

시퀀스

시퀀스의 개념 이해와 시퀀스 생성

❖ 시퀀스의 필요성

- 행을 구분하기 위해 기본키를 사용
- 기본키는 항상 유일한 값을 가져야 함
- 사용자가 유일한 값을 직접 생성하려면 부담이 큼

❖ 시퀀스

- 테이블 내의 유일한 숫자를 자동으로 생성하는 자동 번호 발생기
- 시퀀스 값을 기본키로 사용하면 편리함

시퀀스의 개념 이해와 시퀀스 생성

❖ 시퀀스의 생성

○ 형식

```
CREATE SEQUENCE 시퀀스_이름  
[START WITH n]  
[INCREMENT BY n]  
[{MAXVALUE n | NOMAXVALUE }]  
[{MINVALUE n | NOMINVALUE }]  
[{CYCLE | NOCYCLE }]  
[{CACHE n | NOCACHE }]
```

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

시퀀스 관련 데이터 딕셔너리

❖ 시퀀스 관련 데이터 딕셔너리 - SEQUENCE_NAME

```
DESC USER_SEQUENCES;
```

이름	널	유형
SEQUENCE_NAME	NOT NULL	VARCHAR2(30)
MIN_VALUE		NUMBER
MAX_VALUE		NUMBER
INCREMENT_BY	NOT NULL	NUMBER
CYCLE_FLAG		VARCHAR2(1)
ORDER_FLAG		VARCHAR2(1)
CACHE_SIZE	NOT NULL	NUMBER
LAST_NUMBER	NOT NULL	NUMBER

❖ 시퀀스 관련 데이터 디렉터리 - SEQUENCE_NAME

[illegible]

시퀀스 값을 알아보는 CURRVAL과 NEXTVAL

❖ NEXTVAL, CURVAL의 사용여부 확인하기

- CURRVAL에 새로운 값이 할당되기 위해서는 NEXTVAL로 새로운 값을 생성해야 한다.
 - CURRVAL : 현재 값을 반환한다.
 - NEXTVAL : 현재 시퀀스값의 다음 값을 반환한다.
- NEXTVAL, CURRVAL을 사용할 수 있는 경우
 - 서브쿼리가 아닌 SELECT 문
 - INSERT문의 SELECT 절
 - INSERT문의 VALUE절
 - UPDATE문의 SET절

시퀀스 값을 알아보는 CURRVAL과 NEXTVAL

❖ NEXTVAL, CURVAL의 사용여부 확인하기

- NEXTVAL, CURVAL을 사용할 수 없는 경우
 - VIEW의 SELECT절
 - DISTINCT 키워드가 있는 SELECT문
 - GROUP BY, HAVING, ORDER BY 절이 있는 SELECT 문
 - SELECT, DELETE, UPDATE의 서브 쿼리
 - CREATE TABLE, ALTER TABLE의 DEFAULT 값

시퀀스 값을 알아보는 CURRVAL과 NEXTVAL

❖ CURRVAL, NEXTVAL 사용하기

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

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

- 각각의 문장을 반복해서 실행하면

❖ 시퀀스 사용시 주의점

- 반드시 NEXTVAL을 사용한 후 CURRVAL을 사용
- CURRVAL 값을 조회하려면 NEXTVAL로 새로운 값을 미리 생성해 두어야 한다

시퀀스 실무에 적용하기

❖ 시퀀스의 용도

- 99.9%가 INSERT 연산과 같이 사용되어 컬럼 값을 자동으로 발생시키는 용도로 사용

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

```
DROP TABLE EMP01;
```

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

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

```
SELECT * FROM EMP01;
```

시퀀스 실무에 적용하기

❖ 부서 테이블에 시퀀스 적용하기

- DEPTNO가 기본키이며 10, 20 30, ... 순으로 배정

```
CREATE TABLE DEPT_EXAMPLE(  
    DEPTNO NUMBER(4) PRIMARY KEY,  
    DNAME VARCHAR2(15),  
    LOC VARCHAR2(15));
```

```
CREATE SEQUENCE DEPT_EXAMPLE_SEQ  
    INCREMENT BY 10  
    START WITH 10  
    NOCYCLE;
```

시퀀스 실무에 적용하기

❖ 부서 테이블에 시퀀스 적용하기

- DEPTNO가 기본키이며 10, 20 30, ... 순으로 배정

```
INSERT INTO DEPT_EXAMPLE  
VALUES(DEPT_EXAMPLE_SEQ.NEXTVAL, '인사과', '서울');
```

```
INSERT INTO DEPT_EXAMPLE  
VALUES(DEPT_EXAMPLE_SEQ.NEXTVAL, '경리과', '서울');
```

```
INSERT INTO DEPT_EXAMPLE  
VALUES(DEPT_EXAMPLE_SEQ.NEXTVAL, '총무과', '대전');
```

```
INSERT INTO DEPT_EXAMPLE  
VALUES(DEPT_EXAMPLE_SEQ.NEXTVAL, '기술팀', '인천');
```

```
SELECT * FROM DEPT_EXAMPLE;
```

시퀀스 제거와 수정

❖ 시퀀스를 삭제하는 방법

```
SELECT SEQUENCE_NAME, INCREMENT_BY, MAX_VALUE  
FROM USER_SEQUENCES
```

```
DROP SEQUENCE DEPT_EXAMPLE_SEQ;
```

```
SELECT SEQUENCE_NAME, INCREMENT_BY, MAX_VALUE  
FROM USER_SEQUENCES
```

시퀀스 제거와 수정

❖ 시퀀스를 수정하는 방법

- ALTER SEQUENCE 문 사용
 - START WITH는 변경할 수 없음
 - 다른 번호에서 다시 시작하고 싶다면 이전 시퀀스를 제거하고 다시 생성

```
DROP SEQUENCE DEPT_DEPTNO_SEQ;
```

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

```
SELECT SEQUENCE_NAME, INCREMENT_BY, MAX_VALUE  
FROM USER_SEQUENCES
```

시퀀스 제거와 수정

❖ 시퀀스를 수정하는 방법

```
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;
```

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

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

```
SELECT SEQUENCE_NAME, INCREMENT_BY, MAX_VALUE  
FROM USER_SEQUENCES
```