

```
1 REM Author :
2 REM Date :
3 REM Objective : Chapter 9. DDL
4 REM Environment : Ubuntu Server 20.04 LTS, HeidiSQL 10.2.0, MySQL Community Server 5.7.34.0
5
6
```

7 REM DDL(Data Definition Language)

- 8 -데이터베이스의 **Object** 구조를 생성, 변경, 삭제하는 명령어
- 9 -**CREATE, ALTER, DROP, RENAME, TRUNCATE**

12 REM Naming Convention

13 출처: <https://taptorestart.tistory.com/entry/MySQL-데이터베이스명-테이블명-컬럼명은-어떻게-지어야-할까>

14 1. General

- 15 1)Ensure the name **is unique and** does **not** exist **as** a reserved keyword.
- 16 2)Keep the **length to** a maximum **of 30** bytes—in practice this **is 30** characters unless you **are using** a multi-byte **character set**.
- 17 3)**Names** must **begin with** a letter **and** may **not end with** an underscore.
- 18 4)**Only use** letters, numbers **and** underscores **in names**.
- 19 5)Avoid the **use of** multiple consecutive underscores—these can be hard **to read**.
- 20 6)**Use** underscores **where** you would naturally **include** a **space in** the name (**first** name becomes first_name).
- 21 7)Avoid abbreviations **and if** you have **to use** them make sure they **are** commonly understood.

23 2. Tables

- 24 1)**Use** a collective name **or, less** ideally, a plural form. **For** example (**in order of** preference) staff **and** employees.
- 25 2)**Do not prefix with** tbl **or any** other such descriptive **prefix or** Hungarian notation.
- 26 3)Never give a **table** the same name **as** one **of** its columns **and** vice versa.
- 27 4)Avoid, **where** possible, concatenating two **table names** together **to create** the name **of** a relationship **table**. Rather **than** cars_mechanics prefer services.

29 3. Columns

- 30 1)Always **use** the singular name.
- 31 2)**Where** possible avoid simply **using** id **as** the **primary** identifier **for** the **table**.
- 32 3)**Do not add** a **column with** the same name **as** its **table and** vice versa.
- 33 4)Always **use** lowercase **except where** it may make sense **not to** such **as** proper nouns.

36 REM Database 생성

37 1. Database

- 38 -**Table**이나 **View, Index** 등 데이터베이스 내에 정의하는 모든 것
- 39 2. The **CREATE DATABASE statement is** used **to create** a **new SQL database**.

40 3. Syntax

```
41 CREATE DATABASE [IF NOT EXISTS] db_name
42 [create_option] ...
43
44 create_option: [DEFAULT] {
45     CHARACTER SET [=] charset_name | COLLATE [=] collation_name
46 }
```

48 4. Example

```
49 CREATE DATABASE testDB;
50 CREATE DATABASE study_db default CHARACTER SET UTF8;
```

53 REM Database 삭제

- 54 1. The **DROP DATABASE statement is** used **to drop** an existing **SQL database**.

55 2. Syntax

```
56 DROP DATABASE [IF EXISTS] db_name
```

57 3. Example

```
58 DROP DATABASE testDB;
```

62 REM 테이블의 생성

63 1. CREATE TABLE

- 64 -The **CREATE TABLE statement is** used **to create** a **new table in** a **database**.

65 2. Syntax

```
66 CREATE TABLE [IF NOT EXISTS] table_name
67 (
68     column_name    column_definition
69     [CONSTRAINT] PRIMARY KEY |
70     [CONSTRAINT] UNIQUE |
71     [CONSTRAINT] FOREIGN KEY |
```

```

72     CHECK (expression)
73 )
74 column_definition : {
75     data_type [NOT NULL | NULL] [DEFAULT default_value]
76     [AUTO_INCREMENT] [UNIQUE][PRIMARY KEY]
77     [COMMENT 'string']
78 }
79 table_option : {
80     AUTO_INCREMENT [=] value |
81     [DEFAULT] CHARACTER SET [=] charset_name |
82     COMMENT [=] 'string'
83 }
84

```

1)일반적 생성 방법

```

86 CREATE TABLE table_name ( column_name datatype, ...);
87

```

```

88 CREATE TABLE IF NOT EXISTS dept1
89 (
90     deptno TINYINT,
91     dname VARCHAR(13),
92     loc VARCHAR(14) DEFAULT 'Seoul'
93 )DEFAULT CHARACTER SET UTF8
94

```

2)SUB_QUERY 에 의해 검색된 테이블과 동일한 구조로 생성

-테이블 복사, CTAS(Create Table ~ As Select ~)방법
 -주의할 점은 기존 테이블의 제약조건이 모두 없어진다는 점

```

99 CREATE TABLE dept2
100 AS
101 SELECT * FROM dept;
102
103 CREATE TABLE emp2
104 AS
105 SELECT EMPNO, ENAME FROM EMP;
106
107 CREATE TABLE emp3
108 AS
109 SELECT * FROM emp
110 WHERE DEPTNO = 10;
111
112 CREATE TABLE emp4
113 AS
114 SELECT * FROM emp where 1=0; <-- 구조만 복사
115

```

3)Guide Lines

-지정한 열 이름을 사용하여 테이블을 생성하며 SELECT 문에 의해 검색된 행을 테이블에 삽입.
 -열 사양이 제공되는 경우 열 수는 하위 질의 SELECT 목록의 열 수와 동일해야 한다.
 -열 사양이 제공되지 않는 경우 해당 테이블의 열 이름은 하위 질의의 열 이름과 동일.

4)조회

```

122 SELECT table_name
123 FROM user_tables;
124
125 SELECT object_name, object_type
126 FROM user_objects;
127
128 CREATE TABLE dept30
129 AS
130 SELECT empno, ename, sal * 12 "Annual Salary", hiredate
131 FROM emp
132 WHERE deptno = 30;
133

```

5)사용예

--부서별로 인원수, 평균급여, 급여의 합, 최소급여, 최대급여를 포함하는 emp_calcu table을 생성하라.

```

136 CREATE TABLE emp_calcu
137 AS
138 SELECT deptno, COUNT(*) AS cnt, AVG(sal) AS salavg, SUM(sal) AS salsum,
139     MIN(sal) AS minsal, MAX(sal) AS maxsal
140 FROM emp
141 GROUP BY deptno;
142

```

--사원번호, 이름, 업무, 입사일자, 부서번호만 포함하는 emp_temp table을 생성하는데, 자료는 포함하지 않고 스키마만 생성하시오.

```

145 CREATE TABLE emp_temp

```

```

146 AS
147 SELECT empno, ename, job, hiredate, deptno
148 FROM emp
149 WHERE 1 < 0;
150
151
152 --테이블을 다음차트를 기반으로 DEPARTMENT 테이블을 생성하고, 생성한 후 테이블 구조를 확인하시오.
153 --열이름 :      id      name
154 --데이터유형 :    INT      VARCHAR
155 --길이 :          7      25
156 CREATE TABLE department
157 (
158     id      INT(7),
159     name    VARCHAR(25)
160 );
161
162 DESC department
163
164
165 --테이블을 다음차트를 기반으로 EMPLOY 테이블을 생성하고, 생성한 후 테이블 구조를 확인하시오.
166 --열이름 :      id      last_name  first_name  dept_id
167 --데이터유형 :  INT      VARCHAR   VARCHAR     INT
168 --길이 :        7      25          25          7
169 CREATE TABLE employ
170 (id      INT(7),
171     last_name  VARCHAR(25),
172     first_name VARCHAR(25),
173     dept_id    INT(7)
174 );
175
176 DESC employ
177
178
179
180 REM TABLE 삭제
181 1. The DROP TABLE statement is used to drop an existing table in a database.
182 -table의 모든 구조와 데이터가 삭제
183 -DDL이기 때문에 TRANSACTION 이 COMMIT된다.
184 -영구히 삭제, 되돌릴 수 없다.
185 2. Syntax
186     DROP TABLE table_name;
187
188     DROP TABLE emp30;
189     DROP TABLE dept;
190
191 3. Guideline
192 1) 삭제하려는 테이블의 기본 키나 고유 키를 다른 테이블에서 참조해서 사용하는 경우에는 해당 테이블을 제거할 수 없다.
193 2) 이런 경우에는 참조하는 테이블을 먼저 제거한 후 해당 테이블을 삭제해야 한다.
194
195
196
197 REM ALTER TABLE
198 1. The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.
199 2. The ALTER TABLE statement is also used to add and drop various constraints on an existing table.
200 -테이블의 스키마 변경
201 -Column을 추가/삭제하거나 제약조건을 추가/삭제하는 작업
202
203 3. Column Add
204 -열 추가
205 -새 열은 마지막 열, 위치 지정 불가능
206 -이미 행을 포함하고 있는 테이블이라면 새로 들어갈 열의 모든 행은 초기에 널 값을 갖는다.
207 -Syntax
208     ALTER TABLE table_name
209     ADD [COLUMN] col_name column_definition
210         [FIRST | AFTER col_name]
211         [, column datatype]...);
212
213     ALTER TABLE dept30
214     ADD job VARCHAR(9);
215
216     ALTER TABLE dept30
217     ADD sal INT AFTER ename;
218
219     ALTER TABLE dept1

```

```
220 ADD bigo VARCHAR(15) FIRST;
```

```
221
```

```
222
```

223 2. Column Modify

224 -Column의 데이터 유형, Default값, NOT NULL 제약조건에 대한 변경

225 -Syntax

```
226 ALTER TABLE table_name
```

```
227 MODIFY [COLUMN] column_name column_definition
```

```
228 [FIRST | AFTER col_name]
```

```
229
```

230 -MODIFY Guide Lines

231 --숫자 열의 너비 또는 전체 자릿수를 증가가능.

232 --열이 널 값만 포함하고 테이블에 행이 없는 경우 열의 너비를 줄일 수 있다.

233 --열이 널 값을 포함하면 데이터 유형을 변경할 수 있다.

234 --열의 기본값을 변경하면 변경 이후에 테이블에 삽입되는 항목에만 영향을 준다.

235 --테이블의 구조 변경(데이터 타입, 길이), 만일 기존 데이터가 있을 경우에는 CHAR와 VARCHAR 사이의 타입 변경만 가능

236 --컬럼의 크기 변경 역시 기존에 저장된 데이터의 길이와 같거나 클 경우에만 변경이 가능

237 --즉 해당 컬럼에 자료가 없을 경우에는 데이터타입과 컬럼의 크기를 변경가능

238 --해당 컬럼에 자료가 있을 경우에는 데이터타입 변경 불가능하고 크기는 늘릴 수만 있음.

```
239
```

```
240 ALTER TABLE dept30
```

```
241 MODIFY ename VARCHAR(15);
```

```
242
```

```
243 ALTER TABLE dept1
```

```
244 MODIFY bigo VARCHAR(30);
```

```
245
```

```
246
```

247 -사용예

248 --위 예제에서 생성한 EMPLOY테이블의 last_name열에 긴 성을 가진 사원의 성을 저장할 수 있도록 길이를 50으로 수정한 후, 내용을 확인하시오

```
249
```

```
250
```

251 3. Column Remove

252 -열 삭제

253 -Guide Lines

254 --열은 데이터를 포함하거나 포함하지 않거나 삭제 할 수 있다.

255 --한 번에 하나의 열만 삭제할 수 있다.

256 --테이블에는 최소 하나의 열이 있어야 하므로 열을 삭제하려면 테이블에 열이 둘 이상 있어야 한다.

257 --삭제된 열은 복구할 수 없다.

258 -Syntax

```
259 ALTER TABLE table_name
```

```
260 DROP [COLUMN] col_name
```

```
261
```

```
262 ALTER TABLE emp1
```

```
263 DROP COLUMN JOB;
```

```
264
```

```
265
```

266 4. CHANGE COLUMN

267 -Column의 이름을 변경

268 -Syntax

```
269 ALTER TABLE table_name
```

```
270 CHANGE [COLUMN] old_col_name new_col_name column_definition
```

```
271 [FIRST | AFTER col_name]
```

```
272
```

```
273 ALTER TABLE dept1
```

```
274 CHANGE COLUMN bigo bigo1 VARCHAR(15);
```

```
275
```

```
276
```

```
277
```

278 REM TABLE TRUNCATE

279 1. Empties a table completely.

280 2. ROLLBACK 불가능

281 3. DELETE는 모든 행을 제거할 수 있지만, 저장공간을 해제할 수 없다.

282 4. 모든 행을 제거할 때는 DELETE보다 TRUNCATE TABLE 문을 사용하자.

283 5. Syntax

```
284 TRUNCATE [TABLE] table_name
```

```
285
```

```
286 TRUNCATE TABLE emp30;
```

```
287
```

288 6. DROP TABLE은 테이블 자체를 제거하지만, TRUNCATE는 테이블은 존재하면서 데이터만 제거하기에 구조는 남아있다.

```
289
```

```
290
```

```
291
```

292 REM Rename TABLE

```

293 1. table의 이름 변경
294 2. Syntax
295 RENAME TABLE old_table_name TO new_table_name
296
297 RENAME TABLE dept2 TO dept3;
298
299
300
301 REM 테이블에 주석 문 추가
302 1. Table Comment
303 -CREATE TABLE 또는 ALTER TABLE에 table_option으로 추가
304 COMMENT [=] 'string'
305
306 ALTER TABLE dept1
307 COMMENT = '부서 정보 테이블입니다';
308
309
310 2. Column Comment
311 -CREAET TABLE 또는 ALTER TABLE에 column_definition으로 추가
312 COMMENT 'string'
313
314 ALTER TABLE dept1
315 MODIFY COLUMN deptno TINYINT COMMENT '부서코드'
316
317
318
319 REM DATA TYPE
320 1. CHAR
321 1) 고정길이의 문자 데이터타
322 2) 기본값 및 최소크기 : 1Byte
323 3) 최대크기 125Bytes
324 4) 나머지 공간을 여백으로 채워서 처음 정의된 공간을 모두 사용하는 타입
325
326 2. VARCHAR
327 1) Variable Character의 약자
328 2) 가변길이의 문자 데이터타
329 3) 기본값 및 최소 크기 : 1Byte
330 4) 최대크기 : 65535Bytes
331 5) 여백으로 채우지 않고 필요한 공간만 사용
332
333 ABCDE --> char(8) --> ABCDE__ _
334 ABCDE --> varchar(8) --> ABCDE
335
336 3. 정수
337 1) TINYINT(1), SMALLINT(2), INT(4), BIGINT(8)
338 2) 부호없는 정수를 저장할 때는 UNSIGNED 예약을 뒤에 붙여준다.
339
340 4. 실수
341 1) 숫자값을 -38자리 ~ 38자리 저장
342 2) FLOAT(4), DOUBLE(8), DECIMAL(m, d)
343
344 5. DECIMAL(m, d)
345 1) 전체자리수(m)중 소수점 이하 자리수(d)
346 2) 전체자리수는 십진수의 총 갯수이고, 소수점 이하 자리수는 소수점의 오른쪽에 있는 숫자 갯수.
347
348 6. DATE
349 1) A.D 1000년 1월 1일 ~ A.D. 9999년 12월 31일까지의 날짜
350 2) SELECT CURDATE();
351 3) 'YYYY-MM-DD' 형식
352
353 7. DATETIME
354 1) '1000-01-01 00:00:00.000000' to '9999-12-31 23:59:59.999999'
355 2) 'YYYY-MM-DD HH:MM:SS' 형식

```