

시퀀스

자바 강의실

목차

I. 시퀀스

II. 동의어

III. 데이터베이스 링크

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

- 시퀀스 개념 이해와 시퀀스 생성
 - 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.

1. 시퀀스

- 시퀀스 개념 이해와 시퀀스 생성

- ② INCREMENT BY : 연속적인 시퀀스 번호의 증가치를 지정할 때 사용됩니다. 만일 1씩 증가하는 시퀀스를 생성하려면 INCREMENT BY 1이라고 기술하면 됩니다.
- ③ MAXVALUE n | NOMAXVALUE : MAXVALUE 은 시퀀스가 가질 수 있는 최대값을 지정합니다. 만일 NOMAXVALUE를 지정하게 되면 ASCENDING 순서일 경우에는 1027승이고 DESCENDING 순서일 경우에는 -1로 설정됩니다.
- ④ MINVALUE n | NOMINVALUE : MINVALUE 은 시퀀스가 가질 수 있는 최소값을 지정합니다. 만일 NOMINVALUE을 지정하게 되면 ASCENDING 순서일 경우에는 1이고 DESCENDING 순서일 경우에는 1026승으로 설정됩니다.

1. 시퀀스

- 시퀀스 개념 이해와 시퀀스 생성
 - ⑤ CYCLE | NOCYCLE : CYCLE 은 지정된 시퀀스 값이 최대값 까지 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

예: 시퀀스 생성

질의 : 다음은 직원 번호를 자동으로 부여해주는 시퀀스 객체를 생성하는 문장입니다.

시작 값이 1이고 1씩 증가하는 시퀀스 EMP_ID_SEQ을 생성합니다.

```
CREATE SEQUENCE EMP_ID_SEQ  
INCREMENT BY 1  
START WITH 1;
```

1. 시퀀스

- 시퀀스 관련 데이터 덱서너리
 - 사용자가 작성한 객체들의 정보를 저장하고 있는 데이터 덱서너리의 이름을 보면 다음과 같은 규칙성이 있습니다.
 - 테이블(TABLE) 객체에 대한 정보를 저장하는 데이터 덱서너리는 USER_TABLES이고, 뷰(VIEW)객체에 대한 정보를 저장하는 데이터 덱서너리는 USER_VIEWS이므로 이번 장에서 새로 배운 시퀀스(SEQUENCE)에 대한 자료사전의 이름은 USER_SEQUENCES 이라고 사료될 것입니다.
 - 생성된 시퀀스 객체에 대한 정보를 저장하는 데이터 덱서너리로 는 **USER_SEQUENCES**가 있습니다.

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.

1. 시퀀스

- 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

1. 시퀀스

- 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.

1. 시퀀스

- 시퀀스 수정

- 이번에는 시퀀스를 변경해 봅시다.
- 시퀀스를 변경하려면 ALTER SEQUENCE 문을 사용

```
ALTER SEQUENCE sequence_name  
[INCREMENT BY n]  
[{MAXVALUE n | NOMAXVALUE}]  
[{MINVALUE n | NOMINVALUE}]  
[{CYCLE | NOCYCLE}]  
[{CACHE n | NOCACHE}]
```

- ALTER SEQUENCE는 **START WITH 절이 없다는 점**을 빼고는 CREATE SEQUENCE와 구조가 동일합니다.
- START WITH 옵션은 ALTER SEQUENCE를 써서 변경불가
- **다른 번호에서 다시 시작하려면 이전 시퀀스를 삭제하고 다시 생성해야 합니다.**

2. 동의어

- 동의어의 개념과 종류

- 데이터베이스의 객체에 대한 소유권은 해당 객체를 생성한 사용자에게 있습니다. 따라서 다른 사용자가 객체에 접근하기 위해서는 소유자로부터 접근 권한을 부여받아야 합니다. 또한 다른 사용자가 소유한 객체에 접근하기 위해서는 소유자의 이름을 객체 앞에 지정해야 합니다.
- 이렇게 객체를 조회할 때마다 일일이 객체의 소유자를 지정하는 것이 번거로울 경우 동의어를 정의하면 긴 이름대신 간단한 이름으로 접근할 수 있게 됩니다.
- 동의어는 개별 사용자를 대상으로 하는 비공개 동의어와 전체 사용자를 대상으로 한 공개 동의어가 있습니다.

2. 동의어

- 동의어 종류

- 동의어는 개별 사용자를 대상으로 하는 비공개 동의어와 전체 사용자를 대상으로 한 공개 동의어가 있습니다.
- 비공개 동의어
 - 객체에 대한 접근 권한을 부여받은 사용자가 정의한 동의어로 해당 사용자만 사용할 수 있다.
- 공개 동의어
 - 권한을 주는 사용자가 정의한 동의어로 누구나 사용할 수 있다. 공개 동의어는 DBA 권한을 가진 사용자만이 생성할 수 있다. SYNONYM 앞에 PUBLIC를 붙여서 정의한다.

2. 동의어

- 동의어의 사용예

- DUAL은 원래 SYS가 소유하는 테이블 명이므로 다른 사용자가 DUAL 테이블에 접근하려면 SYS.DUAL로 표현해야 하는 것이 원칙입니다.
- 그럼에도 불구하고 지금까지 모든 사용자가 SYS.을 생략하고 DUAL이라고 간단하게 사용하였습니다.
- 이럴 수 있었던 이유는 공개 동의어로 지정되어있기 때문입니다.

2. 동의어

- 동의어 생성

- 동의어를 정의하기 위한 CREATE SYNONYM 명령어의 기본 형식은 다음과 같습니다.

```
CREATE [PUBLIC] SYNONYM synonym_name  
FOR user_name.object_name;
```

- synonym_name은 user_name.object_name에 대한 별칭입니다.
- user_name은 객체를 소유한 오라클 사용자이고 object_name은 동의어를 만들려는 데이터베이스 객체 이름입니다.

2. 동의어

- 동의어 종류

- 동의어는 개별 사용자를 대상으로 하는 비공개 동의어와 전체 사용자를 대상으로 한 공개 동의어가 있습니다.
- 비공개 동의어
 - 객체에 대한 접근 권한을 부여받은 사용자가 정의한 동의어로 해당 사용자만 사용할 수 있다.
- 공개 동의어
 - 권한을 주는 사용자가 정의한 동의어로 누구나 사용할 수 있다. 공개 동의어는 DBA 권한을 가진 사용자만이 생성할 수 있다. SYNONYM 앞에 PUBLIC를 붙여서 정의한다.

2. 동의어

예: 시스템 계정으로 테이블 생성

질의 : sys 계정으로 SYSTBL 을 생성한다

```
CREATE TABLE SYSTBL (  
    ENO NUMBER(5),  
    ENAME VARCHAR2(20)  
);
```

```
INSERT INTO SYSTBL VALUES(10, '김창원');  
INSERT INTO SYSTBL VALUES(20, '조상규');
```

2. 동의어

예: 권한 부여하기

질의 : hr 사용자에게 SYS_TBL 이란 테이블을 SELECT 할 권한을 부여합니다.

```
GRANT SELECT ON SYSTBL TO hr; -- sys 계정
```

```
SELECT * FROM SYSTBL; -- hr 계정
```

```
SELECT * FROM sys.SYSTBL;
```

2. 동의어

예: 동의어 생성하기

질의 : 비공개 동의어는 권한을 부여받는 사용자인 hr가 정의해야 함으로
hr 계정으로 접속하여 비공개 동의어를 생성합니다

```
CREATE SYNONYM SYNTBL FOR system.SYSTBL; -- hr 계정  
SELECT * FROM SYNTBL; -- hr 계정
```

hr 이 동의어 생성권한이 없는 경우

```
GRANT CREATE SYNONYM TO hr; -- sys 계정
```

2. 동의어

예: 동의어 제거하기

질의 : 동의어를 소유한 사용자로 접속해야만 제거가 가능합니다.

```
DROP SYNONYM SYNTBL;
```

3. 데이터베이스 링크

- 데이터베이스 링크는 클라이언트 또는 현재의 데이터베이스에서 네트워크상의 다른 데이터베이스에 접속하기 위한 접속 설정을 정의하는 오라클 객체 이다.
 - 우선 고려되어야 사항은 ORACLE INSTANCE가 두 개 이상이고, 각각의 HOST NAME과 ORACLE_SID는 다르고 NLS_CHARACTER_SET은 동일하게 되어 있어야 한다.
 - 만약 같은 MACHINE에서 INSTANCE의 ORACLE_SID가 같다면 TNS ERROR가 발생한다.
 - 미래를 위해 다른 MACHINE이라 할지라도 ORACLE_SID는 규칙에 의해 다르게 가져가는 것이 좋다.
 - NLS_CHARACTER_SET이 동일하게 되어있지 않으면 DATA 입출력시 ?????로 나타난다.
 - 데이터베이스 링크로 연결되는 서버에 Listener가 꼭 구동 되어 있어야 한다.