

제약조건이란?

제약조건 종류

제약조건 관리




- 테이블의 해당 컬럼에 사용자가 원치 않은 데이터가 입력, 수정, 삭제되는 것을 방지하기 위한 조건입니다.
- 제약조건에 종속성이 존재할 경우 테이블의 삭제를 방지합니다.
- 다음의 제약조건 유형이 오라클에서 유효합니다.

- NOT NULL
- UNIQUE KEY
- PRIMARY KEY
- FOREIGN KEY
- CHECK

제약조건 조회는...
SELECT * FROM user_constraints;

| 제약조건 | 설명 |
|-------------|--|
| NOT NULL | 이열은 null 값을 포함하지 않음을 지정합니다. |
| UNIQUE KEY | 테이블의 모든 행에 대해 유일해야 하는 값을 가진 열 또는 열의 조합을 지정합니다. |
| PRIMARY KEY | 유일하게 테이블의 각 행을 식별합니다. NOT NULL+ UNIQUE KEY |
| FOREIGN KEY | 열과 참조된 테이블의 열 사이의 외래키 관계를 적용하고 설정합니다. |
| CHECK | 참이어야 하는 조건을 지정합니다. |

- 
- 테이블 레벨 제약조건을 정의할 수 있습니다.
 - 열 레벨 제약조건을 정의할 수 있습니다.

다음 구문은 테이블 생성 중에 제약조건을 정의하는 구문입니다.

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

- 열별로 제약조건을 정의합니다.
- 무결성 제약조건의 어떤 유형도 정의 가능합니다.

테이블을 정의할 때 열을 정의하는 곳에 제약조건을 명시할 수 있습니다. 다음 구문은 열 레벨 제약조건을 정의하기 위한 구문입니다.

```
column datatype [DEFAULT expr]  
                [CONSTRAINT constraint_name] constraint_type,
```

다음 구문은 아래 조건에 맞는 테이블을 생성합니다.

- 직원번호, 이름, 급여, 부서번호를 저장하는 테이블을 생성해야 합니다.
- 직원번호는 PK, 이름은 NN, 급여는 10000이하, 부서번호는 departments 테이블을 참조해서 저장합니다.
- 테이블 이름과 열 이름, 그리고 열의 타입은 임의로 구성합니다.

```
SQL> CREATE TABLE emp4 (
  2   empno  number(4)  (CONSTRAINT emp4_empno_pk) PRIMARY KEY,
  3   ename  varchar2(10) NOT NULL,
  4   sal    number(7,2) CONSTRAINT emp4_sal_ck CHECK(sal<=10000),
  5   deptno number(2)  CONSTRAINT emp4_deptno_dept_deptid_fk
  6           FK      REFERENCES departments(department_id)
  7 );
```

Table EMP4이(가) 생성되었습니다.

- 하나 이상의 열을 참조하고, 테이블의 열 정의와는 개별적으로 정의합니다.
- NOT NULL 제약조건은 정의할 수 없습니다.

테이블을 정의할 때 모든 열들의 이름과 타입을 명시한 다음 테이블 정의 구문의 맨 아래에 제약 조건들을 설정할 수 있습니다. 테이블 레벨 제약조건을 설정할 때에는 열 이름을 명시해야 합니다. NOT NULL 제약조건은 테이블 레벨 제약조건을 사용할 수 없습니다.

```
...  
column datatype [DEFAULT expr], --테이블의 마지막 열  
    [CONSTRAINT constraint_name] constraint_type (column, ...),  
    ...  
);
```

다음 구문은 아래 조건에 맞는 테이블을 생성합니다.

- 사원번호, 이름, 급여, 부서번호를 저장하는 테이블을 생성해야 합니다.
- 사원번호는 PK, 이름은 NN, 급여는 10000이하, 부서번호는 departments 테이블을 참조해서 저장합니다.
- 테이블 이름과 열 이름, 그리고 열의 타입은 임의로 구성합니다.

```
SQL> CREATE TABLE emp5 (  
2     empno number(4),  
3     ename varchar2(10) NOT NULL,  
4     sal number(7,2),  
5     deptno number(2),  
6     CONSTRAINT emp5_empno_pk PRIMARY KEY (empno),  
7     CONSTRAINT emp5_sal_ck CHECK(sal<=10000),  
8     CONSTRAINT emp5_deptno_dept_deptid_fk  
9         FOREIGN KEY (deptno) REFERENCES departments(department_id)  
10 );
```

Table EMP5이(가) 생성되었습니다.

제약조건이란?

제약조건 종류

제약조건 관리



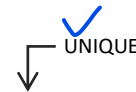
Null 값을 가질 수 없습니다.

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
|-------------|------------|------------|----------|--------------------|------------|------------|--------|----------------|------------|---------------|
| 100 | Steven | King | SKING | 515.123.4567 | 2003-06-17 | AD_PRES | 24000 | | | 90 |
| 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 | 2005-09-21 | AD_VP | 17000 | | 100 | 90 |
| 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 2001-01-13 | AD_VP | 17000 | | 100 | 90 |
| 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 | 2006-01-03 | IT_PROG | 9000 | | 102 | 60 |
| 104 | Bruce | Ernst | BERNST | 590.423.4568 | 2007-05-21 | IT_PROG | 6000 | | 103 | 60 |
| 105 | David | Austin | DAUSTIN | 590.423.4569 | 2005-06-25 | IT_PROG | 4800 | | 103 | 60 |
| 106 | Valli | Pataballa | VPATABAL | 590.423.4560 | 2006-02-05 | IT_PROG | 4800 | | 103 | 60 |
| 107 | Diana | Lorentz | DLORENTZ | 590.423.5567 | 2007-02-07 | IT_PROG | 4200 | | 103 | 60 |
| 108 | Nancy | Greenberg | NGREENBE | 515.124.4569 | 2002-08-17 | FI_MGR | 12008 | | 101 | 100 |
| 109 | Daniel | Faviet | DFAVIET | 515.124.4169 | 2002-08-16 | FI_ACCOUNT | 9000 | | 108 | 100 |
| 110 | John | Chen | JCHEN | 515.124.4269 | 2005-09-28 | FI_ACCOUNT | 8200 | | 108 | 100 |
| 111 | Ismael | Sciarra | ISCIARRA | 515.124.4369 | 2005-09-30 | FI_ACCOUNT | 7700 | | 108 | 100 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 176 | Jonathon | Taylor | JTAYLOR | 011.44.1644.429265 | 2006-03-24 | SA_REP | 8600 | 0.2 | 149 | 80 |
| 177 | Jack | Livingston | JLIVINGS | 011.44.1644.429264 | 2006-04-23 | SA_REP | 8400 | 0.2 | 149 | 80 |
| 178 | Kimberely | Grant | KGRANT | 011.44.1644.429263 | 2007-05-24 | SA_REP | 7000 | 0.15 | 149 | |
| 179 | Charles | Johnson | CJOHNSON | 011.44.1644.429262 | 2008-01-04 | SA_REP | 6200 | 0.1 | 149 | 80 |
| 180 | Winston | Taylor | WTAYLOR | 650.507.9876 | 2006-01-24 | SH_CLERK | 3200 | | 120 | 50 |
| 181 | Jean | Fleaur | JFLEAUR | 650.507.9877 | 2006-02-23 | SH_CLERK | 3100 | | 120 | 50 |
| 182 | Martha | Sullivan | MSULLIVA | 650.507.9878 | 2007-06-21 | SH_CLERK | 2500 | | 120 | 50 |
| 183 | Girard | Geoni | GGEONI | 650.507.9879 | 2008-02-03 | SH_CLERK | 2800 | | 120 | 50 |
| 184 | Nandita | Sarchand | NSARCHAN | 650.509.1876 | 2004-01-27 | SH_CLERK | 4200 | | 121 | 50 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 203 | Susan | Mavris | SMAVRIS | 515.123.7777 | 2002-06-07 | HR_REP | 6500 | | 101 | 40 |
| 204 | Hermann | Baer | HBAER | 515.123.8888 | 2002-06-07 | PR_REP | 10000 | | 101 | 70 |
| 205 | Shelley | Higgins | SHIGGINS | 515.123.8080 | 2002-06-07 | AC_MGR | 12008 | | 101 | 110 |
| 206 | William | Gietz | WGIETZ | 515.123.8181 | 2002-06-07 | AC_ACCOUNT | 8300 | | 205 | 110 |

- 값들이 유일해야 합니다. 열에서 두 개 이상 행이 같은 값을 가질 수 없습니다.
- Null 값을 가질 수 있습니다.

EMPLOYEES

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | ... |
|-------------|------------|-----------|----------|-----|
| 100 | Steven | King | SKING | ... |
| 101 | Neena | Kochhar | NKOCHHAR | ... |
| 102 | Lex | De Haan | LDEHAAN | ... |
| 103 | Alexander | Hunold | AHUNOLD | ... |
| 104 | Bruce | Ernst | BERNST | ... |
| 105 | David | Austin | DAUSTIN | ... |
| 106 | Valli | Pataballa | VPATABAL | ... |
| 107 | Diana | Lorentz | DLORENTZ | ... |
| 108 | Nancy | Greenberg | NGREENBE | ... |
| 109 | Daniel | Faviet | DFAVIET | ... |
| 110 | John | Chen | JCHEN | ... |
| 111 | Ismael | Sciarra | ISCIARRA | ... |
| ... | ... | ... | ... | ... |



INSERT INTO ...

| | | | | |
|-----|-------|------|-------|-----|
| 300 | Sue | King | SKING | ... |
| 310 | Jeong | Heo | JHEO | ... |

EMAIL 열의 값 SKING은 이미 존재합니다. 입력 할 수 없습니다

← 입력 가능합니다.



- 값들이 유일해야 합니다. 열에서 두 개 이상 행이 같은 값을 가질 수 없습니다.
- Null 값을 가질 수 없습니다.

PRIMARY KEY

EMPLOYEES

| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | ... |
|-------------|------------|-----------|----------|-----|
| 100 | Steven | King | SKING | ... |
| 101 | Neena | Kochhar | NKOCHHAR | ... |
| 102 | Lex | De Haan | LDEHAAN | ... |
| 103 | Alexander | Hunold | AHUNOLD | ... |
| 104 | Bruce | Ernst | BERNST | ... |
| 105 | David | Austin | DAUSTIN | ... |
| 106 | Valli | Pataballa | VPATABAL | ... |
| 107 | Diana | Lorentz | DLORENTZ | ... |
| 108 | Nancy | Greenberg | NGREENBE | ... |
| 109 | Daniel | Faviet | DFAVIET | ... |
| 110 | John | Chen | JCHEN | ... |
| 111 | Ismael | Sciarra | ISCIARRA | ... |
| ... | ... | ... | ... | ... |

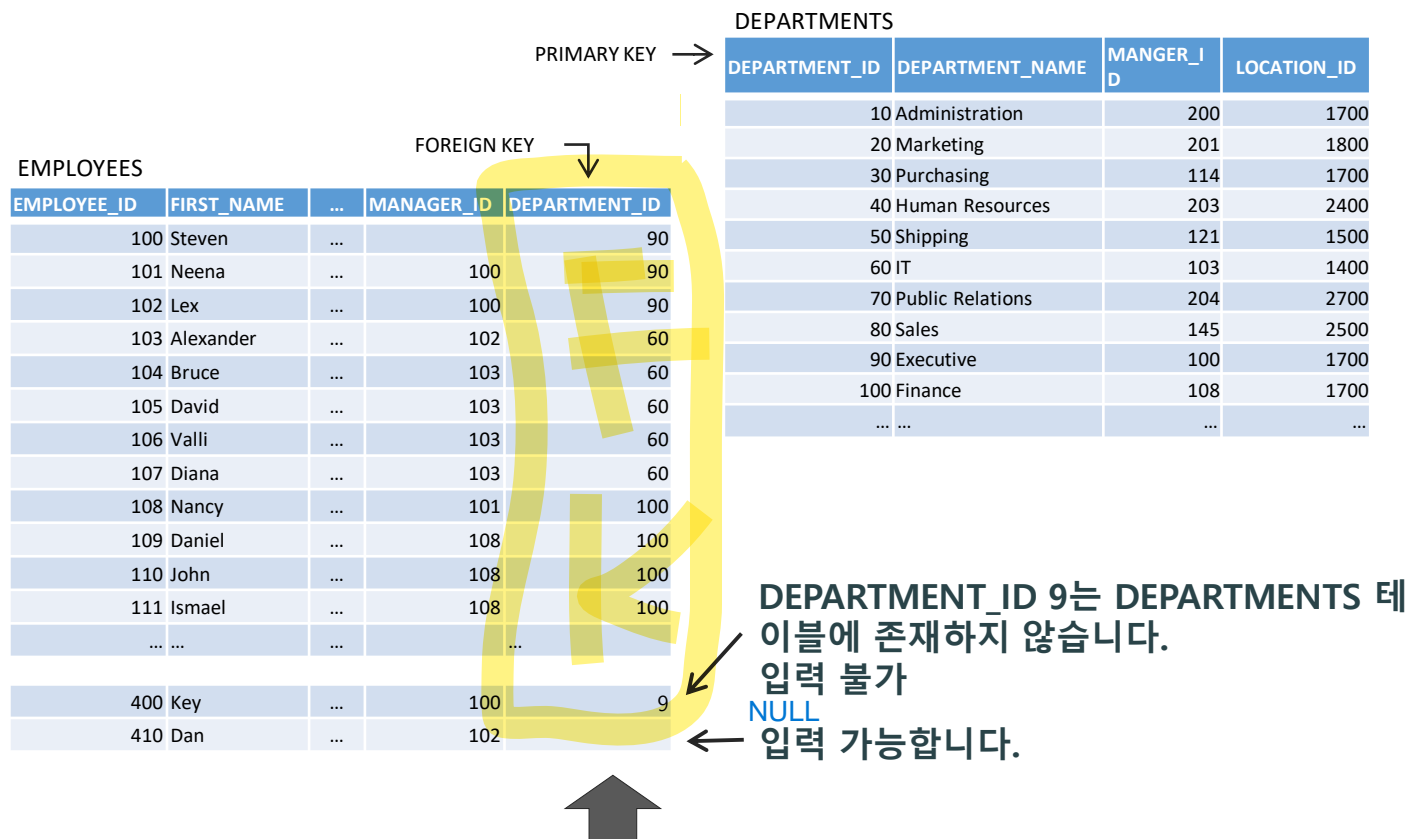
INSERT INTO ...

| | | | | |
|-----|------|------|-------|-----|
| 100 | Eric | Hyun | EHYUN | ... |
| | Dan | June | DJUNE | ... |

사원번호 100은 이미 존재하기 때문에 입력이 안됩니다.

EMPLOYEE_ID 열은 NULL값 입력이 안됩니다.

테이블 간의 관계를 설정합니다.



- 각 행을 만족해야 하는 조건을 정의합니다.

CHECK 제약조건은 각 행이 만족해야 하는 조건을 정의합니다. 조건은 다음의 경우만 제외하고는 질의 조건과 동일한 구성을 사용할 수 있습니다.

- CURRVAL, NEXTVAL, LEVEL, ROWNUM 등 pseudo-column 에 대한 참조
- SYSDATE, UID, USER, USERENV 함수에 대한 호출
- 다른 행의 다른 값을 참조하는 질의

단일 열은 복수 CHECK 제약조건을 가질 수 있습니다. 열에서 정의할 수 있는 CHECK 제약조건의 수에 대한 한계는 없습니다. CHECK 제약조건은 열 레벨 또는 테이블 레벨에서 정의될 수 있습니다.

CREATE TABLE test (where 조건에 들어가는 형식이라고 생각하면 됩니다

id varchar2(30) NOT NULL PRIMARY KEY,
col varchar2(1) CHECK (col IN ('A', 'B', 'C'))

);

CHECK (col < 10)
CHECK (col = 'ABCDE')
CHECK (col BETWEEN 5 AND 6)
CHECK (col LIKE '_A' OR col LIKE 'A%')

제약조건이란?

제약조건 종류

제약조건 관리



- 제약 조건의 추가 또는 삭제는 가능합니다. 그러나 수정을 불가능합니다.
- 제약조건의 활성화 또는 비활성화는 가능합니다.
- MODIFY 절을 사용하여 NOT NULL 조건을 추가할 수 있습니다.

제약조건 추가

```
ALTER TABLE table_name
ADD [CONSTRAINT constraint_name] type (column);
```

제약조건 삭제

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

문제 1.

| M_NAME | M_NUM | REG_DATE | GENDER | LOCA |
|--------|-------|------------|--------|------|
| AAA | 1 | 2018-07-01 | M | 1800 |
| BBB | 2 | 2018-07-02 | F | 1900 |
| CCC | 3 | 2018-07-03 | M | 2000 |
| DDD | 4 | 오늘날짜 | M | 2000 |

다음과 같은 테이블을 생성하고 데이터를 insert하세요 (커밋)

조건) M_NAME 는 가변문자형, 널값을 허용하지 않음

조건) M_NUM 은 숫자형, 이름(mem_memnum_pk) primary key

조건) REG_DATE 는 날짜형, 널값을 허용하지 않음, 이름:(mem_regdate_uk) UNIQUE키

조건) GENDER 가변문자형

조건) LOCA 숫자형, 이름:(mem_loca_loc_locid_fk) foreign key – 참조 locations테이블(location_id)

문제 2.

MEMBERS테이블과 LOCATIONS테이블을 INNER JOIN 하고 m_name, m_mum, street_address, location_id 컬럼만 조회

m_num기준으로 오름차순 조회