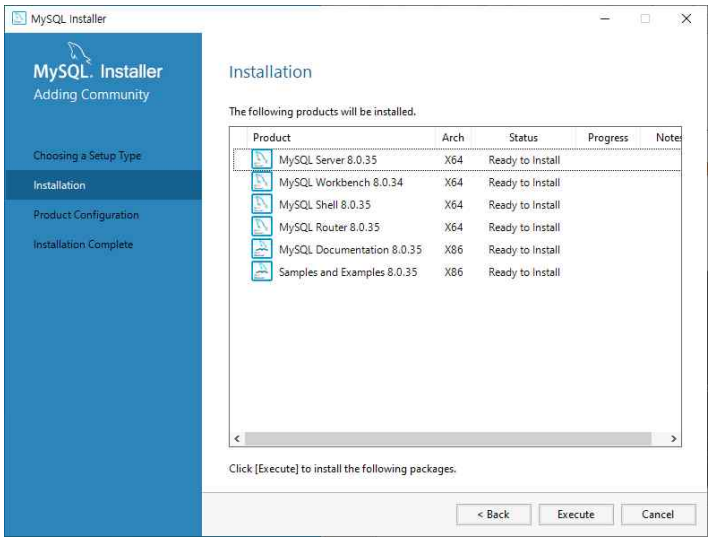
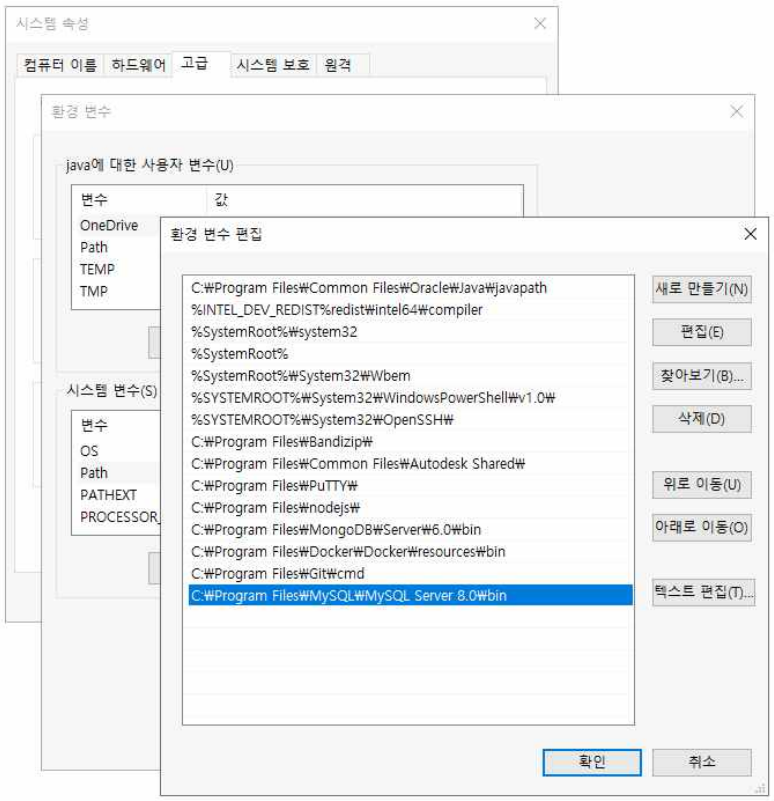
제1장 데이터베이스 설치와 생성

1) MySQL 설치 및 설정

MySQL 설치



환경 변수 설정



2) 데이터베이스 생성 및 제거

데이터베이스 생성

```
CREATE DATABASE 데이터베이스명;
```

데이터베이스 제거

```
DROP DATABASE 데이터베이스명;
```

☞ 실습 1-1. MySQL 접속, 데이터베이스 생성 및 삭제

```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\java> mysql -u root -p
Enter password: ****

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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 `StudyDB`;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| studydb            |
| sys                |
| world              |
+--------------------+
7 rows in set (0.00 sec)

mysql> DROP DATABASE `StudyDB`;
Query OK, 0 rows affected (0.01 sec)
```

3) 관리자 추가

일반 관리자 생성

```
CREATE USER '아이디'@'%' IDENTIFIED BY '비밀번호';
```

권한부여

```
GRANT ALL PRIVILEGES ON 데이터베이스.* TO '아이디'@'%';
```

변경사항 적용

```
FLUSH PRIVILEGES;
```

※ %는 모든 외부 IP 접속을 허용

☞ 실습 1-2. 일반 관리자 생성 및 권한 부여

```
mysql> CREATE USER 'chhak'@'%' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON StudyDB.* TO 'chhak'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

4) 관리자 비밀번호 수정 및 관리자 삭제

관리자 삭제

```
DROP USER '아이디'@'%';
```

☞ 실습 1-3. 사용자 비밀번호 수정 및 삭제

```
mysql> ALTER USER 'chhak'@'%' IDENTIFIED BY 'abc1234';
Query OK, 0 rows affected (0.01 sec)

mysql> DROP USER 'chhak'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

제2장 SQL 기초

1) 테이블 생성 및 제거

테이블 생성

```
CREATE TABLE 테이블명 (칼럼명1 자료형1, 칼럼명2 자료형2, ...);
```

테이블 제거

```
DROP TABLE 테이블명;
DROP TABLE IF EXISTS 테이블명;
```

☞ 실습 2-1. 테이블 생성, 제거

```
mysql> use StudyDB;
Database changedQuery OK

mysql> CREATE TABLE `User1` (
    ->   `uid`  VARCHAR(10),
    ->   `name` VARCHAR(10),
    ->   `hp`   CHAR(13),
    ->   `age`  INT
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> DROP TABLE `User1`;
```

2) 테이블에 데이터 추가

데이터 추가

```
INSERT INTO 테이블명 VALUES (데이터1, 데이터2, 데이터3 ...);
```

칼럼명을 지정해서 데이터 추가

```
INSERT INTO 테이블명 (칼럼명1, 칼럼명2 ... ) VALUES (데이터1, 데이터2 ...);
```

칼럼명을 지정해서 데이터 추가

```
INSERT INTO 테이블명 SET 칼럼명1=데이터1, 칼럼명2=데이터2 ...;
```

☞ 실습 2-2. 데이터 입력

```
mysql> INSERT INTO `User1` VALUES ('A101', '김유신', '010-1234-1111', 25);
mysql> INSERT INTO `User1` VALUES ('A102', '김춘추', '010-1234-2222', 23);
mysql> INSERT INTO `User1` VALUES ('A103', '장보고', '010-1234-3333', 32);
mysql> INSERT INTO `User1` (`uid`, `name`, `age`) VALUES ('A104', '강감찬', 45);
mysql> INSERT INTO `User1` SET `uid`='A105', `name`='이순신', `hp`='010-1234-5555';
```

3) 테이블에 데이터 조회

데이터 조회

```
SELECT 칼럼명1, 칼럼명2 ... FROM 테이블명;
SELECT 칼럼명1, 칼럼명2 ... FROM 테이블명 WHERE 조건;
```

모든 데이터 조회

```
SELECT * FROM 테이블명;
SELECT * FROM 데이터베이스명.테이블명;
```

☞ 실습 2-3. 데이터 조회

```
mysql> SELECT * FROM `User1`;
mysql> SELECT * FROM `User1` WHERE `uid`='A101';
mysql> SELECT * FROM `User1` WHERE `name`='김춘추';
mysql> SELECT * FROM `User1` WHERE `age` < 30;
mysql> SELECT * FROM `User1` WHERE `age` >= 30;
mysql> SELECT `uid`, `name`, `age` FROM `User1`;
```

4) 테이블에 데이터 수정

데이터 수정

```
UPDATE 테이블명 SET 칼럼명1=데이터1, 칼럼명2=데이터2 ...;
```

조건에 일치하는 레코드만 수정

```
UPDATE 테이블명 SET 칼럼명1=데이터1, 칼럼명2=데이터2 ... WHERE 조건;
```

☞ 실습 2-4. 데이터 수정

```
mysql> UPDATE `User1` SET `hp`='010-1234-4444' WHERE `uid`='A104';
mysql> UPDATE `User1` SET `age`=51 WHERE `uid`='A105';
mysql> UPDATE `User1` SET `hp`='010-1234-1001', `age`=27 WHERE `uid`='A101';
```

5) 테이블에 데이터 삭제

데이터 삭제

```
DELETE FROM 테이블명 WHERE 조건;
```

☞ 실습 2-5. 데이터 삭제

```
mysql> DELETE FROM `User1` WHERE `uid`='A101';
mysql> DELETE FROM `User1` WHERE `uid`='A102' AND `age`=25;
mysql> DELETE FROM `User1` WHERE `age` >= 30;
```

6) 테이블 수정

속성(열) 추가

```
ALTER TABLE 테이블명 ADD 속성명 자료형;
```

속성(열) 자료형 변경

```
ALTER TABLE 테이블명 MODIFY 속성명 새로운_자료형;
```

속성(열) 삭제

```
ALTER TABLE 테이블명 DROP COLUMN 속성명;
```

☞ 실습 2-6. 테이블 컬럼 수정

```
mysql> ALTER TABLE `User1` ADD `gender` TINYINT;
mysql> ALTER TABLE `User1` ADD `birth` CHAR(10) AFTER `name`;
mysql> ALTER TABLE `User1` MODIFY `gender` CHAR(1);
mysql> ALTER TABLE `User1` MODIFY `age` TINYINT;
mysql> ALTER TABLE `User1` DROP `gender`;
```

7) 테이블 복사

테이블 복사

```
CREATE TABLE 새로운_테이블명 LIKE 복사할_테이블명;
```

테이블 데이터 복사

```
INSERT INTO 테이블명 SELECT * FROM 복사할_테이블명;
INSERT INTO 테이블명 (칼럼명...) SELECT 칼럼명... FROM 복사할_테이블명;
```

☞ 실습 2-7. 테이블 복사

```
mysql> CREATE TABLE `User1Copy` LIKE `User1`;
mysql> INSERT INTO `User1Copy` SELECT * FROM `User1`;
```

8) 연습문제

☞ 실습 2-8. 아래와 같이 테이블을 생성 후 데이터를 입력하시오.

· 회원 테이블(TblUser)

user_id	user_name	user_hp	user_age	user_addr
p101	김유신	010-1234-1001	25	경남 김해시
p102	김춘추	010-1234-1002	23	경남 경주시
p103	장보고	(NULL)	31	전남 완도군
p104	강감찬	(NULL)	(NULL)	서울시 중구
p105	이순신	010-1234-1005	50	(NULL)

· 제품 테이블(TblProduct)

prod_no	prod_name	prod_price	prod_stock	prod_company	prod_date
1001	냉장고	800,000	25	LG전자	2022-01-06
1002	노트북	1,200,000	120	삼성전자	2022-01-07
1003	모니터	350,000	35	LG전자	2023-01-13
1004	세탁기	1,000,000	80	삼성전자	2021-01-01
1005	컴퓨터	1,500,000	20	삼성전자	2023-10-01
1006	휴대폰	950,000	102	(NULL)	(NULL)

☞ 실습 2-9. 아래 SQL을 실행하시오.

```
> SELECT * FROM `TblUser`;
> SELECT `user_name` FROM `TblUser`;
> SELECT `user_name`, `user_hp` FROM `TblUser`;
> SELECT * FROM `TblUser` WHERE `user_id`='p102';
> SELECT * FROM `TblUser` WHERE `user_id`='p104' OR `user_id`='p105';
> SELECT * FROM `TblUser` WHERE `user_addr`='부산시 금정구';
> SELECT * FROM `TblUser` WHERE `user_age` > 30;
> SELECT * FROM `TblUser` WHERE `user_hp` IS NULL;
> UPDATE `TblUser` SET `user_age`=42 WHERE `user_id`='p104';
> UPDATE `TblUser` SET `user_addr`='부산시 진구' WHERE `user_id`='p105';
> DELETE FROM `TblUser` WHERE `user_id`='p103';

> SELECT * FROM `TblProduct`;
> SELECT `prod_name` FROM `TblProduct`;
> SELECT `prod_name`, `prod_company`, `prod_price` FROM `TblProduct`;
> SELECT * FROM `TblProduct` WHERE `prod_company`='LG전자';
> SELECT * FROM `TblProduct` WHERE `prod_company`='삼성전자';
> UPDATE `TblProduct` SET
                       `prod_company`='삼성전자',
                       `prod_date`='2024-01-01'
                      WHERE
                       `prod_no`=1006;
```

제3장 제약조건

1) 기본키(Primary Key)

☞ 실습 3-1. 기본키 실습하기

```
> CREATE TABLE `User2` (
    `uid`   VARCHAR(10) PRIMARY KEY,
    `name`  VARCHAR(10),
    `birth` CHAR(10),
    `addr`  VARCHAR(50)
    );
```

☞ 실습 3-2. User2 데이터 추가하기

2) 고유키(Unique)

☞ 실습 3-3. 고유키 실습하기

```
> CREATE TABLE `User3` (
    `uid`   VARCHAR(10) PRIMARY KEY,
    `name`  VARCHAR(10),
    `birth` CHAR(10),
    `hp`    CHAR(13) UNIQUE,
    `addr`  VARCHAR(50)
    );
```

☞ 실습 3-4. User3 데이터 추가하기

3) 외래키(Foreign Key)

☞ 실습 3-5. 왜리키 실습하기

```
> CREATE TABLE `Parent` (
    `pid`   VARCHAR(10) PRIMARY KEY,
    `name`  VARCHAR(10),
    `birth` CHAR(10),
    `addr`  VARCHAR(100)
    );

> CREATE TABLE `Child` (
    `cid`    VARCHAR(10) PRIMARY KEY,
    `name`   VARCHAR(10),
    `hp`     CHAR(13) UNIQUE,
    `parent` VARCHAR(10),
    FOREIGN KEY (`parent`) REFERENCES `Parent` (`pid`)
   );
```

☞ 실습 3-6. Parent, Child 데이터 추가하기

4) DEFAULT와 NOT NULL

☞ 실습 3-7

```
> CREATE TABLE `User4`(
    `uid`    VARCHAR(10) PRIMARY KEY,
    `name`   VARCHAR(10) NOT NULL,
    `gender` CHAR(1),
    `age`    INT DEFAULT 1,
    `hp`     CHAR(13) UNIQUE,
    `addr`   VARCHAR(20)
    );
```

☞ 실습 3-8. User4 데이터 추가하기

5) CHECK와 AUTO_INCREMENT

☞ 실습 3-9

```
> CREATE TABLE `User5`(
    `seq`    INT PRIMARY KEY AUTO_INCREMENT,
    `name`   VARCHAR(10) NOT NULL,
    `gender` CHAR(1) CHECK (`gender` IN ('M', 'F')),
    `age`    INT DEFAULT 1 CHECK (`age` > 0 AND `age` < 100),
    `addr`   VARCHAR(20)
    );
```

☞ 실습 3-10. User5 데이터 추가하기

제4장 SQL 고급

1) 실습 테이블 생성

☞ 실습 4-1. 아래 표를 참고하여 테이블을 생성하시오.

· 직원 테이블(Member) 정보

컬럼명(한글)	영문명	데이터유형	길이	NULL허용	기본값
아이디(PK)	uid	문자열	10	X	없음
이름	name	문자열	10	X	없음
휴대폰(UK)	hp	문자열	13	X	없음
직급	pos	문자열	10	X	'사원'
부서번호	dep	숫자	정수	O	NULL
입사일	rdate	날짜시간	날짜시간	X	없음

· 부서 테이블(Department) 정보

컬럼명(한글)	영문명	데이터유형	길이	NULL허용	기본값
부서번호(PK)	depNo	숫자	정수	X	없음
부서명	name	문자열	10	X	없음
부서전화번호	tel	문자열	12	X	없음

· 매출 테이블(Sales) 정보

컬럼명(한글)	영문명	데이터유형	길이	NULL허용	기본값
번호(PK)	seq	숫자	정수	X	AUTO_INCREMENT
직원아이디	uid	문자열	10	X	없음
매출연도	year	년도	년도	X	없음
매출월	month	숫자	정수	X	없음
매출액	sale	숫자	정수	O	NULL

☞ 실습 4-2. 테이블에 아래 그림과 같이 데이터를 입력하시오.

· 직원 테이블(Member)

uid	name	hp	pos	dep	rdate
a101	박혁거세	010-1234-1001	부장	101	2020-11-19 11:39:48
a102	김유신	010-1234-1002	차장	101	2020-11-19 11:39:48
a103	김춘추	010-1234-1003	사원	101	2020-11-19 11:39:48
a104	장보고	010-1234-1004	대리	102	2020-11-19 11:39:48
a105	강감찬	010-1234-1005	과장	102	2020-11-19 11:39:48
a106	이성계	010-1234-1006	차장	103	2020-11-19 11:39:48
a107	정철	010-1234-1007	차장	103	2020-11-19 11:39:48
a108	이순신	010-1234-1008	부장	104	2020-11-19 11:39:48
a109	허균	010-1234-1009	부장	104	2020-11-19 11:39:48
a110	정약용	010-1234-1010	사원	105	2020-11-19 11:39:48
a111	박지원	010-1234-1011	사원	105	2020-11-19 11:39:48

· 매출 테이블(Sales)

seq	uid	year	month	sale
1	a101	2018	1	98,100
2	a102	2018	1	136,000
3	a103	2018	1	80,100
4	a104	2018	1	78,000
5	a105	2018	1	93,000
6	a101	2018	2	23,500
7	a102	2018	2	126,000
8	a103	2018	2	18,500
9	a105	2018	2	19,000
10	a106	2018	2	53,000
11	a101	2019	1	24,000
12	a102	2019	1	109,000
13	a103	2019	1	101,000
14	a104	2019	1	53,500
15	a107	2019	1	24,000
16	a102	2019	2	160,000
17	a103	2019	2	101,000
18	a104	2019	2	43,000
19	a105	2019	2	24,000
20	a106	2019	2	109,000
21	a102	2020	1	201,000
22	a104	2020	1	63,000
23	a105	2020	1	74,000
24	a106	2020	1	122,000
25	a107	2020	1	111,000
26	a102	2020	2	120,000
27	a103	2020	2	93,000
28	a104	2020	2	84,000
29	a105	2020	2	180,000
30	a108	2020	2	76,000

· 부서 테이블(Department)

depNo	name	tel
101	영업1부	051-512-1001
102	영업2부	051-512-1002
103	영업3부	051-512-1003
104	영업4부	051-512-1004
105	영업5부	051-512-1005
106	영업지원부	051-512-1006
107	인사부	051-512-1007

2) 데이터 조회 연산자

조건에 일치하는 투플 조회

```
SELECT 칼럼명1, 칼럼명2 ... FROM 테이블명 WHERE 조건식;
```

☞ 실습 4-3

```
> SELECT * FROM `Member` WHERE `name`='김유신';
> SELECT * FROM `Member` WHERE `pos`='차장' AND dep=101;
> SELECT * FROM `Member` WHERE `pos`='차장' OR  dep=101;
> SELECT * FROM `Member` WHERE `name` != '김춘추';
> SELECT * FROM `Member` WHERE `name` <> '김춘추';
> SELECT * FROM `Member` WHERE `pos`='사원' OR `pos`='대리';
> SELECT * FROM `Member` WHERE `pos` IN('사원', '대리');
> SELECT * FROM `Member` WHERE `dep` IN(101, 102, 103);
> SELECT * FROM `Member` WHERE `name` LIKE '%신';
> SELECT * FROM `Member` WHERE `name` LIKE '김%';
> SELECT * FROM `Member` WHERE `name` LIKE '김_ _';
> SELECT * FROM `Member` WHERE `name` LIKE '_성_';
> SELECT * FROM `Member` WHERE `name` LIKE '정_';
> SELECT * FROM `Sales` WHERE `sale` > 50000;
> SELECT * FROM `Sales` WHERE `sale` >= 50000 AND `sale` < 100000 AND `month`=1;
> SELECT * FROM `Sales` WHERE `sale` BETWEEN 50000 AND 100000;
> SELECT * FROM `Sales` WHERE `sale` NOT BETWEEN 50000 AND 100000;
> SELECT * FROM `Sales` WHERE `year` IN(2020);
> SELECT * FROM `Sales` WHERE `month` IN(1, 2);
```

3) 정렬과 LIMIT

오름/내림차순으로 정렬

```
SELECT 칼럼명1, 칼럼명2 ... FROM 테이블명 ORDER BY 기준_칼럼명 (ASC / DESC);
```

☞ 실습 4-4

```
> SELECT * FROM `Sales` ORDER BY `sale`;
> SELECT * FROM `Sales` ORDER BY `sale` ASC;
> SELECT * FROM `Sales` ORDER BY `sale` DESC;
> SELECT * FROM `Member` ORDER BY `name`;
> SELECT * FROM `Member` ORDER BY `name` DESC;
> SELECT * FROM `Member` ORDER BY `rdate` ASC;
> SELECT * FROM `Sales` WHERE `sale` > 50000 ORDER BY `sale` DESC;
> SELECT * FROM `Sales`
  WHERE `sale` > 50000
  ORDER BY `year`, `month`, `sale` DESC;
```

투플 개수를 제한

```
SELECT 칼럼명1, 칼럼명2 ... FROM 테이블명 LIMIT 제한할_투플;
```

☞ 실습 4-5

```
> SELECT * FROM Sales LIMIT 3;
> SELECT * FROM Sales LIMIT 0, 3;
> SELECT * FROM Sales LIMIT 1, 2;
> SELECT * FROM Sales LIMIT 5, 3;
> SELECT * FROM Sales ORDER BY `sale` DESC LIMIT 3, 5;
> SELECT * FROM Sales WHERE `sale` < 50000 ORDER BY `sale` DESC LIMIT 3;
> SELECT * FROM Sales
  WHERE `sale` > 50000
  ORDER BY `year` DESC, `month`, `sale` DESC
  LIMIT 5;
```

4) SQL 내장함수

칼럼명을 별칭으로 사용하기

```
SELECT 칼럼명 AS 별칭 FROM 테이블명;
```

☞ 실습 4-6

```
> SELECT SUM(sale) AS `합계` FROM `Sales`;
> SELECT AVG(sale) AS `평균` FROM `Sales`;
> SELECT MAX(sale) AS `최대값` FROM `Sales`;
> SELECT MIN(sale) AS `최소값` FROM `Sales`;
> SELECT CEILING(1.2);
> SELECT CEILING(1.8);
> SELECT FLOOR(1.2);
> SELECT FLOOR(1.8);
> SELECT ROUND(1.2);
> SELECT ROUND(1.8);
> SELECT RAND();
> SELECT CEILING(RAND() * 10);
> SELECT COUNT(sale) AS `갯수` FROM `Sales`;
> SELECT COUNT(*) AS `갯수` FROM `Sales`;

> SELECT LEFT('HelloWorld', 5);
> SELECT RIGHT('HelloWorld', 5);
> SELECT SUBSTRING('HelloWorld', 6, 5);
> SELECT CONCAT('Hello', 'World');
> SELECT CONCAT(`uid`, `name`, `hp`) FROM `member` WHERE `uid`='a108';
> SELECT CURDATE();
> SELECT CURTIME();
> SELECT NOW();
> INSERT INTO `Member` VALUES ('a112', '유관순', '010-1234-1012', '대리', 107, NOW());
```

☞ 실습 4-7. 2018년 1월 매출의 총합을 구하시오.
☞ 실습 4-8. 2019년 2월에 5만원 이상 매출에 대한 총합과 평균을 구하시오.
☞ 실습 4-9. 2020년 전체 매출 가운데 최저, 최고, 매출을 구하시오.

5) 그룹화

그룹별로 조회

```
SELECT 칼럼명1, 칼럼명2 ... FROM 테이블명 GROUP BY 그룹화_칼럼명;
```

☞ 실습 4-10

```
> SELECT * FROM `Sales` GROUP BY `uid`;
> SELECT * FROM `Sales` GROUP BY `year`;
> SELECT * FROM `Sales` GROUP BY `uid`, `year`;
> SELECT `uid`, COUNT(*) AS `건수` FROM `Sales GROUP BY `uid`;
> SELECT `uid`, SUM(sale) AS `합계` FROM `Sales` GROUP BY `uid`;
> SELECT `uid`, AVG(sale) AS `평균` FROM `Sales` GROUP BY `uid`;

> SELECT `uid`, `year`, SUM(sale) AS `합계`
  FROM `Sales`
  GROUP BY `uid`, `year`;

> SELECT `uid`, `year`, SUM(sale) AS `합계`
  FROM `Sales`
  GROUP BY `uid`, `year`
  ORDER BY `year` ASC, `합계` DESC;

> SELECT `uid`, `year`, SUM(sale) AS `합계`
  FROM `Sales`
  WHERE `sale` >= 50000
  GROUP BY `uid`, `year`
  ORDER BY `합계` DESC;
```

그룹화에 검색조건 설정

```
SELECT 칼럼명1, 칼럼명2 ... FROM 테이블명 GROUP BY 그룹화_칼럼명 HAVING 조건;
```

☞ 실습 4-11

```
> SELECT `uid`, SUM(sale) AS `합계` FROM `Sales`
  GROUP BY `uid`
  HAVING `합계` >= 200000;

> SELECT `uid`, `year`, SUM(sale) AS `합계`
  FROM `Sales`
  WHERE `sale` >= 100000
  GROUP BY `uid`, `year`
  HAVING `합계` >= 200000
  ORDER BY `합계` DESC;
```

6) 테이블 합치기

2개 이상의 테이블에서 투플 조회하기(중복 데이터 제외)

```
SELECT 칼럼명 FROM 테이블명1 UNION SELECT 칼럼명 FROM 테이블명2;
```

☞ 실습 4-12

```
> CREATE TABLE `Sales2` LIKE `Sales`;
> INSERT INTO `Sales2` SELECT * FROM `Sales`;
> UPDATE `Sales2` SET `year` = `year` + 3;

> SELECT * FROM `Sales` UNION SELECT * FROM `Sales2`;
> SELECT * FROM `Sales` WHERE `sale` >= 100000
  UNION
  SELECT * FROM `Sales2` WHERE `sale` >= 100000;

> SELECT `uid`, `year`, `sale` FROM Sales
  UNION
  SELECT `uid`, `year`, `sale` FROM Sales2;

> SELECT `uid`, `year`, SUM(sale) AS `합계`
  FROM `Sales`
  GROUP BY `uid`, `year`
  UNION
  SELECT `uid`, `year`, SUM(sale) AS `합계`
  FROM `Sales2`
  GROUP BY `uid`, `year`
  ORDER BY `year` ASC, `합계` DESC;
```

7) 테이블 결합

내부조인(INNER JOIN)

```
SELECT 칼럼 FROM 테이블1 INNER JOIN 테이블2 ON 테이블1.칼럼=테이블2.칼럼;
```

※ INNER 생략 가능

☞ 실습 4-13

```
> SELECT * FROM `Sales` INNER JOIN `Member` ON `Sales`.uid = `Member`.uid;
> SELECT * FROM `Member` INNER JOIN `Department` ON `Member`.dep = `Department`.depNo;

> SELECT * FROM `Sales` AS a JOIN `Member` AS b  ON  a.uid = b.uid;
> SELECT * FROM `Member` AS a JOIN `Department` AS b  ON  a.dep = b.depNo;

> SELECT * FROM `Sales` AS a, `Member` AS b WHERE a.uid = b.uid;
> SELECT * FROM `Member` AS a, `Department` AS b WHERE a.dep = b.depNo;

> SELECT a.`seq`, a.`uid`, `sale`, `name`, `pos` FROM `Sales` AS a
  JOIN `Member` AS b  ON  a.`uid` = b.`uid`;

> SELECT a.`seq`, a.`uid`, `sale`, `name`, `pos` FROM `Sales` AS a
  JOIN `Member` AS b  USING (uid);

> SELECT a.`seq`, a.`uid`, `sale`, `name`, `pos` FROM `Sales` AS a
  JOIN `Member` AS b  ON  a.`uid` = b.`uid`
  WHERE `sale` >= 100000;

> SELECT a.`seq`, a.`uid`, b.`name`, b.`pos`, `year`, SUM(`sale`) AS `합계` FROM `Sales` AS a
     JOIN `Member` AS b  ON a.uid = b.uid
     GROUP BY a.`uid`, a.`year`  HAVING `합계` >= 100000
     ORDER BY a.`year` ASC, `합계` DESC;

> SELECT * FROM `Sales` AS a
  JOIN `Member` AS b  ON a.uid = b.uid
  JOIN `Department` AS c  ON b.dep = c.depNo;

> SELECT a.`seq`, a.`uid`, a.`sale`, b.`name`, b.`pos`, c.`name` FROM `Sales` AS a
  JOIN `Member` AS b  ON a.uid = b.uid
  JOIN `Department` AS c ON b.dep = c.depNo;

> SELECT a.`seq`, a.`uid`, a.`sale`, b.`name`, b.`pos`, c.`name` FROM `Sales` AS a
  JOIN `Member` AS b  ON a.uid = b.uid
  JOIN `Department` AS c  ON b.dep = c.depNo
  WHERE `sale` > 100000
  ORDER BY `sale` DESC;
```

외부조인(LEFT, RIGHT JOIN)

```
SELECT 칼럼 FROM 테이블1 (LEFT | RIGHT) JOIN 테이블2 ON 테이블1.칼럼=테이블2.칼럼;
```

※ 외부조인은 결합하는 테이블의 한쪽(LEFT, RIGHT)의 모든 튜플 조회

☞ 실습 4-14

```
> SELECT * FROM `Sales` AS a LEFT JOIN `Member` AS b ON a.uid = b.uid;
> SELECT * FROM `Sales` AS a RIGHT JOIN `Member` AS b ON a.uid = b.uid;
> SELECT a.seq, a.uid, `sale`, `name`, `pos`  FROM Sales AS a
  LEFT JOIN Member AS b USING(uid);

> SELECT a.seq, a.uid, `sale`, `name`, `pos`  FROM Sales AS a
  RIGHT JOIN Member AS b USING(uid);
```

☞ 실습 4-15. 모든 직원의 아이디, 이름, 직급, 부서명을 조회하시오.

☞ 실습 4-16. '김유신' 직원의 2019년도 매출의 합을 조회하시오.

☞ 실습 4-17. 2019년 50,000이상 매출에 대해 직원별 매출의 합이 100,000원 이상인 직원이름, 부서명, 직급, 년도, 매출 합을 조회하시오. 단, 매출 합이 큰 순서부터 정렬

제5장 데이터베이스 개체

1) 인덱스(Index)

☞ 실습 5-1. 인덱스 조회

```
> SHOW INDEX FROM `User1`;
> SHOW INDEX FROM `User2`;
> SHOW INDEX FROM `User3`;
```

☞ 실습 5-2. 인덱스 생성 및 적용

```
> CREATE INDEX `idx_user1_uid` ON `User1`(`uid`);
> ANALYZE TABLE `User1`;
```

☞ 실습 5-3. 인덱스 삭제

```
> DROP INDEX idx_user1_uid ON `User1`;
```

2) 뷰(View)

☞ 실습 5-4. 뷰 생성

```
> CREATE VIEW `vw_user1` AS (SELECT `name`, `hp`, `age` FROM `User1`);
> CREATE VIEW `vw_user4_age_under30` AS (SELECT * FROM `User4` WHERE `age` < 30);
> CREATE VIEW `vw_member_with_sales` AS (
    SELECT
      a.`uid`   AS `직원아이디`,
      b.`name`  AS `직원이름`,
      b.`pos`   AS `직급`,
      c.`name`  AS `부서명`,
      a.`year`  AS `매출년도`,
      a.`month` AS `월`,
      a.`sale`  AS `매출액`
    FROM `Sales` AS a
    JOIN `Member` AS b ON a.uid = b.uid
    JOIN `Department` AS c ON b.dep = c.depNo
    );
```

☞ 실습 5-5. 뷰 조회

```
> SELECT * FROM `vw_user1`;
> SELECT * FROM `vw_user2_age_under30`;
```

☞ 실습 5-6. 뷰 삭제

```
> DROP VIEW `vw_user1`;
> DROP VIEW `vw_user4_age_under30`;
```

3) 저장 프로시저

☞ 실습 5-7. 프로시저 생성 및 실행 기본

```
> DELIMITER $$
    CREATE PROCEDURE proc_test1()
    BEGIN
       SELECT * FROM `Member`;
       SELECT * FROM `Department`;
    END $$
    DELIMITER ;

> CALL proc_test1();
```

☞ 실습 5-8. 매개변수를 갖는 프로시저 생성 및 실행

```
> DELIMITER $$
    CREATE PROCEDURE proc_test2(IN _userName VARCHAR(10))
    BEGIN
       SELECT * FROM `Member` WHERE `name`=_userName;
    END $$
    DELIMITER ;

> CALL proc_test2('김유신');

> DELIMITER $$
    CREATE PROCEDURE proc_test3(IN _pos VARCHAR(10), IN _dep TINYINT)
    BEGIN
       SELECT * FROM `Member` WHERE `pos`=_pos AND `dep`=_dep;
    END $$
    DELIMITER ;

> CALL proc_test3('차장', 101);

> DELIMITER $$
    CREATE PROCEDURE proc_test4(IN _pos VARCHAR(10), OUT _count INT)
    BEGIN
       SELECT COUNT(*) INTO _count FROM `Member` WHERE `pos`=_pos ;
    END $$
    DELIMITER ;

> CALL proc_test4('대리', @_count);
> SELECT CONCAT('_count : ', @_count)
```

☞ 실습 5-9. 프로시저 프로그래밍

```
> DELIMITER $$
    CREATE PROCEDURE proc_test5(IN _name VARCHAR(10))
    BEGIN
       DECLARE userId VARCHAR(10);
       select `uid` into userId from `Member` where `name` = _name;
       select * from `Sales` where `uid`=userId;
    END $$
    DELIMITER ;

> CALL proc_test5('김유신');

> DELIMITER $$
    CREATE PROCEDURE proc_test6()
    BEGIN
       DECLARE num1 INT;
       DECLARE num2 INT;

       SET num1 = 1;
       SET num2 = 2;

       IF (NUM1 > NUM2) THEN
          SELECT 'NUM1이 NUM2보다 크다.' as `결과2`;
       ELSE
          SELECT 'NUM1이 NUM2보다 작다.' as `결과2`;
       END IF;
    END $$
    DELIMITER ;

> CALL proc_test6();

> DELIMITER $$
    CREATE PROCEDURE proc_test7()
    BEGIN
       DECLARE sum INT;
       DECLARE num INT;

       SET sum = 0;
       SET num = 1;

       WHILE (num <= 10) DO
          SET sum = sum + num;
          SET num = num + 1;
       END WHILE;

       SELECT sum AS '1부터 10까지 합계';
    END $$
    DELIMITER ;

> CALL proc_test7();
```

☞ 실습 5-10. 커서를 활용하는 프로시저

```
> DELIMITER $$
    CREATE PROCEDURE proc_test8()
    BEGIN
      # 변수 선언
      DECLARE total INT DEFAULT 0;
      DECLARE price INT;
      DECLARE endOfRow BOOLEAN DEFAULT false;

      # 커서 선언
      DECLARE salesCursor CURSOR FOR
           SELECT `sale` FROM `Sales`;

      # 반복 조건
      DECLARE CONTINUE HANDLER
           FOR NOT FOUND SET endOfRow = TRUE;

      # 커서 열기
      OPEN salesCursor;

      cursor_loop: LOOP
        FETCH salesCursor INTO price;

        IF endOfRow THEN
          LEAVE cursor_loop;
        END IF;

        SET total = total + price;
      END LOOP;

      SELECT total AS '전체 합계';

      CLOSE salesCursor;
    END $$
    DELIMITER ;

> CALL proc_test8();
```

4) 저장 함수

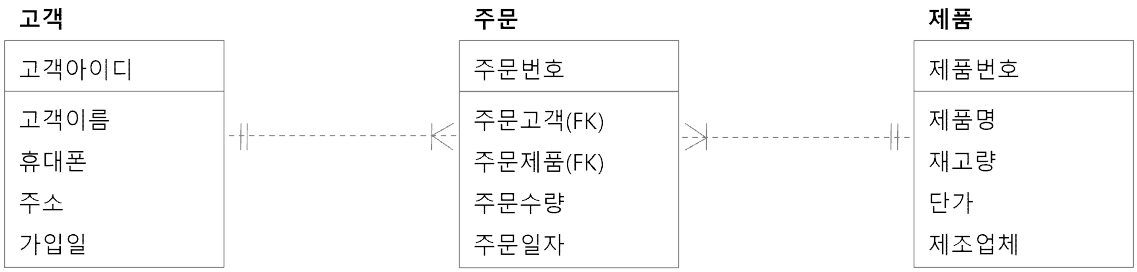
☞ 실습 5-11. 저장 함수 생성 및 호출

```
> DELIMITER $$
    CREATE FUNCTION func_test1(_userid VARCHAR(10)) RETURNS INT
    BEGIN
        DECLARE total INT;

        SELECT SUM(`sale`) INTO total FROM `Sale` WHERE `uid`=_userid;

        RETURN total;
    END $$
    DELIMITER ;

> SELECT func_test1('a101');

> DELIMITER $$
    CREATE FUNCTION func_test2(_sale INT) RETURNS DOUBLE
    BEGIN
        DECLARE bonus DOUBLE;

        IF (_sale >= 100000) THEN
            SET bonus = _sale * 0.1;
        ELSE
            SET bonus = _sale * 0.05;
        END IF;

    RETURN bonus;
    END $$
    DELIMITER ;

> SELECT
    `uid`,
    `year`,
    `month`,
    `sale`,
    func_test2(`sale`) as `bonus`
  FROM `Sale`;
```

제6장 데이터 모델링

1) Shop 모델링 실습

☞ 실습 6-1. 아래 테이블 명세서를 참고해서 ERD 작성하시오.



데이터베이스	ShopERD		테이블	고객(Customer)				
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	고객아이디	custId	varchar(10)	Y			Y	
	이름	name	varchar(10)				Y	
	휴대폰	hp	char(13)		Y			NULL
	주소	addr	varchar(100)					NULL
	가입일	rdate	date				Y	

데이터베이스	ShopERD		테이블	제품(Product)				
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	제품번호	prodNo	int	Y			Y	
	제품명	prodName	varchar(10)				Y	
	재고량	stock	int				Y	0
	단가	price	int					NULL
	제조업체	company	varchar(20)				Y	

데이터베이스	ShopERD		테이블	주문(Order)				
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	주문번호	orderNo	int	Y			Y	AI
	주문고객	orderId	varchar(10)			Y	Y	
	주문제품	orderProduct	int			Y	Y	
	주문수량	orderCount	int				Y	1
	주문일자	orderDate	datetime				Y	

☞ 실습 6-2. 아래와 같이 데이터를 입력하시오.

Customer

custId	name	hp	addr	rdate
c101	김유신	010-1234-1001	경남 김해시	2023-01-01
c102	김춘추	010-1234-1002	경남 경주시	2023-01-02
c103	장보고	010-1234-1003	전남 완도군	2023-01-03
c104	강감찬	010-1234-1004	서울시 관악구	2023-01-04
c105	이순신	010-1234-1005	부산시 금정구	2023-01-05

Product

prodNo	prodName	stock	price	comapny
1	새우깡	5,000	1,500	농심
2	초코파이	2,500	2,500	오리온
3	포카칩	3,600	1,700	오리온
4	양파링	1,250	1,800	농심
5	죠리퐁	2,200	NULL	크라운

Order

orderNo	orderId	orderProduct	orderCount	orderDate
1	c102	3	2	2023-01-01 13:15:10
2	c101	4	1	2023-01-01 13:15:12
3	c102	1	1	2023-01-01 13:15:14
4	c103	6	5	2023-01-01 13:15:16
5	c105	2	1	2023-01-01 13:15:18

☞ 실습 6-3. 다음 데이터를 조회 하시오.

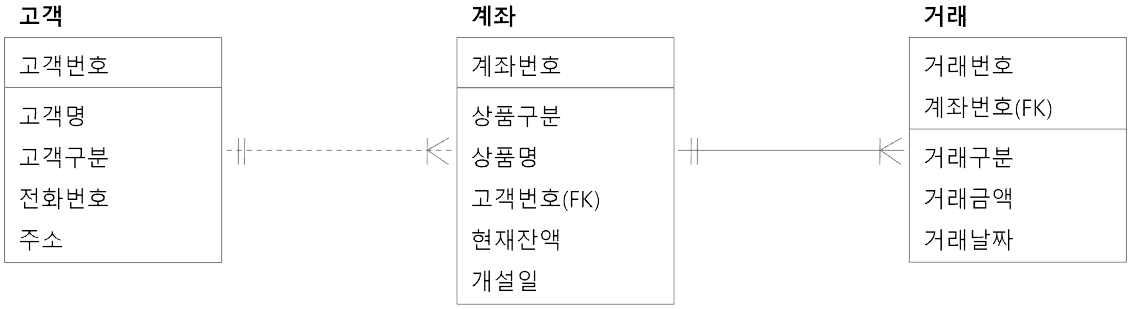
> 모든 주문의 주문번호, 주문한 고객명, 주문한 상품명, 주문 수량, 주문일을 조회하시오.

> 김유신이 주문한 상품의 주문번호, 상품번호, 상품명, 가격, 주문수량, 주문일을 조회하시오.

> 주문한 상품의 총 주문 금액을 조회하시오.

2) Bank 모델링 실습

☞ 실습 6-4. 아래 테이블 명세서를 참고해서 ERD 작성하시오.



데이터베이스	BankERD			테이블			고객(bank_customer)	
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	고객번호	c_no	varchar(14)	Y			Y	
	고객명	c_name	varchar(20)				Y	
	고객구분	c_dist	tinyint				Y	0
	전화번호	c_phone	char(13)				Y	
	주소	c_addr	varchar(50)					NULL

데이터베이스	BankERD			테이블			계좌(bank_account)	
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	계좌번호	a_no	char(11)	Y			Y	
	상품구분	a_item_dist	char(2)				Y	
	상품명	a_item_name	varchar(20)				Y	
	고객번호	a_c_no	varchar(14)			Y	Y	
	현재잔액	a_balance	int				Y	0
	개설일	a_open_date	date				Y	

데이터베이스	BankERD			테이블			거래(bank_transaction)	
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	거래번호	t_no	int	Y			Y	AI
	계좌번호	t_a_no	char(11)			Y	Y	
	거래구분	t__dist	tinyint				Y	
	거래금액	t_amount	int				Y	0
	거래날짜	t_datetime	datetime				Y	

☞ 실습 6-5. 아래와 같이 데이터를 입력하시오.

bank_customer

c_no	c_name	c_dist	c_phone	c_addr
730423-1000001	김유신	1	010-1234-1001	경남 김해시
730423-1000002	김춘추	1	010-1234-1002	경남 경주시
750423-1000003	장보고	1	010-1234-1003	전남 완도군
102-12-51094	(주)정보산업	2	051-500-1004	부산시 부산진구
930423-1000005	이순신	1	010-1234-1005	서울 종로구

c_dist - 1:개인고객, 2:기업고객

bank_account

a_no	a_item_dist	a_item_name	a_c_no	a_balance	a_open_date
101-11-1001	S1	자유저축예금	730423-1000001	1,550,000	2011-04-11
101-11-1002	S1	자유저축예금	930423-1000005	260,000	2011-05-12
101-11-1003	S1	자유저축예금	750423-1000003	75,000	2011-06-13
101-12-1001	S2	기업전용예금	102-12-51094	15,000,000	2011-07-14
101-13-1001	S3	정기저축예금	730423-1000002	1,200,000	2011-08-15

a_item_dist - S1:자유저축예금, S2:기업전용예금, S3:정기저축예금

bank_transaction

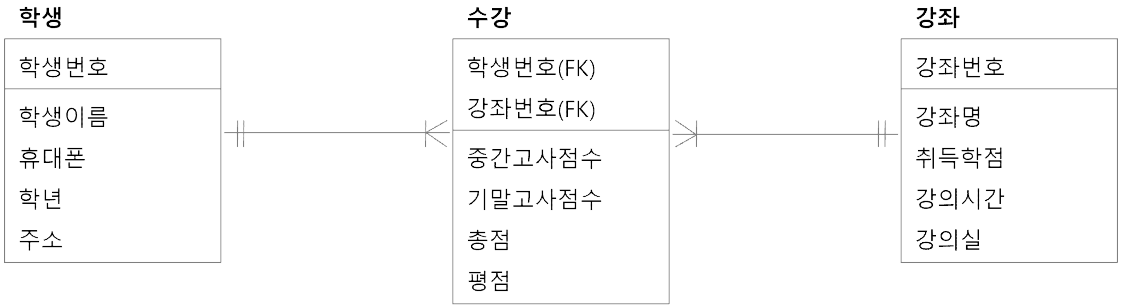
t_no	t_a_no	t_dist	t_amount	t_datetime
1	101-11-1001	1	50,000	2023-01-01 13:15:10
2	101-12-1001	2	1,000,000	2023-01-02 13:15:12
3	101-11-1002	3	260,000	2023-01-03 13:15:14
4	101-11-1002	2	100,000	2023-01-04 13:15:16
5	101-11-1003	3	75,000	2023-01-05 13:15:18
6	101-11-1001	1	150,000	2023-01-05 13:15:28

t_dist - 1:입금, 2:출금, 3:잔액조회

☞ 실습 6-6. 다음 데이터를 조회 하시오.
> 모든 고객의 고객번호, 이름, 연락처, 계좌번호, 계좌상품명, 현재잔액을 조회하시오.
> 이순신 고객명으로 모든 거래내역 중 거래구분, 거래금액, 거래날짜를 조회하시오.
> 개인고객 중 현재잔액이 가장 큰 고객의 주민번호, 고객명, 계좌번호, 현재잔액, 계좌개설일을 조회하시오.

3) College 모델링 실습

☞ 실습 6-7. 아래 테이블 명세서를 참고해서 ERD 작성하시오.



데이터베이스	CollegeERD		테이블	학생(Student)				
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	학생번호	stdNo	char(8)	Y			Y	
	학생이름	stdName	varchar(20)				Y	
	휴대폰	stdHp	char(13)		Y		Y	
	학년	stdYear	tinyint				Y	
	주소	stdAddress	varchar(100)					NULL

데이터베이스	CollegeERD		테이블	강좌(Lecture)				
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	강좌번호	lecNo	int	Y			Y	
	강좌명	lecName	varchar(20)				Y	
	취득학점	lecCredit	tinyint				Y	
	강의시간	lecTime	tinyint				Y	
	강의실	lecClass	varchar(10)					NULL

데이터베이스	CollegeERD		테이블	수강(Register)				
구분	Logical	Physical						
		Column	Data Type	PK	UK	FK	Not null	Default
속성	학생번호	regStdNo	char(8)			Y	Y	
	강좌번호	regLecNo	int			Y	Y	
	중간고사점수	regMidScore	int					NULL
	기말고사점수	regFinalScore	int					NULL
	총점	regTotalScore	int					NULL
	평점	regGrade	char(1)					NULL

☞ 실습 6-8. 아래와 같이 데이터를 입력하시오.

Student

stdNo	stdName	stdHp	stdYear	stdAddress
20201011	김유신	010-1234-1001	3	경남 김해시
20201122	김춘추	010-1234-1002	3	경남 경주시
20210213	장보고	010-1234-1003	2	전남 완도군
20210324	강감찬	010-1234-1004	2	서울 관악구
20220415	이순신	010-1234-1005	1	서울 종로구

Lecture

lecNo	lecName	lecCredit	lecTime	lecClass
101	컴퓨터과학 개론	2	40	본301
102	프로그래밍 언어	3	52	본302
103	데이터베이스	3	56	본303
104	자료구조	3	60	본304
105	운영체제	3	52	본305

Register

regStdNo	regLecNo	regMidScore	regFinalScore	regTotalScore	regGrade
20220415	101	60	30	(NULL)	(NULL)
20210324	103	54	36	(NULL)	(NULL)
20201011	105	52	28	(NULL)	(NULL)
20220415	102	38	40	(NULL)	(NULL)
20210324	104	56	32	(NULL)	(NULL)
20210213	103	48	40	(NULL)	(NULL)

☞ 실습 6-9. 다음 데이터를 조회 하시오.

> 이번 학기에 수강신청 하지 않은 학생의 학번, 이름, 연락처, 학년을 조회하시오.

> 중간고사 점수와 기말고사 점수의 총합을 구하고 등급을 구하시오.

> 2학년 학생의 학번, 이름, 학년, 수강 강좌명, 중간점수, 기말점수, 총합, 등급을 조회하시오.

제7장 정규화

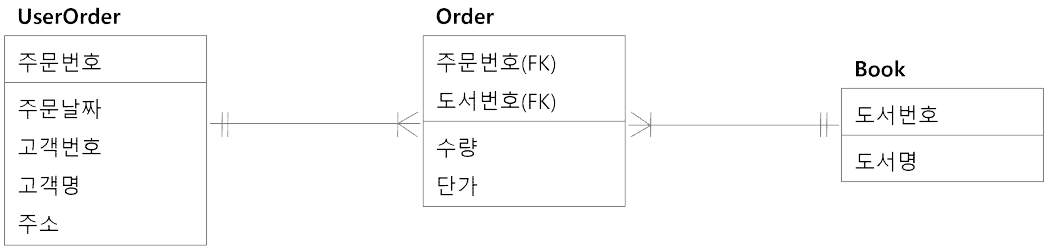
1) 정규화 실습

☞ 실습 7-1. 아래 비정규형 릴레이션을 제1정규화를 수행하시오.

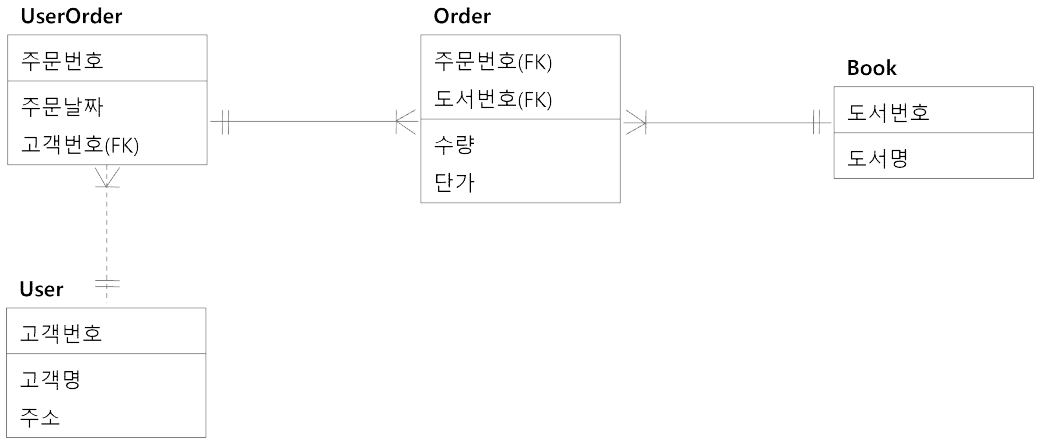
· 도서 구매 정보 테이블

주문번호	주문날짜	고객번호	고객명	주소	도서번호	도서명	수량	가격
10001	2024-01-12	a101	김유신	김해	101	프로그래밍	1	28,000
10002	2024-01-12	a102	김춘추	경주	101, 102	프로그래밍, 자료구조	1, 2	60,000
10003	2024-01-12	a103	장보고	완도	102	자료구조	2	32,000
10004	2024-01-12	a104	강감찬	서울	110	데이터베이스	1	25,000
10005	2024-01-12	a105	이순신	서울	110, 102	데이터베이스, 자료구조	1, 1	41,000

☞ 실습 7-2. 위 비정규형 릴레이션을 제2정규화를 수행하시오.



☞ 실습 7-3. 제1정규형 릴레이션을 제3정규화를 수행하시오.



제8장 트랜잭션과 병행 제어

1) 트랜잭션

☞ 실습 8-1. 트랜잭션 Commit

```
> START TRANSACTION;
> SELECT * FROM `bank_account`;
> UPDATE `bank_account` SET
                          `a_balance` = `a_balance` - 10000
                      WHERE
                          `a_no` = '101-11-1001';
> UPDATE `bank_account` SET
                          `a_balance` = `a_balance` + 10000
                      WHERE
                          `a_no` = '101-11-1212';

> COMMIT;
> SELECT * FROM `bank_account`;
```

☞ 실습 8-2. 트랜잭션 Rollback

```
> START TRANSACTION;
> UPDATE `bank_account` SET
                          `a_balance` = `a_balance` - 10000
                      WHERE
                          `a_no` = '101-11-1001';
> UPDATE `bank_account` SET
                          `a_balance` = `a_balance` + 10000
                      WHERE
                          `a_no` = '101-11-1212';

> SELECT * FROM `bank_account`;

> ROLLBACK;
> SELECT * FROM `bank_account`;
```

☞ 실습 8-3. 커밋 OFF

```
> SELECT @@AUTOCOMMIT;
> SET AUTOCOMMIT = 0;
> UPDATE `bank_account` SET
                          `a_balance` = `a_balance` - 10000
                      WHERE
                          `a_no` = '101-11-1001';

> SELECT * FROM `bank_account`;
```