

11주차 2차시 데이터 삽입

【학습목표】

1. 데이터를 테이블에 직접 삽입하는 방법을 설명할 수 있다.
2. 부속 질의문을 이용하여 만들어진 결과를 테이블에 삽입하는 방법을 설명할 수 있다.

학습내용1 : 데이터 삽입 개요

- SQL문 : INSERT
- 2가지 삽입 방법
 - . 테이블에 튜플을 직접 삽입
 - . 부속 질의문을 이용해 튜플을 삽입

1. 주의 사항

- . 데이터를 삽입하기 위하여, 테이블 속성의 정의를 확인하여야 함.
- . NULL이 가능한 속성은 값을 입력하지 않아도 됨.
- . NOT NULL인 속성은 반드시 값을 입력하여야 함.
- . DBMS는 데이터 유일성을 유지하기 위하여 같은 튜플의 삽입을 제한하므로 데이터 작성시 같은 튜플을 만들지 않도록 하여야 함.
- . DBMS는 데이터 무결성을 유지하기 위하여 각 속성 도메인에 위배되는 데이터의 삽입을 거절하므로, 도메인에 맞는 데이터 작성이 필요함.

학습내용2 : 데이터 직접 삽입

- 데이터 직접 삽입
 - . 기본 형식

```
INSERT INTO   테이블_이름 [ ( 속성_리스트 ) ]  
VALUES       ( 속성값_리스트 ) ;
```

- * 속성_리스트와 속성값_리스트의 개수와 순서가 동일하여야 함
- * 속성_리스트와 속성값_리스트의 각 요소는 일대일 대응
- * 속성값_리스트는 생략 가능
 - . 테이블을 정의할 때 지정한 속성의 순서대로 속성값_리스트를 작성
- * 속성값_리스트의 속성값은 문자나 날짜 타입의 데이터는 작은 따옴표로 묶어야 함

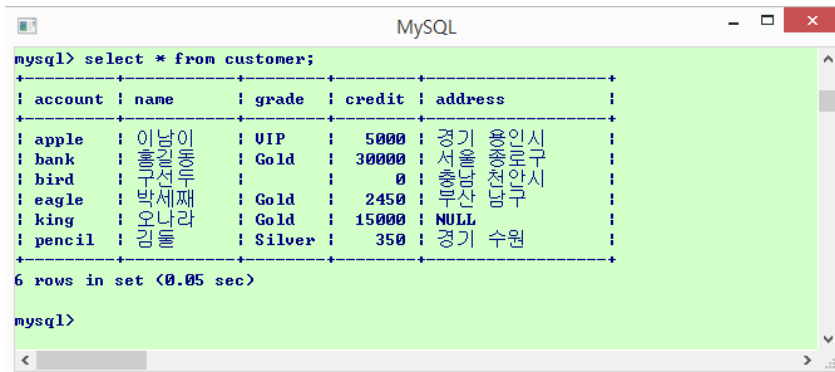
1. 질의 1 : 기본질의

쇼핑몰 데이터베이스의 Customer 테이블에 고객아이디(account)가 moon, 이름(name)이 한가위, 등급(grade)이 New, 적립금(credit)이 1000, 주소(address)가 서울 용산구인 정보를 삽입하시오.

- 질의문

```
INSERT INTO customer ( account, name, grade, credit, address )
VALUES ( 'moon', '한가위', 'New', 1000, '서울 용산구' );
```

- 데이터 삽입전 Customer 테이블 데이터

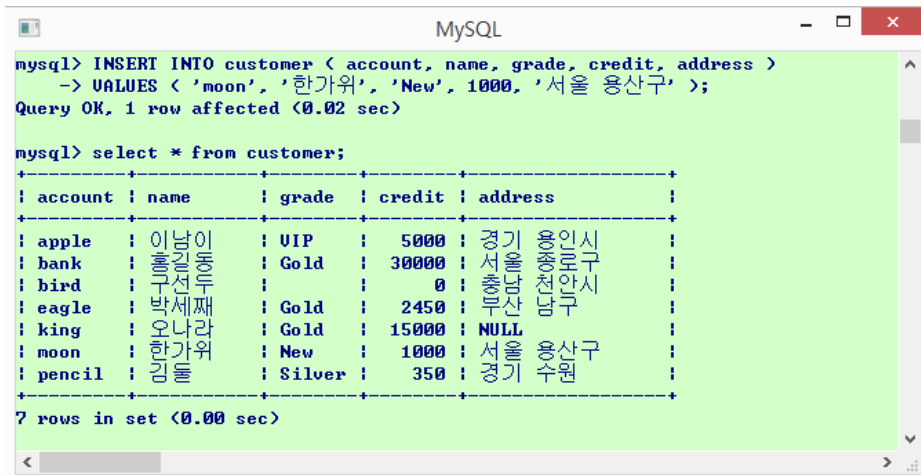


```
mysql> select * from customer;
```

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 종로구
bird	구선두		0	충남 천안시
eagle	박세재	Gold	2450	부산 남구
king	오나라	Gold	15000	NULL
pencil	김돌	Silver	350	경기 수원

6 rows in set (0.05 sec)

- 데이터 삽입 및 데이터 조회



```
mysql> INSERT INTO customer ( account, name, grade, credit, address )
-> VALUES ( 'moon', '한가위', 'New', 1000, '서울 용산구' );
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from customer;
```

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 종로구
bird	구선두		0	충남 천안시
eagle	박세재	Gold	2450	부산 남구
king	오나라	Gold	15000	NULL
moon	한가위	New	1000	서울 용산구
pencil	김돌	Silver	350	경기 수원

7 rows in set (0.00 sec)

2. 질의 2 : 속성리스트

쇼핑몰 데이터베이스의 Customer 테이블에 고객아이디(account)가 sun, 이름(name)이 박태양, 적립금(credit)이 1000, 등급(grade)이 New, 주소(address)가 용인 처인구인 정보를 삽입하시오. 단, 속성리스트를 별도로 지정하지 말고 입력하시오.

- Customer 테이블의 구조로부터 속성리스트의 속성 순서를 확인

```
MySQL
mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| account | varchar(10)   | NO   | PRI | NULL     |       |
| name    | varchar(10)   | NO   |     | NULL     |       |
| grade   | varchar(6)    | NO   |     | New      |       |
| credit  | int(11)       | YES  |     | 0        |       |
| address | varchar(100)  | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.03 sec)

mysql>
```

- 질의문

```
INSERT INTO customer
VALUES ( 'sun', '박태양', 'New', 1000, '용인 처인구' );
```

- 데이터 삽입전 Customer 테이블 데이터

```
MySQL
mysql> select * from customer;
+-----+-----+-----+-----+-----+
| account | name      | grade | credit | address |
+-----+-----+-----+-----+-----+
| apple   | 이남이   | VIP   | 5000   | 경기 용인시 |
| bank    | 홍길동   | Gold  | 30000  | 서울 종로구 |
| bird    | 구선두   |       | 0      | 충남 천안시 |
| eagle   | 박세재   | Gold  | 2450   | 부산 남구   |
| king    | 오나라   | Gold  | 15000  | NULL       |
| moon    | 한가위   | New   | 1000   | 서울 용산구 |
| pencil  | 김돌     | Silver | 350    | 경기 수원   |
+-----+-----+-----+-----+-----+
7 rows in set (0.02 sec)

mysql>
```

- 데이터 삽입 및 조회

```

MySQL
mysql> INSERT INTO customer
-> VALUES ( 'sun', '박태양', 'New', 1000, '용인 처인구' );
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer;
+-----+-----+-----+-----+-----+
| account | name      | grade | credit | address |
+-----+-----+-----+-----+-----+
| apple   | 이남이    | VIP   | 5000   | 경기 용인시 |
| bank    | 홍길동    | Gold  | 30000  | 서울 종로구 |
| bird    | 구선두    |       | 0       | 충남 천안시 |
| eagle   | 박세재    | Gold  | 2450   | 부산 남구   |
| king    | 오나라    | Gold  | 15000  | NULL       |
| moon    | 한가위    | New   | 1000   | 서울 용산구 |
| pencil  | 김민철    | Silver| 350    | 경기 수원시 |
| sun     | 박태양    | New   | 1000   | 용인 처인구 |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>

```

[TIP]

- 테이블에 레코드를 추가할 때 모든 속성값을 입력할 경우, 속성리스트는 기입하지 않고 속성값을 테이블 구조의 필드 순서와 동일하게 속성값리스트에 기입한다.

3. 질의 3 : Default 값

신규 고객의 계정(account)는 table, 이름은 정탁자, 등급(grade)는 New, 적립금(credit)은 0, 주소는 NULL 에 대한 정보를 입력하시오. 단, 테이블의 구조를 보고, Default 값을 사용할 속성은 입력하지 마시오.

- Customer 테이블 생성시 Default의 값
 - . 등급(grade) : New
 - . 적립금(Credit) : 0
 - . 주소(address) : NULL

```

MySQL
mysql> desc customer;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account | varchar(10) | NO   | PRI | NULL    |       |
| name    | varchar(10) | NO   |     | NULL    |       |
| grade   | varchar(6)  | NO   |     | New     |       |
| credit  | int(11)     | YES  |     | 0       |       |
| address | varchar(100) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

- 질의문

```
INSERT INTO customer ( account, name )
VALUES ( 'table', '정탁자' );
```

- 데이터 삽입전 Customer 테이블 데이터

MySQL

```
mysql> select * from customer;
```

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 중로구
bird	구선두		0	충남 천안시
eagle	박세재	Gold	2450	부산 남구
king	오나라	Gold	15000	NULL
moon	한가위	New	1000	서울 용산구
pencil	김돌	Silver	350	경기 수원시
sun	박태양	New	1000	경기 처인구

8 rows in set (0.00 sec)

- 데이터 삽입 및 조회

MySQL

```
mysql> INSERT INTO customer ( account, name )
-> VALUES ( 'table', '정탁자' );
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from customer;
```

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 중로구
bird	구선두		0	충남 천안시
eagle	박세재	Gold	2450	부산 남구
king	오나라	Gold	15000	NULL
moon	한가위	New	1000	서울 용산구
pencil	김돌	Silver	350	경기 수원시
sun	박태양	New	1000	경기 처인구
table	정탁자	New	0	NULL

9 rows in set (0.00 sec)

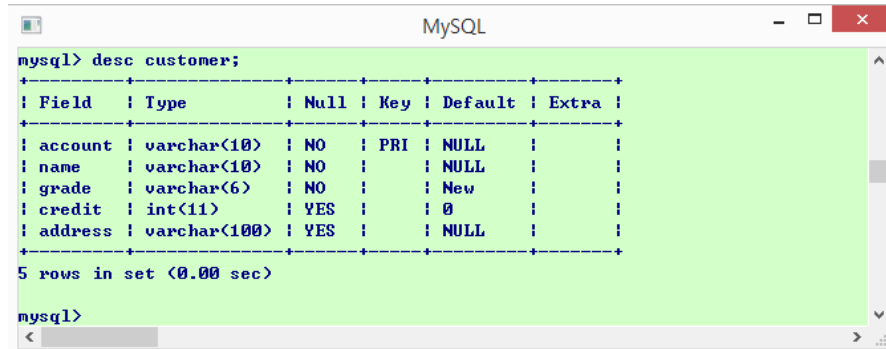
[TIP]

- 테이블에 레코드를 삽입시 속성에 Default 값을 입력하고자 할 경우, 해당 속성을 속성리스트에서 제외하고, 속성값도 입력하지 않는다.
- 질의문에 Default 값이라도 입력하고자 할 경우, 속성값을 입력할 속성들을 속성리스트에 적어야 한다.

4. 질의 4 : NOT NULL

신규 고객의 계정(account)는 watch, 이름은 김정확, 적립금(credit)은 NULL, 주소는 NULL 에 대한 정보를 입력하시오.

- Customer 테이블 구조



```
mysql> desc customer;
```

Field	Type	Null	Key	Default	Extra
account	varchar(10)	NO	PRI	NULL	
name	varchar(10)	NO		NULL	
grade	varchar(6)	NO		New	
credit	int(11)	YES		0	
address	varchar(100)	YES		NULL	

```
5 rows in set (0.00 sec)

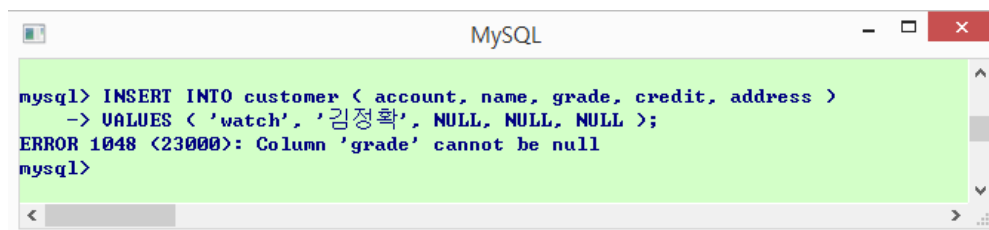
mysql>
```

- . grade 속성 : NOT NULL
- . credit 속성 : NULL 허용
- . address 속성 : NULL 허용

- 질의문

```
INSERT INTO customer ( account, name, grade, credit, address )
VALUES ( 'watch', '김정확', NULL, NULL, NULL );
```

- 질의 결과

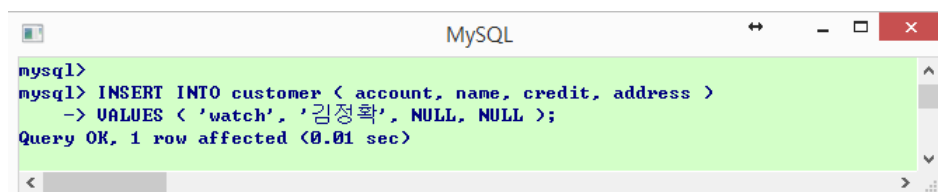


```
mysql> INSERT INTO customer ( account, name, grade, credit, address )
-> VALUES ( 'watch', '김정확', NULL, NULL, NULL );
ERROR 1048 (23000): Column 'grade' cannot be null
mysql>
```

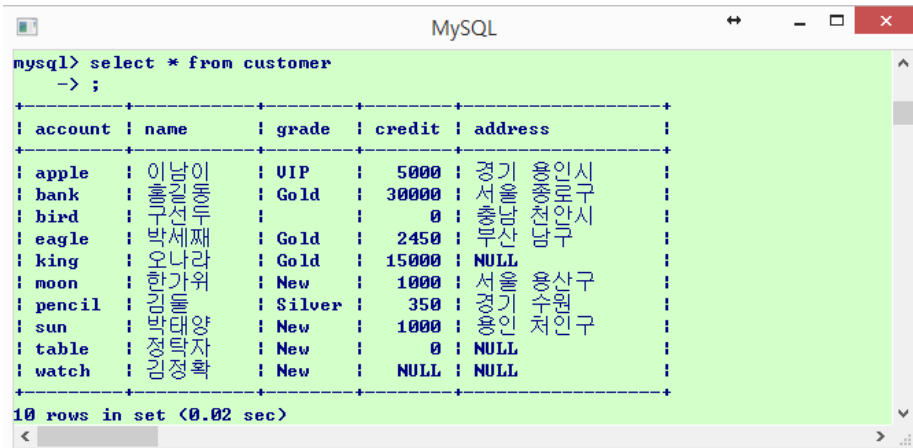
- 질의문

```
INSERT INTO customer ( account, name, credit, address )
VALUES ( 'watch', '김정확', NULL, NULL );
```

- 질의 결과



```
mysql>
mysql> INSERT INTO customer ( account, name, credit, address )
-> VALUES ( 'watch', '김정확', NULL, NULL );
Query OK, 1 row affected (0.01 sec)
```



```
mysql> select * from customer
-> ;
```

account	name	grade	credit	address
apple	이남이	VIP	5000	경기 용인시
bank	홍길동	Gold	30000	서울 중로구
bird	구선두		0	충남 천안시
eagle	박세재	Gold	2450	부산 남구
king	오나라	Gold	15000	NULL
moon	한가위	New	1000	서울 용산구
pencil	김돌	Silver	350	경기 수원시
sun	박태양	New	1000	용인 처인구
table	정탁자	New	0	NULL
watch	김정확	New	NULL	NULL

10 rows in set (0.02 sec)

학습내용3 : 부속 질의문을 이용한 데이터 삽입

- 부속 질의문으로 SELECT문을 이용
- 부속 질의문을 이용해 다른 테이블에서 검색한 데이터를 튜플로 삽입
- 기본 형식

```
INSERT INTO 테이블_이름 [ ( 속성_리스트 ) ]
SELECT 문;
```

1. 질의 1

매출 테이블(Sales)을 만들고, Orders 테이블과 Items 테이블로부터 필요한 데이터를 조회하여 날짜별 매출을 만드는 질의문을 작성하시오. 예제로 2015-06-01의 총 매출을 확인하여 sales 테이블에 저장하시오.

- 날짜별 매출 테이블을 만드시오.
 . 테이블 이름 : sales
 . 속성 : 번호(num) - int, 날짜(date)-date, 매출금액(amount)-int
- Orders 테이블과 Items 테이블을 이용하여 2015-06-01 날짜에 발생한 주문별 매출 질의문을 만들고, 결과를 확인하시오.
- B의 질의문을 이용해서 2015-06-01에 발생한 총 매출을 구하는 질의문을 만들고, 결과를 확인하시오.
- C의 질의문을 부속질의문으로 사용하여 A에서 만든 테이블에 2015-06-01에 발생한 총 매출 튜플을 삽입하시오.

- 질의1-A (매출 테이블 생성) 에 대한 질의문

```
create table sales (
    num int not null auto_increment,
    sdate date not null,
    amount int not null default 0,
    primary key ( num )
```

);

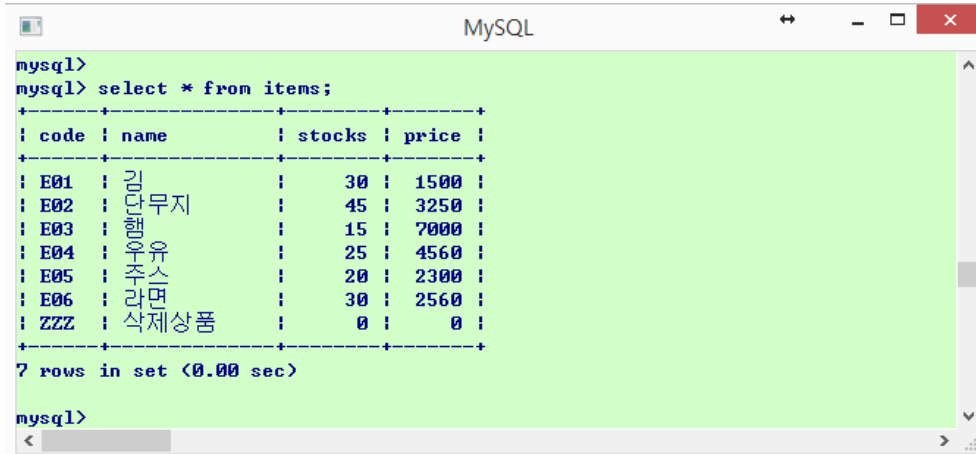
```
mysql> create table sales (
-> num int not null auto_increment,
-> sdate date not null,
-> amount int not null default 0,
-> primary key ( num )
-> );
Query OK, 0 rows affected (0.18 sec)
```

```
mysql> desc sales;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| num   | int(11) | NO   | PRI | NULL    | auto_increment |
| sdate | date   | NO   |     | NULL    |               |
| amount | int(11) | NO   |     | 0       |               |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

- 질의1-B : “2015-06-01 날짜에 발생한 주문별 매출” 질의문
 . 주문별 매출을 계산하기 위하여 필요한 테이블 : Orders, Items
 . Orders 테이블

```
mysql> select * from Orders;
+-----+-----+-----+-----+-----+-----+
| num | customer | item | qty | date       | saddr |
+-----+-----+-----+-----+-----+-----+
| 1   | apple    | E06  | 15  | 2015-06-01 | 경기 용인시 |
| 2   | apple    | E05  | 2   | 2015-06-01 | 경기 용인시 |
| 3   | eagle    | E01  | 3   | 2015-06-03 | 부산 강서구 |
| 4   | bank     | E04  | 5   | 2015-06-03 | 서울 용문로 |
| 5   | bird     | E02  | 10  | 2015-06-04 | 충남 천안시 |
| 6   | bird     | E03  | 1   | 2015-06-04 | 충남 천안시 |
| 7   | apple    | E06  | 10  | 2015-06-10 | NULL    |
| 8   | apple    | E05  | 10  | 2015-06-10 | NULL    |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```


. Items 테이블



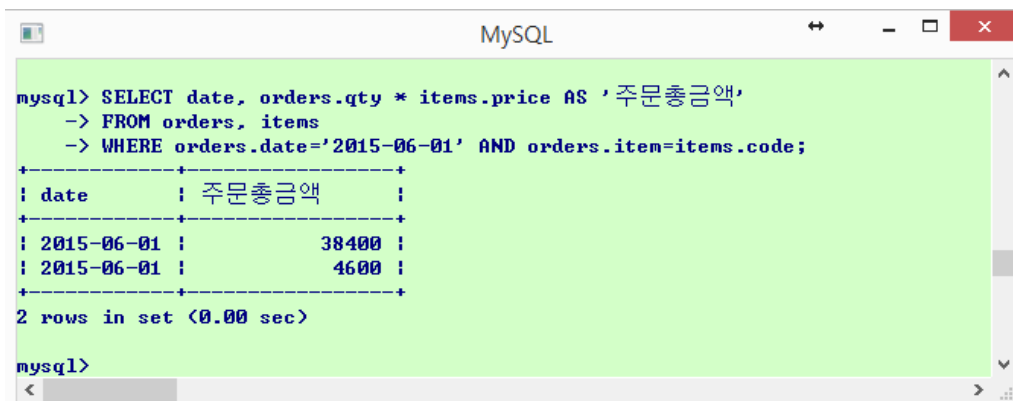
```
mysql> select * from items;
```

code	name	stocks	price
E01	김	30	1500
E02	단무지	45	3250
E03	햄	15	7000
E04	우유	25	4560
E05	주스	20	2300
E06	라면	30	2560
ZZZ	삭제상품	0	0

```
7 rows in set (0.00 sec)
```

. 질의문

```
SELECT date, orders.qty * items.price AS '주문총금액'
FROM orders, items
WHERE orders.date='2015-06-01' AND orders.item=items.code;
```



```
mysql> SELECT date, orders.qty * items.price AS '주문총금액'
-> FROM orders, items
-> WHERE orders.date='2015-06-01' AND orders.item=items.code;
```

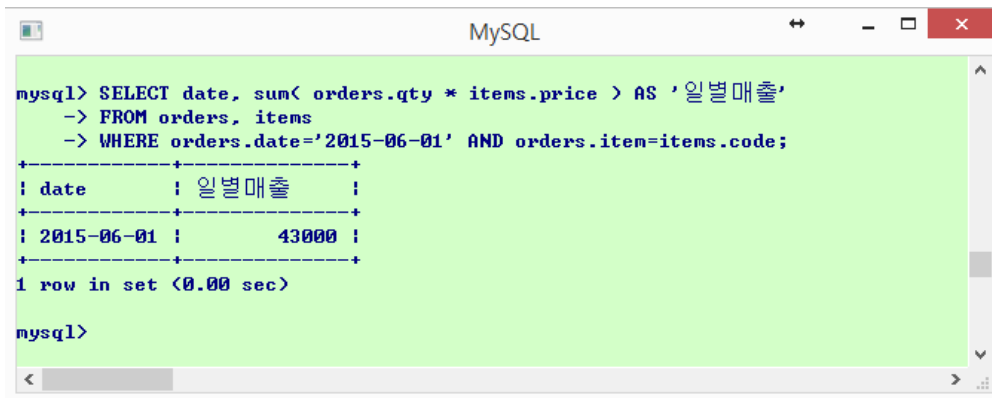
date	주문총금액
2015-06-01	38400
2015-06-01	4600

```
2 rows in set (0.00 sec)
```

- 질의1-C : “2015-06-01 날짜에 발생한 총매출” 질의문
- . 질의1-B의 결과인 각 주문별 매출의 합계를 계산하기 위한 집계 함수 : SUM

. 질의문

```
SELECT date, sum( orders.qty * items.price ) AS '일별매출'
FROM orders, items
WHERE orders.date='2015-06-01' AND orders.item=items.code;
```

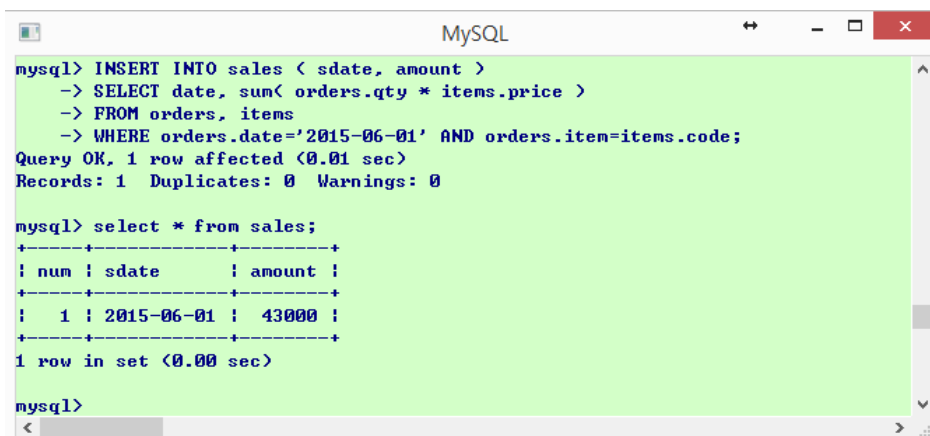


```
mysql> SELECT date, sum( orders.qty * items.price ) AS '일별매출',
-> FROM orders, items
-> WHERE orders.date='2015-06-01' AND orders.item=items.code;
+-----+-----+
| date      | 일별매출 |
+-----+-----+
| 2015-06-01 | 43000    |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

- 질의1-D : 일별 총매출을 Sales 테이블에 삽입
 - . 질의1-C의 결과를 INSERT 질의문을 사용하여 Sales 테이블에 삽입
- . 질의문

```
INSERT INTO sales ( sdate, amount )
SELECT date, sum( orders.qty * items.price )
FROM orders, items
WHERE orders.date='2015-06-01' AND orders.item=items.code;
```



```
mysql> INSERT INTO sales ( sdate, amount )
-> SELECT date, sum( orders.qty * items.price )
-> FROM orders, items
-> WHERE orders.date='2015-06-01' AND orders.item=items.code;
Query OK, 1 row affected (0.01 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> select * from sales;
+----+-----+-----+
| num | sdate      | amount |
+----+-----+-----+
| 1   | 2015-06-01 | 43000  |
+----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

【학습정리】

1. 데이터를 테이블에 삽입하기 위하여 INSERT 키워드를 사용한다.
2. 데이터를 테이블에 삽입하는 방법으로 투플을 직접 삽입하는 방법과 부속 질의문을 이용해 투플을 삽입하는 방법이 있다.