

# 데이터베이스관리

14주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# [Quiz] SELECT

(조인, group by)

담당교수: 김희숙  
(jasmin11@hanmail.net)



## [Quiz 2-01] 집계함수/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

- [Quiz 2-01] 집계함수/group by
- 1) 최대 연봉을 구하시오
- 2) 사원수를 구하시오
- 3) 직책의 개수를 중복없이 검색하시오(DISTINCT)

- 4) 직책별 연봉의 평균을 구하시오
- 5) 직책별 사원수를 구하시오
- 6) 연봉이 2500 이상인 사원이 2명 이상인 직책의 사원수를 구하시오

## [예제] 조인 (문제)

### [Quiz 1-01] 조인

69. 다음 두 릴레이션 R과 S에 대해 R을 왼쪽, S를 오른쪽 릴레이션으로 하고 속성 B를 조인 속성으로 하여 자연 조인(natural join), 왼쪽 외부 조인(left outer join), 전체 외부 조인(full outer join)을 수행하였을 때 결과 릴레이션의 카디널리티(cardinality)는 각각 얼마인가?

R		S	
A	B	B	C
a	2	1	x
b	3	2	y
c	3	3	z
d	5		

-- 결과 릴레이션의  
카디널리티는 각각 무엇인가

- 1) 자연조인
- 2) 왼쪽외부조인
- 3) 완전외부조인

# [Quiz] SELECT

(부)

담당교수: 김희숙  
(jasmin11@hanmail.net)



## [Quiz 3-01] 집계함수 / 뷰(view)

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

-- [Quiz 2-01] 집계함수/group by

-- 1) 다음 실행결과는 무엇인가?

```
SELECT COUNT(*),  
        COUNT(적립금),  
        COUNT(DISTINCT 직업)  
FROM   고객;
```

-- 뷰

CREATE VIEW 고객\_학생

as

```
SELECT 고객아이디, 등급, 직업,적립금  
FROM   고객  
WHERE  직업 = '학생';
```

-- 2) 다음 실행결과는 무엇인가?

```
insert into 고객_학생 values('shinme','gold','학생',500);  
select * from 고객 where 고객아이디='shinme';
```

# [Quiz] SELECT

(관계대수, 하위질의)

담당교수: 김희숙  
(jasmin11@hanmail.net)

## [예제] 집계함수, 그룹화, 하위질의 (문제)

R

A	B
a	2
b	4
c	2
c	4
c	6

S

C	D
a	1
b	2
c	4

-- 48 실행결과는 무엇인가

-- 1)

```
SELECT  COUNT(*)
FROM    R
WHERE   B = SOME ( SELECT D
                    FROM S
                    );
```

-- 48 실행결과는 무엇인가

-- 2)

```
SELECT  D, AVG(B)
FROM    R, S
WHERE   A = C
GROUP BY D
HAVING  COUNT(*) > 1;
```



## [예제] 집계함수, 그룹화, 하위질의 (ans)

R

A	B
a	2
b	4
c	2
c	4
c	6

S

C	D
a	1
b	2
c	4

-- 1)

COUNT(\*)

-- 2)

D AVG(B)

-- 48 실행결과는 무엇인가

-- 1)

```
SELECT  COUNT(*)
FROM    R
WHERE   B = SOME ( SELECT D
                    FROM S
                    );
```

-- 48 실행결과는 무엇인가

-- 2)

```
SELECT  D, AVG(B)
FROM    R, S
WHERE   A = C
GROUP BY D
HAVING  COUNT(*) > 1;
```

## [예제] 집합

-- 60 실행결과 튜플 수는 무엇인가?

-- 1) SQL문을 실행결과 **튜플 수**(카디날리티)

**(SELECT DISTINCT A FROM R)**

**UNION ALL**

**(SELECT A FROM S)**

R

A	B	C
1	a	10
1	a	11
1	a	25
2	b	22
3	b	21
5	c	17

S

A	D	E
1	p	x
1	p	y
2	q	y
4	r	w
6	s	z

## [예제] 관계대수

-- 10. 다음 관계 대수 표현의 의미는?

$$\pi_{\text{제목, 평점}}(\sigma_{\text{상영시간} \geq 150 \wedge \text{감독이름} = \text{'봉준호'}}(\text{영화}))$$

## [예제] 관계대수 (ans)

-- 10. 다음 관계 대수식을 SQL 문법으로 작성하시오

$$\pi_{\text{제목, 평점}}(\sigma_{\text{상영시간} \geq 150 \wedge \text{감독이름} = \text{'봉준호'}}(\text{영화}))$$

```
SELECT  제목, 평점
FROM    영화
WHERE   상영시간 >= 150 AND 감독이름 = '봉준호';
```

## [예제] 하위질의(객관식)

-- 62. 과일 테이블에 대한 SQL 문 내의 비교 조건을 실행한 결과가 참인 것은?

①

```
SELECT *  
FROM 과일  
WHERE 21 NOT IN (SELECT 과일코드 FROM 과일);
```

②

```
SELECT *  
FROM 과일  
WHERE 19 < ANY (SELECT 과일코드 FROM 과일);
```

③

```
SELECT *  
FROM 과일  
WHERE 15 < ALL (SELECT 과일코드 FROM 과일);
```

④

```
SELECT *  
FROM 과일  
WHERE 19 =ALL (SELECT 과일코드 FROM 과일);
```

과일

과일코드	과일명
10	오렌지
15	키위
19	파인애플

## [예제] 하위질의(객관식) (ans)

-- 62. 과일 테이블에 대한 SQL 문 내의 비교 조건을 실행한 결과가 참인 것은?

①

```
SELECT *  
FROM 과일  
WHERE 21 NOT IN (SELECT 과일코드 FROM 과일);
```

②

```
SELECT *  
FROM 과일  
WHERE 19 < ANY (SELECT 과일코드 FROM 과일);
```

③

```
SELECT *  
FROM 과일  
WHERE 15 < ALL (SELECT 과일코드 FROM 과일);
```

④

```
SELECT *  
FROM 과일  
WHERE 19 =ALL (SELECT 과일코드 FROM 과일);
```

과일

과일코드	과일명
10	오렌지
15	키위
19	파인애플

## [예제] 하위질의(객관식)

06. 다음 인사, 차량 테이블을 이용하여 SQL 문을 수행했을 경우의 결과는?

SELECT 종류

FROM 차량

WHERE 사원번호 = (select 사원번호  
from 인사  
where 성명='오형우');

인사

사원번호	성명	소속
25	김미순	총무과
56	박이준	자재과
23	이형주	영업과
43	오형우	교육과

차량

사원번호	종류
25	A
56	B
23	C
43	D

- ① 43
- ② 56
- ③ C
- ④ D

## [예제] 중복(DISTINCT) (객관식)

-- 20. student 테이블은 50개의 튜플이 정의되어 있으며  
-- S-AGE 열의 값은 정수값으로 되어 있다.

-- S-AGE 값이 18인 튜플이 10개,  
--           19인 튜플이 35개,  
--           20인 튜플이 5개일 경우,

-- 다음 SQL문의 실행결과값을 각각 작성하시오

**SELECT DISTINCT S-AGE FROM student;**

**SELECT DISTINCT S-AGE FROM student WHERE S-AGE > 19;**

① 50, 40   ② 50, 5   ③ 3, 5   ④ 3, 1



## [예제] 집계함수

-- 26. student 테이블에 독일어과 학생 50명, 중국어과 학생 30명,  
-- 영어영문학과 학생 50명의 정보가 저장되어 있을 때,  
-- 다음 SQL 문의 실행결과는?  
-- (단, dept 필드는 학과명)

-- 1)

```
SELECT COUNT(dept) FROM student;
```

-- 2)

```
SELECT COUNT(DISTINCT dept) FROM student;
```

## [실습] 관계대수 (ans)

〈사원〉

사원	직급
김형석	대리
홍영선	과장
류기선	부장
김현천	이사

〈직원〉

사원	직급
신원섭	이사
이성호	대리
홍영선	과장
류기선	부장

사원 명단이 사원, 직원 테이블에 저장되어 있다  
두 테이블을 통합하는 SQL문법을 작성하시오  
같은 레코드가 두 번 나오지 않도록 한다

-- 답안

SELECT 사원, 직급 FROM 사원

UNION

SELECT 사원, 직급 FROM 직원

# [Quiz] SELECT

(조인, 하위질의)

담당교수: 김희숙  
(jasmin11@hanmail.net)

## [실습] [Quiz] 의사, 환자

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

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

- 의사 테이블을 생성하는 SQL문을 작성하시오.
- <조건>
- 1-1) 소속 속성의 값을 입력하지 않으면 자동으로 내과가 지정
- 1-2) 근무연수는 1년 이상 50년 이하 범위의 값을 가지도록 지정

- 2) 환자 테이블을 생성하는 SQL문을 작성하시오
- <조건>
- 2-1) 이름 속성은 널 값이 허용되지 않도록 지정
- 2-2) 담당의사 속성을 의사 테이블의 의사번호 속성을 참조하는 외래키 지정

- 3) 다음 SQL문법을 작성하시오
- 3-1) D001 의사가 담당하고
- 나이가 30세 이상인 환자의 환자번호와 환자이름을 검색하라
- 3-2) 소속별로 의사수와 평균 근무연수를 검색하라
- 3-3) 김용욱 환자를 담당하는 의사의
- 의사이름과 소속, 근무연수를 검색하라

## [실습] [Quiz] 의사, 환자 (답안)

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

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

- 의사 테이블을 생성하는 SQL문을 작성하시오.
- <조건>
- 1-1) 소속 속성의 값을 입력하지 않으면 자동으로 내과가 지정
- 1-2) 근무연수는 1년 이상 50년 이하 범위의 값을 가지도록 지정

```
CREATE TABLE 의사 (  
    의사번호      char(4)      NOT NULL ,  
    의사이름      varchar(20) ,  
    소속          varchar(30)  DEFAULT '내과' ,  
    근무연수      int          CHECK (근무연수 >= 1 AND 근무연수 <= 50),  
    PRIMARY KEY(의사번호)  
);
```

## [실습] [Quiz] 의사, 환자 (답안)

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

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

- 2) 환자 테이블을 생성하는 SQL문을 작성하시오
- <조건>
- 2-1) 이름 속성은 널 값이 허용되지 않도록 지정
- 2-2) 담당의사 속성을 의사 테이블의 의사번호 속성을 참조하는 외래키 지정

```
CREATE TABLE 환자 (  
    환자번호      char(4)      NOT NULL ,  
    환자이름      varchar(20)  NOT NULL ,  
    나이          int ,  
    담당의사      char(4)      ,  
    PRIMARY KEY(환자번호) ,  
    FOREIGN KEY(담당의사) REFERENCES 의사(의사번호)  
);
```

## [실습] [Quiz] 의사, 환자

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

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

-- 3) 다음 SQL문법을 작성하시오

- 3-1) D001 의사가 담당하고
- 나이가 30세 이상인 환자의 환자번호와 환자이름을 검색하라
- 3-2) 소속별로 의사수와 평균 근무연수를 검색하라
- 3-3) 김용욱 환자를 담당하는 의사의
- 의사이름과 소속, 근무연수를 검색하라
- 3-3-1) (하위질의 사용)
- 3-3-2) (조인 사용)

## [실습] [Quiz] 의사, 환자 (답안)

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

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

```
-- 3. 하위 질의
SELECT
FROM
WHERE ... ( SELECT ...
              FROM ...
              WHERE ... )
```

- 3-3) 김용욱 환자를 담당하는 의사의 의사이름과 소속, 근무연수를 검색하라
- (하위질의 사용)

```
SELECT 의사.의사이름, 의사.소속, 의사.근무연수
FROM 의사
WHERE 의사번호 IN ( SELECT 담당의사
                    FROM 환자
                    WHERE 환자이름 = '김용욱'
                    );
```

- (조인 사용) 김용욱 환자를 담당하는 의사의 의사이름과 소속, 근무연수
- ```
SELECT 의사.의사이름, 의사.소속, 의사.근무연수
FROM 의사, 환자
WHERE 환자.환자이름 = '김용욱'
      AND 의사.의사번호 = 환자.담당의사;
```



## [실습] [Quiz] 의사, 환자

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

- 4) 다음 SQL문법을 작성하시오
- 4-1) 의사이름과 담당 환자수를 검색하라

환자 테이블

| <u>환자번호</u> | 환자이름 | 나이 | 담당의사 |
|-------------|------|----|------|
| P001        | 오우진  | 31 | D002 |
| P002        | 채광주  | 50 | D001 |
| P003        | 김용욱  | 43 | D003 |

의사 테이블

| <u>의사번호</u> | 의사이름 | 소속   | 근무연수 |
|-------------|------|------|------|
| D001        | 정지영  | 내과   | 5    |
| D002        | 김선주  | 피부과  | 10   |
| D003        | 정성호  | 정형외과 | 15   |

## [실습] [Quiz] 의사, 환자 (답안)

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

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   |

-- 2. 인 라인 뷰(inline view)

```
SELECT
FROM      ( SELECT ...
            FROM ...
            WHERE ...
            ) as T
```

WHERE

-- 인라인뷰) **의사이름과 담당 환자수를 검색**하라

```
SELECT 의사이름, T.담당환자수
FROM  의사 D , ( SELECT 담당의사, COUNT(*) as "담당환자수"
                  FROM 환자
                  GROUP BY 담당의사
                ) as T
```

WHERE D.의사번호 = T.담당의사;

-- 조인) **의사이름과 담당 환자수를 검색**하라

```
SELECT 의사이름, COUNT(*) as "담당환자수"
FROM  의사, 환자
WHERE 의사.의사번호 = 환자.담당의사
GROUP BY 의사이름;
```

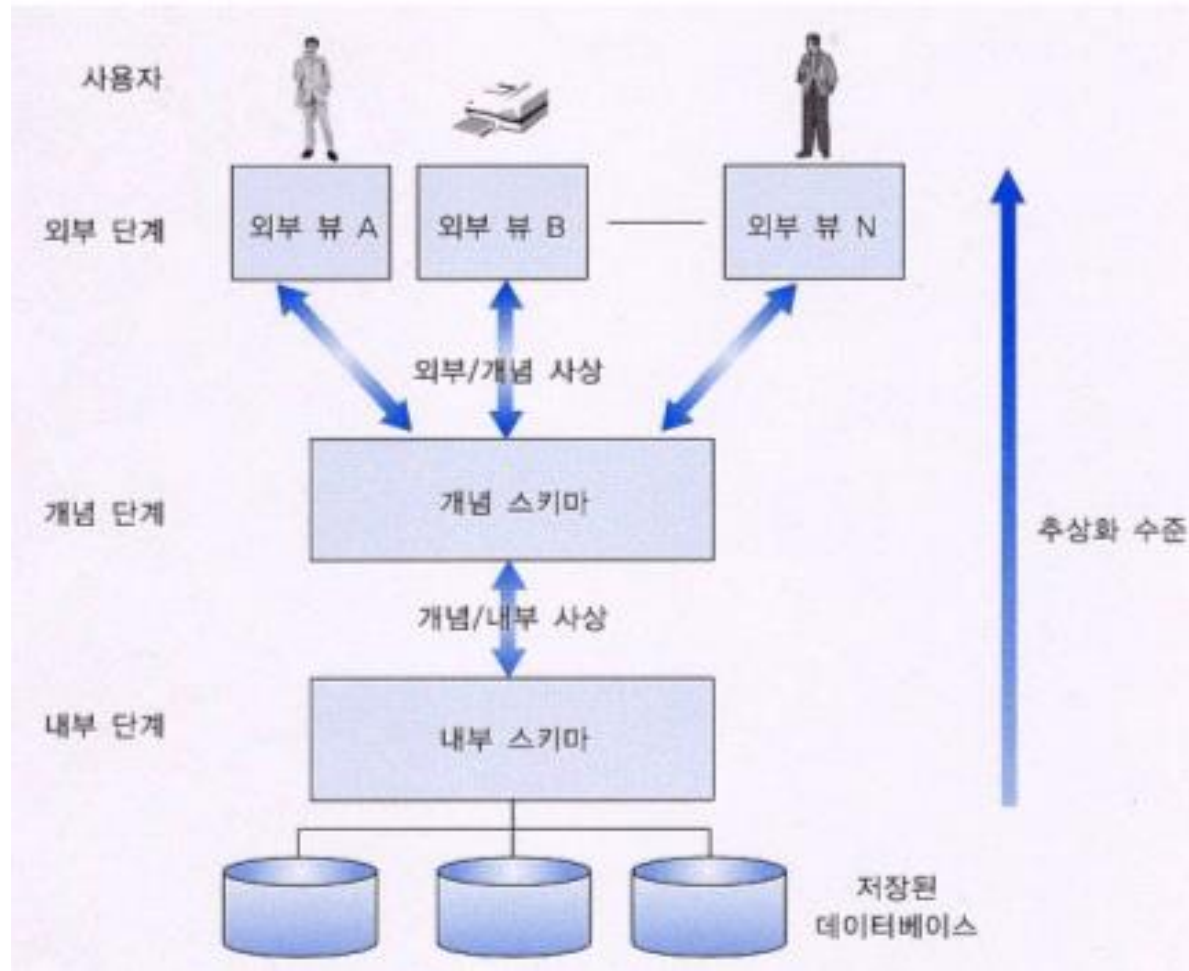
# SQL: 뷰(view)

14주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# [요약] 3단계 구조

ANSI/SPARC 아키텍처 (스키마 3단계 구조)



(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)



## 02 데이터베이스의 구조

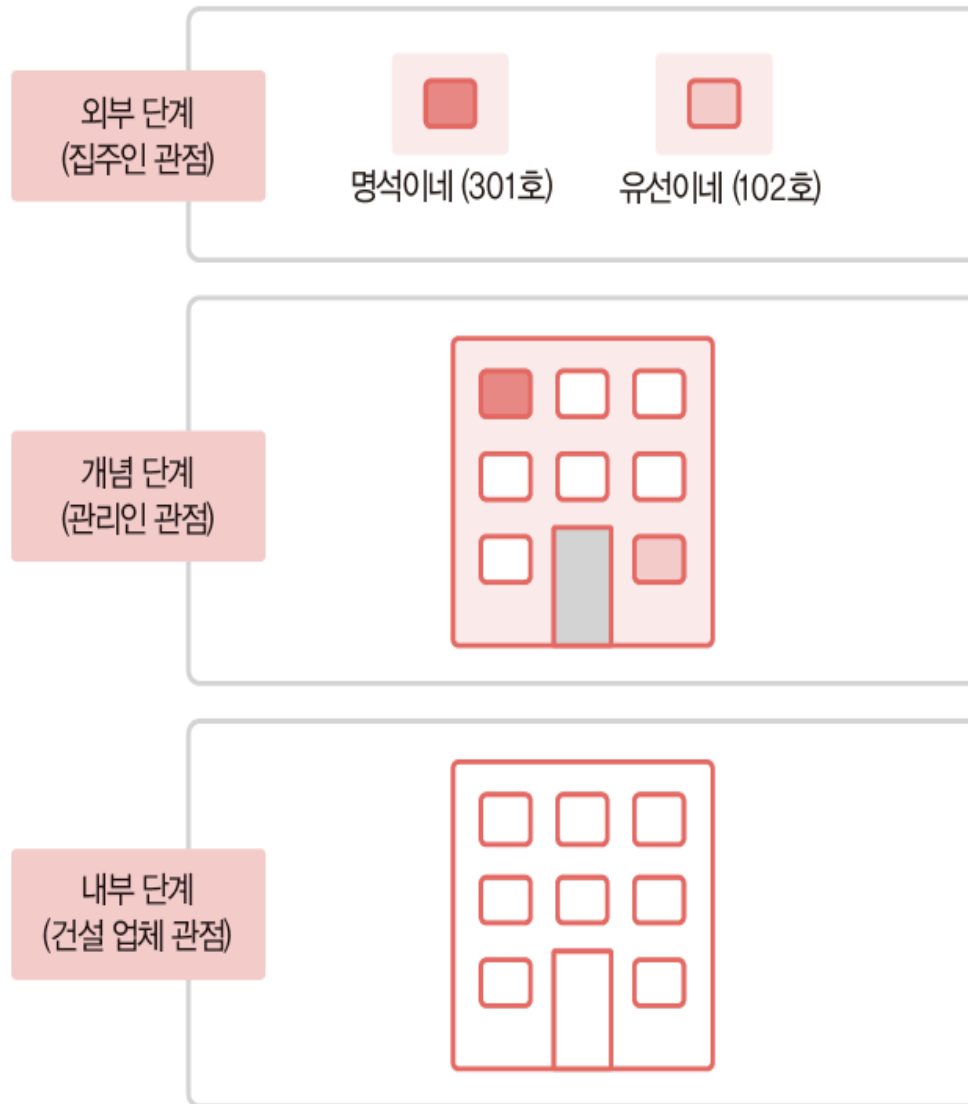


그림 3-3 3단계 데이터베이스 구조의 개념



그림 3-4 3단계 데이터베이스 구조의 이해

# [실습] 고객,주문,제품

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

데이터베이스이름: handb

## 고객, 제품, 주문 테이블 생성

고객 테이블

| 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 | 김용욱  | 22   | silver | 학생  | 0    |
| melon  | 성원용  | 35   | gold   | 회사원 | 5000 |
| peach  | 오형준  | NULL | silver | 의사  | 300  |
| pear   | 채광주  | 31   | silver | 회사원 | 500  |

주문 테이블

| 주문번호 | 주문고객   | 주문제품 | 수량 | 배송지      | 주문일자       |
|------|--------|------|----|----------|------------|
| o01  | apple  | p03  | 10 | 서울시 마포구  | 2019-01-01 |
| o02  | melon  | p01  | 5  | 인천시 계양구  | 2019-01-10 |
| o03  | banana | p06  | 45 | 경기도 부천시  | 2019-01-11 |
| o04  | carrot | p02  | 8  | 부산시 금정구  | 2019-02-01 |
| o05  | melon  | p06  | 36 | 경기도 용인시  | 2019-02-20 |
| o06  | banana | p01  | 19 | 충청북도 보은군 | 2019-03-02 |
| o07  | apple  | p03  | 22 | 서울시 영등포구 | 2019-03-15 |
| o08  | pear   | p02  | 50 | 강원도 춘천시  | 2019-04-10 |
| o09  | banana | p04  | 15 | 전라남도 목포시 | 2019-04-11 |
| o10  | carrot | p03  | 20 | 경기도 안양시  | 2019-05-22 |

제품 테이블

| 제품번호 | 제품명   | 재고량  | 단가   | 제조업체 |
|------|-------|------|------|------|
| p01  | 그냥만두  | 5000 | 4500 | 대한식품 |
| p02  | 매운짬면  | 2500 | 5500 | 민국푸드 |
| p03  | 쿵떡파이  | 3600 | 2600 | 한빛제과 |
| p04  | 맛난초콜릿 | 1250 | 2500 | 한빛제과 |
| p05  | 얼큰라면  | 2200 | 1200 | 대한식품 |
| p06  | 통통우동  | 1000 | 1550 | 민국푸드 |
| p07  | 달콤비스킷 | 1650 | 1500 | 한빛제과 |

# [요약] 뷰(view) 생성



- 뷰(view):
  - 가상의 테이블(다른 테이블을 기반으로 만들어진 가상의 테이블)
  - 장점: **사용의 편의성, 보안**

```
-- 뷰 정의(뷰 생성)
CREATE VIEW 뷰이름
as
SELECT
FROM
[with check option]
```

❖ 뷰 정의할 때, ORDER BY 사용 불가능

```
DROP TABLE 테이블명;
```

```
DROP VIEW 뷰이름;
```

❖ WITH CHECK OPTION:

뷰 삽입/수정할 때,  
뷰의 정의 조건을 위반하면 수행되지 않도록 하는  
제약조건



# [요약] 뷰(view) 삽입/수정/삭제

-- 뷰 삽입/수정/삭제: 실제 테이블에 반영

❖ 뷰 삽입/수정/삭제가 되지 않는 경우:

1. 기본 테이블의 기본키가 누락된 뷰
2. 집계 함수로 작성된 뷰
3. DISTINCT 키워드 포함한 뷰
4. GROUP BY 문법 포함한 뷰
5. 조인으로 작성된 뷰



## [실습] 3단계 구조(외부스키마, 개념스키마, 내부스키마)

[실습] 뷰(schema\_view-ex.sql)

```
1  -- (schema_view-ex.sql)
2  -- 3단계 스키마 구조
3  • use studydb;
4
5  • drop table 가입고객;
6
7  • CREATE TABLE 가입고객 (
8      번호 int NOT NULL,
9      이름 char(10),
10     성별 char(2),
11     나이 int,
12     직업 char(10),
13     주소 char(20),
14     연락처 char(20),
15     PRIMARY KEY(번호)
16 );
17
18 • INSERT INTO 가입고객 VALUES (1, '홍길동', '남', 20, '학생', '서울', NULL),
19                                (2, '임꺽정', '남', 22, '학생', '인천', '010-1111-1111'),
20                                (3, '신아로미', '여', 20, '학생', '서울', '010-2222-2222')
21 ;
22
23 • select * from 가입고객;
```

| Result Grid                             |      |      |      |      |      |               |  |
|-----------------------------------------|------|------|------|------|------|---------------|--|
| Filter Rows:                            |      |      |      |      |      |               |  |
| Edit: Export/Import: Wrap Cell Content: |      |      |      |      |      |               |  |
| 번호                                      | 이름   | 성별   | 나이   | 직업   | 주소   | 연락처           |  |
| 1                                       | 홍길동  | 남    | 20   | 학생   | 서울   | NULL          |  |
| 2                                       | 임꺽정  | 남    | 22   | 학생   | 인천   | 010-1111-1111 |  |
| 3                                       | 신아로미 | 여    | 20   | 학생   | 서울   | 010-2222-2222 |  |
| *                                       | NULL | NULL | NULL | NULL | NULL | NULL          |  |

가입고객 4 x

```
-- 3단계 스키마 구조
use studydb;
```

```
drop table 가입고객;
```

```
CREATE TABLE 가입고객 (
    번호 int NOT NULL,
    이름 char(10),
    성별 char(2),
    나이 int,
    직업 char(10),
    주소 char(20),
    연락처 char(20),
    PRIMARY KEY(번호)
);
```

```
INSERT INTO 가입고객 VALUES (1, '홍길동', '남', 20, '학생', '서울', NULL),
                              (2, '임꺽정', '남', 22, '학생', '인천', '010-1111-1111'),
                              (3, '신아로미', '여', 20, '학생', '서울', '010-2222-2222')
;
```

```
select * from 가입고객;
```

```
-- 가입고객(번호,이름,성별,나이,직업,주소,연락처)
-- 고객분석팀(성별,나이,직업)
-- 상품배송팀(고객번호,고객이름,주소,연락처)
```



-- 뷰 생성

```
CREATE VIEW 고객분석팀
as
```

```
    SELECT 성별, 나이, 직업
    FROM   가입고객;
```

```
select * from 고객분석팀;
```

```
25 -- 가입고객(번호,이름,성별,나이,직업,주소,연락처)
26 -- 고객분석팀(성별,나이,직업)
27 -- 상품배송팀(고객번호,고객이름,주소,연락처)
28
29 -- 뷰 생성
30 • CREATE VIEW 고객분석팀
31   as
32     SELECT 성별, 나이, 직업
33     FROM   가입고객;
34
35 • select * from 고객분석팀;
```

| Result Grid                                                                                                                                                                                        |    |    |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|----|
| Filter Rows: <input type="text"/>                                                                                                                                                                  |    |    |    |
| Export:  Wrap Cell Content:  |    |    |    |
|                                                                                                                                                                                                    | 성별 | 나이 | 직업 |
| ▶                                                                                                                                                                                                  | 남  | 20 | 학생 |
|                                                                                                                                                                                                    | 남  | 22 | 학생 |
|                                                                                                                                                                                                    | 여  | 20 | 학생 |

```
-- 가입고객(번호,이름,성별,나이,직업,주소,연락처)
-- 고객분석팀(성별,나이,직업)
-- 상품배송팀(고객번호,고객이름,주소,연락처)
```

-- 뷰 생성

CREATE VIEW 상품배송팀(고객번호,고객이름,주소,연락처)

as

```
SELECT 번호, 이름, 주소, 연락처
FROM   가입고객;
```

select \* from 상품배송팀;

```
36
37
38
39
40
41
42 • CREATE VIEW 상품배송팀(고객번호,고객이름,주소,연락처)
43   as
44     SELECT 번호, 이름, 주소, 연락처
45     FROM   가입고객;
46
47 • select * from 상품배송팀;
```

Result Grid

|     | 고객<br>번호 | 고객<br>이름 | 주소 | 연락처           |
|-----|----------|----------|----|---------------|
| ▶ 1 |          | 홍길동      | 서울 | NULL          |
| 2   |          | 임꺽정      | 인천 | 010-1111-1111 |
| 3   |          | 신아로미     | 서울 | 010-2222-2222 |

# [예제] 뷰 생성(CREATE VIEW)

- (수정) 예제7-55 고객 테이블에서 등급이 vip인 고객의 고객아이디, 고객이름, 나이, 등급으로 구성된
- 뷰를 우수고객이라는 이름으로 생성해보자
- 우수고객 뷰의 모든 내용을 검색해보자

```
CREATE VIEW 우수고객(고객아이디, 고객이름, 나이, 등급)
as
    SELECT  고객아이디, 고객이름, 나이, 등급
    FROM    고객
    WHERE   등급 = 'vip'
WITH CHECK OPTION;
```

## [실습] (view) 뷰 생성

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

```
-- 우수고객(고객아이디, 고객이름, 나이, 등급)
```

```
-- 뷰 생성
```

```
CREATE VIEW 우수고객(고객아이디, 고객이름, 나이, 등급)
as
```

```
    SELECT 고객아이디, 고객이름, 나이, 등급
    FROM    고객
    WHERE   등급 = 'vip'
```

```
WITH CHECK OPTION;
```

```
select * from 우수고객;
```

```
-- 뷰 필드 생략 가능
```

```
CREATE VIEW 우수고객
```

```
as
```

```
    SELECT 고객아이디, 고객이름, 나이, 등급
```

## [실습] 뷰(view-ex.txt)

```
3  -- 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)
4  -- 제품(제품번호, 제품명, 재고량, 단가, 제조업체)
5  -- 주문(주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자)
6
7  -- 우수고객(고객아이디, 고객이름, 나이, 등급)
8
9  -- (수정) 예제7-55 고객 테이블에서 등급이 vip인 고객의
10 -- |고객아이디, 고객이름, 나이, 등급으로 구성된
11 -- 뷰를 우수고객이라는 이름으로 생성해보자
12 -- 우수고객 뷰의 모든 내용을 검색해보자
13 • drop view 우수고객;
14
15 • CREATE VIEW 우수고객(고객아이디, 고객이름, 나이, 등급)
16 as
17     SELECT  고객아이디, 고객이름, 나이, 등급
18     FROM    고객
19     WHERE   등급 = 'vip'
20     WITH CHECK OPTION;
21
22 • select * from 우수고객;
```

| Result Grid                       |      |    |     |
|-----------------------------------|------|----|-----|
| Filter Rows: <input type="text"/> |      |    |     |
| Export:  Wrap Cell Content:       |      |    |     |
| 고객아이디                             | 고객이름 | 나이 | 등급  |
| ▶ banana                          | 김선우  | 25 | vip |



## [실습] (view) WITH CHECK OPTION

[실습] 뷰(view-ex.txt)

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

```
-- 우수고객(고객아이디, 고객이름, 나이, 등급)
```

```
-- 뷰 생성
```

```
CREATE VIEW 우수고객(고객아이디, 고객이름, 나이, 등급)
as
```

```
    SELECT 고객아이디, 고객이름, 나이, 등급
    FROM    고객
    WHERE   등급 = 'vip'
```

```
WITH CHECK OPTION;
```

```
select * from 우수고객;
```

```
-- WITH CHECK OPTION
```

```
뷰로 작성된 조건에 만족하지 않는 경우,
입력, 수정, 삭제 불가능
```

```
31 -- 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)
32 -- 제품(제품번호, 제품명, 재고량, 단가, 제조업체)
33 -- 주문(주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자)
34
35 -- 우수고객(고객아이디, 고객이름, 나이, 등급)
```

```
37 • select * from 고객;
```

```
39 • select * from 우수고객;
```

```
41 -- 뷰에 입력/수정/삭제 시도(with check option 설정 경우)
42 • insert into 우수고객(고객아이디, 고객이름, 나이, 등급)
43   values('shinmi', '신아로미', 20, 'gold');
```

```
45 • update 우수고객
46   set   등급 = 'gold'
47   where 고객아이디 = 'banana';
```

```
49 • delete
50   from   우수고객
51   where  고객아이디 = 'banana';
```

참조무결성 제약조건 위배

# [예제] 뷰 생성(CREATE VIEW)

- 예제7-56 제품 테이블에서 제조업체별 제품수로 구성된 뷰를
- 업체별제품수라는 이름으로 생성해보자
- 업체별제품수 뷰의 모든 내용을 검색해보자

CREATE VIEW 업체별제품수(제조업체, 제품수)

as

SELECT 제조업체, COUNT(\*)

FROM 제품

GROUP BY 제조업체

## [실습] (view) 집계함수

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

-- 업체별제품수(제조업체, 제품수)

-- 뷰 생성

CREATE VIEW 업체별제품수(제조업체, 제품수)

as

SELECT 제조업체, COUNT(\*)  
 FROM 제품  
 GROUP BY 제조업체

select \* from 업체별제품수;

-- 뷰 필드 생략 불가능

CREATE VIEW 업체별제품수(제조업체, 제품수)

as

SELECT 제조업체, COUNT(\*)

## [실습] 뷰(view-ex.txt)

```

3  -- 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)
4  -- 제품(제품번호, 제품명, 재고량, 단가, 제조업체)
5  -- 주문(주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자)
6
7  • select * from 제품;
8
9  -- 예제7-56 제품 테이블에서 제조업체별 제품수로 구성된 뷰를
10 -- 업체별제품수라는 이름으로 생성해보자
11 -- 업체별제품수 뷰의 모든 내용을 검색해보자
12
13 • CREATE VIEW 업체별제품수(제조업체, 제품수)
14 as
15   SELECT 제조업체, COUNT(*)
16   FROM   제품
17   GROUP BY 제조업체;
18
19 • select * from 업체별제품수;
  
```

| 제조업체 | 제품수 |
|------|-----|
| 대한식품 | 2   |
| 민국푸드 | 2   |
| 한빛제과 | 3   |

| 제품번호 | 제품명   | 재고량  | 단가   | 제조업체 |
|------|-------|------|------|------|
| p01  | 그냥만두  | 5000 | 4500 | 대한식품 |
| p02  | 매운짜면  | 2500 | 5500 | 민국푸드 |
| p03  | 콩떡파이  | 3600 | 2600 | 한빛제과 |
| p04  | 맛난초콜렛 | 1250 | 2500 | 한빛제과 |
| p05  | 얼큰라면  | 2200 | 1200 | 대한식품 |
| p06  | 통통우동  | 1000 | 1550 | 민국푸드 |
| p07  | 달콤비스킷 | 1650 | 1500 | 한빛제과 |
| NULL | NULL  | NULL | NULL | NULL |



# [예제] 뷰 조회(SELECT)

-- 예제7-57 우수고객 뷰에서 나이가 25세 이상인 고객에 대한 모든 내용을 검색해보자

```
SELECT *  
FROM    우수고객  
WHERE   나이 >= 25;
```

## [실습] (view) 뷰 조회

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

```
-- 뷰 조회
SELECT *
FROM 우수고객
WHERE 나이 >= 25;
```

```
select 고객아이디, 고객이름, 나이, 등급
from 고객
where 등급 = 'vip'
AND 나이 >= 25;
```

```
-- 뷰 조회
뷰에 SELECT : 내부적으로 기본테이블에 대한
SELECT 문으로 변환하여 수행
```

## [실습] 뷰(view-ex.txt)

```
3 -- 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)
4 -- 제품(제품번호, 제품명, 재고량, 단가, 제조업체)
5 -- 주문(주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자)
6
7 -- 예제7-57 우수고객 뷰에서 나이가 25세 이상인 고객에 대한 모든 내용을 검색해보자
8
9 • SELECT *
10 FROM 우수고객
11 WHERE 나이 >= 25;
12
13 • select 고객아이디, 고객이름, 나이, 등급
14 from 고객
15 where 등급 = 'vip'
16 AND 나이 >= 25;
```

| Result Grid                             |      |      |      |    |     |  |
|-----------------------------------------|------|------|------|----|-----|--|
| Filter Rows:                            |      |      |      |    |     |  |
| Edit: Export/Import: Wrap Cell Content: |      |      |      |    |     |  |
| 고객아이디                                   | 고객이름 | 나이   | 등급   | 직업 | 적립금 |  |
| ▶ banana                                | 김선우  | 25   | vip  |    |     |  |
| * NULL                                  | NULL | NULL | NULL |    |     |  |

| 고객아이디   | 고객이름 | 나이   | 등급     | 직업   | 적립금  |  |
|---------|------|------|--------|------|------|--|
| ▶ apple | 정소화  | 20   | gold   | 학생   | 1000 |  |
| banana  | 김선우  | 25   | vip    | 간호사  | 2500 |  |
| carrot  | 고명석  | 28   | gold   | 교사   | 4500 |  |
| melon   | 성원용  | 35   | gold   | 회사원  | 5000 |  |
| orange  | 김용욱  | 22   | silver | 학생   | 0    |  |
| peach   | 오형준  | NULL | silver | 의사   | 300  |  |
| pear    | 채광주  | 31   | silver | 회사원  | 500  |  |
| * NULL  | NULL | NULL | NULL   | NULL | NULL |  |

## [예제] 뷰 입력

- 예제7-58 제품번호가 p08, 재고량이 1000, 제조업체가 신선식품인 새로운 제품의 정보를
- 제품1 뷰에 삽입해보자.
- 제품1 뷰에 있는 모든 내용을 검색해보자

```
drop view 제품1;
```

```
CREATE VIEW 제품1
```

```
as
```

```
    SELECT  제품번호, 재고량, 제조업체  
    FROM    제품;
```

```
-- 뷰에 삽입
```

```
insert into 제품1
```

```
values('p08',1000,'신선식품');
```

```
select * from 제품1;
```

```
select * from 제품;
```

## [실습] (view) 뷰 입력

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

-- 뷰 생성

**CREATE VIEW 제품1**

**as**

**SELECT** 제품번호, 재고량, 제조업체  
**FROM** 제품;

-- 뷰 입력

**insert into 제품1 values('p08',1000,'신선식품');**

-- 뷰 조회

**select \* from 제품1;**

-- 기본 테이블(base table) 조회

**select \* from 제품;**

-- 뷰 입력

뷰에 INSERT : 뷰에 입력하지 않은 필드는  
내부적으로 기본 테이블에 **널 값**으로  
입력된다

## [실습] 뷰(view-ex.txt)

-- 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)  
-- 제품(제품번호, 제품명, 재고량, 단가, 제조업체)  
-- 주문(주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자)  
  
-- 예제7-58 제품번호가 p08, 재고량이 1000, 제조업체가 신선식품인  
-- 새로운 제품의 정보를 제품1 뷰에 삽입해보자.  
-- 제품1 뷰에 있는 모든 내용을 검색해보자

**drop view 제품1;**

**CREATE VIEW 제품1**

**as**

**SELECT** 제품번호, 재고량, 제조업체  
**FROM** 제품;

**insert into 제품1 values('p08',1000,'신선식품');**

**select \* from 제품1;**

**select \* from 제품;**

| 제품<br>번호 | 재고량  | 제조<br>업체 |
|----------|------|----------|
| p01      | 5000 | 대한식품     |
| p02      | 2500 | 민국푸드     |
| p03      | 3600 | 한빛제과     |
| p04      | 1250 | 한빛제과     |
| p05      | 2200 | 대한식품     |
| p06      | 1000 | 민국푸드     |
| p07      | 1650 | 한빛제과     |
| p08      | 1000 | 신선식품     |

| 제품<br>번호 | 제품명   | 재고량  | 단<br>가 | 제조<br>업체 |
|----------|-------|------|--------|----------|
| p01      | 그냥만두  | 5000 | 4500   | 대한식품     |
| p02      | 매운쫄면  | 2500 | 5500   | 민국푸드     |
| p03      | 쿵덕파이  | 3600 | 2600   | 한빛제과     |
| p04      | 맛난초콜렛 | 1250 | 2500   | 한빛제과     |
| p05      | 얼큰라면  | 2200 | 1200   | 대한식품     |
| p06      | 통통우동  | 1000 | 1550   | 민국푸드     |
| p07      | 할금비스켓 | 1650 | 1500   | 한빛제과     |
| p08      | NULL  | 1000 | NULL   | 신선식품     |

# [예제] 뷰 조작 불가능한 경우

- 기본키 제외한 필드를 제품2 뷰 생성하고
- 제품2 뷰에 삽입해보자.
- 제품2 뷰에 있는 모든 내용을 검색해보자

```
drop view 제품2;
```

```
CREATE VIEW 제품2
```

```
as
```

```
    SELECT  제품명, 재고량, 제조업체  
    FROM    제품;
```

```
-- 뷰에 삽입
```

```
insert into 제품2
```

```
values('사원냉면',1000,'신선식품');
```

```
select * from 제품2;
```

```
select * from 제품;
```

# SQL: DCL(grant, revoke) 권한관리

14주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# [실습] MySQL

- 1단계: 데이터베이스 생성(스키마 생성)
- 사용할 데이터베이스 선택
- 2단계: 테이블 생성
- 3단계: 데이터 입력

```
-- 데이터베이스(testdb) / 테이블 (직원)  
-- 직원(이름,주소,전화번호,연봉)
```

```
-- 뷰  
-- 직원_뷰(이름, 전화번호)
```

```
-- 데이터베이스(kimdb) / 사용자(kim)
```

# [실습] 권한 관리

-- (MySQL)

1단계: 데이터베이스 생성

CREATE DATABASE testdb

DEFAULT CHARACTER SET utf8mb4 ;

-- 사용할 데이터베이스 선택

USE testdb;

# (root)

# 데이터베이스 생성

testdb

테이블 생성: 직원



# [실습] 권한 관리

# (root)

# 사용자 생성:

kim (kimdb)

-- (MySQL)

# (root)

CREATE DATABASE kimdb

DEFAULT CHARACTER SET 'utf8mb4' ;

-- (MySQL)

# (root)

# 사용자 생성: kim 생성한다

# [실습] 사용자 생성

# (root)

# 사용자 생성:

kim (kimdb)

lee (leedb)



Local instance MySQL80

Users and Privileges

User Accounts

| User             | From Host |
|------------------|-----------|
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| newuser          | %         |
| node             | localhost |
| root             | localhost |
| study            | %         |

Details for account newuser@%

Login Account Limits Administrative Roles Schema Privileges

Login Name:

kim

You may create multiple accounts with the same name to connect from different hosts.

Authentication Type:

Standard

For the standard password and/or host based authentication, select 'Standard'.

Limit to Hosts Matching:

localhost

% and \_ wildcards may be used

Password:

\*\*\*

Type a password to reset it.

Weak password.

Confirm Password:

\*\*\*

Enter password again to confirm.

Expire Password

Add Account

Delete

Refresh

Revert

Apply

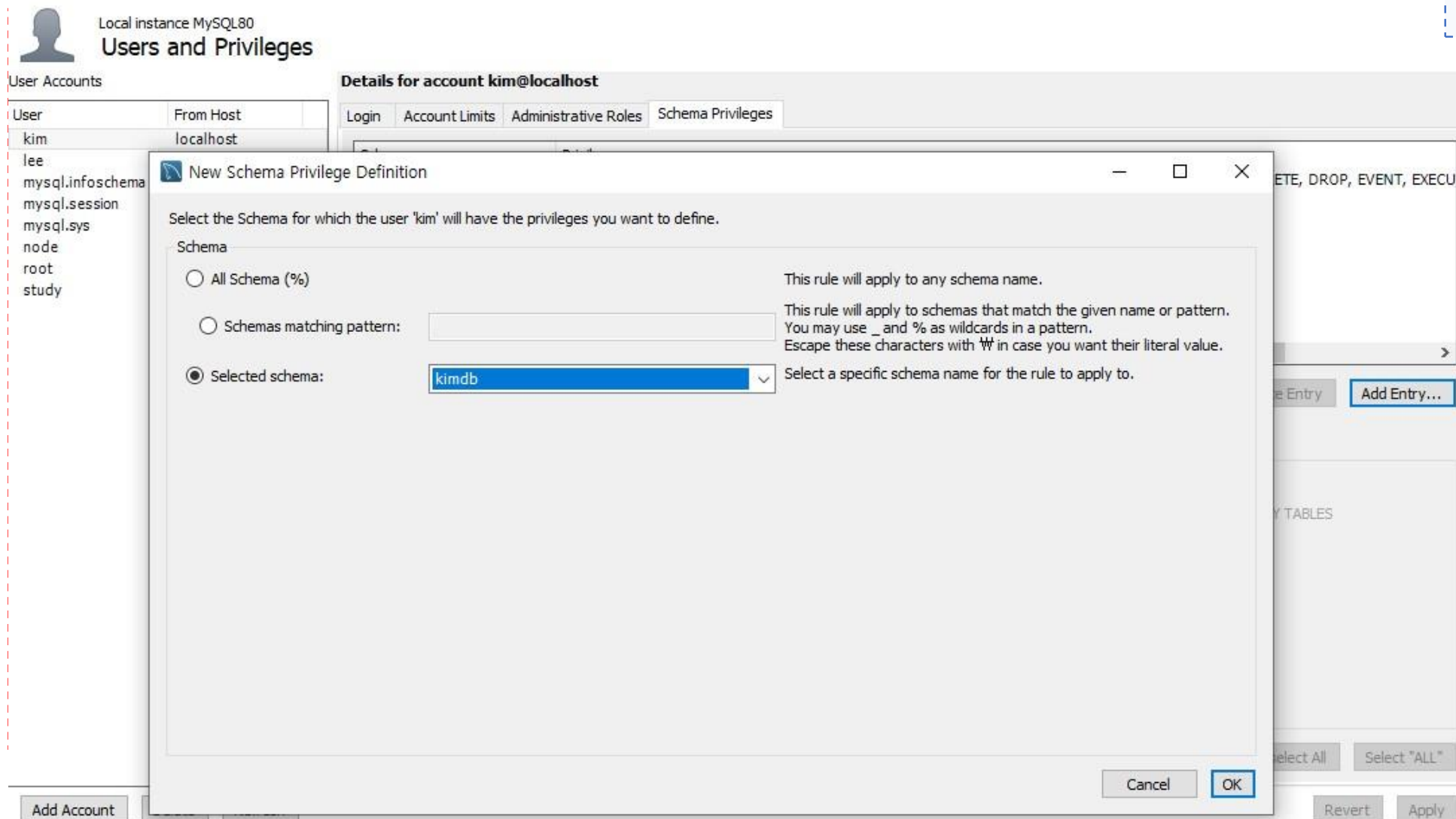
# [실습] 사용자 생성

# (root)

# 사용자 생성:

kim (kimdb)

lee (leedb)



# [실습] 뷰(view) 생성

-- (MySQL)

DROP TABLE if exists 직원;

-- 직원(이름,주소,전화번호,연봉)

CREATE TABLE 직원 (

이름            VARCHAR(20)        NOT NULL ,

주소            VARCHAR(100) ,

전화번호       CHAR(13)            NOT NULL ,

연봉            INT                    DEFAULT 0 ,

PRIMARY KEY(이름)

);

# [실습] 뷰(view) 생성

-- (MySQL)

-- 직원(이름,주소,전화번호,연봉)

INSERT INTO 직원 VALUES( ' 우태하','서울시 서초구','010-1111-1111',1000);

INSERT INTO 직원 VALUES('김선우','서울시 구로구','010-2222-2222',2000);

INSERT INTO 직원 VALUES('이영지','서울시 마포구','010-3333-3333',3000);

INSERT INTO 직원 VALUES('유희정','서울시 마포구','010-4444-4444',4000);

INSERT INTO 직원 VALUES('오형준','부산시 연산구','010-5555-5555',5000);

INSERT INTO 직원 VALUES('고진수','서울시 구로구','010-6666-6666',6000);

select \* from 직원;

# [실습] 뷰

# (root) -> kim  
# WITH GRANT OPTION

-- (MySQL)

# (root)

-- (MySQL) safe mode 해제

SET SQL\_SAFE\_UPDATES = 0;

# 테이블 생성: 직원

# 뷰 생성: 직원\_뷰

# (root) -> kim 에게 모든 권한 부여(직원\_뷰)

GRANT ALL PRIVILEGES on testdb.직원\_뷰 TO 'kim'@'localhost' WITH GRANT OPTION;

# [실습] 뷰

# kim

#

-- (MySQL)

# (kim)

테이블: testdb.직원\_뷰(root) (모든 권한)

-- (오류): 권한 없음

select \* from testdb.직원;

-- (ok)

select \* from testdb.직원\_뷰;

-- 직원\_뷰(이름,전화번호)

insert into testdb.직원\_뷰(이름,전화번호) values('오주원','010-9999-9999');