

# Data Base Assignment due 9/26

정보대학 컴퓨터학과 2017320108 고재영

#1. University 데이터 베이스 관련한 테이블 추가

- 11개의 릴레이션 테이블을 추가:

create table ~ (

); 등으로 각각의 릴레이션 테이블 생성

- 그 이후 foreign key을 추가

alter table ~ add foreign key ~ references ~; 으로 foreign key 추가

- 그 결과, Wd로 다음과 같은 11개 릴레이션 목록을 볼 수 있다.

The screenshot shows a Windows desktop with a Google Chrome browser in the background and a SQL Shell (psql) window in the foreground. The SQL Shell window displays the following text:

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
postgres 사용자의 암호:
psql (11.5)
도움말을 보려면 "help"를 입력하십시오.

postgres=# \d
릴레이션(relation) 목록
스키마 | 이름 | 종류 | 소유주
-----+-----+-----+-----
public | advisor | 테이블 | postgres
public | classroom | 테이블 | postgres
public | course | 테이블 | postgres
public | department | 테이블 | postgres
public | instructor | 테이블 | postgres
public | prereq | 테이블 | postgres
public | section | 테이블 | postgres
public | student | 테이블 | postgres
public | takes | 테이블 | postgres
public | teaches | 테이블 | postgres
public | time_slot | 테이블 | postgres
(11개 행)

postgres=#
```

Below the SQL Shell window, the text "#2. 강의 콘텐츠의 유형 추가" and "#3. 10개 이상의 스테이트먼트 실행" are visible.

```

postgres=# \d classroom
              "public.classroom" 테이블
      컬럼          |      종류      | Collation | NULL 허용 | 초기값
-----|-----|-----|-----|-----
building | character varying(20) |           | not null |
room_no  | character varying(3)  |           | not null |
capacity | character varying(3)  |           |           |
인덱스들:
"classroom_pkey" PRIMARY KEY, btree (building, room_no)
다음에서 참조됨:
TABLE "section" CONSTRAINT "section_building_fkey" FOREIGN KEY (building, room_no) REFERENCES classroom(building, room_no)

postgres=# \d course
              "public.course" 테이블
      컬럼          |      종류      | Collation | NULL 허용 | 초기값
-----|-----|-----|-----|-----
course_id | character varying(8) |           | not null |
title     | character varying(50) |           |           |
dept_name | character varying(20) |           |           |
credits   | numeric(2,0)         |           |           |
인덱스들:
"course_pkey" PRIMARY KEY, btree (course_id)
참조키 제약 조건:
"course_dept_name_fkey" FOREIGN KEY (dept_name) REFERENCES department(dept_name)
다음에서 참조됨:
TABLE "prereq" CONSTRAINT "prereq_course_id_fkey" FOREIGN KEY (course_id) REFERENCES course(course_id)
TABLE "prereq" CONSTRAINT "prereq_prereq_id_fkey" FOREIGN KEY (prereq_id) REFERENCES course(course_id)

postgres=# \d department

```

## #2. 임의 인스턴스 튜플 추가

- 최소 10명의 instructor, 20명의 student, 5개의 class, 10개의 section을 추가하는 지시에 따라야 함.
- 바로 sql shell에 입력하는 데에 안정성이 떨어지기 때문에, 해당 레포트에 instance를 포함한 표를 먼저 적을 것임.
- insert into ~ value ~; 커맨드로 튜플 추가
- 아래의 테이블들은 필자가 임의로 작성함. (주의: semester에 관해 spring, summer가 아니라 1학기, 2학기로 숫자로 생각하여 집어넣었음.)

<department 테이블>

dept_name	building	budget
Physics	Nobel	48500000
Phys. Edu.	Jordan	95000000
History	Chaeo	27000000
Biology	Mendel	50000000
Music	Hendel	30000000
Comp. Sci.	Turing	90000000

<instructor 테이블>

ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Messi	Phys. Edu.	99000
32343	El Said	History	60000
45565	Mozart	Music	80000
10101	Crick	Biology	65000
58594	Eminem	Music	84000
84512	Abraham	Comp. Sci.	92000
14141	Ozil	Phys. Edu.	72000
76543	Gold	Physics	75000
28452	Kroos	Phys. Edu.	80000

<student 테이블>

ID	name	dept_name	tot_cred
31611	Pepe	Physics	95
68733	Park	Music	34
16547	Brandt	Comp. Sci.	94
35455	Eriksen	Music	134
29571	Kim	Physics	130
20697	Hazard	History	105
13682	Havertz	Phys. Edu.	62
55199	Arnold	Physics	102
41666	Douglas	Comp. Sci.	90

14480	Ruiz	History	132
-------	------	---------	-----

10개 더 이어서

ID	name	dept_name	tot_cred
76265	Kanye	Physics	83
65666	Paulo	Biology	96
22590	Claudio	Biology	94
84116	Sergio	Comp. Sci.	61
99995	Ramos	Phys. Edu.	117
29671	Modric	Music	61
42951	Pique	Physics	26
25365	Jesse	Phys. Edu.	89
40387	Pirlo	History	24
22337	Inzaghi	Comp. Sci.	96

<course 테이블>

course_id	title	dept_name	credits
PHEK200	Sport Sociology	Phys. Edu.	2
PHEK201	Sport Pedagogy	Phys. Edu.	3
LIBS200	Human Metabolism	Biology	3
LIBS301	Molecular Biology	Biology	3
HOEW112	Modern Japan History	History	3
HOEW113	Medieval History	History	3

HOEW210	Oriental Civilization	History	2
COSE115	Java	Comp. Sci.	3
COSE214	Data Structure	Comp. Sci.	3
COSE318	Operating System	Comp. Sci.	3
PHYS213	Electromagnetics	Physics	3
PHYS311	Statistical Physics	Physics	3
MUS210	Romanticism	Music	2
PHYS312	Quantum Mechanics	Physics	3
MUS220	Harmony Composition	Music	3

<section 테이블>

course_id	sec_id	semester	year	building	room_no	time_slot_id
PHEK200	00	1	2018	Jordan	101	2
PHEK201	01	1	2016	Jordan	104	5
LIBS200	01	2	2017	Mendel	101	6
LIBS301	02	1	2016	Mendel	102	5
HOEW112	01	2	2018	Chaeho	101	5
HOEW113	02	2	2018	Chaeho	101	6
PHYS312	01	2	2019	Nobel	201	1
MUS210	01	1	2019	Hendel	101	1

COSE214	02	1	2016	Turing	301	2
COSE316	03	2	2017	Turing	201	3

<teaches 테이블>

ID	course_id	sec_id	semester	year
28452	PHEK201	01	1	2016
10101	LIBS200	01	2	2017
58594	MUS210	01	1	2019
84512	COSE214	02	1	2016
84512	COSE316	03	2	2017

<classroom 테이블>

building	room_no	capacity
Nobel	101	110
Nobel	201	125
Nobel	202	140
Jordan	101	150
Jordan	102	200
Jordan	103	80
Jordan	104	120

Chaeho	101	100
Mendel	101	120
Mendel	102	60
Hendel	101	120
Turing	101	100
Turing	201	70
Turing	301	150

<advisor 테이블>

s_id	i_id
31611	22222
68733	58594
16547	84512
35455	45565
29571	76543
20697	32343
13682	28452
55199	22222
41666	84512
14480	32343

s_id	i_id
76265	76543
65666	10101
22590	10101
84116	84512
99995	14141
29671	45565
42951	76543
25365	14141
40387	32343
22337	84512

<department 테이블>

dept_name	building	budget
Physics	Nobel	48500000
Phys. Edu.	Jordan	95000000
History	Chaeho	27000000
Biology	Mendel	50000000
Music	Hendel	30000000
Comp. Sci.	Turing	90000000

#3. 10개 이상의 시퀀스 스테이트먼트를 자유롭게 실행



1. instructor 테이블 내용을 출력하라.

```
postgres=# select *
postgres=# from instructor
postgres=# ;
```

id	name	dept_name	salary
22222	Einstein	Physics	95000.00
12121	Messi	Phys. Edu.	99000.00
32343	El Said	History	60000.00
45565	Mozart	Music	80000.00
10101	Crick	Biology	65000.00
58594	Eminem	Music	84000.00
84512	Abraham	Comp. Sci.	92000.00
14141	Ozil	Phys. Edu.	72000.00
76543	Gold	Physics	75000.00
28452	Kroos	Phys. Edu.	80000.00

(10개 행)

2. 총 학점이 60점 이상인 학생들의 이름을 출력하라.

```
postgres=# select name
postgres=# from student
postgres=# where tot_cred > 60
postgres=# ;
   name
-----
Pepe
Brandt
Eriksen
Kim
Hazard
Havertz
Arnold
Douglas
Ruiz
Kanye
Paulo
Claudio
Sergio
Ramos
Modric
Jesse
Inzaghi
(17개 행)

postgres=#
```

3. 컴퓨터 학과 소속 학생들의 수를 찾아라.

```

postgres=# select count(name)
postgres-# from student
postgres-# where dept_name = 'Comp. Sci.';
count
-----
      4
(1개 행)

postgres=#

```

4. 이름이 P로 시작하는 학생의 이름과 해당 학생이 어느 학과 소속인지를 알아내라.  
 추가적으로, 그 학생들의 이름을 사전순으로 나열하라.

```

postgres=# select name, dept_name
postgres-# from student
postgres-# where name like 'P%'
postgres-# order by name;
name | dept_name
-----+-----
Park | Music
Paulo | Biology
Pepe | Physics
Pique | Physics
Pirlo | History
(5개 행)

```

5. 물리학과 소속 학생들이 졸업에 얼마나 임박해있는지 알기 위해, 물리학과 소속 학생들의 이름과, 총 이수학점을, 이수학점에 관하여 내림차순으로 보여라.

```

postgres=# select name, tot_cred
postgres=# from student
postgres=# where dept_name = 'Physics'
postgres=# order by tot_cred desc;

```

name	tot_cred
Kim	130
Arnold	102
Pepe	95
Kanye	83
Pique	26

(5개 행)

6. 교수자 ID가 시작과 끝, 가운데에 1이 들어가 있는 교수자들의 교수ID, 이름, 학과를 출력하라.

```

postgres=# select ID, name, dept_name
postgres=# from instructor
postgres=# where ID like '1_1_1';

```

id	name	dept_name
12121	Messi	Phys. Edu.
10101	Crick	Biology
14141	Ozil	Phys. Edu.

(3개 행)

7. 학과별 소속 교수들의 평균 연봉을 구하라. 이 때, 연봉에 대하여 오름차순으로 나열하라.

```

postgres=# select dept_name, avg(salary)
postgres=# from instructor
postgres=# group by dept_name
postgres=# order by avg(salary);
 dept_name |          avg
-----+-----
History    | 60000.000000000000
Biology    | 65000.000000000000
Music      | 82000.000000000000
Phys. Edu. | 83666.666666666667
Physics    | 85000.000000000000
Comp. Sci. | 92000.000000000000
(6개 행)

```

8. 학교의 교수진 중 실제로 강의를 하는 교수의 교수ID, 이름, 학과명, 연봉, 코스 ID, 분반 ID, 학기, 강의시행년도를 출력하라.

```

postgres=# select *
postgres=# from instructor natural join teaches;
 id | name | dept_name | salary | course_id | sec_id | semester | year
-----+-----+-----+-----+-----+-----+-----+-----
28452 | Kroos | Phys. Edu. | 80000.00 | PHEK201 | 1 | 1 | 2016
10101 | Crick | Biology | 65000.00 | LIBS200 | 1 | 2 | 2017
58594 | Eminem | Music | 84000.00 | MUS210 | 1 | 1 | 2019
84512 | Abraham | Comp. Sci. | 92000.00 | COSE214 | 2 | 1 | 2016
84512 | Abraham | Comp. Sci. | 92000.00 | COSE316 | 3 | 2 | 2017
(5개 행)

```

9. 전체 학생들 중에서 학과 별 학생수를 백분위로 나타내라.

```

postgres=# select dept_name, count(id)*100/20 as portion
postgres=# from student
postgres=# group by dept_name;
 dept_name | portion
-----+-----
History    | 15
Physics    | 25
Music      | 15
Comp. Sci. | 20
Phys. Edu. | 15
Biology    | 10
(6개 행)

```

10. 물리학과와 어떤 교수보다 더 많은 연봉을 받는 교수들의 이름과 그 교수의 연봉을, 연봉에 관한 오름차순으로 나타내라.

```
postgres=# select name, salary
postgres=# from instructor
postgres=# where salary > some ( select salary
postgres=# from instructor
postgres=# where dept_name = 'Physics')
postgres=# order by salary;
```

name	salary
Mozart	80000.00
Kroos	80000.00
Eminem	84000.00
Abraham	92000.00
Einstein	95000.00
Messi	99000.00

(6개 행)