

# 10강 데이터 무결성과 제약조건

## 데이터 무결성이란

- 무결성은 데이터의 정확성, 일관성, 유효성이 유지되는 것을 말하며 데이터베이스 관리 시스템의 중요한 기능이다.
- 개체 무결성 : 모든 테이블은 기본키로 선택된 필드를 가져야 한다.
- 참조 무결성 : 참조관계에 해당하는 두 테이블의 데이터를 항상 일관된 값을 가져야 한다.
- 도메인 무결성 : 테이블에 존재하는 필드의 무결성을 보장하기 위한 것 NOY NULL, 필드 타입등이 있다
- 무결성 규칙 : 도메인 무결성보다 넓은 개념의 무결성이며 모든 데이터베이스 전체에 공통으로 적용되는 규칙

## 제약조건

- 데이터 무결성 제약조건이란 테이블에 적절하지 않는 값이 입력되는 것을 방지하기 위해 테이블을 생성할 때 각 컬럼에 정의하는 여러가지 규칙을 말한다.
- 오라클에서 제공하는 무결성 제약조건은 다음과 같다

무결성 제약 조건	역할
NOT NULL	컬럼에 NULL 값을 포함하지 못하도록 지정한다.
UNIQUE	테이블의 모든 로우에 대해서 유일한 값을 갖도록 한다.
PRIMARY KEY	테이블의 각 레코드를 식별하기 위해 사용되는 키로 NULL과 중복된 값을 모두 허용하지 않는다 (NOT NULL + UNIQUE)
FOREIGN KEY	참조되는 테이블에 컬럼의 값이 항상 존재해야 한다.
CHECK	저장 가능한 데이터 값의 범위나 조건을 지정하여 설정한 값 만을 허용한다.

## 제약조건 - NOT NULL

- NOT NULL 제약조건을 사용하면 해당 칼럼은 NULL값을 가질 수 없게 된다.
- 반드시 값을 입력해야 하는 칼럼에 지정한다.
- 예제: 아래 조건에 해당하는 테이블(customer)을 만들고 데이터를 입력해본다.

칼럼명	타입	크기	제약조건
id	varchar2	20	not null
pwd	varchar2	20	not null
name	varchar2	20	not null

```
CREATE TABLE customer(  
    id varchar2(20) not null,  
    pwd varchar2(20) not null,  
    name varchar2(20) not null  
);
```

id	pwd	name
asdf	1234	null

오류 발생

```
INSERT INTO customer  
VALUES('asdf','1234',null);
```

## 제약조건 - UNIQUE

- UNIQUE 제약조건을 사용하면 해당 칼럼은 고유한 값을 가지게 할수 있다 - 중복데이터 입력시 오류 발생
- 단 NULL값을 허용한다.
- 예제: 아래 조건에 해당하는 테이블(customer)을 만들고 데이터를 입력해본다. (이미 만들어진 테이블은 삭제한다.)

칼럼명	타입	크기	제약조건
id	varchar2	20	UNIQUE
pwd	varchar2	20	not null
name	varchar2	20	Not null

```
CREATE TABLE customer(  
    id varchar2(20) unique,  
    pwd varchar2(20) not null,  
    name varchar2(20) not null  
);
```

오류 발생

id	pwd	name
asdf	1234	'park'
asdf	6789	'kim'

```
INSERT INTO customer  
VALUES('asdf','1234','park');  
INSERT INTO customer  
VALUES('asdf','6789','kim');
```

## 제약조건 - UNIQUE

- UNIQUE 제약조건은 이름을 명시하지 않으면 자동으로 제약조건의 이름을 부여한다.
- 테이블 생성시 직접 제약조건의 이름을 부여 할 수 있다.
  - 예제: 아래 조건에 해당하는 테이블(customer)을 만들고 제약조건을 확인해본다. (이미 만들어진 테이블은 삭제한다.)

칼럼명	타입	크기	제약조건	제약조건 이름
id	varchar2	20	UNIQUE	customer_id_uk
pwd	varchar2	20	not null	customer_pwd_nn
name	varchar2	20	Not null	customer_name_nn

```
CREATE TABLE customer(  
  id varchar2(20) CONSTRAINT customer_id_uk unique,  
  pwd varchar2(20) CONSTRAINT customer_pwd_nn not null,  
  name varchar2(20) CONSTRAINT customer_name_nn not null  
);
```

```
SELECT table_name, constraint_name  
FROM user_constraints  
WHERE table_name in('CUSTOMER');
```

## 제약조건 – PRIMARY KEY

- PRIMARY KEY 제약조건은 테이블에 기본 키를 생성합니다.
- 테이블에는 반드시 기본키가 존재해야 하며 기본키는 두가지 특성을 가진다. NULL값을 가질 수 없고, 고유해야 한다.  
(NOT NULL + UNIQUE)
- 기본키는 특정 레코드를 고유하게 구분(식별)할 수 있어야 한다.(개체 무결성)

## 제약조건 – PRIMARY KEY

- 예제 : 아래 조건에 해당하는 테이블(customer)을 만들고 제약조건을 확인해본다. (이미 만들어진 테이블은 삭제한다.)
- 그리고 데이터를 추가해서 제약조건을 기능을 살펴본다.

칼럼명	타입	크기	제약조건	제약조건 이름
id	varchar2	20	Primary key	customer_id_pk
pwd	varchar2	20	not null	customer_pwd_nn
name	varchar2	20	Not null	customer_name_nn

id	pwd	name
asdf	1234	'park'
asdf	6789	'kim'
null	1597	'hong'



## 제약조건 – PRIMARY KEY

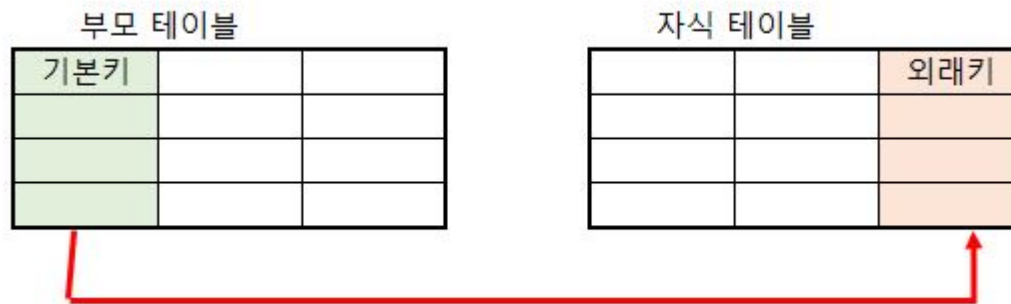
```
CREATE TABLE customer(  
    id varchar2(20),  
    pwd varchar2(20) CONSTRAINT customer_pwd_nn not null,  
    name varchar2(20) CONSTRAINT customer_name_nn not null,  
    CONSTRAINT customer_id_pk PRIMARY KEY(id)  
);
```

```
INSERT INTO customer  
VALUES('asdf','1234','park');  
INSERT INTO customer  
VALUES('asdf','6789','kim');  
INSERT INTO customer  
VALUES(null,'1597','hong');
```

```
SELECT table_name, constraint_name  
FROM user_constraints  
WHERE table_name in('CUSTOMER');
```

## 제약조건 – FOREIGN KEY

- FOREIGN KEY : 참조 무결성을 위한 외래키 제약조건을 생성한다.
  - 외래키 제약 조건을 생성하면 두 테이블 간의 관계가 형성된다.
  - 이때 외래키가 생성되는 테이블을 참조테이블 또는 자식테이블이라고 부른다.
  - 외래키가 가르키는 대상이 되는 테이블을 기준 테이블 또는 부모 테이블이라고 부른다.
  - 참조의 대상이 되는 키는 반드시 기본키나 유니크 제약조건이 있어야 한다.



## 제약조건 – FOREIGN KEY

- 자식 테이블의 외래키는 부모 테이블의 기본키를 참조하고 있다.
- 외래키 제약조건에 걸린 해당 컬럼에 데이터를 입력하기 위해서는 부모의 해당 데이터에 **존재** 해야한다.
- 부모에 존재하지 않는 데이터 입력시 오류를 발생 시킨다.

## 제약조건 – FOREIGN KEY

- 예제 : 아래 조건에 해당하는 테이블(Student, registration)두개를 만들어 본다

칼럼명	타입	크기	제약조건	제약조건 이름
stuno	varchar2	20	primary key	student_id_pk
name	varchar2	20	not null	student_pwd_nn
major	varchar2	20		

칼럼명	타입	크기	제약조건	제약조건 이름
enrollid	varchar2	20	primary key	registration_enrollid_pk
stuno	varchar2	20	foreign key	registration_stuno_fk
subject	varchar2	20	not null	registration_subject_nn

## 제약조건 – FOREIGN KEY

- 예제 : Student테이블에 데이터를 추가하고, registration에 데이터를 추가하면서 제약조건을 확인해보자

stuno	name	major
S001	Ko	'math'
S002	Hong	'physics'
S003	lisa	'computer science'

enrollid	stuno	subject
E001	S001	'linear algebra'
E002	S003	'algorithm'
E003	S005	'constitutional law'



오류 발생

## 제약조건 – FOREIGN KEY

```
CREATE TABLE student(  
    stuNo varchar2(20),  
    name varchar2(20) CONSTRAINT student_name_nn NOT NULL,  
    major varchar2(20),  
    CONSTRAINT student_stuno_pk PRIMARY KEY(stuNo)  
);
```

```
CREATE TABLE registration(  
    enrollid varchar2(20),  
    stuNo varchar2(20),  
    subject varchar2(20) CONSTRAINT registration_subject_nn NOT NULL,  
    CONSTRAINT registration_enrollid_pk PRIMARY KEY(enrollid),  
    CONSTRAINT registration_stuNo_fk FOREIGN KEY(stuNo) REFERENCES student(stuNo)  
);
```

## 제약조건 – FOREIGN KEY

```
INSERT INTO student
VALUES ('S001', 'KO', 'Math');
INSERT INTO student
VALUES ('S002', 'HONG', 'Physics');
INSERT INTO student
VALUES ('S003', 'Lisa Su', 'Computer Science');
```

```
INSERT INTO registration
VALUES ('E001', 'S001', 'Linear Algebra');
INSERT INTO registration
VALUES ('E002', 'S003', 'Algorithm');
INSERT INTO registration
VALUES ('E003', 'S005', 'Constitutional Law');
```



오류 발생

## 제약조건 – CHECK

- CHECK 제약조건은 컬럼에 저장 가능한 데이터의 범위나 조건을 정의한다. 해당 범위나 조건에 벗어나는 값이 들어오면 오류를 발생 시킨다.
- 예제 : 아래 조건에 해당하는 테이블(customer)을 만들고 데이터를 입력해 본다 (이미 만들어진 테이블은 삭제한다.)

컬럼명	타입	크기	제약조건	제약조건 이름
id	varchar2	20	Primary key	customer_id_uk
pwd	varchar2	20	not null	customer_pwd_nn
name	varchar2	20	not null	customer_name_nn
jumsu	number	3	check(0<=jumsu<=100)	customer_jumsu_range

id	Pwd	name	jumsu
asdf	1234	'park'	75
qwer	6789	'kim'	123
zxcv	1597	'hong'	-65



## 제약조건 – CHECK

```
CREATE TABLE customer(  
    id varchar2(20),  
    pwd varchar2(20) CONSTRAINT customer_pwd_nn not null,  
    name varchar2(20) CONSTRAINT customer_name_nn not null,  
    jumsu number(3) CONSTRAINT customer_jumsu_range check(0<=jumsu AND jumsu<=100),  
    CONSTRAINT customer_id_pk PRIMARY KEY(id)  
);
```

```
INSERT INTO customer  
VALUES('asdf','1234','park',75);  
INSERT INTO customer  
VALUES('qwer','6789','kim',123);  
INSERT INTO customer  
VALUES('zxcv','1597','hong',-65);
```

## 제약조건 – DEFAULT

- DEFAULT 제약조건은 아무런 값을 입력하지 않았을 때 디폴트 값으로 입력되게 만든다.
- 예제 : 아래 조건에 해당하는 테이블(customer)을 만들고 데이터를 입력해 본다 (이미 만들어진 테이블은 삭제한다.)

칼럼명	타입	크기	제약조건	제약조건 이름
id	varchar2	20	Primary key	customer_id_uk
pwd	varchar2	20	not null	customer_pwd_nn
name	varchar2	20	default ('홍길동')	

id	pwd	name
asdf	1234	'kim'
hong	6789	null

## 제약조건 – DEFAULT

```
CREATE TABLE customer(  
    id varchar2(20),  
    pwd varchar2(20) CONSTRAINT customer_pwd_nn not null,  
    name varchar2(20) default '홍길동',  
    CONSTRAINT customer_id_pk PRIMARY KEY(id)  
);
```

```
INSERT INTO customer  
VALUES ('asdf', '1234', 'kim');  
INSERT INTO customer(id, pwd)  
VALUES ('hong', '6789');
```

```
SELECT * FROM customer;
```

## 제약조건 변경하기

- 이미 만들어진 제약조건을 중간에 변경할 수 있다
- 제약조건을 변경하는 것은 테이블의 구조를 변경하는 것과 같다. - ALTER TABLE~~
  - 제약 조건 추가 : add
  - 제약 조건 변경 : modify
  - 제약 조건 제거 : drop

## 제약조건 추가

- 예제 : student 테이블과 registration테이블을 복제해서 stu\_copy와 reg\_copy를 만들어 본다

```
CREATE TABLE stu_copy  
AS  
SELECT * FROM student;
```

```
CREATE TABLE reg_copy  
AS  
SELECT * FROM registration;
```

- 기본키를 추가해 본다

```
ALTER TABLE stu_copy  
ADD CONSTRAINT stu_copy_stuno_pk PRIMARY KEY(stuno);
```

```
ALTER TABLE reg_copy  
ADD CONSTRAINT reg_copy_enrollid_pk PRIMARY KEY(enrollid);
```

- 외래키를 추가해 본다.

```
ALTER TABLE reg_copy  
ADD CONSTRAINT reg_copy_stuNo_fk FOREIGN KEY(stuNo) REFERENCES student(stuNo);
```

## 제약조건 수정

- null값을 허용하지 않도록 변경해보자

```
ALTER TABLE stu_copy  
MODIFY major CONSTRAINT stu_copy_major_nn NOT NULL;
```

## 제약조건 삭제

- reg\_copy, stu\_copy 테이블의 기본키를 제거해 본다.

```
ALTER TABLE reg_copy  
DROP PRIMARY KEY;  
  
ALTER TABLE stu_copy  
DROP PRIMARY KEY CASCADE;
```

- stu\_copy 테이블의 major컬럼의 NOT NULL 제약조건을 제거해 본다.

```
ALTER TABLE stu_copy  
DROP CONSTRAINT stu_copy_major_nn;
```

## 제약조건 활성화 비활성화

- student 테이블과 registration테이블에 적용된 제약조건을 확인해 보자

```
SELECT table_name,CONSTRAINT_NAME,status  
FROM user_constraints  
WHERE table_name IN('STUDENT','REGISTRATION');
```

- 외래키 제약조건으로 인해 아래 오류 발생 쿼리를 입력할 수 없다.

enrollid	stuno	subject
E003	S005	'constitutional law'

- 해당 쿼리를 삽입하기 위해 외래키를 잠시 비활성화 한다.

```
ALTER TABLE REGISTRATION  
disable CONSTRAINT REGISTRATION_STUNO_FK;
```



## 제약조건 활성화 비활성화

- 해당 쿼리를 삽입해보자 – 정상적으로 삽입이 된다.

enrollid	stuno	subject
E003	S005	'constitutional law'

```
INSERT INTO registration  
VALUES ('E003','S005','Constitutional Law');
```

- 다시 외래키를 활성화 해보자

에러 발생

```
ALTER TABLE REGISTRATION  
enable CONSTRAINT REGISTRATION_STUNO_FK;
```

- 에러가 발생하는데 이유는 기존에 입력된 데이터의 무결성을 검사하기 때문이다.

```
ALTER TABLE REGISTRATION  
enable NOVALIDATE CONSTRAINT REGISTRATION_STUNO_FK;
```

- 위와 같이 쿼리를 작성하는 경우 기존의 데이터 무결성 체크는 무시하고 이후에 입력되는 데이터의 무결성만 체크한다.