

# MySQL DDL문 완전정복 SQL 독학 강의#21편 -sTricky

sTricky 2020. 6. 12. 17:19

---

## MySQL DDL문 완전정복 SQL 독학 강의#21편 -sTricky



### [콘텐츠 index](#)

0. DDL 문이란?

- 1. CREATE 문
- 2. ALTER 문
- 3. TRUNCATE 문
- 4. DROP 문

## 전편 강의 보러 가기 ##

[2020/06/02 - \[Database/sql 강의\] - insert into on duplicate key MySQL merge SQL 독학 강의#20편 -sTricky](#)

	insert into on duplicate key MySQL merge SQL 독학 강의#20...  stricky.tistory.com
--	---

안녕하세요.

오늘은 MySQL의 DDL문장들에 대해서 강의 포스팅을 작성합니다.

DBMS를 사용하고 관리를 하는 데 있어서 가장 중요한 것이 DDL문이라고도 할 수 있습니다.

일반 사용자라면 잘 모르고 넘어갈 수도 있으나, DB를 관리하거나 개발을 하시는 분들은 꼭 알아야 할 내용입니다.

0. DDL 문이란?

우선 DDL 문장이 무엇이나?

DB내에는 많은 오브젝트들이 있습니다. 이런 오브젝트들을 생성하고 변경하고 관리하는 문이 바로 DDL문입니다.

DDL (Data Definition Language)에는 여러 가지가 있습니다.

DDL 명령어	정의
CREATE	새로운 오브젝트나 스키마등을 생성 할 때 사용하는 명령어
ALTER	이미 만들어져 있는 오브젝트의 내용을 변경 할 때 사용하는 명령어
TRUNCATE	테이블의 데이터를 전부 삭제하고 테이블이 차지하고 있던 공간을 반납 하는 명령어
DROP	테이블 자체를 삭제하는 명령어 입니다. 테이블을 비롯하여 인덱스도 함께 삭제

그럼, 위 4개의 DDL 명령어에 대해서 하나씩 알아보도록 하겠습니다.

## CREATE 문

### 1. CREATE 문

CREATE 문으로 대부분의 MySQL 오브젝트들을 생성할 수 있습니다.

Table, Prodedule, Function, Index, User 등등..

모든 오브젝트들에 대한 CREATE 문을 다 소개해드릴 순 없기 때문에 이 중에서 Table과 user를 생성하는 CREATE 문의 예제로 설명을 드리도록 하겠습니다.

#### 1) CREATE TABLE

우선 CREATE TABLE 구문의 예제를 하나 보겠습니다.

```
create table test.com_com_c
(
  CD_NM          varchar(100) not null comment '코드명',
  KOR_CD_NM      varchar(100) not null comment '한글코드명',
  COM_CD_TYP_CD  varchar(3)   not null comment '공통코드유형코드',
  UPR_CD_NM      varchar(100) null comment '상위코드명',
  CD_TBL_NM      varchar(100) null comment '코드테이블명',
  USE_YN         varchar(1)   not null comment '사용여부',
  RGST_DTM       datetime     not null comment '등록일시',
  RGSTR_ID       varchar(30)   not null comment '등록자ID',
  UPD_DTM        datetime     not null comment '수정일시',
  UPDR_ID        varchar(30)   not null comment '수정자ID',
  UPD_PRGM_ID    varchar(50)   not null comment '수정프로그램ID'
  primary key (CD_NM)
)
comment '공통_공통코드' engine = InnoDB;
```

위에서 예로 든 것은 공통 코드 테이블의 만드는 CREATE TABLE DDL 문입니다.

우선 "create table..."으로 시작을 하고 "스키마명. 테이블명"을 써줍니다. 위 예제를 들어 설명을 하면 "test"라는 스키마에 "com\_com\_c"라는 테이블을 생성하겠다는 의미 이죠.

그리고 "("를 열고 테이블 내 생성할 칼럼을 하나씩 써줍니다. 여기에서 가장 좌측에 "CD\_NM, KOR\_CD\_NM" 등을 컬럼명이라고 하고 그 다음에 나오는 "varchar(100)"을 데이터타입이라고 부릅니다.

이 두 내용을 필수적으로 들어가야 하는 내용입니다. 풀어서 설명하면 CD\_NM이라는 칼럼을 만드는데 varchar타입의 문자열을 저장 할 수 있게끔 컬럼을 만들고 그 길이는 100byte로 하겠다는 의미 입니다. 그 뒤에 나오는것들은 안써도 되지만 좀 더 디테일한 컬럼 정보를 지정해줄 수 있습니다.

뒤에 따라오는 "not null"이란 것은 이 칼럼은 null이 들어올 수 없다, 즉, 어떤 값이라도 꼭 있어야 한다는 의미입니다. 이렇게 테이블을 만들고 나서 insert 문을 통해서 해당 칼럼에 어떤 값도 넣지 않으면 insert시 에러가 나게 됩니다.

반대로 not null이 아니라 null이라고 써두면 null을 허용한다는 의미가 되겠죠.

그리고 마지막에 "comment '코드명'"이라고 되어 있는데, 이건 "CD\_NM"에 '코드명'이라고 코멘트를 붙이겠다는 의미입니다. 이것을 논리 칼럼명이라고 할 수도 있고, column description 이라고도 표현할 수 있습니다.

그리고 쪽~ 아래로 내려오면 ")" 괄호가 닫히기 전 바로 윗줄에 "primary key (CD\_NM)"이라는 부분이 있습니다. "test.com\_com\_c" 테이블의 PK(Primary Key) 칼럼을 "CD\_NM"으로 하겠다는 의미입니다.

만약 두 개 이상 복수의 칼럼을 PK로 설정을 하고 싶다면 "primary key (CD\_NM, other\_column, the\_other\_column)" 이런 식으로 ", "로 구분하여 나열하면 됩니다.

단, PK로 설정하는 칼럼은 위에서 "not null"로 정의되어야 합니다.

이젠 ")" 괄호 밖으로 나와서 "comment '공통\_공통 코드'"라고 되어 있는 구문을 보실 수 있으실 텐데, 이는 테이블의 논리명, 또는 코멘트를 이야기하는 것입니다.

마지막으로 "engine = InnoDB"이라는 것을 볼 수 있습니다. 이 구문은 MySQL에는 두 가지 테이블 스토리지 엔진이 존재하는데, "InnoDB"와 "MyISAM"이 그것입니다. 어떤 스토리지 엔진으로 테이블 데이터를 저장하겠느냐를 선택하는 것입니다.

이를 CREATE TABLE DDL 실행 시에 명시하지 않으면 첫 번째 스키마 내 초기 설정 스토리지 엔진 값, 두 번째 해당 MySQL의 초기 설정 스토리지 엔진 값으로 생성이 됩니다.

두 스토리지 엔진의 차이점에 대해서는 추후 다시 포스팅을 하도록 하겠습니다.

우선 이렇게 간단하게 테이블 생성문에 대해서 설명을 드렸습니다.

## 2) CREATE USER

CREATE USER 부분은 해당 내용에 관해서 이미 포스팅한 게 있어 그것으로 대체하도록 하겠습니다.

아래 링크로 가셔서 확인하시기 바랍니다.

<https://stricky.tistory.com/100>

	[mariadb] DB 계정 별 권한 확인, 적용, 부여, 회수 계정 생성 stricky.tistory.com
--	--

## ALTER 문

### 2. ALTER 문

ALTER 문 역시 ALTER 문을 통해서 MySQL DB 내 다양한 오브젝트들을 수정 및 변경할 수 있으나, 지면(?) 관계상 테이블 내 컬럼들의 다양한 수정 및 변경 예문으로 대신하겠습니다.

우선 다양한 ALTER 문의 예제를 보겠습니다.

```
-- not null 옵션의 컬럼을 null로 변경
alter table send_log modify createdAt datetime default current_timestamp() null;

-- sendType 컬럼명을 sendType2으로 변경
alter table send_log change sendType sendType2 varchar(4) not null;

-- 컬럼 코멘트를 변경
alter table send_log modify fee decimal(20,8) null comment '컬럼코멘트';

-- 신규 컬럼을 추가 하는데, fee 컬럼 뒤로 위치
alter table send_log add column_8 int null after fee;

-- receive 컬럼의 데이터 타입을 변경
alter table send_log modify receive int not null;
```

위 주석에 있는 내용처럼 다양한 방법의 ALTER문을 사용할 수 있습니다.

컬럼명을 바꿀수도 있고, 컬럼의 데이터타입을 변경할 수도 있으며, 컬럼 코멘트를 변경 할 수 있습니다. 그리고 신규 컬럼을 추가하거나 삭제할 때도 ALTER문이 사용됩니다.

## TRUNCATE 문

### 3. TRUNCATE 문

TRUNCATE 문은 테이블의 데이터 삭제와 함께 해당 테이블이 쓰고 있던 디스크 공간을 반납하게 됩니다. TRUNCATE문 자체가 테이블을 DROP 했다가 다시 CREATE 하기 때문입니다. 이점이 delete와 가장 다른 점입니다.

TRUNCATE 문의 예제를 한번 보시겠습니다.

```
TRUNCATE TABLE address_svc_m.testtt;
```

그리고 MySQL에서는 ORACLE과 달리 TRUNCATE시 인덱스가 DROP 되지 않습니다.

또, 기존에 AUTO\_INCREMENT로 지정하여 값이 증가하던 칼럼이 초기화됩니다.

기본적으로 TRUNCATE 사용 시 주의할 점은 DELETE보다 전체 데이터를 삭제할 때는 더 빠르나 이후 데이터 복구가 불가능하다는 단점이 있습니다.

이점을 주의하셔야 합니다.

## DROP 문

### 4. DROP 문

DROP 문은 테이블 및 기타 MySQL 오브젝트들을 삭제하는 명령어입니다. 만약 이 명령어를 통해서 테이블을 삭제한다면 해당 테이블의 row들은 물론 생성되어 있던 인덱스와 함께 저장공간까지 모두 반납되게 됩니다.

DROP 문의 사용 예제를 보시겠습니다.

```
DROP TABLE address_svc_m.testtt;
```

이 DDL명령을 실행하고 나면 테이블이 삭제됩니다.

그러면 안 되겠지만, MySQL에서는 실수로 테이블을 DROP 했을 때는 BINLOG파일을 이용해서 해당 데이터를 살릴 수도 있습니다.

자세한 내용은 추후 포스팅으로 찾아뵙겠습니다.

오늘 이렇게 MySQL의 DDL문에 대해서 공부를 해보았습니다.

물론 앞서 배웠던 SELECT문들도 정말 중요하지만, 데이터 관리 시점에서 볼 때는 DDL문이 훨씬 더 중요합니다.

DB 관리자를 희망하시는 분들은 이 내용을 잘 숙지하시기 바랍니다.

한번 DDL을 잘못 실행하면 정말... 큰 고난을 겪을 수도 있으니, DDL문은 항상 신중하게 두 번 세 번 확인 후 실행하는 습관을 들이셔야 합니다.