2019320097_조이강 데이터베이스 실습

a. Find the titles of courses in the ‘Comp. Sci.’ department that have 3 credits

Query)

select title

from course

where dept_name = ‘Comp. Sci.’ and credits = 3;

```
chapter3=# select title from course where dept_name = 'Comp. Sci.' and c
redits = 3;
          title
-------------------------
 Robotics
 Image Processing
 Database System Concepts
(3개 행)
```

b. Find the IDs of all students who were taught by an instructor named ‘Srinivasan’; make sure there are no duplicates in the result.

Query)

select distinct t.id

from takes t, teaches s

where t.course_id = s.course_id and s.id = (select id from instructor where name = ‘Srinivasan’);

```
chapter3=# select distinct t.id
chapter3-# from takes t, teaches s
chapter3-# where t.course_id = s.course_id
chapter3-# and
chapter3-# s.id =
chapter3-# (select id from instructor where name = 'Srinivasan');
  id
-------
 00128
 12345
 45678
 54321
 76543
 98765
(6개 행)
```

c. Find the ID and name of instructors who have not given grades to students (i.e., where the grade is null in the takesrelation).

Query)

select id, name

from instructor

where id in (

select teaches.id

from teaches, takes

where teaches.course_id = takes.course_id and takes.grade is null);

```
chapter3=# select ID, name from instructor where ID in (
chapter3(# select id from teaches where course_id in(
chapter3(# select course_id from takes where grade is null));
  id   |  name
-------+--------
 76766 | Crick
 15151 | Mozart
(2개 행)


chapter3=# select ID, name from instructor where ID in (
chapter3(# select teaches.id from teaches, takes where teaches.course_id
 = takes.course_id and takes.grade is null);
  id   |  name
-------+--------
 76766 | Crick
 15151 | Mozart
(2개 행)
```

d. Find the name and department name of instructors whose department name starts with “C”, listed alphabetically by department name.

Query)

select name, dept_name

from instructor

where dept_name like ‘C%’

order by dept_name;

```
chapter3=# select name, dept_name from instructor where dept_name like '
C%' order by dept_name;
    name    | dept_name
------------+------------
 Smith      | Chemistry
 Karl       | Chemistry
 Srinivasan | Comp. Sci.
 Katz       | Comp. Sci.
 Brandt     | Comp. Sci.
(5개 행)
```

e. Find the ID and salary of the instructor(s) with the highest salary while satisfying the following conditions.

1. Use Set Operation (“EXCEPT”)

Query)

(select id, salary from intructor)

except

select distinct i.id, i.salary from instructor i, instructor n where i.salary < n.salary;

```
chapter3=# (select id, salary from instructor) except
chapter3-# select distinct i.id, i.salary from instructor i, instructor
n where i.salary < n.salary;
  id   |  salary
-------+----------
 22222 | 95000.00
(1개 행)
```

2. Use Aggregate Function

Query)

select id, salary

from instructor

where salary = (select max(salary) from instructor);

```
chapter3=# select id, salary from instructor where salary = (select max(
salary) from instructor);
  id   |  salary
-------+----------
 22222 | 95000.00
(1개 행)
```

▪ Make a relation $grade\_points(grade, points)$, which provides a conversion from letter grades in the *takes* relation to numeric scores.

▪ The tuples of the $grade\_points$ relation: (A+, 4.3), (A, 4.0), (A-, 3.7), (B+, 3.3), (B, 3.0), (B-, 2.7), (C+, 2.3), (C, 2.0), (C-, 1.7), (D+, 1.3), (D, 1.0), (D-, 0.7), (F, 0.0)

▪ The grade-points a student earns for a course offering (section) are calculated by multiplying the number of credits for the course by the numeric grade points received for that course.

▪ For simplicity, you can assume that no *takes* tuple has a null value for the grade attribute.

Query)

create table grade_points(

grade varchar(2),

points numeric(2,1),

primary key (grade)

);

insert into grade_points values

('A+', 4.3),

('A', 4.0),

('A-', 3.7),
('B+', 3.3),

('B', 3.0),

('B-', 2.7),
('C+', 2.3),

('C', 2.0),

('C-', 1.7),
('D+', 1.3),

('D', 1.0),

('D-', 0.7),
('F', 0.0) ;

```
chapter3=# select * from grade_points
chapter3-# ;
 grade | points
-------+--------
 A+    |    4.3
 A     |    4.0
 A-    |    3.7
 B+    |    3.3
 B     |    3.0
 B-    |    2.7
 C+    |    2.3
 C     |    2.0
 C-    |    1.7
 D+    |    1.3
 D     |    1.0
 D-    |    0.7
 F     |    0.0
(13개 행)
```

a. Find the total grade-points earned by the student with ID 12345, across all courses taken by the student.

Query)

select sum(

(select credits from course where course_id = t.course_id)

*

(select points from grade_points where t.grade = grade_points.grade)

)

from takes t

where t.id = ‘12345’;

```
chapter3=# select sum(
chapter3(# (select credits from course where course_id = t.course_id)
chapter3(# *
chapter3(# (select points from grade_points where t.grade = grade_points
.grade)
chapter3(# )
chapter3-# from takes t
chapter3-# where t.id = '12345';
 sum
------
 48.0
(1개 행)
```

b. Find the grade-point average (GPA) for the above student, that is, the total grade-points divided by the total credits for the associated courses.

- 평균평점= (과목별점수* 과목의학점수) / 전체학점수

Query)

select sum(

(select credits from course where course_id = t.course_id)

*

(select points from grade_points where t.grade = grade_points.grade)

)

/ sum((select credits from course where course_id = t.course_id)) as GPA

from takes t

where t.id = ‘12345’;

```
chapter3=# select sum(
chapter3(# (select credits from course where course_id = t.course_id)
chapter3(# *
chapter3(# (select points from grade_points where t.grade = grade_points
.grade)
chapter3(# )
chapter3-# / sum((select credits from course where course_id = t.course_
id)) as GPA
chapter3-# from takes t
chapter3-# where t.id = '12345';
        gpa
--------------------
 3.4285714285714286
(1개 행)
```

c. Find the ID and the grade-points average of every student.

Query)

select id, sum(

(select credits from course c where c.ourse_id = t.course_id)

*

(select points from grade_points g where t.grade = g.grade)

)

/ sum((select credits from course c where c.course_id = t.course_id)) as GPA

from takes t

where t.grade is not null

group by t.id;

```
chapter3=# select id, sum(
chapter3(# (select credits from course c where c.course_id = t.course_id
)
chapter3(# *
chapter3(# (select points from grade_points g where t.grade = g.grade)
chapter3(# )
chapter3-# / sum((select credits from course c where c.course_id = t.cou
rse_id)) as GPA
chapter3-# from takes t
chapter3-# where t.grade is not null
chapter3-# group by t.id;
  id   |        gpa
-------+--------------------
 00128 | 3.8714285714285714
 12345 | 3.4285714285714286
 19991 | 3.0000000000000000
 23121 | 2.3000000000000000
 44553 | 2.7000000000000000
 45678 | 2.0181818181818182
 54321 | 3.5000000000000000
 55739 | 3.7000000000000000
 76543 | 4.0000000000000000
 76653 | 2.0000000000000000
 98765 | 2.2571428571428571
 98988 | 4.0000000000000000
(12개 행)
```

d. Find the ID and the grade-points average of students whose GPA is greater than 3.0.

```
with gpa as(
select id, sum(
(select credits from course c where c.ourse_id = t.course_id)
*
(select points from grade_points g where t.grade = g.grade)
)
/ sum((select credits from course c where c.course_id = t.course_id)) as GPA
from takes t
where t.grade is not null
group by t.id;
)
```

select gpa.id, gpa.gpa

from gpa

where gpa.gpa > 3;

```
chapter3=# with gpa as (
chapter3(# select id, sum(
chapter3(# (select credits from course c where c.course_id = t.course_id
)
chapter3(# *
chapter3(# (select points from grade_points g where t.grade = g.grade)
chapter3(# )
chapter3(# / sum((select credits from course c where c.course_id = t.cou
rse_id)) as GPA
chapter3(# from takes t
chapter3(# where t.grade is not null
chapter3(# group by t.id
chapter3(# )
chapter3-# select gpa.id, gpa.gpa
chapter3-# from gpa
chapter3-# where gpa.gpa > 3;
  id   |        gpa
-------+--------------------
 00128 | 3.8714285714285714
 12345 | 3.4285714285714286
 54321 | 3.5000000000000000
 55739 | 3.7000000000000000
 76543 | 4.0000000000000000
 98988 | 4.0000000000000000
(6개 행)
```