



# 계약조건 개요



## 제약조건 개요

### 1. 제약조건 개요

#### 1.1 제약조건 정의

- ☐ 데이터베이스는 제약조건을 사용하여 데이터를 일관성 없는 조작으로부터 보호
- ☐ 제약조건은 그 테이블에서 행이 삽입되거나, 갱신되거나, 삭제될 때마다 테이블에서 규칙 적용
- ☐ 제약조건은 종속성이 존재할 경우 테이블의 삭제를 방지



## 제약조건 개요

## 1.2 제약조건 유형

유 형	설 명
NOT NULL	열이 널 (NULL) 값을 포함하지 않도록 지정
UNIQUE	테이블의 모든 행에 대해 고유한 값을 갖는 열 또는 열 조합을 지정
PRIMARY KEY	테이블의 각 행을 고유하게 식별
FOREIGN KEY	한 열과 참조된 테이블의 열 간에 외래 키 관계를 설정하고 시행
CHECK	참 (True) 이어야 하는 조건을 지정

## 제약조건 개요

## 2. 제약조건 생성

### 2.1 제약조건 생성 구문

```
CREATE TABLE [schema.]table  
  (column datatype [DEFAULT expr] [column_constraint],  
  ... ,  
  [table_constraint] [, ...]);
```

```
CREATE TABLE emp  
  (empno      NUMBER(4) ,  
   ename      VARCHAR2(10) ,  
   ...  
   deptno     NUMBER(7,2) NOT NULL,  
   CONSTRAINT emp_empno_pk PRIMARY KEY (empno));
```

## 2.1 제약조건 생성 구문 (계속)

### ☞ 열 레벨 제약조건

- 열별로 정의
- 무결성 제약조건의 어떤 유형도 정의 가능

```
column [CONSTRAINT constraint_name] constraint_type,
```

### ☞ 테이블 레벨 제약조건

- 하나 이상의 열을 참조
- 테이블의 열 정의와는 개별적으로 정의
- NOT NULL을 제외한 임의의 제약조건 정의 가능

```
column, ...,  
[CONSTRAINT constraint_name] constraint_type (column, ...),
```

## 2.2 제약조건 사용 지침

- ☐ 사용자가 제약조건 이름을 지정하지 않으면 오라클 서버는 **sys\_cn** 포맷을 사용하여 이름을 생성
- ☐ 제약조건 생성 시기
  - 테이블이 생성될 때
  - 테이블이 생성된 후
- ☐ 열 레벨 또는 테이블 레벨로 제약조건을 정의
- ☐ 데이터 디렉터리에서 제약조건 확인
  - **USER\_CONSTRAINTS**



# 계약조건 활용



## 1. NOT NULL

### 1.1 NOT NULL 제약조건 개요

- NOT NULL 제약조건은 열에 널 값을 허용하지 않음
- NOT NULL 제약조건이 없는 열은 기본적으로 널 값을 가질 수 있음
- NOT NULL 제약조건은 열 레벨에서만 정의 가능

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81/11/17	5000		10
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7566	JONES	MANAGER	7839	81/04/02	2975		20
7902	FORD	ANALYST	7566	81/10/03	3000		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
...							

NOT NULL 제약조건

(이 열의 모든 행은 널 값을 가질 수 없음)

NOT NULL 제약조건 없음

(이 열의 모든 행은 널 값을 가질 수 있음)



## 1.2 NOT NULL 제약조건 활용

```
SQL> CREATE TABLE emp_test (  
2     empid      NUMBER(5),  
3     empname    VARCHAR2(10) NOT NULL,  
4     duty       VARCHAR2(9),  
5     sal        NUMBER(7,2),  
6     bonus      NUMBER(7,2),  
7     mgr        NUMBER(5),  
8     hire_date  DATE,  
9     deptid     NUMBER(2));
```

```
... ,  
empname    VARCHAR2(10)  
                CONSTRAINT emp_test_empname_nn NOT NULL,  
...
```

## 2. UNIQUE


### 2.1 UNIQUE 제약조건 개요

- ☐ **UNIQUE** 제약조건은 열 또는 열 집합의 모든 값이 고유하게 함
- ☐ **UNIQUE** 제약조건이 하나 이상의 열을 포함하면 해당 열 그룹을 조합 고유 키 (Composite Unique Key)라고 함
- ☐ 동일한 열에 **NOT NULL** 제약조건을 정의하지 않으면 **UNIQUE** 제약조건은 널 값을 허용
- ☐ **UNIQUE** 제약조건은 열 레벨 또는 테이블 레벨로 정의 가능
  - 열 조합의 경우 테이블 레벨로 정의
- ☐ **UNIQUE** 인덱스가 자동으로 생성




## 2.1 UNIQUE 제약조건 개요 (계속)

UNIQUE 제약조건



DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

INSERT INTO



50	SALES	DAEGU
60		SEOUL

False ('DAEGU' 가 이미 있음)

True

## 2.2 UNIQUE 제약조건 활용

```
SQL> CREATE TABLE dept_test (  
2     deptid      NUMBER(2) ,  
3     dname       VARCHAR2(14) ,  
4     loc         VARCHAR2(13) ,  
5     CONSTRAINT dept_test_dname_uk UNIQUE(dname) ) ;
```



### 3. PRIMARY KEY


#### 3.1 PRIMARY KEY 제약조건 개요

- ☐ PRIMARY KEY 제약조건은 테이블의 각 행을 고유하게 식별하는 열 또는 열 집합
- ☐ 기본키 (Primary key)는 테이블 당 하나만 생성
- ☐ 열 또는 열 조합에 고유성을 부여하며 기본키의 일부인 모든 열은 널 값을 가질 수 없음
- ☐ PRIMARY KEY 제약조건은 열 레벨 또는 테이블 레벨에서 정의 가능
  - 열 조합에 대한 PRIMARY KEY 제약조건은 테이블 레벨에서 정의 가능
- ☐ UNIQUE 인덱스가 자동 생성




## 3.1 PRIMARY KEY 제약조건 개요 (계속)

PRIMARY KEY 제약조건



DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

INSERT INTO



20	TEST	DAEGU
	TEST01	SEOUL

 False (DEPTNO '20'이 이미 있음)

 False (DEPTNO가 널임)

### 3.2 PRIMARY KEY 제약조건 활용

```
SQL> CREATE TABLE dept_test (  
2     deptid      NUMBER(2) ,  
3     dname       VARCHAR2(14) ,  
4     loc         VARCHAR2(13) ,  
5     CONSTRAINT dept_test_deptid_pk PRIMARY KEY(deptid) ,  
6     CONSTRAINT dept_test_dname_uk UNIQUE(dname) ) ;
```



## 4. FOREIGN KEY

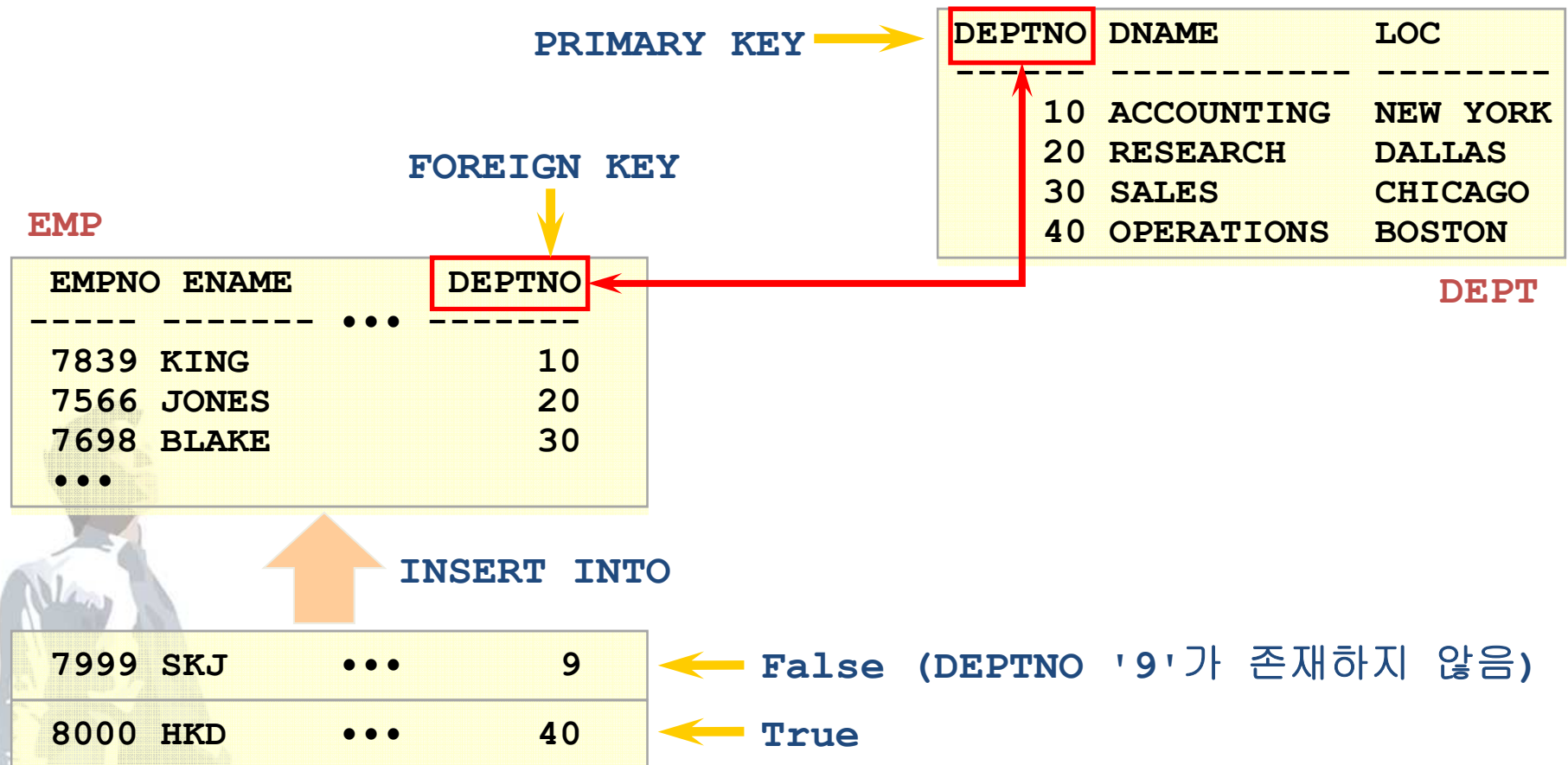
### 4.1 FOREIGN KEY 제약조건 개요

- ☐ **FOREIGN KEY** 제약조건은 열 또는 열 조합을 외래키로 지정하고 동일한 테이블이나 다른 테이블에 있는 기본키 또는 고유키와의 관계 설정
- ☐ **FOREIGN KEY** 값은 부모 테이블의 기존 값과 일치하거나 **NULL**
- ☐ **FOREIGN KEY**는 데이터 값을 기준으로 하며 물리적 포인터가 아닌 논리적 포인터
- ☐ **FOREIGN KEY** 제약조건은 열 레벨 또는 테이블 레벨에서 정의 가능
  - 열 조합에 대한 **FOREIGN KEY** 제약조건은 테이블 레벨에서 정의 가능





## 4.1 FOREIGN KEY 제약조건 개요 (계속)



## 4.2 FOREIGN KEY 제약조건 활용

```
SQL> CREATE TABLE emp_test (  
2     empid      NUMBER(5),  
3     empname    VARCHAR2(10) NOT NULL,  
4     duty       VARCHAR2(9),  
5     sal        NUMBER(7,2),  
6     bonus      NUMBER(7,2),  
7     mgr        NUMBER(5),  
8     hire_date  DATE,  
9     deptid     NUMBER(2),  
10    CONSTRAINT emp_test_deptid_fk FOREIGN KEY (deptid)  
11    REFERENCES dept_test(deptid));
```

### 4.3 FOREIGN KEY 제약조건 키워드

#### ☐ FOREIGN KEY

- 테이블 제약조건 레벨로 자식 테이블의 열을 정의

#### ☐ REFERENCES

- 부모 테이블 및 부모 테이블에서의 해당 열을 식별

#### ☐ ON DELETE CASCADE

- 부모 테이블의 행이 삭제되는 경우, 자식 테이블의 종속 행을 삭제

#### ☐ ON DELETE SET NULL

- 종속 외래키 값을 널로 변환



## 5. CHECK

### 5.1 CHECK 제약조건 개요

- ☐ 각 행을 만족해야 하는 조건을 정의
- ☐ 허용되지 않는 표현식
  - `CURRVAL`, `NEXTVAL`, `LEVEL`, `ROWNUM` 의사열 (pseudocolumns) 참조
  - `SYSDATE`, `UID`, `USER`, `USERENV` 함수 호출
  - 다른 행의 다른 값을 참조하는 질의
- ☐ **CHECK** 제약조건은 열 레벨 또는 테이블 레벨에서 정의 가능



## 5.2 CHECK 제약조건 활용

```
SQL> CREATE TABLE emp_test (  
2     empid      NUMBER(5),  
3     empname    VARCHAR2(10) NOT NULL,  
4     duty       VARCHAR2(9),  
5     sal        NUMBER(7,2),  
6     bonus      NUMBER(7,2),  
7     mgr        NUMBER(5),  
8     hire_date  DATE,  
9     deptid     NUMBER(2),  
10    CONSTRAINT emp_test_deptid_ck  
11           CHECK (deptid BETWEEN 10 AND 99),  
12    CONSTRAINT emp_test_deptid_fk FOREIGN KEY (deptid)  
13           REFERENCES dept_test(deptid));
```

# 계약조건 수정 및 보기



## 제약조건 수정 및 보기

## 1. 제약조건 수정

## 1.1 제약조건 추가

- ☐ 제약조건을 추가, 삭제, 활성화 및 비활성화는 할 수 있지만 구조를 수정할 수는 없음
- ☐ **ALTER TABLE** 문에 **ADD** 절을 사용하여 기존 테이블에 제약조건 추가 가능
- ☐ **ALTER TABLE** 문에 **MODIFY** 절을 사용하여 **NOT NULL** 제약조건 추가 가능
  - 테이블이 비어 있거나 기존 열의 모든 행에 값이 있는 경우에만 **NOT NULL** 제약 조건 추가 가능



## 1.1 제약조건 추가 (계속)

### ☐ 제약조건 추가 구문

```
ALTER TABLE table
```

```
ADD [CONSTRAINT constraint] type (column);
```

```
SQL> ALTER TABLE emp_test
```

```
2 ADD CONSTRAINT emp_test_mgr_fk FOREIGN KEY (mgr)
```

```
3 REFERENCES emp (empno);
```

테이블이 변경되었습니다.

```
SQL> ALTER TABLE emp_test
```

```
2 MODIFY (duty VARCHAR2(9) NOT NULL);
```

테이블이 변경되었습니다.



## 1.2 제약조건 삭제

- ❑ 제약조건을 경우 수정할 수 없기 때문에 기존의 제약조건을 삭제하고 다시 추가
- ❑ **ALTER TABLE** 문을 **DROP** 절과 함께 사용하여 제약조건 삭제
- ❑ **PRIMARY KEY, UNIQUE** 제약조건을 삭제할 경우 **UNIQUE** 인덱스도 삭제
- ❑ **DROP** 절의 **CASCADE** 옵션을 사용하면 종속 제약조건을 모두 삭제



## 1.2 제약조건 삭제 (계속)

### ☐ 제약조건 삭제 구문

```
ALTER TABLE table  
DROP PRIMARY KEY | UNIQUE (column) |  
CONSTRAINT constraint [CASCADE];
```

```
SQL> ALTER TABLE emp_test  
2 DROP CONSTRAINT emp_test_mgr_fk;
```

```
SQL> ALTER TABLE dept_test  
2 DROP PRIMARY KEY CASCADE;
```

## 2. 제약조건 활성화 및 비활성화

### 2.1 제약조건 비활성화

- ❑ ALTER TABLE 문의 **DISABLE** 절을 실행하여 무결성 제약조건을 비활성화
- ❑ CASCADE 옵션을 적용하여 종속 무결성 제약조건을 비활성화
- ❑ UNIQUE 또는 PRIMARY KEY 제약조건을 비활성화하면 인덱스 제거
- ❑ 제약조건 비활성화 구문

```
ALTER TABLE table  
DISABLE CONSTRAINT constraint [CASCADE];
```



## 2.1 제약조건 비활성화 (계속)

```
SQL> ALTER TABLE dept_test  
2 ADD CONSTRAINT dept_test_deptid_pk PRIMARY KEY(deptid);
```

```
SQL> ALTER TABLE emp_test  
2 ADD CONSTRAINT emp_test_deptid_fk FOREIGN KEY(deptid)  
3 REFERENCES dept_test(deptid);
```

```
SQL> ALTER TABLE dept_test  
2 DISABLE CONSTRAINT dept_test_deptid_pk CASCADE;
```

테이블이 변경되었습니다.

## 2.2 제약조건 활성화

- ☐ 현재 테이블 정의에서 비활성화 되어 있는 무결성 제약 조건을 활성화 하려면 **ENABLE** 절을 사용
- ☐ **UNIQUE** 또는 **PRIMARY KEY** 제약조건을 활성화하면 **UNIQUE** 또는 **PRIMARY KEY** 인덱스가 자동으로 생성
- ☐ **CASCADE** 옵션으로 비활성화했던 **PRIMARY KEY** 제약조건을 다시 활성화 할 경우 해당 **PRIMARY KEY**에 종속된 **FOREIGN KEY**는 활성화 안됨
- ☐ 제약조건 활성화 구문

```
ALTER TABLE table  
ENABLE CONSTRAINT constraint;
```

```
SQL> ALTER TABLE dept_test  
2 ENABLE CONSTRAINT dept_test_deptid_pk;
```

### 3. 제약조건 보기

- ☐ USER\_CONSTRAINTS 뷰를 통해 모든 제약조건의 정의 및 이름 확인

```
SQL> SELECT constraint_name, constraint_type,  
2          search_condition  
3 FROM    user_constraints;
```

- 제약조건 유형 (constraint\_type) 에서 C는 CHECK 및 NOT NULL, P는 PRIMARY KEY, R은 FOREIGN KEY, U는 UNIQUE 제약조건을 의미

- ☐ USER\_CONS\_COLUMNS 뷰를 통해 제약조건의 이름 및 연관된 열 확인

```
SQL> SELECT constraint_name, column_name  
2 FROM    user_cons_columns;
```