

15장 시퀀스

15.1. 시퀀스의 개념 이해와 시퀀스 생성

- 오라클에서는 행을 구분하기 위해서 기본 키를 두고 있다. 기본 키는 중복된 값을 가질 수 없으므로 항상 유일한 값을 가져야 한다.
- 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.

```
-- 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

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

- 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.
- ③ MAXVALUE *n* | NOMAXVALUE : MAXVALUE 은 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^{26} 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. NOCYCLE은 증가가 완료되게 되면 에러를 유발시킨다.
- ⑥ CACHE *n* | NOCACHE : CACHE 은 메모리상의 시퀀스 값을 관리하도록 하는 것인데 기본 값은 20이다. NOCACHE는 원칙적으로 메모리 상에서 시퀀스를 관리하지 않는다.

15.2. 시퀀스 관련 데이터 덱서너리

- 생성된 시퀀스 객체에 대한 정보를 저장하는 데이터 덱서너리는 USER_SEQUENCES 이다.

```
DESC USER_SEQUENCES

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

15.3. 시퀀스 값을 알아보는 CURRVAL과 NEXTVAL

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

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

15.4. 시퀀스 실무에 적용하기

- 시퀀스는 99.9%가 INSERT 연산과 같이 사용되어 컬럼 값을 자동으로 발생시키는 용도로 사용된다.
- 아래 예는 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.

```
-- 예: 시퀀스를 테이블의 기본 키에 적용하기  
CREATE SEQUENCE EMP_SEQ  
START WITH 1  
INCREMENT BY 1  
MAXVALUE 100000 ;  
  
DROP TABLE EMP01;  
CREATE TABLE EMP01(  
EMPNO NUMBER(4) PRIMARY KEY,  
ENAME VARCHAR(10),  
HIREDATE DATE  
);  
  
INSERT INTO EMP01  
VALUES(EMP_SEQ.NEXTVAL, 'JULIA' , SYSDATE);  
  
SELECT * FROM EMP01;
```

15.5. 시퀀스 제거와 수정

15.5.1. 시퀀스를 삭제하는 방법

- 생성되어 있는 시퀀스를 제거하려면 DROP SEQUENCE문을 사용해야 한다.

```
SELECT SEQUENCE_NAME, MAX_VALUE, INCREMENT_BY, CYCLE_FLAG  
FROM USER_SEQUENCES;  
  
DROP SEQUENCE DEPT_DEPTNO_SEQ;
```

15.5.2. 시퀀스를 수정하는 방법

- 시퀀스를 변경하려면 ALTER SEQUENCE 문을 사용해야 한다.
- 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.

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

-- 예 : 시퀀스 최대값을 변경하기
DROP SEQUENCE DEPT_DEPTNO_SEQ;
CREATE SEQUENCE DEPT_DEPTNO_SEQ
START WITH 10
INCREMENT BY 10
MAXVALUE 30; -- 10부터 10씩 증가하면서 최대 30까지의 값을 갖는 시퀀스를 생성한다.

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; -- 부서 번호를 계속 생성하다 보면 최대값을 넘게 된다.

ALTER SEQUENCE DEPT_DEPTNO_SEQ
MAXVALUE 100; -- 최대값을 넘을 때까지 시퀀스를 생성한다.

SELECT SEQUENCE_NAME, MAX_VALUE, INCREMENT_BY, CYCLE_FLAG
FROM USER_SEQUENCES; -- USER_SEQUENCES 를 조회하면 시퀀스가 수정되었는지 확인할 수 있다.
```

[과제] 과제-15-01.TXT

1. 최소값 1, 최대값 99999999, 1000부터 시작해서 1씩 증가하는 ORDERS_SEQ 라는 시퀀스를 만들어보자.

<정답>