

# ORACLE® 14. 시퀀스

# 목차

1. 시퀀스 개념
2. 시퀀스 데이터 디렉터리
3. CURRVAL, NEXTVAL
4. 시퀀스 적용하기
5. 시퀀스 제거
6. 시퀀스 수정

# 1. 시퀀스 개념

- ❖ 오라클에서는 행을 구분하기 위해서 기본 키를 두고 있다. 기본 키는 중복된 값을 가질 수 없으므로 항상 유일한 값을 가져야 합니다.
- ❖ 기본 키가 유일한 값을 갖도록 사용자가 직접 값을 생성해내려면 부담이 클 것이다.
- ❖ 시퀀스는 테이블 내의 유일한 숫자를 자동으로 생성하는 자동 번호 발생기이므로 시퀀스를 기본 키로 사용하게 되면 사용자의 부담을 줄일 수 있다.

# 1.1 시퀀스 형식[1]

❖ 다음은 시퀀스 생성을 위한 기본 형식이다.

형식

**CREATE SEQUENCE *sequence\_name***

**[START WITH n]** ①

**[INCREMENT BY n]** ②

**[{MAXVALUE n | NOMAXVALUE}]** ③

**[{MINVALUE n | NOMINVALUE}]** ④

**[{CYCLE | NOCYCLE}]** ⑤

**[{CACHE n | NOCACHE}]** ⑥

❖ 1. START WITH

- 시퀀스 번호의 시작값을 지정할 때 사용된다. 만일 1부터 시작되는 시퀀스를 생성하려면 START WITH 1이라고 기술하면 된다.

❖ 2. INCREMENT BY

- 연속적인 시퀀스 번호의 증가치를 지정할 때 사용된다. 만일 1씩 증가하는 시퀀스를 생성하려면 INCREMENT BY 1이라고 기술하면 된다.

# 1.1 시퀀스 형식[2]

## ❖ 3. MAXVALUE n | NOMAXVALUE

- MAXVALUE는 시퀀스가 가질 수 있는 최대값을 지정한다. 만일 NOMAXVALUE를 지정하게 되면 ASCENDING 순서일 경우에는  $10^{27}$ 승이고 DESCENDING 순서일 경우에는 -1로 설정된다.

## ❖ 4. MINVALUE n | NOMINVALUE

- MINVALUE 은 시퀀스가 가질수 있는 최소값을 지정한다. 만일 NOMINVALUE을 지정하게 되면 ASCENDING 순서일 경우에는 1이고 DESCENDING 순서일 경우에는  $10^{27}$ 승으로 설정된다.

## ❖ 5. CYCLE | NOCYCLE

- CYCLE 은 지정된 시퀀스 값이 최대값까지 증가가 완료되게 되면 다시 START WITH 옵션에 지정한 시작 값에서 다시 시퀀스를 시작하도록 한다. NOCYCLE 은 증가가 완료되게 되면 에러가 발생한다.

## ❖ 6. CACHE n | NOCACHE

- CACHE 은 메모리상의 시퀀스 값을 관리하도록 하는 것인데 기본 값은 20이다. NOCACHE는 원칙적으로 메모리 상에서 시퀀스를 관리하지 않는다.

## 1.1 시퀀스 형식(3)

- ❖ 다음은 부서 번호를 자동으로 부여해주는 시퀀스 객체를 생성하는 문장이다.

**예**

```
CREATE SEQUENCE DEPT_DEPTNO_SEQ  
INCREMENT BY 10  
START WITH 10;
```

- ❖ 시작 값이 10이고 10씩 증가하는 시퀀스 DEPT\_DEPTNO\_SEQ를 생성한다.

## 2. 시퀀스 데이터 디렉터리

- ❖ 사용자가 작성한 객체들의 정보를 저장하고 있는 데이터 디렉터리의 이름을 보면 다음과 같은 규칙성이 있다.
- ❖ 테이블(TABLE) 객체에 대한 정보를 저장하는 데이터 디렉터리는 USER\_TABLES이고, 뷰(VIEW)객체에 대한 정보를 저장하는 데이터 디렉터리는 USER\_VIEWS이므로 이번 장에서 새로 배운 시퀀스(SEQUENCE)에 대한 자료사전의 이름은 USER\_SEQUENCES 이라고 생각할 수 있다.
- ❖ 생성된 시퀀스 객체에 대한 정보를 저장하는 데이터 디렉터리로는 USER\_SEQUENCES가 있다.

## 2.1 시퀀스 데이터 덱서너리 살펴보기(1)

- ❖ ex) SEQUENCE\_NAME은 시퀀스 객체의 이름을 저장하고 MIN\_VALUE는 최소값, MAX\_VALUE는 최대값, INCREMENT\_BY는 증가치에 대한 정보를 가지고 있으며, CYCLE\_FLAG는 CYCLE 옵션을 사용하는지, 하지 않는지에 대한 정보를 가지며 LAST\_NUMBER는 마지막 숫자 값을 가지고 있다.
- ❖ 1. 데이터 덱서너리 USER\_SEQUENCES로 현재 사용 중인 시퀀스 객체의 정보를 살펴보자.

**예**

```
SELECT SEQUENCE_NAME, MIN_VALUE, MAX_VALUE,  
INCREMENT_BY, CYCLE_FLAG  
FROM USER_SEQUENCES;
```



## 2.1 시퀀스 데이터 딥서치 살펴보기(2)

### 3. CURRVAL, NEXTVAL(1)

- ❖ 시퀀스의 현재 값을 알아내기 위해서 CURRVAL를 사용하고, 다음 값을 알아내기 위해서는 NEXTVAL를 사용한다.

CURRVAL : 현재 값을 반환한다.

NEXTVAL : 현재 시퀀스값의 다음 값을 반환한다

- ❖ CURRVAL에 새로운 값이 할당되기 위해서는 NEXTVAL로 새로운 값을 생성해야 한다.
- ❖ 즉, NEXTVAL로 새로운 값을 생성한 다음에 이 값을 CURRVAL에 대체하게 됩니다.

### 3. CURRVAL, NEXTVAL(2)

- ❖ NEXTVAL, CURRVAL을 사용할 수 있는 경우와 사용할 수 없는 경우를 살펴보자.
- ❖ NEXTVAL, CURRVAL을 사용할 수 있는 경우
  - 서브 쿼리가 아닌 SELECT 문
  - INSERT 문의 SELECT 문
  - INSERT 문의 VALUES 절
  - UPDATE 문의 SET 절
- ❖ NEXTVAL, CURRVAL을 사용할 수 없는 경우
  - VIEW의 SELECT 절
  - DISTINCT 키워드가 있는 SELECT 문
  - GROUP BY, HAVING, ORDER BY 절이 있는 SELECT 문
  - SELECT, DELETE, UPDATE의 서브 쿼리
  - CREATE TABLE, ALTER TABLE 명령의 DEFAULT 값

## 3.1 CURRVAL, NEXTVAL 사용하기

❖ ex) CURRVAL, NEXTVAL의 실제 사용 예를 살펴보자.

❖ 1. NEXTVAL로 새로운 값을 생성한다.

**예** `SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;`

❖ 2. 시퀀스의 현재 값을 알아내기 위해서 CURRVAL를 사용한다.

**예** `SELECT DEPT_DEPTNO_SEQ.CURRVAL FROM DUAL;`

## 4. 시퀀스 적용하기

- ❖ 시퀀스는 99.9%가 INSERT 연산과 같이 사용되어 컬럼 값을 자동으로 발생시키는 용도로 사용된다.
- ❖ 사원 테이블을 생성하면서 사원 번호를 기본 키로 설정하였다.
- ❖ 기본 키는 반드시 유일한 값을 가져야 한다. 사용자가 새로운 사원을 추가할 때마다 유일한 사원번호를 INSERT 해야 하는 번거로움이 있다.
- ❖ 사원 번호를 생성하는 시퀀스 객체를 사용하여 사원 번호가 자동 생성되도록 한다면 이러한 번거로움을 덜어줄 수 있다.

## 4.1 기본키에 시퀀스 적용(1)

- ❖ 사원 번호를 생성하는 시퀀스 객체를 생성하여 이를 기본 키인 사원 번호에 사용하여 사용자가 새로운 사원을 추가할 때마다 유일한 사원번호를 INSERT 해야 하는 번거로움을 줄여보자.
- ❖ 1. 시작 값이 1이고 1씩 증가하고, 최댓값이 100000이 되는 시퀀스 EMP\_SEQ 생성한다.

**예**

```
CREATE SEQUENCE EMP_SEQ  
START WITH 1  
INCREMENT BY 1  
MAXVALUE 100000;
```

## 4.1 기본키에 시퀀스 적용[2]

- ❖ 2. 이번에는 생성된 시퀀스를 사용하기 위해서 사원 번호를 기본 키로 설정하여 EMP01란 이름으로 새롭게 생성한다.

```
예 DROP TABLE EMP01;  
  
CREATE TABLE EMP01(  
EMPNO NUMBER(4) PRIMARY KEY,  
ENAME VARCHAR2(10),  
HIREDATE DATE  
);
```

- ❖ 3. 사원 번호를 저장하는 EMPNO 컬럼은 기본 키로 설정하였으므로 중복된 값을 가질 수 없다. 다음은 생성한 EMP\_SEQ 시퀀스로부터 사원번호를 자동으로 할당 받아 데이터를 추가하는 문장이다.

```
예 INSERT INTO EMP01  
VALUES(EMP_SEQ.NEXTVAL, '김씨', SYSDATE);
```

## 4.1 기본키에 시퀀스 적용(3)

- ❖ 4. 데이터를 추가하면서 EMP\_SEQ 시퀀스로부터 직원번호를 자동으로 할당받았는지 EMP01 테이블의 내용을 확인한다.

**예**      **SELECT \* FROM EMP01;**

EMPNO	ENAME	HIREDATE
1	김씨	21/01/10

- ❖ ex) 부서 테이블을 생성한다.

**예**      **CREATE TABLE DEPT\_EXAMPLE(  
DEPTNO NUMBER(4) PRIMARY KEY,  
DNAME VARCHAR2(15),  
LOC VARCHAR2(15)  
);**

- ❖ 부서 번호를 저장하는 DEPTNO 컬럼은 기본 키로 설정하였으므로 중복된 값을 가질 수 없다.



# 문제

1. 위에서 만든 부서 테이블(DEPT\_EXAMPLE)의 DEPTNO 컬럼에 유일한 값을 가질 수 있도록 시퀀스 객체를 생성(DEPT\_EXAMPLE\_SEQ)하고 데이터를 입력 하시오.

DEPTNO	DNAME	LOC
10	경리부	서울
20	인사부	인천
30	영업부	용인
40	전산부	수원

## 5. 시퀀스 제거

- ❖ DROP SEQUENCE문으로 시퀀스를 제거해보자.

**예** DROP SEQUENCE DEPT\_DEPTNO\_SEQ;

Sequence DEPT\_DEPTNO\_SEQ이 (가) 삭제되었습니다.

## 6. 시퀀스 수정[1]

- ❖ 시퀀스를 제거한 후 확인까지 해보았다.. 이번에는 시퀀스를 변경해보자.
- ❖ 시퀀스를 변경하려면 ALTER SEQUENCE 문을 사용해야 한다.

### 형식

```
ALTER SEQUENCE sequence_name  
[INCREMENT BY n]  
[{MAXVALUE n | NOMAXVALUE}]  
[{MINVALUE n | NOMINVALUE}]  
[{CYCLE | NOCYCLE}]  
[{CACHE n | NOCACHE}]
```

- ❖ ALTER SEQUENCE는 START WITH 절이 없다는 점을 빼고는 CREATE SEQUENCE와 구조가 동일하다.
- ❖ START WITH 옵션은 ALTER SEQUENCE를 써서 변경할 수 없다.
- ❖ 다른 번호에서 다시 시작하려면 이전 시퀀스를 삭제하고 다시 생성해야 한다.

## 6. 시퀀스 수정[2]

- ❖ ex) 이미 생성해서 사용하던 시퀀스의 최대값을 변경해 보자
- ❖ 1. 시퀀스는 최대값을 지정하지 않으면 기본적으로  $10^{27}$ 으로 지정된다. 사용자가 임의로 최대값을 지정할 수 있는데 MAXVALUE에 값을 지정하면 된다. 10부터 10씩 증가하면서 최대 30까지의 값을 갖는 시퀀스를 생성한다.

예

```
CREATE SEQUENCE DEPT_DEPTNO_SEQ  
START WITH 10  
INCREMENT BY 10  
MAXVALUE 30;
```

## 6. 시퀀스 수정[3]

- ❖ 2. 부서 번호를 계속 생성하다 보면 최대값을 넘게 된다. 최대값을 넘을 때까지 시퀀스를 생성해 본다.

예

```
SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;  
SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;  
SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;  
SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;
```

NEXTVAL
-----
10
NEXTVAL
-----
20
NEXTVAL
-----
30

명령의 4 행에서 시작하는 중 오류 발생 -

```
SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL
```

오류 보고 -

```
ORA-08004: sequence DEPT_DEPTNO_SEQ.NEXTVAL exceeds MAXVALUE and cannot be instantiated
```

- ❖ 이때 CYCLE 옵션을 지정하지 않으면 기본값으로 NOCYCLE을 갖게 되므로 오류가 발생하게 된다.

## 6. 시퀀스 수정(4)

- ❖ 3. ALTER SEQUENCE문을 사용하여 사용 중이던 DEPT\_DEPTNO\_SEQ 시퀀스의 최대값을 수정해본다.

```
예 ALTER SEQUENCE DEPT_DEPTNO_SEQ  
MAXVALUE 1000;
```

- ❖ 4. USER\_SEQUENCES 를 조회하면 시퀀스가 수정되었는지 확인할 수 있다.

```
예 SELECT SEQUENCE_NAME, MAX_VALUE, INCREMENT_BY,  
    CYCLE_FLAG  
    FROM USER_SEQUENCES;
```

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG
DEPT DEPTNO SEQ	1000	10	N
DEPT EXAMPLE SEQ	99999999999999999999999999999999	10	N
EMP SEQ	100000	1	N