大连理工大学软件学院 数据库

Intermediate SQL-2

上机答案

你的下载是我上传的动力,请不要吝啬一个财富值

Intermediate SQL-2

Using the university schema that you have write the following queries. In some cases you

might need to insert extra data to show the effect of a particular feature.

Recommendation: With clause is strongly recommended for simplifying the query.

1. Find the courses which have been offered for 2 years at least and have sections in spring,

2010. For each course as such, information displayed should involve:

\* Identifier of course(i.e. the primary key for section)

\* Title of the course

\* Number of instructors who in charge of teaching the course in spring ,2010

\* Total salary all over the instructors who in charge of teaching the course in

spring ,2010

\* Total credit hours performed per week( Note: 1 credit hour equals to 50 minutes).

2. USE outer join to construct the following query

Find all information for student registration and course offered. The students who have

never registered for any courses and the courses has never been offered. For each record in

the result, information displayed should involve:

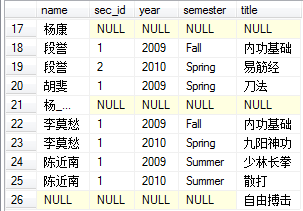
\* Identifier of student(i.e. the primary key for student)

\* Name of student

\* Identifier of section(i.e. the primary key for section)

\* Title of course.

The result should be like the following



3. USE scalar subquery to construct the following query

For all students, grade information of each student is needed. Those students who have

never registered for any section should also be considered. For each student, information

displayed should involve:

\* Identifier of student(i.e. the primary key for student)

\* Name of student

\* Department name of student

\* Number of failure for the student to pass some section. (That is the number of grade ‘F’)

\* Total number of failure of passing sections for the students in the same department as

the current student.

4. Find students who have registered for some but not all courses(PART COURSE, for short)

taught by instructors of department '拳脚学院'. Furthermore, the registration of these

students for such courses (i.e. PART COURSE above) should have grade, even the grade is

‘F’. Do this using the "not exists ... except ..." structure. For each student as such,

information displayed should involve:

\* Identifier of student(i.e. the primary key for student)

\* Name of the student

\* Number of courses, taught by instructors of department '拳脚学院', registered by the

student

5. Use EXISTS or NOT EXISTS clause in WHERE clause to construct following query.

Find those sections which have no instructor as the teacher. Moreover, these sections

should have never been registered by any student. For each section as such, information

displayed should involve:

\* Identifier of student(i.e. the primary key for student)

\* Name of the corresponding course.

\* Credits of the course

--1

with course\_twoYear(course\_id) as

(select course\_id from teaches where

course\_id in

(select course\_id

from teaches where year=2009)

intersect

(select course\_id

from teaches where year=2010 and semester='Spring')),

number(course\_id,ID,num) as (select course\_id,ID,COUNT(sec\_id)from teaches

where year=2010 and semester='Spring' group by course\_id,ID),

tot\_time(course\_id,times) as (select course\_id,SUM(credits\*50) from course group by course\_id)

select course.course\_id ,title ,num,SUM(salary)as tot\_sal,times

from course\_twoYear ,course ,number,instructor,tot\_time

where course.course\_id=course\_twoYear.course\_id and number.course\_id=course.course\_id and

instructor.ID=number.ID and course.course\_id=tot\_time.course\_id

group by course.course\_id ,title ,num,times;

--2

with stu\_take(course\_id,name,sec\_id,year,semester) as

(select course\_id,name,sec\_id,year,semester from student left join takes on student.ID=takes.ID)

select name,sec\_id,year,semester,title from stu\_take full join course on stu\_take.course\_id=course.course\_id;

--3

select student.ID,student.name,student.dept\_name,

(select COUNT(takes.grade)from takes where takes.ID=student.id and takes.grade='F'),

(select COUNT(takes.ID)

from student,takes

where student.ID=takes.ID and takes.grade='F')

from student;

--4

with T(student\_name,total\_course)as

(select student.name,COUNT(course.course\_id)

from student join takes

on student.ID=takes.ID

join course on takes.course\_id=course.course\_id

where course.dept\_name='拳脚学院'

group by student.name)

select ID,name,total\_course

from student join T on student.name=T.student\_name

where not exists(

(select student\_name from T where T.total\_course=(select

COUNT(course\_id)from course where course.dept\_name='拳脚学院'))

except

(select student\_name from T)

)

--5

select course.course\_id,title,credits

from course Full outer join teaches on course.course\_id = teaches.course\_id

Full outer join takes on course.course\_id = takes.course\_id

Full outer join section on course.course\_id = section.course\_id

where course.course\_id not in((select course.course\_id

from teaches ,course ,takes ,section

where teaches.course\_id = section.course\_id and takes.course\_id = course.course\_id

and section.course\_id = takes.course\_id )

)