20200508 DB과제 기초 6

B반 신석환

#UNION

2개 이상의 쿼리 결과를 합쳐서
보여주는 것

```
mysql> SELECT * FROM DIRECTORY;
+----+-------------------+-----------
| NO | WORDING           | CONTENTS
+----+-------------------+-----------
|  1 | PRIMARY KEY       | 다른 항목과
|  2 | FOREIGN KEY       | 테이블 내의
|  3 | SubQuery          | 하나의 SQL문
|  4 | View              | 데이터베이스
|  5 | Stored Procedure  | 여러 SQL 문을
|  6 | JDBC_Connect_Java | 자바에서 데이
|  7 | Trigger           | 테이블에 대한
+----+-------------------+-----------
7 rows in set (0.00 sec)

mysql> SELECT * FROM MYSQL
    -> ;
+----+----------------+-----------------
| No | Wording        | Exp
+----+----------------+-----------------
|  1 | SELECT         | Searching Columr
|  2 | INSERT         | Add Detail
|  3 | CREATE         | Create Somethins
|  4 | DELETE         | Delete Somethin
|  5 | UPDATE         | Changing column
|  6 | TRUNCATE       | Delete Somethins
|  7 | SHOW           | Show databases c
|  8 | USE            | Use a database c
|  9 | ALTER          | Add or Delete Co
| 10 | ORDER BY       | Order a column
| 11 | GROUP BY       | Grouping Details
| 12 | AUTO_INCREMENT | Making No Machar
+----+----------------+-----------------
12 rows in set (0.00 sec)

mysql> SELECT WORDING FROM MYSQL
    -> UNION
    -> SELECT WORDING FROM DIRECTORY;
+--------------------+
| WORDING            |
+--------------------+
| SELECT             |
| INSERT             |
| CREATE             |
| DELETE             |
| UPDATE             |
| TRUNCATE           |
| SHOW               |
| USE                |
| ALTER              |
| ORDER BY           |
| GROUP BY           |
| AUTO_INCREMENT     |
| PRIMARY KEY        |
| FOREIGN KEY        |
| SubQuery           |
| View               |
| Stored Procedure   |
| JDBC_Connect_Java  |
| Trigger            |
+--------------------+
19 rows in set (0.00 sec)
```

```
mysql> SELECT WORDING FROM DIRECTORY
    -> UNION
    -> SELECT WORDING FROM DIRECTORYDB
    -> UNION
    -> SELECT WORDING FROM MYSQL;
+--------------------+
| WORDING            |
+--------------------+
| PRIMARY KEY        |
| FOREIGN KEY        |
| SubQuery           |
| View               |
| Stored Procedure   |
| JDBC_Connect_Java  |
| Trigger            |
| Data Modeling      |
| DBMS               |
| DML                |
| DDL                |
| DCL                |
| TCL                |
| Database           |
| Field              |
| Record             |
| Table              |
| SELECT             |
| INSERT             |
| CREATE             |
| DELETE             |
| UPDATE             |
| TRUNCATE           |
| SHOW               |
| USE                |
| ALTER              |
| ORDER BY           |
| GROUP BY           |
| AUTO_INCREMENT     |
+--------------------+
29 rows in set (0.00 sec)
```

#JOIN – INNER JOIN

양 테이블에서 일치하는 레코드를 출력

```
mysql> SELECT * FROM STUDENT;
+----+---------+----------+-------+----------+--------------+------+
| NO | NAME    | SCHOOL   | GRADE | ADDRESS  | PHONE_NUMBER | ETC  |
+----+---------+----------+-------+----------+--------------+------+
|  1 | William | jamsil   | 3     | Seoul    | 01011112222  |      |
|  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  3 | Seok    | jeju     | 2     | jeju     | 01033334444  |      |
|  4 | Hwan    | busan    | 1     | deagu    | 01044445555  |      |
|  5 | Juan    | jamsil   | 2     | Seoul    | 01055556666  |      |
|  6 | Lee     | jeju     | 3     | canada   | 01066667777  |      |
+----+---------+----------+-------+----------+--------------+------+
6 rows in set (0.03 sec)

mysql> CREATE TABLE EMPLOYEE(NO INT NOT NULL AUTO_INCREMENT PRIMARY KEY, NAME CHAR(10), DEPARTMENT CHAR(30
Query OK, 0 rows affected (1.09 sec)

mysql> INSERT INTO EMPLOYEE(NAME, DEPARTMENT, ADDR) VALUES('William', 'a', 'a'), ('SHIN', 'b', 'b'), ('seo
    -> ('JUAn', 'e', 'e'), ('IEE', 'f', 'f');
Query OK, 6 rows affected (0.07 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM EMPLOYEE;
+----+---------+------------+------+
| NO | NAME    | DEPARTMENT | ADDR |
+----+---------+------------+------+
|  1 | William | a          | a    |
|  2 | SHIN    | b          | b    |
|  3 | seoK    | c          | c    |
|  4 | HWan    | d          | d    |
|  5 | JUAn    | e          | e    |
|  6 | IEE     | f          | f    |
+----+---------+------------+------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM STUDENT INNER JOIN EMPLOYEE ON STUDENT.NAME = EMPLOYEE.NAME;
+----+---------+----------+-------+----------+--------------+------+----+---------+------------+------+
| NO | NAME    | SCHOOL   | GRADE | ADDRESS  | PHONE_NUMBER | ETC  | NO | NAME    | DEPARTMENT | ADDR |
+----+---------+----------+-------+----------+--------------+------+----+---------+------------+------+
|  1 | William | jamsil   | 3     | Seoul    | 01011112222  |      |  1 | William | a          | a    |
|  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |  2 | SHIN    | b          | b    |
|  3 | Seok    | jeju     | 2     | jeju     | 01033334444  |      |  3 | seoK    | c          | c    |
|  4 | Hwan    | busan    | 1     | deagu    | 01044445555  |      |  4 | HWan    | d          | d    |
|  5 | Juan    | jamsil   | 2     | Seoul    | 01055556666  |      |  5 | JUAn    | e          | e    |
|  6 | Lee     | jeju     | 3     | canada   | 01066667777  |      |  6 | IEE     | f          | f    |
+----+---------+----------+-------+----------+--------------+------+----+---------+------------+------+
6 rows in set (0.00 sec)
```

#JOIN – LEFT JOIN

```
mysql> SELECT * FROM STUDENT;
+----+---------+-----------+-------+-----------+--------------+------+
| NO | NAME    | SCHOOL    | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+----+---------+-----------+-------+-----------+--------------+------+
|  1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
|  4 | Hwan    | busan     | 1     | deagu     | 01044445555  |      |
|  5 | Juan    | jamsil    | 2     | Seoul     | 01055556666  |      |
|  6 | Lee     | jeju      | 3     | canada    | 01066667777  |      |
+----+---------+-----------+-------+-----------+--------------+------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEE;
+----+---------+------------+---------+
| NO | NAME    | DEPARTMENT | ADDRESS |
+----+---------+------------+---------+
|  1 | William | a          | a       |
|  2 | SHIN    | b          | b       |
|  3 | seoK    | c          | c       |
|  4 | HWan    | d          | d       |
|  5 | JUAn    | e          | e       |
|  6 | lEE     | f          | f       |
|  7 | William | a          | a       |
|  8 | SHIN    | b          | b       |
|  9 | seoK    | c          | c       |
| 10 | ASDFL   | d          | d       |
+----+---------+------------+---------+
10 rows in set (0.00 sec)
```

왼쪽 테이블의 모든 레코드 출력,
오른쪽 테이블에서는 일치하는 레코드만 출력

```
mysql> SELECT * FROM EMPLOYEE LEFT JOIN STUDENT ON EMPLOYEE.NAME = STUDENT.NAME;
+----+---------+------------+---------+------+---------+-----------+-------+-----------+--------------+------+
| NO | NAME    | DEPARTMENT | ADDRESS | NO   | NAME    | SCHOOL    | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+----+---------+------------+---------+------+---------+-----------+-------+-----------+--------------+------+
|  1 | William | a          | a       |    1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|  2 | SHIN    | b          | b       |    2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  3 | seoK    | c          | c       |    3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
|  4 | HWan    | d          | d       |    4 | Hwan    | busan     | 1     | deagu     | 01044445555  |      |
|  5 | JUAn    | e          | e       |    5 | Juan    | jamsil    | 2     | Seoul     | 01055556666  |      |
|  6 | lEE     | f          | f       |    6 | Lee     | jeju      | 3     | canada    | 01066667777  |      |
|  7 | William | a          | a       |    1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|  8 | SHIN    | b          | b       |    2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  9 | seoK    | c          | c       |    3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
| 10 | ASDFL   | d          | d       | NULL | NULL    | NULL      | NULL  | NULL      | NULL         | NULL |
+----+---------+------------+---------+------+---------+-----------+-------+-----------+--------------+------+
10 rows in set (0.00 sec)
```

#JOIN – RIGHT JOIN

```
mysql> SELECT * FROM STUDENT;
+----+---------+-----------+-------+-----------+--------------+------+
| NO | NAME    | SCHOOL    | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+----+---------+-----------+-------+-----------+--------------+------+
|  1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
|  3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
|  4 | Hwan    | busan     | 1     | deagu     | 01044445555  |      |
|  5 | Juan    | jamsil    | 2     | Seoul     | 01055556666  |      |
|  6 | Lee     | jeju      | 3     | canada    | 01066667777  |      |
+----+---------+-----------+-------+-----------+--------------+------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEE;
+----+---------+------------+---------+
| NO | NAME    | DEPARTMENT | ADDRESS |
+----+---------+------------+---------+
|  1 | William | a          | a       |
|  2 | SHIN    | b          | b       |
|  3 | seoK    | c          | c       |
|  4 | HWan    | d          | d       |
|  5 | JUAn    | e          | e       |
|  6 | IEE     | f          | f       |
|  7 | William | a          | a       |
|  8 | SHIN    | b          | b       |
|  9 | seoK    | c          | c       |
| 10 | ASDFL   | d          | d       |
+----+---------+------------+---------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEE RIGHT JOIN STUDENT ON EMPLOYEE.NAME = BINARY(STUDENT.NAME);
+------+---------+------------+---------+----+---------+-----------+-------+-----------+--------------+------+
| NO   | NAME    | DEPARTMENT | ADDRESS | NO | NAME    | SCHOOL    | GRADE | ADDRESS   | PHONE_NUMBER | ETC  |
+------+---------+------------+---------+----+---------+-----------+-------+-----------+--------------+------+
|    1 | William | a          | a       |  1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
|    7 | William | a          | a       |  1 | William | jamsil    | 3     | Seoul     | 01011112222  |      |
| NULL | NULL    | NULL       | NULL    |  2 | Shin    | seoungnam | 2     | SeoungNam | 01022223333  |      |
| NULL | NULL    | NULL       | NULL    |  3 | Seok    | jeju      | 2     | jeju      | 01033334444  |      |
| NULL | NULL    | NULL       | NULL    |  4 | Hwan    | busan     | 1     | deagu     | 01044445555  |      |
| NULL | NULL    | NULL       | NULL    |  5 | Juan    | jamsil    | 2     | Seoul     | 01055556666  |      |
| NULL | NULL    | NULL       | NULL    |  6 | Lee     | jeju      | 3     | canada    | 01066667777  |      |
+------+---------+------------+---------+----+---------+-----------+-------+-----------+--------------+------+
7 rows in set (0.00 sec)
```

오른쪽 테이블의 모든 레코드 출력,
왼쪽 테이블에서는 일치하는 레코드만 출력

#순서도 (Flowchart)

순서도 (flowchart)

순서도(flowchart)는 다이어그램의 종류중 하나로 어떤 일을 처리할때 여러 종류의 상자와 ○
를 이어주는 화살표를 이용해
명령의 순서를 보여주는 **알고리즘(algorithm)**[1] 혹은 **프로세스(process)**[2]를 말한다

순서도 기호

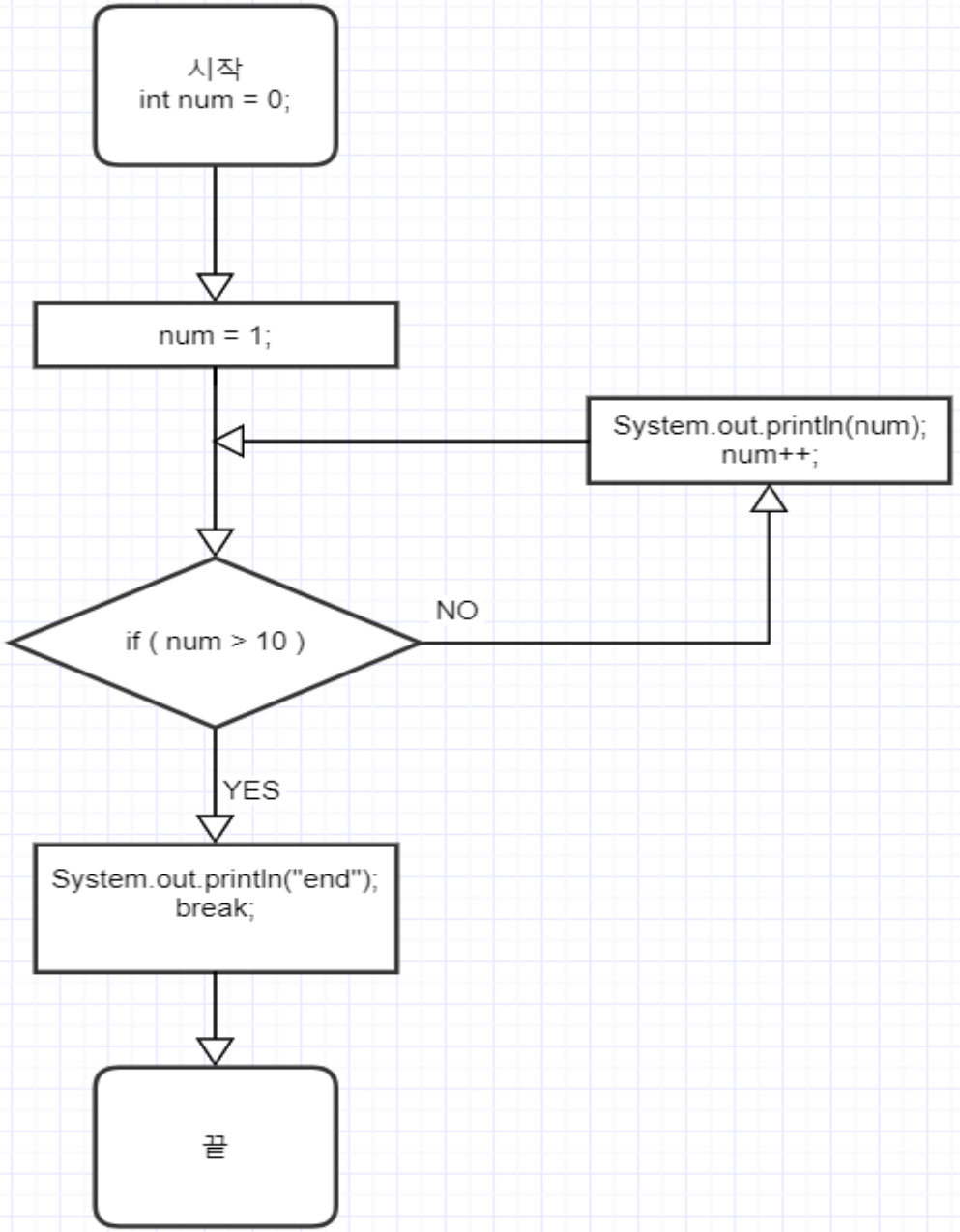
■ 순서도 기호 ■

	단말	순서도의 시작과 끝		카드입력	카드리더(card reader)를 통한 입력
	흐름선	작업 흐름을 명시		수동입력	키보드를 통한 입력
	준비	작업 단계 시작 전 준비 (변수 및 초기치 선언 등)		서브루틴	정의하여 둔 부프로그램의 호출
	처리	처리하여야 할 작업을 명시 (변수에 계산 값 입력 등)		페이지 내 연결자	한 페이지 내의 순서도 연결
	입출력	일반적인 데이터의 입력 또는 결과의 출력		페이지 간 연결자	페이지가 다른 순서도의 연결
	판단	조건에 따라 흐름선을 선택 (일반적으로 참, 거짓 구분)		화면표시	처리결과 또는 메시지를 모니터를 이용하여 출력
	프린트	프린터를 이용한 출력 (서류 등의 지면에 출력)		결합	기본 흐름선에 다른 흐름선 합류

#순서도 (Flowchart)

```java
public class methods {
    public static void main(String[] args) {
        int num = 0;
        num = 1;
        while(true) {
            if(num > 10) {
                System.out.println("end");
                break;
            }
            System.out.println(num);
            num++;
        }

    }
}
```



#복습과제 – RENTAL_CAL COMPANY

1. 렌터카 업체 Table 만들기

1) Table 구성 및 내용

(1) 회원정보 테이블 : No, 이름, 전화번호 등 (Column 7개)

(2) 차량별 예약 테이블 : No, 차량명, 예약일, 예약여부, 예약자명

2) 데이터

(1) 회원정보 최소 5건 이상

(2) 렌터카 최소 3대, 렌터카별 예약일 오늘 기준 앞뒤로 5일

3) 아래 내용 출력하기

(1) JOIN을 사용하여 이름을 기준으로 렌터카와 회원정보 전체 동시 출력

(2) 서브쿼리를 사용하여 A라는 렌터카의 예약현황을 출력할 때, 회원의 이름이 “김 " 으로 시작하는 사람 출력

#복습과제 – RENTAL_CAL COMPANY

1

```
mysql> INSERT INTO CUSTOMER(NAME, NUMBER, LICENSE_NUMBER, SEX, ADDRESS, CUSTOMER_NUMBER) VALUES
    -> ('SHIN', '01011112222', '13323', 'M', 'SEOUL', '1'),
    -> ('SEOK', '01011113333', '13322', 'M', 'SEOUL', '2'),
    -> ('HWAN', '01011114444', '13333', 'M', 'BUSAN', '3'),
    -> ('WILLIAM', '01011115555', '13324', 'M', 'JEON_JU', '4'),
    -> ('JUAN', '01011116666', '13344', 'F', 'SEONG_NAM', '5');
Query OK, 5 rows affected (0.11 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM CUSTOMER;
+----+---------+-------------+----------------+------+-----------+-----------------+
| NO | NAME    | NUMBER      | LICENSE_NUMBER | SEX  | ADDRESS   | CUSTOMER_NUMBER |
+----+---------+-------------+----------------+------+-----------+-----------------+
|  1 | SHIN    | 01011112222 | 13323          | M    | SEOUL     | 1               |
|  2 | SEOK    | 01011113333 | 13322          | M    | SEOUL     | 2               |
|  3 | HWAN    | 01011114444 | 13333          | M    | BUSAN     | 3               |
|  4 | WILLIAM | 01011115555 | 13324          | M    | JEON_JU   | 4               |
|  5 | JUAN    | 01011116666 | 13344          | F    | SEONG_NAM | 5               |
+----+---------+-------------+----------------+------+-----------+-----------------+
5 rows in set (0.00 sec)
```

2

```
mysql> INSERT INTO BOOK(CAR_NAME, BOOK_DATE, WHETHER_BOOK, BOOKING_NAME) VALUES
    -> ('SONATA', '2020-05-02', 'Y', 'WILLIAM'),
    -> ('TICO', '2020-05-07', 'N', 'SHIN'),
    -> ('MINI', '2020-05-04', 'Y', 'JUAN'),
    -> ('AVANTE', '2020-05-10', 'N', 'HWAN'),
    -> ('SONATA', '2020-05-11', 'Y', 'SEOK');
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM BOOK;
+----+----------+-------------+---------------+--------------+
| NO | CAR_NAME | BOOK_DATE   | WHETHER_BOOK  | BOOKING_NAME |
+----+----------+-------------+---------------+--------------+
|  1 | SONATA   | 2020-05-02 | Y             | WILLIAM      |
|  2 | TICO     | 2020-05-07 | N             | SHIN         |
|  3 | MINI     | 2020-05-04 | Y             | JUAN         |
|  4 | AVANTE   | 2020-05-10 | N             | HWAN         |
|  5 | SONATA   | 2020-05-11 | Y             | SEOK         |
+----+----------+-------------+---------------+--------------+
5 rows in set (0.00 sec)
```

3-1

```
mysql> SELECT * FROM CUSTOMER LEFT JOIN BOOK ON CUSTOMER.NAME = BOOK.NAME
    -> UNION
    -> SELECT * FROM CUSTOMER RIGHT JOIN BOOK ON CUSTOMER.NAME = BOOK.NAME;
+------+---------+-------------+----------------+------+----------+-----------------+------+----------+------------+-------------+---------+
| NO   | NAME    | NUMBER      | LICENSE_NUMBER | SEX  | ADDRESS  | CUSTOMER_NUMBER | NO   | CAR_NAME | BOOK_DATE  | WHETHER_BOOK | NAME    |
+------+---------+-------------+----------------+------+----------+-----------------+------+----------+------------+-------------+---------+
|    1 | SHIN    | 01011112222 | 13323          | M    | SEOUL    | 1               |    2 | TICO     | 2020-05-07 | N           | SHIN    |
|    2 | SEOK    | 01011113333 | 13322          | M    | SEOUL    | 2               |    5 | SONATA   | 2020-05-11 | Y           | SEOK    |
|    3 | HWAN    | 01011114444 | 13333          | M    | BUSAN    | 3               |    4 | AVANTE   | 2020-05-10 | N           | HWAN    |
|    4 | WILLIAM | 01011115555 | 13324          | M    | JEON_JU  | 4               |    1 | SONATA   | 2020-05-02 | Y           | WILLIAM |
|    5 | JUAN    | 01011116666 | 13344          | F    | SEONG_NAM | 5               |    3 | MINI     | 2020-05-04 | Y           | JUAN    |
|    6 | KIM     | 01011113333 | 13322          | M    | SEOUL    | 6               |    6 | MINI     | 2020-05-04 | Y           | KIM     |
+------+---------+-------------+----------------+------+----------+-----------------+------+----------+------------+-------------+---------+
6 rows in set (0.00 sec)
```

3-2

```
mysql> SELECT * FROM CUSTOMER;
+----+--------+-------------+----------------+------+---------+-----------------+
| NO | NAME   | NUMBER      | LICENSE_NUMBER | SEX  | ADDRESS | CUSTOMER_NUMBER |
+----+--------+-------------+----------------+------+---------+-----------------+
|  1 | SHIN   | 01011112222 | 13323          | M    | SEOUL   | 1               |
|  2 | SEOK   | 01011113333 | 13322          | M    | SEOUL   | 2               |
|  3 | HWAN   | 01011114444 | 13333          | M    | BUSAN   | 3               |
|  4 | WILLIAM| 01011115555 | 13324          | M    | JEON_JU | 4               |
|  5 | JUAN   | 01011116666 | 13344          | F    | SEONG_NAM| 5              |
|  6 | KIM    | 01011113333 | 13322          | M    | SEOUL   | 6               |
+----+--------+-------------+----------------+------+---------+-----------------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM BOOK;
+----+----------+------------+--------------+---------+
| NO | CAR_NAME | BOOK_DATE  | WHETHER_BOOK | NAME    |
+----+----------+------------+--------------+---------+
|  1 | SONATA   | 2020-05-02 | Y            | WILLIAM |
|  2 | TICO     | 2020-05-07 | N            | SHIN    |
|  3 | MINI     | 2020-05-04 | Y            | JUAN    |
|  4 | AVANTE   | 2020-05-10 | N            | HWAN    |
|  5 | SONATA   | 2020-05-11 | Y            | SEOK    |
|  6 | MINI     | 2020-05-04 | Y            | KIM     |
+----+----------+------------+--------------+---------+
6 rows in set (0.00 sec)
mysql> SELECT * FROM BOOK;
+----+----------+------------+--------------+---------+
| NO | CAR_NAME | BOOK_DATE  | WHETHER_BOOK | NAME    |
+----+----------+------------+--------------+---------+
|  1 | SONATA   | 2020-05-02 | Y            | WILLIAM |
|  2 | TICO     | 2020-05-07 | N            | SHIN    |
|  3 | MINI     | 2020-05-04 | Y            | JUAN    |
|  4 | AVANTE   | 2020-05-10 | N            | HWAN    |
|  5 | SONATA   | 2020-05-11 | Y            | SEOK    |
|  6 | MINI     | 2020-05-04 | Y            | KIM     |
+----+----------+------------+--------------+---------+
6 rows in set (0.00 sec)
mysql> SELECT WHETHER_BOOK FROM BOOK WHERE NAME = (SELECT NAME FROM CUSTOMER WHERE NAME = 'KIM');
+--------------+
| WHETHER_BOOK |
+--------------+
| Y            |
+--------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM BOOK WHERE NAME = (SELECT NAME FROM CUSTOMER WHERE NAME LIKE '%KIM%');
+----+----------+------------+--------------+------+
| NO | CAR_NAME | BOOK_DATE  | WHETHER_BOOK | NAME |
+----+----------+------------+--------------+------+
|  6 | MINI     | 2020-05-04 | Y            | KIM  |
+----+----------+------------+--------------+------+
1 row in set (0.00 sec)
```

#복습과제 – 순서도

[복습]

2. 실시간 검색어 차트를 위한 순서도 및 테이블 구성하기

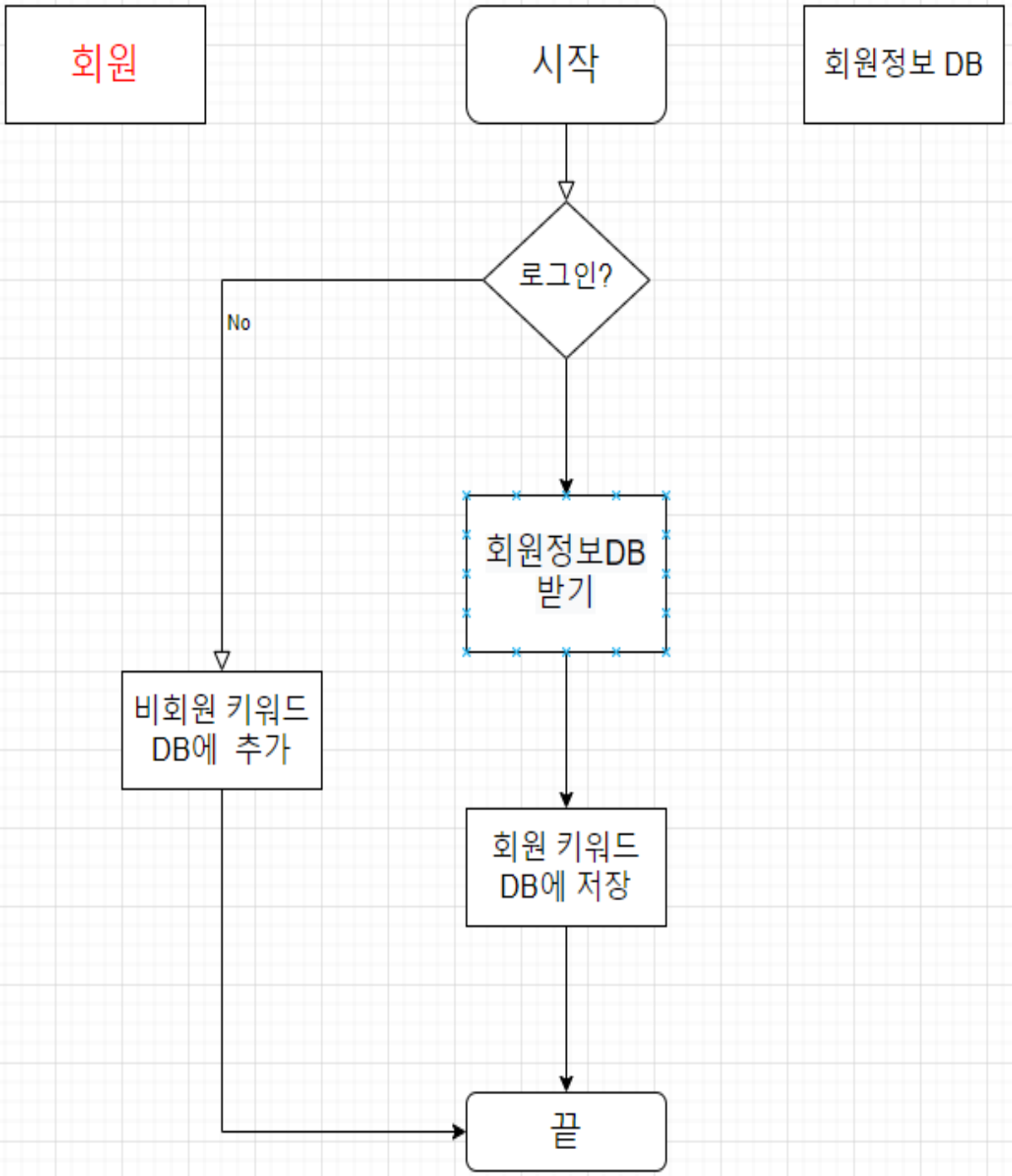
: 시간대별, 연령별 키워드 순위 1 ~ 10위가 나올 수 있도록 구성하기, 추가(개인별 자유)

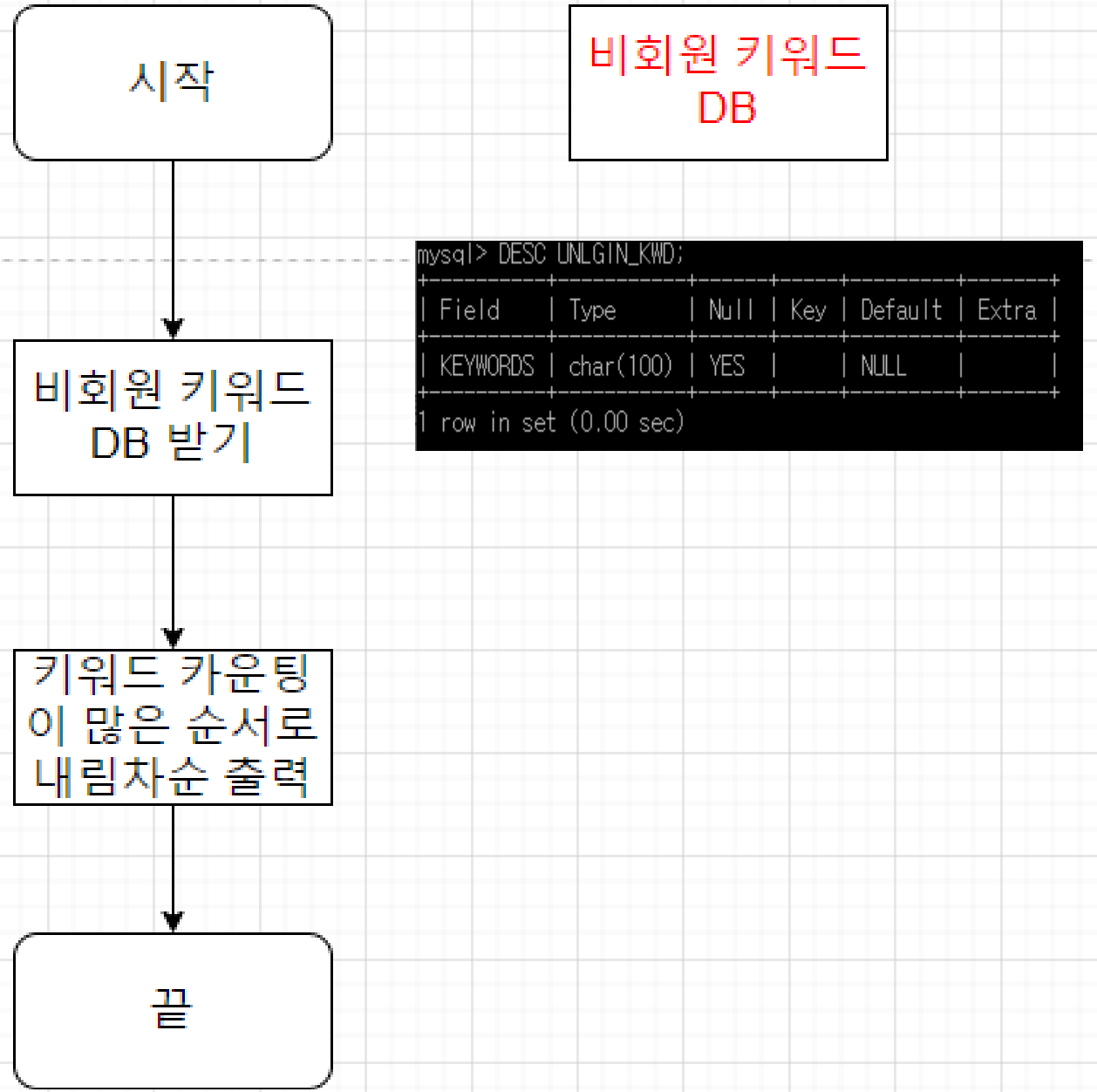
3. 실시간 뉴스 차트를 위한 순서도 및 테이블 구성하기

: 댓글 많은 순, 조회 많은 순, 추가(개인별 자유)

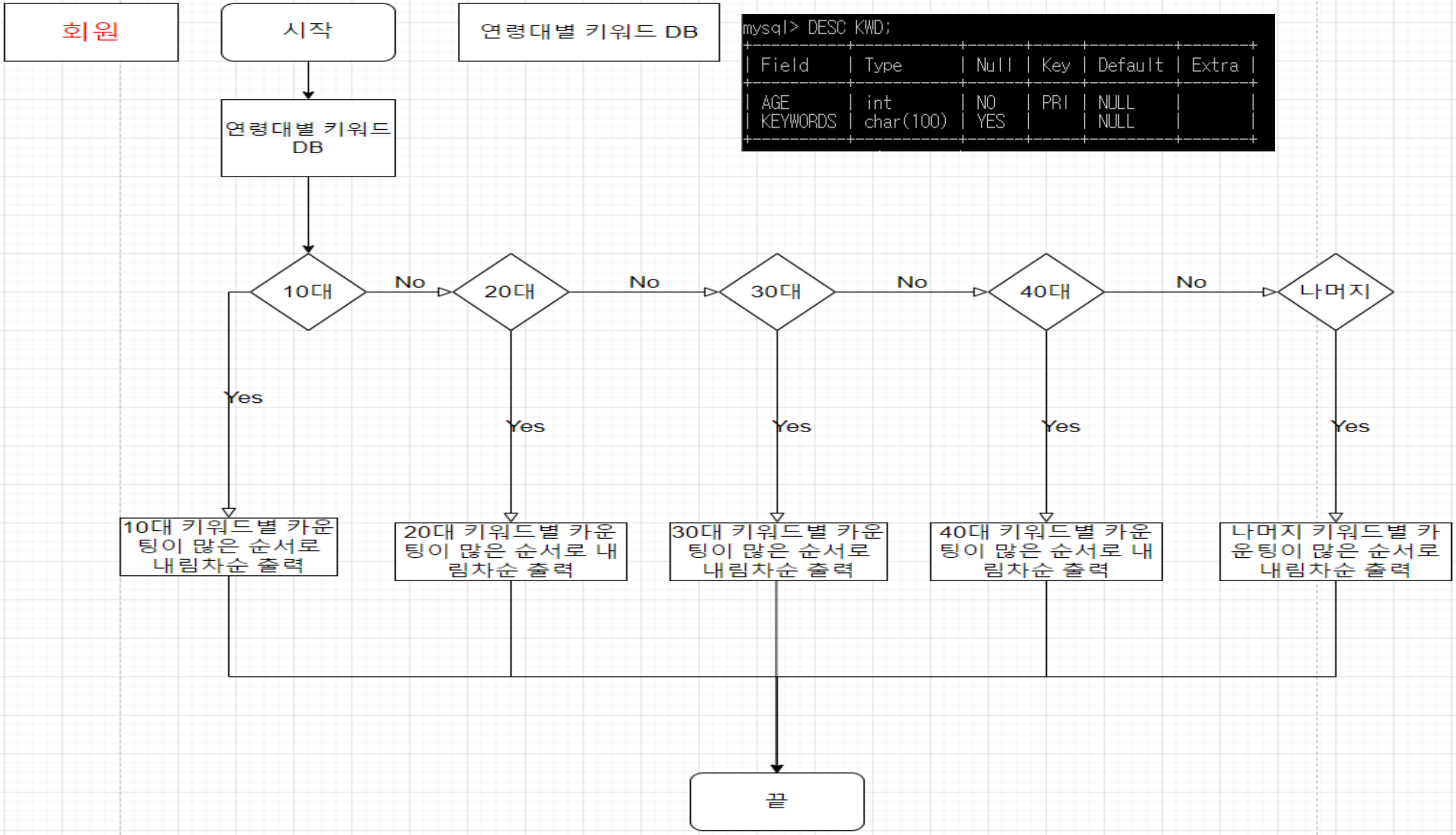
4. 쇼핑 카테고리별 차트를 위한 순서도 및 테이블 구성하기

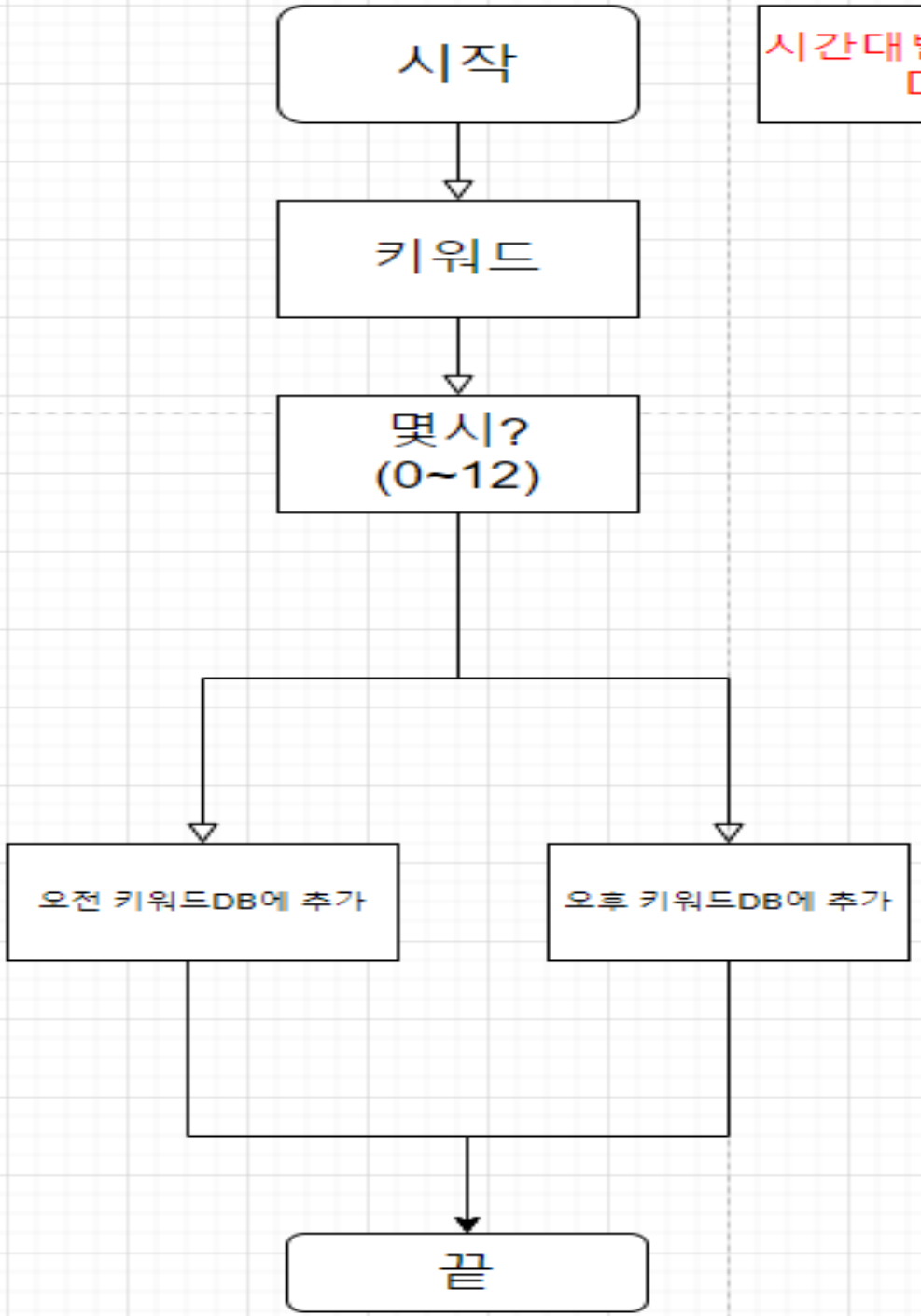
: 판매량 순, 누적 후기 순, 추가(개인별 자유)



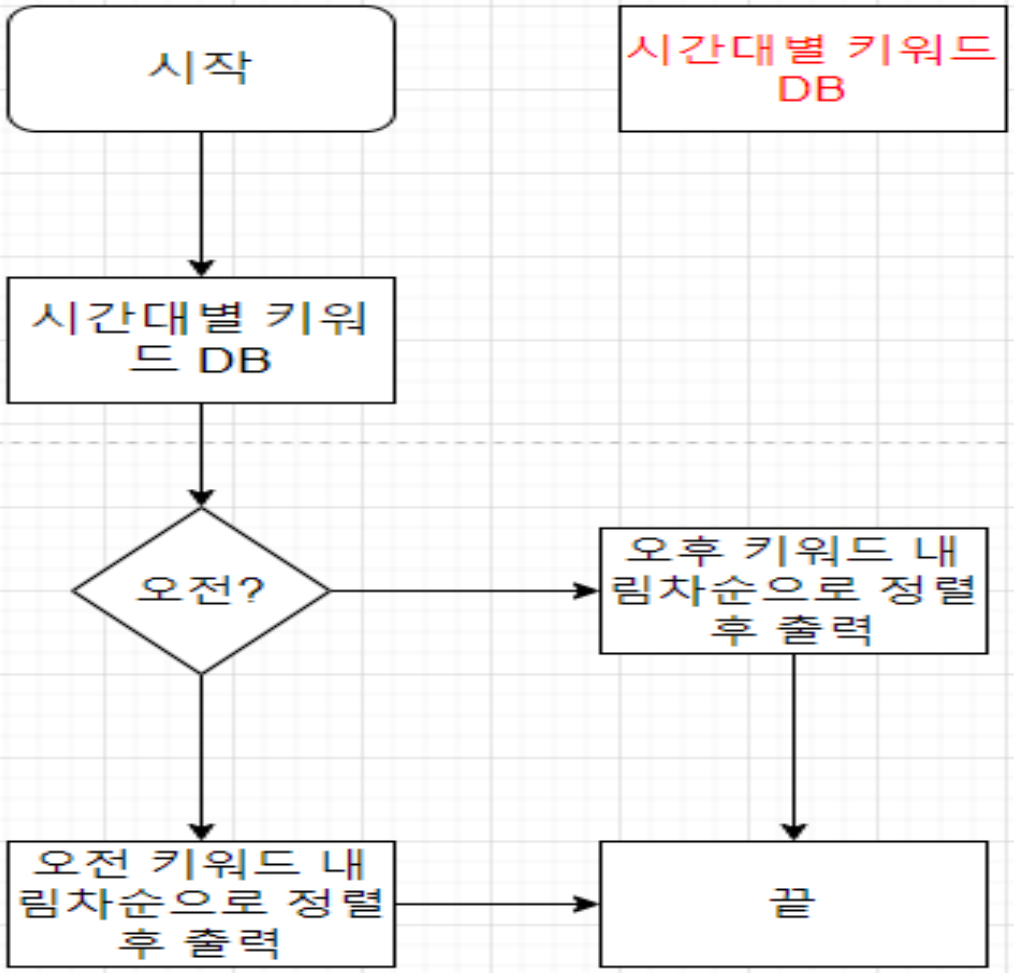



```
mysql> DESC UNLGIN_KWD;
+----------+-----------+------+-----+---------+-------+
| Field    | Type      | Null | Key | Default | Extra |
+----------+-----------+------+-----+---------+-------+
| KEYWORDS | char(100) | YES  |     | NULL    |       |
+----------+-----------+------+-----+---------+-------+
1 row in set (0.00 sec)
```


회원
시작
연령대별 키워드 DB
mysql> DESC KWD;
+----------+----------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+----------+----------+------+-----+---------+-------+
| AGE | int | NO | PRI | NULL | |
| KEYWORDS | char(100) | YES | | NULL | |
+----------+----------+------+-----+---------+-------+
연령대별 키워드 DB
10대
No
20대
No
30대
No
40대
No
나머지
Yes
Yes
Yes
Yes
Yes
10대 키워드별 카운팅이 많은 순서로 내림차순 출력
20대 키워드별 카운팅이 많은 순서로 내림차순 출력
30대 키워드별 카운팅이 많은 순서로 내림차순 출력
40대 키워드별 카운팅이 많은 순서로 내림차순 출력
나머지 키워드별 카운팅이 많은 순서로 내림차순 출력
끝

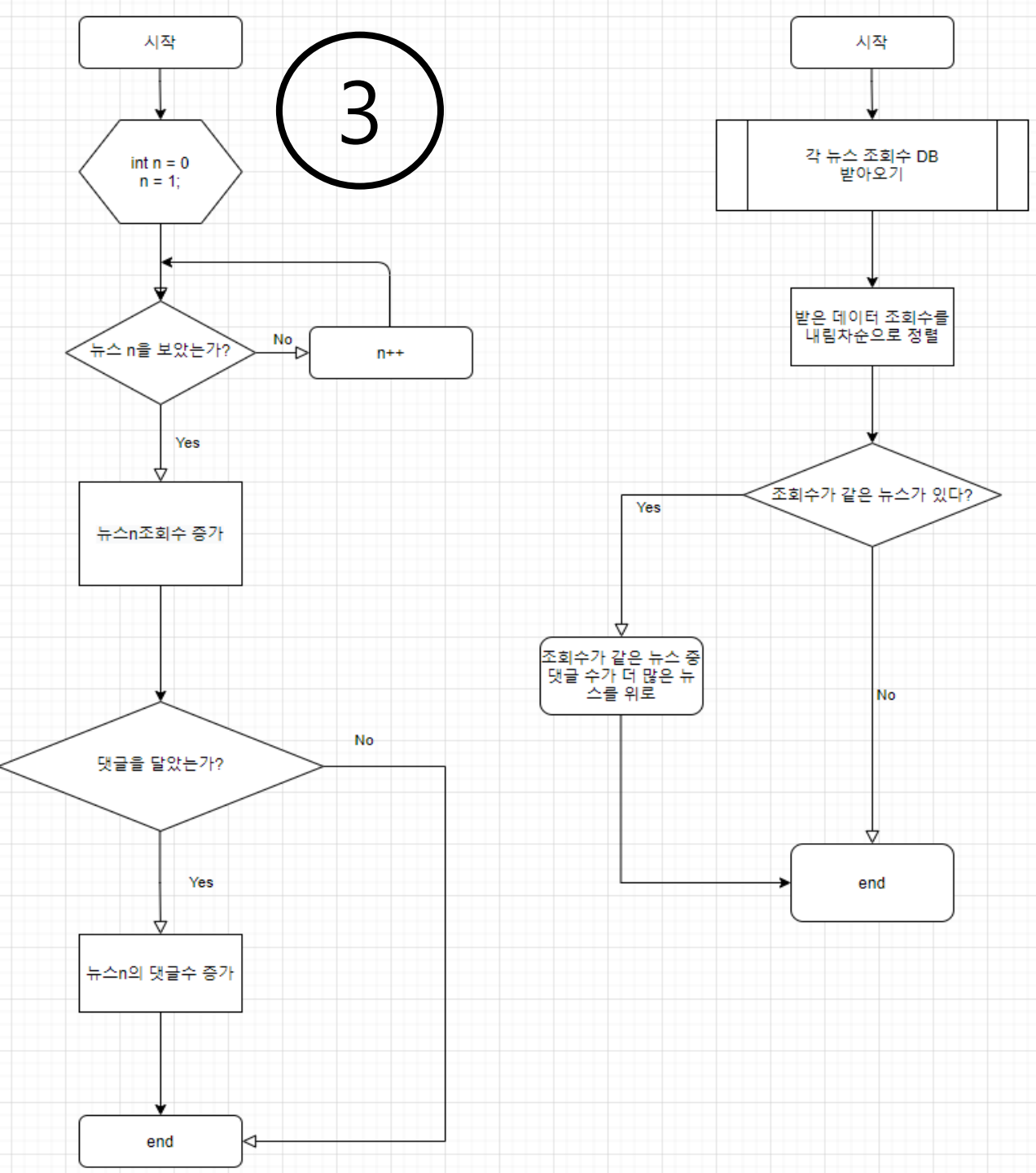


시간대별 키워드 DB



시간대별 키워드 DB

```
mysql> DESC TIME_KWD;
+---------+------+------+-----+---------+-------+
| Field   | Type | Null | Key | Default | Extra |
+---------+------+------+-----+---------+-------+
| TIME_AM | int  | YES  |     | NULL    |       |
| ITEM_PM | int  | YES  |     | NULL    |       |
+---------+------+------+-----+---------+-------+
```




```
mysql> SELECT * FROM NEW_COUNTING;
+------+--------------+---------+
| ID   | WATCHED_NEW  | COMENT  |
+------+--------------+---------+
|    5 | SHIN         | N       |
|    7 | SHIN         | N       |
|   10 | SEOK         | Y       |
|   13 | SEOK         | Y       |
|    1 | SHIN         | Y       |
|    2 | SHIN         | N       |
|    4 | SEOK         | N       |
|    3 | HWAN         | N       |
|    6 | JUAN         | Y       |
|    8 | JUAN         | Y       |
|    9 | SCHOOL       | Y       |
|   11 | SCHOOL       | N       |
|   12 | GRADE        | N       |
|   14 | SEOK         | N       |
+------+--------------+---------+
14 rows in set (0.00 sec)

mysql> SELECT * FROM NEWS;
+--------------+---------------+------------------+
| WATCHED_NEW  | ALL_COUNTING  | COMENT_COUNTING  |
+--------------+---------------+------------------+
| SHIN         |             4 |                1 |
| SEOK         |             4 |                2 |
| HWAN         |             1 |                0 |
| JUAN         |             2 |                2 |
| SCHOOL       |             2 |                1 |
| GRADE        |             1 |                0 |
+--------------+---------------+------------------+
6 rows in set (0.00 sec)
```

[예습]

- View, Index, Stored Procedure에 대한 정의 조사, 예문 3개씩 실행하기

View : 데이터베이스에 존재하는 일종의 가상 테이블

VIEW

```
mysql> CREATE VIEW TEST AS
    -> SELECT NO, WORDING
    -> FROM DIRECTORY
    -> WHERE NO = 2;
Query OK, 0 rows affected (0.23 sec)

mysql> SELECT * FROM TEST;
+----+-------------+
| NO | WORDING     |
+----+-------------+
|  2 | FOREIGN KEY |
+----+-------------+
1 row in set (0.01 sec)
```

```
+---------+------+
| CARNAME | NAME |
+---------+------+
| A       | JUNG |
| B       | KIM  |
| A       | KIM  |
+---------+------+
3 rows in set (0.00 sec)

mysql> CREATE VIEW VT SELECT
ERROR 1064 (42000): You have
 for the right syntax to use
mysql> CREATE VIEW VT AS SEL
Query OK, 0 rows affected (0

mysql> SELECT * FROM VT;
+---------+------+
| CARNAME | NAME |
+---------+------+
| A       | JUNG |
| A       | KIM  |
+---------+------+
```

```
+----+------+
| ID | NAME |
+----+------+
|  1 | KIM  |
|  3 | LEE  |
+----+------+
2 rows in set (0.02 sec)

mysql> CREATE VIEW VTS AS
Query OK, 0 rows affected

mysql> SELECT * FROM VTS;
+------+----+
| NAME | ID |
+------+----+
| KIM  |  1 |
+------+----+
```

Index : 대부분의 책의 제일 뒤에 붙어 있는 '찾아보기'와 같은 개념

INDEX

```
mysql> SELECT * FROM FRUITS;
+----+-----------+-------+------+
| NO | ITEM      | PRICE | ETC  |
+----+-----------+-------+------+
|  1 | APPLE     |  2000 | NULL |
|  2 | STRAWBERRY |  3000 | NULL |
|  3 | PINEAPPLE |  2500 | NULL |
|  4 | ORANGE    |  1500 | NULL |
|  5 | BANANA    |  4000 | NULL |
+----+-----------+-------+------+
5 rows in set (0.03 sec)

mysql> CREATE INDEX INDEX_TEST ON FRUITS(ITEM);
Query OK, 0 rows affected (0.39 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW INDEXT FROM FRUITS;
ERROR 1064 (42000): You have an error in your SQL syntax; c
mysql> SHOW INDEX FROM FRUITS;
+--------+------------+------------+--------------+--------
| Table  | Non_unique | Key_name   | Seq_in_index | Column_
+--------+------------+------------+--------------+--------
| fruits |          0 | PRIMARY    |            1 | NO
| fruits |          1 | INDEX_TEST |            1 | ITEM
+--------+------------+------------+--------------+--------
2 rows in set (0.09 sec)
```

```
mysql> CREATE INDEX IND_TEST ON SCHOOL(NAME);
Query OK, 0 rows affected (0.58 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW INDEX FROM SCHOOL;
+--------+------------+----------+--------------+-
| Table  | Non_unique | Key_name | Seq_in_index | C
+--------+------------+----------+--------------+--
| school |          0 | PRIMARY  |            1 | I
| school |          1 | IND_TEST |            1 | N
+--------+------------+----------+--------------+--
2 rows in set (0.12 sec)
```

```
mysql> SELECT * FROM TEST;
+----+-----------------------------------------------------------------------
-------------+--------+------------+-----------+----------------+
| NO | TITLE
             | C3     | C4         | MY_Answer | Answer         |
+----+-----------------------------------------------------------------------
-------------+--------+------------+-----------+----------------+
| 15 | A software system that enable user to define, create and maintain tl
             | DBMS   | SQLite     | NULL      | DBMS           |
| 16 | A structured set of data held in a computer, especially one that is
ming language | Server | Android    | NULL      | Database       |
| 17 | A process of creating a data model for the data to be stored in a d
deling       | Coding | Data mining | NULL     | Data modeling  |
| 18 | The SQL commands that deals with the manipulation of data present i
             | DCL    | DDL        | NULL      | DML            |
| 19 | The command used for adding, deleting, dropping or modifying column
             | ALTER  | CREATE     | NULL      | ALTER          |
| 20 | Which word does not belong in the following list?
te           | Tuple  | Record     | NULL      | Attribute      |
| 21 | A function that operates on numeric data types and automatically ge
CREMENT       | int    | char       | NULL      | AUTO_INCREMENT |
+----+-----------------------------------------------------------------------
-------------+--------+------------+-----------+----------------+
7 rows in set (0.02 sec)

mysql> CREATE INDEX IND ON TEST(TITLE);
Query OK, 0 rows affected (0.45 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW INDEX FROM TEST;
+-------+------------+----------+--------------+-------------+-----------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation |
+-------+------------+----------+--------------+-------------+-----------+
| test  |          0 | PRIMARY  |            1 | NO          | A         |
| test  |          1 | IND      |            1 | TITLE       | A         |
+-------+------------+----------+--------------+-------------+-----------+
2 rows in set (0.07 sec)
```

Stored Procedure : MYSQL에서 제공해주는 프로그래밍 기능. 즉 SQL을 하나로 묶어서 편리하게 사용하는 기능

PROCEDURE

```
mysql> delimiter go
mysql> create procedure pr1()
    ->  begin
    ->   select * from price;
    ->   select * from fruits;
    -> end;
    -> delimiter go
Query OK, 0 rows affected (0.21 sec)

mysql> CALL PR1;
+----+--------+-------+
| No | name   | price |
+----+--------+-------+
|  1 | Pencil |  1000 |
|  2 | pEn    |  2000 |
|  3 | noTe   |   500 |
|  4 | maSk   |  1500 |
|  5 | Shose  |  5000 |
|  6 | pencil |  1000 |
|  7 | pen    |  2000 |
|  8 | note   |   500 |
|  9 | mask   |  1500 |
| 10 | shoes  |  5000 |
+----+--------+-------+
10 rows in set (0.01 sec)

+----+-------------+-------+------+
| NO | ITEM        | PRICE | ETC  |
+----+-------------+-------+------+
|  1 | APPLE       |  2000 | NULL |
|  2 | STRAWBERRY  |  3000 | NULL |
|  3 | PINEAPPLE   |  2500 | NULL |
|  4 | ORANGE      |  1500 | NULL |
|  5 | BANANA      |  4000 | NULL |
+----+-------------+-------+------+
5 rows in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)
```

```
mysql> SELECT * FROM STUDENT;
+----+-------+------+-----------+-------------------+
| NO | NAME  | ETC  | SALARY    | DEPARTMENT        |
+----+-------+------+-----------+-------------------+
|  1 | KIM   | NA   | 200000000 | CEO               |
|  2 | KIM3  | NA   |     50000 | STAFF             |
|  3 | WANG3 | NA   |  25000000 | ASSISTANT MANAGER |
|  4 | LEE   | NA   |    700000 | HEAD MANAGER      |
+----+-------+------+-----------+-------------------+
4 rows in set (0.02 sec)

mysql> DELIMITER //
mysql> CREATE PROCEDURE PROC()
    -> BEGIN
    -> SELECT NAME FROM STUDENT;
    -> SELECT NO FROM STUDENT WHERE SALARY = 50000;
    -> END//
Query OK, 0 rows affected (0.12 sec)

mysql> DELIMITER ;
mysql> ;
ERROR:
No query specified

mysql> CALL PROC;
+-------+
| NAME  |
+-------+
| KIM   |
| KIM3  |
| WANG3 |
| LEE   |
+-------+
4 rows in set (0.00 sec)

+----+
| NO |
+----+
|  2 |
+----+
```

```
mysql> SELECT * FROM ORDER_NUM;
+----+------------+---------+-------+
| ID | ORDER_MENU | TABLEID | PRICE |
+----+------------+---------+-------+
|  1 | 1          |       1 | 30000 |
|  2 | 10         |       4 | 55000 |
|  3 | 5          |       1 | 25000 |
|  4 | 6          |       4 | 15000 |
|  5 | 2          |       3 | 45000 |
|  6 | 3          |       3 | 18000 |
+----+------------+---------+-------+
6 rows in set (0.02 sec)

DELIMITER //
CREATE PROCEDURE PROCE()
BEGIN
SELECT ID FROM ORDER_NUM WHERE TABLEID = 4;
SELECT NO FROM STUDENT WHERE SALARY = 50000;
END//
OK, 0 rows affected (0.10 sec)

DELIMITER ;
+----+
| ID |
+----+
|  2 |
|  4 |
+----+
2 rows in set (0.01 sec)

+----+
| NO |
+----+
|  2 |
+----+
1 row in set (0.02 sec)
```