

학습목표

1. 스키마 수정
2. 스키마 삭제

학습내용

- ALTER/DROP 명령어에 대해 알 수 있습니다.
- 테이블 삭제 명령어에 대해 알 수 있습니다.

사전퀴즈

1. 테이블 스키마를 수정하기 위해서는 ALTER TABLE/DATABASE 명령어를 사용한다.

정답: O

해설 : ALTER TABLE/DATABASE는 이미 생성된 테이블이나 DB의 스키마를 변경할 때 사용한다.

2. TRUNCATE TABLE은 테이블 데이터와 테이블을 삭제한다.

정답 : X

해설 : 테이블을 삭제하려면 DROP TABLE을 사용한다. 테이블은 놔두고 데이터만 삭제하려면 TRUNCATE TABLE을 사용한다.

수업

1. 스키마 수정

이미 생성된 스키마에 대해 수정할 경우 사용한다.

테이블 컬럼 추가/삭제/수정

- ALTER TABLE 테이블명 ADD 컬럼명 데이터타입
- ALTER TABLE 테이블명 DROP COLUMN 컬럼명
- ALTER TABLE 테이블명 **CHANGE** 컬럼명 new_컬럼명 데이터타입(**컬럼명변경**)
- ALTER TABLE 테이블명 **MODIFY** 컬럼명 데이터타입(**컬럼타입변경**)

기본키 제약조건 추가/기본키 제약조건 삭제

- ALTER TABLE 테이블명 ADD PRIMARY KEY(**컬럼명**)
- ALTER TABLE 테이블명 DROP PRIMARY KEY

UNIQUE 제약조건 추가/삭제

- **하나의 컬럼**에 대해 UNIQUE 제약조건 추가
ALTER TABLE 테이블명 **ADD UNIQUE**(**컬럼명**)
- **두 개 이상의 컬럼(복합컬럼)**에 대해 UNIQUE 제약조건 추가
ALTER TABLE 테이블명 **ADD CONSTRAINT 제약조건명 UNIQUE**(**컬럼명1, 컬럼명2**)
- ALTER TABLE 테이블명 DROP UNIQUE 제약명

외래키 제약조건 추가/삭제

- ALTER TABLE 테이블명 ADD FOREIGN KEY(**컬럼명**) REFERENCES 원테이블명(**원컬럼명**)
- ALTER TABLE 테이블명 DROP FOREIGN KEY 컬럼명

테이블명 수정

- ALTER TABLE 테이블명 RENAME new_테이블명;

CHECK 제약조건 추가/삭제

- ALTER TABLE 테이블명 ADD CHECK 조건
- ALTER TABLE 테이블명 ADD CONSTRAINT 조건명 CHECK(조건절)
- ALTER TABLE 테이블명 DROP CHECK 조건명

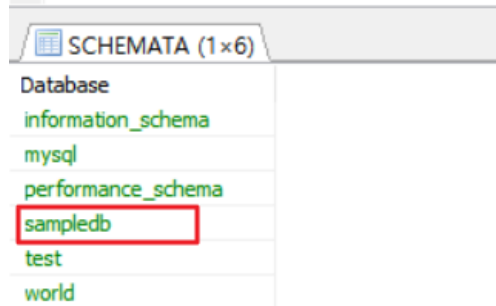
DEFAULT 제약조건 추가/삭제

- ALTER TABLE 테이블명 ALTER 컬럼명 SET DEFAULT 기본값
- ALTER TABLE 테이블명 ALTER 컬럼명 DROP DEFAULT

예제 1-1

sampleDB 데이터베이스를 정의하고 생성하시오

```
1 create database sampleDB;
2
3 show databases;
```



Database
information_schema
mysql
performance_schema
sampledb
test
world

예제 1-2

컬럼으로 이름(Name), 주소(Address), 전화번호(Telephone)를 가지는 BusinessCard 테이블을 정의하고 생성하시오.
단, 이름, 주소, 전화번호는 문자열(varchar) 최대 길이 255자로 지정함

```

7 create table BusinessCard(
8     Name varchar(255),
9     Address varchar(255),
10    Telephone varchar(255)
11 )
12 ;
13
14 insert into BusinessCard
15 values('Bob', 'Seoul', '123-456')
16 ;
17
18 insert into BusinessCard
19 values('Sam', 'Seoul', '789-456')
20 ;
21
22 select *
23 from BusinessCard
24 ;

```

businesscard (3×2)		
Name	Address	Telephone
Bob	Seoul	123-456
Sam	Seoul	789-456

예제 1-3

테이블 컬럼 추가(ID int)

```

MariaDB [world]> alter table BusinessCard add column id int;
Query OK, 0 rows affected (0.014 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [world]> show create table BusinessCard\G;
***** 1. row *****
      Table: BusinessCard
Create Table: CREATE TABLE `businesscard` (
  `Name` varchar(255) DEFAULT NULL,
  `Address` varchar(255) DEFAULT NULL,
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) DEFAULT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.000 sec)

```

* 참고 *

컬럼의 데이터 타입이 int일 때, 데이터 크기를 지정하지 않았음에도 자동으로 데이터크기가 11이 지정된다.

이와 관련한 좋은 설명글을 구글링하다가 발견

출처: <http://great-artist.tistory.com/84> [Good artists copy, great artists steal.]

숫자 컬럼에 INT를 선언할 경우 자동적으로 INT(11)이 입력된다. INT형 옆의 11이란 숫자는 입력된 값의 자릿수를 의미한다. 해당값은 ZEROFILL (이하 ZF) 속성이 지정됐을 경우에만 영향을 받는다. 즉 ZF 속성을 사용하지 않는다면 의미가 없다. 그렇다면 ZF의 기능은 무엇일까? 입력된 값의 자릿수를 일관되게 맞추려는 목적으로 사용된다. INT(5) 약 해당 컬럼에 입력된 값의 자릿수가 지정된 자릿수보다 작다면 좌측부터 0으로 채워진다. 만약 지정된 자릿수보다 크다면 아무런 영향을 받지 않는다.

테스트




테이블을 만들어서 테스트 해보자. INT형에 ZEROFILL값을 1, 2, 3, 4, 5, 11로 설정 ZF(ZEROFILL)에 체크한다. ZF에 체크를 하면 mysql은 자동적으로 UN(Unsigned) 속성을 추가한다.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
int_1	INT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
int_2	INT(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
int_3	INT(3)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
int_4	INT(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
int_5	INT(5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
int	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

테이블에 아래의 값을 입력해 보고 나서 테이블의 값을 확인해보자.

1	1	1	1	1
12	12	12	12	12
123	123	123	123	123
1234	1234	1234	1234	1234
12345	12345	12345	12345	12345

숫자값의 자릿수가 ZF값보다 작다면 왼쪽부터 비어있는 곳은 0으로 채워진다.

Result Grid				Filter Rows:	<input type="text"/>	Export:	
	int_1	int_2	int_3	int_4	int_5	int	
▶	1	01	001	0001	00001	1	
	12	12	012	0012	00012	12	
	123	123	123	0123	00123	123	
	1234	1234	1234	1234	01234	1234	
	12345	12345	12345	12345	12345	12345	
	123456	123456	123456	123456	123456	123456	

결론

MySQL은 자동적으로 UNSIGNED 속성을 추가한다. 좌측부터 0으로 채우려면 양수로만 값이 입력되어야 하기 때문이다. 추가로 ZF는 보여지는 형식을 정의하는 것이지 실제 저장되는 방식과는 무관하다. int형을 입력하면 자동적으로 INT(11)이 된다. 기본값이 11인 이유는 int형의 최대값이 10자리이기 때문으로 보인다. 11자리로 해놓으면 최대값에 도달한다고 하더라도 자릿수가 보기 좋게 0부터 시작해서 정렬되기 때문으로 보인다.

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
INT	4	-2147483648	2147483647
		0	4294967295

예제 1-4

컬럼명 변경(Address → Address1)

```

MariaDB [world]> show create table BusinessCard\G;
***** 1. row *****
      Table: BusinessCard
Create Table: CREATE TABLE `businesscard` (
  `Name` varchar(255) DEFAULT NULL,
  `Address` varchar(255) DEFAULT NULL,
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.000 sec)

ERROR: No query specified

MariaDB [world]> alter table BusinessCard change Address Address1 varchar(255);
Query OK, 0 rows affected (0.039 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [world]> show create table BusinessCard\G;
***** 1. row *****
      Table: BusinessCard
Create Table: CREATE TABLE `businesscard` (
  `Name` varchar(255) DEFAULT NULL,
  `Address1` varchar(255) DEFAULT NULL,
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.000 sec)

```

예제 1-5

컬럼타입 변경(Address1 varchar(255) → char(255))

```

MariaDB [world]> alter table BusinessCard modify Address1 char(255);
Query OK, 2 rows affected (0.077 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [world]> show create table BusinessCard\G;
***** 1. row *****
      Table: BusinessCard
Create Table: CREATE TABLE `businesscard` (
  `Name` varchar(255) DEFAULT NULL,
  `Address1` char(255) DEFAULT NULL,
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.005 sec)

```

예제 1-6

ID 컬럼 PK, Unique 설정

```

MariaDB [world]> alter table BusinessCard add primary key(id);
Query OK, 0 rows affected (0.059 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [world]> show create table BusinessCard\G;
***** 1. row *****
      Table: BusinessCard
Create Table: CREATE TABLE `businesscard` (
  `Name` varchar(255) DEFAULT NULL,
  `Address1` char(255) DEFAULT NULL,
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.000 sec)

```

```

MariaDB [world]> alter table BusinessCard add unique(id);
Query OK, 0 rows affected (0.025 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [world]> show create table BusinessCard\G;
***** 1. row *****
      Table: BusinessCard
Create Table: CREATE TABLE `businesscard` (
  `Name` varchar(255) DEFAULT NULL,
  `Address1` char(255) DEFAULT NULL,
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.000 sec)

```

예제 1-7

Address1 기본값 설정('Seoul')

```

MariaDB [world]> alter table BusinessCard alter Address1 set default 'Seoul';
Query OK, 0 rows affected (0.031 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [world]> show create table BusinessCard\G;
***** 1. row *****
      Table: BusinessCard
Create Table: CREATE TABLE `businesscard` (
  `Name` varchar(255) DEFAULT NULL,
  `Address1` char(255) DEFAULT 'Seoul',
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.000 sec)

```

예제 1-8

테이블명 변경 (BusinessCard → BusinessCards)

```

MariaDB [world]> alter table BusinessCard rename BusinessCards;
Query OK, 0 rows affected (0.013 sec)

MariaDB [world]> show create table BusinessCard##G;
ERROR 1146 (42S02): Table 'world.businesscard' doesn't exist
ERROR: No query specified

MariaDB [world]> show create table BusinessCards##G;
***** 1. row *****
Table: BusinessCards
Create Table: CREATE TABLE `businesscards` (
  `Name` varchar(255) DEFAULT NULL,
  `Address1` char(255) DEFAULT 'Seoul',
  `Telephone` varchar(255) DEFAULT NULL,
  `id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.006 sec)

```

2. 스키마 삭제

데이터베이스 삭제

- DROP DATABASE 데이터베이스명

테이블 삭제

- DROP TABLE 테이블명

내용과 테이블 전체 삭제

- DELETE FROM 테이블명

레코드를 일일이 하나씩 지움. 테이블 스키마는 유지

- TRUNCATE TABLE 테이블명

테이블 내용만 지움. 테이블 스키마는 유지. 전용명령어

예제 2-1

BusinessCard 테이블 생성/내용 추가한 후 테이블 스키마는 남겨두고 레코드를 모두 비우시오(Truncate)

```

1 create table BusinessCard(
2     Name varchar(255),
3     Address varchar(255),
4     Telephone varchar(255)
5 )
6 ;
7
8 insert into BusinessCard
9 values('Bob', 'Seoul', '123-456')
10 ;
11
12 insert into BusinessCard
13 values('Sam', 'Seoul', '789-456')
14 ;
15
16 select *
17 from BusinessCard
18 ;
19

```

businesscard (3×2)		
Name	Address	Telephone
Bob	Seoul	123-456
Sam	Seoul	789-456

```

20 truncate table BusinessCard
21 ;
22
23 select *
24 from BusinessCard
25 ;

```

businesscard (3×0)		
Name	Address	Telephone

예제 2-2

테이블을 삭제하시오

```

27 drop table BusinessCard
28 ;
29
30 select *
31 from BusinessCard
32 ;

```

```

146
147 select * from BusinessCard ;
148 /* SQL 오류 (1146): Table 'world.businesscard' doesn't exist */
149 /* 영향 받은 행: 0 잘못된 행: 0 경고: 0 지속 시간 0 of 1 쿼리: 0.000 sec. */

```

DB 스키마를 변경해야 할 때 기존에 들어있는 데이터는 어떻게 될까?(해야 할까?)

전문가의견

기존에 존재하지 않는 컬럼이 추가되는 경우에는 주로 NULL 값이 추가됩니다. 하지만 존재하던 데이터의 값이 사라질(삭제 될) 경우에는 데이터가 삭제되면서 문제가 발생할 경우도 있습니다. 그 경우에는 미리 영향받을 데이터를 백업한 후 나중에 별

도로 추가해 주어야 하는 경우가 있으니 조심해야 합니다. 특히, 스키마 변경에는 항상 영향받을 범위를 한정해 주는 것이 좋습니다.