

데이터베이스관리

10주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[용어] 릴레이션(테이블)

- 부서(부서번호 pk, 부서이름)
- 직원(직원번호 pk, 직원이름, 소속부서 fk)

[복습] (테이블 2개) 관계형 데이터 모델: (릴레이션)

부서

부서번호	부서이름	부서위치
1	인사부	서울
2	연구부	대전
3	홍보부	서울

[용어] 릴레이션(테이블)

[실습] 관계형 데이터 모델: (릴레이션)

부서

a)

b)

부서번호	부서이름	부서위치
1	인사부	서울
2	연구부	대전
3	홍보부	서울

릴레이션(테이블)

스키마
인스턴스

튜플, 레코드, 행
속성, 필드, 열

도메인

※ 관계 스키마 표기법

부서(부서번호, 부서이름, 부서위치)
pk

[실습] 필드 추가/데이터 추가

[Quiz 1-01] 스키마 수정(필드 추가) / 데이터 추가(레코드 입력)

-- 부서(부서번호 pk, 부서이름)



-- 부서(부서번호 pk, 부서이름, **부서위치**)

부서

부서번호	부서이름
1	인사부
2	연구부
3	홍보부



필드 추가

부서위치
서울
대전
서울

레코드 추가

4	영업부
---	-----

[실습] [실습] 필드 추가/데이터 추가 (ans)

-- [Quiz 1-01] 부서

```

/* --- 부서 스키마 ----- */
drop table 부서;

CREATE TABLE 부서 (
    부서번호 int          NOT NULL ,
    부서이름 varchar(10) ,
    PRIMARY KEY(부서번호)
);

insert into 부서 values(1,'인사부');
insert into 부서 values(2,'연구부');
insert into 부서 values(3,'홍보부');
/* ----- */
    
```

-- 부서(부서번호,부서이름)

부서

부서번호	부서이름
1	인사부
2	연구부
3	홍보부

레코드 추가

4	영업부
---	-----

```

-- 레코드 추가
insert into 부서
values(4,'영업부');
    
```

필드 추가

-- 부서(부서번호,부서이름,부서위치)

부서

부서번호	부서이름	부서위치
1	인사부	서울
2	연구부	대전
3	홍보부	서울

```

-- 필드 추가
ALTER TABLE 부서
ADD 부서위치 varchar(10);
    
```

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**

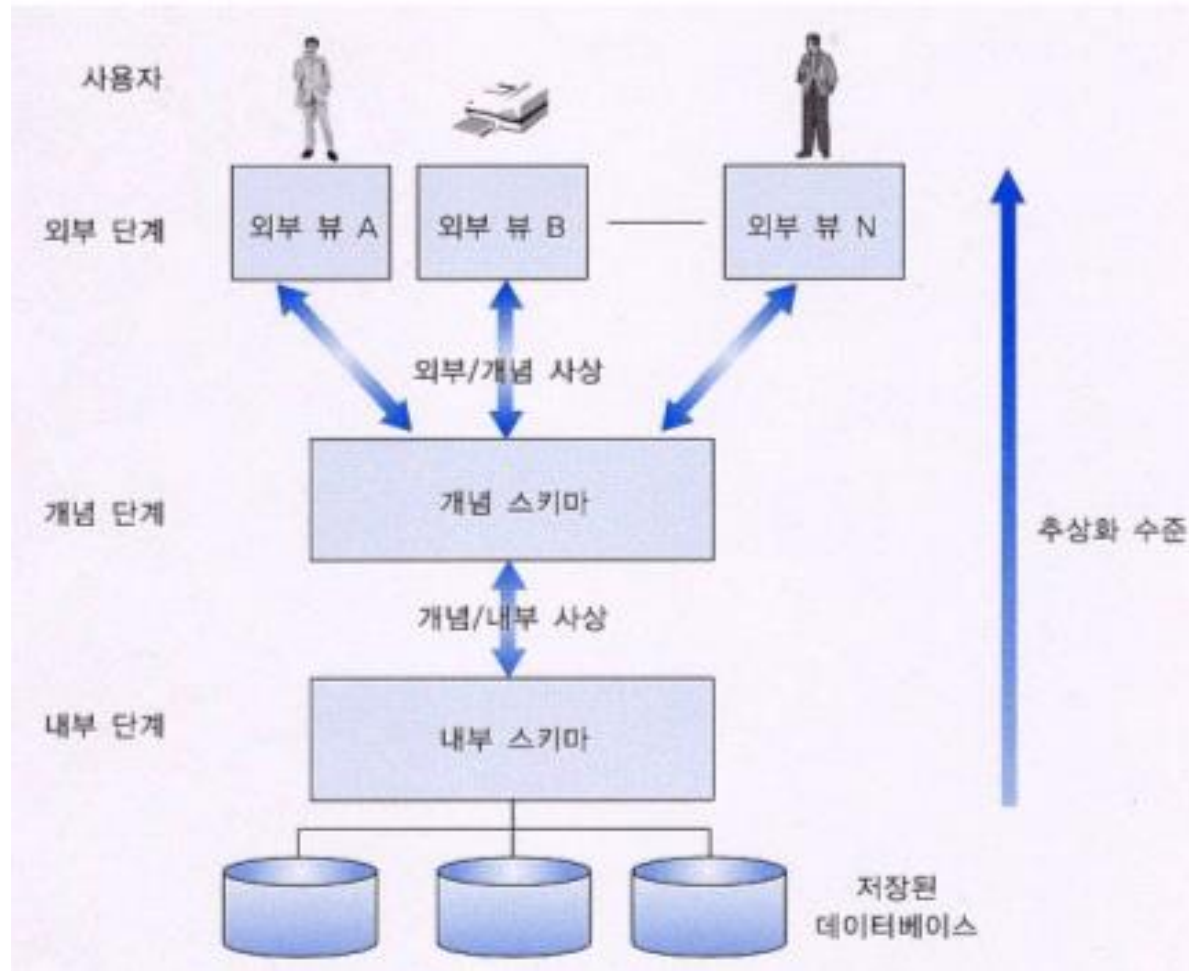


- * **파일시스템 단점:**

- 1) 데이터 **중복성**
- 2) 데이터 **종속성**

[요약] 3단계 구조

ANSI/SPARC 아키텍처 (스키마 3단계 구조)



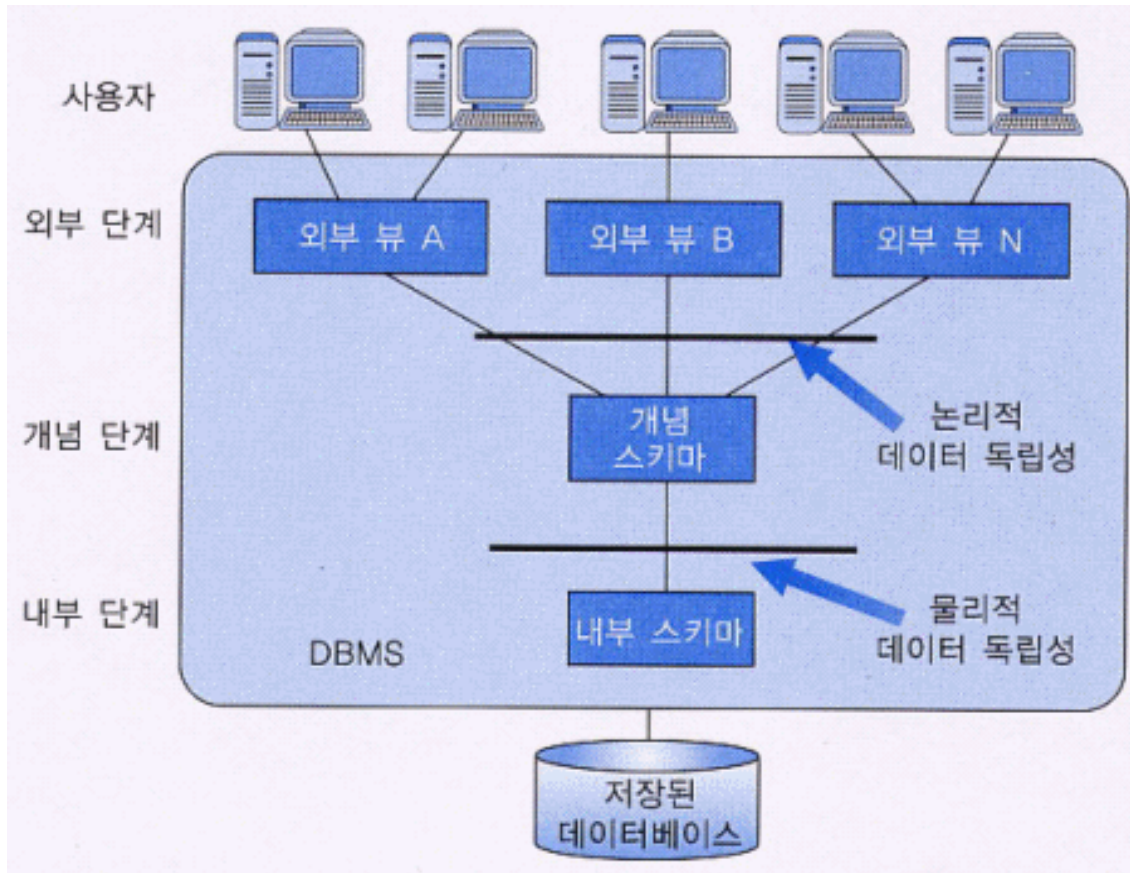
(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)



[요약] 데이터 독립성

* 3단계 스키마 구조:

- 1) 외부 스키마
- 2) 개념 스키마
- 3) 내부 스키마



- **논리적 데이터 독립성:**

개념스키마가 변화해도
외부스키마는 영향 받지 않는다

- **물리적 데이터 독립성:**

내부스키마가 변화해도
개념스키마, 외부스키마는 영향 받지 않는다

(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)

[요약] 키(Key)

□키

- 각 튜플을 고유하게 식별할 수 있는 하나 이상의 애트리뷰트 들의 모임
- ✓ 수퍼키(super key)
- ✓ 후보키(candidate key)
- ✓ 기본키(primary key)
- ✓ 대체키(alternate key)
- ✓ 외래키(foreign key)

키	특성
수퍼키	유일성
후보키	유일성, 최소성
기본키	중복불가, 필수입력
대체키	
외래키	

[요약] 데이터 무결성(integrity)

1. 무결성 제약조건

- 1) **개체 무결성 제약조건**: 기본키는 널 값을 가질 수 없다(기본키)
- 2) **참조 무결성 제약조건**: (외래키)
참조하는 테이블의 외래키 값은
참조되는 테이블의 기본키 값과 같다.
- 3) **키 제약조건**: 키는 중복된 값을 갖지 않는다
- 4) **도메인 제약조건**: CHECK, DEFAULT 문법

Integrity Constraint

- 1) entity integrity constraint
- 2) referential integrity constraint
- 3) uniqueness integrity constraint
- 4) domain integrity constraint

[실습] 테이블 생성(테이블 3개)

[실습 1-2] (테이블 3개) 학생, 수강, 과목 테이블 생성(기본키, 외래키)

❖ 기본키가 복합키인 경우

```
drop table ;
drop table ;
drop table 학생;
```

```
--학생(학번, 학생명, 학년)
 학생 (
    학번      char(4) NOT NULL,
    학생명    varchar(12),
    학년      int,
     KEY(학번)
);
```

```
--과목(과목번호, 과목명)
 과목 (
    과목번호  char(5),
    과목명    varchar(30),
     KEY(과목번호)
);
```

```
--수강(학번, 과목번호, 성적)
 수강 (
    학번      ,
    과목번호  char(5),
    성적      int,
     KEY() ,
     KEY (학번) REFERENCES ,
     KEY (과목번호) REFERENCES 
);
```

[3개 테이블 : 기본키, 외래키]

학생

학번	학생명	학년
1111	홍길동	1
2222	김윤식	3
3333	이정진	2
4444	홍진아	1

수강

학번	과목번호	성적
1111	CS100	98
1111	CS102	88
2222	CS102	90
3333	CS100	92

과목

과목번호	과목명
CS100	데이터베이스
CS101	운영체제
CS102	자료구조

실습

담당교수: 김희숙
(jasmin11@hanmail.net)

SQL: 그룹화(group by)

10주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 집계함수

-- 사원(직원코드,성명,직책,연봉)

[실습] 집계함수, 그룹화(GROUP BY)

사원

직원코드	성명	직책	연봉
161353	김미나	대리	2300
181323	최영락	사원	1900
151453	홍진호	과장	2800
135485	구준표	과장	3000
104895	강나래	팀장	3600
165484	김하늘	대리	2400

COUNT(*): 널 값 포함
COUNT(필드): 널 값 제외

SUM()
AVG()
MAX()
MIN()

SELECT
FROM
GROUP BY
HAVING

SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY

[실습 1-01] SELECT (집계함수)

[실습] (집계함수)

-- 집계함수

CREATE TABLE 성적(

이름 varchar(9) NOT NULL primary key,

점수 int

);

INSERT INTO 성적 (이름, 점수) VALUES ('홍길동', 87);

INSERT INTO 성적 (이름, 점수) VALUES ('임꺽정', 60);

INSERT INTO 성적 (이름, 점수) VALUES ('박찬호', 75);

INSERT INTO 성적 (이름, 점수) VALUES ('선동열', 70);

INSERT INTO 성적 (이름, 점수) VALUES ('홍명보', 90);

INSERT INTO 성적 (이름, 점수) VALUES ('차범근', 75);

INSERT INTO 성적 (이름, 점수) VALUES ('강성범', 68);

INSERT INTO 성적 (이름, 점수) VALUES ('신동엽', null);

18 • select * from 성적;

Result Grid		Filter Rows:
	이름	점수
▶	강성범	68
	박찬호	75
	선동열	70
	신동엽	NULL
	임꺽정	60
	차범근	75
	홍길동	87
	홍명보	90
*	NULL	NULL

[실습 1-01] SELECT (집계함수)

[실습] (집계함수)

-- 성적(이름, 점수)

-- 1-1) 최고 점수를 검색하라

-- 1-2) 최저 점수를 검색하라

-- 1-3) 점수합계를 검색하라

-- 1-4) 평균점수를 검색하라

-- 1-5) 학생수는 모두 몇 명인지 검색하라

-- 1-6) 시험에 응시한 학생수는 모두 몇 명인지 검색하라

18 • select * from 성적;

	이름	점수
▶	강성범	68
	박찬호	75
	선동열	70
	신동엽	NULL
	임꺽정	60
	차범근	75
	홍길동	87
	홍명보	90
•	NULL	NULL

[실습 1-02] SELECT (그룹화)

[실습] (그룹화)

-- 그룹화

```
CREATE TABLE 성적2 (
  이름 varchar(9) NOT NULL primary key ,
  과목 varchar(8),
  점수 int
);

INSERT INTO 성적2 VALUES ('홍길동', '영어',87 );
INSERT INTO 성적2 VALUES ('임꺽정', '수학',60 );
INSERT INTO 성적2 VALUES ('박찬호', '국어',75 );
INSERT INTO 성적2 VALUES ('선동열', '영어',70 );
INSERT INTO 성적2 VALUES ('홍명보', '수학',90 );
INSERT INTO 성적2 VALUES ('차범근', '수학',75 );
INSERT INTO 성적2 VALUES ('강성범', '수학',68 );
INSERT INTO 성적2 VALUES ('신동엽', '영어',null);
```

40 • SELECT * FROM 성적2;

< Result Grid Filter Rows:

	이름	과목	점수
▶	강성범	수학	68
	박찬호	국어	75
	선동열	영어	70
	신동엽	영어	NULL
	임꺽정	수학	60
	차범근	수학	75
	홍길동	영어	87
	홍명보	수학	90
*	NULL	NULL	NULL

[실습 1-02] SELECT (그룹화)

[실습] 그룹화, 부분합(GROUP BY)

- 성적2(이름, 과목, 점수)
- 2-1) 각 과목수는 몇 개인지 검색하라(DISTINCT 사용)
- 2-2) 과목별 수강생은 몇 명인지 검색하라(GROUP BY)
- 2-3) 과목별 평균점수를 검색하라(GROUP BY)

SELECT 과목, COUNT(점수)

FROM 성적2

GROUP BY 과목;

- 2-4) 과목별 평균점수 75 보다 높은 학생의
- 과목별 평균점수를 검색하라(HAVING)
- 2-5) 점수가 70 이상인 과목이름, 과목 평균점수를
- 과목의 과목별 평균점수가 75 이상인 것만
- 과목별 평균점수가 높은 순으로 검색하라(ORDER BY)

40 • SELECT * FROM 성적2;

Result Grid Filter Rows:

	이름	과목	점수
▶	강성범	수학	68
	박찬호	국어	75
	선동열	영어	70
	신동엽	영어	NULL
	임꺽정	수학	60
	차범근	수학	75
	홍길동	영어	87
	홍명보	수학	90
*	NULL	NULL	NULL

[실습] [실습 1-01](MySQL) 집계함수(성적) (ans)

-- [실습] (집계함수)

-- 성적(이름, 점수)

-- 1-1) 최고 점수를 검색하라

```
SELECT MAX(점수)
FROM 성적;
```

-- 1-2) 최저 점수를 검색하라

```
SELECT MIN(점수)
FROM 성적;
```

-- 1-3) 점수합계를 검색하라

```
SELECT SUM(점수)
FROM 성적;
```

-- 1-4) 평균점수를 검색하라

```
SELECT AVG(점수)
FROM 성적;
```

-- [실습] (집계함수)

-- 성적(이름, 점수)

-- 1-5) 학생수는 모두 몇 명인지 검색하라

```
SELECT COUNT(*)
FROM 성적;
```

-- 1-6) 시험에 응시한 학생수는 모두 몇 명인지 검색하라

-- (MySQL/MS SQL)

```
SELECT COUNT(점수) as '응시 학생수'
FROM 성적;
```

-- (Oracle) 출력 필드명 띄어쓰기

```
SELECT COUNT(점수) as "응시 학생수"
FROM 성적;
```

* COUNT(*) : NULL 포함하여 계산

* COUNT(필드) : NULL 제외하여 계산

18 • select * from 성적;

Result Grid | Filter Rows:

	이름	점수
▶	강성범	68
	박찬호	75
	선동열	70
	신동엽	NULL
	임꺽정	60
	차범근	75
	홍길동	87
	홍명보	90
★	NULL	NULL

[실습] [실습 1-02](MySQL) 그룹화(성적2) (ans)

-- [실습] (그룹화)

-- 성적2(이름, 과목, 점수)

-- 2-1) 각 과목수는 몇 개인지 검색하라(DISTINCT 사용)

```
SELECT COUNT(DISTINCT 과목)
FROM 성적2
```

-- 2-2) 과목별 수강생은 몇 명인지 검색하라(GROUP BY)

```
SELECT 과목, COUNT(점수)
FROM 성적2
GROUP BY 과목;
```

-- 2-3) 과목별 평균점수를 검색하라(GROUP BY)

```
SELECT 과목, AVG(점수)
FROM 성적2
GROUP BY 과목;
```

40 • SELECT * FROM 성적2;

< Result Grid Filter Rows:

	이름	과목	점수
▶	강성범	수학	68
	박찬호	국어	75
	선동열	영어	70
	신동엽	영어	NULL
	임꺽정	수학	60
	차범근	수학	75
	홍길동	영어	87
	홍명보	수학	90
*	NULL	NULL	NULL

[실습] [실습 1-02](MySQL) 그룹화(성적2) (ans)

-- [실습] (그룹화)

-- 성적2(이름, 과목, 점수)

-- 2-4) 과목별 평균점수 75 보다 높은 학생의 과목별 평균점수를 검색하라
(HAVING)

```
SELECT 과목, AVG(점수)
FROM 성적2
GROUP BY 과목
HAVING AVG(점수) >= 75 ;
```

-- 2-5) 점수가 70 이상인 과목이름, 과목 평균점수를 과목의 과목별 평균점수가 75 이상인 것만 과목별 평균점수가 높은 순으로 검색하라(ORDER BY)

```
SELECT 과목, AVG(점수)
FROM 성적2
WHERE 점수 >= 70
GROUP BY 과목
HAVING AVG(점수) >= 75
ORDER BY AVG(점수) ASC;
```

40 • SELECT * FROM 성적2;

	이름	과목	점수
▶	강성범	수학	68
	박찬호	국어	75
	선동열	영어	70
	신동엽	영어	NULL
	임격정	수학	60
	차범근	수학	75
	홍길동	영어	87
	홍명보	수학	90
	NULL	NULL	NULL

/* SELECT 문법 순서 */

```
SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY
```

-- ROUND() 함수

```
select 과목, ROUND(AVG(점수))
from 성적2
group by 과목;
```

실습(Quiz)

담당교수: 김희숙
(jasmin11@hanmail.net)

[실습] [Quiz 1] fruit 테이블(MySQL)

[Quiz 1]

1) 다음과 같은 fruit 테이블을 생성하시오
(기본키: 과일번호)

데이터 입력하는 SQL문법을 작성하라

fruit

과일번호	과일이름	가격	색상	원산지
1	사과	2000	빨간색	한국
2	배	5000	노란색	한국
3	바나나	1500	노란색	인도네시아
4	키위	4000	초록색	호주
5	수박	6000	초록색	한국

- 2-1) 노란색 과일의 이름, 색상을 검색하라.
- 2-2) 과일이름을 오름차순으로 정렬하여 모든 필드를 검색하라
- 2-3) 가격이 5000 이하인 빨간색 과일의 이름을 검색하라
- 2-4) 가격이 2000 ~ 5000 사이의 과일이름과 가격, 원산지를 검색하라(BETWEEN AND)
- 2-5) 색상이 노란색이거나 초록색인 과일이름, 색상을 검색하라 (IN 사용)
- 2-6) 색상이 노란색이거나 초록색인 과일이름, 색상을 검색하라 (논리연산자 OR 사용)
- 2-7) 원산지가 한국이 아닌 과일이름, 원산지를 검색하라

- 3-1) UPDATE 문을 이용하여 사과의 가격을 2배로 하여 수정하라
- 3-2) DELETE 문을 이용하여 원산지가 호주인 과일의 목록을 삭제하라
- 3-3) ALTER TABLE 문을 이용하여 과일정보 필드를 추가하라 (데이터형식 varchar(20))

[실습] [Quiz 2] 고객 테이블(MySQL) (stu)

고객

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용축	22	silver	학생	NULL
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

- 1) 직업이 학생, 간호사, 교사인 고객의 이름, 직업, 등급을 검색
- 2) 적립금이 300 이상인 고객 중에서 나이를 알 수 없는 고객아이디, 나이, 적립금을 검색하라
- 3) 등급이 gold인 고객 중에서 고객아이디, 등급, 적립금을 검색하라. 단, 등급을 오름차순 정렬하고 등급이 같으면 적립금의 내림차순 정렬하라
- 4) 등급별 적립금 평균을 검색하라,
(단, 적립금 평균이 1000원 이상인 등급에 대해 검색)

- 고객 테이블은
- 고객아이디, 고객이름, 나이, 등급, 직업, 적립금 속성으로 구성되고,
- 고객아이디 속성이 기본키다.
- 고객이름과 등급 속성은 값을 반드시 입력해야 하고,
- 적립금 속성은
- 값을 입력하지 않으면 0이 기본으로 입력되도록 한다

[과제02] (데이터베이스 관리)

교재 7장 SELECT 예제 (01)

담당교수: 김희숙
(jasmin11@hanmail.net)

[과제02] (MySQL)

(MySQL)
실습하기
(실습 화면캡처하여
워드에 작성)

□[과제02] **MySQL 프로그램 사용**

□**SELECT** 문제, **SQL** 답안, 실행결과 각각 화면캡처하여 워드에 작성

1) 제출방법: eClass-과제

첨부파일로 제출

2) 파일명: 과제02-분반_이름_학번-작성날짜.doc (또는 hwp)

(예. 과제02-YA_홍길동_23229999-제출날짜)

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

제출방법:
eClass [과제] 제출

[과제02] (실습)

(MySQL)
실습하기
(화면캡처 워드작성)

[과제02] 다음을 SQL문법으로 작성하여 실습하시오(MySQL 실습)

* (문제파일: HW02-handb-select-stu)

```
use handb;
```

- 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)
- 제품(제품번호, 제품명, 재고량, 단가, 제조업체)
- 주문(주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자)

(MySQL) 실습하기 (화면캡처 워드작성)

[실습 1] 다음에 대하여 SQL문법과 실행결과를 작성하라

기본검색	예제7-10) 고객 테이블에서 고객아이디, 고객이름, 등급 속성을 검색한다 예제7-11) 고객 테이블에 존재하는 모든 속성을 검색한다
중복	예제7-13) 제품 테이블에서 제조업체를 검색한다 예제7-14) 제품 테이블에서 제조업체를 검색하되, ALL 키워드를 사용한다 예제7-15) 제품 테이블에서 제조업체 속성을 중복 없이 검색한다(중복 제거) 예제7-16) 제품 테이블에서 제품명과 단가를 검색하되, 단가를 가격이라는 새 이름으로 출력한다
필드명명	예제7-17) 제품 테이블에서 제품명과 단가 속성을 검색하되, 단가에 500원을 더해 조정단가라는 새 이름으로 출력한다
조건식	예제7-18) 제품 테이블에서 한빛제과가 제조한 제품의 제품명, 재고량, 단가를 검색한다 예제7-19) 주문 테이블에서 apple 고객이 15개 이상 주문한 주문제품, 수량, 주문일자를 검색한다 예제7-20) 주문 테이블에서 apple 고객이 주문했거나 15개 이상 주문된 제품의 주문제품, 수량, 주문일자, 주문고객을 검색한다 예제7-21) 제품 테이블에서 단가가 2000원 이상이면서 3000원 이하인 제품의 제품명, 단가, 제조업체를 검색한다
LIKE	예제7-22) 고객 테이블에서 성이 김씨인 고객의 고객이름, 나이, 등급, 적립금을 검색한다 예제7-23) 고객 테이블에서 고객아이디가 5자인 고객의 고객아이디, 고객이름, 등급을 검색한다
NULL	예제7-24) 고객 테이블에서 나이가 아직 입력되지 않은 고객의 고객이름을 검색한다 예제7-25) 고객 테이블에서 나이가 이미 입력된 고객의 고객이름을 검색한다
정렬	예제7-26) 고객 테이블에서 고객이름, 등급, 나이를 검색하되, 나이를 기준으로 내림차순 정렬한다 예제7-27) 주문 테이블에서 수량이 10개 이상인 주문의 주문고객, 주문제품, 수량, 주문일자를 검색한다. 단, 주문제품을 기준으로 오름차순 정렬하고, 동일 제품은 수량을 기준으로 내림차순 정렬한다

[과제02] 다음을 SQL문법으로 작성하여 실습하시오(MySQL 실습)

[실습 2] 집계함수, group by

집계함수	예제7-28) 제품 테이블에서 모든 제품의 단가 평균을 검색한다
	예제7-29) 한빛제과에서 제조한 제품의 재고량 합계를 제품 테이블에서 검색한다
	예제7-30) 고객 테이블에 고객이 몇 명 등록되어 있는지 검색한다
	--1) 고객아이디 속성을 이용해 계산하는 경우
	--2) 나이 속성을 이용해 계산하는 경우(널 값 제외)
	--3) * 을 이용해 계산하는 경우
	예제7-31) 제품 테이블에서 제조업체의 수를 검색한다
	예제7-32) 주문 테이블에서 주문제품별 수량의 합계를 검색한다
	예제7-33) 제품 테이블에서 제조업체별로 제조한 제품의 개수와 제품 중 가장 비싼 단가를 검색하되, 제품의 개수는 제품수라는 이름으로 출력하고 가장 비싼 단가는 최고가라는 이름으로 출력한다
	예제7-34) 제품 테이블에서 제품을 3개 이상 제조한 제조업체별로 제품의 개수와 제품 중 가장 비싼 단가를 검색한다
group by	예제7-35) 고객 테이블에서 적립금 평균이 1000원 이상인 등급에 대해 등급별 고객 수와 적립금 평균을 검색한다
	예제7-36) 주문 테이블에서 각 주문고객이 주문한 제품의 총주문수량을 주문제품별로 검색한다