



06강.oracle

- 시퀀스

1 시퀀스

목차

1 시퀀스 개념 이해와 시퀀스 생성

2 시퀀스 관련 데이터 디렉터리

3 CURRVAL, NEXTVAL

4 시퀀스 실무에 적용하기

5 시퀀스 제거

6 시퀀스 수정

1 시퀀스

시퀀스 개념 이해와 시퀀스 생성

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

1 시퀀스

뷰의 기본 테이블

❖ 다음은 시퀀스를 생성하기 위한 기본 형식입니다.

```
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 사용됩니다. 만일 1부터 시작되는 시퀀스를 생성하려면 START WITH 1이라고 기술하면 됩니다.

② INCREMENT BY

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

1 시퀀스

뷰의 기본 테이블

③ MAXVALUE n | NOMAXVALUE

- MAXVALUE 은 시퀀스가 가질 수 있는 최대값을 지정합니다. 만일 NOMAXVALUE를 지정하게 되면 ASCENDING 순서일 경우에는 10^{27} 승이고 DESCENDING 순서일 경우에는 -1로 설정됩니다.

④ MINVALUE n | NOMINVALUE

- MINVALUE 은 시퀀스가 가질수 있는 최소값을 지정합니다. 만일 NOMINVALUE을 지정하게 되면 ASCENDING 순서일 경우에는 1 이고 DESCENDING 순서일 경우에는 10^{26} 승으로 설정됩니다.

⑤ CYCLE | NOCYCLE

- CYCLE 은 지정된 시퀀스 값이 최대값까지 증가가 완료되게 되면 다시 START WITH 옵션에 지정한 시작 값에서 다시 시퀀스를 시작하도록 합니다. NOCYCLE은 증가가 완료되게 되면 에러를 유발시킵니다.

⑥ CACHE n | NOCACHE

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

1 시퀀스

예제

```
CREATE SEQUENCE board_seq          --시퀀스이름 board_seq
INCREMENT BY 1                    --증감숫자 1
START WITH 1                      --시작숫자 1
MINVALUE 1                        --최소값 1
MAXVALUE 1000                     --최대값 1000
NOCYCLE                           --순환하지않음
CACHE;                            --메모리에 시퀀스값 미리할당
```

```
CREATE TABLE EX_TABLE (BOARD_NUM NUMBER(4) NOT NULL);

INSERT INTO EX_TABLE(BOARD_NUM) VALUES(EX_SEQ.NEXTVAL);
```


1 시퀀스

뷰의 기본 테이블

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

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

- ❖ 시작 값이 1이고 1씩 증가하는 시퀀스 EMP_SEQ을 생성합니다.



1 시퀀스

CURRVAL, NEXTVAL

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

CURRVAL : 현재 값을 반환한다.

NEXTVAL : 현재 시퀀스값의 다음 값을 반환한다

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

CURRVAL, NEXTVAL

NEXTVAL, CURRVAL을 사용할 수 있는 경우와 사용할 수 없는 경우를 살펴봅시다.

❖ NEXTVAL, CURRVAL을 사용할 수 있는 경우

- 서브 쿼리가 아닌 SELECT 문
- INSERT 문의 SELECT 절
- INSERT 문의 VALUE절
- UPDATE문의 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

아래와 같이 출력하시오.

1. 시작 값이 1이고 1씩 증가하고, 최댓값이 100000이 되는 시퀀스 EMP_SEQ 생성합니다.

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

1 실습.

아래와 같이 출력하시오.

2. 이번에는 생성된 시퀀스를 사용하기 위해서 사원 번호를 기본 키로 설정하여 EMP01란 이름으로 새롭게 생성합니다.

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

3. 사원 번호를 저장하는 EMPNO 컬럼은 기본 키로 설정하였으므로 중복된 값을 가질 수 없습니다. 다음은 생성한 EMP_SEQ 시퀀스로부터 사원번호를 자동으로 할당받아 데이터를 추가하는 문장입니다.

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

1 시퀀스

시퀀스 수정

- ❖ 시퀀스를 제거한 후 확인까지 해보았습니다. 이번에는 시퀀스를 변경해 봅시다.
- ❖ 시퀀스를 변경하려면 ALTER SEQUENCE 문을 사용해야 합니다.

```
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 동일합니다.
- ❖ START WITH 옵션은 ALTER SEQUENCE를 써서 변경할 수 없습니다.
- ❖ 다른 번호에서 다시 시작하려면 이전 시퀀스를 삭제하고 다시 생성해야 합니다.

1 실습.

아래와 같이 출력하시오.

9. EMP 테이블의 자료를 입사일을 오름차순으로 정렬하여 최근 입사한 직원을 먼저 출력하되 사원번호, 사원명, 직급, 입사일 칼럼을 출력하는 쿼리문을 작성하시오.

THANK YOU