, class c
where f.ssn=c.instructor_ssn
group by f.ssn, f_name
having count(distinct c.class_no)>=2
order by count(distinct c.class_no), f_name;

Database Management - GMU. Prof. Alex Brodsky. Module 8

21

Query 5

List the students (SSN and Name) along with the number of classes they are taking. If a student is not taking any class, the student should also be listed (with 0 as the number of classes he/she is taking). The list should be ordered by the number of classes (in an ascending order), and in case of a tie, by the SSN of the students.

```
select ssn, s_name, count(distinct class_no)
from student s, enrollment e
where s.ssn = e.student_ssn (+)
group by ssn, s_name
order by count(distinct class_no), ssn;
```

Database Management - GMU. Prof. Alex Brodsky. Module 8

22

Query 6

List the faculty members (F_Name) who teach more than twice as many classes as Professor Smith (F_Name='Smith') is teaching. (Note that if Professor Smith is not teaching anything, then any professor who teaches at least one class will satisfy the above query.)

```
select f_name
from faculty f, class c
where f.ssn=c.instructor_ssn
group by f.ssn, f_name
having count(distinct c.class_no) > (select 2*count(class_no)
                                    from faculty f, class c
                                    where f.f_name='Smith'
                                    and f.ssn=c.instructor_ssn);
```

Database Management - GMU. Prof. Alex Brodsky. Module 8

23

Query 7

Find the number of departments which do not have a chairman (Chair_ssn is 'NULL').

```
select count(d_code)
from department
where chair_ssn is NULL;
```

Database Management - GMU. Prof. Alex Brodsky. Module 8

24

Query 8

For each department (d_code), give the number of graduate students (status='Grad') and the number of other students (status <> 'Grad'). The two numbers must be shown in the same row as the department code. Hint: use Decode.

select major, sum(decode(status, 'Grad', 1,0)) as Grad,
        sum(decode(status, 'Grad', 0, 1)) as NoGrad
from student
group by major;

Database Management - GMU. Prof. Alex Brodsky. Module 8

25

Query 9

For each department (d_code), give the highest rank of the professors in the department along with the number of the faculty with that highest rank. The output contains one row for each department. Assume Full>Associate>Assistant, i.e., lexicographic order is fine. Note that some departments may not have professors in some ranks (e.g., a department may not have full, or associate or assistant professors).

```
select f.D_Code as dept, f.maxrank, count(e.ssn) as num
from  (select d_code, max(rank) as maxrank
        from faculty group by d_code) f,
        faculty e
where f.d_code=e.d_code and e.rank=f.maxrank
group by f.d_code, f.maxrank
order by f.d_code
```

Database Management - GMU. Prof. Alex Brodsky. Module 8

26

Conclusion

❖ Done with most SQL stuff

❖ More on programming with SQL
  – In lecture week 14 (advanced topics)

❖ More practice!

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Database Management - GMU. Prof. Alex Brodsky. Module 8

27