

# DB프로그래밍

4주차


담당교수: 김희숙  
(jasmin11@hanmail.net)

# 시작하기

4주차 1-01

담당교수: 김희숙  
(jasmin11@hanmail.net)

# Erwin과 SQL문법 자동생성

- 
- 1) Database / Database Connection
  - 2) Tools / Forward Engineer / Schema generation  
Options / Column / **Physical Order** 체크해제(필드순서 바꾸기 때문)  
Options / **Trigger** 체크 모두 해제
  - 3) <Preview> - 스크립트 확인 후 Close → <Generate> → <OK>

# [실습] (Erwin+Oracle)

1. 다음을 Oracle 로 작성한다 (book 테이블)

1) SQL 문법 사용하여 테이블 작성과 데이터 입력 : [메모장] 에서 작성한다

/\* Oracle 실습 (book-ora.sql) \*/

drop table book;

--book 테이블 생성

CREATE TABLE book (

bookid int NOT NULL,

bookname varchar(50),

publisher varchar(30),

price int,

PRIMARY KEY (bookid)

);

--book(bookid, bookname, publisher, price)

insert into book values(1, '축구의 역사', '굿스포츠', 7000);

insert into book values(2, '축구 아는 여자', '나무수', 13000);

insert into book values(3, '축구의 이해', '대한미디어', 22000);

insert into book values(4, '골프 바이블', '대한미디어', 35000);

insert into book values(5, '피겨 교본', '굿스포츠', 8000);

--데이터 조회

select \* from book;

--형식 명령어

(한 줄에 레코드가 나타나지 않을 때 활용)

SCOTT>COL bookid FOR 999999

SCOTT>COL bookname FOR a20

SCOTT>COL publisher FOR a10

SCOTT>COL price FOR 9999999

SCOTT>

SCOTT>

SCOTT>select \* from book;

BOOKID	BOOKNAME	PUBLISHER	PRICE
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

book

bookid

bookname

publisher

price

book

bookid: int

bookname: varchar(50)

publisher: varchar(30)

price: int

2) 메모장에 작성한 스크립트 파일을 c 드라이브의 **DBDATA**란 폴더 만들고 그 아래 저장하고 파일실행

SQL>@ c:/DBDATA/book-ora.sql;



# [실습] (Oracle)

```
--(oracle) 에서 직접 입력하여 실습(emp_dept-ora.sql)

drop table employee;
drop table department;

CREATE TABLE department (
    deptno          CHAR(2) NOT NULL,
    deptname        VARCHAR(18) ,
    PRIMARY KEY (deptno)
);

CREATE TABLE employee (
    empno           CHAR(4) NOT NULL,
    ename           VARCHAR(12) ,
    deptno          CHAR(2) NOT NULL,
    tel             CHAR(4) ,
    salary          int ,
    PRIMARY KEY (empno) ,
    FOREIGN KEY (deptno) REFERENCES department (deptno)
);

--department(deptno, deptname)

insert into department values('AA', '총무부');
insert into department values('BB', '영업부');
insert into department values('CC', '기획부');
insert into department values('DD', '생산부');

--employee(empno, ename, deptno, tel, salary)

insert into employee values('100', '김기훈', 'BB', '1241', 200);
insert into employee values('101', '홍성범', 'CC', '5621', 200);
insert into employee values('102', '이만수', 'BB', '5251', 250);
insert into employee values('103', '강나미', 'DD', '1231', 300);

COL deptno FOR a6

COL empno FOR a6

select * from department;
select * from employee;
```



# 기본키/외래키



EMPLOYEE

empno	ename	dept	tel	salary
100	김기훈	영업	1241	200
101	홍성범	기획	5621	200
102	이만수	영업	5251	250
103	강나미	생산	1231	300



EMPLOYEE

empno	ename	dept	tel	salary
100	김기훈	BB	1241	200
101	홍성범	CC	5621	200
102	이만수	BB	5251	250
103	강나미	DD	1231	300

DEPARTMENT

deptno	deptname
AA	총무
BB	영업
CC	기획
DD	생산

[생능오세종, p96~97]

일기내용

날짜	일기메모	날씨
2011-10-01	과제물이 너무 많은	맑음
2011-10-02	하루만 참아보자	맑음
2011-10-03	오늘은 복권에 당첨	흐림

일정정보

날짜	시간	일정내용
2011-10-01	11	영어 스터디
2011-10-01	12	창수 점심약속
2011-10-03	9	과제 제출
2011-10-03	15	은행 방문
2011-10-03	19	가족모임(신촌)



# SQL 문법(테이블 생성)

```
SCOTT>--mybook(bookid, price)
SCOTT>--mycustomer(cno, cname)
SCOTT>--myorders(orderid, cno, bookid, price)
```

```
SCOTT>select * from mybook;
```

BOOKID	PRICE
B1	1000
B2	2000
B3	3000

```
SCOTT>select * from mycustomer;
```

CNO	CNAME
A	박지성
B	김연아
C	장미란

```
SCOTT>select * from myorders;
```

ORDERID	CNO	BOOKID	PRICE
1	A	B1	1000
2	A	B2	2000
3	B	B1	1000

```
/* (mybook-sub-ora.sql) */
```

```
drop table myorders;
```

```
drop table mybook;
```

```
drop table mycustomer;
```

--mybook, mycustomer, myorders 테이블 생성

```
CREATE TABLE mybook (
```

```
    bookid    varchar(6) PRIMARY KEY,
```

```
    price     INT
```

```
);
```

```
CREATE TABLE mycustomer (
```

```
    cno        varchar(3) PRIMARY KEY,
```

```
    cname      VARCHAR(10)
```

```
);
```

```
CREATE TABLE myorders (
```

```
    orderid    INT PRIMARY KEY,
```

```
    cno        varchar(3) REFERENCES mycustomer(cno),
```

```
    bookid     varchar(6) REFERENCES mybook(bookid),
```

```
    price      INT
```

```
);
```



# SQL 문법(테이블 생성)

```
SCOTT>--mybook(bookid, price)
SCOTT>--mycustomer(cno, cname)
SCOTT>--myorders(orderid, cno, bookid, price)
```

```
SCOTT>select * from mybook;
```

BOOKID	PRICE
B1	1000
B2	2000
B3	3000

```
SCOTT>select * from mycustomer;
```

CNO	CNAME
A	박지성
B	김연아
C	장미란

```
SCOTT>select * from myorders;
```

ORDERID	CNO	BOOKID	PRICE
1	A	B1	1000
2	A	B2	2000
3	B	B1	1000

--데이터 입력

```
INSERT INTO mybook VALUES('B1', 1000);
```

```
INSERT INTO mybook VALUES('B2', 2000);
```

```
INSERT INTO mybook VALUES('B3', 3000);
```

```
INSERT INTO mycustomer VALUES ('A', '박지성');
```

```
INSERT INTO mycustomer VALUES ('B', '김연아');
```

```
INSERT INTO mycustomer VALUES ('C', '장미란');
```

```
INSERT INTO myorders VALUES (1, 'A','B1', 1000);
```

```
INSERT INTO myorders VALUES (2, 'A','B2', 2000);
```

```
INSERT INTO myorders VALUES (3, 'B','B1',1000);
```

```
select * from mybook;
```

```
select * from mycustomer;
```

```
select * from myorders;
```





# 데이터 모델링

번호	이름	연락처	지역	응모분야
1	서민석	345-1212	서울	전자제품
2	최지혜	241-2423	대구	자전거
3	이지민	267-2313	서울	자전거
4	박병준	412-2331	부산	전자제품
5	박진서	342-5621	대구	주방용품
...	...	...	...	...



지역

지역번호	지역
1	서울
2	대구
3	부산
...	...

응모분야

응모분야번호	응모분야명
1	전자제품
2	자전거
3	주방용품
...	...

응모자

번호	이름	연락처	지역	응모분야
1	서민석	345-1212	1	1
2	최지혜	241-2423	2	2
3	이지민	267-2313	1	2
4	박병준	412-2331	3	1
5	박진서	342-5621	2	3
...	...	...	...	...

## SQL문법

```
/* (freegift01-ora.sql) */
```

```
-- 응모(번호,이름,연락처,지역,응모분야)
```

번호	이름	연락처	지역	응모분야
1	서민석	345-1212	서울	전자제품
2	최지혜	241-2423	대구	자전거
3	이지민	267-2313	서울	자전거
4	박병준	412-2331	부산	전자제품
5	박진서	342-5621	대구	주방용품
...	...	...	...	...

## SQL문법

```
/* (freegift02-ora.sql) */
```

```
-- 지역(지역번호, 지역)
```

```
-- 응모분야(응모분야번호, 응모분야명)
```

```
-- 응모자(번호, 이름, 연락처, 지역, 응모분야)
```

지역

지역번호	지역
1	서울
2	대구
3	부산
...	...

응모분야

응모분야번호	응모분야명
1	전자제품
2	자전거
3	주방용품
...	...

응모자

번호	이름	연락처	지역	응모분야
1	서민석	345-1212	1	1
2	최지혜	241-2423	2	2
3	이지민	267-2313	1	2
4	박병준	412-2331	3	1
5	박진서	342-5621	2	3
...	...	...	...	...

# 관계 종류

4주차 1-02

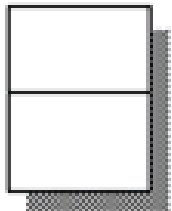
담당교수: 김희숙  
(jasmin11@hanmail.net)

# 기본 개념

- 스키마
    - 개체(Entity)
    - 속성(Attribute)
    - 관계(Relationship)
    - 제약조건
- 데이터베이스를 구성하는 데이터구조와 제약조건에 대한 명세를 구체적으로 기술한 것

-ERwin 사용하여 논리적 설계

E/1



- 엔티티 명
- 기본키 영역
- 일반 속성(Attribute) 영역

## ❖관계

### 1. 관계 차수

1진 관계, 2진 관계, 3진 관계

### 2. 관계 카디널리티

일대일 관계(1:1)

일대다 관계(1:N)

다대다 관계(M:N)

### 3. 관계 존재성

필수(Mandatory)

선택(Optional)

# 관계의 종류

## ❖관계

### 1. 관계 차수

1진 관계, 2진 관계, 3진 관계

### 2. 관계 카디널리티

일대일 관계(1:1)

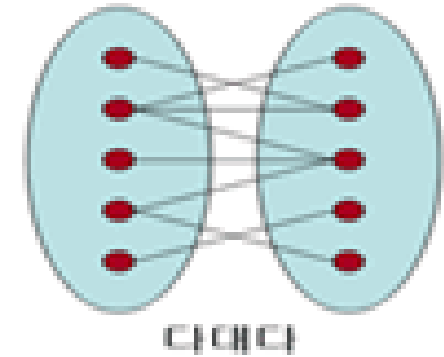
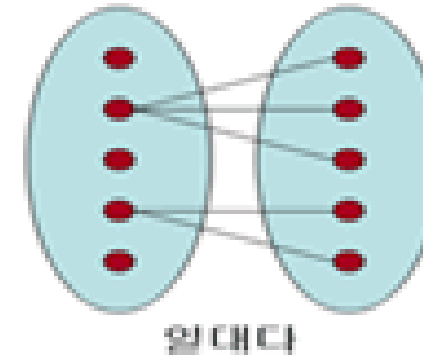
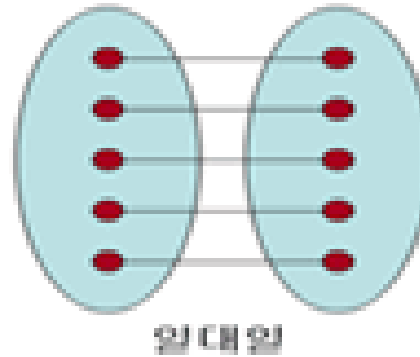
일대다 관계(1:N)

다대다 관계(M:N)

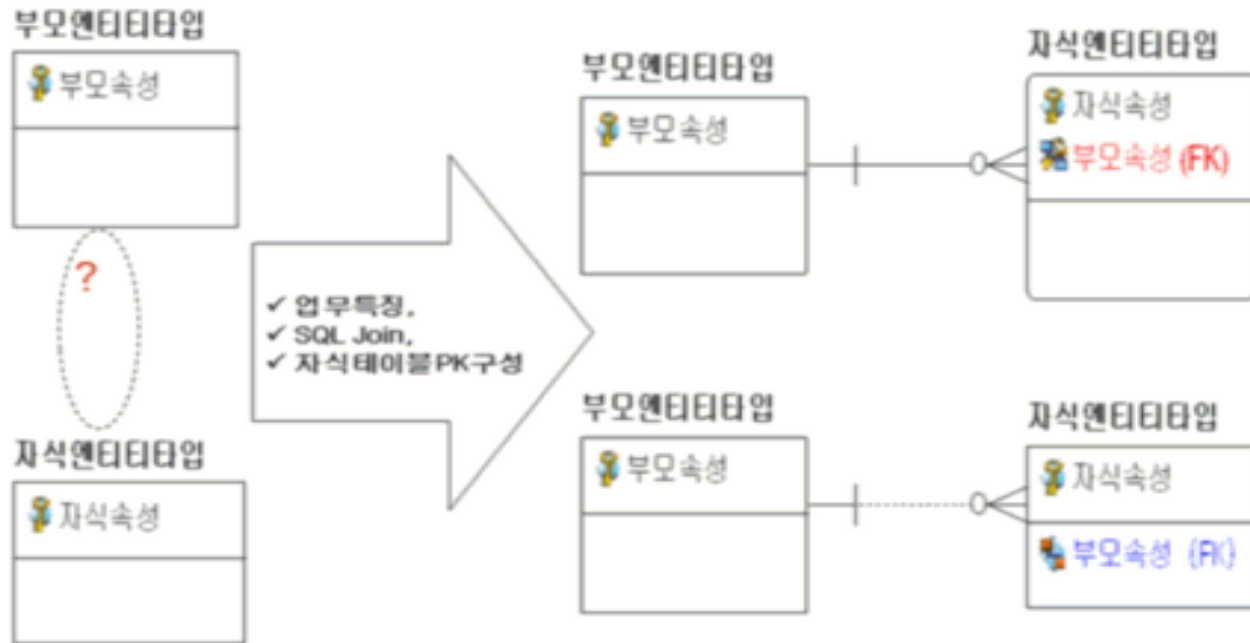
### 3. 관계 존재성

필수(Mandatory)

선택(Optional)



# 식별 관계/비식별 관계



## ❖ 관계(Erwin)

### 1. 식별관계 (identifying)

부모 테이블의 기본키가 자식 테이블의 기본키 구성원으로 전이

### 2. 비식별관계 (non-identifying)

부모 테이블의 기본키가 자식 테이블의 일반속성으로 전이

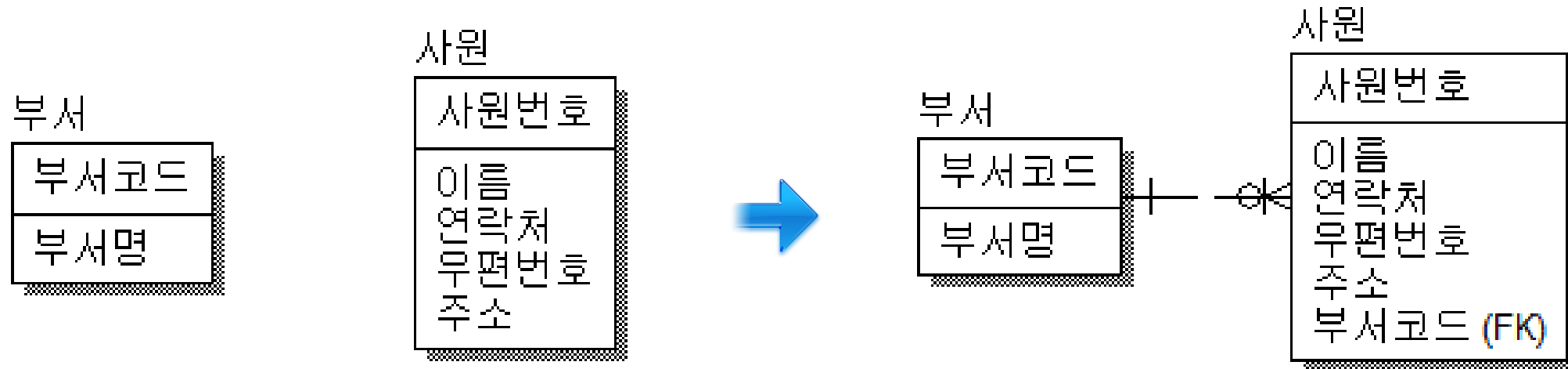
# 일대다 관계

4주차 2-01

담당교수: 김희숙  
(jasmin11@hanmail.net)



# 일대다 관계(비식별 관계)



# 일대다 관계(비식별 관계)

환자

환자번호	이름	질병코드	나이
P1001	김철수	A01	30
P1002	양길현	A03	29
P1003	임영수	A01	50
Q1001	박한나	A04	40

질병

질병코드	질병명	증상
A01	뇌졸중	머지럼증
A02	콜레라	설사
A03	기관지염	발열
A04	장티푸스	발열

<조건>

단, **각 환자는 반드시 하나의 질병만 있는 것으로 간주한다**

질병에 걸린 여러 명의 환자가 있다

환자가 없는 질병이 있을 수도 있다

# 일대다 관계(식별 관계)

학생

학번
학번 이름 주소 전공

학생취미

취미



학생

학번
학번 이름 주소 전공

학생취미

학번 (FK) 취미



학생

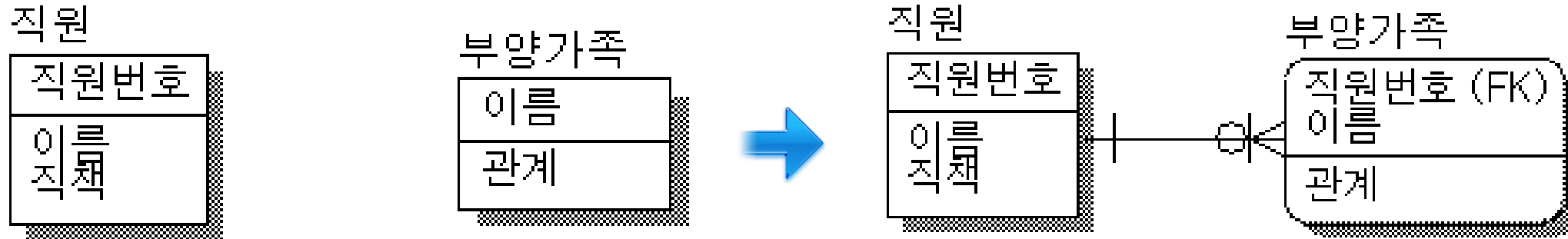
학번	이름	주소	전공
21001	김철수	서울	영문학
21002	양길현	인천	컴퓨터
21003	임영수	광주	화학
21004	박한나	부산	수학

학생취미

학번	취미
21002	낚시
21002	등산
21003	낚시
21004	여행



# 일대다 관계(식별 관계)



# 일대다 관계(식별 관계)

student

학번	이름	전공
20140001	홍길동	소프트웨어정보
20140002	박지성	소프트웨어정보

student\_course

학번	수강과목	학점
20140001	데이터베이스	A0
20140001	알고리즘	B+
20140002	데이터베이스	B0

# 다대다 관계

4주차 2-02

담당교수: 김희숙  
(jasmin11@hanmail.net)



# 다대다 관계

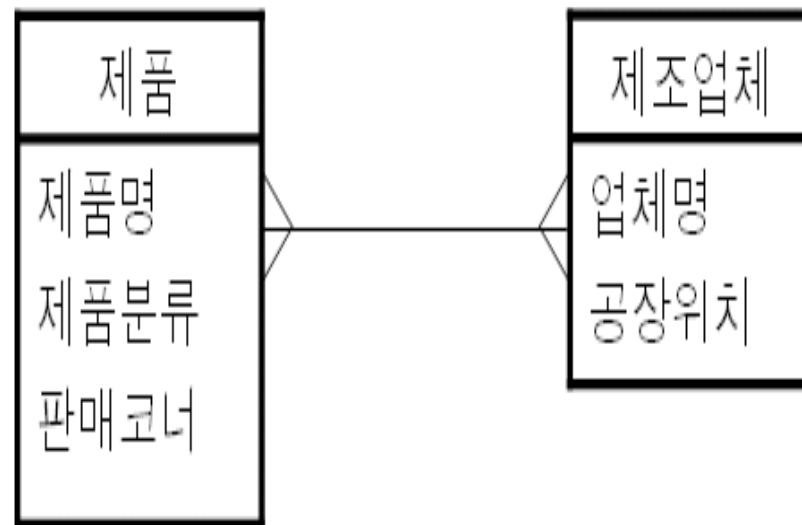


제품

제품명	제품분류	판매코너
TV	가전	1-C
냉장고	가전	1-D
MP3	미디어	2-A
세탁기	가전	1-F

제조업체

업체명	공장위치
삼성	수원
LG	강릉
대우	부산
현대	광주



# 다대다 관계

학생

학번	이름
11002	이홍근
24036	김순미
30419	박상웅

수강

학번	과목번호	학점
11002	CS310	A0
11002	CS313	B+
24036	CS345	B0
30419	CS310	A+

과목

과목번호	과목이름
CS310	데이터베이스
CS313	운영체제
CS345	자료구조
CS326	자바



# 다대다 관계(다대다 교차해소)

학생

학번
이름 주소 연락처

과목

과목코드
과목명 과목개요

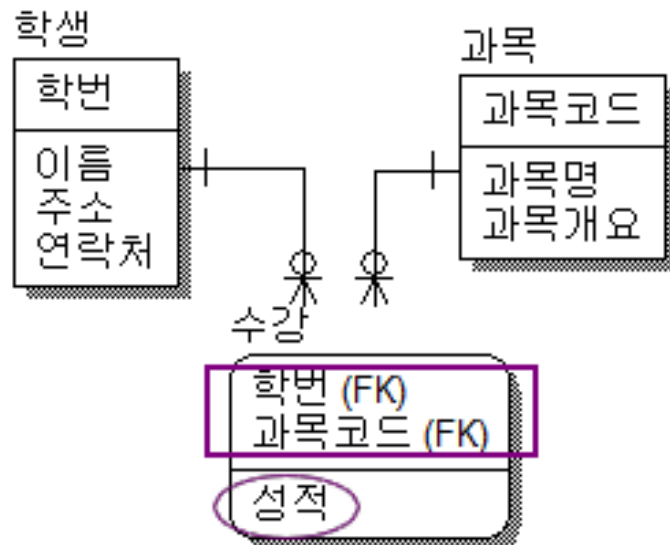
학생

학번
이름 주소 연락처

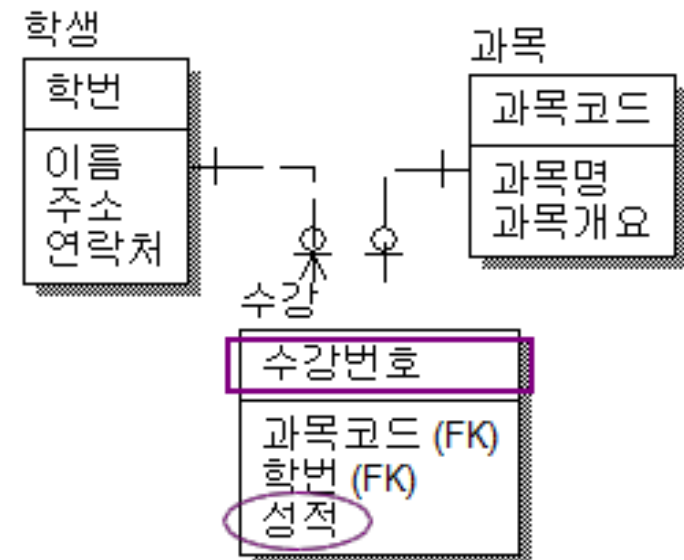
과목

과목코드
과목명 과목개요

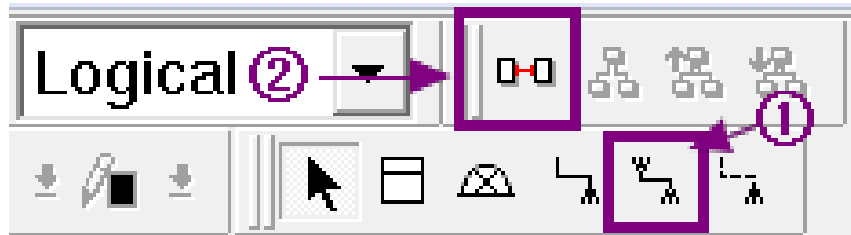
## A. 식별관계 (복합키)



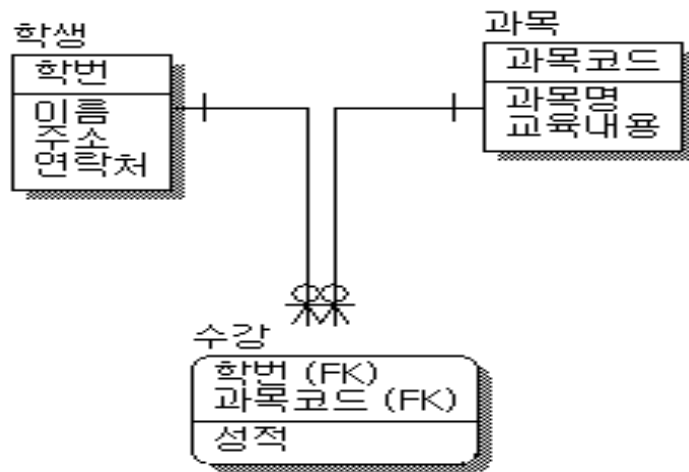
## B. 비식별관계 (대리키)



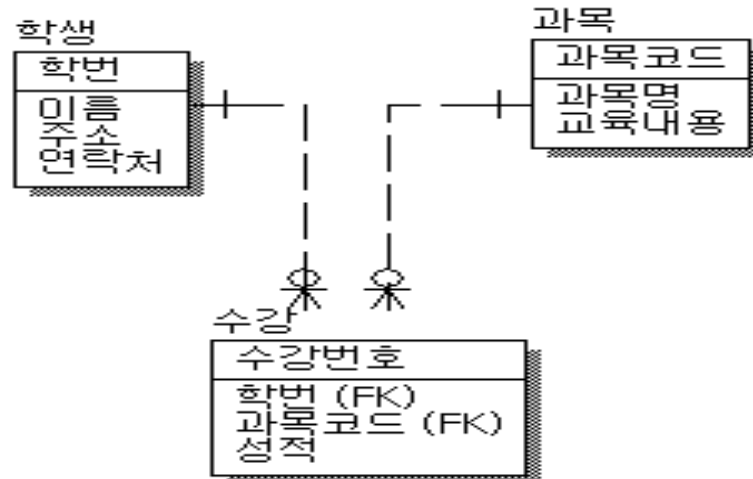
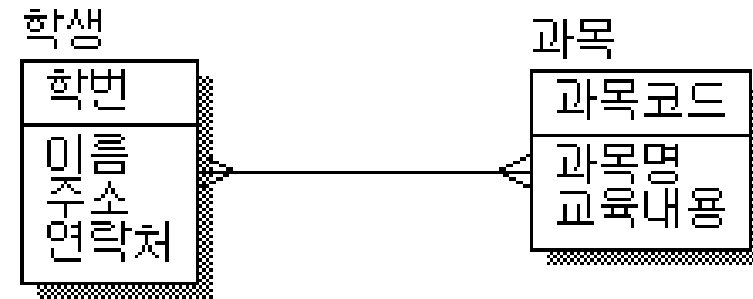
# [실습] (Erwin) 다대다 관계(학생/수강/과목)



(다대다 교차해소)



<식별관계>



<비식별관계>

# 실습 영상

## (다대다 관계: 학생, 과목)





# 일대일 관계

4주차 3-01

담당교수: 김희숙  
(jasmin11@hanmail.net)

# 일대일 관계

<테이블 A>

사원번호	이름	부서명
1	김하나	영업1과
2	이두한	영업2과
3	박태성	영업3과

<테이블 B>

사원번호	출신학교(최종)	성적
1	한국대학교	B <sup>o</sup>
2	신한고등학교	90
3	제일대학교	B <sup>+</sup>

사원

사원번호
이름 부서명

부가정보

출신학교 성적
------------

사원

사원번호
이름 부서명

부가정보

사원번호 (FK)
출신학교 성적



# 일대일 관계



학생

학번	이름	주소	전공
21001	김철수	서울	영문학
21002	양길현	인천	컴퓨터
21003	임영수	광주	화학
21004	박한나	부산	수학

신체정보

학번	키	몸무게	혈액형
21001	175	70	A
21002	169	65	B
21003	180	60	O
21004	170	85	B

사원

사번
성명

신체사항

키
몸무게

사원

사번
성명

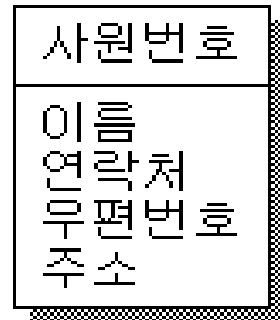
신체사항

사번 (FK)
키
몸무게

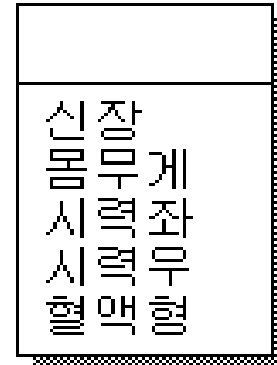


# 일대일 관계(Erwin)

사원



신체정보



Relationship Cardinality

Summary: One-to-Exactly-1

Cardinality

☐ Zero, One or More

☐ One or More (P)

☐ Zero or One (Z)

☒ Exactly: 1

Relationship Type

☒ Identifying

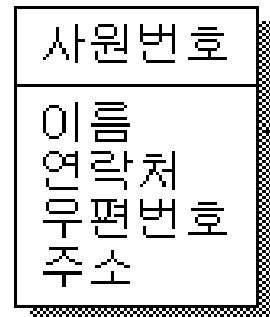
☐ Non-Identifying

Nulls

☐ Nulls Allowed

☐ No Nulls

사원



신체정보

