

# 11. 무결성 제약 조건

컴퓨터공학부

김은경

## [학습 목표]

1. 무결성 제약 조건의 다섯 가지 유형을 나열할 수 있다.
2. 다섯 가지 유형의 제약 조건을 정의할 수 있다.
3. 제약 조건을 추가할 수 있다.
4. 제약 조건을 삭제할 수 있다.
5. 정의된 제약 조건을 비활성화 시킬 수 있다.

# 1. 데이터 무결성

## ★ 데이터 무결성(Data Integrity)이란?

- 데이터 상호 간에 모순이 없는 완전무결한 데이터 상태를 의미함

## ★ 데이터 무결성의 유형

유형	특징
널 값	널 값을 포함하는 행 삽입 또는 변경의 허용 여부에 따라 데이터 무결성이 보장됨
고유 열 값	열 또는 열의 집합이 고유한 값을 가질 때만 행 삽입 또는 변경을 허용함으로써 데이터 무결성을 보장함
기본 키 값	해당 키 값이 모든 행을 테이블 내에서 고유하게 식별할 수 있을 때만 행 삽입 또는 변경을 허용함으로써 데이터 무결성이 보장됨
참조 무결성	해당 키 값이 참조하는 관련 테이블의 키 값과 일치할 때만 행 삽입 또는 변경을 허용하고, 참조하는 열이 있는 경우 삭제를 허용하지 않음으로써 데이터 무결성이 보장됨
복합 무결성	열 또는 열 집합이 포함하는 값에 따라 행 삽입 또는 변경의 허용 여부를 결정함으로써 데이터 무결성이 보장되는 것으로, 사용자가 정의하게 됨

# 1. 데이터 무결성

## ★ 제약 조건 정의 방법

- CREATE TABLE 명령을 사용해서 테이블을 생성할 때 제약 조건도 함께 정의
- CONSTRAINT라는 예약어 뒤에 열 또는 테이블에 대한 제약조건 기술함

```
CREATE TABLE 테이블이름  
  (열이름 데이터형 [DEFAULT 표현식] [CONSTRAINT 열_제약조건],  
    ...  
  [CONSTRAINT 테이블_제약조건] );
```

# 1. 데이터 무결성

## ★ 열 제약 조건 기술 형식

- 열을 정의하면서 바로 뒤에 열에 대한 제약 조건을 기술함
- 어떤 유형의 제약 조건도 정의할 수 있음

[CONSTRAINT 제약\_이름] 제약조건\_유형

## ★ 테이블 제약 조건 기술 형식

- 모든 열이 정의된 다음 맨 마지막에 테이블에 대한 제약 조건을 기술함
- 하나 이상의 열을 참조할 수 있고, NOT NULL을 제외한 모든 유형의 제약 조건을 정의할 수 있음

[CONSTRAINT 제약\_이름] 제약조건\_유형(열이름, ...)

# 1. 데이터 무결성

## ✓ 제약 조건 정의 예제

```
CREATE TABLE test_emp (  
    emp_id    NUMBER(4),  
    emp_name  VARCHAR2(10),  
    job       VARCHAR2(10),  
    mgr_id    NUMBER(4),  
    hiredate  DATE,  
    dept_id   NUMBER(3) NOT NULL,  
    CONSTRAINT test_emp_pk PRIMARY KEY (emp_id));
```

```
DESC test_emp;
```

Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
EMP_NAME		VARCHAR2(10)
JOB		VARCHAR2(10)
MGR_ID		NUMBER(4)
HIREDATE		DATE
DEPT_ID	NOT NULL	NUMBER(3)

# 1. 데이터 무결성

## ★ 제약 조건 관련 기타 사항들

### ▪ 제약 조건 정의의 필요성

- ① 테이블에 부적합한 데이터가 삽입되는 것을 방지함
- ② 데이터의 무결성을 해치는 데이터 변경이나 삭제를 방지함
- ③ 다른 테이블에 종속성이 존재할 경우, 테이블이 삭제되는 것을 방지함

### ▪ 제약 조건 정의 지침

- ① 열 또는 테이블 수준에서 제약 조건을 정의함
- ② 사용자가 내용을 짐작할 수 있도록 제약 조건에 이름을 부여하는 것이 바람직함
- ③ 사용자가 이름을 생략하면, 오라클 서버가 SYS Cn(n: 정수) 형태로 자동으로 명명함

# 1. 데이터 무결성

## ★ 제약 조건 관련 기타 사항들

### ▪ 제약 조건 생성 시점

- ① 테이블이 생성될 때 동시에 생성할 수 있음
  - CREATE TABLE 명령으로 테이블을 생성할 때 제약 조건을 포함해서 생성하는 함
- ② 테이블이 생성된 다음 생성할 수 있음
  - ALTER TABLE 명령으로 제약 조건을 추가함  
(단, 추가되는 제약 조건을 위배하는 데이터가 있는 경우 제약 조건을 추가할 수 없음)

### ▪ 제약 조건 정의 확인

- 데이터 사전의 USER\_CONSTRAINTS 테이블이나 USER\_CONS\_COLUMNS 뷰가 정의된 제약 조건에 대한 모든 정보를 포함하고 있음



## 2. 무결성 제약 조건

### ★ 무결성 제약 조건의 유형

- ① 오라클 서버 차원에서 지원하는 제약 조건 : NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY
- ② 사용자가 정의하는 제약 조건 : CHECK

유형	특징	데이터 무결성
NOT NULL	. 널 값이 허용하지 않음 . 열에 대해서만 정의할 수 있음	널 값
UNIQUE	. 테이블의 모든 행에 대해 유일한 값을 가져야 함 . 단, 널 값 및 널 값의 중복은 허용함	고유 열 값
PRIMARY KEY	. 테이블의 각 행을 유일하게 식별함 (NOT NULL 제약 조건 + UNIQUE 제약 조건)	기본 키 값
FOREIGN KEY	. 참조하는 테이블에 있는 값만 허용함	참조 무결성
CHECK	. 참이 되어야 하는 조건을 지정함	복합 무결성

## 2. 무결성 제약 조건

### ★ NOT NULL 제약 조건

- 어떤 열이 널 값을 포함하면 행 삽입 또는 변경을 허용하지 않기 위해 사용되는 제약 조건

#### ▪ NOT NULL 제약 조건의 특징

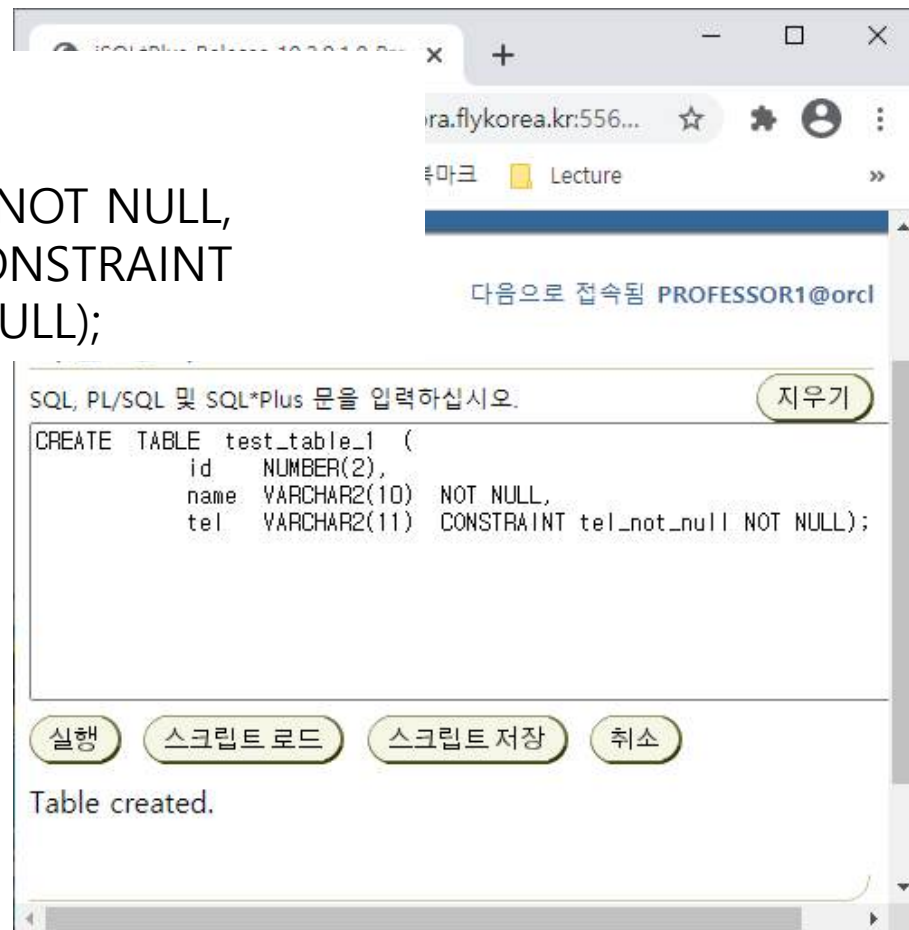
- ① 특정 열에 대해 널 값이 허용되지 않도록 보장함
- ② 열에 대해서만 정의할 수 있음

## 2. 무결성 제약 조건

### ✓ NOT NULL 제약 조건 정의 예

- id와 name, tel이라는 세 개의 열을 갖는 test\_table\_1이라는 테이블을 생성하되, name과 tel이 널 값을 갖는 행은 삽입될 수 없도록 정의하시오.

```
CREATE TABLE test_table_1 (  
    id    NUMBER(2),  
    name  VARCHAR2(10) NOT NULL,  
    tel   VARCHAR2(11) CONSTRAINT  
        tel_not_null NOT NULL);
```



## 2. 무결성 제약 조건

→ tel이 널 값을 갖는 행을 하나 삽입하기 :

```
INSERT INTO test_table_1 (id, name)
VALUES (11, '김경아');
```

Tel이 널 값을 갖는 행은  
삽입될 수 없음



## 2. 무결성 제약 조건

### ★ UNIQUE 제약 조건

- 어떤 열의 값이 테이블에 있는 모든 행에 대해 유일할 때만 행 삽입 또는 변경을 허용하기 위해 사용되는 제약 조건

#### ▪ UNIQUE 제약 조건의 특징

- ① 테이블의 모든 행에 대해서 유일한 값을 갖도록 보장함
- ② 열 또는 열의 집합에 대해서 중복된 값을 허용하지 않음
- ③ NOT NULL 제약 조건을 정의하지 않으면, 널 값을 허용하고, 널 값의 중복도 허용함
- ④ 테이블이나 열에 대해서 정의할 수 있음
- ⑤ 이미 중복된 값이 있는 열에 이 제약조건이 추가되면, 추후 추가되는 데이터에 대해서만 이 제약 조건이 적용됨
- ⑥ UNIQUE 제약 조건이 정의된 열에는 인덱스가 자동으로 생성됨

## 2. 무결성 제약 조건

### ✓ UNIQUE 제약 조건 정의 예

- num과 name, loc라는 세 개의 열을 갖는 test\_table\_2라는 테이블을 생성하되, loc가 모든 행에 대해서 유일한 값을 가질 때만 삽입이 허용되도록 정의하기

```
CREATE TABLE test_table_2 (  
    num    NUMBER(3),  
    name   VARCHAR2(12),  
    loc    VARCHAR2(20) UNIQUE);
```



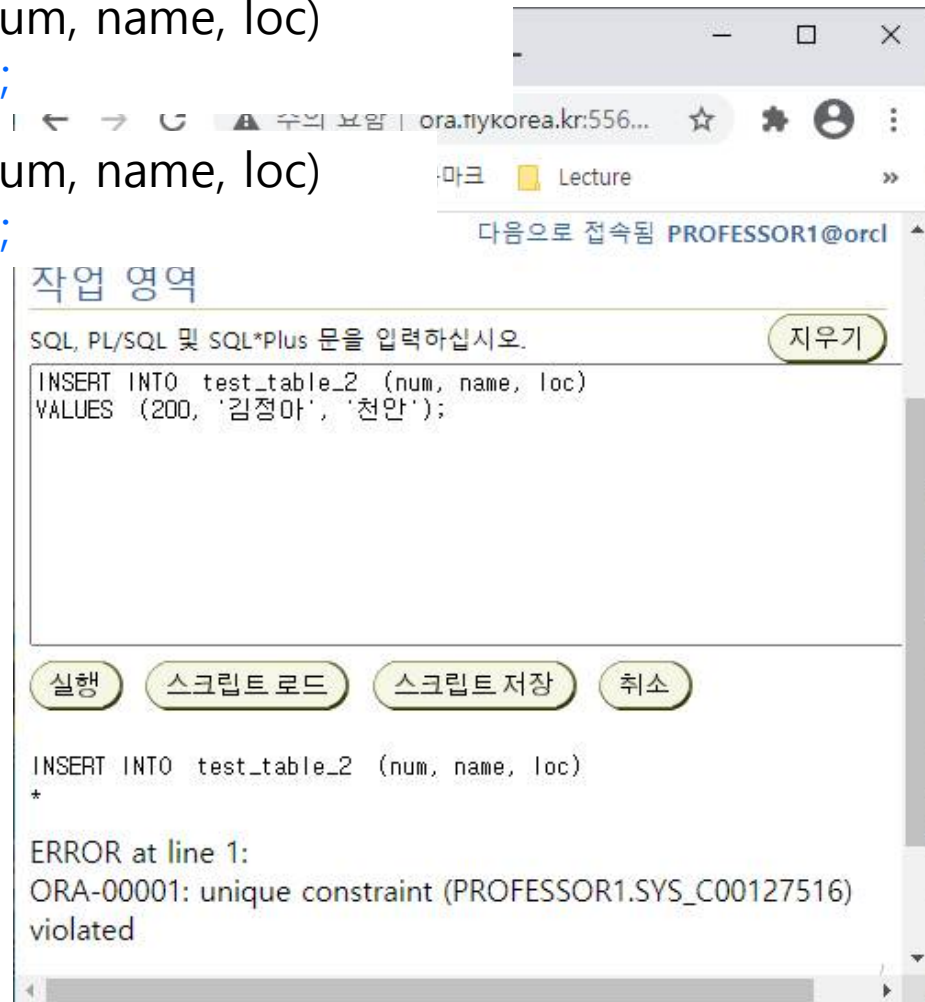
## 2. 무결성 제약 조건

→ loc가 동일한 값을 갖는 행 삽입하기 :

```
INSERT INTO test_table_2 (num, name, loc)  
VALUES (100, '박홍수', '천안');
```

```
INSERT INTO test_table_2 (num, name, loc)  
VALUES (200, '김정아', '천안');
```

loc는 유일한 값을 가져야 하므로 '천안' 이라는 동일한 값이 입력되면 오류가 발생함



## 2. 무결성 제약 조건

### ★ PRIMARY KEY 제약 조건

- 어떤 열이 기본 키인 경우, 테이블에 있는 모든 행에 대해 유일하면서 널 값을 갖지 않을 때만 행 삽입 또는 변경을 허용하기 위한 제약 조건

#### ▪ PRIMARY KEY 제약 조건의 특징

- ① 테이블의 각 행을 유일하게 식별할 수 있는 기본 키를 생성하도록 함
- ② 열 또는 열 집합에 대해서 유일성을 요구하고, 널 값을 허용하지 않음  
(UNIQUE 제약 조건 + NOT NULL 제약 조건)
- ③ 테이블이나 열에 대해서 정의 가능함
- ④ PRIMARY KEY 제약 조건이 정의된 열에는 인덱스가 자동으로 생성됨
- ⑤ **하나 이상의 복합 속성이 기본 키**를 이룰 때는, 괄호 안에 하나 이상의 열 이름을 콤마로 구분해서 기술하면 됨

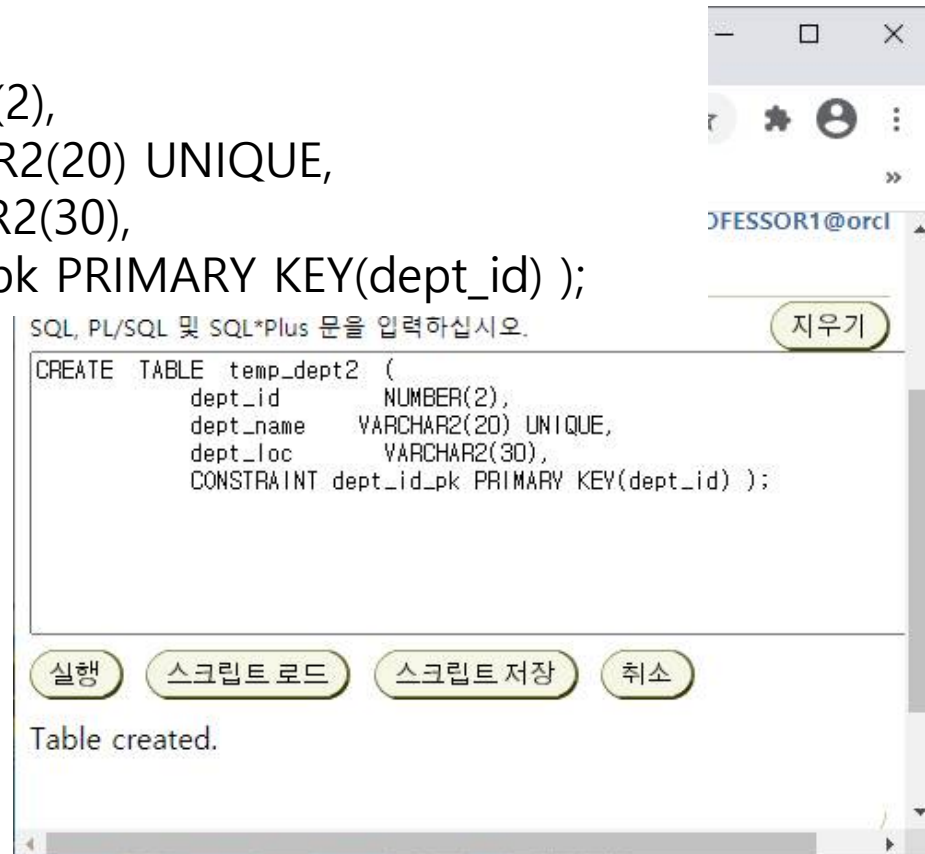


## 2. 무결성 제약 조건

### ✓ PRIMARY KEY 제약 조건 정의 예

- dept\_id와 dept\_name, dept\_loc라는 세 개의 열을 갖는 temp\_dept2라는 테이블을 생성하되, dept\_id를 기본 키로 정의하고, dept\_name이 유일한 값을 가질 때만 삽입이 허용되도록 정의하기

```
CREATE TABLE temp_dept2 (  
    dept_id      NUMBER(2),  
    dept_name    VARCHAR2(20) UNIQUE,  
    dept_loc     VARCHAR2(30),  
    CONSTRAINT dept_id_pk PRIMARY KEY(dept_id) );
```



## 2. 무결성 제약 조건

→ 기본 키 제약조건을 위반하는 임의의 행 삽입하기 :

```
INSERT INTO temp_dept2 (dept_name, dept_loc)  
VALUES ('관리부', '부산');
```

기본 키는 널 값을 가질 수  
없으므로 오류가 발생함



## 2. 무결성 제약 조건

### ★ FOREIGN KEY 제약 조건

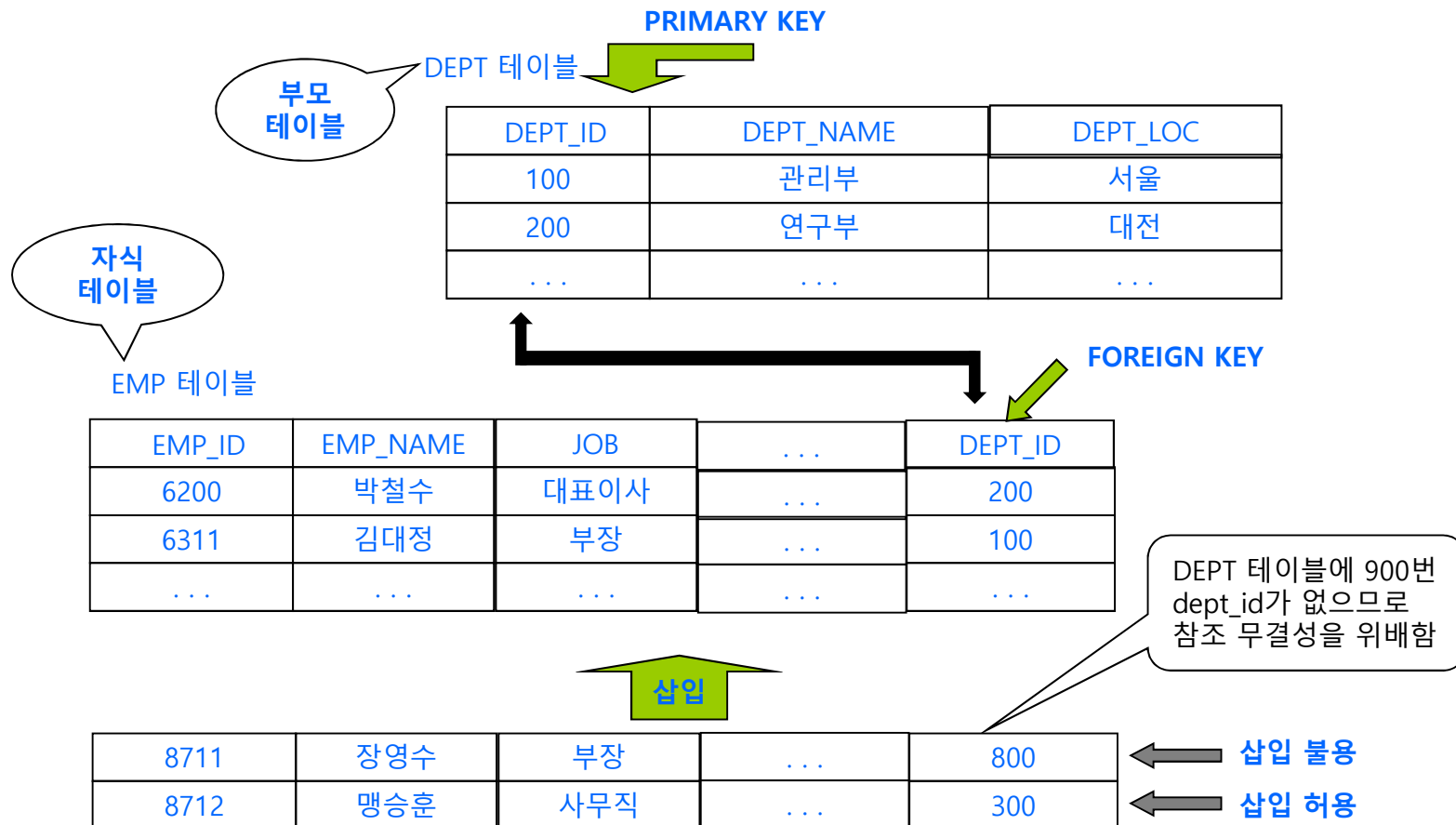
- 어떤 열이 같은 테이블 또는 다른 테이블의 열을 참조하는 경우, 해당 열이 참조하는 테이블에 있는 값을 가질 때만 행 삽입 또는 변경을 허용하고, 또 참조되는 테이블에서는 참조하는 열이 있는 경우 삭제를 허용하지 않도록 하기 위해 사용하는 제약 조건

#### ▪ FOREIGN KEY 제약 조건의 특징

- ① 외래 키 즉, 참조 키를 정의하는 것으로, 참조하는 테이블에 있는 값만 갖도록 보장함
- ② 동일 테이블 또는 다른 테이블에 있는 기본 키나 유일 키 사이의 관계를 설정함
- ③ 자식 테이블 : 외래 키를 정의한 테이블을 칭함
- ④ 부모 테이블 : 외래 키가 참조하는 열을 포함하는 테이블을 칭함
- ⑤ 테이블이나 열에 대해서 정의할 수 있음
- ⑥ 자식 테이블이 참조하고 있는 행을 부모 테이블에서 삭제하는 것은 참조 무결성을 위배하게 되므로 허용되지 않음

## 2. 무결성 제약 조건

### ✓ 부모 테이블과 자식 테이블



## 2. 무결성 제약 조건

### ★ FOREIGN KEY 제약 조건을 정의하는 키워드

- ① FOREIGN KEY : 자식 테이블의 열을 정의함
- ② REFERENCES : 부모 테이블과 그 테이블에서 참조할 열을 정의함
- ③ ON DELETE CASCADE : 부모 테이블에 있는 행이 삭제되면, 자식 테이블에 있는 종속적인 행도 함께 삭제되도록 허용하는 옵션

### ✓ FOREIGN KEY 제약 조건 정의 예

```
CREATE TABLE temp_emp (  
    emp_id    NUMBER(3),  
    emp_name  VARCHAR2(10) NOT NULL,  
    job       VARCHAR2(20),  
    sal       NUMBER(6),  
    mgr_id    NUMBER(3),  
    dept_id   NUMBER(2) NOT NULL,  
    CONSTRAINT e_id_pk PRIMARY KEY (emp_id),  
    CONSTRAINT d_id_fk FOREIGN KEY(dept_id)  
        REFERENCES temp_dept2(dept_id) );
```

## 2. 무결성 제약 조건

### ★ CHECK 제약 조건

- 어떤 열의 값이 사용자가 정의한 특정 조건을 만족할 때만 행 삽입 또는 변경을 허용하도록 하기 위해서 사용하는 제약 조건

#### ▪ CHECK 제약 조건의 특징

- ① 사용자가 각 열의 값이 만족해야 하는 제약 조건을 지정함
- ② 각 열에 정의할 수 있는 CHECK 제약 조건의 수는 제한이 없음
- ③ 열 또는 테이블에 대해 정의 가능함

## 2. 무결성 제약 조건

### ✓ CHECK 제약 조건 정의 예

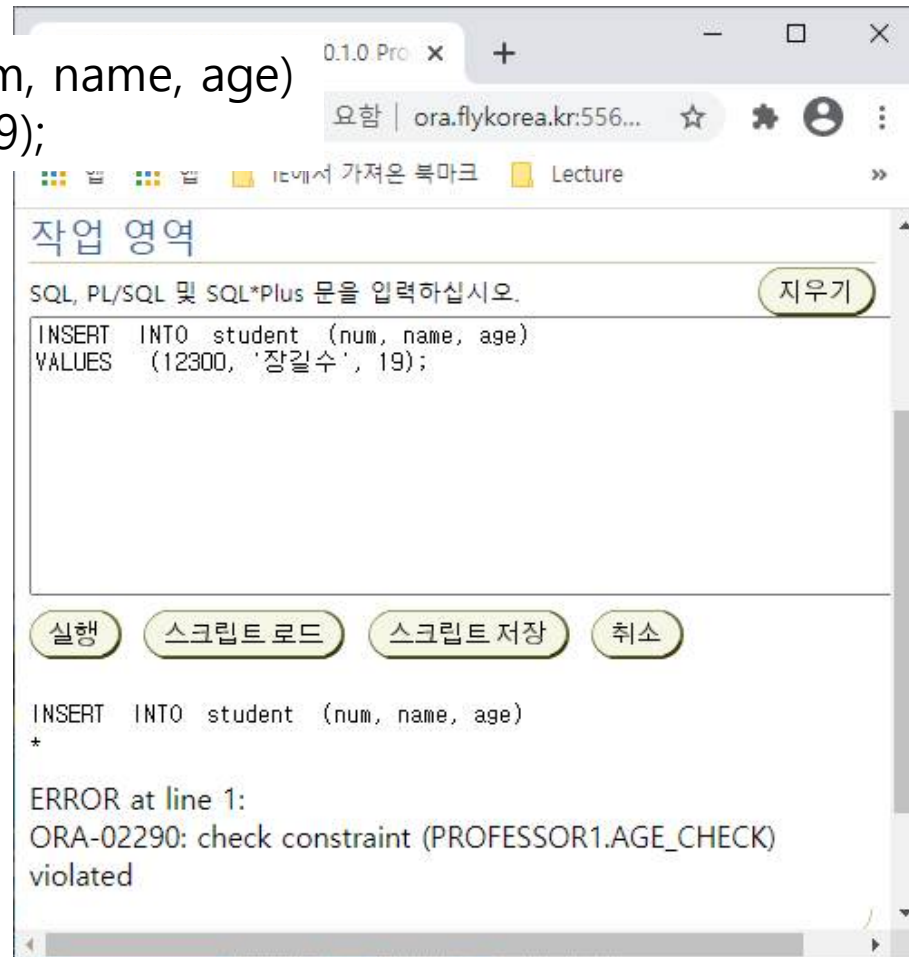
```
CREATE TABLE student (  
    num    NUMBER(5) PRIMARY KEY,  
    name   VARCHAR2(10),  
    age    NUMBER(2),  
    CONSTRAINT age_check  
        CHECK (age BETWEEN 20 AND 29));
```



## 2. 무결성 제약 조건

→ age가 유효 범위를 벗어나는 데이터 삽입하기 :

```
INSERT INTO student (num, name, age)
VALUES (12300, '장길수', 19);
```



CHECK 제약조건에서 정의된 age의 유효 범위를 벗어나면 오류가 발생함



### 3. 제약 조건 추가

#### ★ 제약 조건 추가 방법

- ALTER TABLE 명령의 ADD나 MODIFY 절을 이용해서 테이블을 생성할 때 미처 정의하지 못한 새로운 제약 조건을 추가할 수 있음

```
ALTER TABLE 테이블_이름  
ADD | MODIFY  
[CONSTRAINT 제약조건_이름] 제약조건_유형 (열_이름);
```

- ① ADD 절 : 테이블의 특정 열에 **NOT NULL**을 제외<sup>한</sup> 제약 조건 추가
- ② MODIFY 절 : 열에 대해서 NOT NULL 제약 조건 추가

#### ★ 제약 조건 변경 시 주의사항

- ① 제약 조건의 직접적인 수정은 불가능하며, 해당 제약 조건을 삭제한 후에 새로 추가해야 함
- ② 추가되는 제약 조건을 위배하는 데이터가 있는 경우, 그 데이터를 삭제하거나 수정하기 전에는 제약 조건을 추가할 수 없음

### 3. 제약 조건 추가

#### ★ 테이블에 대한 제약 조건 추가 방법

- NOT NULL을 제외한 제약 조건은 ADD 절에 추가할 제약 조건의 유형과 열 이름을 기술해서 추가함

```
ALTER TABLE 테이블_이름  
ADD [CONSTRAINT 제약조건_이름] 제약조건_유형 (열_이름);
```

#### ★ 열에 대한 제약 조건 추가 방법

- NOT NULL 제약 조건은 MODIFY 절에 열 이름과 함께 기술해서 추가함

```
ALTER TABLE 테이블_이름  
MODIFY 열_이름 [CONSTRAINT 제약조건_이름] NOT NULL;
```

### 3. 제약 조건 추가

#### ✓ 테이블 제약 조건 추가 사례

- temp\_emp 테이블에서 관리자 번호(mgr\_id)가 같은 테이블에 있는 다른 사원번호(emp\_id)를 참조하도록 mgr\_id\_fk라는 이름의 제약 조건을 추가한 다음, 데이터 사전의 USER\_CONSTRAINTS 테이블에서 확인하시오.

```
ALTER TABLE temp_emp
ADD CONSTRAINT mgr_id_fk
FOREIGN KEY(mgr_id) REFERENCES temp_emp(emp_id);
```

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

CONSTRAINT_NAME	CON	SEARCH_CONDITION
D_ID_FK	R	
E_ID_PK	P	
SYS_C00127522	C	"DEPT_ID" IS NOT NULL
SYS_C00127521	C	"EMP_NAME" IS NOT NULL
MGR_ID_FK	R	

#### [참고]

CONSTRAINT_TYPE	의미
P	PRIMARY KEY
R	FOREIGN KEY
U	UNIQUE
C	CHECK, NOT NULL

### 3. 제약 조건 추가

#### ✓ 열 제약 조건 추가 사례

- temp\_emp 테이블에서 job이 널 값을 가질 수 없도록 제약 조건을 추가하기

```
ALTER TABLE temp_emp  
    MODIFY job CONSTRAINT job_not_null NOT NULL;
```

#### [주의 사항]

- 만약 job 열이 널 값을 갖고 있는 행이 있는 경우, 해당 행을 삭제해야 위의 제약 조건을 추가할 수 있음

## 4. 제약 조건 삭제

### ★ 제약 조건 삭제 방법

- ALTER TABLE 명령의 DROP 절을 이용해서 제약 조건을 삭제할 수 있음

```
ALTER TABLE 테이블_이름  
DROP PRIMARY KEY | UNIQUE(열_이름) |  
CONSTRAINT 제약조건_이름 [CASCADE];
```

### ★ 제약 조건 삭제 시 주의사항

- ① 제약조건\_이름은 데이터 사전의 USER\_CONSTRAINTS 테이블이나 USER\_CONS\_COLUMNS 뷰에서 확인할 수 있음
- ② PRIMARY KEY는 한 테이블에 하나 밖에 없으므로 제약조건\_이름이나 열\_이름 없이도 삭제할 수 있음
- ③ CASCADE 옵션을 지정하면, 모든 종속적인 제약 조건을 한꺼번에 삭제하게 됨

## 4. 제약 조건 삭제

### ✓ 제약 조건 삭제 사례

- foreign\_emp 테이블에서 sal에 대한 CHECK 제약 조건 삭제하기

```
ALTER TABLE foreign_emp  
DROP CONSTRAINT sal_check;
```

Table altered.

[참고] sal에 대한 CHECK 제약 조건 이름 확인하기

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

CONSTRAINT_NAME	CON	SEARCH_CONDITION
FOREMP_ID_PK	P	
SYS_C00127497	C	"SAL" IS NOT NULL
SYS_C00127496	C	"JOB" IS NOT NULL
SYS_C00127495	C	"EMP_NAME" IS NOT NULL
SAL_CHECK	C	sal >= 2500

## 4. 제약 조건 삭제

### ✓ 종속적인 제약 조건 삭제 사례

- temp\_dept2 테이블에서 기본 키 제약 조건을 삭제하되, temp\_emp 테이블의 dept\_id 열에 정의한 이 기본 키에 대한 참조 키 제약 조건도 함께 삭제하도록 삭제하시오.

```
ALTER TABLE temp_dept2  
        DROP PRIMARY KEY CASCADE;
```

Table altered.

→ temp\_dept2 테이블의 제약조건 확인하기 :

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE, SEARCH_CONDITION  
FROM   USER_CONSTRAINTS  
WHERE  TABLE_NAME = 'TEMP_DEPT2';
```

CONSTRAINT_NAME	CON	SEARCH_CONDITION
SYS_C00127518	U	

PRIMARY KEY 제약조건은 삭제되었고, dept\_name에 대한 UNIQUE 제약조건만 남아 있음

## 4. 제약 조건 삭제

→ temp\_dept2 테이블의 기본 키인 dept\_id를 참조하고 있는 temp\_emp 테이블의 제약조건 확인하기 :

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

CONSTRAINT_NAME	CON	SEARCH_CONDITION
E_ID_PK	P	
SYS_C00127522	C	"DEPT_ID" IS NOT NULL
SYS_C00127521	C	"EMP_NAME" IS NOT NULL
MGR_ID_FK	R	
JOB_NOT_NULL	C	"JOB" IS NOT NULL

CASCADE 옵션에 의해 외래 키 제약조건(R)이 함께 삭제되었음



## 5. 제약 조건 비활성화

### ★ 제약 조건 비활성화 방법

- ALTER TABLE 명령의 DISABLE CONSTRAINT 절을 이용해서 이미 정의된 제약 조건이 일시적으로 적용되지 않도록 지정할 수 있음

```
ALTER TABLE 테이블_이름  
        DISABLE CONSTRAINT 제약조건_이름 [CASCADE];
```

### ★ 제약 조건 비활성화 시 주의사항

- ① CASCADE 옵션을 지정하면 모든 종속적인 제약 조건을 비활성화 시킴  
단, 다시 활성화시킬 때는 각각 ENABLE 시켜야 함
- ② 테이블에 많은 데이터를 삽입할 때, 매번 제약 조건을 체크 하게 되면  
성능이 저하되므로 일시적으로 비활성화 시키는 것이 효율적임  
단, 삽입되는 데이터에 오류가 없어야 함
- ③ UNIQUE나 PRIMARY KEY 제약 조건이 비활성화 되면, 자동으로 생성된  
인덱스가 삭제됨

## 5. 제약 조건 비활성화

### ✓ 제약 조건 비활성화 예

- temp\_emp 테이블에서 emp\_name의 NOT NULL 제약 조건 비활성화 시키기

```
ALTER TABLE temp_emp  
DISABLE CONSTRAINT sys_c00127521;
```

[참고] temp\_emp 테이블에 대한 제약 조건 이름 확인하기

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

CONSTRAINT_NAME	CON	SEARCH_CONDITION
E_ID_PK	P	
SYS_C00127522	C	"DEPT_ID" IS NOT NULL
SYS_C00127521	C	"EMP_NAME" IS NOT NULL
MGR_ID_FK	R	
JOB_NOT_NULL	C	"JOB" IS NOT NULL

## 6. 제약 조건 활성화

### ★ 제약 조건 활성화 방법

- ALTER TABLE 명령의 ENABLE CONSTRAINT 절을 이용해서 현재 비활성화된 제약 조건을 다시 활성화 시킬 수 있음

```
ALTER TABLE 테이블_이름  
        ENABLE CONSTRAINT 제약조건_이름;
```

### ★ 제약 조건 활성화 시 주의사항

- ① CASCADE 옵션으로 연쇄적으로 비활성화 된 제약 조건도 활성화시킬 때는 각각 활성화시켜야 함
- ② UNIQUE나 PRIMARY KEY 제약 조건이 활성화되면, 인덱스가 자동 생성됨

## 6. 제약 조건 활성화

### ✓ 제약 조건 활성화 예

- temp\_emp 테이블에서 비활성화 시킨 emp\_name의 NOT NULL 제약 조건을 다시 활성화 시키기

```
ALTER TABLE temp_emp  
    ENABLE CONSTRAINT sys_c00127521;
```

**[주의]** emp\_name이 널 값을 갖는 행이 삽입된 경우, NOT NULL 제약 조건을 다시 활성화 시킬 수 없음

## 7. 제약 조건 확인

### ★ 테이블의 제약 조건 확인 방법

- 데이터 사전의 USER\_CONSTRAINTS 테이블에서 확인할 수 있음
- USER\_CONSTRAINTS 테이블에서 유용한 열 정보
  - ① CONSTRAINT\_NAME : 사용자나 오라클 서버가 부여한 제약조건 이름 표시
  - ② CONSTRAINT\_TYPE : 제약 조건의 유형 표시
  - ③ SEARCH\_CONDITION : NOT NULL을 포함해서 CHECK 제약 조건 표시

### ★ 열의 제약 조건 확인 방법

- 데이터 사전의 USER\_CONS\_COLUMNS 뷰에서 확인할 수 있음
- USER\_CONS\_COLUMNS 뷰에서 유용한 열 정보
  - ① CONSTRAINT\_NAME : 사용자나 오라클 서버가 부여한 열 제약조건 이름 표시
  - ② COLUMN\_NAME : 제약 조건이 정의된 열 이름 표시
- 'NOT NULL' 제약 조건은 DESCRIBE 명령으로도 확인할 수 있음

## 7. 제약 조건 확인

### ★ 제약 조건의 유형

- ① C : CHECK (NOT NULL 포함)
- ② P : PRIMARY Key
- ③ U : UNIQUE Key
- ④ R : REFERENCE 무결성

## 7. 제약 조건 확인

### ✓ 테이블 제약 조건 확인 사례

- 데이터 사전에서 emp 테이블에 정의된 제약 조건의 이름과 유형, CHECK 제약 조건 확인하기

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

CONSTRAINT_NAME	CON	SEARCH_CONDITION
EMP_DEPT_ID_FK	R	
EMP_ID_PK	P	
SYS_C00127487	C	"HIREDATE" IS NOT NULL
SYS_C00127486	C	"SAL" IS NOT NULL
SYS_C00127485	C	"JOB" IS NOT NULL
SYS_C00127484	C	"EMP_NAME" IS NOT NULL
EMP_SELF_KEY	R	

7 rows selected.

오라클 서버가 부여한  
제약 조건 이름  
(SYS\_로 시작)

사용자가 부여한  
제약 조건 이름

NOT NULL 제약 조건도  
유형을 C로 표시함

PRIMARY KEY는  
NOT NULL 포함

[비교] DESC emp;

Name	Null?	Type
EMP_ID	NOT NULL	NUMBER(4)
EMP_NAME	NOT NULL	VARCHAR2(10)
JOB	NOT NULL	VARCHAR2(10)
DEPT_ID		NUMBER(3)
SAL	NOT NULL	NUMBER(4)
BONUS		NUMBER(4)
MGR_ID		NUMBER(4)
HIREDATE	NOT NULL	DATE

## 7. 제약 조건 확인

### ✓ 열 제약 조건 확인 사례

- 데이터 사전에서 emp 테이블의 각 열에 정의된 제약 조건의 이름과 해당 열 이름 확인하기

```
SELECT CONSTRAINT_NAME, COLUMN_NAME  
FROM   USER_CONS_COLUMNS  
WHERE  TABLE_NAME = 'EMP';
```

CONSTRAINT_NAME	COLUMN_NAME
SYS_C00127484	EMP_NAME
SYS_C00127485	JOB
SYS_C00127486	SAL
SYS_C00127487	HIREDATE
EMP_ID_PK	EMP_ID
EMP_DEPT_ID_FK	DEPT_ID
EMP_SELF_KEY	MGR_ID

7 rows selected.





**Q & A**