

9장 테이블 생성과 관리

□ 목 적

- 주된 데이터베이스 객체를 기술
- 테이블 생성
- 열 정의를 명시할 때, 사용 가능한 데이터형을 기술
- 테이블 정의를 변경
- 테이블을 삭제, 이름 변경, 자르기

1. 데이터 베이스 객체

객 체	설 명
Table	행과 열로 구성된 기본적인 저장매체의 단위
View	하나 이상의 테이블로부터 데이터의 부분집합을 논리적으로 표현
Sequence	기본 키 값을 발생
Index	어떤 질의의 성능을 향상
Synonym	객체에 대체 이름을 부여

2. 이름 지정 규칙

- 문자로 시작해야 함.
- 문자 길이는 1~30 이어야 함.
- 오직 **A-Z, a-z, 0-9, _, \$** 그리고 **#**만을 포함할 수 있음.
 - **#**은 유일한 문자이지만 권장되지 않음.
- 동일한 사용자가 소유한 객체의 이름은 중복되어서는 안됨.
- 오라클 서버 예약어는 안됨.

주) 이름은 대소문자를 구분하지 않는다.

3. CREATE TABLE 문장

다음을 가져야 함.

- CREATE TABLE 권한
- 저장 영역

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

다음을 명시해야 함.

- 테이블 이름
- 열 이름, 열 데이터형 및 열 크기

4. 다른 사용자 테이블 참조

- 사용자의 스키마에 없는, 다른 사용자에게 속하는 테이블
- 테이블의 접두어로서 소유자의 이름을 사용해야 함.
 - ◆스키마는 객체의 모음.
 - ◆ 스키마 객체는 데이터베이스에서 바로 참조하는 논리적인 구조.
 - ◆ 스키마 객체는 테이블, 뷰, 동의어, 시퀀스, 내장프로시저, 인덱스, 클러스터 그리고 데이터베이스 링크를 포함.

5. DEFAULT 옵션

- 삽입 시에 열에 대한 디폴트 값 명시.

```
_hiredate DATE DEFAULT SYSDATE, _
```

#가능한 값은 리터럴 값, 표현식 또는 SQL 함수

#불가능한 값은 다른 열의 이름이나 의사열
((ex)NEXTVAL, CURRVAL)

#디폴트 데이터형은 열의 데이터형과 일치해야 함.

6. 테이블 생성

- 테이블 생성

```
SQL> CREATE TABLE dept
2          (deptno NUMBER(2),
3          dname  VARCHAR2(14),
4          loc    VARCHAR2(13));
Tbale created.
```

#테이블 생성 확인

```
SQL> DESCRIBE  dapt;
```

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VHARCHAR2(14)
LOC		VHARCHAR2(13)

7. 데이터 사전 질의

- 사용자가 소유한 테이블을 기술

```
SQL> SELECT *  
2 FROM user_tables;
```

- 사용자가 소유한 별개의 다른 객체 형을 보여줌.

```
SQL> SELECT DISTINCT object_type  
2 FROM user_objects;
```

- 사용자가 소유한 테이블, 뷰, 동의어, 시퀀스를 보여줌.

```
SQL> SELECT *  
2 FROM user_catalog;
```

8. 데이터 형

데이터형	설 명
VARCHAR2(size)	가변길이 문자 데이터
CHAR(size)	고정길이 문자 데이터
NUMBER(p, s)	가변길이 숫자 데이터
DATE	날짜와 시간 값
LONG	2기가 바이트까지의 가변길이 문자 데이터
CLOB	4기가 바이트까지의 단일 바이트 문자 데이터
RAW and LONG RAW	원시 이진 데이터
BLOB	4기가 바이트까지의 이진 데이터
BFILE	4기가 바이트까지의 외부 파일에 저장된 이진 데이터

9. 서브쿼리를 사용한 테이블 생성(1)

- # CREATE TABLE 문장과 AS subquery 옵션을 조합하여 테이블을 생성하고 행을 삽입함.

```
CREATE TABLE table  
    [column( , column.....)]  
As subquery;
```

- # 서브쿼리 열의 개수와 명시된 열의 개수를 일치시킴.
- # 열 이름과 디폴트 값을 가진 열을 정의

9. 서브쿼리를 사용한 테이블 생성(2)

```
SQL> CREATE TABLE dept30
  2  AS
  3      SELECT empno, ename, sal* 12 annsal, hiredate
  4      FROM emp
  5      WHERE deptno = 30;
Table created.
```

```
SQL> DESCRIBE dept30;
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
ANNSAL		NUMBER
HIREDATE		DATE

10. ALTER TABLE 문장

- 다음에 **ALTER TABLE** 문장 사용.
 - 새로운 열 추가
 - 기존 열 변경
 - 새로운 열에 대한 디폴트 값 정의

```
ALTER TABLE table  
ADD          (column datatype [DEFAULT expr]  
              [ , column datatype] .....);
```

```
ALTER TABLE table  
MODIFY       (column datatype [DEFAULT expr]  
              [ , column datatype] .....);
```

11. 열 추가

- **ADD** 절을 사용하여 열을 추가함.

```
SQL> ALTER TABLE datp30  
2  ADD ( job VARCHAR2(9) );  
Table altered.
```

새로운 열이 마지막 열이 됨.

EMPNO	ENAME	ANNSAL	HIREDATE	JOB
7798	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	

...
6 rows selected.

12. 열 수정

- 열의 데이터형, 크기, 디폴트 값을 변경.

```
ALTER TABLE datp30  
MODIFY      (ename VARCHAR2(15));  
Table altered.
```

- ✦ 숫자 열의 정밀도나 폭을 증가시킴.
- ✦ 열이 오직 null 값만을 포함하거나 테이블에 행이 없으면 열의 폭을 감소시킴.
- ✦ 열이 null 값을 포함하면 데이터형을 변경시킴.
- ✦ 열이 null 값을 포함하거나 크기를 변경하지 않으면 CHAR 열을 VARCHAR2 데이터형으로 변경하거나 VARCHAR2 열을 CHAR 데이터형으로 변경.
- ✦ 열의 디폴트 값을 변경시키는 것은 오직 테이블에 가해지는 이후의 삽입에만 영향을 미침.

13. 테이블 삭제(1)

- # 테이블의 모든 데이터와 구조가 삭제.
- # 어떤 결정되지 않은 트랜잭션이 커밋.
- # 모든 인덱스가 삭제.
- # 이 문장은 롤백할 수 없음.

```
SQL> DROP TABLE datp30;  
Table dropped.
```


13. 테이블 삭제(2)

- **TRUNCATE TABLE 문장:**
 - 테이블의 모든 행을 삭제.
 - 해당 테이블에 사용한 기억 공간을 해제.

```
SQL>TRUNCATE TABLE department;  
Table truncated.
```

- # TRUNCATE를 사용하여 삭제한 행을 롤백 할 수 없음.
- # 대안적으로 DELETE 문장을 사용하여 행을 삭제함.
 - ◆ DELETE 문장은 테이블의 모든 행을 삭제할 수 있지만, 저장공간을 해제할 수는 없다.

14. 객체 이름 변경

테이블 이름, 뷰, 시퀀스 또는 동의어를 변경하기 위해, **RENAME** 문장을 실행.

```
SQL> RENAME dept TO ddepartment;  
Table renamed.
```

객체의 소유자 이어야 함.

15. 테이블에 주석문 추가

COMMENT 문장을 사용하여 테이블이나 열에 주석문을 추가할 수 있음.

```
SQL> COMMENT ON TABLE emp  
2 IS 'Employee Information';  
Comment created.
```

주석문은 데이터 뷰를 통해서 볼 수 있음.

- ALL_COL_COMMENTS
- USER_COL_COMMENTS
- ALL_TAB_COMMENTS
- USER_TAB_COMMENTS

제약조건이란?

- 오라클 서버는 부적합한 데이터가 테이블에 삽입되는 것을 방지하기 위해 **constraint** 를 사용
 - 그 테이블에서 행이 삽입, 갱신, 삭제될 때 마다 테이블에서 규칙을 적용
 - 다른 테이블에 종속성이 있다면 테이블의 제거를 방지
 - **Developer/200**같은 오라클 툴들에 대한 규칙을 제공
- 데이터 무결성 제약 조건

제약조건	기 술
NOT NULL	이 열은 null 값을 포함하지 않음을 지정
UNIQUE KEY	테이블의 모든 행에 대해 유일해야 하는 값을 가진 열 또는 열의 조합을 지정
PRIMARY KEY	유일하게 테이블의 각 행을 식별
FOREIGN KEY	열과 참조된 테이블의 열 사이의 외래키 관계를 적용하고 설정
CHECK	참이어야 하는 조건을 지정

제약조건 지침

- 사용자가 제약조건을 명명하거나 오라클 서버는 **SYS_Cn** 포맷을 사용하여 이름을 생성
- 제약조건 생성
 - 테이블이 생성되는 시간과 동일
 - 테이블이 생성된 후에 정의될 수 있음
- 열 또는 테이블 레벨에서 제약조건을 정의
- **USER_CONSTRAINTS** 데이터 사전 뷰에서 지정 테이블에 대해 정의된 제약조건을 봄

제약조건 정의

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

- **Schema** 소유자 명과 동일
- **Table** 테이블의 이름
- **DEFAULT expr** 값이 **INSERT** 문장에서 생략되면 디폴트 값을 지정
- **Column** 열의 이름
- **Datatype** 열의 데이터 유형과 길이
- **Column_constraint** 열 정의의 일부로서 무결성 제약조건
- **Table_constraint** 테이블 정의의 일부로서 무결성 제약 조건

제약조건 정의

- 열 제약조건 레벨
 - 열별로 정의 무결성 제약조건의 어떤 유형도 정의 가능

```
Column [CONSTRAINT constraint_name] constraint_type
```

■ 테이블 제약조건 레벨

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

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

NOT NULL 제약조건

- 열에 대해 **null** 값이 허용되지 않도록 보증
- **NOT NULL** 제약조건이 없는 열은 디폴트로 **null** 값을 포함할 수 있음
- 테이블 레벨이 아닌 열 레벨에서 정의

```
SQL> CREATE TABLE emp (  
2         empno    NUMBER(4),  
3         ename    VARCHAR2(10) NOT NULL,  
4         job      VARCHAR2(9),  
5         mgr      NUMBER(4),  
6         hiredate DATE,  
7         sal      NUMBER(7,2),  
8         comm     NUMBER(7,2),  
9         deptno   NUMBER(7,2) NOT NULL ;
```


UNIQUE KEY 제약조건

- **UNIQUE key** 무결성 제약조건은 열 또는 열의 집합의 모든 테이블 값들이 유일해야 함을 요구 즉 지정된 열 또는 열의 집합에서 중복 값을 가지는 테이블의 두 행은 없다
- **UNIQUE key** 제약조건을 정의에 포함된 열은 **unique key**라 부름
- **UNIQUE key**가 하나이상의 열을 포함한다면, 열 그룹은 **composite unique key**라 부름
- **UNIQUE key** 제약조건은 동일 열에 대해 **NOT NULL** 제약조건을 정의하지 않으면 **null**의 출력을 허용
 - 행은 **null**이 어느 것과도 동일하게 여겨지지 않기 때문에 **NOT NULL** 제약조건이 없으면 열에 대해 **null**을 포함할 수 있다
 - 열에서 **null**은 항상 **UNIQUE key** 제약조건을 만족함

UNIQUE key 제약조건

- 열 또는 테이블 레벨에서 정의될 수 있음
- 조합 유일 키는 테이블 레벨 정의를 사용하여 생성됨

```
SQL> CREATE TABLE dept (  
2         deptno NUMBER(2),  
3         dname  VARCHAR2(14),  
4         loc    VARCHAR2(13),  
5         CONSTRAINT dept dname uk UNIQUE(dname));
```

PRIMARY Key 제약조건

- **PRIMARY Key** 제약조건은 테이블에 대해 생성될 수 있음
- **PRIMARY Key** 제약조건은 테이블에서 각 행을 유일하게 식별하는 열 또는 열의 집합
- 이 제약조건은 열 또는 열의 집합의 유일성을 요구하고 **null** 값을 포함할 수 없음을 보증함

PRIMARY Key 제약조건

- 열 레벨 또는 테이블 레벨에서 정의될 수 있음
- 조합 **PRIMARY Key**는 테이블 정의 레벨을 사용하여 생성됨

```
SQL> CREATE TABLE dept (  
2         deptno  NUMBER(2),  
3         ename   VARCHAR2(14),  
4         loc     VARCHAR2(13),  
5         CONSTRAINT dept_dname_uk UNIQUE(dname),  
6         CONSTRAINT dept_deptno_pk PRIMARY Key(deptno));
```

FOREIGN Key 제약조건

- **FOREIGN Key**, 또는 참조 무결성 제약조건은 알 또는 열의 집합을 **foreign key**로 지정하여 동일 테이블 또는 다른 테이블 간의 기본 키 또는 유일 키 사이의 관계를 설정
- 오래 키 값은 부모 테이블에서 존재하는 값과 일치해야 하거나 **NULL**이 되어야 함
- 왜래 키는 데이터 값을 기초로 하여 순전히 논리적이거나 물리적이거나 포인터가 아님

FOREIGN Key 제약조건

- 열 또는 테이블 제약조건 레벨에서 정의될 수 있음
- 조합 왜래 키는 테이블 레벨 정의를 사용하여 생성됨

```
SQL> CREAT TABLE emp (  
2      empno    NUMBER(4),  
3      ename     VARCHAR2(10) NOT NULL,  
4      job       VARCHAR2(9),  
5      mgr       NUMBER(4),  
6      hiredate  DATE,  
7      sal       NUMBER(7,2),  
8      comm      NUMBER(7,2),  
9      deptno    NUMBER(7,2) NOT NULL  
10     CONSTRAINT emp_deptno_fk FOREIGN key (deptno)  
11           REFERENCES dept (DEPTNO));
```

FOREIGN Key 제약조건 키워드

- **FOREIGN Key**
 - 테이블 제약조건 레벨에서 자식 테이블 열을 정의함
- **REFERENCES**
 - 부모 테이블에서 테이블과 열을 식별함
- **ON DELETE CASCADE**
 - 자식 테이블에서 행의 종속을 삭제하고 부모 테이블에서 삭제를 허용함

CHECK 제약 조건

- 각 행이 만족해야 하는 조건을 정의
- 조건은 다음의 경우만을 제외하고 질의 조건과 동일한 구성을 사용할 수 있음
 - **CURRVAL, NEXTVAL, LEVEL, ROWNUM, pseudocolumn** 에 대한 참조
 - **SYSDATE, UID, USER, USERENV** 함수에 대한 호출
 - 다른 행에 있는 다른 값을 참조 하는 질의
- 단일 열은 복수 **CHECK** 제약 조건을 가질 수 없다
- 열에서 정의할 수 있는 **CHECK** 제약조건의 수에 대한 한계는 없다

제약 조건 추가

```
ALTER TABLE table
```

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

- **Add** 절을 가지는 **ALTER TABLE** 문장을 사용하여 기존의 테이블에 대한 제약조건을 추가할 수 있음
- 제약조건을 추가 또는 삭제가능, 수정은 불가능
- 제약조건을 활성화 또는 비활성화
- **MODIFY** 절을 사용하여 **NOT NULL** 제약조건을 추가

제약조건 추가

- **EMP** 테이블 안에 유효한 종업원으로 이미 존재해야 하는 관리자를 나타내는 **EMP** 테이블에 대하여 **FOREIGN key** 제약조건을 추가할 수 있다.

```
SQL> ALTER TABLE      emp
      2      ADD CONSTRAINT emp_mgr_fk
      3      FOREIGN key(mgr) REPERENCES emp(empno);
Table altered.
```

제약조건 삭제

- 제약조건을 삭제하기 위해서 **USER_CONSTRAINTS** 와 **USER_CONS_COLUMNS** 데이터 사전 뷰에서 제약조건 이름을 식별할 수 있음.
- **DROP** 절과 **ALTER TABLE** 문장을 사용
- **DROP**의 **CASCADE** 옵션은 모든 종속적인 제약조건이 삭제되게 함

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

- Table : 테이블의 이름 - column : 제약조건에 의해 영향 받은 이름 - constraint : 제약조건의 이름

- ✦ 무결성 제약조건을 삭제할 때, 그 제약조건은 더 이상 오라클 서버에 의해 적용되지 않으며 데이터 사전에서 찾을 수 있다.

제약조건 사용불가

- 무결성 제약조건을 비활성화 하기 위해 **ALTER TABLE** 문장의 **DISABLE** 절을 실행
- 종속적인 무결성 제약조건을 비 활성화 하기위해서 **CASCADE** 옵션을 적용함

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

- Table : 테이블의 이름 - constraint : 제약조건의 이름

- ❏ CREATE TABLE 문장과 ALTER TABLE 문장으로 DISABLE 절을 사용할 수 있다.
- ❏ CASCADE 절은 종속적인 무결성 제약조건을 비활성화 함

제약조건의 사용가능

- **ENABLE** 절을 가진 **ALTER TABLE** 문장을 사용하여 삭제 또는 재 생성 없이 제약조건을 활성화 할 수 있음

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

- Table : 테이블의 이름 - constraint : 제약조건의 이름

- ⌘ 제약조건이 활성화 되면 그 제약조건은 테이블의 모든 데이터에 대해 적용됨
- ⌘ 테이블의 모든 데이터는 제약조건과 일치해야 함
- ⌘ UNIQUE key 또는 PRIMARY key 제약조건이 활성화 된다면, UNIQUE 또는 PRIMARY key 인덱스는 자동으로 생성됨.
- ⌘ CREATE TABLE 문장과 ALTER TABLE 문장으로 ENABLE 절을 사용할 수 있다.

제약조건 보기

- 모든 제약조건의 정의와 이름을 보기위해 USER_CONSTRAINTS 테이블을 질의함

```
SQL> SELECT      constraint_name, constraint_type,  
2                search_condition  
3      FROM      user_constraints  
4      WHERE      table_name = 'EMP';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION
SYS_C00674	C	EMPNO IS NOT NULL
SYS_C00675	C	DEPTNO IS NOT NULL
EMP_EMPNO_PK	P	
...		

제약조건과 연관된 열 보기

USER_CONS_COLUMNS 뷰에서 제약조건 이름과 관련된 열을 봄

```
SQL> SELECT      constraint_name, column_name
  2      FROM      user_con_columns
  3      WHERE      table_name = 'EMP';
```

CONSTRAINT_NAME	COLUMN_NAME
EMP_DEPTNO_FK	DEPTNO
EMP_EMPNO_PK	EMPNO
EMP_MGR_FK	MGR
SYS_C00674	EMPNO
SYS_C00675	DEPTNO

요약

- 다음 제약조건 유형을 생성함
 - **NOT NULL**
 - **UNIQUE Key**
 - **PRIMARY Key**
 - **FOREIGN Key**
 - **CHECK**
- 모든 제약조건 정의와 이름을 보기 위해 **USER_CONSTRAINTS** 테이블을 질의합니다.

4-5) 행 갱신 : 무결성 제약조건 에러

```
SQL> UPDATE emp  
2> SET      deptno = 55  
3> WHERE    deptno = 10;
```

```
UPDATE emp  
      *  
ERROR at line  
ORA-02291 : integrity constraint ( USR.EMP_DEPTNO_FK)  
Violated parent key not found.
```

Department number 55 does not exist.

5-4) 행 삭제 : 무결성 제약조건 에러

```
SQL> DELETE FROM dept  
2> WHERE deptno = 10;
```

```
DELETE FROM dept
```

*

ERROR at line 1 :

ORA-02292 : integrity constraint (USR.EMP_DEPTNO_FK)
violated - failed record found.

You cannot delete a row that contains a primary key that is used as a foreign key in another table.

□ 요 약

문 장	설 명
CREATE TABLE	테이블을 생성
ALTER TABLE	테이블 구조를 변경
DROP TABLE	행과 테이블 구조를 삭제
RENAME	테이블 이름, 뷰, 시퀀스 또는 동의어 변경
TRUNCATE	테이블로부터 모든 행을 제거하고 저장공간을 해제
COMMENT	테이블이나 뷰에 주석문 추가