

```

1 REM Author :
2 REM Date :
3 REM Objective : Chapter 10. Constraints
4 REM Environment : Ubuntu Server 20.04 LTS, HeidiSQL 10.2.0, MySQL Community Server 5.7.34.0
5
6
7 REM Column 의 옵션
8 1. DEFAULT OPTION
9 2. Constraints
10
11
12 REM DEFAULT OPTION
13 1. 열에 기본값을 부여할 수 있다.
14 2. 이 옵션은 열 값이 없는 행을 삽입할 경우 열에 널 값이 입력되는 것을 방지
15 3. 리터럴, 표현식을 기본값으로 사용할 수 있다.
16 4. 기본 표현식은 해당 열의 데이터 유형과 일치해야 한다.
17 5. ex)
18 hiredate DATETIME DEFAULT CURRENT_TIMESTAMP <--- NOW() 불가
19
20 6. 테이블을 생성할 때 혹은 스키마를 변경할 때 칼럼에 직접 적용
21 CREATE TABLE table_name
22 (column datatype [DEFAULT expression][,...]);
23
24 7. 실습1
25
26 CREATE TABLE Department
27 AS
28 SELECT * FROM dept
29 WHERE 0 > 1;
30
31 ALTER TABLE Department
32 ADD COLUMN hiredate DATETIME DEFAULT CURRENT_TIMESTAMP;
33
34 INSERT INTO Department(deptno, hiredate)
35 VALUES(10, DEFAULT);
36
37 8. 실습2
38
39 CREATE TABLE Jusorok
40 (
41     bunho SMALLINT,
42     gender CHAR(6) DEFAULT '남자'
43 );
44
45 INSERT INTO Jusorok VALUES (1, '여자');
46 INSERT INTO Jusorok VALUES (2, DEFAULT);
47 SELECT * FROM jusorok;
48
49
50 REM Constraints(제약조건)
51 -사용자가 원하는 조건의 데이터만 유지하기 위한 즉, 데이터의 무결성을 유지하기 위한 데이터베이스의 보편적인 방법
52 -테이블의 특정 칼럼에 설정하는 제약이다.
53
54 INSERT INTO dept
55 VALUES(10, 'TEST', 'SEOUL'); --무결성 제약조건 위배
56 --Dept 테이블에 이미 10번 부서가 존재하고 있기 때문
57
58 INSERT INTO dept
59 VALUES(NULL, NULL, 'SEOUL'); --오류
60 --NULL 을 부서번호에 삽입할 수 없다.
61
62 1. 특징
63 1) 사용자에 의해 발생한 잘못된 DML 문이 실행되지 않는다.
64 2) 제약 조건에 대한 모든 정보가 자료사전에 저장된다.
65 3) 원할 때는 언제든지 기능을 비활성화할 수 있고, 또한 활성화 할 수 있다.
66 4) 처리 결과가 즉시 사용자에게 보여진다.
67 5) 하나의 칼럼에 여러 개의 제약 조건을 설정할 수 있다.
68 6) 부적절한 데이터의 입력, 수정, 삭제를 방지할 목적
69 7) 제약조건은 테이블레벨/컬럼레벨 제약조건이 있다.
70 8) 제약조건은 종속성이 있을 경우 삭제를 방지한다.
71 9) 테이블에서 삽입, 수정, 삭제를 할 때 마다 조건에 대한 규칙을 적용한다.
72 10) 제약조건은 각각의 RDBMS 마다 다를 수 있다.
73 11) Oracle 에서는 제약 조건 이름을 지정하지 않으면, 자동으로 SYS-Cnumber 형식으로 생성한다.
74 12) 사용자가 해당 제약조건에 위배되는 명령을 요청했을 때, 오라클은 제약조건 이름과 함께 에러를 내보낸다.

```

13) DESC 명령어는 NOT NULL 제약 조건을 확인할 수 있지만, 다른 제약조건을 확인할 수 없다.

## 2. 제약조건의 종류

### 1) Column Level Constraint

- 컬럼단위로 제약조건을 부여할 때
- 5가지의 제약조건이 모두 가능
- NOT NULL은 컬럼레벨에서만 부여가능
- Syntax

```
column_name data_type  
[CONSTRAINT] constraint_type
```

```
id VARCHAR2(10) CONSTRAINT PRIMARY KEY,  
id VARCHAR2(10) PRIMARY KEY
```

```
CREATE TABLE Student
```

```
(  
    hakbun CHAR(4) PRIMARY KEY,  
    name VARCHAR(20) NOT NULL,  
    kor TINYINT NOT NULL  
)
```

```
SELECT CONSTRAINT_NAME, constraint_type, TABLE_name  
FROM information_schema.table_constraints  
WHERE TABLE_NAME = 'student';
```

### 2) Table Level Constraint

- 각각의 컬럼의 정의와 개별적으로 정의한다.
- 하나 이상의 컬럼을 지정가능하다.
- NOT NULL을 제외한 나머지 제약조건의 정의가 가능
- Syntax

```
column_name data_type,  
...  
...  
[CONSTRAINT constraint_name] constraint_type
```

```
id VARCHAR2(10),  
name VARCHAR2(20),  
age NUMBER(3),  
CONSTRAINT testpk_id_pk PRIMARY KEY(id)
```

```
CREATE TABLE Student
```

```
(  
    hakbun CHAR(4),  
    name VARCHAR(20) NOT NULL,  
    kor TINYINT(3) NOT NULL,  
    CONSTRAINT Student_hakbun_pk PRIMARY KEY(hakbun)  
);
```

## 3. 데이터 무결성 제약조건 5가지

- PRIMARY KEY(PK)
- FOREIGN KEY(FK)
- UNIQUE(UK)
- NOT NULL(NN)
- CHECK(CK)

## 4. Constraints Naming Convention guide lines

- 제약조건의 이름은 Oracle과 달리 Query에서는 사용할 일은 없다.
- constraint\_name : tablename\_columnname\_constrainttype
- ex : emp\_empno\_pk, emp\_deptno\_fk

REM PRIMARY KEY

EMPNO	ENAME	JOB	DEPTNO
7499	ALLEN	SALESMAN	30
	JONES	MANAGER	20
	JONES	SALESMAN	10

-현재 EMPNO 가 **UNIQUE** 로 설정된 상태임  
-**UNIQUE** 는 **NULL** 이 허용되기 때문에 추가적으로 **NOT NULL**이 필요  
-그래서 **PRIMARY KEY** 가 필요함.

1. 테이블에 저장된 행 데이터를 고유하게 식별하기 위한 기본키
2. 기본적으로 테이블마다 주 식별자는 있어야 한다.
3. 하나의 테이블에 하나의 기본키 제약만 정의할 수 있다.
4. **NOT NULL** 제약조건과 **NO DUPLICATE** 제약조건(**UNIQUE**)이 부여된다.
5. **UNIQUE INDEX** 가 자동으로 생성
6. 테이블 레벨 제약조건과 컬럼 레벨 제약조건 모두 가능
7. Syntax

#### 1) Column Level Constraints

```
CREATE TABLE Test
(
    id NUMBER(3) PRIMARY KEY,
    name VARCHAR2(20)
);
```

#### 2) Table Level Constraints

```
CREATE TABLE Test1
(
    id NUMBER(3),
    name VARCHAR2(20),
    CONSTRAINT test1_id_pk PRIMARY KEY(id)
);
```

### REM FOREIGN KEY

1. 관계형 데이터베이스에서 테이블 간의 관계를 정의하기 위해 기본키를 다른 테이블의 외래키로 복사하는 경우 생성.
2. 자식테이블(참조하는 쪽)에서 정의한다.
3. MASTER TABLE(부모테이블) vs DETAIL TABLE(자식테이블)  
-MASTER TABLE(parent) 은 참조 당하는 쪽(예:dept table)을 의미하고 DETAIL TABLE(child)은 참조하는 쪽(예:emp table)을 의미한다.
4. 외래키가 바라보는(참조하는) 부모테이블(마스터테이블)의 키는 **PRIMARY KEY, UNIQUE KEY**로 정의된 열을 지정할 수 있으며, 데이터타입이 일치해야 하고, **NULL** 일 수 있다.
5. **ON DELETE CASCADE**를 지정하면, 부모테이블의 레코드를 삭제할 때, 참조된 행을 삭제할 수 있다.

#### 6. Syntax

##### 1) Column Level Constraint

```
column_name data_type
[CONSTRAINT] FOREIGN KEY
REFERENCES table_name (column_name)
[ON DELETE CASCADE]
```

```
ex) deptno NUMBER(2) FOREIGN KEY REFERENCES dept (deptno);
```

##### 2) Table Level Constraint

```
column_name data_type,
...,
...,
[CONSTRAINT constraint_name] FOREIGN KEY(column_name)
REFERENCES table_name (column_name) [ON DELETE CASCADE]
```

```
ex) deptno NUMBER(2),
    FOREIGN KEY(deptno) REFERENCES dept (deptno)
```

#### 7. FOREIGN KEY 의 주의점

- 1) **FOREIGN KEY** 값은 MASTER TABLE에서 존재하는 값과 일치해야하거나 **NULL**이 되어야 한다.
- 2) MASTER TABLE을 먼저 생성해야 한다.
- 3) MASTER TABLE에 **PRIMARY KEY** 또는 **UNIQUE KEY** 로 설정된 열을 DETAIL TABLE에서 참조해야 한다.
- 4) MASTER TABLE 과 DETAIL TABLE 에서 참조하는 쪽과 참조 당하는 쪽의 열은 자료형이 서로 일치해야 한다.

#### 8. 실습

-부모 테이블 먼저 생성  
**CREATE TABLE** dept10

```
(
    deptno TINYINT PRIMARY KEY,
```

```

221     dname VARCHAR(15),
222     LOCAL VARCHAR(1)
223 );
224
225 --자식테이블 생성
226 CREATE TABLE emp10
227 (
228     empno INT PRIMARY KEY,
229     ename VARCHAR(15),
230     deptno TINYINT,
231     CONSTRAINT FOREIGN KEY(deptno) REFERENCES dept10(deptno)
232 );
233
234 CREATE TABLE dept_copy
235 (
236     deptno    NUMBER(2),
237     dname     VARCHAR2(14),
238     loc       VARCHAR2(13),
239     CONSTRAINT dept_copy_deptno_pk PRIMARY KEY(deptno)
240 );
241
242 CREATE TABLE emp_copy
243 (
244     empno INT,
245     ename VARCHAR(10),
246     hiredate DATE,
247     deptno SMALLINT,
248     CONSTRAINT emp_copy_empno_pk PRIMARY KEY(empno),
249     CONSTRAINT emp_copy_deptno_fk FOREIGN KEY(deptno)
250     REFERENCES dept_copy (deptno)
251 );
252
253

```

#### REM UNIQUE KEY

EMPNO	ENAME	JOB	DEPTNO
7499	ALLEN	SALESMAN	30
7499	JONES	MANAGER	20

- 열 또는 열 집합 모든 값들의 유일성을 보장하기 위한 키
- 중복된 값을 가질 수 없다는 것을 보증한다.
- 하나의 테이블에서 여러 칼럼에 명시할 수 있다.
- PRIMARY KEY와 유사하지만, NULL 허용이 된다는 것이 차이다.
- 자동으로 INDEX가 부여된다.
- Syntax
  - Column Level Constraint  
column\_name data\_type [CONSTRAINT] UNIQUE,
  - Table Level Constraint  
column\_name data\_type,  
[CONSTRAINT constraint\_name] UNIQUE(column\_name)
- 실습

```

274
275 CREATE TABLE dept_clone
276 (
277     deptno SMALLINT,
278     dname  VARCHAR(20),
279     loc    VARCHAR(20),
280     CONSTRAINT dept_clone_deptno_uk UNIQUE(deptno)
281 );
282
283 CREATE TABLE unitest
284 (
285     deptno SMALLINT UNIQUE,
286     dname  CHAR(14),
287     loc    CHAR(13)
288 );
289
290 --OR
291 CREATE TABLE unitest
292 (
293     deptno SMALLINT,
294     dname  CHAR(14),

```

```

295         loc CHAR(13),
296     CONSTRAINT unittest_deptno_uk UNIQUE (deptno),
297 );
298
299
300 REM NOT NULL
301 1. 값이 NULL 이 되지 않는다는 것을 보장한다.
302 2. INSERT, UPDATE 시 NULL 을 허용하지 않겠다는 의미
303 3. NOT NULL 제약조건이 없는 열은 기본적으로 NULL 이 허용된다.
304 4. 반드시 컬럼레벨 제약조건에서만 지정가능
305 5. PRIMARY KEY 는 기본적으로 NOT NULL 을 가지고 있음.
306 6. Syntax
307     column datatype NOT NULL
308
309 7. 실습
310
311 CREATE TABLE dept_copy1
312 (
313     deptno SMALLINT,
314     dname VARCHAR(20) NOT NULL DEFAULT 'Accounting',
315     loc VARCHAR(20),
316     CONSTRAINT dept_copy1_deptno_pk PRIMARY KEY(deptno),
317     CONSTRAINT dept_copy1_loc_uk UNIQUE(loc)
318 );
319
320 INSERT INTO dept_copy1(deptno, loc)
321 VALUES(20, 'Pusan');
322
323 CREATE TABLE dept14
324 (
325     deptno SMALLINT,
326     dname VARCHAR(15),
327     LOCAL CHAR(1),
328     CONSTRAINT dept14_dname_nn NOT NULL(dname)
329 ); -- ERROR 발생
330
331 CREATE TABLE NULLDemo
332 (
333     deptno SMALLINT,
334     dname VARCHAR(10) NOT NULL,
335     loc VARCHAR(10),
336     CONSTRAINT nulldemo_deptno_pk PRIMARY KEY(deptno)
337 );
338
339 INSERT INTO nulldemo VALUES (1, 'aaa', '');
340 INSERT INTO nulldemo VALUES (2, 'bbb', null);
341 INSERT INTO nulldemo VALUES (3, NULL, 'seoul');
342
343
344 REM CHECK
345 1. 행이 만족해야 하는 조건을 정의하는 것이다.
346 2. Syntax
347     1) Column Level Constraint
348         column_name data_type CHECK(condition)
349     2) Table Level Constraint
350         -column_name data_type,
351         ""
352         [CONSTRAINT constraint_name] CHECK(condition)
353
354 3. 실습
355 CREATE TABLE dept_copy
356 (
357     deptno SMALLINT,
358     dname VARCHAR(20),
359     loc VARCHAR(20),
360     CONSTRAINT dept_copy_deptno_pk PRIMARY KEY(deptno),
361     CONSTRAINT dept_copy_deptno_ck CHECK(deptno IN(10,20,30,40,50))
362 );
363
364 --사원번호, 사원명, 급여, 성별 4개의 칼럼을 갖는 테이블을 생성하시오,
365 --단 사원번호가 기본키로, 사원명은 NOT NULL로, 급여는 500에서 5000사이의 값만
366 --저장할 수 있고, 성별은 남자는 M, 여자는 F 둘중의 하나만 저장할 수 있어야 한다.
367 CREATE TABLE emp2
368 (

```

```

369 empno INT PRIMARY KEY,
370 ename VARCHAR(10) NOT NULL,
371 sal DECIMAL(7,2) CHECK(sal BETWEEN 500 AND 5000),
372 gender CHAR(1) CHECK(gender IN('M', 'F'))
373 );
374
375 CREATE TABLE zippost
376 (
377     post1 CHAR(3),
378     post2 CHAR(3),
379     address VARCHAR(100) NOT NULL,
380     CONSTRAINT zippost_post_pk PRIMARY KEY(post1, post2)
381 );
382
383 CREATE TABLE member
384 (
385     id INT,
386     name VARCHAR(10) NOT NULL,
387     gender CHAR(1),
388     jumin1 CHAR(6),
389     jumin2 CHAR(7),
390     tel CHAR(13),
391     post1 CHAR(3),
392     post2 CHAR(3),
393     address VARCHAR(100),
394     CONSTRAINT member_id_pk PRIMARY KEY(id),
395     CONSTRAINT member_gender_ck CHECK(gender IN ('1', '2')),
396     CONSTRAINT member_tel_uk UNIQUE(tel),
397     CONSTRAINT member_post_fk FOREIGN KEY(post1, post2)
398         REFERENCES zippost(post1, post2)
399 );
400
401
402 REM DICTIONARY 에서 제약 조건 검색하기
403 1. DESC 명령어는 NOT NULL 제약 조건을 확인할 수 있지만, 다른 제약조건을 확인할 수 없다.
404 2. 제약조건 확인
405     information_schema.table_constraints;
406     -constraint_catalog, constraint_schema, constraint_name, table_schema, table_name, constraint_type
407
408 3. DESC information_schema.table_constraints
409     --CONSTRAINT_TYPE
410     --PRIMARY KEY
411     --FOREIGN KEY
412     --UNIQUE
413
414
415 REM 제약 조건 추가
416 1. 제약 조건의 추가, 삭제는 가능하지만, 변경은 불가능
417 2. 제약 조건의 활성화, 비활성화 가능
418 3. NOT NULL 제약 조건은 MODIFY 절을 이용하여 추가
419 4. NOT NULL 은 Column Level 에서만 가능
420 5. Syntax
421     ALTER TABLE table_name
422     ADD [CONSTRAINT constraint_name] constraint_type(column);
423
424 6. 실습
425
426 DROP TABLE emp1;
427
428 CREATE TABLE emp1
429 AS
430 SELECT empno, ename, job, deptno
431 FROM emp;
432
433 ALTER TABLE emp1
434 ADD
435     CONSTRAINT emp1_deptno_fk FOREIGN KEY(deptno)
436     REFERENCES dept(deptno);
437
438 CREATE TABLE emp_clone
439 AS
440 SELECT empno, ename, job
441 FROM emp
442 WHERE deptno = 10;

```

```

443
444 ALTER TABLE emp_clone
445 ADD CONSTRAINT emp_clone_empno_pk PRIMARY KEY(empno);
446
447 ALTER TABLE emp_clone
448 MODIFY ename VARCHAR(10) CONSTRAINT emp_clone_nn NOT NULL;
449
450
451 1. 아래와 같이 dept 및 emp 테이블의 제약조건의 이름을 나열하는 질의를 작성하시오.
452 CONSTRAINT_NAME
453 -----
454 dept_deptno_pk
455 emp_empno_pk
456 emp_deptno_fk
457 emp_mgr_fk
458
459
460 2. 사원테이블을 emp_copy라는 테이블을 생성하시오. 사원이름에 UNIQUE키를 부여하시오.
461 CREATE TABLE emp_copy
462 AS
463 SELECT empno, ename, sal
464 FROM emp
465 WHERE deptno = 20;
466
467 ALTER TABLE emp_copy
468 ADD CONSTRAINT emp_copy_ename_uk UNIQUE (ename);
469
470
471 REM 제약조건의 삭제
472 1. ALTER TABLE의 DROP 절을 사용하여 기존의 테이블에 있는 제약조건을 삭제할 수 있다.
473 2. Syntax
474 ALTER TABLE table_name
475 DROP CONSTRAINT constraint_name
476 OR
477 DROP [{ PRIMARY KEY(column_name) | FOREIGN KEY(column_name) | UNIQUE(column_name) }]
478
479 3. 실습
480
481 ALTER TABLE emp_copy
482 DROP CONSTRAINT emp_copy_ename_uk;
483
484 ALTER TABLE emp_copy
485 ADD PRIMARY KEY(empno);
486
487 ALTER TABLE emp_copy
488 DROP PRIMARY KEY;
489
490 ALTER TABLE emp1
491 DROP CONSTRAINT emp1_ename_nn;

```