

12장 시퀀스와 인덱스

이번 장에서는 자동 번호 발생기 역할을 하는 시퀀스와 조회의 성능을 향상시키는 인덱스에 대해서 살펴보도록 하겠습니다.

학습 내용

- ❖ 시퀀스 생성
- ❖ 시퀀스 수정 및 제거
- ❖ 조회시 성능 향상을 위한 인덱스
- ❖ 인덱스의 종류

학습목표

- ❖ PRIMARY KEY로 지정한 칼럼에 일련번호를 자동으로 부여받기 위해 시퀀스의 생성과 사용 방법을
- ❖ 학습합니다.
- ❖ 시퀀스를 수정, 삭제하는 방법을 학습합니다.
- ❖ 조회의 성능을 향상시키기 위해서 인덱스 객체의 사용 방법을 학습합니다.

01. 시퀀스 생성

❖ 시퀀스는 테이블 내의 유일한 숫자를 자동으로 생성해 줍니다.

```
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.

01. 시퀀스 생성

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.

시퀀스 관련 데이터 디렉터리

```
create sequence sample_seq  
increment by 10  
start with 10;
```

- ❖ 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.

시퀀스 관련 데이터 사전

- ❖ **SEQUENCE_NAME**은 시퀀스 객체의 이름
- ❖ **MIN_VALUE**는 최소값, **MAX_VALUE**는 최대값.
- ❖ **INCREMENT_BY**는 증가치에 대한 정보
- ❖ **CYCLE_FLAG**는 **CYCLE**옵션을 사용하는지, 하지 않는지에 대한 정보

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

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.

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**

시퀀스를 기본 키에 접목하기

- ❖ 부서 테이블의 부서번호는 시퀀스가 알아서 자동으로 생성해 주고 나머지 칼럼에 대해서만 값을 INSERT하면 보다 간편하게 쿼리문을 작성할 수 있게 됩니다.

```
create sequence dno_seq  
increment by 10  
start with 10;
```

```
insert into dept_second  
values(dno_seq.nextval, 'ACCOUNTING', 'NEW YORK');
```

02. 시퀀스 수정 및 제거

- ❖ 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.

02. 시퀀스 수정 및 제거

❖ 시퀀스 변경하기

```
alter sequence dno_seq  
maxvalue 50;
```

❖ 시퀀스 제거하기

```
drop sequence dno_seq;
```

03. 조회시 성능 향상을 위한 인덱스

- ❖ 인덱스는 검색 속도를 향상시키기 위해서 사용
- ❖ 인덱스 객체에 대한 정보는 USER_COLUMNS와 USER_IND_COLUMNS 데이터 사전을 통해서 살펴볼 수 있습니다.
- ❖ 기본 키나 유일 키는 인덱스가 자동으로 생성됩니다.

03. 조회시 성능 향상을 위한 인덱스

- ❖ 제약 조건에 의해 자동으로 생성되는 인덱스 외에 CREATE INDEX 명령어로 직접 인덱스를 생성할 수도 있습니다.
- ❖ 다음은 인덱스를 생성하기 위한 기본 형식입니다.

```
CREATE INDEX index_name  
ON table_name (column_name);
```

- CREATE INDEX 다음에 인덱스 객체 이름을 지정합니다. 어떤 테이블의 어떤 컬럼에 인덱스를 설정할 것인지를 결정하기위해서 ON 절 다음에 테이블 이름과 컬럼 이름을 기술합니다.

03. 조회시 성능 향상을 위한 인덱스

❖ 인덱스 생성하기

```
create index idx_employee_ename  
on employee(ename);
```

❖ USER_IND_COLUMNS 데이터 사전으로 인덱스 생성 확인하기

```
select index_name, table_name, column_name  
from user_ind_columns  
where table_name in('EMPLOYEE');
```

03. 조회시 성능 향상을 위한 인덱스

- ❖ 인덱스를 삭제하기 위해서 오라클은 DROP INDEX 명령어를 제공합니다.

```
DROP INDEX index_name;
```


인덱스의 사용 여부를 결정하기 위한 조건

인덱스를 사용해야 하는 경우	인덱스를 사용하지 말아야 하는 경우
테이블에 행의 수가 많을 때	테이블에 행의 수가 적을 때
WHERE 문에 해당 컬럼이 많이 사용될 때	WHERE 문에 해당 컬럼이 자주 사용되지 않을 때
검색 결과가 전체 데이터의 2%~4% 정도 일 때	검색 결과가 전체 데이터의 10%~15% 이상 일 때
JOIN에 자주 사용되는 컬럼이나 NULL을 포함하는 컬럼이 많은 경우	테이블에 DML 작업이 많은 경우 즉, 입력 수정 삭제 등이 자주 일어날 때

04. 인덱스의 종류

❖ 인덱스는 다음과 같이 구분할 수 있습니다.

유일한 값을 갖는지 여부에 따른

고유 인덱스(Unique Index) /비 고유 인덱스(NonUnique Index)

인덱스를 구성하는 칼럼의 개수에 따라

단일 인덱스(Single Index)/결합 인덱스(Composite Index)

함수에 인덱스를 지정하는

함수 기반 인덱스(Function Based Index)