



15장 시퀀스

이번 장에서는 자동 번호 발생기 역할을 하는 시퀀스에 대해서
살펴보겠습니다.

이 장에서 다룰 내용



1 시퀀스 개념 이해와 시퀀스 생성

2 시퀀스 관련 데이터 디렉터리

3 CURRVAL, NEXTVAL

4 시퀀스 실무에 적용하기

5 시퀀스 제거

6 시퀀스 수정

01. 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.

1.1 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.

1.1 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 1이고 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.

1.1 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 10이고 1씩 증가하는 시퀀스 EMP_SEQ을 생성합니다.

02. 시퀀스 관련 데이터 디렉터리



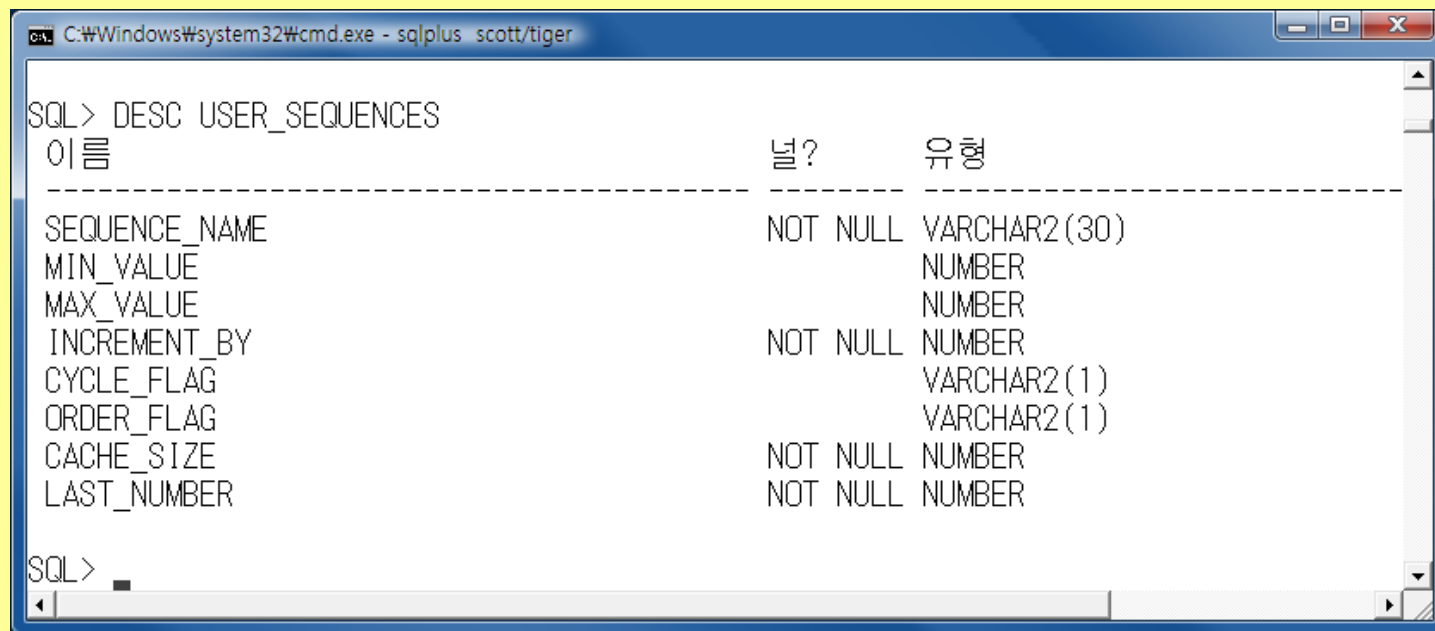
- ❖ 사용자가 작성한 객체들의 정보를 저장하고 있는 데이터 디렉터리의 이름을 보면 다음과 같은 규칙성이 있습니다.
- ❖ 테이블(TABLE) 객체에 대한 정보를 저장하는 데이터 디렉터리는 USER_TABLES이고, 뷰(VIEW)객체에 대한 정보를 저장하는 데이터 디렉터리는 USER_VIEWS이므로 이번 장에서 새로 배운 시퀀스(SEQUENCE)에 대한 자료사전의 이름은 USER_SEQUENCES 이라고 사료될 것입니다.
- ❖ 생성된 시퀀스 객체에 대한 정보를 저장하는 데이터 디렉터리로는 USER_SEQUENCES가 있습니다.

<<실습하기> 시퀀스 관련 데이터 덱서너리

시퀀스 객체의 정보를 살펴봅시다.

1. 우선 데이터 덱서너리 USER_SEQUENCES의 테이블 구조를 먼저 살펴봅시다.

DESC USER_SEQUENCES



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger

SQL> 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

SQL>
```


< <실습하기> 시퀀스 관련 데이터 딕셔너리

SEQUENCE_NAME은 시퀀스 객체의 이름을 저장하고

MIN_VALUE는 최소값, MAX_VALUE는 최대값.

INCREMENT_BY는 증가치에 대한 정보를 가지고 있으며,

CYCLE_FLAG는 CYCLE옵션을 사용하는지, 하지 않는지에 대한 정보를 가지며

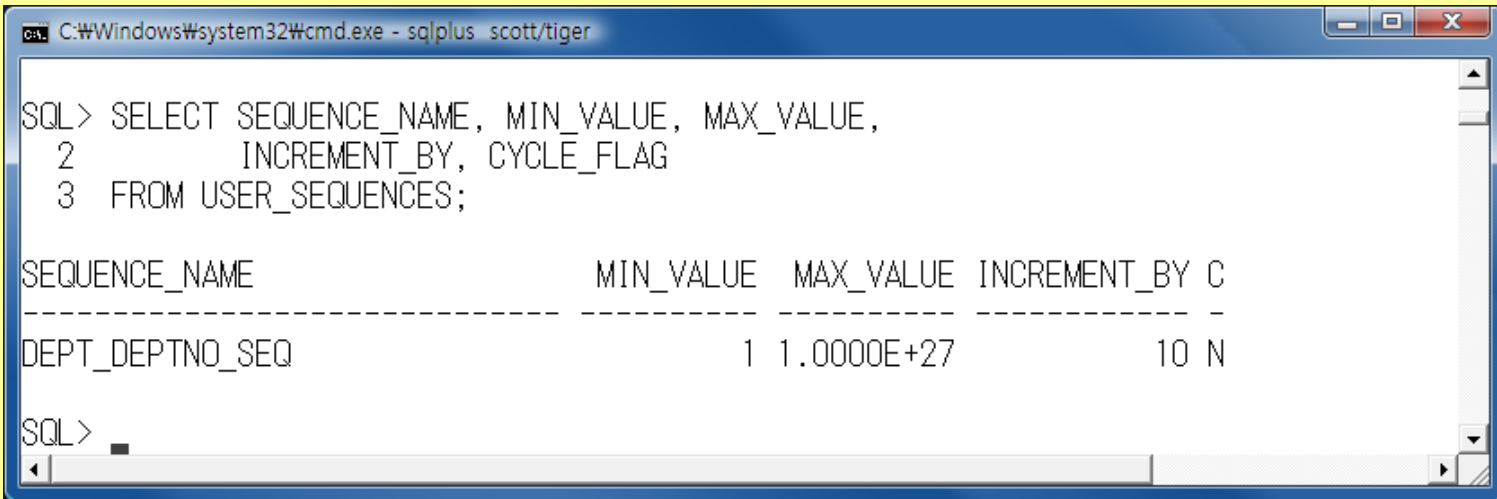
LAST_NUMBER는 마지막 숫자 값을 가지고 있습니다.

2. 데이터 딕셔너리 USER_SEQUENCES로 현재 사용 중인 시퀀스 객체의 정보를 살펴봅시다.

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

<<실습하기> 시퀀스 관련 데이터 덱서너리

지금까지 생성한 시퀀스 객체는 DEPT_DEPTNO_SEQ 한 개이기에 데이터 덱서너리에도 DEPT_DEPTNO_SEQ 시퀀스 객체 하나에 대한 정보만 나타납니다.



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger

SQL> SELECT SEQUENCE_NAME, MIN_VALUE, MAX_VALUE,
2          INCREMENT_BY, CYCLE_FLAG
3          FROM USER_SEQUENCES;

SEQUENCE_NAME          MIN_VALUE  MAX_VALUE  INCREMENT_BY C
-----
DEPT_DEPTNO_SEQ        1 1.0000E+27      10 N

SQL>
```

INCREMENT_BY가 증가치를 지정하는 컬럼으로 10씩 증가한다고 증가치가 설정되어 있고 MIN_VALUE와 MAX_VALUE, CYCLE 옵션 값은 따로 지정하지 않았으므로 기본 값인 1과 10^{27} , N(사이클을 사용하지 않겠다)으로 지정되어 있습니다.

03. 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.

03. 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

〈실습하기〉 CURRVAL, NEXTVAL 사용하기

CURRVAL, NEXTVAL의 실제 사용 예를 살펴봅시다.

1. NEXTVAL로 새로운 값을 생성해야 합니다.

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

2. 시퀀스의 현재 값을 알아내기 위해서 CURRVAL를 사용합니다.

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

04. 시퀀스 실무에 적용하기



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

〈실습하기〉 시퀀스를 테이블의 기본 키에 접목하기

사원 번호를 생성하는 시퀀스 객체를 생성하여 이를 기본 키인 사원 번호에 사용하여 사용자가 새로운 사원을 추가할 때마다 유일한 사원번호를 INSERT 해야 하는 번거로움을 줄입니다.

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 ;
```

〈실습하기〉 시퀀스를 테이블의 기본 키에 접목하기

사원 번호를 생성하는 시퀀스 객체를 생성하여 이를 기본 키인 사원 번호에 사용하여 사용자가 새로운 사원을 추가할 때마다 유일한 사원번호를 INSERT 해야 하는 번거로움을 줄입니다.

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 ;
```


〈실습하기〉 시퀀스를 테이블의 기본 키에 접목하기

사원 번호를 생성하는 시퀀스 객체를 생성하여 이를 기본 키인 사원 번호에 사용하여 사용자가 새로운 사원을 추가할 때마다 유일한 사원번호를 INSERT 해야 하는 번거로움을 줄입니다.

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 ;
```

〈실습하기〉 시퀀스를 테이블의 기본 키에 접목하기

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

```
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);
```

〈실습하기〉 뷰 정의하기

뷰를 생성하려는데 다음과 같이 권한을 설정하였다면 30번 부서에 소속된 사원들의 사번과 이름과 부서번호를 출력하기 위한 SELECT문을 하나의 뷰로 다시 정의해 봅시다.

1. 뷰를 정의합시다.

```
CREATE VIEW EMP_VIEW30  
AS  
SELECT EMPNO, ENAME, DEPTNO  
FROM EMP_COPY  
WHERE DEPTNO=30;
```

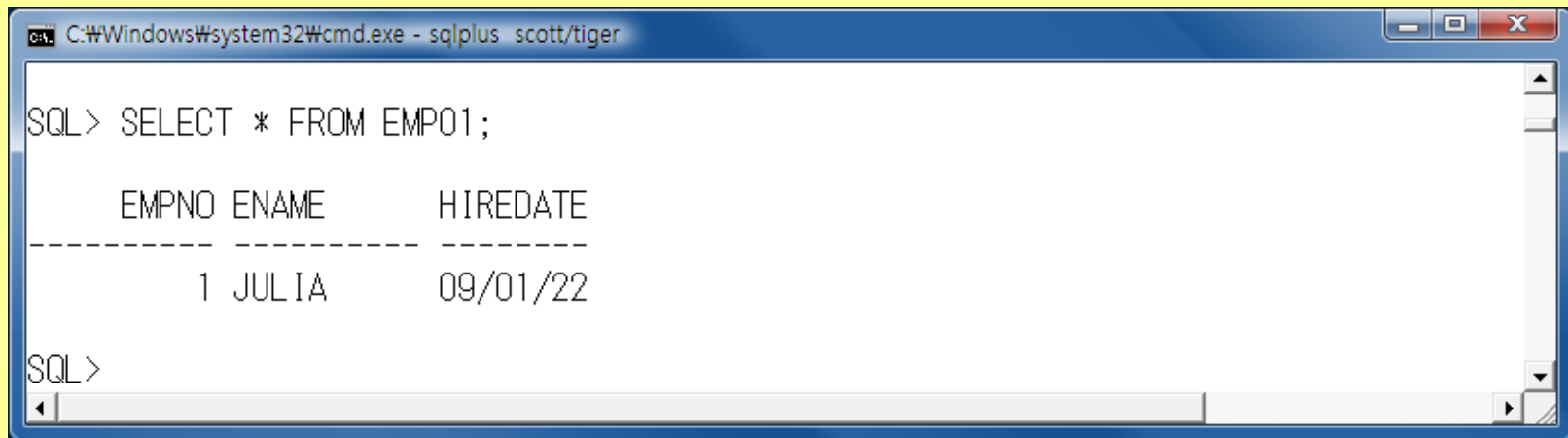
2. 뷰를 생성할 때 컬럼 이름을 명시하지 않으면 뷰(EMP_VIEW30)를 정의하는 기본 테이블(EMP_COPY)의 컬럼 명을 상속받아 사용합니다. 다음은 생성된 뷰의 구조를 살펴봅시다.

```
DESC EMP_VIEW30
```

<실습하기> 뷰 정의하기

5. 데이터를 추가하면서 EMP_SEQ 시퀀스로부터 직원번호를 자동으로 할당받았는지 EMP01 테이블의 내용을 확인합시다.

```
SELECT * FROM EMP01;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the command "SELECT * FROM EMP01;". The output is a table with three columns: EMPNO, ENAME, and HIREDATE. The first row of data shows EMPNO 1, ENAME JULIA, and HIREDATE 09/01/22. The prompt "SQL>" is visible at the bottom left of the window.

```
SQL> SELECT * FROM EMP01;
```

EMPNO	ENAME	HIREDATE
1	JULIA	09/01/22

```
SQL>
```

<스스로 해 보기>

1. 부서 번호를 생성하는 시퀀스 객체를 생성하여 시퀀스 객체를 이용하여 부서 번호를 자동 생성하도록 해 봅시다.

이 문제를 풀기 위해서 다음과 같이 부서 테이블을 생성합니다.

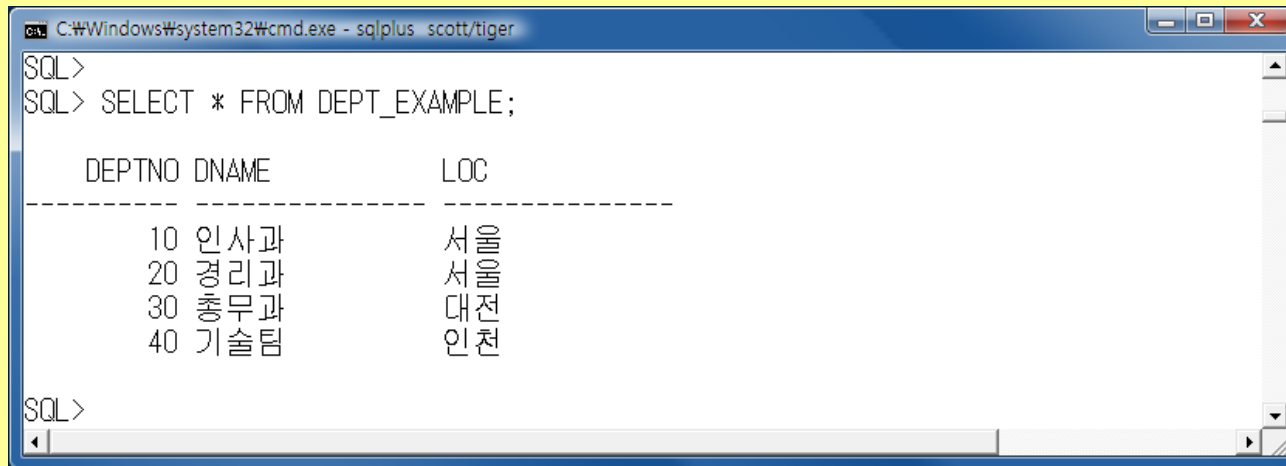
```
CREATE TABLE DEPT_EXAMPLE(  
  DEPTNO NUMBER(4) PRIMARY KEY,  
  DNAME VARCHAR(15),  
  LOC VARCHAR(15)  
);
```

부서 번호를 저장하는 DEPTNO 컬럼은 기본 키로 설정하였으므로 중복된 값을 가질 수 없습니다.

<스스로 해 보기>

다음은 시퀀스 객체 생성 후 로우를 추가한 결과를 확인한 결과 화면입니다.

```
SELECT * FROM DEPT_EXAMPLE;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the command "SELECT * FROM DEPT_EXAMPLE;". The output is a table with three columns: DEPTNO, DNAME, and LOC. The data is as follows:

DEPTNO	DNAME	LOC
10	인사과	서울
20	경리과	서울
30	총무과	대전
40	기술팀	인천

The prompt "SQL>" is visible at the bottom of the window.

추가된 행을 살펴보면 시퀀스 객체가 발생시킨 일련번호가 부서 번호에 적용된 것을 확인할 수 있습니다.

시퀀스 객체의 초기 값을 10으로 하여 10씩 증가하도록 했기 때문에 증가치를 지정했기 때문에 부서 번호가 10, 20, 30, 40으로 지정된 것을 확인할 수 있습니다.

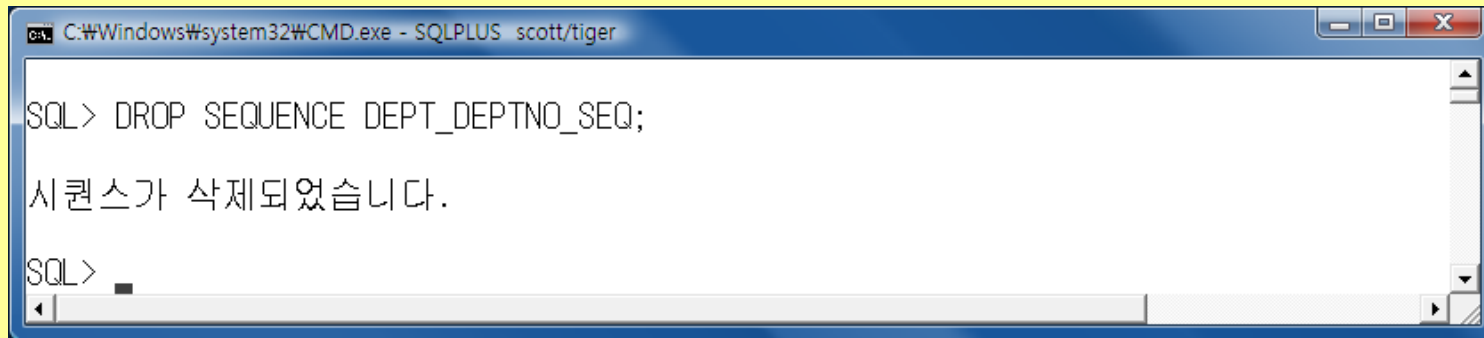
<탄탄히 다지기>

- 1.1 DEPTNO 컬럼에 유일한 값을 가질 수 있도록 시퀀스 객체 생성(시퀀스 이름 : DEPT_EXAMPLE_SEQ)해 봅시다.
- 1.2 새로운 로우를 추가할 때마다 시퀀스에 의해서 다음과 같이 부서 번호가 자동 부여되도록 해 봅시다.

〈실습하기〉 시퀀스 제거하기

DROP SEQUENCE문으로 시퀀스를 제거해 봅시다.

```
DROP SEQUENCE DEPT_DEPTNO_SEQ;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger". The command prompt displays the following text:

```
SQL> DROP SEQUENCE DEPT_DEPTNO_SEQ;  
시퀀스가 삭제되었습니다.  
SQL>
```


05. 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.

〈실습하기〉 시퀀스 최대값을 변경하기

이미 생성해서 사용하던 시퀀스의 최대값을 변경해 봅시다.

1. 시퀀스는 최대값을 지정하지 않으면 기본적으로 10^{27} 으로 지정됩니다. 사용자가 임의로 최대값을 지정할 수 있는데 MAXVALUE에 값을 지정하면 됩니다. 10부터 10씩 증가하면서 최대 30까지의 값을 갖는 시퀀스를 생성합니다.

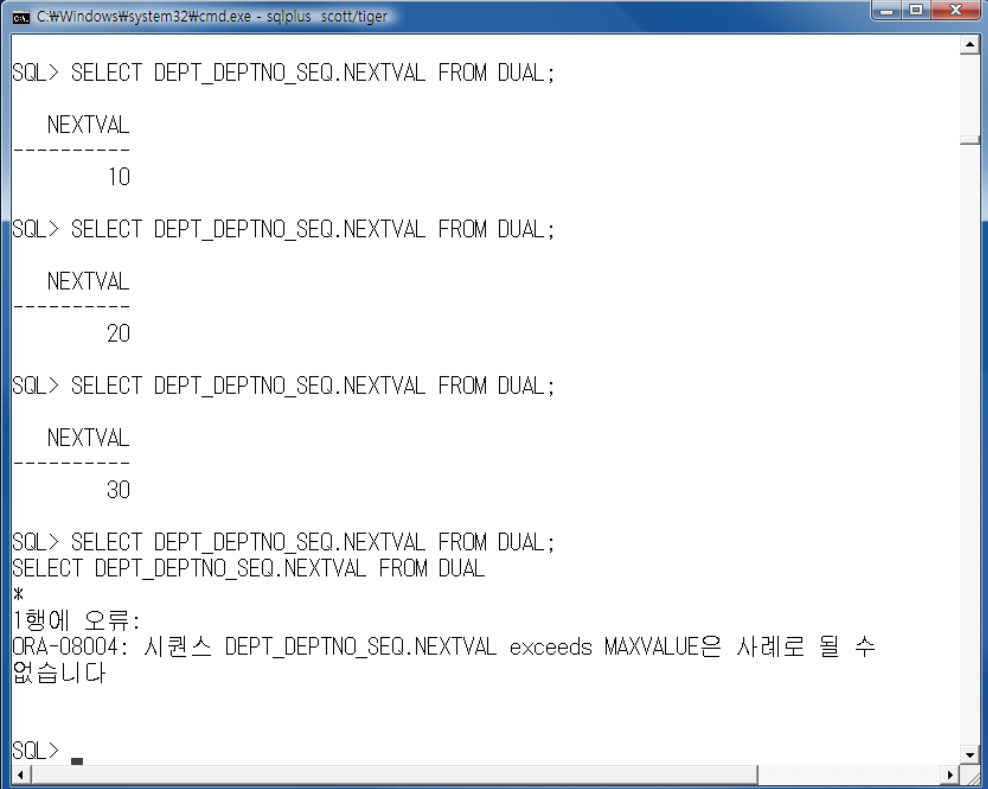
```
DROP SEQUENCE DEPT_DEPTNO_SEQ;  
CREATE SEQUENCE DEPT_DEPTNO_SEQ  
START WITH 10  
INCREMENT BY 10  
MAXVALUE 30;
```

〈실습하기〉 시퀀스 최대값을 변경하기

2. 부서 번호를 계속 생성하다 보면 최대값을 넘게 됩니다. 최대값을 넘을 때까지 시퀀스를 생성해 봅시다.

```
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;
```

이때 CYCLE 옵션을 지정하지 않으면 기본값으로 NOCYCLE를 갖게 되므로 오류가 발생하게 됩니다.



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;  
  
NEXTVAL  
-----  
10  
  
SQL> SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;  
  
NEXTVAL  
-----  
20  
  
SQL> SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;  
  
NEXTVAL  
-----  
30  
  
SQL> SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL;  
SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL  
*  
1행에 오류:  
ORA-08004: 시퀀스 DEPT_DEPTNO_SEQ.NEXTVAL exceeds MAXVALUE은 사례로 될 수  
없습니다  
  
SQL>
```

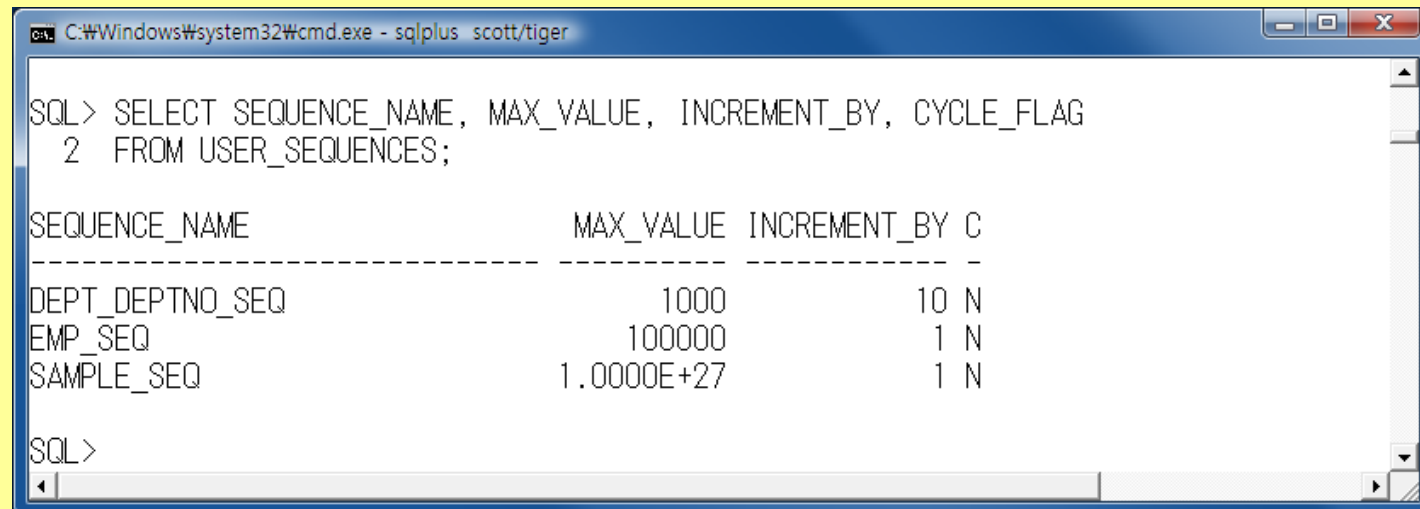
〈실습하기〉 시퀀스 최대값을 변경하기

3. ALTER SEQUENCE문을 사용하여 사용 중이던 DEPT_DEPTNO_SEQ 시퀀스의 최대값을 수정해 봅시다.

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

4. USER_SEQUENCES 를 조회하면 시퀀스가 수정되었는지 확인할 수 있습니다.

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



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT SEQUENCE_NAME, MAX_VALUE, INCREMENT_BY, CYCLE_FLAG  
2 FROM USER_SEQUENCES;  
  
SEQUENCE_NAME          MAX_VALUE INCREMENT_BY C  
-----  
DEPT_DEPTNO_SEQ         1000      10 N  
EMP_SEQ                 100000    1 N  
SAMPLE_SEQ              1.0000E+27 1 N  
  
SQL>
```



Thank You !

Dynamic_오라클 11g + PL/SQL 입문 15장