

7주차 1차시 데이터 정의(2/3)

【학습목표】

- 1. SQL을 사용한 데이터 정의 방법 중 키 정의를 설명할 수 있다.
- 2. SQL을 사용하여 제약 조건을 테이블 생성시 적용하는 방법을 설명할 수 있다.

학습내용1 : 테이블 생성 - 키 정의

- 기본키, 대체키, 외래키를 지정

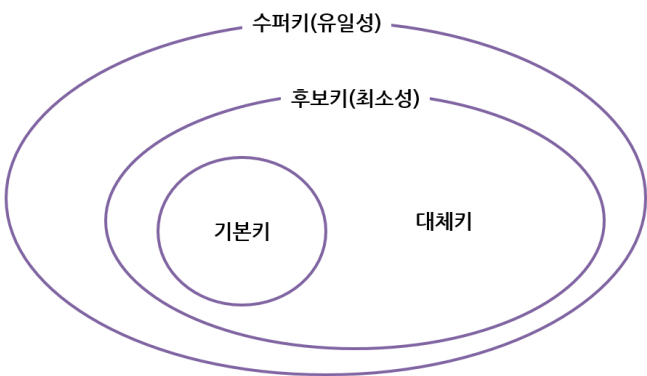


그림. 키의 관계

Key 종류	설명
PRI	기본키, 여러 개의 기본키 중에 하나
UNI	첫번째 대체키 NULL 값을 허용하지만, NULL 허용 여부를 지정할 수 있다.
MUL	같은 값이 다수 발생할 수 있는 속성 중 인덱스로 사용되는 첫 번째 속성

그림. MySQL에서 사용되는 Key 종류

1. 기본키

- PRIMARY KEY를 사용해 지정
- 모든 테이블에 기본키는 반드시 하나만 지정
 - . 예 : PRIMARY KEY (account) <== 고객의 계정(account)를 기본키로 지정
- 여러 개의 속성으로 구성할 수 있음
 - . 예 : CONSTRAINT pk_orderID PRIMARY KEY (customer, item)
 <== 기본키로 pk_orderID가 지정되었고, 기본키의 값은 주문을 한 고객 (customer)과 주문제품 (item)으로 구성됨
- 기본키를 지정하지 않아도 됨
- 기본키는 하나의 튜플을 선택할 수 있음

표. CREATE TABLE 예 : Orders (기본키 : 1개 속성)

```
CREATE TABLE Orders (
  num      int NOT NULL,
  customer varchar(10) NOT NULL,
  item     char(3),
  qty      int NOT NULL DEFAULT 1,
  date     date,
  saddr    varchar(100),
  PRIMARY KEY ( num )
);
```

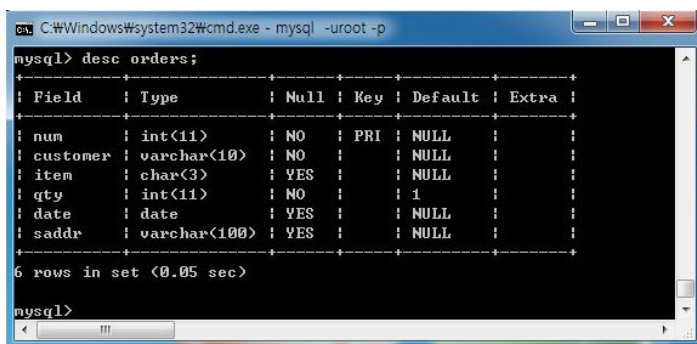


그림. MySQL 적용 후 Orders 릴레이션 구조(기본키 : 1개 속성)

표. CREATE TABLE 예 : Orders (기본키 : 2개의 속성)

```
CREATE TABLE Orders (
  num      int NOT NULL,
  customer varchar(10) NOT NULL,
  item     char(3),
  qty      int NOT NULL DEFAULT 1,
  date     date,
  saddr    varchar(100),
  PRIMARY KEY ( customer, item )
);
```

Field	Type	Null	Key	Default	Extra
num	int(11)	NO		NULL	
customer	varchar(10)	NO	PRI	NULL	
item	char(3)	NO	PRI		
qty	int(11)	NO		1	
date	date	YES		NULL	
saddr	varchar(100)	YES		NULL	

6 rows in set (0.01 sec)

그림. MySQL 적용 후 Orders 릴레이션 구조(기본키 : 2개의 속성)**2. 대체키**

- UNIQUE를 사용해 지정
- 기본키와 같이 각 투플을 유일하게 식별
- 대체키의 값은 테이블 내에서 중복되면 안됨 (유일성)
- NULL 값을 가질 수 있음 (기본키 : NULL 값을 가질 수 없음)
- 한 테이블에서 여러 개 지정할 수 있음
- 예 : UNIQUE (name) <== 고객이름(name) 속성을 대체키로 지정
- 예 : UNIQUE (customer, item)
<== 기본키의 값은 주문을 한 고객(customer)과 주문제품 (item)으로 구성됨

표. CREATE TABLE 예 : Orders (대체키 : 1개의 속성)

```
CREATE TABLE Orders (
  num      int NOT NULL,
  customer varchar(10) NOT NULL,
  item     char(3),
  qty      int NOT NULL DEFAULT 1,
  date     date,
  saddr    varchar(100),
  UNIQUE   ( num )
);
```

```

mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| num   | int(11)       | NO   | PRI | NULL    |       |
| customer | varchar(10)  | NO   |     | NULL    |       |
| item  | char(3)       | YES  |     | NULL    |       |
| qty   | int(11)       | NO   |     | 1        |       |
| date  | date          | YES  |     | NULL    |       |
| saddr | varchar(100)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

```

그림. MySQL 적용 후 Orders 릴레이션 구조(대체키 : 1개의 속성)

표. CREATE TABLE 예 : Orders (기본키, 대체키)

```

CREATE TABLE Orders (
  num      int NOT NULL,
  customer varchar(10) NOT NULL,
  item     char(3),
  qty      int NOT NULL DEFAULT 1,
  date     date,
  saddr    varchar(100),
  PRIMARY KEY ( num ),
  UNIQUE   ( customer )
);

```

```

mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| num   | int(11)       | NO   | PRI | NULL    |       |
| customer | varchar(10)  | NO   | UNI | NULL    |       |
| item  | char(3)       | YES  |     | NULL    |       |
| qty   | int(11)       | NO   |     | 1        |       |
| date  | date          | YES  |     | NULL    |       |
| saddr | varchar(100)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.03 sec)

```

그림. MySQL 적용 후 Orders 릴레이션 구조(기본키, 대체키)

표. CREATE TABLE 예 : Orders (2 대체키)

```

CREATE TABLE Orders (
  num      int NOT NULL,
  customer varchar(10) NOT NULL,
  item     char(3),
  qty      int NOT NULL DEFAULT 1,
  date     date,
  saddr    varchar(100),
  UNIQUE   ( customer ),
  UNIQUE   ( item )
);

```

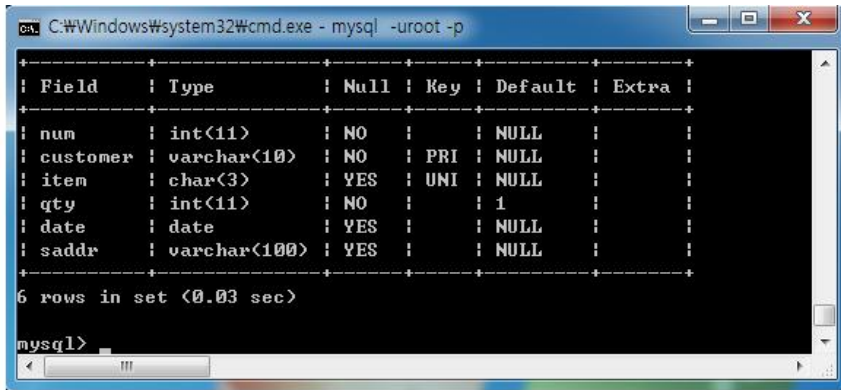


그림. MySQL 적용 후 Orders 릴레이션 구조(2 대체키)

3. 외래키

- FOREIGN KEY를 사용해 지정
- 참조 테이블을 분명히 밝혀야 함.
 - . 참조 무결성 제약 조건을 유지
 - . REFERENCES 다음에 제시
 - . 예 : FOREIGN KEY (customer) REFERENCES Customer(account)
- 참조되는 테이블에서 튜플을 삭제하거나 변경시 제약이 따름
- 참조 무결성 제약 조건 옵션

표. 참조 무결성 제약 조건 옵션

옵션	처리 방법
ON DELETE NO ACTION	튜플을 삭제하지 못하게 함
ON DELETE CASCADE	관련 튜플을 함께 삭제함
ON DELETE SET NULL	관련 튜플의 외래키 값을 NULL로 변경
ON DELETE SET DEFAULT	관련 튜플의 외래키 값을 미리 지정한 기본 값으로 변경

- 참조 무결성 제약 조건 옵션을 선택하지 않은 경우
 - . ON DELETE NO ACTION 이 자동 선택됨
- ON DELETE NO ACTION 실행

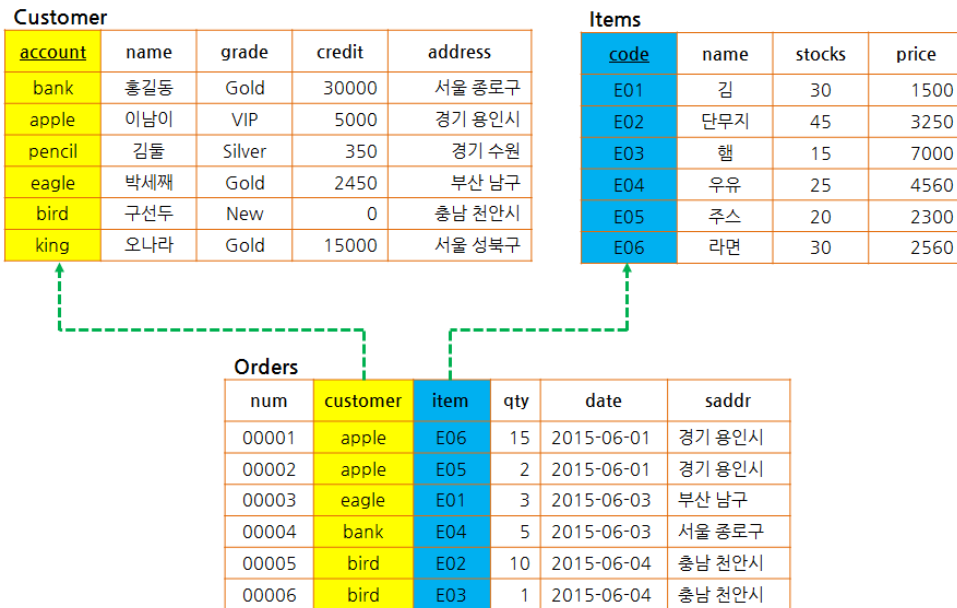


그림. 릴레이션간의 참조키

표. CREATE TABLE 예 : Orders (외래키)

```
CREATE TABLE Orders (
  num      int NOT NULL,
  customer varchar(10) NOT NULL,
  item     char(3) NOT NULL,
  qty      int NOT NULL DEFAULT 1,
  date     date,
  saddr    varchar(100),
  PRIMARY KEY ( num ),
  FOREIGN KEY ( customer ) REFERENCES Customer ( account ),
  FOREIGN KEY ( item ) REFERENCES Items( code )
);
```

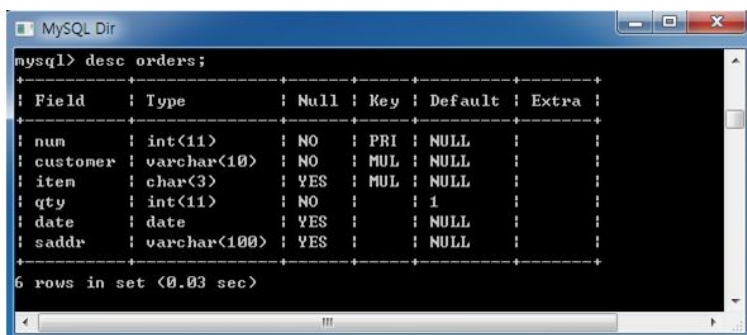


그림. MySQL 적용 후 Orders 릴레이션 구조(외래키)

표. orders 릴레이션 데이터 입력

```

insert into orders values (1, 'apple', 'E06', 15, '2015-06-01', '경기 용인시');
insert into orders values (2, 'apple', 'E05', 2, '2015-06-01', '경기 용인시');
insert into orders values (3, 'eagle', 'E01', 3, '2015-06-03', '부산 남구');
insert into orders values (4, 'bank', 'E04', 5, '2015-06-03', '서울 종로구');
insert into orders values (5, 'bird', 'E02', 10, '2015-06-04', '충남 천안시');
insert into orders values (6, 'bird', 'E03', 1, '2015-06-04', '충남 천안시');

```

```

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 남구   |
| 4   | bank    | E04  | 5   | 2015-06-03 | 서울 종로구 |
| 5   | bird    | E02  | 10  | 2015-06-04 | 충남 천안시 |
| 6   | bird    | E03  | 1   | 2015-06-04 | 충남 천안시 |
+----+-----+-----+----+-----+-----+
6 rows in set (0.00 sec)

```

그림. customer 릴레이션 데이터 확인

code = 'E06' 인 튜플 삭제 요청

삭제 요청

Orders						Items			
num	customer	item	qty	date	saddr	code	name	stocks	price
00001	apple	E06	15	2015-06-01	경기 용인시	E01	김	30	1500
00002	apple	E05	2	2015-06-01	경기 용인시	E02	단무지	45	3250
00003	eagle	E01	3	2015-06-03	부산 남구	E03	햄	15	7000
00004	bank	E04	5	2015-06-03	서울 종로구	E04	우유	25	4560
00005	bird	E02	10	2015-06-04	충남 천안시	E05	주스	20	2300
00006	bird	E03	1	2015-06-04	충남 천안시	E06	라면	30	2560

```

mysql> delete from items where code='E06';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails ('itbank'. 'orders', CONSTRAINT 'orders_ibfk_2' FOREIGN KEY ('item') REFERENCES 'items' ('code'))
mysql>

```

그림. ON DELETE NO ACTION 옵션 실행

```
mysql> show create table Orders;
+-----+
| Table | Create Table |
+-----+
| Orders | CREATE TABLE `orders` (
  `num` int(11) NOT NULL,
  `customer` varchar(10) NOT NULL,
  `item` char(3) DEFAULT NULL,
  `qty` int(11) NOT NULL DEFAULT '1',
  `date` date DEFAULT NULL,
  `saddr` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`num`),
  KEY `customer` (`customer`),
  KEY `item` (`item`),
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`customer`) REFERENCES `customer` (`account`),
  CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`item`) REFERENCES `items` (`code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+
1 row in set (0.01 sec)
```

그림. MySQL에서 테이블 생성시 인식한 외래키 선언부

- ON DELETE CASCADE 실행

표. CREATE TABLE 예 : Orders (외래키)

```
CREATE TABLE Orders (
  num      int NOT NULL,
  customer  varchar(10) NOT NULL,
  item      char(3),
  qty       int NOT NULL DEFAULT 1,
  date      date,
  saddr     varchar(100),
  PRIMARY KEY ( num ),
  FOREIGN KEY ( customer ) REFERENCES Customer ( account ),
  FOREIGN KEY ( item ) REFERENCES Items( code ) ON DELETE CASCADE
);
```

```
mysql> desc orders;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| num | int(11) | NO | PRI | NULL | |
| customer | varchar(10) | NO | MUL | NULL | |
| item | char(3) | YES | MUL | NULL | |
| qty | int(11) | NO | | 1 | |
| date | date | YES | | NULL | |
| saddr | varchar(100) | YES | | NULL | |
+-----+
6 rows in set (0.02 sec)
```

그림. MySQL 적용 후 Orders 릴레이션 구조(외래키)

code = 'E06' 인 튜플 삭제 요청

삭제 요청

Orders

num	customer	item	qty	date	saddr
00001	apple	E06	15	2015-06-01	경기 용인시
00002	apple	E05	2	2015-06-01	경기 용인시
00003	eagle	E01	3	2015-06-03	부산 남구
00004	bank	E04	5	2015-06-03	서울 종로구
00005	bird	E02	10	2015-06-04	충남 천안시
00006	bird	E03	1	2015-06-04	충남 천안시

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

그림. ON DELETE CASCADE 옵션 실행 전

```

C:\Windows\system32\cmd.exe - mysql -uroot -p
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 남구 |
| 4   | bank     | E04  | 5   | 2015-06-03 | 서울 종로구 |
| 5   | bird     | E02  | 10  | 2015-06-04 | 충남 천안시 |
| 6   | bird     | E03  | 1   | 2015-06-04 | 충남 천안시 |
+----+-----+-----+----+-----+-----+
6 rows in set (0.00 sec)

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  |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

그림. ON DELETE CASCADE 옵션 실행 전

num	customer	item	qty	date	saddr
00002	apple	E05	2	2015-06-01	경기 용인시
00003	eagle	E01	3	2015-06-03	부산 남구
00004	bank	E04	5	2015-06-03	서울 종로구
00005	bird	E02	10	2015-06-04	충남 천안시
00006	bird	E03	1	2015-06-04	충남 천안시

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

그림. ON DELETE CASCADE 옵션 실행 후

```

C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> delete from items where code='E06';
Query OK, 1 row affected (0.17 sec)

mysql> select * from items;
+----+-----+-----+-----+
| code | name  | stocks | price |
+----+-----+-----+-----+
| E01  | 김단무지 | 30    | 1500  |
| E02  | 햄     | 45    | 3250  |
| E03  | 유유   | 15    | 7000  |
| E04  | 추스   | 25    | 4560  |
| E05  | 추스   | 20    | 2300  |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

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

```

그림. ON DELETE CASCADE 옵션 실행 후

- ON DELETE SET NULL 실행

표. CREATE TABLE 예 : Orders (외래키)

```

CREATE TABLE Orders (
  num      int NOT NULL,
  customer varchar(10) NOT NULL,
  item     char(3),
  qty      int NOT NULL DEFAULT 1,
  date     date,
  saddr    varchar(100),
  PRIMARY KEY ( num ),
  FOREIGN KEY ( customer ) REFERENCES Customer ( account ),
  FOREIGN KEY ( item ) REFERENCES Items( code ) ON DELETE SET NULL
);

```

```

MySQL Dir
mysql> desc orders;
+----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| num   | int(11)   | NO   | PRI | NULL    |       |
| customer | varchar(10) | NO   | MUL | NULL    |       |
| item  | char(3)   | YES  | MUL | NULL    |       |
| qty   | int(11)   | NO   |     | 1       |       |
| date  | date      | YES  |     | NULL    |       |
| saddr | varchar(100) | YES  |     | NULL    |       |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

```

그림. MySQL 적용 후 Orders 릴레이션 구조(외래키)

code = 'E03' 인 튜플 삭제 요청

삭제 요청

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

Orders

num	customer	item	qty	date	saddr
00002	apple	E05	2	2015-06-01	경기 용인시
00003	eagle	E01	3	2015-06-03	부산 남구
00004	bank	E04	5	2015-06-03	서울 종로구
00005	bird	E02	10	2015-06-04	충남 천안시
00006	bird	E03	1	2015-06-04	충남 천안시

그림. ON DELETE SET NULL 옵션 실행 전

```

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

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  |
+----+-----+-----+-----+
5 rows in set (0.06 sec)

mysql>

```

그림. ON DELETE SET NULL 옵션 실행 전

Items

code	name	stocks	price
E01	김	30	1500
E02	단무지	45	3250
E04	우유	25	4560
E05	주스	20	2300

Orders

num	customer	item	qty	date	saddr
00002	apple	E05	2	2015-06-01	경기 용인시
00003	eagle	E01	3	2015-06-03	부산 남구
00004	bank	E04	5	2015-06-03	서울 종로구
00005	bird	E02	10	2015-06-04	충남 천안시
00006	bird	NULL	1	2015-06-04	충남 천안시

그림. ON DELETE SET NULL 옵션 실행 후

```

mysql> delete from items where code='E03';
Query OK, 1 row affected (0.05 sec)

mysql> select * from items;
+----+-----+-----+-----+
| code | name  | stocks | price |
+----+-----+-----+-----+
| E01  | 김    | 30     | 1500  |
| E02  | 단무지 | 45     | 3250  |
| E04  | 우유  | 25     | 4560  |
| E05  | 주스  | 20     | 2300  |
+----+-----+-----+-----+
4 rows in set (0.00 sec)

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

```

그림. ON DELETE SET NULL 옵션 실행 후

- ON DELETE SET DEFAULT

. MySQL에서는 Parser는 인식하지만, 테이블 정의는 거절됨

- **SET DEFAULT**: This action is recognized by the MySQL parser, but **InnoDB** rejects table definitions containing **ON DELETE SET DEFAULT** or **ON UPDATE SET DEFAULT** clauses.

그림. MySQL의 Foreign Key Constraint의 내용

학습내용2 : 테이블 생성 - 제약 조건 정의

1. 제약 조건 정의

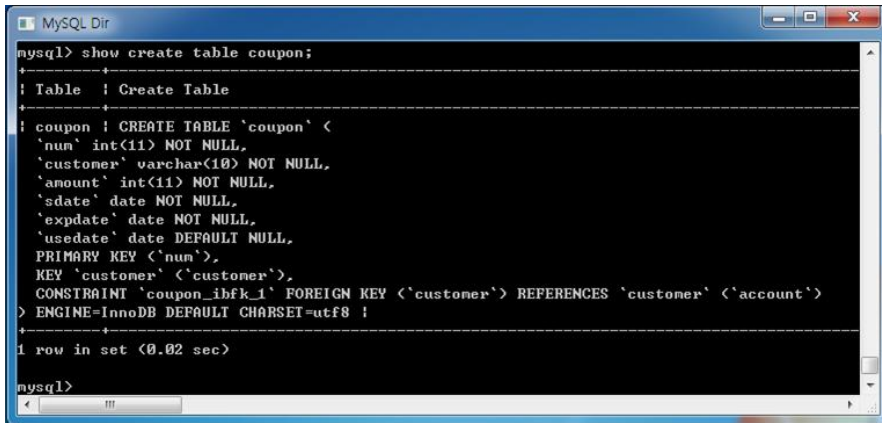
- 테이블을 정의할 때, 특정 속성에 대하여 제약 조건을 지정할 수 있음
- CHECK 키워드를 사용
- 지정한 제약 조건을 만족하는 튜플만 존재할 수 있음.
- 튜플을 삽입하거나 수정할 때, 이 제약조건을 지켜야 함.
- ==> 데이터의 무결성을 위한 제약 조건을 표현하는 방법
- 예 : CHECK (재고량 >= 0 AND 재고량 <= 5000)
 <= 모든 제품의 재고량은 항상 0개 이상이고, 5000개 이하

- CONSTRAINT 키워드와 함께 고유의 이름을 부여
- . 제약 조건을 여러 개 지정할 때 사용
- . 제약 조건을 수정하거나 제거할 때 식별하기 쉽다.

. 예 : CONSTRAINT Chk_date CHECK ((sdate<=usedate) AND (usedate<=expdate))

표. CREATE TABLE 예 : Coupon 릴레이션 (제약조건)

```
CREATE TABLE Coupon (
  num          int NOT NULL,
  customer     varchar(10) NOT NULL,
  amount       int NOT NULL,
  sdate        date NOT NULL,
  expdate      date NOT NULL,
  usedate      date,
  PRIMARY KEY ( num ),
  FOREIGN KEY ( customer ) REFERENCES Customer ( account ),
  CHECK ( amount > 0 )
);
```

```

mysql> show create table coupon;
+-----+-----+
| Table | Create Table |
+-----+-----+
| coupon | CREATE TABLE `coupon` (
  `num` int(11) NOT NULL,
  `customer` varchar(10) NOT NULL,
  `amount` int(11) NOT NULL,
  `sdate` date NOT NULL,
  `expdate` date NOT NULL,
  `usedate` date DEFAULT NULL,
  PRIMARY KEY (`num`),
  KEY `customer` (`customer`),
  CONSTRAINT `coupon_ibfk_1` FOREIGN KEY (`customer`) REFERENCES `customer` (`account`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+-----+
1 row in set (0.02 sec)

mysql>

```

그림. MySQL에서 CHECK에 대한 제한 조건 무시

【학습정리】

1. 테이블을 생성시, 기본키와 대체키를 지정하여 하나의 튜플을 선택하기 위한 속성으로 사용한다.
2. 외래키를 사용할 때는 참조 테이블을 분명히 밝혀야 하며, 참조 무결성 제약 조건 옵션을 사용하여 참조되는 테이블에서 튜플을 삭제하거나 변경시 외래키가 처리되는 방법을 지정하여야 한다.