

# REPORT

## 데이터베이스 과제 #2

제출일 : 2019.10.15. ~ 2019.10.27. 23:59



이름 : 구명회  
학번 : 20154215  
분반 : 01분반

## 과제 Index

- 과제 1. 학생, 학과, 코스, 강의, 강사, 수강, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해 데이터 모델링하기(7단계)
- 과제 2. 학생, 학과, 코스, 강의, 강사, 수강, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해 ERwin을 사용해서 E-R Diagram 설계하기
- 과제 3. 학생, 학과, 코스, 강의, 강사, 수강, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해 E-R Diagram 설계한 것을 Table로 변환하기
- 과제 4. 학생, 학과, 코스, 강의, 강사, 수강, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해 각 Table마다 샘플 데이터를 입력하고 질의 수행하기
- 과제 5. 소감
- 과제 0. Code

## 참고 사이트

1. 테이블, 속성, 관계 : <https://victorydntmd.tistory.com/126>
2. 테이블, 속성, 관계 : <http://home.zany.kr:9003/zDirectPages/ERWin/chap03.html>
3. 테이블, 속성, 관계 : <https://loveiskey.tistory.com/46>
4. 테이블, 속성, 관계 : <https://m.blog.naver.com/gongtong/150135598792>
5. 조인 : <http://blog.naver.com/PostView.nhn?blogId=islove8587&logNo=220932123363&parentCategoryNo=&categoryNo=&viewDate=&isShowPopularPosts=true&from=search>

## 사용 버전

ERwin : Erwin Academic  
ORACLE : 11G R2

## 준비

과제2 시작하기 전에, 과제 1에서 생성했던 계정 MH에 접속했습니다.

```
SQL*Plus: Release 11.2.0.1.0 Production on 일 10월 27 22:06:45 2019
Copyright (c) 1982, 2010, Oracle. All rights reserved.

사용자명 입력: MH
비밀번호 입력:

다음에 접속됨:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> _
```

과제 1. 학생, 학과, 과정, 교수, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해  
데이터 모델링하기(7단계)

1. 데이터 모델링 7단계

업무분석 > 개체도출 > 속성도출 > 속성 중 식별자 도출 > 관계 설정 > 관계 차수 설정 > 관계 식별 파악

1] 업무분석

**학생, 학과, 코스, 강사, 성적, 장학금** 등의 정보로 구성되는 대학교 정보 데이터베이스를 만들어야 합니다.  
**학생**은 **학생고유번호**, **학생이름**, **학생전화번호**, **학생주소**를 가지고, **학과**는 **학과고유번호**, **학과이름**을 가지며  
**강사**는 **강사고유번호**, **강사이름**, **강사전화번호**를 가지고 있습니다. **학생과 강사**는 **학과**를 정할 수 있습니다.  
**강의**는 **강의고유번호**, **강의이름**과 **강의수강료**를 가지며 **성적**은 **성적의 범위** 정보를 가지고 있습니다.  
**강의**는 여러 분반으로 나뉘는 **강의**가 있을 수 있고, **가르치는 교수와 학생의 정보와 성적이 저장됩니다.**  
**장학금**은 **학생 한사람이 하나의 장학금만 받을 수 있습니다.**

2] 실체도출

학생, 학과, 강의, 강사, 성적, 장학금

3] 속성도출

학생(STUDENT) - 학생고유번호, 학생이름, 학생주소-구, 학생주소-동  
학과(DEPARTMENT) - 학과고유번호, 학과이름  
강의(COURSE) - 강의고유번호, 강의이름, 강의수강료  
강사(PROFESSOR) - 강사고유번호, 강사이름, 강사전화번호  
성적(STUDENT\_GRADE) - 성적고유번호, 성적의 등급  
장학금(SCH) - 장학이름, 장학금액, 학생이름

4] 속성 중 식별자 도출

학생(STUDENT) - 학생고유번호  
학과(DEPARTMENT) - 학과고유번호  
강의(COURSE) - 강의고유번호  
강사(PROFESSOR) - 강사고유번호  
성적(STUDENT\_GRADE) - 성적고유번호  
장학금(SCH) - 학생 고유번호, 장학이름

5, 6] 관계 설정, 관계 차수 설정

과정(강의)는 수강료를 받습니다. 1:1

학생은 장학금을 하나만 받을 수 있고, 장학금은 여러 학생들에게 줄 수 있습니다. 1:N

학생은 여러 개의 학과를 선택할 수 있고, 학과는 여러 학생에게 신청받을 수 있습니다. N:M

교수는 여러 의 학과를 선택할 수 있고, 학과는 여러 교수에게 신청받을 수 있습니다. N:M

학생은 여러 과정을 수강할 수 있고, 과정(강의)은 여러 학생이 동시에 수강등록을 할 수 있습니다. N:M

교수는 여러 과정을 강의할 수 있고, 과정(강의)은 여러 교수가 동시에 강의등록을 할 수 있습니다. N:M

성적은 학생이 수강한 과정을 평가할 수 있습니다. N:M

7] 관계 식별 여부 파악 (identification / non-identification)

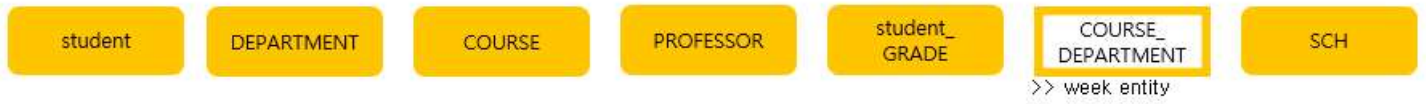
장학금과(SCH) WEEK\_ENTITY인 COURSE\_DIVIDE(CD)를 제외하고 모두 비식별관계로 설정했습니다.

+ WEEK\_ENTITY를 표현하기 위해..

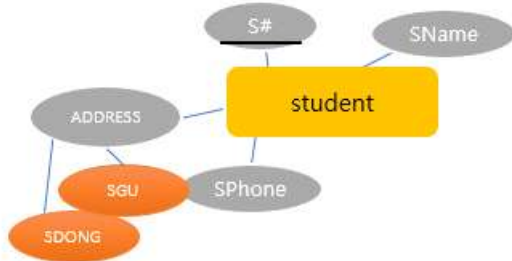
COURSE와 식별관계로 연결한 새로운 RELATION에 OWNER인 COURSE의 기본키를 속성으로 추가했습니다.

## 2] 실체 도출

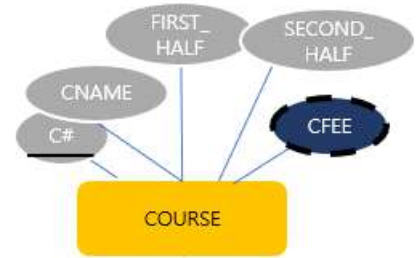
학생, 학과, 강의, 교수, 성적, 분반, 장학금



## 3, 4] 속성 도출, 속성중 식별자 도출 중에서 유도속성, 복합속성, 다치속성



[ADDRESS : 복합속성으로, SGU와 SDONG을 가집니다 A의 주소 : (광산구) + (신가동)]

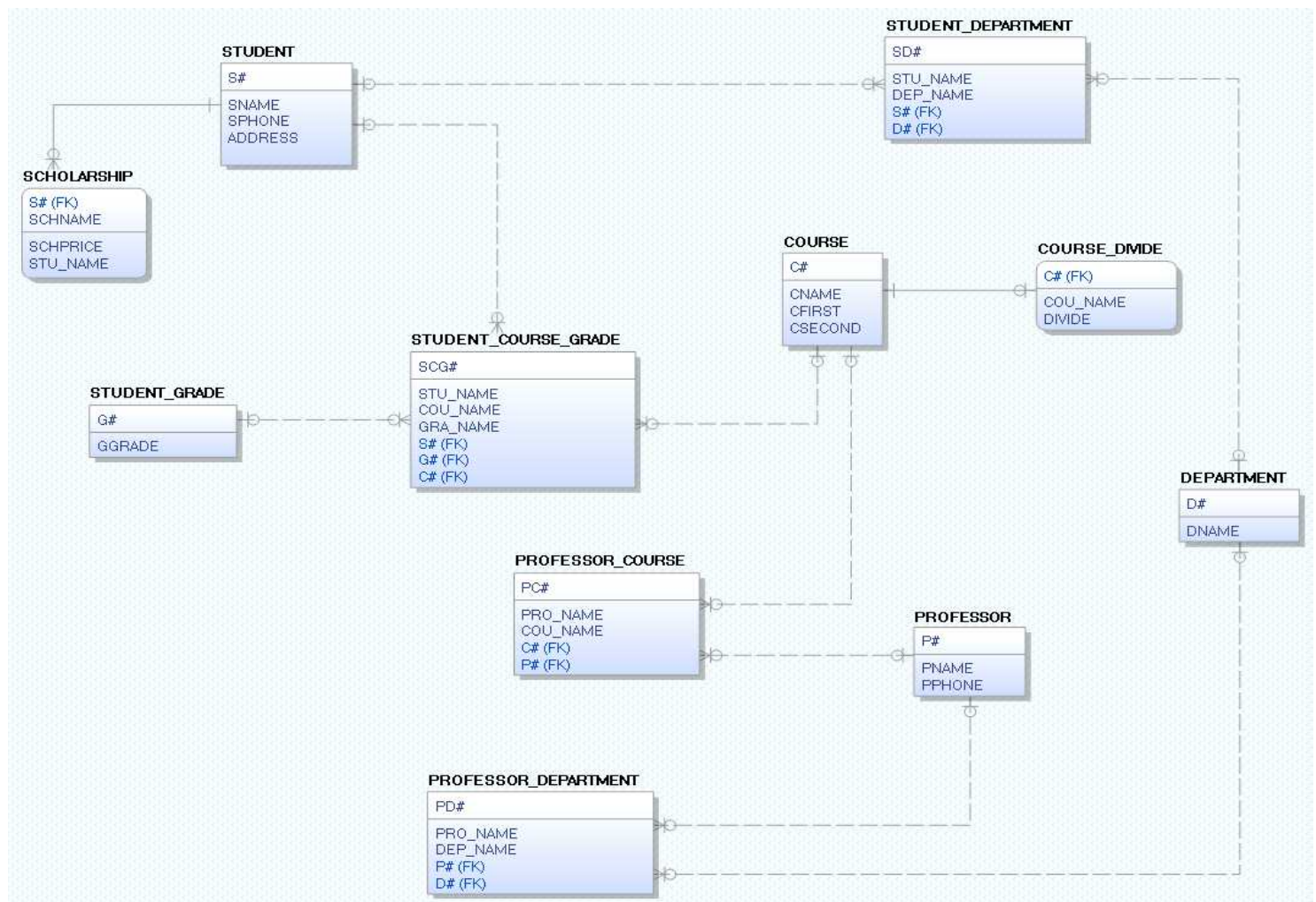


[CFEE : 유도속성으로, FIRST\_HALF와 SECOND\_HALF 값을 합쳐서 계산됩니다.]

[DMDE : 다치 속성으로, 학과의 분반(DB-A, DB-B, ...)의 값을 가집니다.]



과제 2. 학생, 학과, 과정, 교수, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해 ERwin을 사용해서 E-R Diagram 설계하기



### 과제 3. 학생, 학과, 과정, 교수, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해

#### E-R Diagram 설계한 것을 Table로 변환하기

[Entity] 학생, 학과, 과정, 교수, 성적, 분반(CD, WEEK\_ENTITY), 장학금(SCH)

[RELATION] SD, PD, SCG, PC

```
CREATE TABLE STUDENT(
S# CHAR(5),
SNAME CHAR(10),
SPHONE CHAR(15),
SGU CHAR(10),
SDONG CHAR(10),
PRIMARY KEY(S#));
```

```
CREATE TABLE DEPARTMENT(
D# CHAR(5),
DNAME CHAR(20),
PRIMARY KEY(D#));
```

```
CREATE TABLE COURSE(
C# CHAR(5),
CNAME CHAR(10),
CFIRST NUMERIC(10),
CSECOND NUMERIC(10),
PRIMARY KEY(C#));
```

```
CREATE TABLE PROFESSOR(
P# CHAR(5),
PNAME CHAR(15),
PPHONE CHAR(15),
PRIMARY KEY(P#));
```

```
CREATE TABLE STUDENT_GRADE(
G# CHAR(5),
GGRADE CHAR(6),
PRIMARY KEY(G#));
```

```
CREATE TABLE SD(
SD# CHAR(5),
S# CHAR(5),
STU_NAME CHAR(10),
D# CHAR(5),
DEP_NAME CHAR(20),
PRIMARY KEY(SD#),
FOREIGN KEY(S#) REFERENCES STUDENT,
FOREIGN KEY(D#) REFERENCES DEPARTMENT);
```

```
CREATE TABLE PD(
PD# CHAR(5),
P# CHAR(5),
PRO_NAME CHAR(15),
D# CHAR(5),
DEP_NAME CHAR(20),
PRIMARY KEY(PD#),
FOREIGN KEY(P#) REFERENCES PROFESSOR,
FOREIGN KEY(D#) REFERENCES DEPARTMENT);
```

```
CREATE TABLE PC(
PC# CHAR(5),
P# CHAR(5),
PRO_NAME CHAR(15),
C# CHAR(5),
COU_NAME CHAR(10),
PRIMARY KEY(PC#),
FOREIGN KEY(P#) REFERENCES PROFESSOR,
FOREIGN KEY(C#) REFERENCES COURSE);
```

```
CREATE TABLE SCG(
SC# CHAR(5),
S# CHAR(5),
STU_NAME CHAR(10),
C# CHAR(5),
COU_NAME CHAR(10),
G# CHAR(5),
GRA_NAME CHAR(6),
PRIMARY KEY(SC#),
FOREIGN KEY(S#) REFERENCES STUDENT,
FOREIGN KEY(C#) REFERENCES COURSE,
FOREIGN KEY(G#) REFERENCES STUDENT_GRADE);
```

```
CREATE TABLE SCH (
SCHNAME CHAR(10),
SCHPRICE NUMERIC(10),
S# CHAR(5),
STU_NAME CHAR(10),
PRIMARY KEY(S#),
PRIMARY KEY(SCHNAME),
FOREIGN KEY(S#) REFERENCES STUDENT);
```

```
CREATE TABLE CD(
C# CHAR(5),
COU_NAME CHAR(10),
DIVIDE CHAR(6),
FOREIGN KEY(C#) REFERENCES COURSE);
```

### 과제 4. 학생, 학과, 과정, 교수, 성적 등의 정보로 구성되는 대학교 정보 데이터베이스에 대해

#### 각 Table마다 샘플 데이터를 입력하고 질의 수행하기

##### 4.1) A학생이 수강하는 강의의 이름과 성적은?

SQL> select stu\_name, cou\_name, gra\_name from scg where stu\_name = '구명회';

STU\_NAME    COU\_NAME    GRA\_NAME

```
-----
구명회      DATABASE    A+
구명회      IoT          B+
구명회      BIGDATA     A0
구명회      OS            A0
```

SQL> select stu\_name, cou\_name, gra\_name from scg where stu\_name = '김은수';

STU\_NAME    COU\_NAME    GRA\_NAME

```
-----
김은수      DATABASE    A0
김은수      BIGDATA     C+
김은수      OS            A+
```

SQL> select stu\_name, cou\_name, gra\_name from scg where stu\_name = '정초희';

STU\_NAME    COU\_NAME    GRA\_NAME

```
-----
정초희      IoT          B+
정초희      SOFTWARE    B0
```

#### 4.2) A 학생이 수강하는 강의의 이름과 학과는?

```
SQL> select scg.cou_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.stu_name = '구명회';
```

```
COU_NAME    DEP_NAME
```

```
-----
OS           컴퓨터공학과
BIGDATA      컴퓨터공학과
IoT          컴퓨터공학과
DATABASE     컴퓨터공학과
```

```
SQL> select scg.cou_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.stu_name = '공성현';
```

```
COU_NAME    DEP_NAME
```

```
-----
SOFTWARE     생명공학과
BIGDATA      생명공학과
IoT          생명공학과
```

```
SQL> select scg.cou_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.stu_name = '강윤정';
```

```
COU_NAME    DEP_NAME
```

```
-----
BIGDATA      광학공학과
IoT          광학공학과
```

#### 4.3) A 강의를 수강하는 학생의 이름과 학과는?

```
SQL> select scg.stu_name, sd.dep_name from scg join sd on scg.s#=sd.s# where scg.cou_name='DATABASE';
```

```
STU_NAME    DEP_NAME
```

```
-----
구명회      컴퓨터공학과
김은수      컴퓨터공학과
이현지      물리공학과
김영훈      토목공학과
```

```
SQL> select scg.stu_name, sd.dep_name from scg join sd on scg.s#=sd.s# where scg.cou_name='BIGDATA';
```

```
STU_NAME    DEP_NAME
```

```
-----
구명회      컴퓨터공학과
김은수      컴퓨터공학과
공성현      생명공학과
이현지      물리공학과
강윤정      광학공학과
김영훈      토목공학과
```

6 개의 행이 선택되었습니다.

```
SQL> select scg.stu_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.cou_name = 'IoT';
```

```
STU_NAME    DEP_NAME
```

```
-----
구명회      컴퓨터공학과
정초희      전자공학과
공성현      생명공학과
강윤정      광학공학과
```

#### 4.4) A 강사가 강의하는 과목의 관리 학과는?

```
SQL> select pc.pro_name, pc.cou_name, pd.dep_name
       from pc join pd on pc.p# = pd.p# where pc.pro_name = '박중혁';
```

PRO_NAME	COU_NAME	DEP_NAME
----------	----------	----------

박중혁	OS	컴퓨터공학과
박중혁	BIGDATA	컴퓨터공학과
박중혁	DATABASE	컴퓨터공학과

```
SQL> select pc.pro_name, pc.cou_name, pd.dep_name
       from pc join pd on pc.p# = pd.p# where pc.pro_name = '조정희';
```

PRO_NAME	COU_NAME	DEP_NAME
----------	----------	----------

조정희	SOFTWARE	컴퓨터공학과
-----	----------	--------

```
SQL> select pc.pro_name, pc.cou_name, pd.dep_name
       from pc join pd on pc.p# = pd.p# where pc.pro_name = '김도훈';
```

PRO_NAME	COU_NAME	DEP_NAME
----------	----------	----------

김도훈	IoT	전자공학과
-----	-----	-------

#### 4.5) 질문1 수강신청한 과목의 개수?

```
SQL> select count(cou_name) from scg where stu_name = '구명희';
```

COUNT(COU_NAME)
-----------------

4
---

```
SQL> select count(cou_name) from scg where stu_name = '정초희';
```

COUNT(COU_NAME)
-----------------

2
---

```
SQL> select count(cou_name) from scg where stu_name = '김영훈';
```

COUNT(COU_NAME)
-----------------

4
---

#### 4.6) 질문2 수강신청한 과목의 리스트?

```
SQL> select cou_name from scg where stu_name = '구명희';
```

COU_NAME
----------

DATABASE
----------

IoT
-----

BIGDATA
---------

OS
----

```
SQL> select cou_name from scg where stu_name = '정초희';
```

```
COU_NAME
```

```
-----
```

```
IoT
```

```
SOFTWARE
```

```
SQL> select cou_name from scg where stu_name = '김영훈';
```

```
COU_NAME
```

```
-----
```

```
DATABASE
```

```
IoT
```

```
BIGDATA
```

```
OS
```

## 과제 5. 소감

제가 과제를 열추 정답에 근접하게 만들었다는 가정을 하고.. 소감을 작성하겠습니다!

1) 관계 식별하는게 어려웠습니다.

데이터 모델링 7단계 중에서 4단계 식별자까지 잘 설계했는데 테이블을 생성하고 데이터 삽입중에 속성을 잘못 설정한 것도 있었고 설정해둔 관계가 다른 관계와 중복되는 부분이 있어서 삭제하고 수정하는 등, 관계를 설정하는 과정에서 시간이 많이 걸렸습니다. 그리고 저번 과제에서는 테이블을 생성하고 그 테이블 안에서만 데이터를 만지기만 했었는데, 이번 과제에서는 아주 간단한 질의지만, JOIN 질의를 수행하면서 데이터베이스를 실습하고 있다는 생각이 들었습니다.

2) 사실 어렵기도 했지만 이렇게 시간이 오래 걸릴 과제는 아니었는데 설계 과정에서 익숙하지 않아 많이 헤멘 것이 원인이었던 것 같습니다. 시험공부할 때 시간이 너무 없어서 데이터모델링 7단계만 외워서 갔는데 예상을 벗어나는 엄청난 문제가 나왔었는데 이걸 해보고 갔다면 쉽게 그릴수 있었을 것 같습니다. 물론 모두 제가 과제를 안한탓이지만, 과제가 일주일만 빨리 나왔어도 풀어봤을텐데 시험기간 한 주 전에 과제를 내주신것도 조금 아쉬웠습니다..

과제 기한을 시험 끝난주까지 내신 교수님들이 많아서 시간이 부족해 테이블 작성하는것부터는 TEST 테이블 만들어서 다른 명령어들도 수행해볼 기회가 없어 아쉬웠습니다. 과제들이 열추 마무리되면 실습해보겠습니다.

3) 문제점

COURSE\_DIVIDE를 분반을 표시하도록 약한 개체로 설계했습니다.

약한 개체로 설정하기 위해서는 ER-Diagram에서 부모와 자식간의 관계를 식별 관계로 표현해야 하는데,

Table Property에서 C#이 기본키로 설정되어 ERwin의 관계는 그대로 두고 SQL문에서 기본키인 C#을 제거했습니다.

The screenshot shows the Oracle Table 'COURSE\_DIVIDE' Editor. The top section displays table properties for COURSE, COURSE\_DIVIDE, DEPARTMENT, and PROFESSOR. The bottom section shows the SQL DDL for creating the COURSE\_DIVIDE table.

Physical Name	Owner	Physical Only	Generate
COURSE		<input type="checkbox"/>	<input checked="" type="checkbox"/>
COURSE_DIVIDE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DEPARTMENT		<input type="checkbox"/>	<input checked="" type="checkbox"/>
PROFESSOR		<input type="checkbox"/>	<input checked="" type="checkbox"/>

Table Options: General Supplemental Logging Validation Materialized View log SQL Style Icon

```
CREATE TABLE COURSE_DIVIDE(
  DIVIDE          CHAR(6) NULL ,
  COU_NAME        CHAR(10) NULL ,
  C#              CHAR(5) NOT NULL ,
  PRIMARY KEY (C#),
  FOREIGN KEY (C#) REFERENCES COURSE (C#)
);

COMMENT ON TABLE COURSE_DIVIDE IS '과목의 분반을 표현
"DATABASE"의 분반은 DB_A, DB_B, DB_C가 있다.';

COMMENT ON COLUMN COURSE_DIVIDE.C# IS 'WEEK ENTITY
COURSE 기본키 C#를 속성으로 가짐';
```



## 과제 0. 실습화면

### - STUDENT 테이블과 STUDENT\_GRADE 테이블의 데이터 확인

```
SQL>
SQL> SELECT * FROM STUDENT;
```

S#	SNAME	SPHONE	SGU	SDONG
S1	구명회	010-1234-5670	광산구	운남동
S2	김은수	010-1234-5671	광동구	서남동
S3	정조희	010-1234-5672	광동구	지산동
S4	홍성현	010-1234-5673	광산구	신가동
S5	이현지	010-1234-5674	서구	화정동
S6	강윤정	010-1234-5675	서구	동천동
S7	김영훈	010-1234-5676	서구	치평동

7 개의 행이 선택되었습니다.

```
SQL> SELECT * FROM STUDENT_GRADE;
```

G#	GGRADE
G1	A+
G2	A0
G3	B+
G4	B0
G5	C+
G6	C0
G7	D+
G8	D0
G9	F

9 개의 행이 선택되었습니다.

### - SCH 테이블과 DEPARTMENT 테이블의 데이터 확인

```
SQL> SELECT * FROM SCH;
```

SCHNAME	SCHPRICE	S#	STU_NAME
사랑	1000000	S2	김은수
용기	500000	S5	이현지
이해	300000	S3	정조희
관공	200000	S7	김영훈
의면	100000	S1	구명회

```
SQL> SELECT * FROM DEPARTMENT;
```

D#	DNAME
D1	컴퓨터공학과
D2	전자공학과
D3	전기공학과
D4	토목공학과
D5	생명공학과
D6	화학공학과
D7	물리공학과
D8	항공우주공학과
D9	광학공학과

9 개의 행이 선택되었습니다.

### - COURSE 테이블과 PROFESSOR 테이블의 데이터 확인

```
SQL> SELECT * FROM COURSE;
```

C#	CNAME	CFEE
C1	DATABASE	200000
C2	IoT	100000
C3	BIGDATA	150000
C4	SOFTWARE	100000
C5	OS	200000

```
SQL> SELECT * FROM PROFESSOR;
```

P#	PNAME	PPHONE
P1	박중혁	010-2345-6780
P2	신재빈	010-2345-6781
P3	조정희	010-2345-6782
P4	김도훈	010-2345-6783
P5	오찬양	010-2345-6785

- SD 테이블과 PD 테이블, PC 테이블의 데이터 확인

SQL> SELECT \* FROM SD;

SD#	S#	STU_NAME	D#	DEP_NAME
SD1	S1	구명회	D1	컴퓨터공학과
SD2	S2	김영수	D1	컴퓨터공학과
SD3	S3	김정호	D2	전자공학과
SD4	S4	김성현	D5	생명과학과
SD5	S5	이현지	D7	화학공학과
SD6	S6	강윤정	D9	한글학과
SD7	S7	김영훈	D4	컴퓨터공학과

7 개의 행이 선택되었습니다.

SQL> SELECT \* FROM PD;

PD#	P#	PRO_NAME	D#	DEP_NAME
PD1	P1	박종현	D1	컴퓨터공학과
PD2	P2	신재빈	D3	전기공학과
PD3	P3	조정희	D1	컴퓨터공학과
PD4	P4	김도훈	D2	전자공학과
PD5	P5	오찬양	D1	컴퓨터공학과

SQL> SELECT \* FROM PC;

PC#	P#	PRO_NAME	C#	COU_NAME
PC1	P1	박종현	C1	DATABASE
PC2	P1	박종현	C3	BIGDATA
PC3	P1	박종현	C5	OS
PC4	P2	신재빈	C2	IoT
PC5	P2	신재빈	C3	BIGDATA
PC6	P3	조정희	C4	SOFTWARE
PC7	P4	김도훈	C2	IoT
PC8	P5	오찬양	C3	BIGDATA
PC9	P5	오찬양	C4	SOFTWARE

9 개의 행이 선택되었습니다.

- CD 테이블과 SCG 테이블의 데이터 확인

SQL>  
SQL> SELECT \* FROM CD;

C#	COU_NAME	DIVIDE
C1	DATABASE	DB_A
C1	DATABASE	DB_B
C1	DATABASE	DB_C
C2	IoT	IoT_A
C2	IoT	IoT_B
C3	BIGDATA	BD_A
C3	BIGDATA	BD_B
C3	BIGDATA	BD_C
C4	SOFTWARE	SW_A
C5	OS	OS_A
C5	OS	OS_B
C5	OS	OS_C

12 개의 행이 선택되었습니다.

SQL> SELECT \* FROM SCG;

SC#	S#	STU_NAME	C#	COU_NAME	G#	GRA_NA
SC1	S1	구명회	C1	DATABASE	G1	A+
SC2	S1	구명회	C2	IoT	G3	B+
SC3	S1	구명회	C3	BIGDATA	G2	A0
SC4	S1	구명회	C5	OS	G2	A0
SC5	S2	김영수	C1	DATABASE	G2	A0
SC6	S2	김영수	C3	BIGDATA	G5	C+
SC7	S2	김영수	C5	OS	G1	A+
SC8	S3	김정호	C2	IoT	G3	B+
SC9	S3	김정호	C4	SOFTWARE	G4	B0
SC10	S4	김성현	C2	IoT	G2	A0
SC11	S4	김성현	C3	BIGDATA	G6	C0
SC12	S4	김성현	C4	SOFTWARE	G3	B+
SC13	S5	이현지	C1	DATABASE	G2	A0
SC14	S5	이현지	C3	BIGDATA	G1	A+
SC15	S5	이현지	C4	SOFTWARE	G2	A0
SC16	S6	강윤정	C2	IoT	G1	A+
SC17	S6	강윤정	C3	BIGDATA	G3	B+
SC18	S7	김영훈	C1	DATABASE	G9	F
SC19	S7	김영훈	C2	IoT	G9	F
SC20	S7	김영훈	C3	BIGDATA	G9	F
SC21	S7	김영훈	C5	OS	G9	F

21 개의 행이 선택되었습니다.

#### 실습 4.1. A학생이 수강하는 강의의 이름과 성적은?

```
SQL> select stu_name, cou_name, gra_name from scg where stu_name = '구명회';
```

STU_NAME	COU_NAME	GRA_NAME
구명회	DATABASE	A+
구명회	IoT	B+
구명회	BIGDATA	A0
구명회	OS	A0

```
SQL> select stu_name, cou_name, gra_name from scg where stu_name = '김은수';
```

STU_NAME	COU_NAME	GRA_NAME
김은수	DATABASE	A0
김은수	BIGDATA	C+
김은수	OS	A+

```
SQL> select stu_name, cou_name, gra_name from scg where stu_name = '정초희';
```

STU_NAME	COU_NAME	GRA_NAME
정초희	IoT	B+
정초희	SOFTWARE	B0

#### 실습 4.2. A 학생이 수강하는 강의의 이름과 학과는?

```
SQL>
```

```
SQL> select scg.cou_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.stu_name = '구명회';
```

COU_NAME	DEP_NAME
OS	컴퓨터공학과
BIGDATA	컴퓨터공학과
IoT	컴퓨터공학과
DATABASE	컴퓨터공학과

```
SQL> select scg.cou_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.stu_name = '공성현';
```

COU_NAME	DEP_NAME
SOFTWARE	생명공학과
BIGDATA	생명공학과
IoT	생명공학과

```
SQL> select scg.cou_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.stu_name = '강윤정';
```

COU_NAME	DEP_NAME
BIGDATA	광학공학과
IoT	광학공학과

#### 4.3. A 강의를 수강하는 학생의 이름과 학과는?

```
SQL>
SQL> select scg.stu_name, sd.dep_name from scg join sd on scg.s#=sd.s# where scg.cou_name='DATABASE';
```

STU_NAME	DEP_NAME
구명희	컴퓨터공학과
김영수	컴퓨터공학과
이현지	메이킹학과
김영훈	토목공학

```
SQL> select scg.stu_name, sd.dep_name from scg join sd on scg.s#=sd.s# where scg.cou_name='BIGDATA';
```

STU_NAME	DEP_NAME
구명희	컴퓨터공학과
김영수	컴퓨터공학과
이현지	생명공학
이현정	메이킹학과
김영훈	광학
김영훈	토목공학

6 개의 행이 선택되었습니다.

```
SQL> select scg.stu_name, sd.dep_name from scg join sd on scg.s# = sd.s# where scg.cou_name = 'IoT';
```

STU_NAME	DEP_NAME
구명희	컴퓨터공학과
조정희	전자공학과
김영수	생명공학
김영훈	광학
김영훈	토목공학

#### 4.4. A 강사가 강의하는 과목의 관리학과는?

```
SQL> select pc.pro_name, pc.cou_name, pd.dep_name
2         from pc join pd on pc.p# = pd.p# where pc.pro_name = '박종혁';
```

PRO_NAME	COU_NAME	DEP_NAME
박종혁	OS	컴퓨터공학과
박종혁	BIGDATA	컴퓨터공학과
박종혁	DATABASE	컴퓨터공학과

```
SQL> select pc.pro_name, pc.cou_name, pd.dep_name
2         from pc join pd on pc.p# = pd.p# where pc.pro_name = '조정희';
```

PRO_NAME	COU_NAME	DEP_NAME
조정희	SOFTWARE	컴퓨터공학과

```
SQL> select pc.pro_name, pc.cou_name, pd.dep_name
2         from pc join pd on pc.p# = pd.p# where pc.pro_name = '김도훈';
```

PRO_NAME	COU_NAME	DEP_NAME
김도훈	IoT	전자공학과

#### 4.5. 질문1 수강신청한 과목의 개수?

```
SQL>
SQL> select count(cou_name) from scg where stu_name = '구명회';

COUNT(COU_NAME)
-----
                4

SQL> select count(cou_name) from scg where stu_name = '정초희';

COUNT(COU_NAME)
-----
                2

SQL> select count(cou_name) from scg where stu_name = '김영훈';

COUNT(COU_NAME)
-----
                4
```

#### 4.6. 질문2 수강신청한 과목의 리스트?

```
SQL>
SQL> select cou_name from scg where stu_name = '구명회';

COU_NAME
-----
DATABASE
IoT
BIGDATA
OS

SQL> select cou_name from scg where stu_name = '정초희';

COU_NAME
-----
IoT
SOFTWARE

SQL> select cou_name from scg where stu_name = '김영훈';

COU_NAME
-----
DATABASE
IoT
BIGDATA
OS
```

**과제 0. code** 과제 3 ~ 4의 코드를 주석 추가해서 순서대로 첨부했습니다.

//

[Create Table]

```
SQL> CREATE TABLE STUDENT(           // STUDENT라는 테이블을 만들었습니다.
  2  S# CHAR(5),                       // 학생의 고유번호를 나타낼 CHAR형 S#
  3  SNAME CHAR(10),                   // 학생의 이름을 나타낼 CHAR형 SNAME
  4  SPHONE CHAR(15),                  // 학생의 전화번호를 나타낼 CHAR형 SPHONE
  5  SGU CHAR(10),                     // 학생의 주소(구)를 나타낼 CHAR형 SGU
  6  SDONG CHAR(10),                   // 학생의 주소(동)를 나타낼 CHAR형 SDONG
  7  PRIMARY KEY(S#));                 // 학생의 고유번호를 기본키로 설정
```

테이블이 생성되었습니다.

```
SQL> CREATE TABLE SCH(               // SCH라는 테이블을 만들었습니다.
  2  SCHNAME CHAR(10),                 // 장학금의 이름을 나타낼 CHAR형 SCHNAME
  3  SCHPRICE NUMERIC(10),             // 장학금의 금액을 나타낼 NUMERIC형 SCHPRICE
  4  S# CHAR(5),                       // 학생의 고유번호를 나타낼 CHAR형 S#
  5  STU_NAME CHAR(10),                // 학생의 이름을 나타낼 CHAR형 STU_NAME
  6  PRIMARY KEY(S#,SCHNAME),          // 학생의 고유번호와 장학금의 이름을 기본키로 설정
  7  FOREIGN KEY(S#) REFERENCES STUDENT); // 학생의 고유번호를 외래키로 설정
```

테이블이 생성되었습니다.

```
SQL> CREATE TABLE DEPARTMENT(       // DEPARTMENT라는 테이블을 만들었습니다.
  2  D# CHAR(5),                       // 학과의 고유번호를 나타낼 CHAR형 D#
  3  DNAME CHAR(20),                   // 학과의 이름을 나타낼 CHAR형 DNAME
  4  PRIMARY KEY(D#));                 // 학과의 고유번호를 기본키로 설정
```

테이블이 생성되었습니다.

```
SQL> CREATE TABLE COURSE(           // COURSE라는 테이블을 만들었습니다.
  2  C# CHAR(5),                       // 과정(강의)의 고유번호를 나타낼 CHAR형 C#
  3  CNAME CHAR(10),                   // 과정(강의)의 이름을 나타낼 CHAR형 CNAME
  4  CFIRST NUMERIC(10),               // 과정(강의)의 상반기 수강료를 나타낼 NUMERIC형 CFIRST
  5  CSECOND NUMERIC(10),              // 과정(강의)의 하반기 수강료를 나타낼 NUMERIC형 CSECOND
  6  PRIMARY KEY(C#));                 // 과정(강의)의 고유번호를 기본키로 설정
```

테이블이 생성되었습니다.

```
SQL> CREATE TABLE PROFESSOR(        // PROFESSOR라는 테이블을 만들었습니다.
  2  P# CHAR(5),                       // 교수의 고유번호를 나타낼 CHAR형 P#
  3  PNAME CHAR(15),                   // 교수의 이름을 나타낼 CHAR형 PNAME
  4  PPHONE CHAR(15),                  // 교수의 전화번호를 나타낼 CHAR형 PPHONE
  5  PRIMARY KEY(P#));                 // 교수의 고유번호를 기본키로 설정
```

테이블이 생성되었습니다.

```
SQL> CREATE TABLE STUDENT_GRADE(    // STUDENT_GRADE라는 테이블을 만들었습니다.
  2  G# CHAR(5),                       // 학생 성적의 고유번호를 나타낼 CHAR형 G#
  3  GGRADE CHAR(6),                   // 학생 성적의 범위를 나타낼 CHAR형 GGRADE
  4  PRIMARY KEY(G#));                 // 학생 성적의 고유번호를 기본키로 설정
```

테이블이 생성되었습니다.

////////////////////////////////////

```
SQL> CREATE TABLE SD(           // SD라는 STUDENT와 DEPARTMENT의 관계테이블을 만들었습니다.
  2  SD# CHAR(5),                // CHAR형 SD#
  3  S# CHAR(5),
  4  STU_NAME CHAR(10),          // 학생의 이름을 나타낼 CHAR형 STU_NAME
  5  D# CHAR(5),
  6  DEP_NAME CHAR(20),          // 학과의 이름을 나타낼 CHAR형 DEP_NAME
  7  PRIMARY KEY(SD#),
  8  FOREIGN KEY(S#) REFERENCES STUDENT,      // STUDENT의 기본키 S#을 외래키로 설정
  9  FOREIGN KEY(D#) REFERENCES DEPARTMENT);  // DEPARTMENT의 기본키 D#을 외래키로 설정
테이블이 생성되었습니다.
```

```
SQL> CREATE TABLE PC(           // PC라는 PROFESSOR와 COURSE의 관계 테이블을 만들었습니다.
  2  PC# CHAR(5),
  3  P# CHAR(5),
  4  PRO_NAME CHAR(15),         // 교수의 이름을 나타낼 CHAR형 PRO_NAME
  5  C# CHAR(5),
  6  COU_NAME CHAR(10),         // 과정(강의)의 이름을 나타낼 CHAR형 COU_NAME
  7  PRIMARY KEY(PC#),
  8  FOREIGN KEY(P#) REFERENCES PROFESSOR,         // PROFESSOR의 기본키 P#을 외래키로 설정
  9  FOREIGN KEY(C#) REFERENCES COURSE);           // COURSE의 기본키 C#을 외래키로 설정

테이블이 생성되었습니다.
```

```
SQL> CREATE TABLE SCG(      // SCG라는 STUDENT와 COURSE와 GRADE의 관계 테이블을 만들었습니다.
  2 SC# CHAR(5),
  3 S# CHAR(5),
  4 STU_NAME CHAR(10),      // 학생의 이름을 나타낼 CHAR형 STU_NAME
  5 C# CHAR(5),
  6 COU_NAME CHAR(10),      // 과정(강의)의 이름을 나타낼 CHAR형 COU_NAME
  7 G# CHAR(5),
  8 GRA_NAME CHAR(6),      // 학생의 성적을 나타낼 GRA_NAME
  9 PRIMARY KEY(SC#),
10 FOREIGN KEY(S#) REFERENCES STUDENT,      // 학생의 기본키 S#을 외래키로 설정
11 FOREIGN KEY(C#) REFERENCES COURSE,      // 과정(강의)의 기본키 C#을 외래키로 설정
12 FOREIGN KEY(G#) REFERENCES STUDENT_GRADE);      // 학생 성적의 기본키 G#을 외래키로 설정
테이블이 생성되었습니다.
```

////////////////////////////////////

[Insert Data, Select Data]

※ 데이터의 삽입과 검색에 대해서는 중복되는 SQL문이 많으므로 하나하나 설명하지 않고  
 맨 위에 Insert into~와 Select from~을 어떻게 사용하는지만 적었습니다.  
 그리고 삽입하는 SQL문이 많아서 처음과 끝부분만 남기고 제거했습니다.  
 처음과 끝부분이 포함된 코드를 txt파일에 정리해서 제출하겠습니다.

```
// Insert into [Table]      // data를 삽입하고싶은 Table
// Values([column]);      // column 형식에 맞는 값들
SQL> INSERT INTO STUDENT VALUES('S1', '구명회', '010-1234-5670', '광산구', '운남동');
1 개의 행이 만들어졌습니다.
...
SQL> INSERT INTO STUDENT VALUES('S7', '김영훈', '010-1234-5676', '서구', '치평동');
1 개의 행이 만들어졌습니다.
```

```
SQL> SELECT * FROM STUDENT;
S#      SNAME      SPHONE      SGU      SDONG
-----
S1      구명회      010-1234-5670      광산구      운남동
S2      김은수      010-1234-5671      동구      서남동
S3      정초희      010-1234-5672      동구      지산동
S4      공성현      010-1234-5673      광산구      신가동
S5      이현지      010-1234-5674      서구      화정동
S6      강윤정      010-1234-5675      서구      동천동
S7      김영훈      010-1234-5676      서구      치평동
7 개의 행이 선택되었습니다.
```

```
SQL> INSERT INTO DEPARTMENT VALUES('D1', '컴퓨터공학과');
1 개의 행이 만들어졌습니다.
...
SQL> INSERT INTO DEPARTMENT VALUES('D9', '광학공학과');
1 개의 행이 만들어졌습니다.
```



SQL> SELECT \* FROM DEPARTMENT;

D#	DNAME
----	-------

D1	컴퓨터공학과
D2	전자공학과
D3	전기공학과
D4	토목공학과
D5	생명공학과
D6	화학공학과
D7	물리공학과
D8	항공우주공학과
D9	광학공학과

9 개의 행이 선택되었습니다.

SQL> INSERT INTO COURSE VALUES('C1', 'DATABASE', 100000, 100000);

1 개의 행이 만들어졌습니다.

...

SQL> INSERT INTO COURSE VALUES('C5', 'OS', 80000, 80000);

1 개의 행이 만들어졌습니다.

SQL> SELECT \* FROM COURSE;

C#	CNAME	CFIRST	CSECOND
C1	DATABASE	100000	100000
C2	IoT	40000	60000
C3	BIGDATA	50000	80000
C4	SOFTWARE	35000	60000
C5	OS	80000	80000

SQL> INSERT INTO PROFESSOR VALUES('P1', '박종혁', '010-2345-6780');

1 개의 행이 만들어졌습니다.

...

SQL> INSERT INTO PROFESSOR VALUES('P5', '오찬양', '010-2345-6785');

1 개의 행이 만들어졌습니다.

SQL> SELECT \* FROM PROFESSOR;

P#	PNAME	PPHONE
P1	박종혁	010-2345-6780
P2	신재빈	010-2345-6781
P3	조정희	010-2345-6782
P4	김도훈	010-2345-6783
P5	오찬양	010-2345-6785

SQL> INSERT INTO STUDENT\_GRADE VALUES('G1', 'A+');

1 개의 행이 만들어졌습니다.

...

SQL> INSERT INTO STUDENT\_GRADE VALUES('G9', 'F');

1 개의 행이 만들어졌습니다.

SQL> SELECT \* FROM STUDENT\_GRADE;

G#	GGRADE
----	--------

G1	A+
----	----

G2	A0
----	----

G3	B+
----	----

G4	B0
----	----

G5	C+
----	----

G6	C0
----	----

G7	D+
----	----

G8	D0
----	----

G9	F
----	---

9 개의 행이 선택되었습니다.

SQL> INSERT INTO SD VALUES('SD1', 'S1', '구명희', 'D1', '컴퓨터공학과');

1 개의 행이 만들어졌습니다.

...

SQL> INSERT INTO SD VALUES('SD7', 'S7', '김영훈', 'D4', '토목공학과');

1 개의 행이 만들어졌습니다.

SQL> SELECT \* FROM SD;

SD#	S#	STU_NAME	D#	DEP_NAME
-----	----	----------	----	----------

SD1	S1	구명희	D1	컴퓨터공학과
-----	----	-----	----	--------

SD2	S2	김은수	D1	컴퓨터공학과
-----	----	-----	----	--------

SD3	S3	정초희	D2	전자공학과
-----	----	-----	----	-------

SD4	S4	공성현	D5	생명공학과
-----	----	-----	----	-------

SD5	S5	이현지	D7	물리공학과
-----	----	-----	----	-------

SD6	S6	강윤정	D9	광학공학과
-----	----	-----	----	-------

SD7	S7	김영훈	D4	토목공학과
-----	----	-----	----	-------

7 개의 행이 선택되었습니다.

SQL> INSERT INTO PD VALUES('PD1', 'P1', '박종혁', 'D1', '컴퓨터공학과');

1 개의 행이 만들어졌습니다.

...

SQL> INSERT INTO PD VALUES('PD5', 'P5', '오찬양', 'D1', '컴퓨터공학과');

1 개의 행이 만들어졌습니다.

SQL> SELECT \* FROM PD;

PD#	P#	PRO_NAME	D#	DEP_NAME
-----	----	----------	----	----------

PD1	P1	박종혁	D1	컴퓨터공학과
-----	----	-----	----	--------

PD2	P2	신재빈	D3	전기공학과
-----	----	-----	----	-------

PD3	P3	조정희	D1	컴퓨터공학과
-----	----	-----	----	--------

PD4	P4	김도훈	D2	전자공학과
-----	----	-----	----	-------

PD5	P5	오찬양	D1	컴퓨터공학과
-----	----	-----	----	--------

```
SQL> INSERT INTO PC VALUES('PC1', 'P1', '박중혁', 'C1', 'DATABASE');
```

1 개의 행이 만들어졌습니다.

...

```
SQL> INSERT INTO PC VALUES('PC9', 'P5', '오찬양', 'C4', 'SOFTWARE');
```

1 개의 행이 만들어졌습니다.

```
SQL> SELECT * FROM PC;
```

PC#	P#	PRO_NAME	C#	COU_NAME
PC1	P1	박중혁	C1	DATABASE
PC2	P1	박중혁	C3	BIGDATA
PC3	P1	박중혁	C5	OS
PC4	P2	신재빈	C2	IoT
PC5	P2	신재빈	C3	BIGDATA
PC6	P3	조정희	C4	SOFTWARE
PC7	P4	김도훈	C2	IoT
PC8	P5	오찬양	C3	BIGDATA
PC9	P5	오찬양	C4	SOFTWARE

9 개의 행이 선택되었습니다.

```
SQL> INSERT INTO CD VALUES('C1', 'DATABASE', 'DB_A');
```

1 개의 행이 만들어졌습니다.

...

```
SQL> INSERT INTO CD VALUES('C5', 'OS', 'OS_C');
```

1 개의 행이 만들어졌습니다.

```
SQL> SELECT * FROM CD;
```

C#	COU_NAME	DIVIDE
C1	DATABASE	DB_A
C1	DATABASE	DB_B
C1	DATABASE	DB_C
C2	IoT	IoT_A
C2	IoT	IoT_B
C3	BIGDATA	BD_A
C3	BIGDATA	BD_B
C3	BIGDATA	BD_C
C4	SOFTWARE	SW_A
C5	OS	OS_A
C5	OS	OS_B
C5	OS	OS_C

12 개의 행이 선택되었습니다.

```
SQL> INSERT INTO SCG VALUES('SC1', 'S1', '구멍회', 'C1', 'DATABASE', 'G1', 'A+');
```

1 개의 행이 만들어졌습니다.

...

```
SQL> INSERT INTO SCG VALUES('SC21', 'S7', '김영훈', 'C5', 'OS', 'G9', 'F');
```

1 개의 행이 만들어졌습니다.

SQL> SELECT \* FROM SCG;

SC#	S#	STU_NAME	C#	COU_NAME	G#	GRA_NA
-----	----	----------	----	----------	----	--------

SC1	S1	구명회	C1	DATABASE	G1	A+
SC2	S1	구명회	C2	IoT	G3	B+
SC3	S1	구명회	C3	BIGDATA	G2	A0
SC4	S1	구명회	C5	OS	G2	A0
SC5	S2	김은수	C1	DATABASE	G2	A0
SC6	S2	김은수	C3	BIGDATA	G5	C+
SC7	S2	김은수	C5	OS	G1	A+
SC8	S3	정초희	C2	IoT	G3	B+
SC9	S3	정초희	C4	SOFTWARE	G4	B0
SC10	S4	공성현	C2	IoT	G2	A0
SC11	S4	공성현	C3	BIGDATA	G6	C0
SC12	S4	공성현	C4	SOFTWARE	G3	B+
SC13	S5	이현지	C1	DATABASE	G2	A0
SC14	S5	이현지	C3	BIGDATA	G1	A+
SC15	S5	이현지	C4	SOFTWARE	G2	A0
SC16	S6	강윤정	C2	IoT	G1	A+
SC17	S6	강윤정	C3	BIGDATA	G3	B+
SC18	S7	김영훈	C1	DATABASE	G9	F
SC19	S7	김영훈	C2	IoT	G9	F
SC20	S7	김영훈	C3	BIGDATA	G9	F
SC21	S7	김영훈	C5	OS	G9	F

21 개의 행이 선택되었습니다.