

데이터베이스관리

3주차

담당교수: 김희숙
(jasmin11@hanmail.net)

복습

3주차 3-01

담당교수: 김희숙
(jasmin11@hanmail.net)

데이터베이스 관리 시스템(DBMS)



- **데이터베이스:**

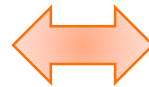
- 한 조직의 여러 응용시스템들이 데이터를 **공유**할 수 있도록 **통합**, **저장**된 **운영**데이터의 집합체

- * **DBMS:**

- 1) 데이터베이스를 관리하는 시스템 소프트웨어
- 2) 사용자 또는 응용프로그램과 데이터베이스 간의 인터페이스 역할을 하는 논리적인 프로그램

- * **DBMS 장점:**

- 1) 중복 최소화
- 2) 데이터 **독립성**



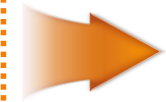
- * **파일시스템 단점:**

- 1) 데이터 **중복성**
- 2) 데이터 **종속성**



데이터베이스 관리 시스템(DBMS)

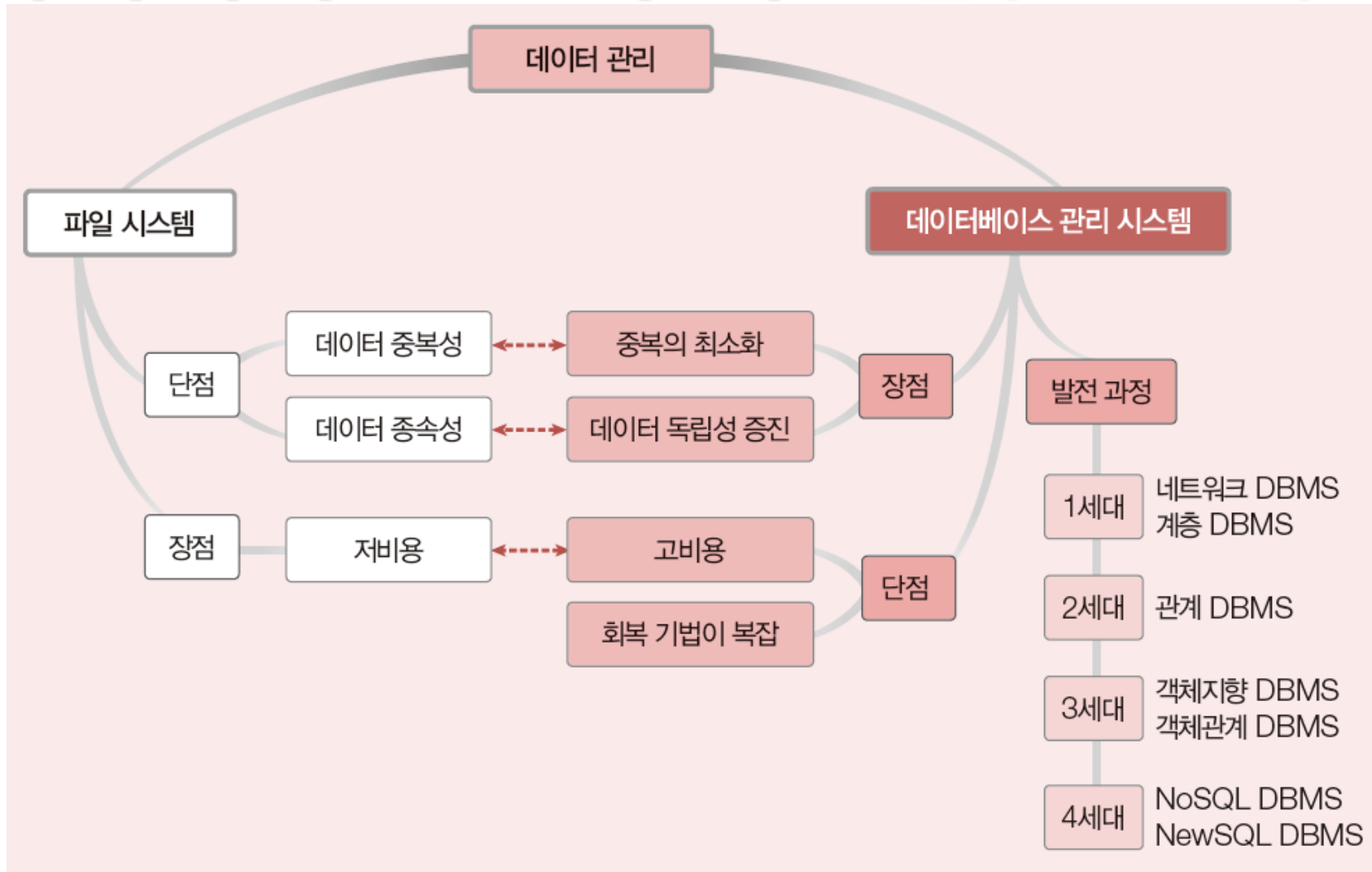
DBMS 기능



정의 기능	데이터베이스 구조를 정의하거나 수정할 수 있다.
조작 기능	데이터를 삽입·삭제·수정·검색하는 연산을 할 수 있다.
제어 기능	데이터를 항상 정확하고 안전하게 유지할 수 있다.

- 정의 :
 - 조직에 필요한 데이터를 저장하기 적합한 **데이터베이스 구조를 정의**하거나, 이미 정의된 **구조를 수정**할 수 있다
- 조작 :
 - 데이터베이스에 **저장된 데이터에 접근**하여 사용할 수 있는 기능을 제공
 - 사용자 요구에 따라 **데이터를 삽입, 삭제, 수정, 검색하는 연산**을 효율적으로 처리
- 제어 :
 - 데이터를 여러 사용자가 공유해도 항상 정확하고 안전하게 유지하는 기능을 제공
 - 데이터를 삽입, 삭제, 수정하는 연산을 한 후에도 내용이 일관되면서 무결성을 유지
 - 장애가 발생해도 회복이 가능하도록 제어
 - 권한이 있는 사용자에게만 데이터 접근을 허용하여 보안이 유지되도록 제어
 - 여러 사용자가 데이터베이스에 동시에 접근하여 데이터를 처리할 수 있도록 제어

데이터베이스 관리 시스템(DBMS)



1. 다음이 설명하는 파일 시스템의 문제점은 무엇인가?

응용 프로그램이 파일에 직접 접근하여 데이터를 처리해야 하므로 파일의 데이터 구성 방법이나 물리적인 저장 구조에 맞게 응용 프로그램을 작성해야 하고, 파일의 구조가 변경되면 응용 프로그램도 함께 변경해야 한다.

- | | |
|-----------|-----------|
| ① 데이터 중복성 | ② 데이터 변경성 |
| ③ 데이터 독립성 | ④ 데이터 종속성 |

2. 데이터베이스 관리 시스템의 주요 기능이 아닌 것은?

- ① 정의 기능
 - ② 조작 기능
 - ③ 제어 기능
 - ④ 절차 기능

3. 다음은 데이터베이스 관리 시스템의 주요 기능 중 무엇에 대한 설명인가?

사용자 요구에 따라 데이터베이스에 저장된 데이터에 접근하여 삽입·삭제·수정·검색 연산을 정확하고 효율적으로 수행한다.

- ① 정의 기능
 - ② 조작 기능
 - ③ 제어 기능
 - ④ 연산 기능

4. 데이터베이스 관리 시스템의 주요 기능인 제어 기능에 대한 설명으로 옳지 않은 것은?

- ① 연산을 수행한 후에도 데이터의 일관성과 무결성을 유지한다.
- ② 여러 사용자가 데이터베이스에 동시에 접근하여 데이터를 처리할 수 있도록 제어한다.
- ③ 데이터베이스의 구조를 정의하거나 수정한다.
- ④ 정당한 사용자가 허가된 데이터에만 접근할 수 있도록 보안을 유지한다.

5. 데이터베이스 관리 시스템의 장점으로 보기 어려운 것은?

- ① 데이터 중복을 통제할 수 있다.
- ② 데이터 보안이 향상된다.
- ③ 백업과 회복 방법이 간단하다.
- ④ 데이터 무결성을 유지할 수 있다.

6. 데이터베이스 관리 시스템의 단점으로 가장 적합한 것은?

- ① 표준화가 어렵다.
- ② 응용 프로그램의 개발 비용이 많이 든다.
- ③ 데이터 독립성을 확보하기 어렵다.
- ④ 백업과 회복 기법이 복잡하다.

7. 다음과 같은 특성이 있는 데이터베이스 관리 시스템의 유형은?

데이터베이스를 노드와 간선을 이용한 그래프 형태로 구성하는 데이터 모델을 사용한다.
데이터베이스의 구조가 복잡하고 변경이 어렵다는 단점이 있다.

- ① 네트워크 데이터베이스 관리 시스템
- ② 계층 데이터베이스 관리 시스템
- ③ 관계 데이터베이스 관리 시스템
- ④ 객체지향 데이터베이스 관리 시스템

8. 다음은 어떤 유형의 데이터베이스 관리 시스템으로 분류할 수 있는가?

오라클, MS SQL 서버, 액세스, 인포믹스, MySQL

- ① 네트워크 데이터베이스 관리 시스템
- ② 계층 데이터베이스 관리 시스템
- ③ 관계 데이터베이스 관리 시스템
- ④ 객체지향 데이터베이스 관리 시스템

교재 2장 데이터베이스 관리 시스템

기출문제(p54~56) 풀어 보세요

👉 문제풀이: 네이버카페 강의 게시판 작성
문제번호, 문제, 답안 댓글로 작성하세요

[Quiz02]

[Quiz02] MySQL에서 실습하시오

아이디	비밀번호	이름	연락처	주소	적립금
apple	1234	정소화	02-111-1111	서울시 마포구	1000
banana	9876	김선우	02-222-2222	경기도 부천시	500

테이블 이름	열 이름	데이터 형식	NULL 유무	기본키
	아이디			
	비밀번호			
	이름			
	연락처			
	주소			
	적립금			

관계 데이터 모델

3주차 3-02

담당교수: 김희숙
(jasmin11@hanmail.net)



관계형 데이터 모델

□ 관계형 데이터베이스

- ✓ 연관된 데이터가 하나의 테이블에 모두 들어 있는 것이 아니라,
- ✓ 여러 개의 서로 다른 테이블로 관리하고 데이터 사이에 어떠한 **관계**를 부여함으로써 데이터를 조직화

	부서코드	부서명
▶	AA	총무부
	BB	영업부
	CC	기획부

	사원번호	이름	부서코드
▶	1111	홍길동	AA
	2222	임꺽정	AA
	3333	박찬호	BB
	4444	선동열	BB
	5555	차두리	AA
	6666	신동엽	BB



관계형 데이터 모델

□ 관계형 데이터베이스

테이블 1개로 설계

사원

사원번호	이름	부서코드
1111	홍길동	총무부
2222	임꺽정	총무부
3333	박찬호	영업부
4444	선동열	영업부
5555	차두리	총무부
6666	신동엽	영업부

영업지원부?

테이블 2개로 설계

	부서코드	부서명
▶	AA	총무부
	BB	영업부
	CC	기획부

	사원번호	이름	부서코드
▶	1111	홍길동	AA
	2222	임꺽정	AA
	3333	박찬호	BB
	4444	선동열	BB
	5555	차두리	AA
	6666	신동엽	BB



[요약] 테이블(릴레이션)

-테이블 (릴레이션)

학생 테이블 <출력 형태>

데이터베이스

테이블
릴레이션

필드 (속성, 열, 애트리뷰트) column

학생

번호	이름	학년	분반	학과번호
1	하지혜	1	YB	1
2	이정우	1	YA	1
3	오지영	2	J1	2
4	강재미	1	YB	1
5	박철호	2	J1	2

레코드
(튜플, 행)

row

테이블(table) : 릴레이션(Relation)
필드(field) : 속성, 열, 애트리뷰트
column
레코드(record) : 튜플(tuple), 행
row

학과 테이블 <출력 형태>

학과

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	정보통신과

학생(번호, 이름, 학년, 분반, 학과번호)

학과(학과번호, 학과명)

스키마(schema)



□스키마

- ✓데이터베이스를 구성하는 데이터의 구조와 제약조건에 대한 명세를 구체적으로 기술한 것

스키마 DEPARTMENT (DEPTNO, DEPTNAME, FLOOR) EMPLOYEE (EMPNO, EMPNAME, TITLE, DNO, SALARY)	데이터베이스스키마 (schema)	데이터베이스상태 (state)
	내포(intension)	외연(extension)
	정적	동적

데이터베이스상태

DEPARTMENT	DEPTNO	DEPTNAME	FLOOR
	1	영업	8
	2	기획	10
	3	개발	9

EMPLOYEE	EMPNO	EMPNAME	TITLE	DNO	SALARY
	2106	김창섭	대리	2	2000000
	3426	박영권	과장	3	2500000
	3011	이수민	부장	1	3000000
	1003	조민희	대리	1	2000000

스키마 표기법



스키마 표기법

DEPARTMENT	DEPTNO	DEPTNAME	FLOOR
	1	영업	8
	2	기획	10
	3	개발	9

DEPARTMENT(deptno, deptname, floor)

EMPLOYEE	EMPNO	EMPNAME	TITLE	DNO	SALARY
	2106	김창섭	대리	2	2000000
	3426	박영권	과장	3	2500000
	3011	이수민	부장	1	3000000
	1003	조민희	대리	1	2000000
	3427	최종철	사원	3	1500000

EMPLOYEE(empno, empname, title, **dno**, salary)
fk

(그림 출처: "데이터베이스 배움터 3판", 홍의경 저, 생능출판사, 2012)



스키마 vs. 인스턴스

* 릴레이션 스키마와 인스턴스

- 데이터베이스 스키마(Schema) : 데이터베이스의 구조와 제약조건에 대하여 기술한 것
- 데이터베이스 상태(State) : 특정시점의 데이터베이스 내용



용어



- **릴레이션(Relation)** : 행과 열로 구성된 2차원의 테이블
- **튜플(tuple)** : 특정 인스턴스에 관한 사실(값)들의 모임
- **도메인(Domain)** : 한 애트리뷰트에 나타날 수 있는 값들의 집합

- **차수(Degree)** : 한 릴레이션에 나타날 수 있는 애트리뷰트의 수
- **카디날리티(Cardinality)** : 튜플 수

[Quiz] 신입생

1. 신입생 테이블의 스키마 표기법을 적으시오
2. 차수(degree) 는?
3. 카디널리티(cardinality) 는?
4. 기본키(Primary key) 는?

학번	주민등록번호	이름	주소	학과명
1292001	900424-1825409	김광식	서울	컴퓨터공학과
1292002	900305-1730021	김정현	서울	컴퓨터공학과
1292003	891021-2308302	김현정	대전	컴퓨터공학과
1292301	890902-2704012	김현정	대구	산업공학과

관계 데이터 모델 개념

- 관계 데이터 모델 기본 개념

- 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델
- 하나의 개체에 대한 데이터를 하나의 릴레이션에 저장

열(속성, 애트리뷰트)

고객아이디	고객이름	나이	등급	직업	적립금
CHAR(20)	CHAR(20)	INT	CHAR(10)	CHAR(10)	INT
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

← 도메인

행(튜플)

그림 5-1 릴레이션의 예 : 고객 릴레이션

관계 데이터 모델 개념

- 관계 데이터 모델 기본 용어

열(속성, 애트리뷰트)

고객아이디	고객이름	나이	등급	직업	적립금
CHAR(20)	CHAR(20)	INT	CHAR(10)	CHAR(10)	INT
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

도메인

행(튜플)

그림 5-1 릴레이션의 예 : 고객 릴레이션

<고객 릴레이션의 차수는 ? 카디널리티는 ?

관계 데이터 모델 개념

- 릴레이션의 특성

- 튜플의 유일성

- 하나의 릴레이션에는 동일한 튜플이 존재할 수 없다.

- 튜플의 무순서

- 하나의 릴레이션에서 튜플 사이의 순서는 무의미하다.

- 속성의 무순서

- 하나의 릴레이션에서 속성 사이의 순서는 무의미하다.

- 속성의 원자성

- 속성 값으로 원자 값만 사용할 수 있다.

관계 데이터 모델 개념

- 릴레이션의 특성

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	회사원, 학생	0

그림 5-4 다중 값 속성을 포함하는 릴레이션의 예 : 고객 릴레이션

키(Key)



□기본키(Primary Key)

- ✓ 각 튜플을 유일하게 구별할 수 있는 하나 이상의 속성의 집합
- ✓ 중복불가
- ✓ 필수입력

테이블
릴레이션

학생

필드 (속성, 열, 애트리뷰트) column

번호	이름	학년	분반	학과번호
1	한지혜	1	YB	1
2	이정우	1	YA	1
3	오지영	2	J1	2
4	강재미	1	YB	1
5	박철호	2	J1	2

레코드
(튜플, 행)
row

키(Key)



□외래키(Foreign Key)

- ✓한 릴레이션의 기본키를 참조하는 키
- ✓참조되는 릴레이션의 기본키 값에 반드시 존재해야 한다

(외래키는 널 값 입력허용)

- ✓참조무결성 제약조건

고객 테이블

고객 ID	이름	주민등록번호	전화번호
1	홍길동	111111-1111111	011-234-5678
2	원도우	222222-2222222	012-546-8526
3	김한빛	333333-3333333	014-654-8752

주문 테이블

물품 ID	물품 이름	고객 ID
1	A	1
2	B	1
3	C	2



키(Key): 기본키

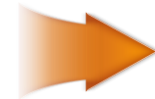
[동일한 이름 입력]

직원

성명	직급	급여	부서명
김사랑	사원	300	인사부
한예슬	대리	250	영업부
오세오	과장	500	영업부
구병서	부장	600	인사부

데이터입력

오세오	사원	300	경리부
-----	----	-----	-----



[기본키]

직원

사번	성명	직급	급여	부서명
111	김사랑	사원	300	인사부
222	한예슬	대리	250	영업부
333	오세오	과장	500	영업부
444	구병서	부장	600	인사부

데이터입력

555	오세오	사원	300	경리부
-----	-----	----	-----	-----

키(Key): 기본키

[복합키]

판매 실적

도시	지점명	판매액
서울	강남지점	330
서울	강북지점	330
광주	광주지점	190
서울	강서지점	150
서울	강동지점	180
대전	대전지점	240



키(Key): 외래키

[중복 데이터 발생]

거래처직원

사번	성명	직급	급여	부서명	위치
111	김사랑	사원	300	인사부	수원
222	한예슬	대리	250	영업부	용인
333	오세오	과장	500	영업부	용인
444	구병서	부장	600	인사부	수원

[중복 데이터 분리] 기본키, 외래키

거래처직원

사번	성명	직급	급여	부서
111	김사랑	사원	300	10
222	한예슬	대리	250	20
333	오세오	과장	500	20
444	구병서	부장	600	10

거래처부서

부서번호	부서명	위치
10	인사부	수원
20	영업부	용인
30	총무부	서울

[Quiz] 키(Key)

기본키:

인사기록
가족관계
거래처

인사기록

사원번호	성명	근무부서	직책	생년월일	주소	전화번호
A-1001	홍길동	기획전략실	대리	1979-08-04	서울시 종로구	010-0000-0000
A-1002	정명선	기획전략실	사원	1984-05-13	서울시 동대문구	010-1111-1111
A-1003	황성인	기획전략실	사원	1986-10-26	서울시 금천구	010-2222-2222
A-1004	방진석	기획전략실	과장	1973-02-09	서울시 성북구	010-3333-3333

가족관계

사원번호	이름	관계	직업	생년월일
A-1001	홍재민	부	회사원	1949-04-14
A-1001	소유진	모	가사	1954-09-21
A-1001	홍길순	여동생	회사원	1982-02-06
A-1002	이길자	모	가사	1958-11-17
A-1002	정명재	오빠	연구원	1982-06-28
A-1002	정명수	남동생	대학생	1989-03-09

거래처

사원번호	사원명	거래처	담당자	거래처주소	거래처전화번호	거래체결날짜
A-1001	홍길동	동일물산	이효린	서울시 서초구	02-1111-0000	2000-06-05
A-1001	홍길동	백일상사	김병선	서울시 영등포구	02-1111-1111	2000-10-18
A-1002	정명선	선진기업	안지환	서울시 강서구	02-1111-2222	2000-03-25
A-1002	정명선	대성기획	나윤구	서울시 노원구	02-1111-3333	2001-08-02
A-1003	황성인	명진건설	장철수	서울시 종로구	02-1111-4444	2000-02-15
A-1003	황성인	제일설비	김혜성	서울시 양천구	02-1111-5555	2002-05-28



[Quiz] 키(Key)

[2개 테이블 : 기본키, 외래키]

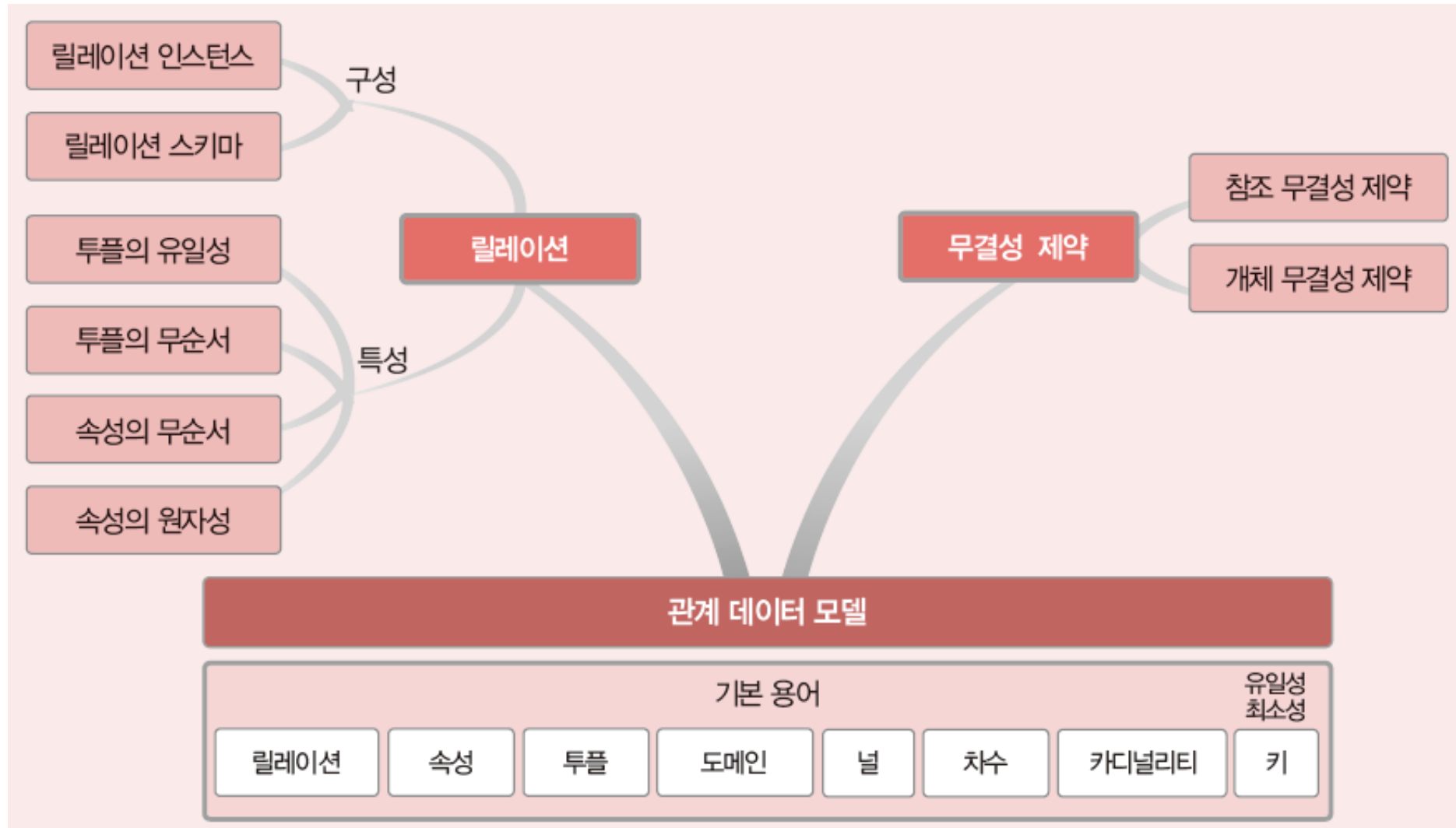
간호사

사번	성명	나이	업무
1111	홍길동	35	외래
2222	김윤식	46	병동
3333	이정진	32	종환자실
4444	홍진아	28	외래

연락처

사번	성별	연락처
1111	남	111-1111
2222	남	222-2222
4444	여	444-4444

관계 데이터 모델



실습

3주차 3-03

담당교수: 김희숙
(jasmin11@hanmail.net)

SQL



❖ SQL(Structured Query Language) : 데이터베이스 언어

종류	약어	명령어	비고
데이터 정의어	DDL (Data Definition Language)	CREATE ALTER DROP	객체 생성 객체 수정 객체 삭제
데이터 조작어	DML (Data Manipulation Language)	SELECT INSERT UPDATE DELETE	데이터 검색 데이터 입력 데이터 수정 데이터 삭제
데이터 제어어	DCL (Data Control Language)	GRANT REVOKE	권한부여 권한취소
	TCL (Transaction Control Language)	COMMIT; ROLLBACK; SAVEPOINT;	작업완료 작업취소 저장포인트

데이터 입력/수정/삭제/조회(login.sql)

- 0) login 테이블을 생성하라
- 1) login 테이블에 데이터 입력하라 (INSERT)
- 2) login 테이블의 아이디 admin 의 암호를 admin 으로 수정하라 (UPDATE)
- 3) login 테이블의 아이디 kkk 를 삭제하라 (DELETE)
- 4) 데이터 조회하라

[실습 1] 1-1) login 테이블 생성하고 데이터 입력하기

테이블 생성, 데이터 입력	login 테이블을 생성하고 데이터를 입력하는 SQL문법								
login <table border="1"> <thead> <tr> <th>L_id</th><th>L_pw</th></tr> </thead> <tbody> <tr> <td>admin</td><td>1234</td></tr> <tr> <td>guest</td><td>guest</td></tr> <tr> <td>kkk</td><td>kkk</td></tr> </tbody> </table>	L_id	L_pw	admin	1234	guest	guest	kkk	kkk	--login 테이블 생성 CREATE TABLE login (L_id char(8) NOT NULL PRIMARYKEY, L_pw varchar(10) NOT NULL) --login 데이터 입력 INSERT INTO login VALUES ('admin', '1234') INSERT INTO login VALUES ('guest', 'guest') INSERT INTO login VALUES ('kkk', 'kkk')
L_id	L_pw								
admin	1234								
guest	guest								
kkk	kkk								

1-2) login 테이블에서 데이터 수정하기

데이터 수정 (UPDATE)	특정 아이디의 패스워드를 수정하는 SQL문법								
login <table border="1"> <thead> <tr> <th>L_id</th><th>L_pw</th></tr> </thead> <tbody> <tr> <td>admin</td><td>admin</td></tr> <tr> <td>guest</td><td>guest</td></tr> <tr> <td>kkk</td><td>kkk</td></tr> </tbody> </table>	L_id	L_pw	admin	admin	guest	guest	kkk	kkk	--데이터 수정 UPDATE login SET L_pw='admin' WHERE L_id='admin' --데이터 조회 SELECT * FROM login
L_id	L_pw								
admin	admin								
guest	guest								
kkk	kkk								

1-3) login 테이블에서 데이터 삭제하기

데이터 삭제 (DELETE)	특정 아이디를 삭제하는 SQL문법						
login <table border="1"> <thead> <tr> <th>L_id</th><th>L_pw</th></tr> </thead> <tbody> <tr> <td>admin</td><td>admin</td></tr> <tr> <td>guest</td><td>guest</td></tr> </tbody> </table>	L_id	L_pw	admin	admin	guest	guest	--데이터 삭제 DELETE FROM login WHERE L_id='kkk' --데이터 조회 SELECT * FROM login
L_id	L_pw						
admin	admin						
guest	guest						

[예제] SQL: 테이블 생성(기본키, 외래키)

테이블 명세서

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열 이름	비고
학과	학과번호	int	NOT NULL	PK				
	학과명	varchar(50)						

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열 이름	비고
학생	번호	int	NOT NULL	PK				
	이름	varchar(12)						
	학년	int						
	분반	char(2)						
	학과번호	int			FK	학과	학과번호	

학과

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	정보통신과

학생

번호	이름	학년	분반	학과번호
1	한지혜	1	YB	1
2	이정우	1	YA	1
3	오지영	2	J1	2
4	강재미	1	YB	1
5	박철호	2	J1	2



[예제] SQL: 테이블 생성(기본키, 외래키)

```
CREATE TABLE 학과 (  
    학과번호 int NOT NULL ,  
    학과명 varchar(50) ,  
    PRIMARY KEY(학과번호)  
);
```

학과

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	정보통신과

학생

번호	이름	학년	분반	학과번호
1	한지혜	1	YB	1
2	이정우	1	YA	1
3	오지영	2	J1	2
4	강재미	1	YB	1
5	박철호	2	J1	2

```
CREATE TABLE 학생 (  
    번호 int NOT NULL ,  
    이름 varchar(12) ,  
    학년 int ,  
    분반 char(2) ,  
    학과번호 int ,  
    PRIMARY KEY(번호) ,  
    FOREIGN KEY (학과번호)  
        REFERENCES 학과(학과번호)  
);
```


--학과(학과번호, 학과명)

insert into 학과 values(1,'컴퓨터소프트웨어공학과');

insert into 학과 values(2,'컴퓨터정보공학과');

insert into 학과 values(3,'정보통신과');

--학생(번호, 이름, 학년, 분반, 학과번호)

insert into 학생 values(1,'한지혜',1,'YB',1);

insert into 학생 values(2,'이정우',1,'YA',1);

insert into 학생 values(3,'오지영',2,'J1',2);

insert into 학생 values(4,'강재미',1,'YB',1);

insert into 학생 values(5,'박철호',2,'Y1',2);

commit;

select * from 학과;

select * from 학생;

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열 이름	비고
학과	학과번호	int	NOT NULL	PK				
	학과명	varchar(50)						

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열 이름	비고
학생	번호	int	NOT NULL	PK				
	이름	varchar(12)						
	학년	int						
	분반	char(2)						
	학과번호	int			FK	학과	학과번호	

학과

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	정보통신과

학생

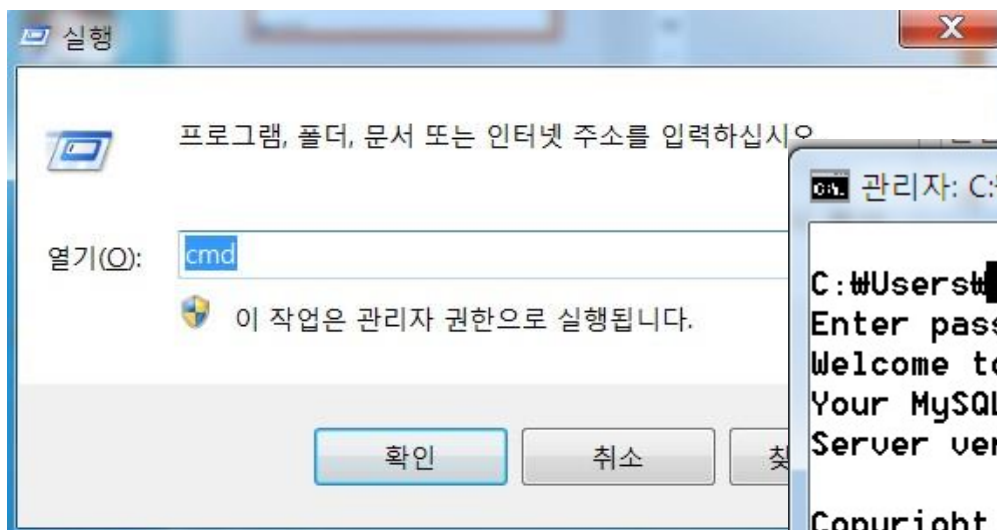
번호	이름	학년	분반	학과번호
1	한지혜	1	YB	1
2	이정우	1	YA	1
3	오지영	2	J1	2
4	강재미	1	YB	1
5	박철호	2	J1	2

[실습] MySQL 실행: cmd

[실습 1] SQL 문법을 사용하여 테이블 생성하고 데이터 입력하시오 (dept-mysql.sql)

--부서(부서코드, 부서명, 위치)	데이터 조회(MySQL)
<pre>/* 부서(부서코드, 부서명, 위치) */ drop table if exists 부서; -- 부서 테이블 생성 CREATE TABLE 부서 (부서코드 char(2) NOT NULL, 부서명 char(9), 위치 char(6), PRIMARY KEY (부서코드));</pre>	<pre>--파일(스크립트 파일)로 실행하는 방법 c:/>mysql -uroot -p -- 1단계: 데이터베이스 생성 CREATE DATABASE testdb default character set utf8 collate utf8_bin; -- 사용할 데이터베이스 선택 USE testdb;</pre>
<pre>-- 부서 데이터입력 insert into 부서 values('AA', '총무부', '서울'); insert into 부서 values('BB', '영업부', '대전'); insert into 부서 values('CC', '기획부', '서울'); select * from 부서;</pre>	<pre>mysql>source c:/DBDATA/dept-mysql.sql; --위의 문법을 실습한 후에 테이블을 삭제하시오 --(MySQL) mysql>drop table 부서; mysql>drop database testdb;</pre>

[실습] (예제) MySQL 실행(cmd)



```
관리자: C:\Windows\system32\cmd.exe - mysql -uroot -p

C:\Users\st[redacted]>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database testdb
-> default character set utf8
-> collate utf8_bin;
Query OK, 1 row affected (0.00 sec)

mysql> use testdb;
Database changed
mysql>
```

[실습] (예제) MySQL 실행(cmd)

dept-mysql - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

/* 부서(부서코드, 부서명, 위치) */

drop table if exists 부서;

-- 부서 테이블 생성

```
CREATE TABLE 부서 (  
    부서코드      char(2) NOT NULL,  
    부서명        char(9),  
    위치          char(6),  
    PRIMARY KEY (부서코드)  
);
```

-- 부서 데이터 입력

```
insert into 부서 values('AA','총무부','서울');  
insert into 부서 values('BB','영업부','대전');  
insert into 부서 values('CC','기획부','서울');
```

select * from 부서;

관리자: C:\Windows\system32\cmd.exe - mysql -uroot -p

Query OK, 1 row affected (0.00 sec)

mysql> use testdb;

Database changed

mysql> source c:/dbdata/dept-mysql.sql;

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.30 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

부서코드	부서명	위치
AA	총무부	서울
BB	영업부	대전
CC	기획부	서울

3 rows in set (0.00 sec)

mysql>

[실습] (예제) MySQL 실행(cmd)

dept-mysql - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

/* 부서(부서코드, 부서명, 위치) */

drop table if exists 부서;

-- 부서 테이블 생성

```
CREATE TABLE 부서 (  
    부서코드      char(2) NOT NULL,  
    부서명        char(9),  
    위치          char(6),  
    PRIMARY KEY (부서코드)  
);
```

-- 부서 데이터입력

```
insert into 부서 values('AA', '총무부', '서울');  
insert into 부서 values('BB', '영업부', '대전');  
insert into 부서 values('CC', '기획부', '서울');
```

select * from 부서;

관리자: C:\Windows\system32\cmd.exe

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

부서코드	부서명	위치
AA	총무부	서울
BB	영업부	대전
CC	기획부	서울

3 rows in set (0.00 sec)

mysql> -- 부서 테이블 삭제

mysql> drop table 부서;

Query OK, 0 rows affected (0.01 sec)

mysql> -- testdb 데이터베이스 삭제

mysql> drop database testdb;

Query OK, 0 rows affected (0.00 sec)

mysql> exit

Bye

C:\Userst >



[실습] 테이블 생성: 기본키, 외래키

[실습 2] 부서, 사원 테이블 생성 하고 데이터 입력하시오 (empdept-non.sql)

기본키 설정	외래키 설정
<pre>/* 부서(부서코드, 부서명) */ /* 사원(사원번호, 이름, 부서코드) */ drop table 사원; drop table 부서; CREATE TABLE 부서 (부서코드 char(2) NOT NULL , 부서명 varchar(10), PRIMARY KEY(부서코드));</pre>	<pre>CREATE TABLE 사원 (사원번호 char(4) NOT NULL , 이름 varchar(12) , 부서코드 char(2), PRIMARY KEY(사원번호), FOREIGN KEY (부서코드) REFERENCES 부서(부서코드)); --데이터 조회 select * from 부서; select * from 사원;</pre>

[실습] MySQL 실행: Workbench



-- MySQL Workbench 실행

-- (MySQL)

drop database testdb;

```
CREATE DATABASE `testdb`  
  DEFAULT CHARACTER SET utf8;
```

use testdb;

1단계: 스키마/데이터베이스 생성 testdb

2단계: 테이블 생성

학과, 학생

3단계: 데이터 입력

-- 1단계: 스키마 생성

```
CREATE SCHEMA `testdb` DEFAULT CHARACTER SET utf8 ;
```

-- 사용할 데이터베이스 선택

```
USE testdb;
```

-- 데이터베이스 삭제

-- (MySQL)

```
drop database testdb;
```

[실습] (예제) MySQL 실행

[실습 2] 부서, 사원 테이블 생성 하고 데이터 입력하시오 (empdept-non.sql)

기본키 설정	외래키 설정
<pre> /* 부서(부서코드, 부서명) */ /* 사원(사원번호, 이름, 부서코드) */ drop table 사원; drop table 부서; CREATE TABLE 부서 (부서코드 char(2) NOT NULL , 부서명 varchar(10), PRIMARY KEY(부서코드)); </pre>	<pre> CREATE TABLE 사원 (사원번호 char(4) NOT NULL , 이름 varchar(12), 부서코드 char(2), PRIMARY KEY(사원번호), FOREIGN KEY (부서코드) REFERENCES 부서(부서코드)); --데이터 조회 select * from 부서; select * from 사원; </pre>

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열이름	비고
부서	부서코드	char(2)	NOT NULL	PK				
	부서명	varchar(10)	NULL					

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열이름	비고
사원	사원번호	char(4)	NOT NULL	PK				
	사원명	varchar(12)	NULL					
	부서코드	char(2)	NULL		FK	부서	부서코드	

```

insert into 부서(부서코드, 부서명) values('AA','총무부');
insert into 부서(부서코드, 부서명) values('BB','영업부');
insert into 부서(부서코드, 부서명) values('CC','기획부');

```

```

insert into 사원 values('1111','홍길동','AA');
insert into 사원 values('2222','임꺽정','AA');
insert into 사원 values('3333','박찬호','BB');
insert into 사원 values('4444','선동열','BB');
insert into 사원 values('5555','차두리','AA');
insert into 사원 values('6666','신동엽','BB');

```

```

select * from 부서;
select * from 사원;

```

	부서코드	부서명
▶	AA	총무부
	BB	영업부
	CC	기획부

	사원번호	이름	부서코드
▶	1111	홍길동	AA
	2222	임꺽정	AA
	3333	박찬호	BB
	4444	선동열	BB
	5555	차두리	AA
	6666	신동엽	BB

[과제01] (데이터베이스 관리)

3주차

담당교수: 김희숙
(jasmin11@hanmail.net)



[과제01] (MySQL)

SQL작성(MySQL)
실습하기
(화면캡처 워드작성)

□[과제01] **MySQL 프로그램 사용**하여 테이블 각각 작성하고
화면캡처하여 워드에 작성하시오

1) 제출방법: 원격강의-과제

첨부파일로 제출

2) 파일명: 과제01-분반_이름_학번-작성날짜.doc (또는 hwp)

(예. 과제01-YA_홍길동_20209999-제출날짜)

3) 제출마감일: 20/09/28 (월) 23:55 까지

* (작성한 SQL파일은 저장해 두세요: 각자 별도로 보관해 두시기 바랍니다)

제출방법:
원격강의 [과제] 제출

[과제01] (실습)

SQL작성(MySQL) 실습하기 (화면캡처 워드작성)

[과제01] 다음을 SQL문법으로 작성하여 실습하시오(MySQL 실습)

* (문제파일: HW1-MySQL-테이블2개-stu.pdf)

[과제 1-1] 다음 테이블 2개 생성
(기본키, 외래키 작성) 하고 데이터 입력

환자 테이블

환자번호	환자이름	나이	담당의사
P001	오우진	31	D002
P002	채광주	50	D001
P003	김용욱	43	D003

의사 테이블

의사번호	의사이름	소속	근무연수
D001	정지영	내과	5
D002	김선주	피부과	10
D003	정성호	정형외과	15

SQL 작성

[과제 1-2] 테이블 생성

- 1-1) 테이블 생성 → 다음 테이블을 생성하는 SQL 문법 작성(기본키 설정)
- 1-2) 데이터 입력 → 본인 이름을 입력한 레코드를 입력하시오
- 1-3) 데이터 조회 → 테이블 조회한 결과를 화면캡처하여 작성하시오
- 1-4) 차수(degree) 는 얼마인가?
- 1-5) 카디날리티(cardinality)는 얼마인가?
- 1-6) 사원이름 속성은 기본키로 적합하지 않다. 그 이유는 무엇인가?
- 1-7) 기본키로 적합한 속성은 무엇인가?

사원번호	사원이름	나이	주소	직급
E001	홍준화	30	서울시 마포구	대리
E002	김연주	28	서울시 영등포구	사원
E003	이명기	32	서울시 강남구	사원