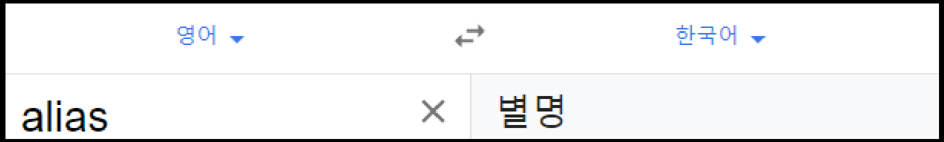
20200511 DB과제

AS(Alias), PROCEDURE, TRIGGER

B반 신석환

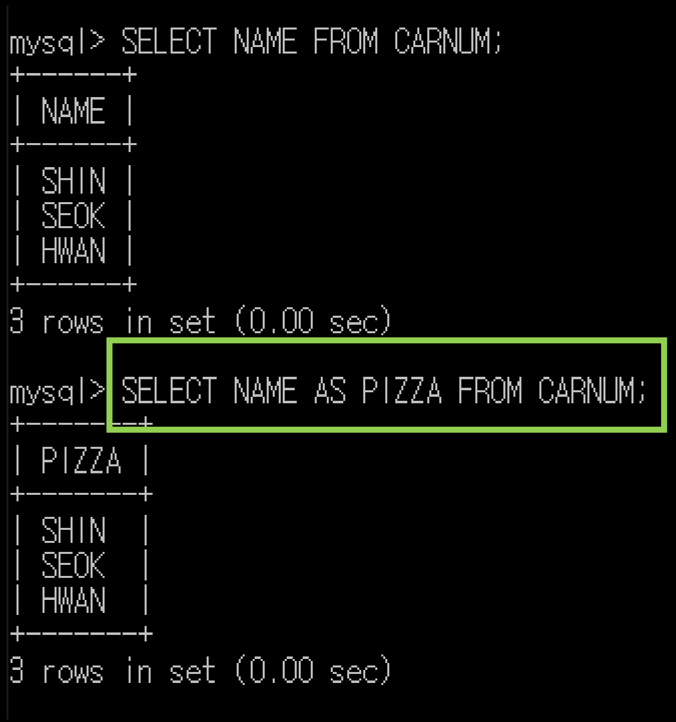
#AS(Alias)



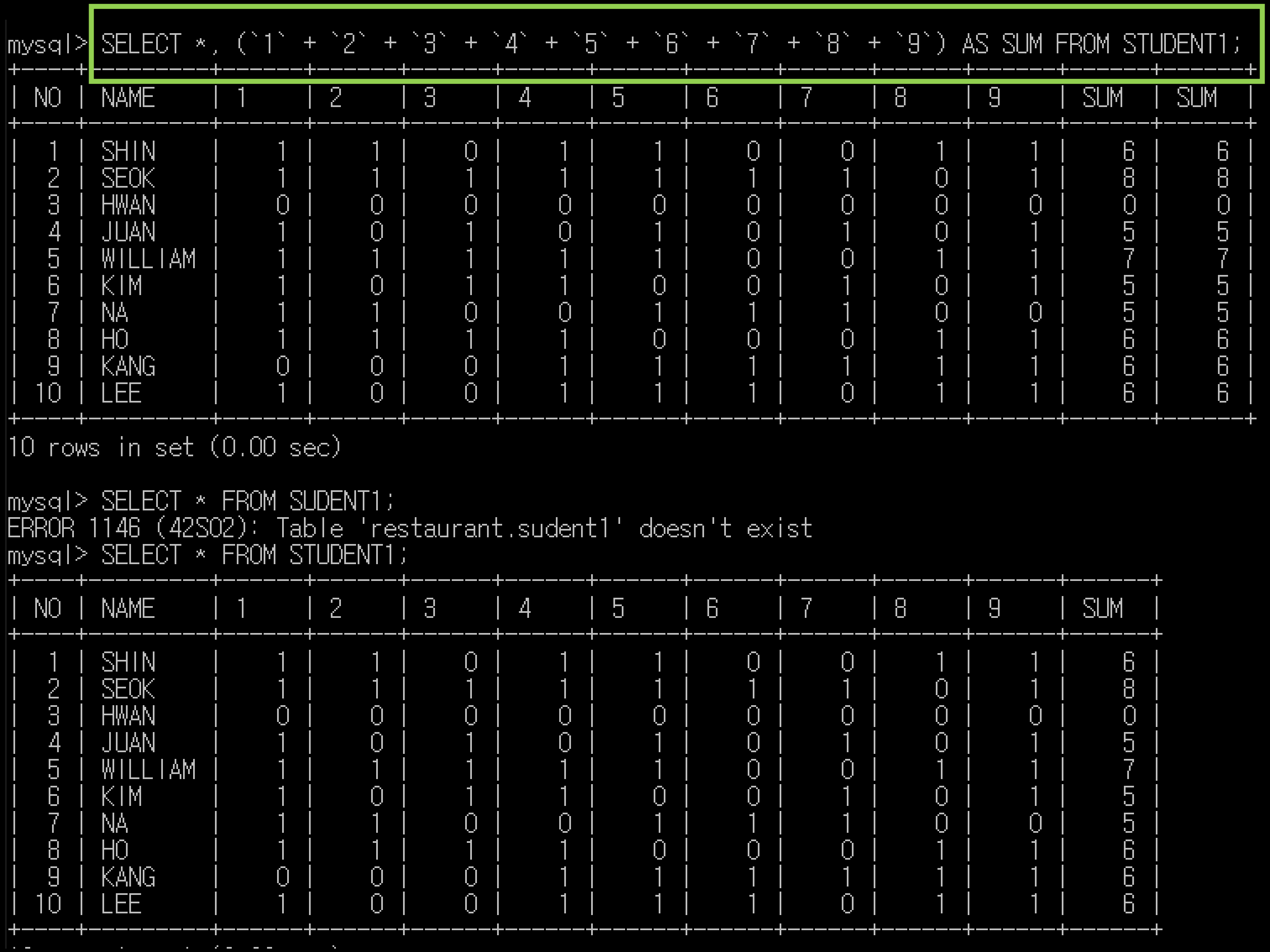
alias

별명

필드명에 별칭을 붙여서 출력 하는 것



```
mysql> SELECT NAME FROM CARNUM;
+------+
| NAME |
+------+
| SHIN |
| SEOK |
| HWAN |
+------+
3 rows in set (0.00 sec)

mysql> SELECT NAME AS PIZZA FROM CARNUM;
+-------+
| PIZZA |
+-------+
| SHIN  |
| SEOK  |
| HWAN  |
+-------+
3 rows in set (0.00 sec)
```

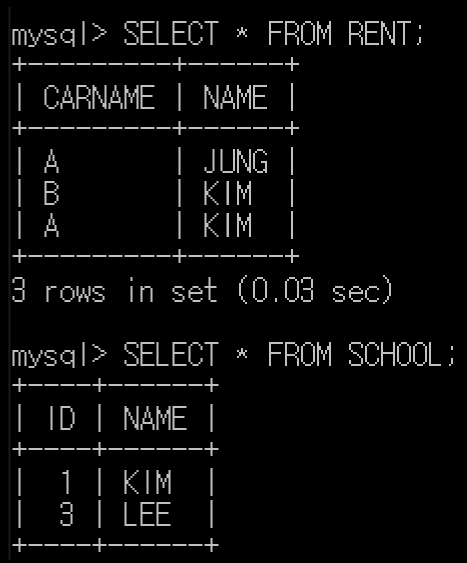


```
mysql> SELECT *, (`1` + `2` + `3` + `4` + `5` + `6` + `7` + `8` + `9`) AS SUM FROM STUDENT1;
+----+---------+------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM SUDENT1;
ERROR 1146 (42S02): Table 'restaurant.sudent1' doesn't exist
mysql> SELECT * FROM STUDENT1;
+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
```

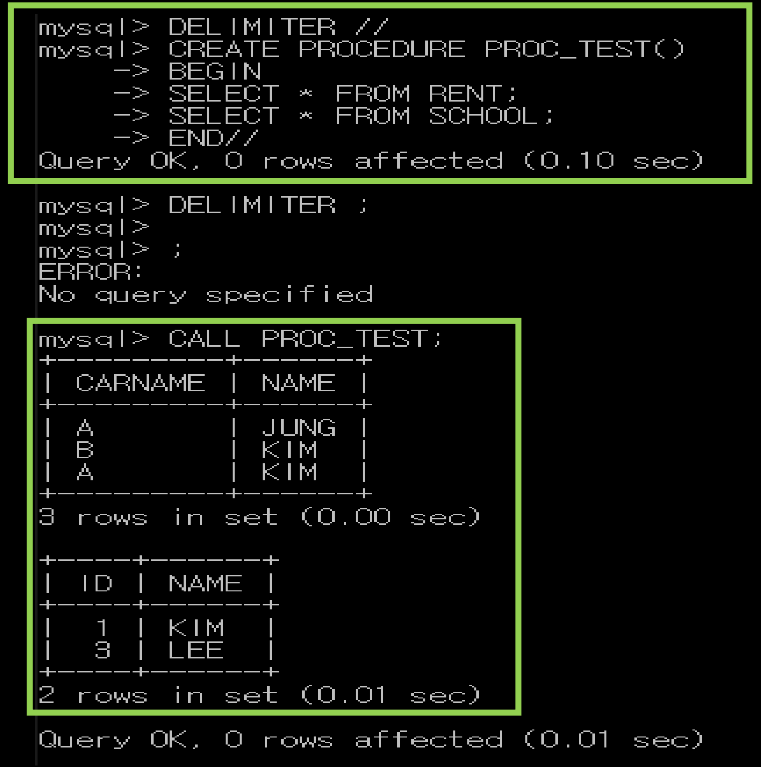
#Stored Procedure

여러 개의 동작을 실행 할 수 있게 만들어주는 것

두개의 테이블을 동시에 SELECT 할 수 있게
PROCEDURE을 만듬 (함수 역할)



```
mysql> SELECT * FROM RENT;
+---------+------+
| CARNAME | NAME |
+---------+------+
| A       | JUNG |
| B       | KIM  |
| A       | KIM  |
+---------+------+
3 rows in set (0.03 sec)

mysql> SELECT * FROM SCHOOL;
+----+------+
| ID | NAME |
+----+------+
|  1 | KIM  |
|  3 | LEE  |
+----+------+
```

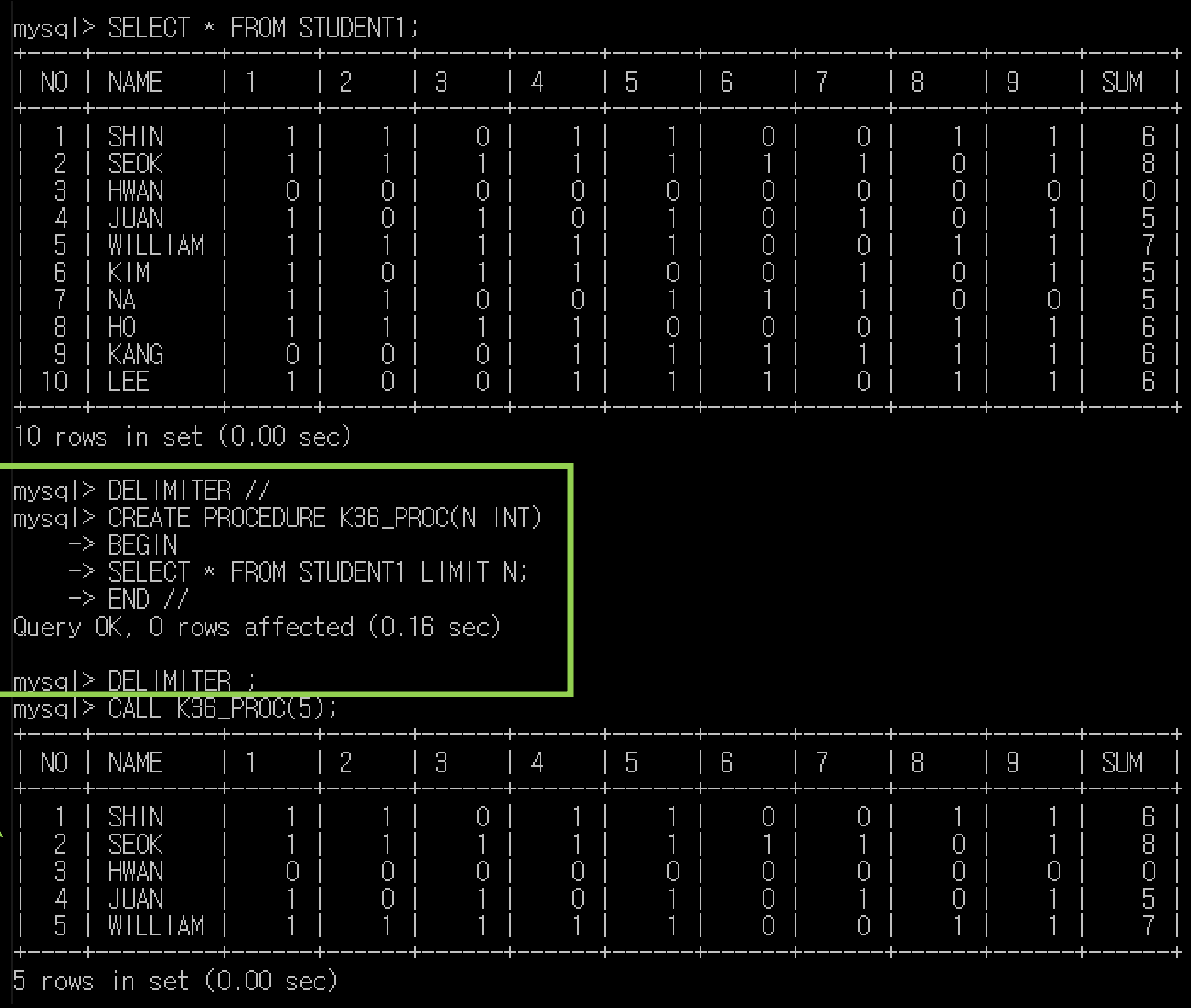


```
mysql> DELIMITER //
mysql> CREATE PROCEDURE PROC_TEST()
    -> BEGIN
    -> SELECT * FROM RENT;
    -> SELECT * FROM SCHOOL;
    -> END//
Query OK, 0 rows affected (0.10 sec)

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

mysql> CALL PROC_TEST;
+---------+------+
| CARNAME | NAME |
+---------+------+
| A       | JUNG |
| B       | KIM  |
| A       | KIM  |
+---------+------+
3 rows in set (0.00 sec)

+----+------+
| ID | NAME |
+----+------+
|  1 | KIM  |
|  3 | LEE  |
+----+------+
2 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

#Stored Procedure (매개변수 사용)

괄호 안에 매개변수를 입력하여 그에 맞는 값을 출력할 수 있다

CALL K36_PROC(5)를 입력하여
1~5까지 출력함

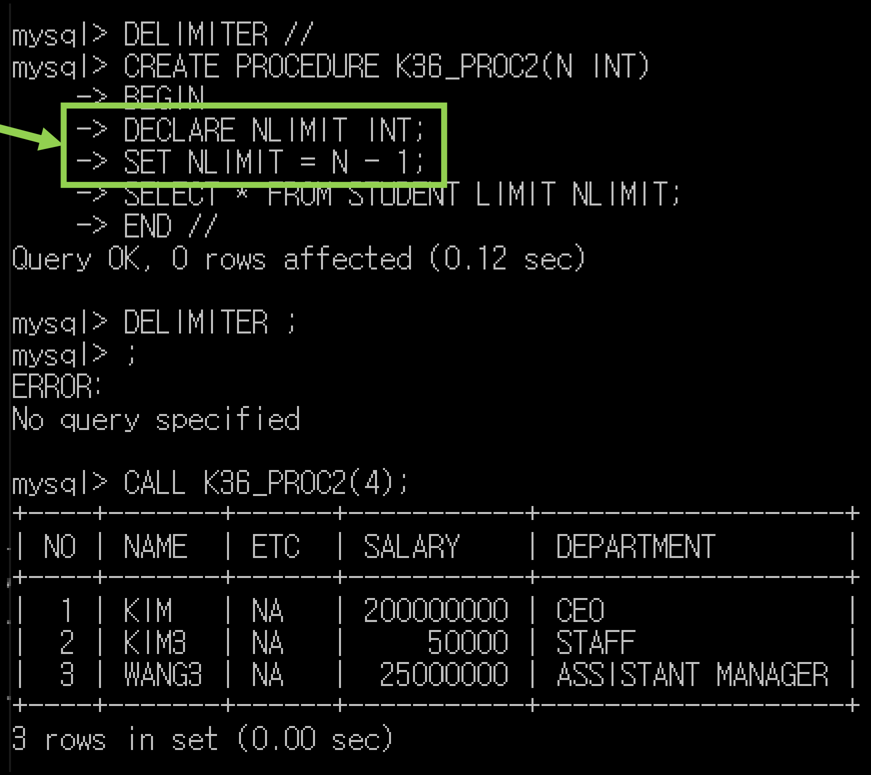


```
mysql> SELECT * FROM STUDENT1;
+----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
| NO | NAME    | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | SUM     |
+----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  1 | SHIN    |       1 |       1 |       0 |       1 |       1 |       0 |       0 |       1 |       1 |       6 |
|  2 | SEOK    |       1 |       1 |       1 |       1 |       1 |       1 |       1 |       0 |       1 |       8 |
|  3 | HWAN    |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |
|  4 | JUAN    |       1 |       0 |       1 |       0 |       1 |       0 |       1 |       0 |       1 |       5 |
|  5 | WILLIAM |       1 |       1 |       1 |       1 |       1 |       0 |       0 |       1 |       1 |       7 |
|  6 | KIM     |       1 |       0 |       1 |       1 |       0 |       0 |       1 |       0 |       1 |       5 |
|  7 | NA      |       1 |       1 |       0 |       0 |       1 |       1 |       1 |       0 |       0 |       5 |
|  8 | HO      |       1 |       1 |       1 |       1 |       0 |       0 |       0 |       1 |       1 |       6 |
|  9 | KANG    |       0 |       0 |       0 |       1 |       1 |       1 |       1 |       1 |       1 |       6 |
| 10 | LEE     |       1 |       0 |       0 |       1 |       1 |       1 |       0 |       1 |       1 |       6 |
+----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
10 rows in set (0.00 sec)

mysql> DELIMITER //
mysql> CREATE PROCEDURE K36_PROC(N INT)
    -> BEGIN
    -> SELECT * FROM STUDENT1 LIMIT N;
    -> END //
Query OK, 0 rows affected (0.16 sec)

mysql> DELIMITER ;
mysql> CALL K36_PROC(5);
+----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
| NO | NAME    | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | SUM     |
+----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  1 | SHIN    |       1 |       1 |       0 |       1 |       1 |       0 |       0 |       1 |       1 |       6 |
|  2 | SEOK    |       1 |       1 |       1 |       1 |       1 |       1 |       1 |       0 |       1 |       8 |
|  3 | HWAN    |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |       0 |
|  4 | JUAN    |       1 |       0 |       1 |       0 |       1 |       0 |       1 |       0 |       1 |       5 |
|  5 | WILLIAM |       1 |       1 |       1 |       1 |       1 |       0 |       0 |       1 |       1 |       7 |
+----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
5 rows in set (0.00 sec)
```

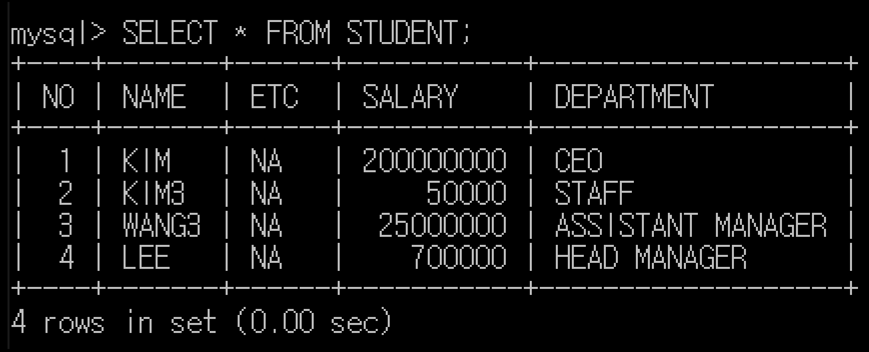
#Stored Procedure (변수 사용)

DECLARE 뒤에 변수를 선언하고

SET 으로 변수 값을 정한다.



```
mysql> DELIMITER //
mysql> CREATE PROCEDURE K36_PROC2(N INT)
    -> BEGIN
    -> DECLARE NLIMIT INT;
    -> SET NLIMIT = N - 1;
    -> SELECT * FROM STUDENT LIMIT NLIMIT;
    -> END //
Query OK, 0 rows affected (0.12 sec)

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

mysql> CALL K36_PROC2(4);
+----+-------+------+-----------+-------------------+
| NO | NAME  | ETC  | SALARY    | DEPARTMENT        |
+----+-------+------+-----------+-------------------+
|  1 | KIM   | NA   | 200000000 | CEO               |
|  2 | KIM3  | NA   |     50000 | STAFF             |
|  3 | WANG3 | NA   |  25000000 | ASSISTANT MANAGER |
+----+-------+------+-----------+-------------------+
3 rows in set (0.00 sec)
```

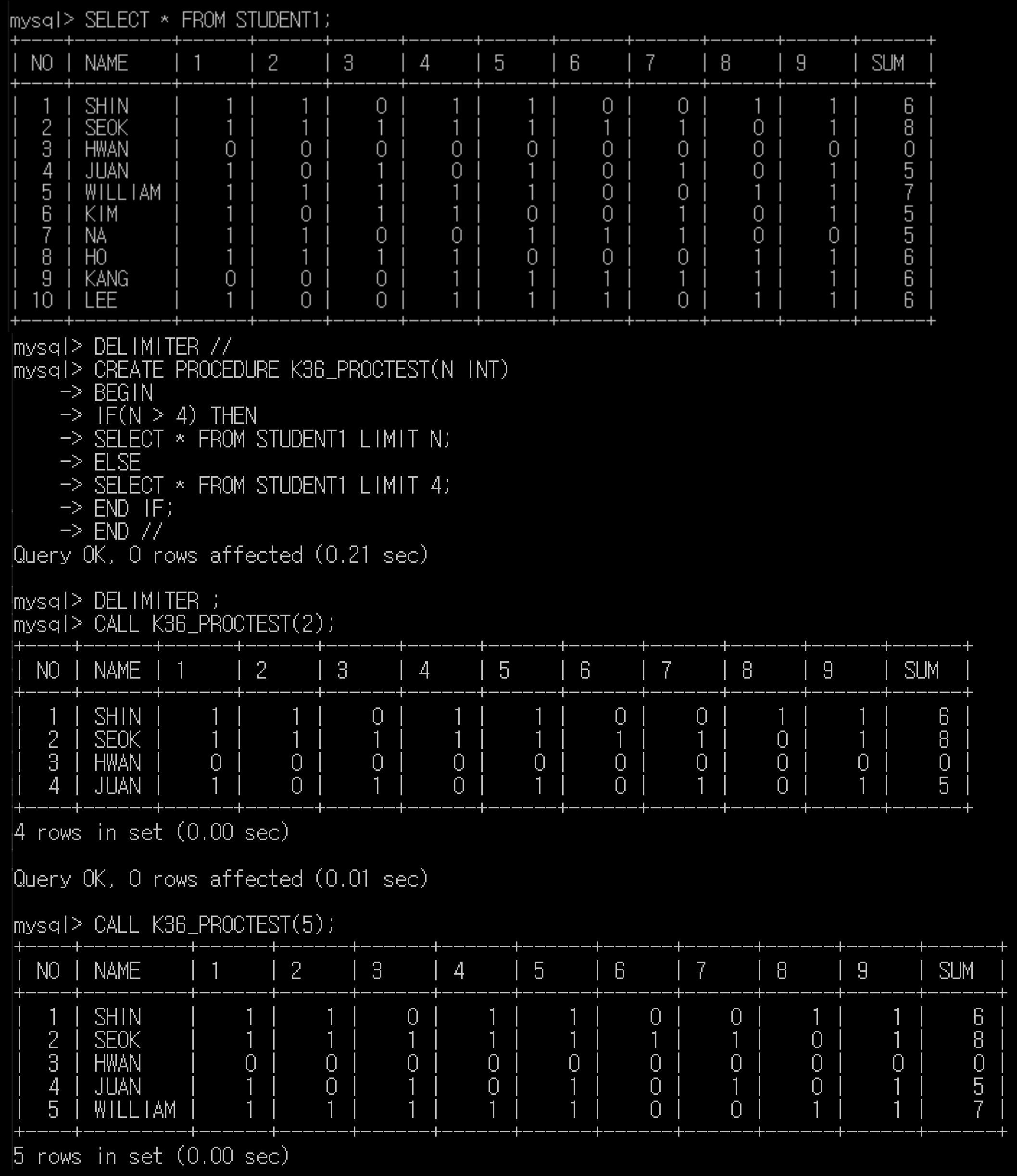


```
mysql> SELECT * FROM STUDENT;
+----+-------+------+-----------+-------------------+
| NO | NAME  | ETC  | SALARY    | DEPARTMENT        |
+----+-------+------+-----------+-------------------+
|  1 | KIM   | NA   | 200000000 | CEO               |
|  2 | KIM3  | NA   |     50000 | STAFF             |
|  3 | WANG3 | NA   |  25000000 | ASSISTANT MANAGER |
|  4 | LEE   | NA   |    700000 | HEAD MANAGER      |
+----+-------+------+-----------+-------------------+
4 rows in set (0.00 sec)
```

#Stored Procedure (조건문)

조건문을 사용하여 Procedure을 만들 수 있다

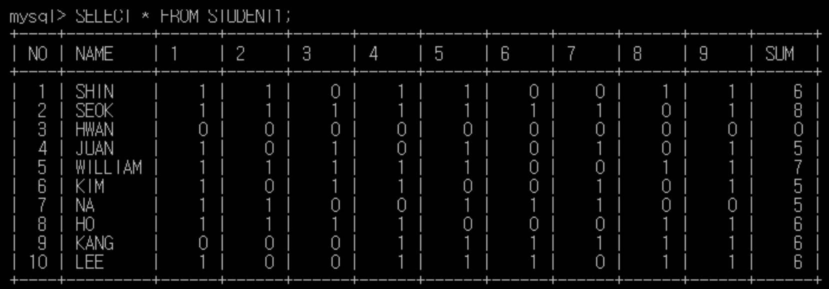
```
DELIMITER //
CREATE PROCEDURE K36_TEST(매개변수)
BEGIN
IF(조건) THEN
넣고 싶은 값
ELSE
넣고 싶은 값
END IF;
END //
DELIMITER ;

CALL K36_TEST(?)
```



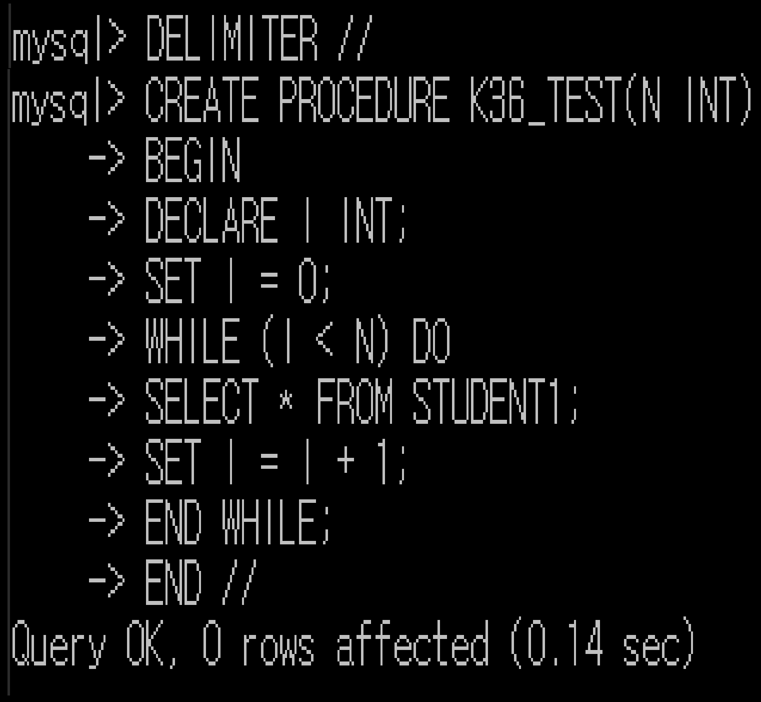
```
mysql> SELECT * FROM STUDENT1;
+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
mysql> DELIMITER //
mysql> CREATE PROCEDURE K36_PROCTEST(N INT)
    -> BEGIN
    -> IF(N > 4) THEN
    -> SELECT * FROM STUDENT1 LIMIT N;
    -> ELSE
    -> SELECT * FROM STUDENT1 LIMIT 4;
    -> END IF;
    -> END //
Query OK, 0 rows affected (0.21 sec)

mysql> DELIMITER ;
mysql> CALL K36_PROCTEST(2);
+----+------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
+----+------+------+------+------+------+------+------+------+------+------+------+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL K36_PROCTEST(5);
+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
5 rows in set (0.00 sec)
```

#Stored Procedure (반복문)

반복문을 사용하여 Procedure을 만들 수 있다



```
mysql> SELECT * FROM STUDENT1;
+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
```



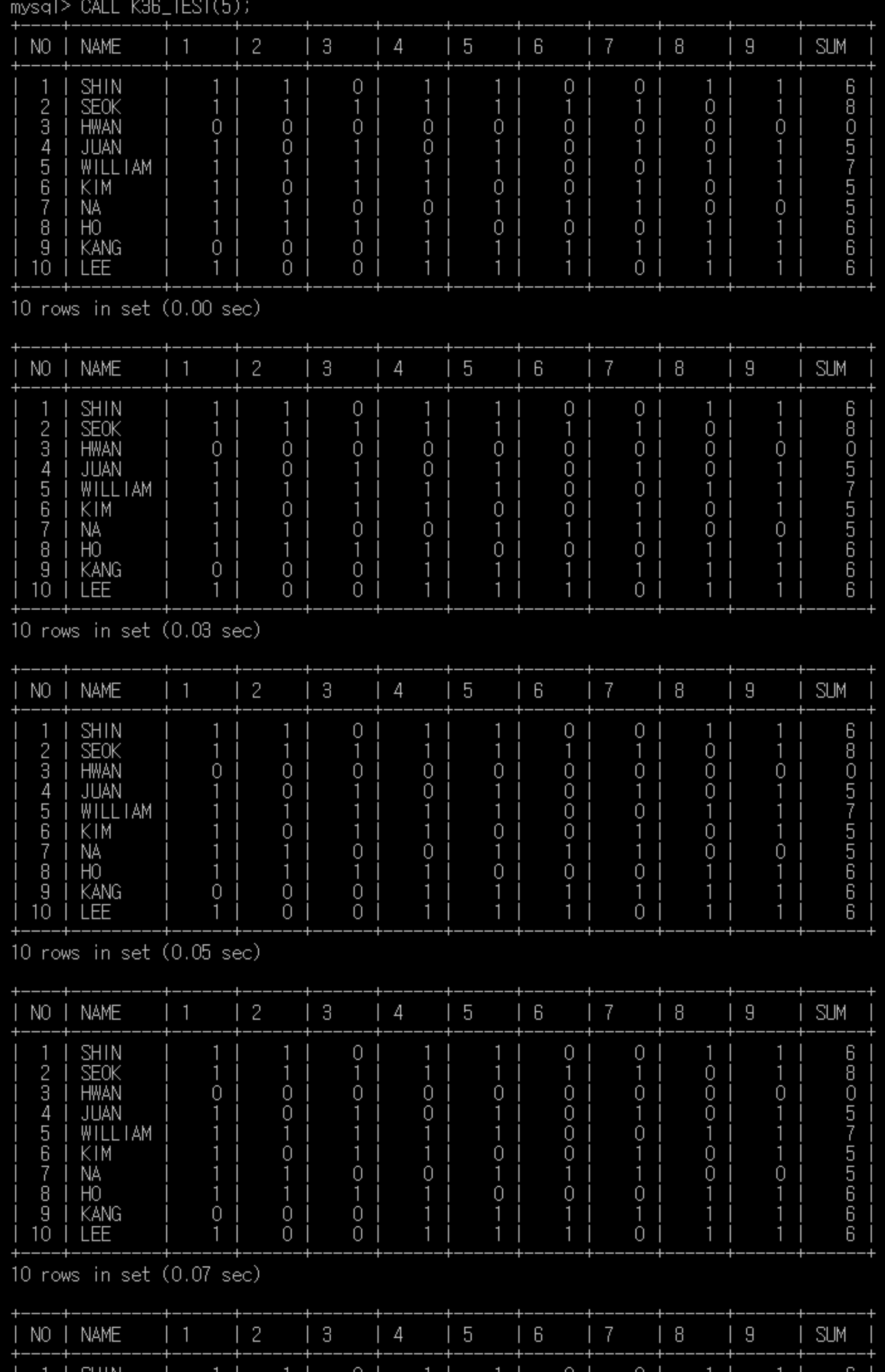
```
mysql> DELIMITER //
mysql> CREATE PROCEDURE K36_TEST(N INT)
    -> BEGIN
    -> DECLARE I INT;
    -> SET I = 0;
    -> WHILE (I < N) DO
    -> SELECT * FROM STUDENT1;
    -> SET I = I + 1;
    -> END WHILE;
    -> END //
Query OK, 0 rows affected (0.14 sec)
```

-DECLARE로 변수 선언하고

-SET으로 변수의 값을 정해주고

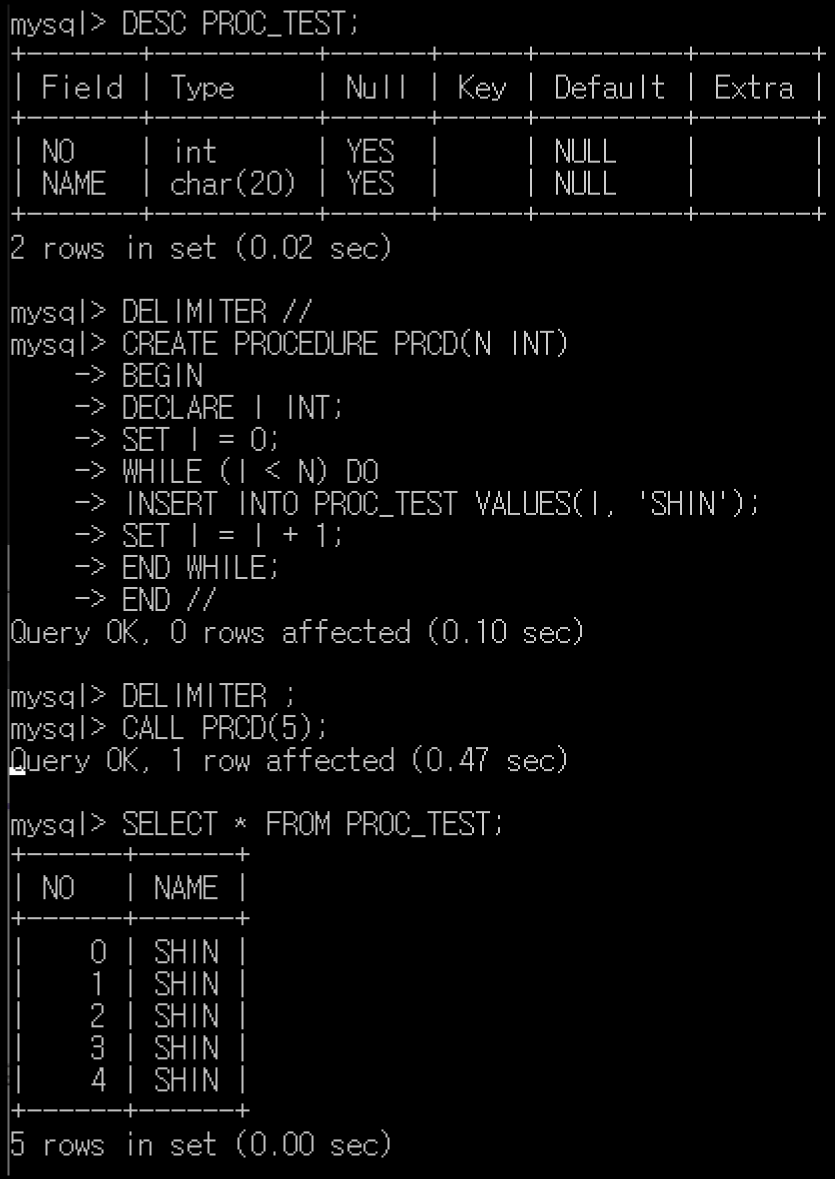
-WHILE로 I < N이면 계속 실행하는 쿼리를 작성한다

DECLARE 변수창출
SET 변수 값 지정
WHILE(조건) DO
쿼리
END WHILE
END //
DELIMITER ;

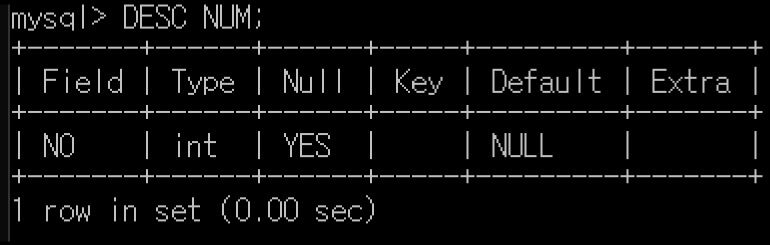


```
mysql> CALL K36_TEST(5);
+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
10 rows in set (0.00 sec)

+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
10 rows in set (0.03 sec)

+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
10 rows in set (0.05 sec)

+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
|  2 | SEOK    |    1 |    1 |    1 |    1 |    1 |    1 |    1 |    0 |    1 |    8 |
|  3 | HWAN    |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|  4 | JUAN    |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    0 |    1 |    5 |
|  5 | WILLIAM |    1 |    1 |    1 |    1 |    1 |    0 |    0 |    1 |    1 |    7 |
|  6 | KIM     |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    0 |    1 |    5 |
|  7 | NA      |    1 |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    0 |    5 |
|  8 | HO      |    1 |    1 |    1 |    1 |    0 |    0 |    0 |    1 |    1 |    6 |
|  9 | KANG    |    0 |    0 |    0 |    1 |    1 |    1 |    1 |    1 |    1 |    6 |
| 10 | LEE     |    1 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |    1 |    6 |
+----+---------+------+------+------+------+------+------+------+------+------+------+
10 rows in set (0.07 sec)

+----+---------+------+------+------+------+------+------+------+------+------+------+
| NO | NAME    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | SUM  |
+----+---------+------+------+------+------+------+------+------+------+------+------+
|  1 | SHIN    |    1 |    1 |    0 |    1 |    1 |    0 |    0 |    1 |    1 |    6 |
```

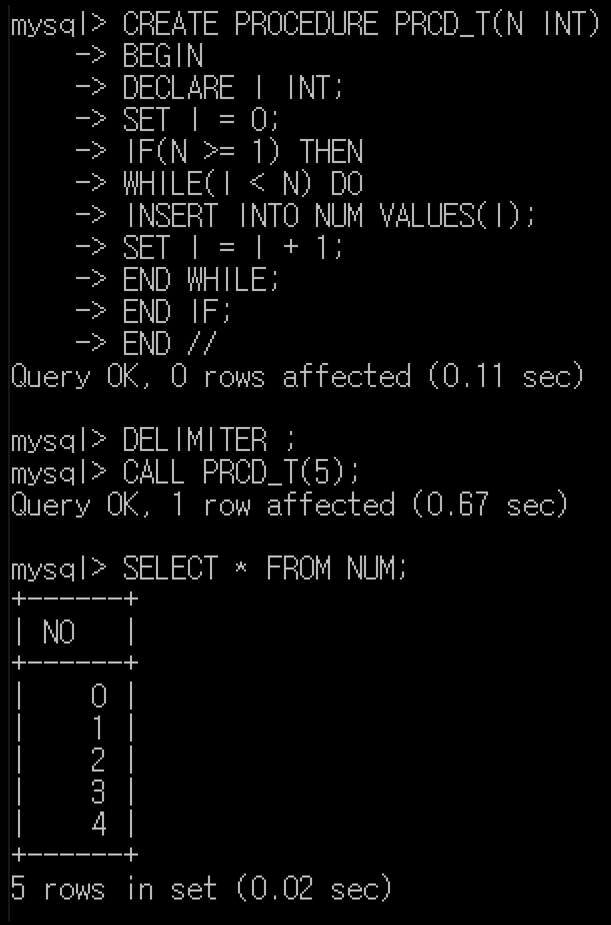
#Stored Procedure 실습



```
mysql> DESC PROC_TEST;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| NO    | int      | YES  |     | NULL    |       |
| NAME  | char(20) | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
2 rows in set (0.02 sec)

mysql> DELIMITER //
mysql> CREATE PROCEDURE PRCD(N INT)
    -> BEGIN
    -> DECLARE I INT;
    -> SET I = 0;
    -> WHILE (I < N) DO
    -> INSERT INTO PROC_TEST VALUES(I, 'SHIN');
    -> SET I = I + 1;
    -> END WHILE;
    -> END //
Query OK, 0 rows affected (0.10 sec)

mysql> DELIMITER ;
mysql> CALL PRCD(5);
Query OK, 1 row affected (0.47 sec)

mysql> SELECT * FROM PROC_TEST;
+------+------+
| NO   | NAME |
+------+------+
|    0 | SHIN |
|    1 | SHIN |
|    2 | SHIN |
|    3 | SHIN |
|    4 | SHIN |
+------+------+
5 rows in set (0.00 sec)
```



```
mysql> DESC NUM;
+-------+------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+------+------+-----+---------+-------+
| NO    | int  | YES  |     | NULL    |       |
+-------+------+------+-----+---------+-------+
1 row in set (0.00 sec)
```



```
mysql> CREATE PROCEDURE PRCD_T(N INT)
    -> BEGIN
    -> DECLARE I INT;
    -> SET I = 0;
    -> IF(N >= 1) THEN
    -> WHILE(I < N) DO
    -> INSERT INTO NUM VALUES(I);
    -> SET I = I + 1;
    -> END WHILE;
    -> END IF;
    -> END //
Query OK, 0 rows affected (0.11 sec)

mysql> DELIMITER ;
mysql> CALL PRCD_T(5);
Query OK, 1 row affected (0.67 sec)

mysql> SELECT * FROM NUM;
+------+
| NO   |
+------+
|    0 |
|    1 |
|    2 |
|    3 |
|    4 |
+------+
5 rows in set (0.02 sec)
```

#Trigger (after)

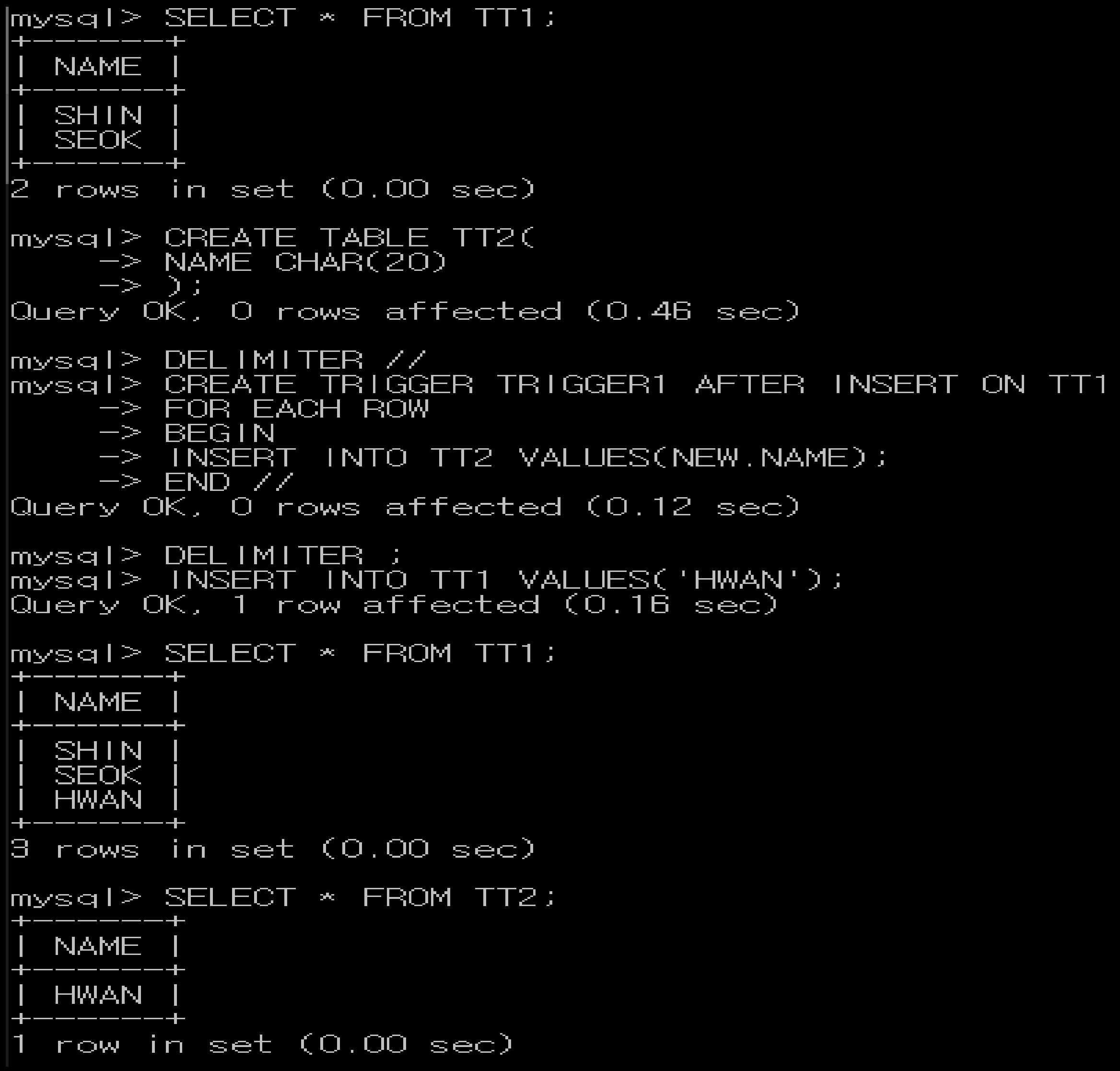
특정 이벤트에 대해 자동으로 실행되는 작업(before, after, instead of, when 등)

*공식

```
DELIMITER //
CREATE TRIGGER 트리거이름 AFTER INSERT ON 테이블1

FOR EACH ROW
BEGIN
INSERT INTO 테이블2 VALUES(NEW.NAME);

END //

DELIMITER ;
```

-테이블 1에 INSERT한 후에 트리거를 실행한다.

-테이블 2에 테이블 1에 새로(NEW) INSERT한 이름(NAME)을 INSERT한다.



```
mysql> SELECT * FROM TT1;
+--------+
| NAME  |
+--------+
| SHIN  |
| SEOK  |
+--------+
2 rows in set (0.00 sec)

mysql> CREATE TABLE TT2(
    -> NAME CHAR(20)
    -> );
Query OK, 0 rows affected (0.46 sec)

mysql> DELIMITER //
mysql> CREATE TRIGGER TRIGGER1 AFTER INSERT ON TT1
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO TT2 VALUES(NEW.NAME);
    -> END //
Query OK, 0 rows affected (0.12 sec)

mysql> DELIMITER ;
mysql> INSERT INTO TT1 VALUES('HWAN');
Query OK, 1 row affected (0.16 sec)

mysql> SELECT * FROM TT1;
+--------+
| NAME  |
+--------+
| SHIN  |
| SEOK  |
| HWAN  |
+--------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM TT2;
+--------+
| NAME  |
+--------+
| HWAN  |
+--------+
1 row in set (0.00 sec)
```

#Trigger (after)

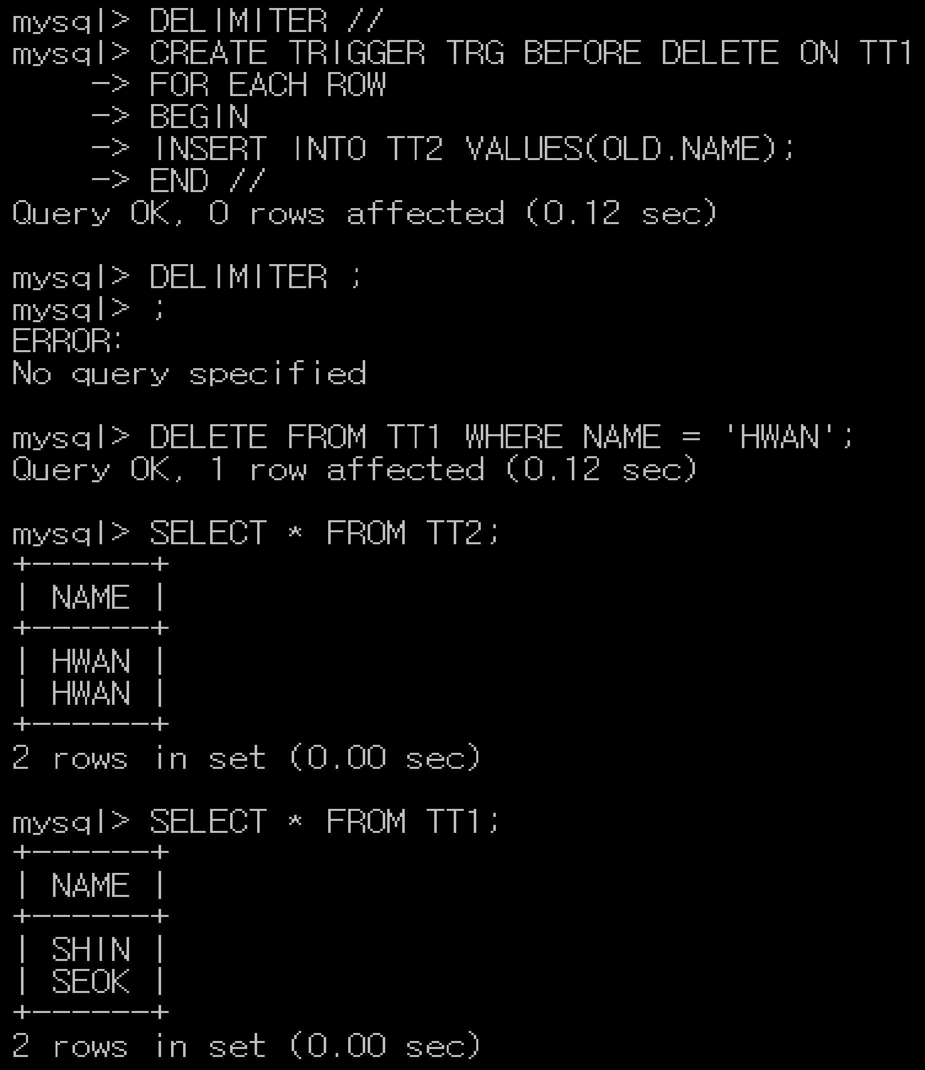
특정 이벤트에 대해 자동으로 실행되는 작업(before, after, instead of, when 등)

*공식

```
DELIMITER //
CREATE TRIGGER 트리거이름 BEFORE DELETE ON 테이블1

FOR EACH ROW
BEGIN
INSERT INTO 테이블2 VALUES(OLD.NAME);

END //

DELIMITER ;
```

-테이블 1에 DELETE 하기 전에트리거를 실행한다.

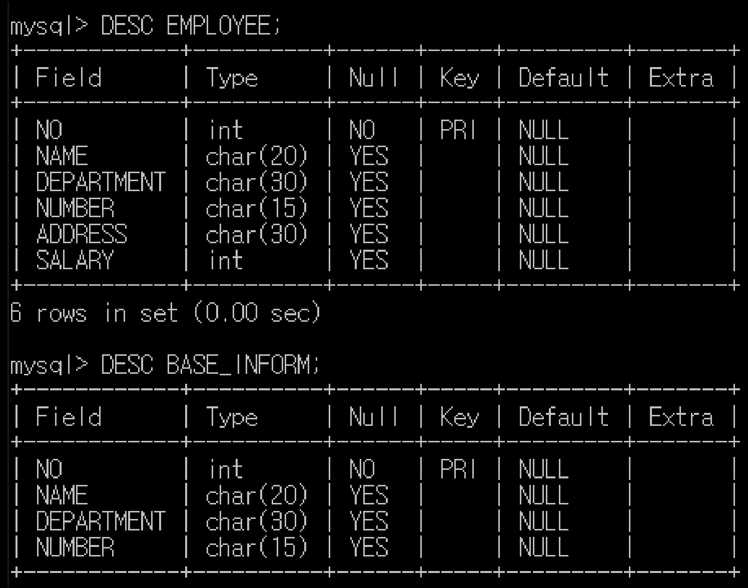
-테이블 2에 테이블 1에 있던(OLD) 이름(NAME)
을 테이블 1에서 DELETE 하기 전에 INSERT한다.



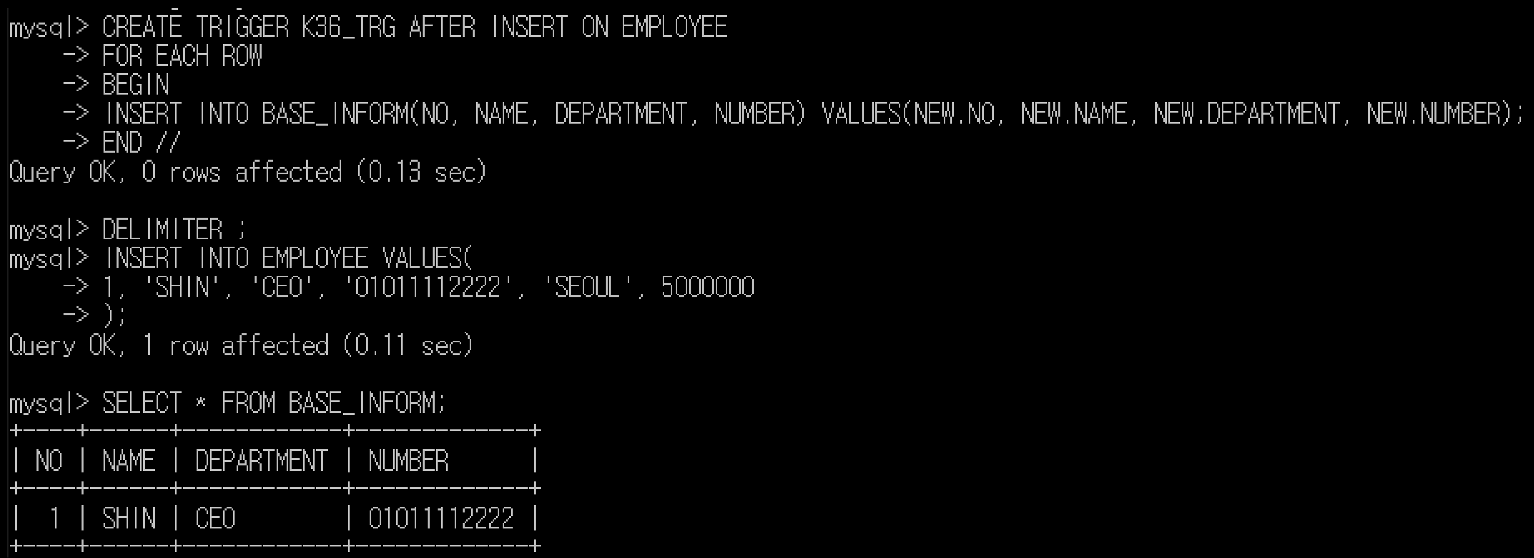
```
mysql> DELIMITER //
mysql> CREATE TRIGGER TRG BEFORE DELETE ON TT1
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO TT2 VALUES(OLD.NAME);
    -> END //
Query OK, 0 rows affected (0.12 sec)

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

mysql> DELETE FROM TT1 WHERE NAME = 'HWAN';
Query OK, 1 row affected (0.12 sec)

mysql> SELECT * FROM TT2;
+-------+
| NAME  |
+-------+
| HWAN  |
| HWAN  |
+-------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM TT1;
+-------+
| NAME  |
+-------+
| SHIN  |
| SEOK  |
+-------+
2 rows in set (0.00 sec)
```

#Trigger 실습

- 테이블 두개를 만들고
- 테이블 EMPLOYEE에 입력된 후에 BASE_INFORM에 같은 정보를 입력해주는 트리거 생성



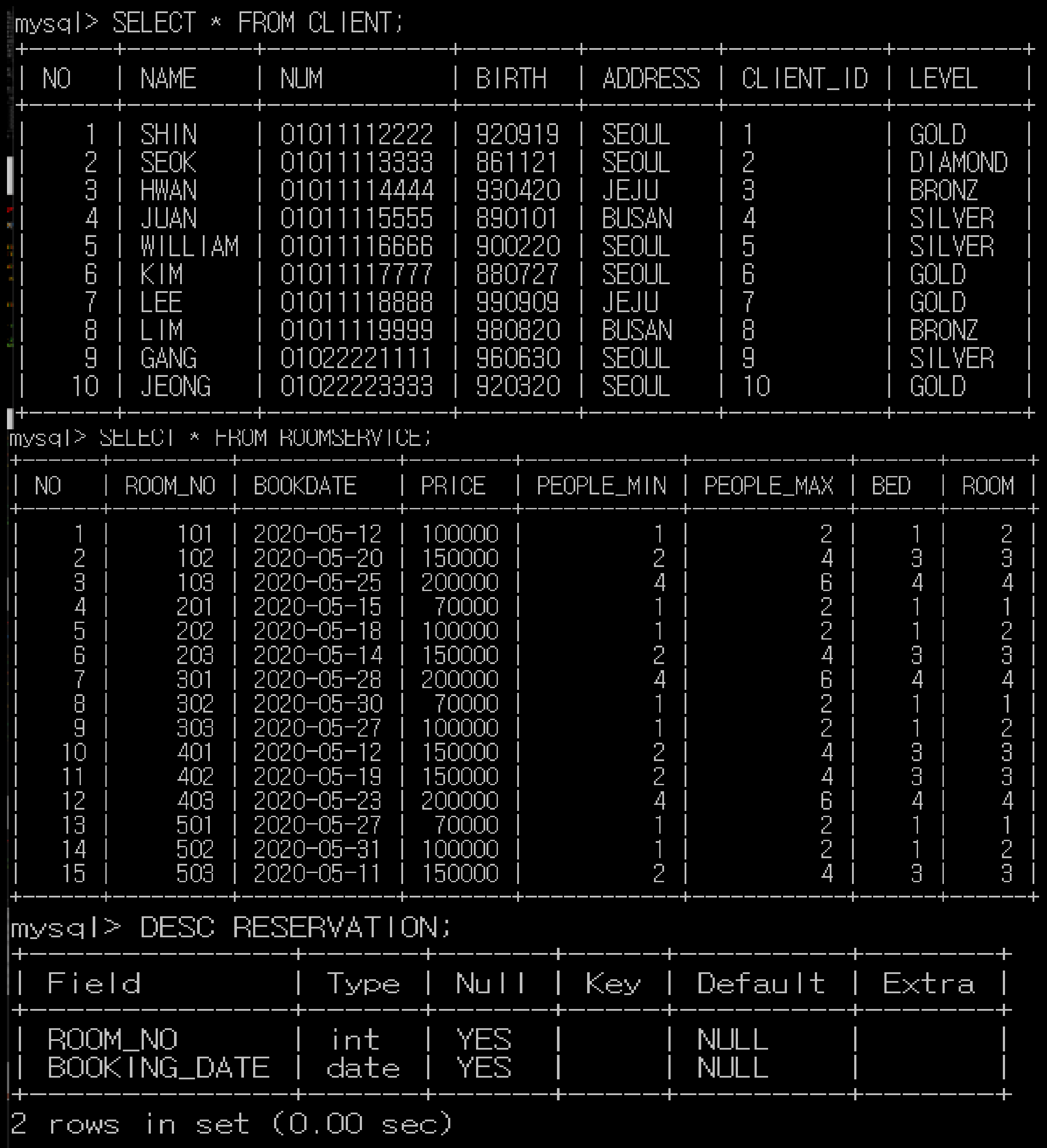
```
mysql> DESC EMPLOYEE;
+------------+----------+------+-----+---------+-------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+---------+-------+
| NO         | int      | NO   | PRI | NULL    |       |
| NAME       | char(20) | YES  |     | NULL    |       |
| DEPARTMENT | char(30) | YES  |     | NULL    |       |
| NUMBER     | char(15) | YES  |     | NULL    |       |
| ADDRESS    | char(30) | YES  |     | NULL    |       |
| SALARY     | int      | YES  |     | NULL    |       |
+------------+----------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> DESC BASE_INFORM;
+------------+----------+------+-----+---------+-------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+---------+-------+
| NO         | int      | NO   | PRI | NULL    |       |
| NAME       | char(20) | YES  |     | NULL    |       |
| DEPARTMENT | char(30) | YES  |     | NULL    |       |
| NUMBER     | char(15) | YES  |     | NULL    |       |
+------------+----------+------+-----+---------+-------+
```



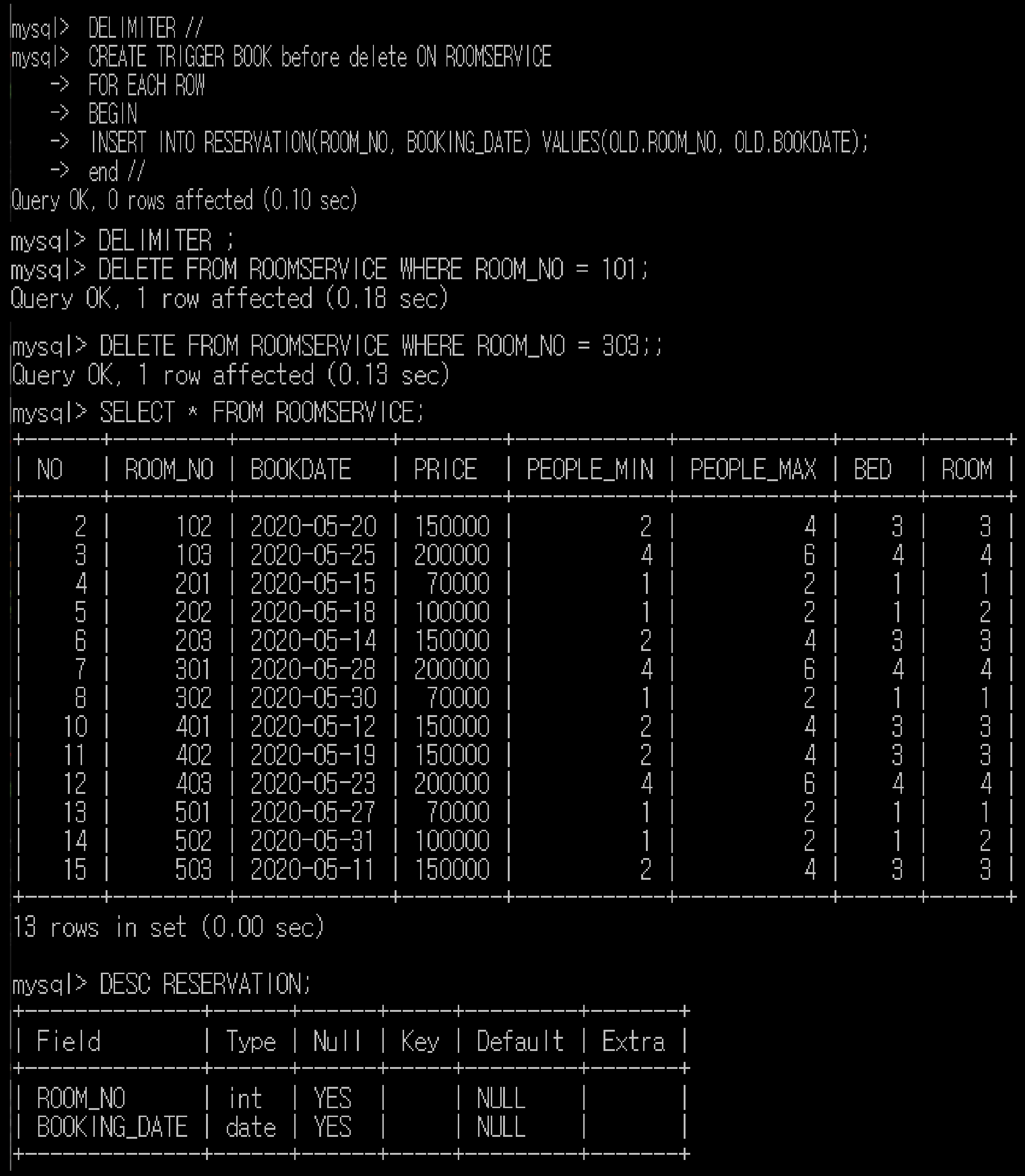
```
mysql> CREATE TRIGGER K36_TRG AFTER INSERT ON EMPLOYEE
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO BASE_INFORM(NO, NAME, DEPARTMENT, NUMBER) VALUES(NEW.NO, NEW.NAME, NEW.DEPARTMENT, NEW.NUMBER);
    -> END //
Query OK, 0 rows affected (0.13 sec)

mysql> DELIMITER ;
mysql> INSERT INTO EMPLOYEE VALUES(
    -> 1, 'SHIN', 'CEO', '01011112222', 'SEOUL', 5000000
    -> );
Query OK, 1 row affected (0.11 sec)

mysql> SELECT * FROM BASE_INFORM;
+----+------+------------+-------------+
| NO | NAME | DEPARTMENT | NUMBER      |
+----+------+------------+-------------+
|  1 | SHIN | CEO        | 01011112222 |
+----+------+------------+-------------+
```

#복습과제

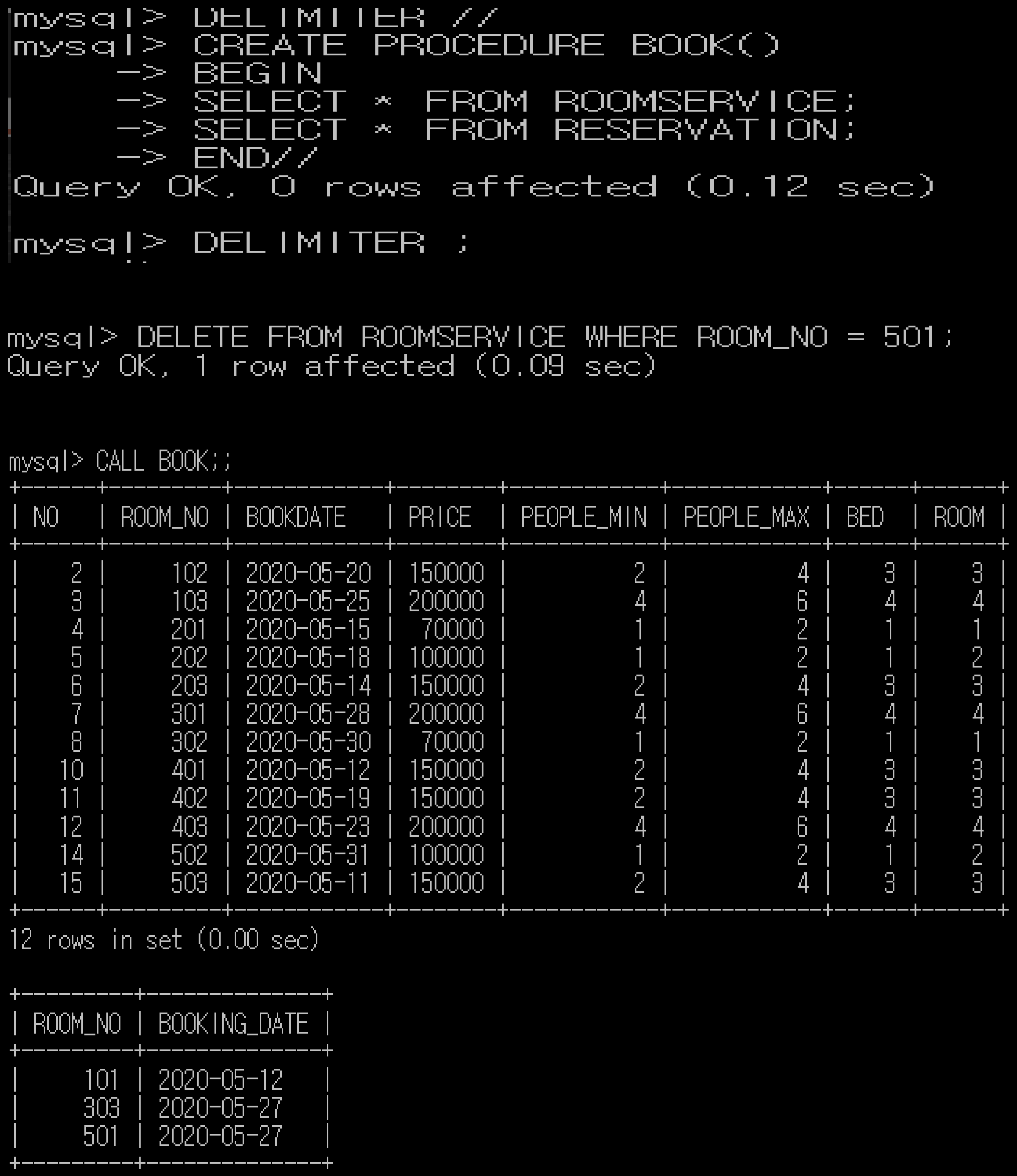
- 호텔 예약 Table 만들기
  1. 회원 명부 Table(Column 7개 이상)
  2. 호텔 Room 관리 Table(Column 7개 이상)
  3. 데이터는 아래와 같이 넣는다
     1) 회원 최소 10명 이상
     2) Room 최소 10개 이상, 예약가능일 최소 15일 이상
     3) 최소 7건 이상 예약처리함
  4. 아래 내용을 출력한다
     1) 회원 명부 리스트 전체 출력
     2) Room 리스트 전체 출력
     3) Room 예약된 것만 전체 출력



```
mysql> SELECT * FROM CLIENT;
+------+---------+-------------+--------+---------+-----------+---------+
| NO   | NAME    | NUM         | BIRTH  | ADDRESS | CLIENT_ID | LEVEL   |
+------+---------+-------------+--------+---------+-----------+---------+
|    1 | SHIN    | 01011112222 | 920919 | SEOUL   | 1         | GOLD    |
|    2 | SEOK    | 01011113333 | 861121 | SEOUL   | 2         | DIAMOND |
|    3 | HWAN    | 01011114444 | 930420 | JEJU    | 3         | BRONZ   |
|    4 | JUAN    | 01011115555 | 890101 | BUSAN   | 4         | SILVER  |
|    5 | WILLIAM | 01011116666 | 900220 | SEOUL   | 5         | SILVER  |
|    6 | KIM     | 01011117777 | 880727 | SEOUL   | 6         | GOLD    |
|    7 | LEE     | 01011118888 | 990909 | JEJU    | 7         | GOLD    |
|    8 | LIM     | 01011119999 | 980820 | BUSAN   | 8         | BRONZ   |
|    9 | GANG    | 01022221111 | 960630 | SEOUL   | 9         | SILVER  |
|   10 | JEONG   | 01022223333 | 920320 | SEOUL   | 10        | GOLD    |
+------+---------+-------------+--------+---------+-----------+---------+
mysql> SELECT * FROM ROOMSERVICE;
+-----+---------+------------+--------+------------+------------+-----+------+
| NO  | ROOM_NO | BOOKDATE   | PRICE  | PEOPLE_MIN | PEOPLE_MAX | BED | ROOM |
+-----+---------+------------+--------+------------+------------+-----+------+
|   1 |     101 | 2020-05-12 | 100000 |          1 |          2 |   1 |    2 |
|   2 |     102 | 2020-05-20 | 150000 |          2 |          4 |   3 |    3 |
|   3 |     103 | 2020-05-25 | 200000 |          4 |          6 |   4 |    4 |
|   4 |     201 | 2020-05-15 |  70000 |          1 |          2 |   1 |    1 |
|   5 |     202 | 2020-05-18 | 100000 |          1 |          2 |   1 |    2 |
|   6 |     203 | 2020-05-14 | 150000 |          2 |          4 |   3 |    3 |
|   7 |     301 | 2020-05-28 | 200000 |          4 |          6 |   4 |    4 |
|   8 |     302 | 2020-05-30 |  70000 |          1 |          2 |   1 |    1 |
|   9 |     303 | 2020-05-27 | 100000 |          1 |          2 |   1 |    2 |
|  10 |     401 | 2020-05-12 | 150000 |          2 |          4 |   3 |    3 |
|  11 |     402 | 2020-05-19 | 150000 |          2 |          4 |   3 |    3 |
|  12 |     403 | 2020-05-23 | 200000 |          4 |          6 |   4 |    4 |
|  13 |     501 | 2020-05-27 |  70000 |          1 |          2 |   1 |    1 |
|  14 |     502 | 2020-05-31 | 100000 |          1 |          2 |   1 |    2 |
|  15 |     503 | 2020-05-11 | 150000 |          2 |          4 |   3 |    3 |
+-----+---------+------------+--------+------------+------------+-----+------+
mysql> DESC RESERVATION;
+--------------+------+------+-----+---------+-------+
| Field        | Type | Null | Key | Default | Extra |
+--------------+------+------+-----+---------+-------+
| ROOM_NO      | int  | YES  |     | NULL    |       |
| BOOKING_DATE | date | YES  |     | NULL    |       |
+--------------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



```
mysql> DELIMITER //
mysql> CREATE TRIGGER BOOK before delete ON ROOMSERVICE
    -> FOR EACH ROW
    -> BEGIN
    -> INSERT INTO RESERVATION(ROOM_NO, BOOKING_DATE) VALUES(OLD.ROOM_NO, OLD.BOOKDATE);
    -> end //
Query OK, 0 rows affected (0.10 sec)

mysql> DELIMITER ;
mysql> DELETE FROM ROOMSERVICE WHERE ROOM_NO = 101;
Query OK, 1 row affected (0.18 sec)

mysql> DELETE FROM ROOMSERVICE WHERE ROOM_NO = 303;;
Query OK, 1 row affected (0.13 sec)

mysql> SELECT * FROM ROOMSERVICE;
+------+---------+------------+--------+------------+------------+------+------+
| NO   | ROOM_NO | BOOKDATE   | PRICE  | PEOPLE_MIN | PEOPLE_MAX | BED  | ROOM |
+------+---------+------------+--------+------------+------------+------+------+
|    2 |     102 | 2020-05-20 | 150000 |          2 |          4 |    3 |    3 |
|    3 |     103 | 2020-05-25 | 200000 |          4 |          6 |    4 |    4 |
|    4 |     201 | 2020-05-15 |  70000 |          1 |          2 |    1 |    1 |
|    5 |     202 | 2020-05-18 | 100000 |          1 |          2 |    1 |    2 |
|    6 |     203 | 2020-05-14 | 150000 |          2 |          4 |    3 |    3 |
|    7 |     301 | 2020-05-28 | 200000 |          4 |          6 |    4 |    4 |
|    8 |     302 | 2020-05-30 |  70000 |          1 |          2 |    1 |    1 |
|   10 |     401 | 2020-05-12 | 150000 |          2 |          4 |    3 |    3 |
|   11 |     402 | 2020-05-19 | 150000 |          2 |          4 |    3 |    3 |
|   12 |     403 | 2020-05-23 | 200000 |          4 |          6 |    4 |    4 |
|   13 |     501 | 2020-05-27 |  70000 |          1 |          2 |    1 |    1 |
|   14 |     502 | 2020-05-31 | 100000 |          1 |          2 |    1 |    2 |
|   15 |     503 | 2020-05-11 | 150000 |          2 |          4 |    3 |    3 |
+------+---------+------------+--------+------------+------------+------+------+
13 rows in set (0.00 sec)

mysql> DESC RESERVATION;
+--------------+------+------+-----+---------+-------+
| Field        | Type | Null | Key | Default | Extra |
+--------------+------+------+-----+---------+-------+
| ROOM_NO      | int  | YES  |     | NULL    |       |
| BOOKING_DATE | date | YES  |     | NULL    |       |
+--------------+------+------+-----+---------+-------+
```



```
mysql> DELIMITER //
mysql> CREATE PROCEDURE BOOK()
    -> BEGIN
    -> SELECT * FROM ROOMSERVICE;
    -> SELECT * FROM RESERVATION;
    -> END//
Query OK, 0 rows affected (0.12 sec)

mysql> DELIMITER ;

mysql> DELETE FROM ROOMSERVICE WHERE ROOM_NO = 501;
Query OK, 1 row affected (0.09 sec)

mysql> CALL BOOK;;
+------+---------+------------+--------+------------+------------+------+------+
| NO   | ROOM_NO | BOOKDATE   | PRICE  | PEOPLE_MIN | PEOPLE_MAX | BED  | ROOM |
+------+---------+------------+--------+------------+------------+------+------+
|    2 |     102 | 2020-05-20 | 150000 |          2 |          4 |    3 |    3 |
|    3 |     103 | 2020-05-25 | 200000 |          4 |          6 |    4 |    4 |
|    4 |     201 | 2020-05-15 |  70000 |          1 |          2 |    1 |    1 |
|    5 |     202 | 2020-05-18 | 100000 |          1 |          2 |    1 |    2 |
|    6 |     203 | 2020-05-14 | 150000 |          2 |          4 |    3 |    3 |
|    7 |     301 | 2020-05-28 | 200000 |          4 |          6 |    4 |    4 |
|    8 |     302 | 2020-05-30 |  70000 |          1 |          2 |    1 |    1 |
|   10 |     401 | 2020-05-12 | 150000 |          2 |          4 |    3 |    3 |
|   11 |     402 | 2020-05-19 | 150000 |          2 |          4 |    3 |    3 |
|   12 |     403 | 2020-05-23 | 200000 |          4 |          6 |    4 |    4 |
|   14 |     502 | 2020-05-31 | 100000 |          1 |          2 |    1 |    2 |
|   15 |     503 | 2020-05-11 | 150000 |          2 |          4 |    3 |    3 |
+------+---------+------------+--------+------------+------------+------+------+
12 rows in set (0.00 sec)

+---------+--------------+
| ROOM_NO | BOOKING_DATE |
+---------+--------------+
|     101 | 2020-05-12   |
|     303 | 2020-05-27   |
|     501 | 2020-05-27   |
+---------+--------------+
```

20200511 DB과제

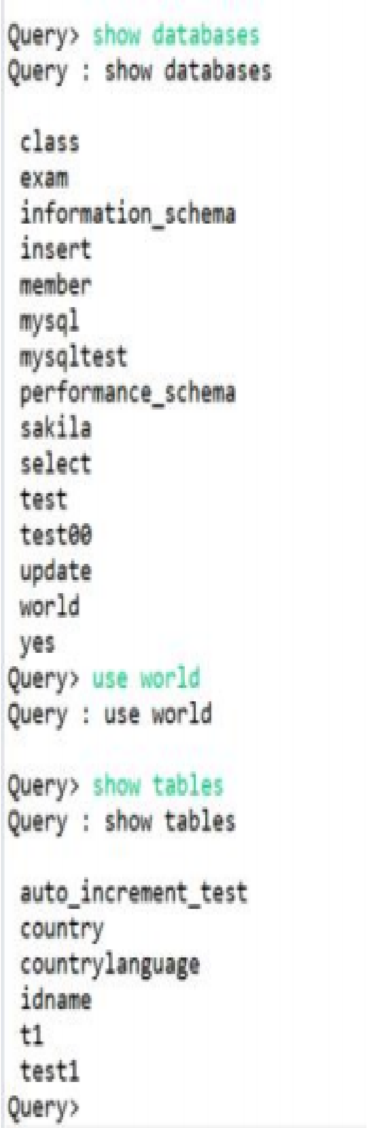
JBDC

B반 신석환

[복습]

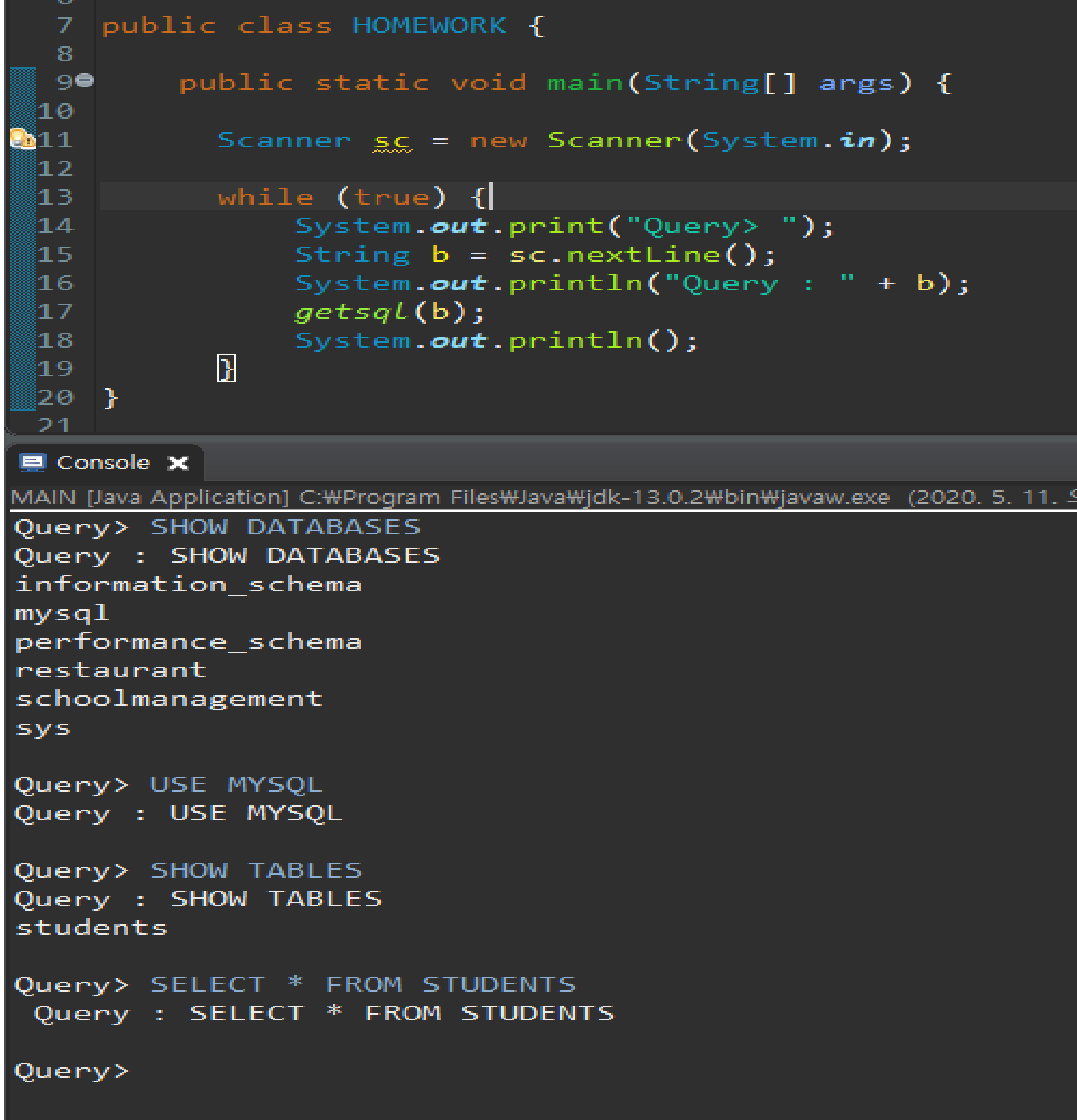
1. 아래 시나리오를 만족시키는 프로그램을 작성하시오

1) 메뉴 구성



```
Query> show databases
Query : show databases

 class
 exam
 information_schema
 insert
 member
 mysql
 mysqltest
 performance_schema
 sakila
 select
 test
 test00
 update
 world
 yes
Query> use world
Query : use world

Query> show tables
Query : show tables

 auto_increment_test
 country
 countrylanguage
 idname
 t1
 test1
Query>
```

(1) 위와 같이 Query를 입력하면 결과를 가져오도록 함(무한반복)



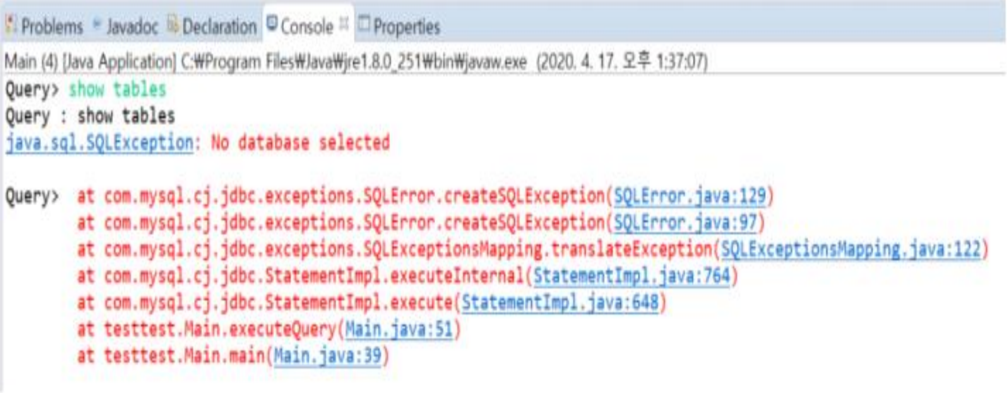
```
public class HOMEWORK {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        while (true) {
            System.out.print("Query> ");
            String b = sc.nextLine();
            System.out.println("Query : " + b);
            getsql(b);
            System.out.println();
        }
}
```

```
MAIN [Java Application] C:₩Program Files₩Java₩jdk-13.0.2₩bin₩javaw.exe  (2020. 5. 11.
Query> SHOW DATABASES
Query : SHOW DATABASES
information_schema
mysql
performance_schema
restaurant
schoolmanagement
sys

Query> USE MYSQL
Query : USE MYSQL

Query> SHOW TABLES
Query : SHOW TABLES
students

Query> SELECT * FROM STUDENTS
 Query : SELECT * FROM STUDENTS

Query>
```

[복습]

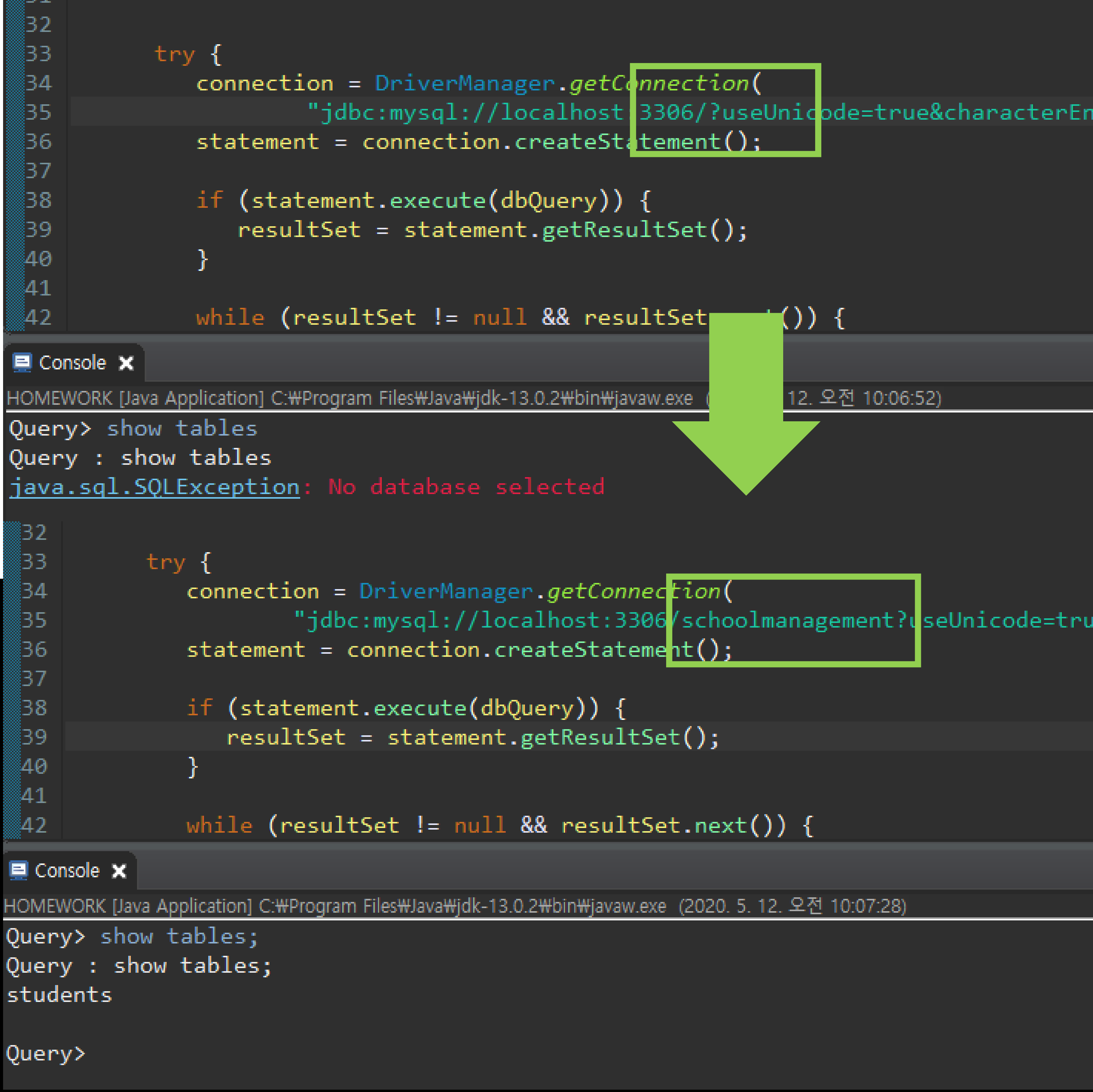
2. 앞 과제에서 작성한 프로그램의 오류를 해결하시오

1) 아래와 같이 Exception이 발생함



```
Main (4) [Java Application] C:₩Program Files₩Java₩jre1.8.0_251₩bin₩javaw.exe  (2020. 4. 17. 오후 1:37:07)
Query> show tables
Query : show tables
java.sql.SQLException: No database selected

Query>  at com.mysql.cj.jdbc.exceptions.SQLError.createSQLException(SQLError.java:129)
        at com.mysql.cj.jdbc.exceptions.SQLError.createSQLException(SQLError.java:97)
        at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
        at com.mysql.cj.jdbc.StatementImpl.executeInternal(StatementImpl.java:764)
        at com.mysql.cj.jdbc.StatementImpl.execute(StatementImpl.java:648)
        at testtest.Main.executeQuery(Main.java:51)
        at testtest.Main.main(Main.java:39)
```

Hint) Try ~ Catch 구문 활용



```java
        try {
            connection = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/?useUnicode=true&characterEn
            statement = connection.createStatement();

            if (statement.execute(dbQuery)) {
                resultSet = statement.getResultSet();
            }

            while (resultSet != null && resultSet.next()) {
```

```
HOMEWORK [Java Application] C:₩Program Files₩Java₩jdk-13.0.2₩bin₩javaw.exe  (     12. 오전 10:06:52)
Query> show tables
Query : show tables
java.sql.SQLException: No database selected
```

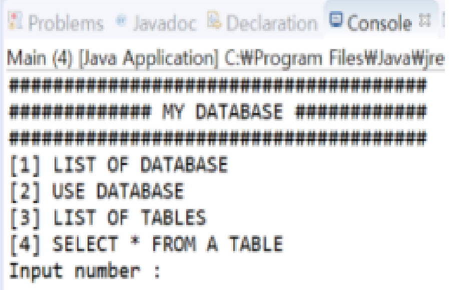
```java
        try {
            connection = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/schoolmanagement?useUnicode=tru
            statement = connection.createStatement();

            if (statement.execute(dbQuery)) {
                resultSet = statement.getResultSet();
            }

            while (resultSet != null && resultSet.next()) {
```

```
HOMEWORK [Java Application] C:₩Program Files₩Java₩jdk-13.0.2₩bin₩javaw.exe  (2020. 5. 12. 오전 10:07:28)
Query> show tables;
Query : show tables;
students

Query>
```

[복습]

3. 아래 시나리오를 만족시키는 프로그램을 작성하시오

1) 메뉴 구성

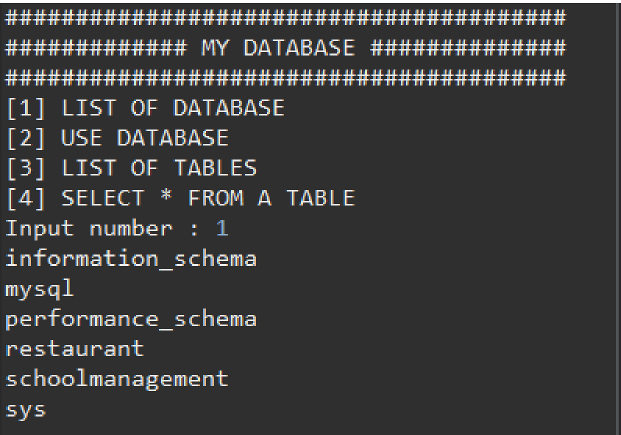


```
Main (4) [Java Application] C:₩Program Files₩Java₩jre
########################################
############# MY DATABASE ############
########################################
[1] LIST OF DATABASE
[2] USE DATABASE
[3] LIST OF TABLES
[4] SELECT * FROM A TABLE
Input number :
```

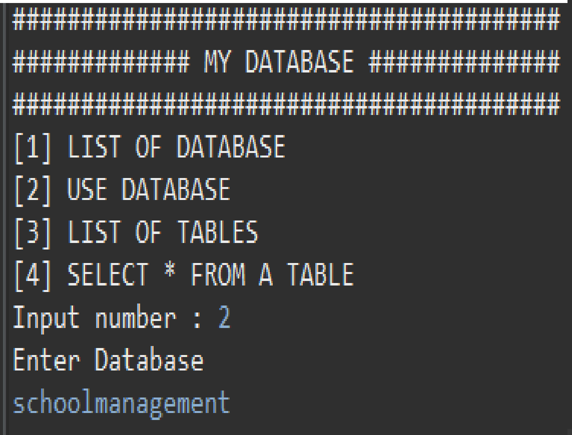
(1) 위와 같이 메뉴 4개를 구성

(2) 사용자의 Input numbe를 받아 각 메뉴에서 실행해야 하는 쿼리문 실행

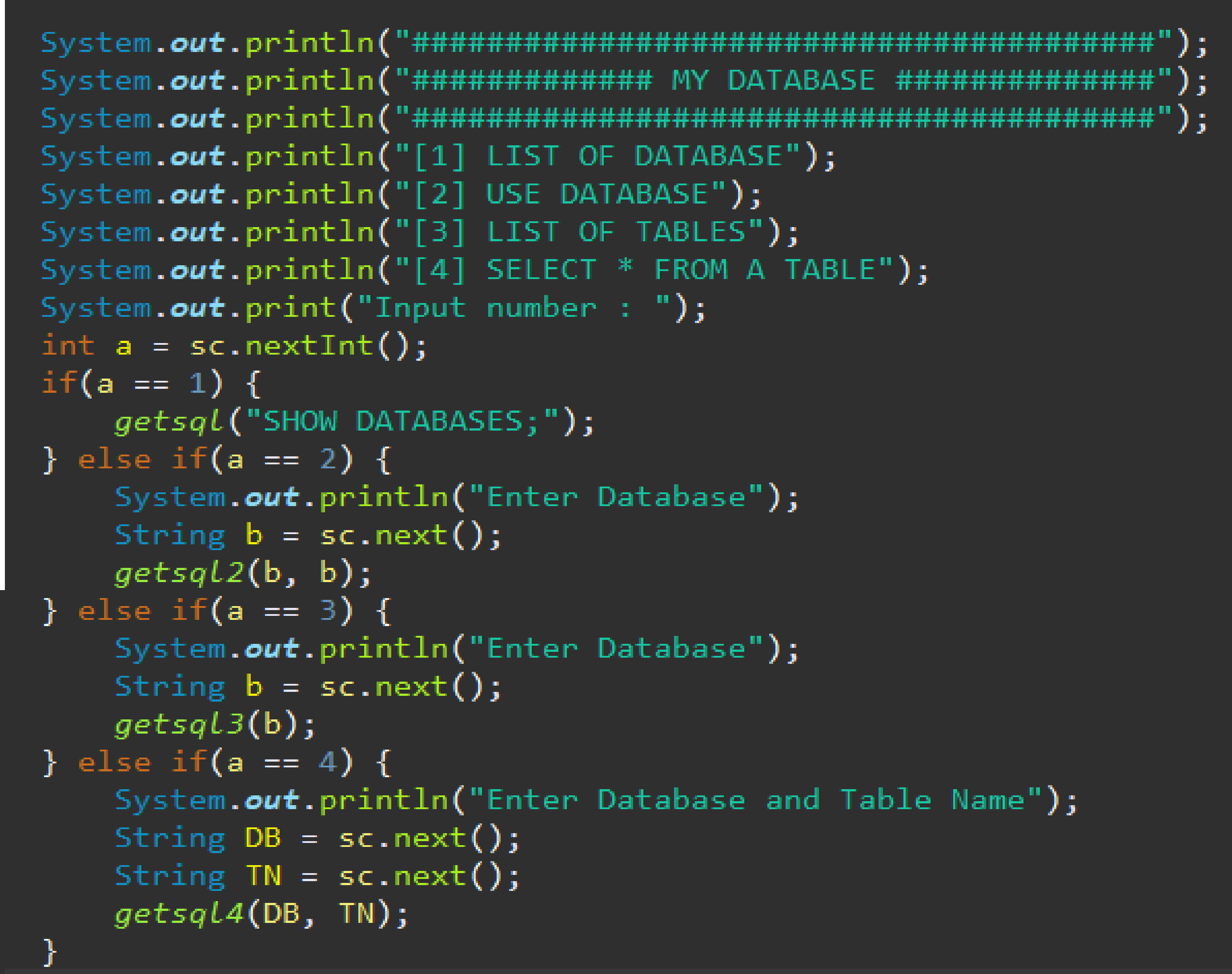
※ 필요에 따라 DATABASE나 TABLE 이름도 추가적으로 입력받아야 함



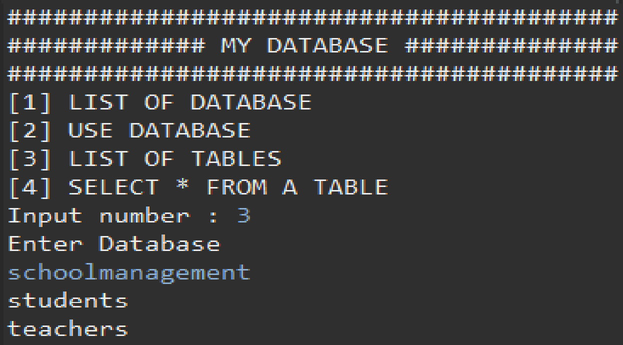
```
########################################
############# MY DATABASE ##############
########################################
[1] LIST OF DATABASE
[2] USE DATABASE
[3] LIST OF TABLES
[4] SELECT * FROM A TABLE
Input number : 1
information_schema
mysql
performance_schema
restaurant
schoolmanagement
sys
```



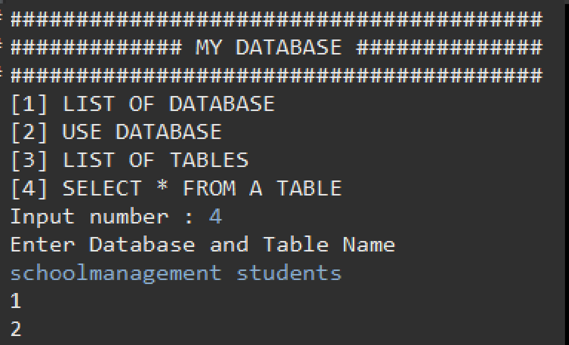
```
########################################
############# MY DATABASE ##############
########################################
[1] LIST OF DATABASE
[2] USE DATABASE
[3] LIST OF TABLES
[4] SELECT * FROM A TABLE
Input number : 2
Enter Database
schoolmanagement
```



```java
System.out.println("########################################");
System.out.println("############# MY DATABASE ##############");
System.out.println("########################################");
System.out.println("[1] LIST OF DATABASE");
System.out.println("[2] USE DATABASE");
System.out.println("[3] LIST OF TABLES");
System.out.println("[4] SELECT * FROM A TABLE");
System.out.print("Input number : ");
int a = sc.nextInt();
if(a == 1) {
    getsql("SHOW DATABASES;");
} else if(a == 2) {
    System.out.println("Enter Database");
    String b = sc.next();
    getsql2(b, b);
} else if(a == 3) {
    System.out.println("Enter Database");
    String b = sc.next();
    getsql3(b);
} else if(a == 4) {
    System.out.println("Enter Database and Table Name");
    String DB = sc.next();
    String TN = sc.next();
    getsql4(DB, TN);
}
```



```
########################################
############# MY DATABASE ##############
########################################
[1] LIST OF DATABASE
[2] USE DATABASE
[3] LIST OF TABLES
[4] SELECT * FROM A TABLE
Input number : 3
Enter Database
schoolmanagement
students
teachers
```



```
########################################
############# MY DATABASE ##############
########################################
[1] LIST OF DATABASE
[2] USE DATABASE
[3] LIST OF TABLES
[4] SELECT * FROM A TABLE
Input number : 4
Enter Database and Table Name
schoolmanagement students
1
2
```

4번을 선택 했을 때 테이블 전체가 출력이 되지 않습니다.. 죄송합니다 해결 방법을 모르겠습니다.