SEQUENCE

[정의] 일련번호(고유성) 생성기

[용도] 의미가 없지만 고유성을 식별하는 일련번호(ex 접수번호,배송번호,등록번호)에 사용

일련번호 자동생성 방식: ① MAX(SEQ) + 1 방식,

② 채번 TABLE 방식

3 SEQUENCE

① CREATE SEQUENCE SCOTT.ORDER_SEQ

INCREMENT BY 1

START WITH 1

MAXVALUE 99999999999

MINVALUE 1

NOCYCLE

CACHE 30;

// ?? 2 SELECT ORDER_SEQ.CURRVAL FROM DUAL;

3 SELECT ORDER_SEQ.NEXTVAL FROM DUAL;

SELECT ORDER_SEQ.CURRVAL FROM DUAL;

SELECT ORDER_SEQ.CURRVAL FROM DUAL;

// ?? ROLLBACK;

4 SELECT ORDER_SEQ.NEXTVAL FROM DUAL;

SELECT ORDER_SEQ.NEXTVAL FROM DUAL;

⑤ SELECT EMPNO, ENAME, ORDER_SEQ. NEXTVAL FROM EMP;

Connect scott/tiger

6 SELECT ORDER_SEQ.NEXTVAL FROM DUAL;

8 SELECT ORDER_SEQ.NEXTVAL FROM DUAL;

SELECT ORDER_SEQ.CURRVAL FROM DUAL;

Connect scott/tiger

SELECT ORDER_SEQ.NEXTVAL FROM DUAL;

(11) SELECT ORDER_SEQ.CURRVAL FROM DUAL;

② SELECT ORDER_SEQ.CURRVAL FROM DUAL;

SELECT ORDER_SEQ.CURRVAL FROM DUAL;

```
// 테이블과는 독립적으로 생성되고 사용된다.
여러 세션(여러 Application)에서 독립적으로 동시에 사용하는 공용 오브젝트
```

ED C:₩03_SQL₩ORDER_TBL.SQL

```
DROP TABLE SCOTT.ORDERS;
CREATE TABLE SCOTT.ORDERS
                                                     // ORDER 테이블명을 사용할수 없는 이유는?
( ORDER_ID
              NUMBER(12)
                           CONSTRAINT ORDER_ID_PK PRIMARY KEY,
                            DEFAULT SYSDATE CONSTRAINT ORDER_DATE_NN NOT NULL,
  ORDER_DATE DATE
  ORDER_MODE VARCHAR2(8) CONSTRAINT ORDER_MODE_CHK CHECK(ORDER_MODE IN ('direct', 'online')),
                            CONSTRAINT ORDER_CUSTOMER_ID_NN NOT NULL,
  CUSTOMER_ID NUMBER(6)
  ORDER_STATUS NUMBER(2), -- 0: 주문중 , 1: 주문완료, 2: 취소 , 3: 발송 4: 수취
  SALES_ID
                NUMBER(6), -- 판매사원번호
  CONSTRAINT ORDER_SALES_ID_FK FOREIGN KEY(SALES_ID) REFERENCES SCOTT.EMP(EMPNO)
):
DESC SCOTT.ORDERS
```

- @ C:₩03_SQL₩ORDER_TBL.SQL
- // EMP 테이블의 EMPNO 컬럼에 필요시 p.k 제약사항 추가
- ① INSERT INTO ORDERS(ORDER_ID,ORDER_MODE,CUSTOMER_ID,ORDER_STATUS,SALES_ID)

 VALUES(ORDER_SEQ_NEXTVAL, 'direct',166,1,7499);
- ② INSERT INTO SCOTT.ORDERS(ORDER_ID,ORDER_DATE,ORDER_MODE,CUSTOMER_ID,ORDER_STATUS,SALES_ID) VALUES(SCOTT.ORDER_SEQ.NEXTVAL,SYSDATE,'online',200,3,7521);
- 3 COMMIT;
- INSERT INTO ORDERS(ORDER_ID,ORDER_DATE,ORDER_MODE,CUSTOMER_ID,ORDER_STATUS,SALES_ID)
 VALUES(ORDER_SEQ.NEXTVAL,SYSDATE,'online',135,2,7844);
- ⑤ ROLLBACK;

// ORDER_SEQ.NEXTVAL도 취소가 되는지?

® SELECT ORDER_ID FROM ORDERS;
SELECT ORDER_SEQ.CURRVAL FROM DUAL;

// Hole 발생

- INSERT INTO ORDERS(ORDER_ID,ORDER_DATE,ORDER_MODE,CUSTOMER_ID,ORDER_STATUS,SALES_ID)
 VALUES(ORDER_SEQ.NEXTVAL,SYSDATE,'direct',135,4,7844);
- ® // MAX+1 방식 , 문제점은 ?
 INSERT INTO scott.ORDERS(ORDER_ID,ORDER_MODE,CUSTOMER_ID,ORDER_STATUS,SALES_ID)

```
VALUES((SELECT MAX(ORDER_ID)+1 FROM scott.ORDERS), 'direct',335,1,7654);
ED C:\(\Psi_03_\text{SQL}\(\Psi_0\text{PRDER_ID_TBL.SQL}\)
```

```
DROP TABLE SCOTT.ORDER_ID;

CREATE TABLE SCOTT.ORDER_ID

( ORDER_SEQ NUMBER(12)
);
INSERT INTO SCOTT.ORDER_ID VALUES(100);

COMMIT;
```

@ C:₩03_SQL₩ ORDER_ID_TBL.SQL

1

```
DECLARE
```

V_ORDER_SEQ NUMBER(12);

BEGIN

SELECT ORDER_SEQ INTO V_ORDER_SEQ

FROM ORDER_ID FOR UPDATE;

INSERT INTO ORDERS(ORDER_ID,ORDER_MODE,CUSTOMER_ID,ORDER_STATUS,SALES_ID)

VALUES(V_ORDER_SEQ, 'direct', 3320, 2, 7677);

UPDATE ORDER_ID SET ORDER_SEQ = ORDER_SEQ +1;

COMMIT;

END;

/

// 에러를 해결 하십시요

[요구]

- @ SEQUENCE의 START 값을 10에서 다시 시작하는 방법은?
- ⓑ 서로 다른 SESSION에서 SEQUENCE를 공유하여 사용해보고 결과 발표
- ⓒ SEQUENCE를 생성하여 00000001, 00000002와 같은 8자리 접수번호를 사용하는 SQL작성 하십시요
- ⓓ 위의 주문(ORDERS) 테이블에 추가해야할 컬럼을 고려해서 스크립트를 수정 하십시요