

5-2.SEQUENCE 와 SYNONYM

1. SEQUENCE

- SEQUENCE 생성 및 옵션 확인하기
- 값이 감소하는 SEQUENCE 생성 및 사용하기
- SEQUENCE 조회 및 수정하기

2. SYNONYM

- SYNONYM 생성
- SYNONYM 조회

1. SEQUENCE

1. SEQUENCE (시퀀스)

- 연속적인 번호를 만들어 주는 기능
- 문 법

CREATE SEQUENCE sequence_name

[INCREMENT BY n] <- 시퀀스 번호의 증가 값으로 기본값은 1

[START WITH n] <- 시퀀스 시작번호로 기본값은 1

[MAXVALUE n | NOMAXVALUE] <- 생성 가능한 시퀀스 최대값

[MINVALUE n | NOMINVALUE] <- CYCLE일 경우 새로 시작되는 값

[CYCLE | NOCYCLE] <- 시퀀스 번호를 순환 사용할 것인지 지정

[CACHE n | NOCACHE] <- 시퀀스 생성속도를 개선하기 위해 캐싱여부 지정

- 사용 예제:

- 아래의 조건으로 제품 주문번호를 생성하기 위해 사용할 SEQUENCE 를 만드세요.
- * SEQUENCE 명: seq_jumun_no
- * 시작 번호: 1000
- * 끝 번호: 1010
- * 증가 값: 1
- * 반복 되고 캐싱은 2개씩 되도록 하세요.

```
CREATE SEQUENCE seq_jumun_no
2 INCREMENT BY 1
3 START WITH 1000
4 MAXVALUE 1010
5 MINVALUE 990
5 CYCLE
6 CACHE 2 ;
```

1) SEQUENCE 생성 및 옵션 확인하기

```
CREATE TABLE jumun  
2 (j_no number(4) ,  
3 j_name varchar2(10)) ;
```

예제용 테이블 생성

```
INSERT INTO jumun  
2 VALUES(seq_jumun_no.NEXTVAL, 'AAAA');
```

1 row created.

```
SCOTT>SELECT * FROM jumun ;
```

J_NO	J_NAME
1000	AAAA

```
INSERT INTO jumun  
2 VALUES(seq_jumun_no.NEXTVAL, 'BBBB') ;
```

1 row created.

```
SCOTT>SELECT * FROM jumun ;
```

J_NO	J_NAME
1000	AAAA
1001	BBBB

```
SELECT seq_jumun_no.CURRVAL FROM dual;
```

CURRVAL
1001

NEXTVAL : 다음 값
CURRVAL : 현재 값

Step 3. MAXVALUE / MINVALUE 항목과 CYCLE 값을 테스트 합니다

```
BEGIN
2  FOR i in 1..11 LOOP
3    INSERT INTO jumun VALUES(seq_jumun_no.NEXTVAL ,
4                               DBMS_RANDOM.STRING('A',4) );
5  END LOOP;
6  COMMIT ;
7  END ;
8 /
```

시퀀스 번호를 모두 소진되게끔 데이터를
11건 입력하는 PL/SQL 입니다

PL/SQL procedure successfully completed.

```
SELECT * FROM jumun ;
```

J_NO	J_NAME
1000	AAAA
1001	BBBB
1002	ldXB
1003	UEJk
1004	ywWy
1005	hHAy
1006	wvhJ
1007	pWvQ
1008	ueod
1009	Iqeg
1010	UwuX

11건의 데이터가 입력되었음이
확인됩니다.

11 rows selected.

```
SCOTT>INSERT INTO jumun  
2 VALUES(seq_jumun_no.NEXTVAL,'DDDD') ;
```

1 row created.

```
SCOTT>SELECT * FROM jumun ;
```

J_NO	J_NAME
1000	AAAA
1001	BBBB
(중간생략)	
990	DDDD

시퀀스 번호가 모두 소진되어
CYCLE 옵션으로 준 MINVALUE 값이
입력이 되었습니다.

Step 4. NOCYCLE 설정과 CACHE 설정을 테스트 합니다.

```
CREATE TABLE jumun2  
2 (j_no number(4) ,  
3 j_name varchar2(4) );
```

Table created.

```
CREATE SEQUENCE seq_jumun2_no  
2 INCREMENT BY 1  
3 START WITH 100  
4 MAXVALUE 105  
5 CACHE 2 ;
```

Sequence created.

실습을 위해 테이블과
시퀀스를 하나씩 더 생
성합니다.

```
INSERT INTO jumun2  
VALUES(seq_jumun2_no.NEXTVAL,'AAAA') ;  
1 row created.
```

```
COMMIT;  
Commit complete.
```

```
CONN / AS SYSDBA ;  
Connected.
```

```
shutdown abort;  
ORACLE instance shut down.
```

```
STARTUP  
ORACLE instance started.
```

```
...  
Database opened.
```

CACHE 2 로 생성된 부분을 테스트 하기 위해 1개의 데이터만 입력한 후 데이터베이스를 강제 종료를 시키고 다시 재시작을 시켰습니다.

```
SYS>CONN scott/tiger
Connected.
```

```
SCOTT>SELECT * FROM jumun2;
```

J_NO	J_NAME
100	AAAA

```
INSERT INTO jumun2
VALUES(seq_jumun2_no.NEXTVAL,'BBBB');
```

1 row created.

```
SCOTT>SELECT * FROM jumun2;
```

J_NO	J_NAME
100	AAAA
102	BBBB

원래 101 번이 입력되어야 하지만 시퀀스 생성할 때 CACHE 값을 2로 주어서 100, 101 번이 캐싱되어있다가 서버가 비정상 종료되어서 캐싱되었던 101 번이 사라지고 102 번이 입력이 된 것입니다.

NOCACHE 속성일 경우 초과되는 값이 요청되면 아래와 같이 에러가 발생합니다.

```
INSERT INTO jumun2  
  2 VALUES(seq_jumun2_no.NEXTVAL , 'FFFF') ;  
VALUES(seq_jumun2_no.NEXTVAL , 'FFFF')  
      *
```

ERROR at line 2:

ORA-08004: sequence SEQ_JUMUN2_NO.NEXTVAL exceeds MAXVALUE and cannot be instantiated

2) 값이 감소하는 SEQUENCE 생성 및 사용하기

```
SCOTT>CREATE SEQUENCE s_rev
```

```
2 INCREMENT BY -1
```

```
3 MINVALUE 0
```

```
4 MAXVALUE 10
```

```
5 START WITH 5 ;
```

Sequence created.

```

COL 이름 FOR a15 ;
SELECT sequence_name "이름" ,
2      MIN_VALUE "MIN 값" ,
3      MAX_VALUE "최대값" ,
4      INCREMENT_BY "증가값" ,
5      CYCLE_FLAG "CYCLE?" ,
6      ORDER_FLAG "정렬" ,
7      CACHE_SIZE "캐싱값" ,
8      LAST_NUMBER "마지막생성값"
9 FROM user_sequences
10 WHERE sequence_name='SEQ_JUMUN2_NO' ;

```

SEQUENCE 조회 및 수정하기

이름	MIN 값	최대값	증가값	C	정	캐싱값	마지막생성값
SEQ_JUMUN2_NO	1	105	1	N	N	2	106

```

SCOTT>ALTER SEQUENCE seq_jumun2_no
2 MAXVALUE 110
3 CACHE 5 ;

```

Sequence altered.

SEQUENCE 를 변경할 때 START WITH 값은 변경이
안 됩니다.

3) SEQUENCE 삭제하기

```
SCOTT>DROP SEQUENCE seq_jumun2_no ;
```

2. SYNONYM (동의어)

SYNONYM (동의어)

```
CREATE [PUBLIC] SYNONYM synonym이름  
FOR [schema.]대상객체 ;
```

```
SCOTT>CONN / AS SYSDBA  
Connected.
```

```
SYS>GRANT create synonym TO scott ;  
SYS>GRANT create public synonym TO scott ;
```

```
SYS>CONN scott/tiger  
Connected.
```


2) 생성 예 :

1. Scott 사용자의 emp2 테이블의 동의어를 e2 로 생성하세요.

```
SCOTT>CREATE SYNONYM e2 FOR emp2 ;
```

2. Scott 사용자의 department 테이블의 동의어를 d2 로 생성하되 모든 사용자들이 사용 할 수 있도록 생성하세요.

```
SCOTT>CREATE PUBLIC SYNONYM d2 FOR department ;
```

3) SYNONYM 조회하기

```
SELECT synonym_name, table_owner , table_name  
2 FROM user_synonyms  
3 WHERE table_name='EMP2' ;
```

SYNONYM_NAME	TABLE_OWNER	TABLE_NAME
-----	-----	-----
E2	SCOTT	EMP2

4) SYNONYM 삭제하기

```
SCOTT>DROP SYNONYM e2 ;
```