



## 06. Constraint

# ● 1. 데이터 무결성

---

## ❑ 무결성

무효갱신으로 부터 테이블내의 데이터를 위/변조,훼손하지 못하도록 제약(Constraint)하여 데이터의 유효성, 일관성, 완전성,정확성,정밀성을 보장하는 성질

## ❑ 무결성 제약사항 구현 방법 3가지

- ① 선언적 무결성 제약사항(Declarative Integrity Constraint)
- ② Trigger (PL/SQL)
- ③ Application Logic (Coding)

## ❑ 선언적 무결성 제약사항

선언적이란 프로그램 코딩을 하지 않고 테이블 생성시 또는 생성후 컬럼이나 테이블에 무결성 제약사항을 표기하여 정의(define)하는 방식

- ① **PRIMARY KEY** : 대표성,고유성,존재성(Not NULL)을 보장하며 **테이블당 1개만 정의**,  
**Unique Index 자동 생성** , Unique Key 와 NOT NULL의 결합형태
- ② **UNIQUE KEY** : 데이터의 **고유성 보장**, 테이블에 **N개의 정의 가능**,  
**NULL 허용,Unique Index 자동 생성**
- ③ CHECK : 값의 범위 나 조건을 지정,**Boolean 연산**
- ④ NOT NULL : NULL(결측치)을 허용하지 않는 **필수입력사항정의** ⑤  
FOREIGN KEY : 테이블 **개체간의참조관계** 정의, **내용에의한참조**

## ❑ LEVEL 과 생성시기

- LEVEL ① TABLE : 테이블에 정의하는 제약 사항으로 여러 컬럼이 제약사항에 관여하는 경우 사용
- ② COLUMN : 특정 컬럼에 정의하는 제약 사항

생성시기 ① TABLE 생성시 생성 ② TABLE 생성후 임의의 시점에 추가

## ● 2. NOT NULL

### ❏ NOT NULL

NOT NULL은 NULL을 허용하지 않는 즉 데이터가 필수적으로 존재하도록 하는 제약 사항

#### ① 테스트 대상 테이블 생성

```
CREATE TABLE CUSTOMER(  
    ID          VARCHAR2(8)          NOT NULL,  
    PWD         VARCHAR2(8)          CONSTRAINT CUSTOMER_PWD_NN NOT NULL,  
    NAME        VARCHAR2(20),        -- 이름  
    SEX         CHAR(1),              -- 성별      [M|F] M:Male   F: Female  
    AGE         NUMBER(3) -- 나이  
);  
DESC CUSTOMER
```

#### ② INSERT INTO CUSTOMER(ID,PWD,NAME,SEX, AGE) VALUES('xman','ok','kang', 'M',21);

```
INSERT INTO CUSTOMER(ID,PWD,NAME,SEX,AGE) VALUES('XMAN','no','kim', 'T',-20);
```

- 정상적 입력

#### ③ INSERT INTO CUSTOMER(ID,NAME,AGE) VALUES('zman','son',99);

```
INSERT INTO CUSTOMER(ID,PWD,NAME,AGE) VALUES('rman',NULL,'jjang',24);
```

```
INSERT INTO CUSTOMER(ID,PWD,NAME,AGE) VALUES(' ', 'pwd', 'jjang',24);
```

- PWD,SEX 컬럼에 암시적/명시적 NULL을 삽입하려 하지만 PWD컬럼의 NOT NULL 제약사항으로 INSERT는 에러 발생.

- DML 연산시 데이터를 변경함에 따라 부적절한 데이터가 발생할수 있다.

DML 연산으로 데이터를 변경 하기전에 무결성 제약사항을 위반하는지 여부를 점검한후  
무결성 제약사항을 위반하게 되면 해당 DML 연산을 수행하지 않고 취소 처리.

## ● 2. NOT NULL

---

### ❏ NOT NULL

NOT NULL은 NULL을 허용하지 않는 즉 데이터가 필수적으로 존재하도록 하는 제약 사항

- ④ UPDATE CUSTOMER SET PWD = NULL WHERE ID = 'XMAN'; -- ID가 XMAN인 ROW만 수정  
SELECT \* FROM CUSTOMER;

- UPDATE연산은 PWD의 NOT NULL 제약사항으로 에러가 발생

- ⑤ 데이터딕셔너리(시스템 카타로그)에서 제약사항 확인하기

```
SELECT TABLE_NAME,CONSTRAINT_NAME,CONSTRAINT_TYPE,SEARCH_CONDITION  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME = 'CUSTOMER';
```

```
SELECT TABLE_NAME,CONSTRAINT_NAME,POSITION,COLUMN_NAME  
FROM USER_CONS_COLUMNS  
WHERE TABLE_NAME = 'CUSTOMER' ORDER BY CONSTRAINT_NAME,POSITION;
```

## ● 3. CHECK

### □ CHECK

\* 설정된 조건을 체크하여 Boolean 결과가 TRUE인 경우에만 DML연산 허용.

① 테이블 생성후 제약사항 추가

```
ALTER TABLE CUSTOMER ADD CONSTRAINT CUSTOMER_SEX_CK CHECK (SEX IN ('M','F'));
```

- SQL 오류: ORA-02293: (SCOTT.CUSTOMER\_SEX\_CK)을 검증할 수 없습니다
- 잘못된 제약사항 확인

② UPDATE CUSTOMER SET SEX='M' WHERE SEX='T';  
COMMIT;

```
ALTER TABLE CUSTOMER ADD CONSTRAINT CUSTOMER_SEX_CK CHECK (SEX IN ('M','F'));
```

- 제약사항 추가시 기존의 데이터에 대한 무결성 검사.
- 기존 데이터가 추가되는 제약사항을 위반하여 에러가 발생하여 데이터 처리후 제약사항을 추가한다.

③ INSERT INTO CUSTOMER(ID,PWD,NAME,SEX, AGE) VALUES('xman','ok','kang', 'M',21);  
INSERT INTO CUSTOMER(ID,PWD,NAME,SEX,AGE) VALUES('xman','ok', 'jjang','M',20);

- ID가 중복되지만 입력

④ INSERT INTO CUSTOMER(ID,PWD,NAME,AGE) VALUES('asura','ok', 'joo',99);

- 성별(SEX) 컬럼에 NOT NULL 제약사항이 없기 때문에 NULL 입력.

⑤ INSERT INTO CUSTOMER(ID,PWD,NAME,SEX,AGE) VALUES('harisu','ok', 'susu','T',33);

- 성별(SEX) 컬럼에 Check 제약사항으로 M 과 F만 입력 가능 에러발생

⑥ INSERT INTO CUSTOMER(ID,PWD,NAME,SEX,AGE) VALUES('shinsun','ok', '도사', 'M',999);

- AGE컬럼은 NUMBER(3)으로 정의되어 999가 입력된다.

## ● 3. CHECK

### ☐ CHECK

#### ⑦ UPDATE CUSTOMER SET AGE = AGE + 1;

- 1년이 지나 고객의 나이를 +1 하려고 UPDATE 연산을 수행하는데 회원 shisun은 1000으로 자리수를 초과하는 에러가 발생하여 문장 수준 롤백(Statement Level Rollback) 발생.

ID	AGE
-----	-----
xman	21
xman	20
asura	99

## ● 4. Unique Key

### ❑ Unique Key

\* 컬럼 데이터의 고유성을 보장하고 NULL을 허용하여 데이터가 없을수 있지만 데이터가 존재하는 경우 고유해야 한다.

#### ① CUSTOMER 테이블 삭제후 생성

ID는 필수 입력사항이면서 고유성을 가져야 해서 NOT NULL + UNIQUE 제약사항 정의  
MOBILE(핸드폰)은 핸드폰이 없는 고객도 있지만 핸드폰이 있는경우 고유한 번호를 가져야 하기 때문에 UNIQUE 제약사항 정의

```
DROP TABLE CUSTOMER;
```

```
CREATE TABLE CUSTOMER(  
    ID          VARCHAR2(8)          NOT NULL CONSTRAINT CUSTOMER_ID_UK UNIQUE,  
    PWD         VARCHAR2(8)          NOT NULL,  
    NAME        VARCHAR2(20),  
    SEX         CHAR(1) DEFAULT 'M'  
                CONSTRAINT CUSTOMER_SEX_CK CHECK (SEX IN ('M','F')),  
    MOBILE      VARCHAR2(14)         UNIQUE,  
    AGE         NUMBER(3)            DEFAULT 18  
);
```

#### ② INSERT INTO CUSTOMER(ID,PWD,NAME,MOBILE, AGE) VALUES('xman','ok','kang', '011-3333',21);

-성별 컬럼에 암시적으로 NULL이 지정되지만 DEFAULT에 의해서 'M' 값이 저장된다.

#### ③ INSERT INTO CUSTOMER(ID,PWD,NAME, MOBILE,AGE) VALUES('XMAN','yes','kim','011-3334',33);

- 데이터는 대소문자를 구분하기 때문에 xman 과 XMAN은 다른 데이터

## ● 4. Unique Key

④ INSERT INTO CUSTOMER(ID,PWD,NAME, MOBILE,AGE) VALUES('xman','yes','lee', '011-3335',-21);  
- ID중복으로 Unique 제약 사항 위반 에러 발생 INSERT 실패(회원가입 실패)

⑤ INSERT INTO CUSTOMER(ID,PWD,NAME, MOBILE,AGE) VALUES('yman','yes','lee', '011-3333',28);  
- 핸드폰 번호 중복으로 Unique 제약 사항 위반 에러 발생 INSERT 실패(회원가입 실패)

⑥ INSERT INTO CUSTOMER(ID,PWD,NAME, MOBILE) VALUES('무명인','yes',NULL, NULL);  
- ID컬럼의 NOT NULL 제약사항으로 NULL을 허용하지 않는다.

⑤ ALTER TABLE CUSTOMER ADD CONSTRAINT CUSTOMER\_NAME\_SEX\_UK UNIQUE(NAME,SEX);  
ALTER TABLE CUSTOMER MODIFY(NAME NOT NULL);  
- 테이블 생성후 제약사항 신규 추가  
- 2개 컬럼을 조합하여 Unique제약 사항을 생성

⑥ INSERT INTO CUSTOMER(ID,PWD,NAME, SEX) VALUES('rman','yes','syo', 'M');  
INSERT INTO CUSTOMER(ID,PWD,NAME, SEX ) VALUES('Rman','yes','syo', 'F');  
INSERT INTO CUSTOMER(ID,PWD,NAME, SEX) VALUES('RmaN','yes','syo', 'M');  
SELECT \* FROM CUSTOMER;  
- NAME+SEX 조합의 Unique 제약사항

⑦ INDEX 생성 여부 확인  
SELECT INDEX\_NAME,INDEX\_TYPE,UNIQUENESS FROM USER\_INDEXES  
WHERE TABLE\_NAME = 'CUSTOMER';

SELECT INDEX\_NAME,COLUMN\_POSITION,COLUMN\_NAME FROM USER\_IND\_COLUMNS  
WHERE TABLE\_NAME = 'CUSTOMER' ORDER BY INDEX\_NAME,COLUMN\_POSITION;

[참고] UNIQUE 제약사항 정의시 해당 컬럼에 INDEX 가 자동 생성된다. INDEX는 Data에 대한 빠른 접근(Quick Search)를 할수 있도록 하는 데이터베이스 Object로 Unique 제약사항 정의시 Index 자동 생성되는 이유는 데이터 입력,수정시 중복 여부를 빠르게 확인 하기 위해서 생성.



## ● 5. Primary Key

### ❑ Primary Key

- \* PRIMARY는 주요한, 기본적인 이라는 사전적 의미를 가지며 데이터베이스에서 PRIMARY KEY는 주키 또는 기본키 라고 한다. 기본키는 테이블내의 각 레코드(행)을 고유하게 식별하는 값을 가진 컬럼(들)의 조합으로 컬럼 데이터의 고유성(Uniqueness) 과 존재성(Not Null)을 동시에 보장하는 제약사항.
- \* 테이블내에 레코드(행)을 고유하게 식별 할수 있는 식별자 후보가 여러 개 있는 경우 대표성을 가진 후보가 PRIMARY KEY(기본키)가 되고 나머지 후보자들은 UNIQUE KEY가 된다.  
기본키는 테이블에 1개만 정의할수 있고 UNIQUE KEY는 테이블에 N개 정의 가능.

- ① ID는 필수 입력사항이면서 고유성을 가져야 해서 PRIMARY KEY로 제약사항 정의  
MOBILE(핸드폰)은 핸드폰이 없는 고객도 있지만 핸드폰이 있는경우 고유한 번호를 가져야 하기 때문에 UNIQUE KEY제약사항 정의.

```
DROP TABLE CUSTOMER;
```

```
CREATE TABLE CUSTOMER(  
    ID          VARCHAR2(8)          CONSTRAINT CUSTOMER_ID_PK PRIMARY KEY,  
    PWD         VARCHAR2(8)          NOT NULL,  
    NAME        VARCHAR2(20),  
    SEX         CHAR(1)  DEFAULT 'M'  
                                CONSTRAINT CUSTOMER_SEX_CK CHECK (SEX IN ('M','F')),  
    MOBILE      VARCHAR2(14)         CONSTRAINT CUSTOMER_MOBILE_UK UNIQUE,  
    AGE         NUMBER(3)            DEFAULT 18  
);
```

## ● 5. Primary Key

---

### ❑ Primary Key

- ② 성별(SEX),나이(AGE) 컬럼에 정의된 DEFAULT는 입력시 컬럼의 값이 지정되지 않는 암시적인 NULL이 지정되면 NULL 대신 저장하는 값으로 성별에는 'M', 나이는 18 입력

```
INSERT INTO CUSTOMER(ID,PWD,NAME,MOBILE) VALUES('zman','ok','한국','011');
```

- ③ MOBILE 컬럼에는 UNIQUE KEY 제약사항이 정의되었지만 NULL 허용.

```
INSERT INTO CUSTOMER(ID,PWD,NAME) VALUES('xman','ok','king');
```

- ④ ID중복으로 에러가 발생한다.

SQL 오류: ORA-00001: 무결성 제약 조건(SCOTT.CUSTOMER\_ID\_PK)에 위배됩니다

0001. 00000 - "unique constraint (%s.%s) violated"

```
INSERT INTO CUSTOMER(ID,PWD,NAME) VALUES('xman','power','zzang');
```

- ⑤ 데이터는 대소문자를 구분한다.

'xman' 와 'Xman'은 다른 데이터로 중복 에러가 발생하지 않는다.

```
INSERT INTO CUSTOMER(ID,PWD,NAME) VALUES('Xman','korea','dbzzang');
```

- ⑥ VALUES절에 함수 사용 가능하다, ID 중복으로 무결성 위반 에러 발생

```
INSERT INTO CUSTOMER(ID,PWD,NAME) VALUES(lower('xMan'),'ok','zzang');
```

## ● 5. Primary Key

---

### ❑ Primary Key

- ⑦ INSERT시 컬럼 생략시 해당 컬럼에 암시적 NULL이 지정된다.

PRIMARY KEY는 NULL을 허용하지 않는다.

SQL 오류: ORA-01400: NULL을 ("SCOTT"."CUSTOMER"."ID") 안에 삽입할 수 없습니다

```
INSERT INTO CUSTOMER(PWD,NAME) VALUES('ok','kim');
```

- ⑧ PRIMARY KEY의 유일성(UNIQUENESS) 과 존재성(NOT NULL)을 위반 하는 경우  
UPDATE 명령이 수행되지 않는다.

```
UPDATE CUSTOMER SET ID = NULL;                                -- 존재성(NOT NULL)
UPDATE CUSTOMER SET ID = 'XMAN';                               -- 유일성(UNIQUENESS)
```

- ⑨ USER\_CONSTRAINTS는 사용자 소유(Owner)의 모든 제약사항(CONSTRAINT)를 조회 할수 있다  
USER\_CONS\_COLUMNS는 제약사항에 관련된 컬럼(COLUMN)의 정보를 조회할수 있다.  
USER\_INDEXES는 사용자 소유(Owner)의 모든 인덱스(INDEX)를 조회 할수 있다  
USER\_IND\_COLUMNS는 인덱스에 관련된 컬럼(COLUMN)의 정보를 조회할수 있다.  
개발이나 튜닝시 빈번하게 사용하게됨으로 실습을 통해 익혀두어야 한다.

## ● 5. Primary Key

---

### ❑ Primary Key

```
SELECT TABLE_NAME,CONSTRAINT_NAME,CONSTRAINT_TYPE,SEARCH_CONDITION FROM USER_CONSTRAINTS  
WHERE TABLE_NAME = 'CUSTOMER';
```

```
SELECT TABLE_NAME,CONSTRAINT_NAME,POSITION,COLUMN_NAME FROM USER_CONS_COLUMNS  
WHERE TABLE_NAME = 'CUSTOMER' ORDER BY CONSTRAINT_NAME,POSITION;
```

```
SELECT INDEX_NAME,INDEX_TYPE,UNIQUENESS  
FROM USER_INDEXES  
WHERE TABLE_NAME = 'CUSTOMER';
```

```
SELECT INDEX_NAME,COLUMN_POSITION,COLUMN_NAME  
FROM USER_IND_COLUMNS  
WHERE TABLE_NAME = 'CUSTOMER' ORDER BY INDEX_NAME,COLUMN_POSITION;
```

## ● 6. Foreign Key

### ❑ Foreign Key

- \* 연관성 있는 **테이블간(내) 참조 무결성(Referential Integrity)** 보장.
- \* 참조 무결성이란 테이블 사이의 관계에서 발생한다. 이전 예제를 보면 부서 테이블과 사원 테이블은 부서번호라는 공통의 데이터 속성을 가지고 관계 형성.
- \* 부서번호 속성(**P.K or U.K**)은 원래 부서 개체(테이블)에서 정의 된후 사원 개체(테이블)와 관계를 형성하기 위해 사원 개체(테이블)에게 상속된(비식별관계) 속성.
- \* 각 테이블에 존재하는 공통된 데이터 속성간의 데이터 무결성을 보장하는 것이 참조 무결성 제약 사항.

```
① CREATE TABLE 부서(부서번호    VARCHAR2(2)  CONSTRAINT 부서_부서번호_PK  PRIMARY KEY,
                        부서명      VARCHAR2(10)  CONSTRAINT 부서_부서명_NN   NOT NULL
);

CREATE TABLE 사원(사번          VARCHAR2(8)   PRIMARY KEY ,
                  이름           VARCHAR2(10),
                  부서번호       VARCHAR2(2) ,
                  CONSTRAINT 사원_부서_부서번호_FK FOREIGN KEY(부서번호)
                        REFERENCES 부서(부서번호) [ON DELETE CASCADE | SET NULL]
);

② DESCRIBE 부서                                // not null , 축약
desc      사원

SELECT CONSTRAINT_NAME,CONSTRAINT_TYPE,TABLE_NAME,SEARCH_CONDITION,R_CONSTRAINT_NAME,DELETE_RULE,STATUS
FROM   USER_CONSTRAINTS
WHERE  TABLE_NAME IN ('부서','사원');          // CONSTRAINT_NAME, CONSTRAINT_TYPE, R_CONSTRAINT_NAME,DELETE_RULE  관찰
```

## ● 6. Foreign Key

---

③ 부서 데이터가 존재 하지 않는 상황에서 사원 정보 입력시 참조 무결성 에러 발생

```
INSERT INTO 사원(사번,이름,부서번호) VALUES('XMAN', 'TUNER','10');
```

- SQL 오류: ORA-02291: 무결성 제약조건(SCOTT.사원\_부서\_부서번호\_FK)이 위배되었습니다. **부모 키가 없습니다**

```
INSERT INTO 부서(부서번호,부서명) VALUES('10','관리');
```

// 부서(Parent data) 정보 입력

```
INSERT INTO 부서(부서번호,부서명) VALUES('20','전산');
```

```
INSERT INTO 부서(부서번호,부서명) VALUES('50','영업');
```

④ 10번,20번 부서에 근무하는 사원

// 사원(Child data) 정보 입력

```
INSERT INTO 사원(사번,이름,부서번호) VALUES('XMAN','TUNER',10);
```

```
INSERT INTO 사원(사번,이름,부서번호) VALUES('YMAN','DBA',20);
```

⑤ 30번 부서(존재하지 않는 부서)에 근무하는 사원 입력시 참조 무결성 에러 발생

```
INSERT INTO 사원(사번,이름,부서번호) VALUES('ZMAN', 'DEVELOPER',30);
```

- SQL 오류: ORA-02291: 무결성 제약조건(SCOTT.사원\_부서\_부서번호\_FK)이 위배되었습니다. **부모 키가 없습니다**

⑥ 근무자가 (Child data) 없는 50번부서 폐지

```
DELETE FROM 부서 WHERE 부서번호 = 50;
```

⑦ 근무자가 있는 10번부서 폐지시 참조 무결성으로 에러 발생. // on delete a) restrict b) cascade c) set null

```
DELETE FROM 부서 WHERE 부서번호 = 10;
```

- SQL 오류: ORA-02292: 무결성 제약조건(SCOTT.사원\_부서\_부서번호\_FK)이 위배되었습니다. **자식 레코드가 발견 되었습니다**

```
SELECT * FROM 사원;
```

// 데이터관찰

## ● 6. Foreign Key

---

⑧ 근무자가 있는 10번부서 폐지 방법은?

① on delete restrict → 해당 부서 근무자를 퇴사(DELETE)처리 하거나 다른 부서로 이관(UPDATE)하고 폐지한다.

UPDATE 사원 SET 부서번호=20 WHERE 부서번호=10;

DELETE FROM 부서 WHERE 부서번호 = 10;

② on delete cascade

DROP TABLE 사원;

CREATE TABLE 사원 ~ on delete cascade 생성후 ② ~ ⑦ 재 실행 , ③ 생략

③ set null

DROP TABLE 사원;

CREATE TABLE 사원 ~ on delete set null 생성후 ② ~ ⑦ 재 실행 , ③ 생략

### □ 과제

1) ON UPDATE CASCADE

- Oracle DBMS는 지원하지 않으나 PL/SQL로 구현하는 방법 설명
- ON UPDATE CASCADE를 지원하는 DBMS를 찾아 해당 SQL 구문 설명

2) Parent Table(ex 부서)을 drop 하는 SQL 작성