

7주차 2차시 데이터 정의(3/3)

【학습목표】

1. 정의된 테이블에 새로운 속성을 추가, 삭제, 수정하는 방법을 설명할 수 있다.
2. 테이블의 제약 조건을 추가하거나 삭제하는 방법을 설명할 수 있다.
3. 테이블의 이름을 변경하고 테이블을 제거하는 방법을 설명할 수 있다.

학습내용1 : 테이블 변경

- ALTER TABLE를 사용하여 변경
- 변경 내용
 - . 새로운 속성 추가
 - . 기존 속성 삭제
 - . 기존 속성 수정
 - . 새로운 제약조건 추가
 - . 기존 제약조건 삭제

A. ALTER TABLE 기본 형식

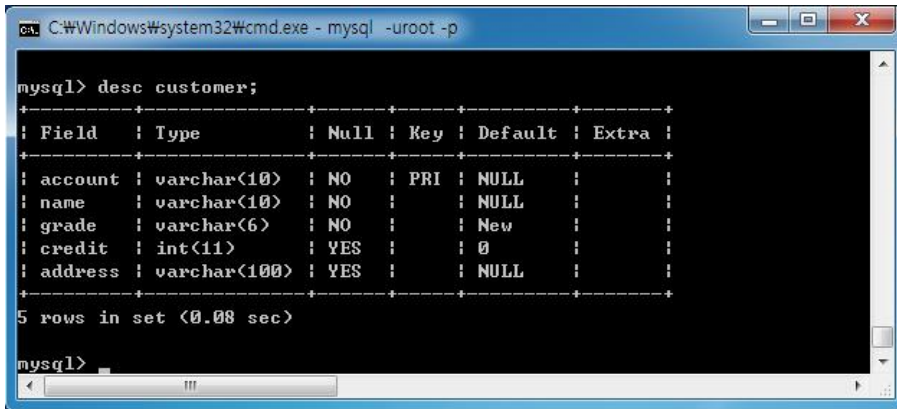
표. ALTER TABLE 기본 형식

ALTER TABLE	테이블이름	
ADD	속성이름	속성정의 [FIRST AFTER 속성이름]
DROP	속성이름	
MODIFY	속성이름	속성정의 [FIRST AFTER 속성이름]
CHANGE	구_속성이름 신_속성이름	속성정의 [FIRST AFTER 속성이름]
RENAME	신_테이블이름	

표. ALTER TABLE 기본 형식 - 속성정의

데이터타입 [NOT NULL | NULL] [DEFAULT 기본값] [AUTO_INCREMENT]

A. 새로운 속성 추가 (ADD)



```

C:\Windows\system32\cmd.exe - mysql -uroot -p

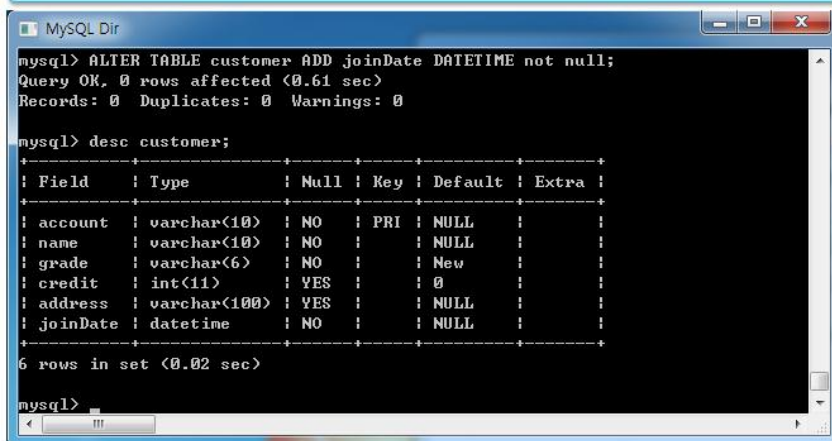
mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| account | varchar(10) | NO   | PRI | NULL    |      |
| name    | varchar(10) | NO   |     | NULL    |      |
| grade   | varchar(6)  | NO   |     | New     |      |
| credit  | int(11)     | YES  |     | 0       |      |
| address | varchar(100) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.08 sec)

mysql>

```

그림. MySQL 적용 전 Customer 릴레이션 구조

ALTER TABLE customer ADD joinDate DATETIME not null;



```

MySQL Dir

mysql> ALTER TABLE customer ADD joinDate DATETIME not null;
Query OK, 0 rows affected (0.61 sec)
Records: 0 Duplicates: 0 Warnings: 0

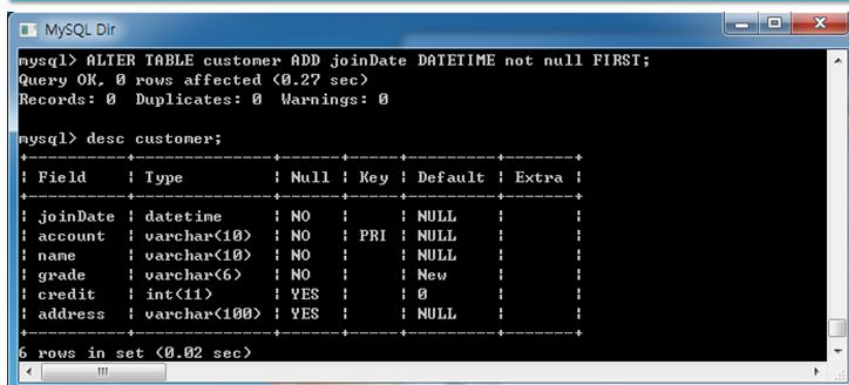
mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| account | varchar(10) | NO   | PRI | NULL    |      |
| name    | varchar(10) | NO   |     | NULL    |      |
| grade   | varchar(6)  | NO   |     | New     |      |
| credit  | int(11)     | YES  |     | 0       |      |
| address | varchar(100) | YES  |     | NULL    |      |
| joinDate | datetime    | NO   |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

mysql>

```

그림. MySQL 적용 후 Customer 릴레이션 구조 - ADD

ALTER TABLE customer ADD joinDate DATETIME not null FIRST;



```

MySQL Dir

mysql> ALTER TABLE customer ADD joinDate DATETIME not null FIRST;
Query OK, 0 rows affected (0.27 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| joinDate | datetime    | NO   |     | NULL    |      |
| 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    |      |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

mysql>

```

그림. MySQL 적용 후 Customer 릴레이션 구조 - ADD

ALTER TABLE customer ADD joinDate DATETIME not null AFTER credit;

```
mysql> ALTER TABLE customer ADD joinDate DATETIME not null AFTER credit;
Query OK, 0 rows affected (0.97 sec)
Records: 0 Duplicates: 0 Warnings: 0

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	
joinDate	datetime	NO		NULL	
address	varchar(100)	YES		NULL	

6 rows in set (0.30 sec)

그림. MySQL 적용 후 Customer 릴레이션 구조 - ADD

ALTER TABLE customer ADD joinDate DATETIME not null
DEFAULT '0000-00-00 00:00:00' AFTER credit;

```
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	
joinDate	datetime	NO		0000-00-00 00:00:00	
address	varchar(100)	YES		NULL	

6 rows in set (0.05 sec)

그림. MySQL 적용 후 Customer 릴레이션 구조 - ADD

ALTER TABLE customer ADD jDate DATETIME not null
DEFAULT '0000-00-00' AFTER credit;

```
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	
jDate	datetime	NO		0000-00-00 00:00:00	
joinDate	datetime	NO		0000-00-00 00:00:00	
address	varchar(100)	YES		NULL	

7 rows in set (0.01 sec)

그림. MySQL 적용 후 Customer 릴레이션 구조 - ADD

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

account	name	grade	credit	jDate	joinDate	address
apple	이남이	VIP	5000	0000-00-00 00:00:00	0000-00-00 00:00:00	경기 용인시
bank	홍길동	Gold	30000	0000-00-00 00:00:00	0000-00-00 00:00:00	서울 영등포구
bird	구선두	Gold	0	0000-00-00 00:00:00	0000-00-00 00:00:00	충남 천안시
calculator	박계산	Gold	100	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL
eagle	박세채	Gold	2450	0000-00-00 00:00:00	0000-00-00 00:00:00	부산 남구
king	오나라	Gold	15000	0000-00-00 00:00:00	0000-00-00 00:00:00	서울 영등포구
pencil	김돌	Silver	350	0000-00-00 00:00:00	0000-00-00 00:00:00	경기 수원시
tons	노마진	New	0	0000-00-00 00:00:00	0000-00-00 00:00:00	NULL

8 rows in set (0.00 sec)

그림. MySQL 적용 후 Customer 속성 값 변화 - ADD

```
ALTER TABLE customer ADD joinD DATETIME DEFAULT 'AAAA' AFTER credit;
ALTER TABLE customer ADD joinD DATETIME DEFAULT '' AFTER credit;
ALTER TABLE customer ADD joinD DATETIME DEFAULT NULL AFTER credit;
```

```
mysql> ALTER TABLE customer ADD joinD DATETIME DEFAULT 'AAAA' AFTER credit;
ERROR 1067 (42000): Invalid default value for 'joinD'
mysql> ALTER TABLE customer ADD joinD DATETIME DEFAULT '' AFTER credit;
ERROR 1067 (42000): Invalid default value for 'joinD'
mysql> ALTER TABLE customer ADD joinD DATETIME DEFAULT NULL AFTER credit;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from customer;
```

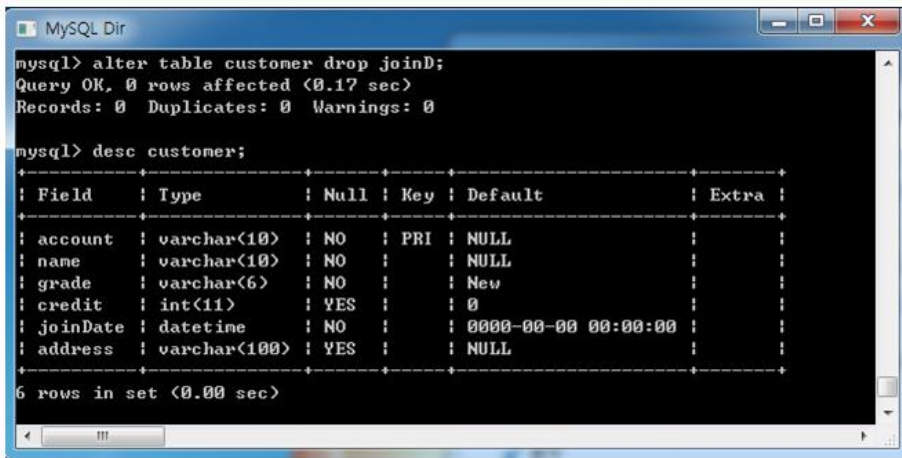
account	name	grade	credit	joinD	joinDate	address
apple	이남이	VIP	5000	NULL	0000-00-00 00:00:00	경기 용인시
bank	홍길동	Gold	30000	NULL	0000-00-00 00:00:00	서울 영등포구
bird	구선두	Gold	0	NULL	0000-00-00 00:00:00	충남 천안시
calculator	박계산	Gold	100	NULL	0000-00-00 00:00:00	NULL
eagle	박세채	Gold	2450	NULL	0000-00-00 00:00:00	부산 남구
king	오나라	Gold	15000	NULL	0000-00-00 00:00:00	서울 영등포구
pencil	김돌	Silver	350	NULL	0000-00-00 00:00:00	경기 수원시
tons	노마진	New	0	NULL	0000-00-00 00:00:00	NULL

8 rows in set (0.00 sec)

그림. MySQL 적용 후 Customer 릴레이션 구조 - ADD

B. 속성 삭제 (DROP)

```
ALTER TABLE customer DROP joinD;
```



```
mysql> alter table customer drop joinD;
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0

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	
joinDate	datetime	NO		0000-00-00 00:00:00	
address	varchar(100)	YES		NULL	

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

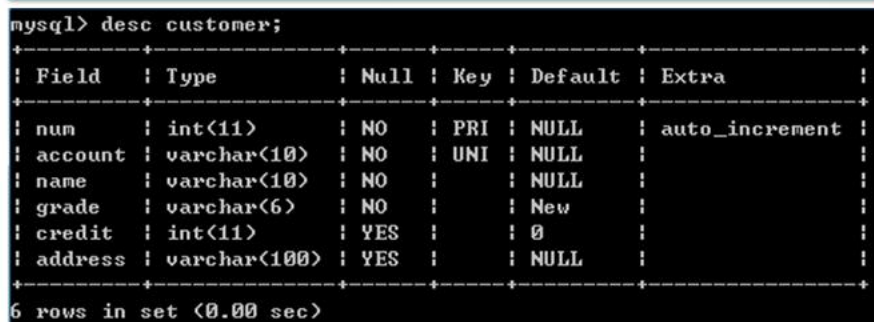
그림. MySQL 적용 후 Customer 릴레이션 구조 - DROP

C. 속성 변경 (CHANGE / MODIFY)

(1) 속성 수정 (CHANGE)

- 속성의 이름과 데이터 타입을 수정함
- 명령어 : ALTER TABLE 테이블명 CHANGE 이전필드명 새로운필드명 데이터타입;

```
ALTER TABLE customer CHANGE credit deposit char(11);
```



```
mysql> desc customer;
```

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

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

그림. 속성 수정 전 Customer 릴레이션 구조

```
ALTER TABLE customer CHANGE credit deposit char(11);
```

```
mysql> desc customer;
```

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
account	varchar(10)	NO	UNI	NULL	
name	varchar(10)	NO		NULL	
grade	varchar(6)	NO		New	
deposit	char(11)	YES		NULL	
address	varchar(100)	YES		NULL	

6 rows in set (0.03 sec)

그림. 속성 수정 후 Customer 릴레이션 구조

```
ALTER TABLE customer CHANGE credit deposit char(11);
```

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

num	account	name	grade	deposit	address
100	bank	홍길동	Gold	30000	서울 종로구
101	apple	이남이	VIP	5000	경기 용인시
102	pencil	김돌	Silver	350	경기 수원시
103	eagle	박세재	Gold	2450	부산 남구
104	bird	구선두		0	충남 천안시
105	king	오나라	Gold	15000	서울 성북구

6 rows in set (0.00 sec)

그림. 속성 수정 후 Customer 데이터

(2) 속성의 데이터타입 수정 (MODIFY)

- 속성의 데이터타입만 수정함
- 명령어 : ALTER TABLE 테이블명 MODIFY 필드명 새로운데이터타입;

```
ALTER TABLE customer MODIFY deposit int;
```

```
mysql> alter table customer modify deposit int;
Query OK, 6 rows affected (0.13 sec)
Records: 6  Duplicates: 0  Warnings: 0

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

그림. 속성 데이터타입 수정 후 Customer 릴레이션구조

D. 제약 조건의 추가 및 삭제 (ADD / DROP)

(1) 기존의 제약 조건 확인하기

- CREATE TABLE을 사용하여 테이블 생성시 적용된 모든 속성, 제약조건 확인
- CREATE TABLE의 SQL에서 선언되지 않은 제약조건이름을 확인할 수 있음
- 명령어 : SHOW CREATE TABLE 테이블이름
- 외래키 제약조건 : MySQL 내부에서 사용된 제약조건이름 확인
· 제약조건을 삭제하기 위하여 확인한 제약조건이름을 사용하여야 함

표. 외래키 제약 조건을 설정을 위한 CREATE TABLE Orders

```
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
);
```

```

C:\Windows\system32\cmd.exe - mysql -uroot -p
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) NOT 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`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+

1 row in set (0.00 sec)

mysql>

```

그림. orders 릴레이션의 제약 조건 확인 하기

(2) 기존의 외래키 제약 조건 삭제하기

- “SHOW CREATE TABLE 테이블명” 질의로 확인한 제약조건이름을 사용하여 삭제
- 명령어 : ALTER TABLE 테이블명 DROP FOREIGN KEY 제약조건이름

ALTER TABLE orders DROP FOREIGN KEY orders_ibfk_2;

```

C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> ALTER TABLE orders DROP FOREIGN KEY orders_ibfk_2;
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>

```

그림. orders 릴레이션의 item 외래키 삭제하기

```

C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> show create table orders;

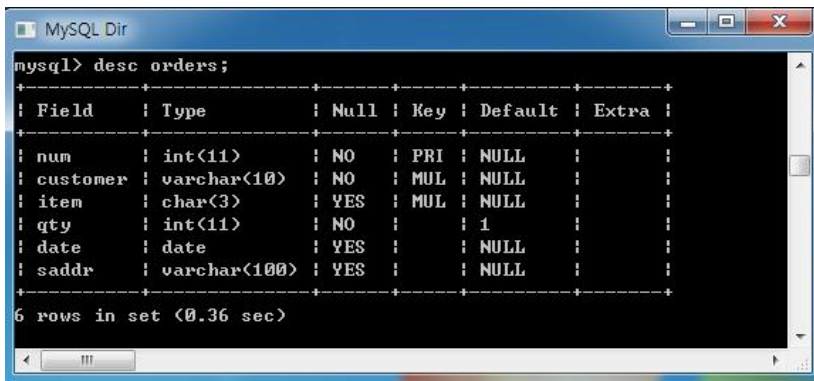
+-----+
| Table | Create Table |
+-----+
| orders | CREATE TABLE `orders` (
  `num` int(11) NOT NULL,
  `customer` varchar(10) NOT NULL,
  `item` char(3) NOT NULL,
  `qty` int(11) NOT NULL DEFAULT '1',
  `date` date DEFAULT NULL,
  `saddr` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`num`),
  KEY `customer` (`customer`),
  KEY `item` (`item`),
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`customer`) REFERENCES `customer` (`account`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+

1 row in set (0.00 sec)

mysql>

```

그림. orders 릴레이션의 item 외래키 제약조건 삭제후 확인

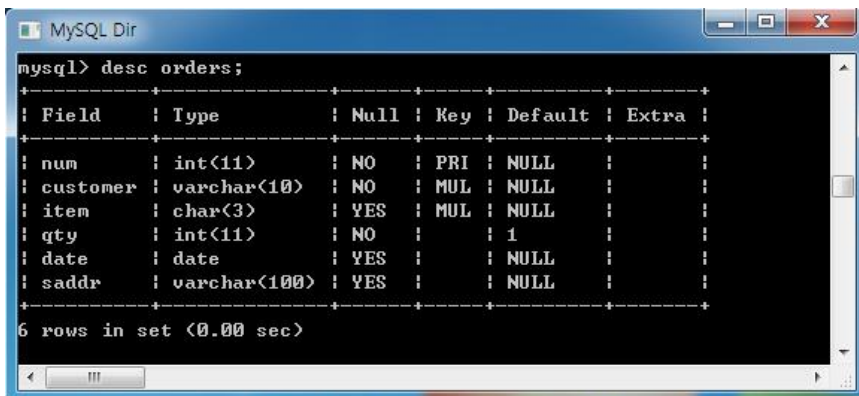


```
mysql> desc orders;
```

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 (0.36 sec)

그림. item 외래키 제약조건 삭제전 테이블 구조



```
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 (0.00 sec)

그림. item 외래키 제약조건 삭제후 테이블 구조
(테이블 구조에는 변화가 없음)

(3) 새로운 외래키 제약 조건 추가

- 새로운 외래키 제약 조건을 테이블에 추가하기 위하여 사용
- 명령어 : ALTER TABLE 테이블명 ADD 외래키제약조건

ALTER TABLE orders ADD FOREIGN KEY (item)
REFERENCES items(code) ON DELETE CASCADE;



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> ALTER TABLE orders ADD FOREIGN KEY ( item )
-> REFERENCES items( code ) ON DELETE CASCADE;
Query OK, 6 rows affected (0.38 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

그림. orders 릴레이션의 외래키 제약조건 옵션 추가하기

```

C:\Windows\system32\cmd.exe - mysql -uroot -p
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) NOT 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`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8
+-----+
1 row in set (0.02 sec)

```

E. 테이블 관련 기타 질의

(1) 데이터베이스내의 모든 테이블 상태 확인하기

- 명령어 : SHOW TABLE STATUS FROM 데이터베이스명
- 확인 가능 정보
 - * 레코드 수
 - * 평균 레코드 길이
 - * 전체 데이터 길이
 - * AUTO_INCREMENT 값
 - * 생성 날짜
 - * 사용언어

```
mysql> show table status from itbank;
```

Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length
customer	InnoDB	10	Compact	6	2730	16384
items	InnoDB	10	Compact	0	0	16384

2 rows in set (0.00 sec)

Max_data_length	Index_length	Data_free	Auto_increment	Create_time
0	16384	0	106	2015-06-14 22:23:47
0	0	0	NULL	2015-06-14 12:48:37

Update_time	Check_time	Collation	Checksum	Create_options	Comment
NULL	NULL	utf8_general_ci	NULL		
NULL	NULL	utf8_general_ci	NULL		

그림. SHOW TABLE STATUS FROM 데이터베이스명의 결과

학습내용2 : 테이블 이름변경 및 삭제

1. 테이블 이름변경

- 명령어 : ALTER TABLE 구_테이블명 RENAME 신_테이블명
- 이름이 변경된 테이블을 참조하는 외래키가 있는 경우
 . 외래키 선언 부분에 테이블 이름 변경이 반영됨

ALTER TABLE customer RENAME customers;

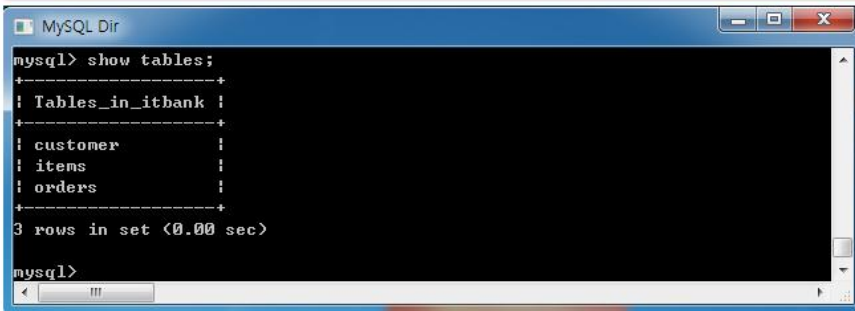


그림. 테이블 이름 변경전 테이블 리스트

ALTER TABLE customer RENAME customers;

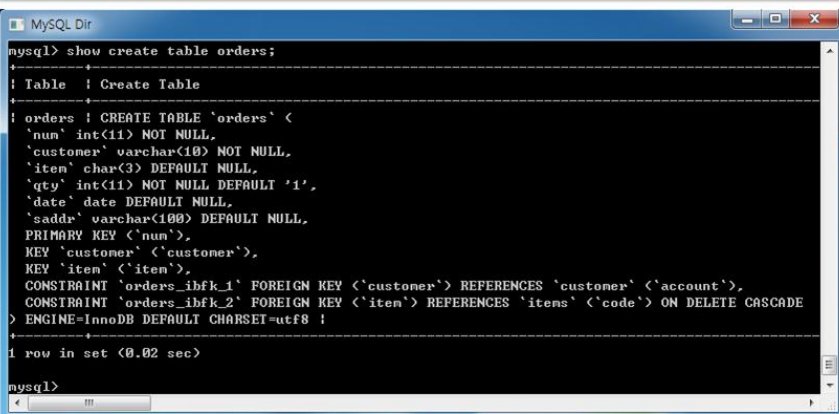


그림. 외래키가 선언된 테이블 생성 질의 (테이블 이름 변경 전)

ALTER TABLE customer RENAME customers;

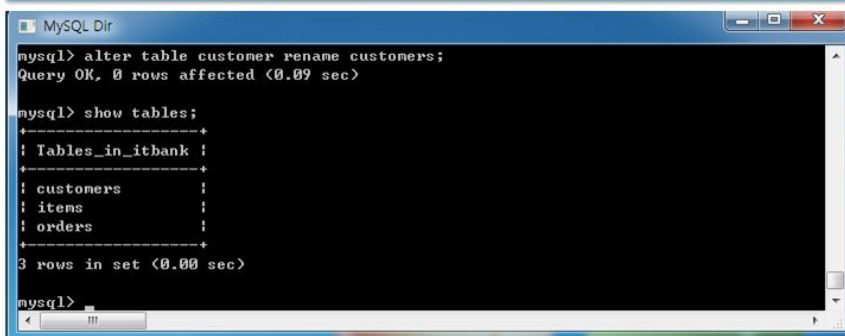
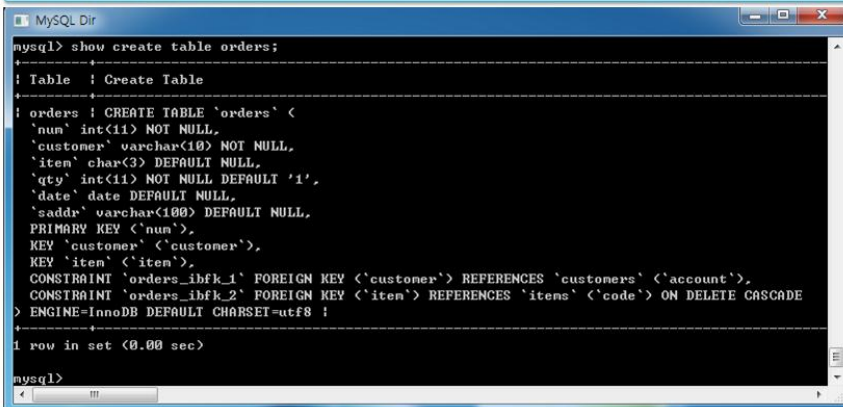


그림. 테이블 이름 변경후 테이블 리스트

ALTER TABLE customer RENAME customers;

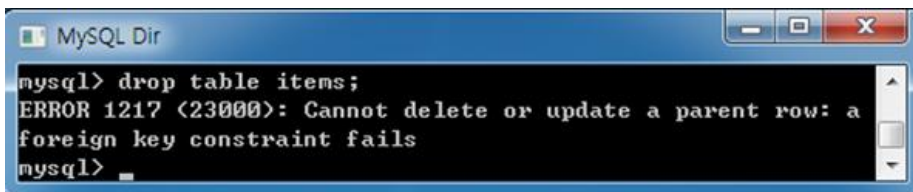


```
mysql> show create table orders;
+-----+-----+
| Table | Create Table |
+-----+-----+
| orders | CREATE TABLE `orders` (
  `num` int(11) NOT NULL,
  `customer` varchar(10) NOT NULL,
  `item` char(3) DEFAULT NULL,
  `qty` int(11) NOT NULL DEFAULT '1',
  `date` date DEFAULT NULL,
  `addr` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`num`),
  KEY `customer` (`customer`),
  KEY `item` (`item`),
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`customer`) REFERENCES `customers` (`account`),
  CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`item`) REFERENCES `items` (`code`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

그림. 외래키가 선언된 테이블 생성 질의 (테이블 이름 변경 후)

2. 테이블 삭제

- 명령어 : DROP TABLE 테이블명 [, 테이블명] [CASCADE | RESTRICT];
- 삭제하는 테이블을 참조하는 테이블이 있는 경우
 - * CASCADE : 같이 삭제
 - * RESTRICT : 삭제 취소

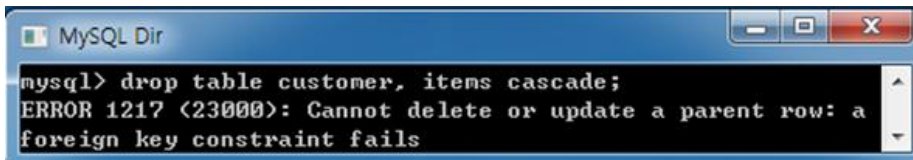


```
mysql> drop table items;
ERROR 1217 (23000): Cannot delete or update a parent row: a
foreign key constraint fails
mysql>
```



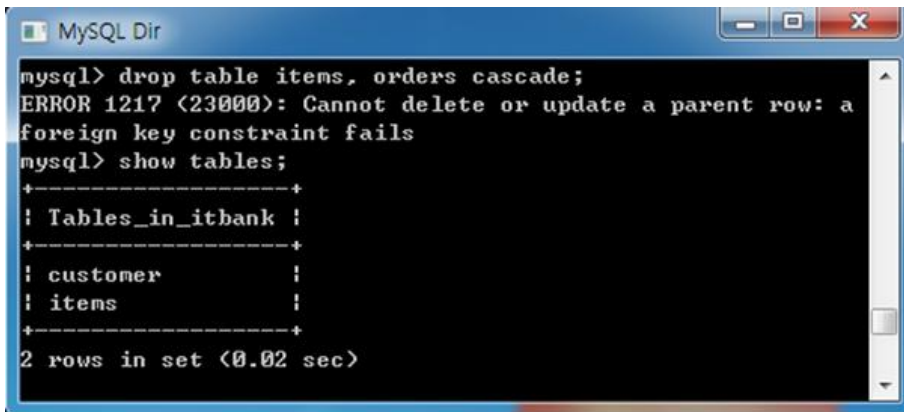
```
mysql> drop table items restrict;
ERROR 1217 (23000): Cannot delete or update a parent row: a
foreign key constraint fails
mysql>
```

그림. 참조하는 테이블이 있는 경우(삭제 취소됨)



```
mysql> drop table customer, items cascade;
ERROR 1217 (23000): Cannot delete or update a parent row: a
foreign key constraint fails
```

그림. 참조하는 테이블이 있는 경우
(MySQL에서 CASCADE가 동작하지 않음)

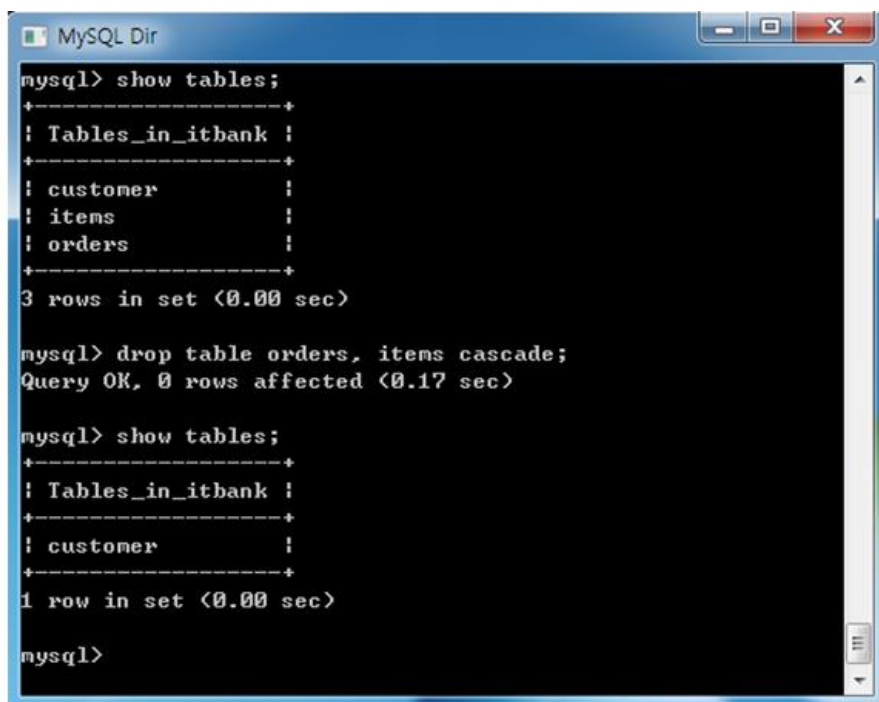


```

mysql> drop table items, orders cascade;
ERROR 1217 (23000): Cannot delete or update a parent row: a
foreign key constraint fails
mysql> show tables;
+-----+
| Tables_in_itbank |
+-----+
| customer          |
| items              |
+-----+
2 rows in set (0.02 sec)

```

그림. 참조하는 테이블이 있는 경우
(items는 삭제 취소, orders는 삭제됨)



```

mysql> show tables;
+-----+
| Tables_in_itbank |
+-----+
| customer          |
| items              |
| orders             |
+-----+
3 rows in set (0.00 sec)

mysql> drop table orders, items cascade;
Query OK, 0 rows affected (0.17 sec)

mysql> show tables;
+-----+
| Tables_in_itbank |
+-----+
| customer          |
+-----+
1 row in set (0.00 sec)

mysql>

```

그림. 참조하는 테이블이 있는 경우
(참조하는 테이블이 먼저 삭제되므로,
참조되는 테이블도 삭제됨)

【학습정리】

1. 테이블을 수정하기 위하여 ALTER TABLE 명령어를 사용한다.
2. 새로운 속성을 추가하기 위하여 ADD), 기존의 속성을 삭제하기 위하여 DROP, 속성을 수정하기 위하여 CHANGE와 MODIFY 사용한다.
3. 테이블을 만들 때 적용하였던 제약조건은 수정할 수 없고, 삭제 후 다시 추가하여야 한다.