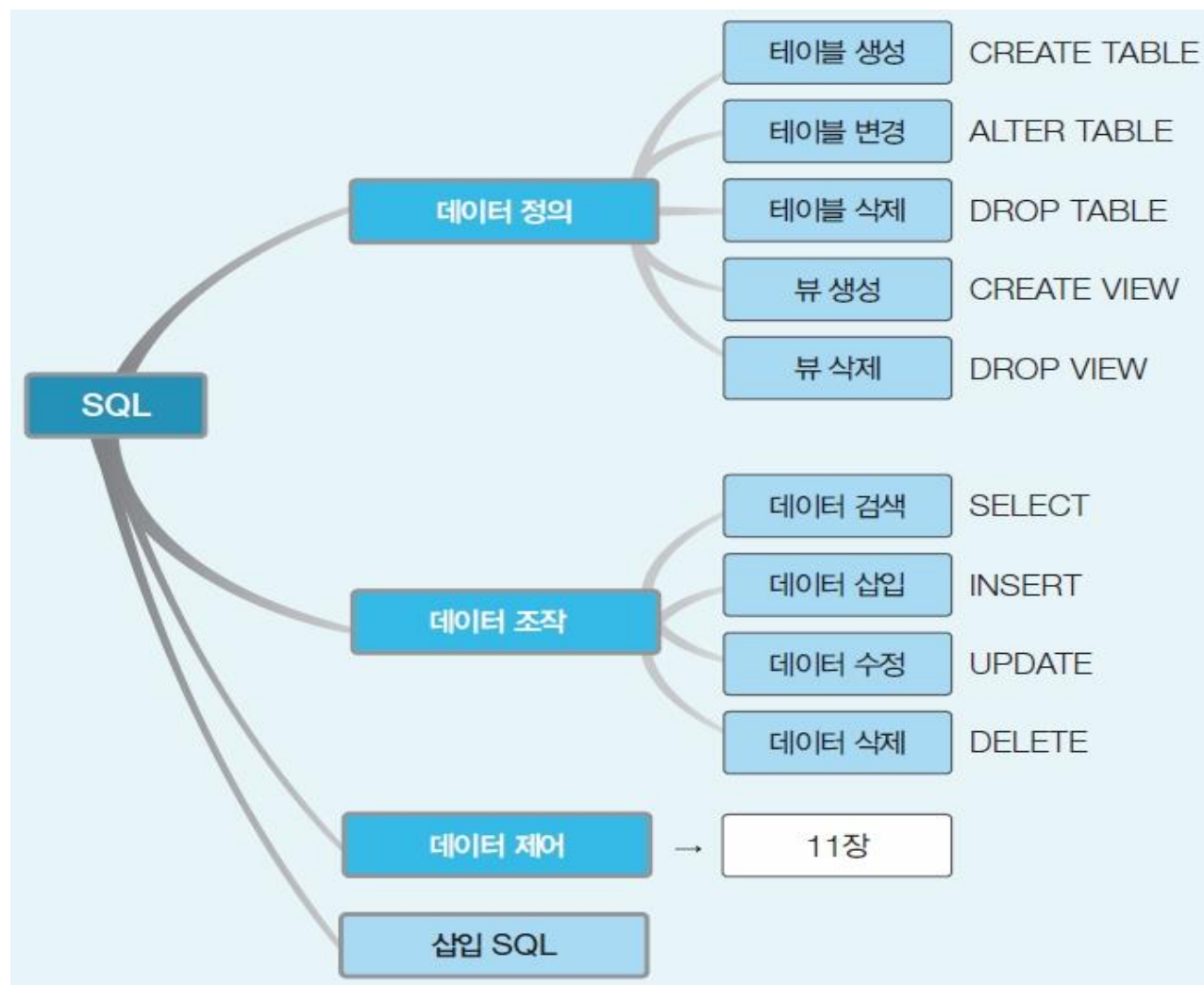


# 데이터베이스관리

9주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# [요약] SQL



(출처: 데이터베이스개론, 김연희, 한빛아카데미)  
김희숙

# [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

담당교수: 김희숙  
(jasmin11@hanmail.net)

# [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

-- [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
1	인사부
2	연구부
3	홍보부

레코드 추가

4	영업부
---	-----

필드 추가

부서위치
서울
대전
서울

# [실습] 필드 추가/데이터 추가

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

```
-- [Quiz 1-01] (부서)
-- 부서(부서번호, 부서이름)

-- 1) 부서 테이블을 생성하라(CREATE)
-- 2) 부서 테이블에 데이터 입력하라(INSERT)
-- 부서번호는 4, 부서이름이 영업부

-- 3) 데이터 조회하라
select * from 부서;

-- 4) 부서 테이블에 필드 추가하라(ALTER)
-- 필드명 부서위치, 데이터 형식 varchar(10)

-- 5) 테이블 구조 조회하라(DESC)
desc 부서;
```

```
-- 2) 부서 테이블에 데이터 입력하라(INSERT)
-- 부서번호는 4, 부서이름이 영업부

-- 3) 데이터 조회하라
select * from 부서;

-- 4) 부서 테이블에 필드 추가하라(ALTER)
-- 필드명 부서위치, 데이터 형식 varchar(10)

-- 5) 테이블 구조 조회하라(DESC)
desc 부서;
```

# [실습] 필드 추가/데이터 추가

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

- 6) 부서 테이블 데이터 수정하라(UPDATE)
- 부서번호 1인 레코드의 부서위치 필드는 서울
- 부서번호 2인 레코드의 부서위치 필드는 대전
- 부서번호 3인 레코드의 부서위치 필드는 서울

-- 7) 데이터 조회하라  
select \* from 부서;

-- 8) 테이블 구조 조회하라(DESC)  
desc 부서;

-- 9) 부서 테이블 삭제하라(drop)

-- 6) 부서 테이블 데이터 수정하라(UPDATE)

-- SAFE\_UPDATE 모드 해제  
SET SQL\_SAFE\_UPDATES = 0;

UPDATE 부서

SET  
WHERE

UPDATE  
SET  
WHERE

UPDATE  
SET  
WHERE



# [실습] 필드 추가/데이터 추가 (ans)

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

-- SAFE\_UPDATE 모드 해제  
SET SQL\_SAFE\_UPDATES = 0;

-- [Quiz 1-01] (부서)  
-- 부서(부서번호, 부서이름)  
  
-- 1) 부서 테이블을 생성하라(CREATE)  
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');
```

-- 2) 부서 테이블에 데이터 입력하라(INSERT)  
-- 부서번호는 4, 부서이름이 영업부  
insert into 부서 values(4,'영업부');

-- 4) 부서 테이블에 필드 추가하라(ALTER)  
-- 필드명 부서위치, 데이터 형식 varchar(10)  
ALTER TABLE 부서  
ADD 부서위치 varchar(10);

-- 6) 부서 테이블 데이터 수정하라(UPDATE)  
-- 부서번호 1인 레코드의 부서위치 필드는 서울

```
UPDATE 부서  
SET 부서위치 = '서울'  
WHERE 부서번호 = 1;
```

# [실습] 데이터 조작(테이블 2개)

[실습 2-01] (테이블 2개) 부서, 사원 테이블에 대하여 다음을 실습하시오

부서

부서번호	부서이름
1	인사부
2	연구부
3	홍보부

사원

사원번호	사원이름	소속부서
1	홍길동	3
2	임꺽정	1
3	차명석	1

외래키

- 부서(부서번호 pk, 부서이름)
- 사원(사원번호 pk, 사원이름, 소속부서 fk)

```
-- (MySQL) safe mode  
SET SQL_SAFE_UPDATES = 0;
```

-- 다음 문제를 실습하시오

- 홍보부를 삭제하라는 실행가능한가?
- 1) 부서번호 3인 홍보부를 삭제하라

## [실습] 데이터 조작(테이블 2개)

-- [실습 2-01] 부서, 사원

-- 부서(부서번호 pk, 부서이름)  
-- 사원(사원번호 pk, 사원이름, 소속부서 fk)

drop table if exists 사원;  
drop table if exists 부서;

```
CREATE TABLE 부서 (
    부서번호 INT NOT NULL ,
    부서이름 VARCHAR(10) ,
    PRIMARY KEY (부서번호)
);
```

부서

부서번호	부서이름
1	인사부
2	연구부
3	홍보부

사원

사원번호	사원이름	소속부서
1	홍길동	3
2	임꺽정	1
3	차명석	1

외래키

-- 다음 문제를 실습하시오

-- 부서번호 3인 홍보부를 삭제하라

```
CREATE TABLE 사원 (
    사원번호 INT NOT NULL ,
    사원이름 VARCHAR(10) ,
    소속부서 INT ,
    PRIMARY KEY (사원번호) ,
    FOREIGN KEY (소속부서)
    REFERENCES 부서(부서번호)
);
```

-- 부서(부서번호 pk, 부서이름)  
-- 사원(사원번호 pk, 사원이름, 소속부서 fk)  
insert into 부서 values(1,'인사부');  
insert into 부서 values(2,'연구부');  
insert into 부서 values(3,'홍보부');

insert into 사원 values(1001,'홍길동',3);  
insert into 사원 values(1002,'임꺽정',1);  
insert into 사원 values(1003,'차명석',1);

select \* from 부서;  
select \* from 사원;

김희죽

# [실습] 데이터 조작(테이블 2개) CASCADE

[실습 2-02] (테이블 2개) 부서, 사원 테이블에 대하여 다음을 실습하시오

부서

부서번호	부서이름
1	인사부
2	연구부
3	홍보부

사원

사원번호	사원이름	소속부서
1	홍길동	3
2	임꺽정	1
3	차명석	1

외래키

-- 부서(부서번호 pk, 부서이름)  
-- 사원(사원번호 pk, 사원이름, 소속부서 fk)

-- 부서번호 3인 홍보부를 삭제하라  
DELETE  
FROM 부서  
WHERE 부서번호 = 3;

-- 홍보부를 삭제하라는 실행가능한가?  
-- 1) on delete no action  
-- 2) on delete cascade  
-- 3) on delete set null

**ON DELETE NO ACTION** : 부서 테이블의 튜플을 삭제할 수 없다

**ON DELETE CASCADE** : 사원 테이블에서 홍보부에 근무하는 임꺽정 레코드도 함께 삭제

**ON DELETE SET NULL** : 사원 테이블에서 홍보부에 근무하는 임꺽정 의 소속부서 속성 값을 NULL로 수정

**ON DELETE SET DEFAULT** : 사원 테이블에서 임꺽정 의 소속부서 속성 값을 기본 값으로 변경

## [실습] 데이터 조작(테이블 2개) CASCADE

-- [실습 2-02] 부서, 사원 (CASCADE)

-- 부서(부서번호 pk, 부서이름)  
-- 사원(사원번호 pk, 사원이름, 소속부서 fk)

drop table if exists 사원;  
drop table if exists 부서;

```
CREATE TABLE 부서 (  
    부서번호 INT NOT NULL ,  
    부서이름 VARCHAR(10) ,  
    PRIMARY KEY (부서번호)  
);
```

부서

부서번호	부서이름
1	인사부
2	연구부
3	홍보부

사원

사원번호	사원이름	소속부서
1	홍길동	3
2	임꺽정	1
3	차명석	1

외래키

-- 다음 문제를 실습하시오

-- 부서번호 3인 홍보부를 삭제하라

```
CREATE TABLE 사원 (  
    사원번호 INT NOT NULL ,  
    사원이름 VARCHAR(10) ,  
    소속부서 INT ,  
    PRIMARY KEY (사원번호) ,  
    FOREIGN KEY (소속부서)  
        REFERENCES 부서(부서번호)  
    ON DELETE CASCADE  
);
```

-- 부서(부서번호 pk, 부서이름)  
-- 사원(사원번호 pk, 사원이름, 소속부서 fk)  
insert into 부서 values(1,'인사부');  
insert into 부서 values(2,'연구부');  
insert into 부서 values(3,'홍보부');

insert into 사원 values(1001,'홍길동',3);  
insert into 사원 values(1002,'임꺽정',1);  
insert into 사원 values(1003,'차명석',1);

select \* from 부서;  
select \* from 사원;

김희죽

## [실습] 데이터 조작(테이블 2개) SET NULL

-- [실습 2-02] 부서, 사원 (SET NULL)

-- 부서(부서번호 pk, 부서이름)  
-- 사원(사원번호 pk, 사원이름, 소속부서 fk)

drop table if exists 사원;  
drop table if exists 부서;

```
CREATE TABLE 부서 (
    부서번호 INT NOT NULL ,
    부서이름 VARCHAR(10) ,
    PRIMARY KEY (부서번호)
);
```

부서

부서번호	부서이름
1	인사부
2	연구부
3	홍보부

사원

사원번호	사원이름	소속부서
1	홍길동	3
2	임꺽정	1
3	차명석	1

외래키

-- 다음 문제를 실습하시오

-- 부서번호 3인 홍보부를 삭제하라

```
CREATE TABLE 사원 (
    사원번호 INT NOT NULL ,
    사원이름 VARCHAR(10) ,
    소속부서 INT ,
    PRIMARY KEY (사원번호) ,
    FOREIGN KEY (소속부서)
    REFERENCES 부서(부서번호)
    ON DELETE SET NULL
);
```

-- 부서(부서번호 pk, 부서이름)  
-- 사원(사원번호 pk, 사원이름, 소속부서 fk)  
insert into 부서 values(1,'인사부');  
insert into 부서 values(2,'연구부');  
insert into 부서 values(3,'홍보부');

insert into 사원 values(1001,'홍길동',3);  
insert into 사원 values(1002,'임꺽정',1);  
insert into 사원 values(1003,'차명석',1);

select \* from 부서;  
select \* from 사원;

김희죽

# SQL: SELECT 문법(SELECT 기초)

9주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# 데이터베이스 언어(SQL)

## □ 데이터베이스 언어(SQL)

- ✓ 데이터 정의어(DDL)
- ✓ 데이터 조작어(DML)
- ✓ 데이터 제어어(DCL)

SQL	명령어
DDL	CREATE ALTER DROP
DML	INSERT UPDATE DELETE SELECT
DCL	GRANT REVOKE



# SQL



# 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

# [예제] SELECT (데이터 조회)

```
CREATE TABLE customer (
  cno    char(4)      NOT NULL ,
  cname  varchar(10)  NOT NULL ,
  city   varchar(20) ,
  point  int ,
  CONSTRAINT pk_customer_cno PRIMARY KEY(cno)
);
```

테이블 이름	필드 이름	데이터 형식	NULL 유무	기본키
customer	cno	char(4)	NOT NULL	PK
	cname	varchar(10)	NOT NULL	
	city	varchar(20)		
	point	int		

Result Grid				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

```
insert into customer values('c101','홍길동','서울',500);
```

```
insert into customer values('c102','임꺽정','인천',300);
```

```
insert into customer values('c103','박찬호','안양',800);
```

```
insert into customer values('c204','신동엽','과천',350);
```

```
insert into customer values('c205','정진우','고양',400);
```

```
/* 새 레코드를 추가하고 SELECT문 예제 실습하시오 */
```

```
insert into customer values('c307','정동우','서울', NULL);
```

# [실습1] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

```
-- 고객(고객번호, 고객명, 거주지, 포인트)
```

```
-- 1-1) 테이블의 모든 열을 검색하라
```

```
-- 1-2) 테이블의 모든 열을 검색(필드명 사용)
```

```
-- 1-3) 고객의 고객명, 거주지를 검색하라(테이블의 특정 열을 검색)
```

```
-- 1-4) cname 은 성명, city는 거주지로 출력하라
```

```
-- (화면에 표시되는 열 이름 변경하여 검색)
```

```
-- 1-5) customer 테이블에서 거주지를 검색하라
```

```
-- 1-6) 거주지를 검색하는데 중복 행을 제거하여 한 번씩만 검색하라
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL

# [실습1] SELECT (데이터 조회)

[실습 1]

-- customer(cno, cname, city, point)

-- 1-1) 테이블의 모든 열을 검색하라

```
SELECT *  
FROM customer;
```

[실습] (customer.sql)

Result Grid				
Filter Rows: <input type="text"/>				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL



# [실습1] SELECT (데이터 조회)

[실습 1]

-- customer(cno, cname, city, point)  
-- 1-2) 테이블의 모든 열을 검색(필드명 사용)

```
SELECT cno, cname, city, point  
FROM customer;
```

[실습] (customer.sql)

Result Grid				
Filter Rows: <input type="text"/>				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL

# [실습1] SELECT (데이터 조회)

[실습 1]

- customer(cno, cname, city, point)
- 1-3) 고객의 고객명, 거주지를 검색하라(테이블의 특정 열을 검색)

```
SELECT  cname, city
FROM    customer;
```

[실습] (customer.sql)

Result Grid				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cname	city
▶	홍길동	서울
	임꺽정	인천
	박찬호	안양
	신동엽	과천
	정진우	고양
	정동우	서울

# [실습1] SELECT (데이터 조회)

[실습 1]

- customer(cno, cname, city, point)
- 1-4) cname 은 성명, city는 거주지로 출력하라
- (화면에 표시되는 열 이름 변경하여 검색)

```
SELECT  cname 성명, city 거주지
FROM    customer;
```

[실습] (customer.sql)

Result Grid				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	성명	거주지
▶	홍길동	서울
	임꺽정	인천
	박찬호	안양
	신동엽	과천
	정진우	고양
	정동우	서울



# [실습1] SELECT (데이터 조회)

[실습 1]

-- customer(cno, cname, city, point)  
-- 1-5) customer 테이블에서 거주지를 검색하라

```
SELECT city  
FROM customer;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	city
▶	서울
	인천
	안양
	과천
	고양
	서울

# [실습1] SELECT (데이터 조회)

[실습 1]

-- customer(cno, cname, city, point)

-- 1-6) 거주지를 검색하는데 중복 행을 제거하여 한 번씩만 검색하라

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임걱정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
*	NULL	NULL	NULL	NULL	



```
SELECT DISTINCT city
FROM customer;
```

	city
▶	서울
	인천
	안양
	과천
	고양

# [실습1] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 1]

-- customer(cno, cname, city, point)

-- 1-6) 거주지를 검색하는데 중복 행을 제거하여 한 번씩만 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL

```
SELECT city
FROM customer;
```



```
SELECT DISTINCT city
FROM customer;
```

	city
▶	서울
	인천
	안양
	과천
	고양
	서울

	city
▶	서울
	인천
	안양
	과천
	고양

# [실습2] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

```
-- 고객(고객번호, 고객명, 거주지, 포인트)
```

```
-- 2-1) 고객번호가 c101 인 고객의 모든 정보를 검색하라
```

```
-- 2-2) 포인트가 400 이하인 고객의 모든 정보를 검색하라
```

```
-- 2-3) 거주지가 서울 이면서 포인트가 500 이상인 고객의 이름, 거주지, 포인트를 검색하라
```

```
-- 2-4) 거주지가 서울 이거나 포인트가 500 이상인 고객의 이름, 거주지, 포인트를 검색하라
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임격정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

## ❖ 관계 연산자

> >= = <> <= <

## ❖ 논리 연산자

NOT

AND

OR

# [실습2] SELECT (데이터 조회)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-1) 고객번호가 c101 인 고객의 모든 정보를 검색하라

```
SELECT *  
FROM customer  
WHERE cno = 'c101';
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c101	홍길동	서울	500

# [실습2] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-2) 포인트가 400 이하인 고객의 모든 정보를 검색하라

```
SELECT *  
FROM customer  
WHERE point <= 400;
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c102	임꺽정	인천	300
	c204	신동엽	과천	350
	c205	정진우	고양	400

# [실습2] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-3) 거주지가 서울 이면서 포인트가 500 이상인 고객의 이름, 거주지, 포인트를  
검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT  cname, city, point
FROM    customer
WHERE   city = '서울' AND point >= 500;
```

	cname	city	point
▶	홍길동	서울	500



# [실습2] SELECT (데이터 조회)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-4) 거주지가 서울 이거나 포인트가 500 이상인 고객의 이름, 거주지, 포인트를 검색하라

```
SELECT  cname, city, point
FROM    customer
WHERE   city = '서울' OR point >= 500;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cname	city	point
▶	홍길동	서울	500
	박찬호	안양	800
	정동우	서울	NULL



# [실습2] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

```
-- 고객(고객번호, 고객명, 거주지, 포인트)
```

```
-- 2-5) 포인트가 350 부터 500 사이인 고객이름, 거주지, 포인트를 검색하라
```

```
-- 부등호 사용
```

```
-- BETWEEN ... AND 사용
```

```
-- 2-6) 거주지가 서울 이거나 안양인 고객번호, 이름, 거주지를 검색하라
```

```
-- 부등호 사용
```

```
-- IN 사용
```

```
-- 2-7) 거주지가 서울이 아니거나 안양이 아닌 고객번호, 이름, 거주지를 검색하라
```

```
-- 부등호 사용
```

```
-- NOT IN 사용
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

❖ SELECT 문법

**BETWEEN ... AND**

**IN**

# [실습2] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 2]

- customer(cno, cname, city, point)
- 2-5) 포인트가 350 부터 500 사이인 고객이름, 거주지, 포인트를 검색하라
- 부등호 사용
- BETWEEN ... AND 사용

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT  cname, city, point
FROM    customer
WHERE   point >= 350 AND point <= 500;
```

```
SELECT  cname, city, point
FROM    customer
WHERE   point BETWEEN 250 AND 500;
```

	cname	city	point
▶	홍길동	서울	500
	신동엽	과천	350
	정진우	고양	400

# [실습2] SELECT (데이터 조회)

## [실습 2]

- customer(cno, cname, city, point)
- 2-6) 거주지가 서울 이거나 안양인 고객번호, 이름, 거주지를 검색하라
- 부등호 사용
- IN 사용

```
SELECT cno, cname, city
FROM customer
WHERE city = '서울' OR city = '안양';
```

```
SELECT cno, cname, city
FROM customer
WHERE city IN ('서울', '안양');
```

## [실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city
▶	c101	홍길동	서울
	c103	박찬호	안양
	c307	정동우	서울

# [실습2] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 2]

- customer(cno, cname, city, point)
- 2-6) 거주지가 서울 이거나 안양인 고객번호, 이름, 거주지를 검색하라
- 부등호 사용
- IN 사용

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT cno, cname, city
FROM customer
WHERE city <> '서울' AND city <> '안양';
```

```
SELECT cno, cname, city
FROM customer
WHERE city NOT IN ('서울', '안양');
```

	cno	cname	city
▶	c102	임꺽정	인천
	c204	신동엽	과천
	c205	정진우	고양

# [실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

```
-- 고객(고객번호, 고객명, 거주지, 포인트)
```

```
-- 3-1) 정씨 성을 가진 고객의 모든 열을 검색하라
```

```
-- 3-2) 이름에 '동' 자가 들어가는 고객의 모든 열을 검색하라
```

```
-- 3-3) 이름의 세번째 글자가 '우' 자가 들어가는 고객의 모든 열을 검색하라
```

```
-- 3-4) 성이 홍씨, 박씨, 정씨인 고객을 검색하라
```

```
-- 성이 홍씨, 박씨, 정씨가 아닌 고객을 검색하라
```

```
-- 3-5) 포인트가 없는 고객의 번호, 이름, 포인트를 검색하라
```

```
-- 포인트가 있는 고객의 번호, 이름, 포인트를 검색하라
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임격정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL

❖ 문자열 검색

% 0개 이상 검색

\_ 1개 검색

❖ NULL 값 검색

IS NULL

IS NOT NULL

# [실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

-- customer(cno, cname, city, point)

-- 3-1) 정씨 성을 가진 고객의 모든 열을 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



SELECT \*

FROM customer

WHERE cname LIKE '정%';

	cno	cname	city	point
▶	c205	정진우	고양	400
	c307	정동우	서울	NULL



# [실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

-- customer(cno, cname, city, point)

-- 3-2) 이름에 '동' 자가 들어가는 고객의 모든 열을 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



SELECT \*

FROM customer

WHERE cname LIKE '%동%';

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c204	신동엽	과천	350
	c307	정동우	서울	NULL

# [실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

-- customer(cno, cname, city, point)

-- 3-3) 이름의 세번째 글자가 '우' 자가 들어가는 고객의 모든 열을 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



```
SELECT *  
FROM customer  
WHERE cname LIKE '__우';
```

	cno	cname	city	point
▶	c205	정진우	고양	400
	c307	정동우	서울	NULL



# [실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

- customer(cno, cname, city, point)
- 3-4) 성이 홍씨, 박씨, 정씨인 고객을 검색하라
- 성이 홍씨, 박씨, 정씨가 아닌 고객을 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT *  
FROM customer  
WHERE cname LIKE '홍%' or cname LIKE '박%' or cname LIKE '정%';
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c103	박찬호	안양	800
	c205	정진우	고양	400
	c307	정동우	서울	NULL

```
SELECT *  
FROM customer  
WHERE cname NOT LIKE '홍%' AND cname NOT LIKE '박%' AND cname NOT LIKE '정%';
```

	cno	cname	city	point
▶	c102	임꺽정	인천	300
	c204	신동엽	과천	350

# [실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

- customer(cno, cname, city, point)
- 3-5) 포인트가 없는 고객의 번호, 이름, 포인트를 검색하라
- 포인트가 있는 고객의 번호, 이름, 포인트를 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT cno, cname, point
FROM customer
WHERE point IS NULL;
```

	cno	cname	point
▶	c307	정동우	NULL

```
SELECT cno, cname, point
FROM customer
WHERE point IS NOT NULL;
```

	cno	cname	point
▶	c101	홍길동	500
	c102	임꺽정	300
	c103	박찬호	800
	c204	신동엽	350
	c205	정진우	400

# [요약] SELECT (정렬)

❖ 정렬: **ORDER BY**

**오름차순** (Ascending order)    **ASC**

**내림차순** (Descending order)    **DESC**

**SELECT**

**FROM**

**ORDER BY**

**SELECT**

**FROM**

**WHERE**

**GROUP BY**

**HAVING**

**ORDER BY**

# [실습4] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

```
-- 고객(고객번호, 고객명, 거주지, 포인트)
```

```
-- 4-1) 고객 테이블에서 이름을 오름차순 정렬하라
```

```
-- 4-2) 거주지가 서울인 고객의 모든 데이터를 검색하는데, 이름의 오름차순 정렬하여 출력하라
```

```
-- 4-3) 거주지의 오름차순으로 정렬하고, 거주지가 같으면 포인트의 내림차순으로 정렬하라
```

```
-- 4-4) 포인트가 많은 순으로(내림차순) 먼저 정렬하고, 같은 포인트는 이름의 오름차순으로 정렬하고
```

```
-- 이름이 같으면 거주지의 오름차순으로 정렬하여 검색하라
```

```
-- 4-5) 다음의 의미는?
```

```
SELECT      cno, cname, city, point
```

```
FROM        customer
```

```
ORDER BY 3;
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

# [실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)

-- 4-1) 고객 테이블에서 이름을 오름차순 정렬하라

```
SELECT *  
FROM customer  
ORDER BY cname ASC;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c102	임꺽정	인천	300
	c307	정동우	서울	NULL
	c205	정진우	고양	400
	c101	홍길동	서울	500

# [실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)  
-- 4-2) 거주지가 서울인 고객의 모든 데이터를 검색하는데,  
이름의 오름차순 정렬하여 출력하라

```
SELECT *  
FROM customer  
WHERE city = '서울'  
ORDER BY cname;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c307	정동우	서울	NULL
	c101	홍길동	서울	500

# [실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)

-- 4-3) 거주지의 오름차순으로 정렬하고, 거주지가 같으면  
포인트의 내림차순으로 정렬하라

```
SELECT *  
FROM customer  
ORDER BY city ASC, point DESC;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c205	정진우	고양	400
	c204	신동엽	과천	350
	c101	홍길동	서울	500
	c307	정동우	서울	NULL
	c103	박찬호	안양	800
	c102	임꺽정	인천	300



# [실습4] SELECT (정렬)

[실습 4]

- customer(cno, cname, city, point)
- 4-4) 포인트가 많은 순으로(내림차순) 먼저 정렬하고,
- 같은 포인트는 이름의 오름차순으로 정렬하고
- 이름이 같으면 거주지의 오름차순으로 정렬하여 검색하라

```
SELECT *  
FROM customer  
ORDER BY point DESC, cname ASC, city ASC;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c103	박찬호	안양	800
	c101	홍길동	서울	500
	c205	정진우	고양	400
	c204	신동엽	과천	350
	c102	임꺽정	인천	300
	c307	정동우	서울	NULL



# [실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)

-- 4-5) 다음의 의미는?

```
SELECT    cno, cname, city, point
FROM      customer
ORDER BY 3;
```

```
SELECT    cno, cname, city, point
FROM      customer
ORDER BY 3;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c205	정진우	고양	400
	c204	신동엽	과천	350
	c101	홍길동	서울	500
	c307	정동우	서울	NULL
	c103	박찬호	안양	800
	c102	임꺽정	인천	300