

다양한 예제로 쉽게 배우는

오라클 SQL 과 PL/SQL

8장 제약조건을 배웁니다

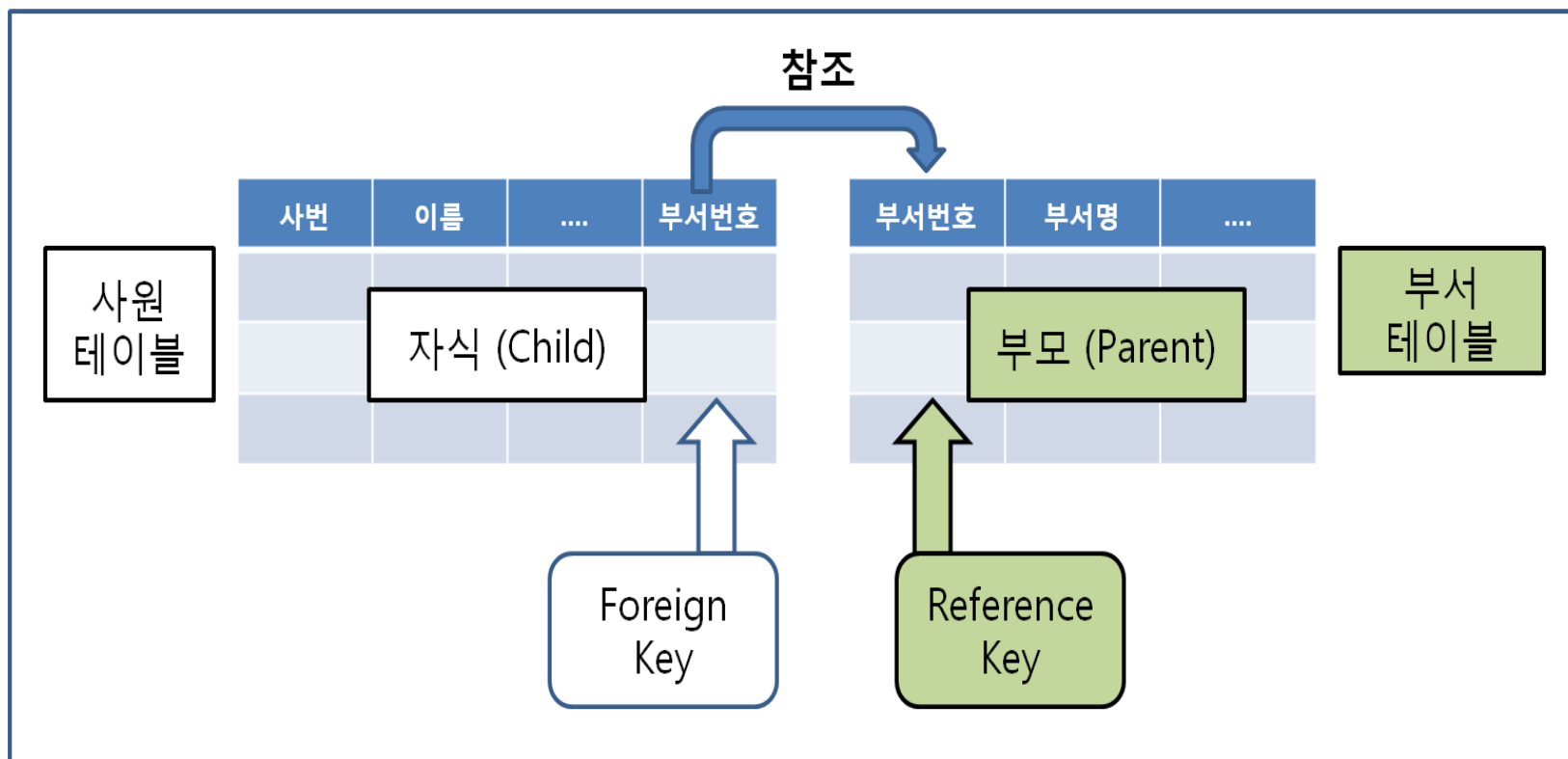
8. 제약조건을 배웁니다

1. 제약 조건의 종류

조건 이름	의 미
NOT NULL	이 조건이 설정 된 컬럼에는 NULL 값이 입력되지 못합니다.
UNIQUE	이 조건이 설정 된 컬럼에는 중복된 값이 입력되지 못합니다.
PRIMARY KEY	이 조건은 NOT NULL + UNIQUE 의 의미를 가지며 테이블 내에서 데이터들끼리의 유일성을 보장하는 컬럼에 설정 할 수 있으며 테이블 당 1개만 설정할 수 있습니다.
FOREIGN KEY	이 조건은 다른 테이블의 컬럼을 참조해서 무결성 검사를 하게 됩니다.
CHECK	이 조건으로 설정된 값만 입력을 허용하고 나머지는 거부됩니다.

8. 제약조건을 배웁니다

- FOREIGN KEY 제약조건



8. 제약조건을 배웁니다

2. 각 제약 조건의 설정 방법

1) 테이블 생성시에 동시에 설정하기

```
SCOTT> CREATE TABLE emp3
2 ( no NUMBER(4)
3     CONSTRAINT emp3_no_pk PRIMARY KEY ,
4   name VARCHAR2(10)
5     CONSTRAINT emp3_name_nn NOT NULL,
6   jumin VARCHAR2(13)
7     CONSTRAINT emp3_jumin_nn NOT NULL
8     CONSTRAINT emp3_jumin_uk UNIQUE ,
9   area NUMBER(1)
10     CONSTRAINT emp3_area_ck CHECK ( area < 5 ),
11  deptno VARCHAR2(6)
12     CONSTRAINT emp3_deptno_fk REFERENCES dept2(dcode)
13 );
```

8. 제약조건을 배웁니다

- 약식으로 생성하기

```
SCOTT> CREATE TABLE emp4
2 ( no NUMBER(4) PRIMARY KEY ,
3   name VARCHAR2(10) NOT NULL,
4   jumin VARCHAR2(13) NOT NULL UNIQUE ,
5   area NUMBER(1) CHECK ( area < 5 ),
6   deptno VARCHAR2(6) REFERENCES dept2(dcode)
7 );
```

8. 제약조건을 배웁니다

2) 테이블 생성 후 추가하기

```
SCOTT>ALTER TABLE emp4  
2 ADD CONSTRAINT emp4_name_uk UNIQUE(name) ;
```

```
SCOTT>ALTER TABLE emp4  
2 ADD CONSTRAINT emp4_area_nn NOT NULL(area) ;  
ADD CONSTRAINT emp4_area_nn NOT NULL(area)  
*
```

ERROR at line 2:
ORA-00904: : invalid identifier

NULL <-> NOT NULL 변경방법

```
SCOTT>ALTER TABLE emp4  
2 MODIFY (area constraint emp4_area_nn NOT NULL) ;
```

Table altered.

8. 제약조건을 배웁니다

- FOREIGN KEY 추가하기

-emp4 테이블의 no 컬럼이 emp2 테이블의 empno 컬럼의 값을 참조하도록 참조키 제약조건을 설정하세요.(emp4 테이블이 자식테이블입니다)

```
SCOTT> ALTER TABLE emp4  
2 ADD CONSTRAINT emp4_no_fk FOREIGN KEY(no)  
3 REFERENCES emp2(empno) ;
```


8. 제약조건을 배웁니다

- FOREIGN KEY 추가 시 주의사항

```
SCOTT>ALTER TABLE emp4
  2 ADD CONSTRAINT emp4_name_fk FOREIGN KEY(name)
  3 REFERENCES emp2(name) ;
REFERENCES emp2(name)
      *
```

ERROR at line 3:

ORA-02270: no matching unique or primary key for this column-list

```
SCOTT>ALTER TABLE emp2
  2 ADD CONSTRAINT emp2_name_uk UNIQUE(name);
```

Table altered.

부모테이블 컬럼에 Unique 나 Pk 컬럼이어야 함

```
SCOTT>ALTER TABLE emp4
  2 ADD CONSTRAINT emp4_name_fk FOREIGN KEY(name)
  3 REFERENCES emp2(name) ;
```

Table altered.

8. 제약조건을 배웁니다

- ON DELETE CASCADE

FOREIGN KEY 를 설정 후 부모 테이블 의 데이터를 지우고 싶는데 만약 자식 테이블에서 부모테이블의 해당 데이터를 참조하고 있을 경우 지울 수가 없습니다. 이럴 경우를 대비해서 FOREIGN KEY를 생성할 때 설정함.
이 옵션을 주면 부모테이블의 데이터가 지워지면 자식 테이블의 데이터도 함께 지우라는 의미임.

-ON DELETE SET NULL

이 옵션은 부모테이블의 데이터가 지워질 경우 자식테이블의 값을 NULL 로 설정하라는 의미임.

8. 제약조건을 배웁니다

3. 제약조건 관리하기

1) 제약 조건 DISABLE 하기 (NOVALIDATE)

```
SCOTT>INSERT INTO test_novvalidate VALUES(1,'DDD');
INSERT INTO test_novvalidate VALUES(1,'DDD')
```

*

ERROR at line 1:

ORA-00001: unique constraint (SCOTT.SYS_C0014418) violated

Primary Key 가 설정되어 있는 컬럼이므로 중복된 데이터가 입력이 안됩니다.

```
SCOTT>ALTER TABLE test_novvalidate
```

```
2  DISABLE NOVALIDATE CONSTRAINT SYS_C0014418 ;
```

Table altered.

```
SCOTT>INSERT INTO test_novvalidate VALUES(1,'DDD');
```

1 row created. <- 정상적으로 입력됩니다.

DISABLE
NOVALIDATE

8. 제약조건을 배웁니다

- Disable VALIDATE

```
SCOTT>INSERT INTO test_validate VALUES(4,NULL);
INSERT INTO test_validate VALUES(4,NULL)
```

*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("SCOTT"."TEST_VALIDATE"."NAME")

NAME 컬럼에 NOT NULL 제약 조건이 설정 되어 있어서 null 값을 허용하지 않습니다. 이 NOT NULL 제약조건을 DISABLE VALIDATE 한 후 다시 입력해 보겠습니다.

```
SCOTT>ALTER TABLE test_validate
2  DISABLE VALIDATE CONSTRAINT tv_name_nn ;
```

Table altered.

8. 제약조건을 배웁니다

- Disable VALIDATE

```
SCOTT>INSERT INTO test_validate VALUES(4,NULL);  
INSERT INTO test_validate VALUES(4,NULL)
```

*

ERROR at line 1:

ORA-25128: No insert/update/delete on table with constraint
(SCOTT.TV_NAME_NN)
disabled and validated

여전히 입력이 안됩니다. 에러 내역을 보면 해당 컬럼은
Insert/update/delete 작업을 수행할 수 없다는 내용을 보여줍니다. 즉 이
옵션은 해당 컬럼의 데이터를 변경할 수 없게 하는 옵션입니다.

8. 제약조건을 배웁니다

(3) DISABLE VALIDATE 옵션 설정 후 다른 칼럼 내용 변경하기

```
SCOTT>INSERT INTO test_validate VALUES(4,'DDD');  
INSERT INTO test_validate VALUES(4,'DDD')
```

```
*  
ERROR at line 1:  
ORA-25128: No insert/update/delete on table with constraint (SCOTT.TV_NAME_NN)  
disabled and validated
```

-- NULL 값 입력 하기

```
SCOTT>INSERT INTO test_validate VALUES(4,NULL);  
INSERT INTO test_validate VALUES(4,NULL)
```

```
*  
ERROR at line 1:  
ORA-25128: No insert/update/delete on table with constraint (SCOTT.TV_NAME_NN)  
disabled and validated
```

8. 제약조건을 배웁니다

-- 다른 컬럼에만 데이터 입력하기

```
SCOTT>INSERT INTO test_validate(no)
```

```
  2  VALUES(4) ;
```

```
INSERT INTO test_validate(no)
```

```
*
```

```
ERROR at line 1:
```

```
ORA-25128: No insert/update/delete on table with constraint
```

```
(SCOTT.TV_NAME_NN)
```

```
disabled and validated
```

위 테스트로 DISABLE VALIDATE 옵션은 테이블의 변경을 막는 것임을 알 수 있습니다.

8. 제약조건을 배웁니다

2) 제약 조건 ENABLE 하기

- **ENABLE NOVALIDATE** : 기존 데이터 검사 안하고 신규 데이터만 검사함

```
SCOTT>INSERT INTO test_enable VALUES(1,'AAA');
1 row created.
```

```
SCOTT>INSERT INTO test_enable VALUES(2,'BBB');
1 row created.
```

```
SCOTT>INSERT INTO test_enable VALUES(3, NULL);
INSERT INTO test_enable VALUES(3, NULL)
*
```

ERROR at line 1:

ORA-01400: cannot insert NULL into ("SCOTT"."TEST_ENABLE"."NAME")

NOT NULL 제약조건 때문에 3 번 값이 입력이 안되고 있습니다.

8. 제약조건을 배웁니다

```
SCOTT>ALTER TABLE test_enable  
2 DISABLE CONSTRAINT te_name_nn ;
```

Table altered.

```
SCOTT>INSERT INTO test_enable VALUES(3,NULL);
```

1 row created. ← 잘 입력됩니다. 그러나 이 데이터는 잘 못 된 데이터입니다.

현재 이 테이블에 3번 행은 NULL 값이 들어갈 수 가 없으나 제약조건을 DISABLE 해서 틀린 데이터가 들어 간 것입니다. 이런 상황에서 제약조건을 ENABLE 하면서 테스트를 진행합니다.

8. 제약조건을 배웁니다

(1) ENABLE NOVALIDATE 로 name 컬럼 제약조건 ENABLE 하기

```
SCOTT>ALTER TABLE test_enable
2  ENABLE NOVALIDATE CONSTRAINT te_name_nn ;
```

Table altered.

```
SCOTT>SELECT * FROM test_enable ;
```

NO	NAME
1	AAA
2	BBB
3	

<- 이 행이 잘못된 데이터입니다

```
SCOTT>INSERT INTO test_enable VALUES(4,NULL);
INSERT INTO test_enable VALUES(4,NULL)
*
```

ERROR at line 1:
ORA-01400: cannot insert NULL into ("SCOTT"."TEST_ENABLE"."NAME")

ENABLE NOVALIDATE 옵션은
기존 데이터는 확인 안하고
신규로 입력되는 데이터만
확인하는 옵션입니다.

8. 제약조건을 배웁니다

(2) ENABLE VALIDATE 로 name 컬럼 제약조건 ENABLE 하기

```
SCOTT>ALTER TABLE test_enable
2 DISABLE CONSTRAINT te_name_nn ;
```

Table altered.

```
SCOTT>ALTER TABLE test_enable
2 ENABLE VALIDATE CONSTRAINT te_name_nn ;
ENABLE VALIDATE CONSTRAINT te_name_nn
```

*

ERROR at line 2:

ORA-02293: cannot validate (SCOTT.TE_NAME_NN) - CHECK constraint violated

ENABLE VALIDATE 옵션은 기존 데이터도 체크하고 신규 데이터도 체크함.
현재 기존 데이터에 문제가 있어서
ENABLE 이 안됨.

8. 제약조건을 배웁니다

3) EXCEPTIONS 테이블을 사용하여 ENABLE VALIDATE 하기

실습은 교재 295 – 297 페이지를 참고하세요

8. 제약조건을 배웁니다

4) 제약조건 조회하기

```
SCOTT>SELECT owner, constraint_name, constraint_type, status  
2 FROM user_constraints  
3 WHERE table_name='EMP4';
```

위 칼럼 중에 constraint_type 의 타입 값 중 P: Primary Key ,
U :Unique , C: CHECK , R:외래키 를 의미합니다.

8. 제약조건을 배웁니다

- 사용 예 2: FOREIGN KEY 조회하기

```
SCOTT> SELECT a.table_name "Child_Table",
2         c.column_name "Child_Column",
3         a.constraint_name "Child_Cons_name",
4         b.table_name "Parent_Table" ,
5         a.r_constraint_name "Parent_Cons_name" ,
6         d.column_name "Parent_Column"
7 FROM user_constraints a , user_constraints b , user_cons_columns c,
8      (SELECT constraint_name, column_name, table_name
9       FROM user_cons_columns) d
10 WHERE a.r_constraint_name=b.constraint_name
11 AND   a.constraint_name=c.constraint_name
12 AND   a.r_constraint_name=d.constraint_name
13 AND   a.constraint_type='R' ;
```

8. 제약조건을 배웁니다

5) 제약조건 삭제하기

```
SCOTT>ALTER TABLE emp4  
2 DROP CONSTRAINT emp4_name_fk ;
```