

Introduction to Oracle9i : SQL

Chapter10. 기타 객체의 생성 및 관리

- ▶ 테이블과 뷰를 제외한 나머지 객체의 종류를 알아본다.
- ▶ 유형별 객체의 생성 방법 및 관리 방법을 익힌다.

※ 데이터베이스 객체

객체	의미
테이블	행과 열로 구성된 기본적인 저장 구조
뷰	하나 이상의 테이블에서 데이터의 부분집합을 논리적으로 표현
시퀀스	고유한 번호를 자동으로 발생시키는 객체로 주로 PK 값 생성에 사용
인덱스	질의(SELECT) 성능을 향상시키기 위하여 사용하는 물리적인 저장 구조
시노님	객체에 대한 이름을 부여

1. 시퀀스

(1) 시퀀스의 특징

- 자동적으로 유일 번호를 생성
- 공유 가능한 객체
- 주로 기본 키 값을 생성하기 위해 사용
- 어플리케이션 코드를 대체
- 메모리에 CACHE되면 시퀀스 값 액세스 효율성 향상

1. 시퀀스

(2) 시퀀스 생성

```
CREATE SEQUENCE 시퀀스명  
    [INCREMENT BY n]  
    [START WITH n]  
    [{MAXVALUE n | NOMAXVALUE}]  
    [{MINVALUE n | NOMINVALUE}]  
    [{CYCLE | NOCYCLE}]  
    [{CACHE | NOCACHE}];
```

```
CREATE SEQUENCE dept_deptno  
    INCREMENT BY 1  
    START WITH 91  
    MAXVALUE 99  
    NOCACHE  
    NOCYCLE ;
```

1. 시퀀스

(3) 시퀀스 확인

```
SELECT sequence_name,min_value,max_value,increment_by,last_number  
FROM    user_sequences ;
```

1. 시퀀스

(4) 시퀀스 사용

1) NEXTVAL과 CURRVAL 의사열

- 특징

- NEXTVAL는 다음 사용 가능한 시퀀스 값 반환
- 시퀀스가 참조될 때마다 다른 사용자에게도 유일 값 반환
- CURRVAL은 현재 시퀀스 값 반환

1. 시퀀스

(4) 시퀀스 사용

2) NEXTVAL과 CURRVAL의 사용 규칙

- NEXTVAL과 CURRVAL을 사용할 수 있는 경우
 - 서브쿼리가 아닌 SELECT문
 - INSERT문의 SELECT문
 - INSERT문의 VALUES절
 - UPDATE문의 SET절

1. 시퀀스

(4) 시퀀스 사용

2) NEXTVAL과 CURRVAL의 사용 규칙

- NEXTVAL과 CURRVAL사용할 수 없는 경우
 - 뷰 생성 시 SELECT문
 - DISTINCT 키워드를 사용한 SELECT문
 - GROUP BY, HAVING, ORDER BY를 이용한 SELECT문
 - SELECT, DELETE, UPDATE문장에서의 서브쿼리
 - CREATE TABLE, ALTER TABLE명령문의 DEFAULT절

1. 시퀀스

(4) 시퀀스 사용

```
INSERT INTO dept  
VALUES (dept_deptno.NEXTVAL, '영업부', '분당구 정자동' );
```

```
SELECT dept_deptno.CURRVAL  
FROM dual ;
```

1. 시퀀스

(5) 시퀀스의 CACHE

- 빠른 액세스를 위해 메모리에 SEQUENCE를 CACHE

```
CREATE SEQUENCE dept_deptno  
    INCREMENT BY 1  
    START WITH 91  
    MAXVALUE 99  
    CACHE  
    NOCYCLE;
```

1. 시퀀스

(6) 시퀀스 값의 간격 발생

- ROLLBACK
- SYSTEM CRASH
- 시퀀스가 다른 테이블에서 사용될 때

1. 시퀀스

(7) 시퀀스 수정

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

- 시퀀스에 대한 ALTER 권한을 가지거나 소유자만 수행 가능
- 이후의 시퀀스 번호만 영향
- 다른 번호에서 시작하려면 시퀀스 제거 후 다시 생성

1. 시퀀스

(8) 시퀀스 삭제

```
DROP SEQUENCE 시퀀스명;
```

```
DROP SEQUENCE dept_deptno;
```

2. 인덱스

(1) 인덱스의 특징

- 테이블의 값을 빠르게 액세스 하도록 하는 데이터베이스 객체
- B*TREE을 써서 디스크 입출력 횟수 감소
- 오라클 서버가 인덱스를 자동적으로 사용하고 유지 보수
- 논리적 물리적으로 테이블과 독립적

2. 인덱스

(2) 인덱스 생성 방법

1) 자동 생성

- PRIMARY KEY나 UNIQUE 제약 조건을 정의 시 자동 생성

2) 사용자 생성

- 행에 대한 접근 속도를 향상시키고자 사용자가 직접 생성

2. 인덱스

(3) 사용자 인덱스 생성

```
CREATE INDEX 인덱스명  
ON 테이블명 (컬럼명1[,컬럼명2,.....]);
```

```
CREATE INDEX emp_ename_idx  
ON emp (ename) ;
```

2. 인덱스

(4) 인덱스 생성 지침

1) 언제 인덱스를 생성하면 좋은가?

- WHERE절 또는 조인 조건에서 자주 사용되는 컬럼
- 광범위한 값을 포함하는 컬럼
- 많은 수의 null값을 포함하는 컬럼
- 테이블이 크고 대부분 질의 결과 전체 행의 2~4%보다 적게 검색

2. 인덱스

(4) 인덱스 생성 지침

1) 언제 인덱스를 생성하면 좋지 않은가?

- 작은 테이블
- 질의의 조건으로 자주 사용되지 않는 컬럼
- 대부분의 질의 결과 전체 행의 2~4%이상을 검색
- 자주 갱신되는 테이블

2. 인덱스

(5) 인덱스 생성 확인

```
SELECT c.index_name, c.column_name, c.column_position, i.uniqueness  
FROM   user_indexes i, user_ind_columns c  
WHERE  c.index_name = i.index_name  
AND    c.table_name = 'EMP';
```

2. 인덱스

(6) 함수 기반 인덱스 생성

- 표현식을 기반에 둔 인덱스
- 적절히 활용할 경우 검색속도를 더 향상

```
CREATE INDEX upper_emp_ename_idx  
ON emp (UPPER(ename)) ;
```

2. 인덱스

(7) 인덱스 삭제

- DROP ANY INDEX 권한 필요

```
DROP INDEX 인덱스명;
```

```
DROP INDEX emp_ename_idx;
```

3. 시노님

(1) 시노님의 특징

- 스키마 이름까지 명시하는 수고를 감소
- 테이블, 뷰, 시퀀스, 프로시저 등 객체의 다른 이름 제공
- 긴 이름을 가진 객체에 유용하게 사용

3. 시노님

(2) 시노님 생성

```
CREATE [PUBLIC] SYNONYM 시노님명  
FOR      객체명 ;
```

```
CREATE SYNONYM gubun  
FOR          salgrade;
```



```
SELECT *  
FROM   gubun;
```


3. 시노님

(3) 시노님 삭제

```
DROP [PUBLIC] SYNONYM 시노님명;
```

```
DROP SYNONYM gubun;
```

- PUBLIC 시노님은 DBA만 제거 가능