

Oracle Database Objects

기타 데이터베이스 객체

- 시퀀스 생성, 유지 관리 및 사용
- 인덱스 생성 및 유지 관리
- 공용 동의어 생성

데이터 베이스 객체

객체	설명
테이블	기본 저장 단위며 행과 열로 구성
뷰	논리적으로 하나 이상의 테이블에 있는 데이터의 부분 집합
시퀀스	기본 키 값을 생성
인덱스	질의 성능 향상

■ 시퀀스란?

- 고유 번호를 자동으로 생성
- 공유 가능한 객체
- 일반적으로 기본 키 값을 생성하는 데 사용
- 응용 프로그램 코드를 대체
- 시퀀스 값을 메모리에 캐시하면 액세스 효율이 높아짐

■ 시퀀스를 정의하여 시퀀스 번호를 자동생성 하는 예

CREATE SEQUENCE sequence

[INCREMENT BY n] **n은 정수(default n=1)**

[START WITH n] **시작 번호(default n=1)**

[{MAXVALUE n | NOMAXVALUE}]

(default 오름차순 최대 10^{27} , 내림차순 최대 -1)

[{MINVALUE n | NOMINVALUE}]

(default 오름차순 최소 10^{27} , 내림차순 최소-1)

[{CYCLE | NOCYCLE}]

(최대,최소값 도달 후 계속 생성 여부 default=NOCYCLE 권장)

[{CACHE n | NOCACHE}] **(메모리에 저장할 개수, 최대 20)**

■ 시퀀스 생성

DEPARTMENTS 테이블의 기본 키로 사용할 DEPT_DEPTID_SEQ 생성

```
CREATE SEQUENCE dept_deptid_seq  
    INCREMENT BY 10  
    START WITH 120  
    MAXVALUE 9999  
    NOCYCLE  
    NOCACHE;
```

Sequence created

■ USER_SEQUENCES 데이터 디렉터리 테이블에서 시퀀스 값을 확인

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

NOCACHE가 지정된 경우 LAST_NUMBER열에 사용 가능한 다음 시퀀스 번호가 표시

■ NEXTVAL 및 CURRVAL 의사 열

- NEXTVAL은 사용 가능한 다음 시퀀스 값을 반환
참조될 때마다(서로 다른 사용자 일지라도) 고유한 값을 반환
- CURRVAL은 현재 시퀀스 값을 반환
- CURRVAL이 값을 포함하려면 먼저 해당 시퀀스에 대해
NEXTVAL이 실행되어야 한다

■ NEXTVAL 및 CURRVAL 사용 규칙

- 다음과 같은 경우 NEXTVAL 및 CURRVAL 사용 가능
 - 서브쿼리에 속하지 않은 SELECT 문의 SELECT 목록
 - INSERT 문에 있는 서브쿼리의 SELECT 목록
 - INSERT 문의 VALUES 절
 - UPDATE 문의 SET 절
- 다음과 같은 경우에는 NEXTVAL 및 CURRVAL 사용 불가
 - 뷰의 SELECT 목록
 - DISTINCT 키워드가 있는 SELECT 문
 - GROUP BY, HAVING ,ORDER BY 절이 있는 SELECT 문
 - SELECT, DELETE, UPDATE 문의 서브 쿼리
 - CREATE TABLE , ALTER TABLE 문의 DEFAULT 표현식

■ 시퀀스 사용

위치 ID 2500에 “Support” 라는 이름의 새 부서를 추가

```
INSERT INTO departments (department_id, department_name,  
                        location_id)  
VALUES      (dept_deptid_seq.NEXTVAL, 'Support', 2500);
```

1 row created

DEPT_DEPTID_SEQ시퀀스의 현재 값을 볼 때

```
SELECT dept_deptid_seq.CURRVAL FROM dual;
```

시퀀스 사용

- 시퀀스 값을 메모리에 캐시하면 해당 값을 더 빠르게 액세스 할 수 있다
- 다음과 같은 경우 시퀀스 값 사이에 공백이 생긴다
 - 롤백이 발생하는 경우
 - 시스템이 고장난 경우
 - 시퀀스가 다른 테이블에서 사용되는 경우
- NOCACHE로 시퀀스를 생성한 경우 USER_SEQUENCES테이블을 질의하여 사용 가능한 다음 값을 볼 수 있다

- 증분 값, 최대값, 최소값, CYCLE 옵션 또는 CACHE 옵션을 변경

```
ALTER SEQUENCE      dept_deptid_seq  
                    INCREMENT BY 20  
                    MAXVALUE 999999  
                    NOCYCLE  
                    NOCACHE;
```

Sequence altered

시퀀스 수정에 대한 지침

- 시퀀스 소유자이거나 시퀀스에 대한 ALTER 권한이 있어야 함
- 이후 시퀀스 번호에만 영향을 준다
- 시퀀스를 다른 번호로 다시 시작하려면 시퀀스를 삭제 한 후 다시 생성 후 사용
- 일부 검증이 수행
 - 예를 들어, 새로운 MAXVALUE 값을 현재 시퀀스 번호보다 작게 지정할 수 없다.

시퀀스 제거

- DROP SEQUENCE문을 사용하여 데이터 디렉터리에서 제거
- 제거된 시퀀스는 더 이상 참조할 수 없다

```
DROP SEQUENCE dept_deptid_seq;
```

Sequence dropped

인덱스란?

- 스키마 객체이다
- Oracle server에서 포인터를 사용하여 행 검색 소도를 높이기 위해
- 데이터 위치를 빠르게 찾는 신속한 경로 액세스 방법을 사용 디스크 I/O를 줄여 준다
- 인덱스화된 테이블과 독립되어 존재
- Oracle server에 의해 사용되며 자동으로 유지 관리

■ 인덱스 생성 방법

자동: 테이블 정의에 PRIMARY KEY 또는 UNIQUE 제약 조건을 정의하면
고유 인덱스가 자동으로 생성

수동: 사용자가 열에 고유하지 않은 인덱스를 생성하여 행에 대한 액세스
시간을 줄일 수 있다

- 하나 이상의 열에 대해 인덱스를 생성

```
CREATE INDEX index  
ON table (column,...);
```

- EMPLOYEES 테이블의 LAST_NAME 열에 대한 질의 액세스 속도를 향상시킨다

```
CREATE INDEX emp_last_name_dix  
ON          employees(last_name);
```

Index created.

■ 인덱스 생성이 필요한 경우

- 열에 광범위한 값이 포함된 경우
- 열에 널 값이 많이 포함된 경우
- WHERE 절 또는 조인 조건에서 하나 이상의 열이 함께 자주 사용되는 경우
- 큰 테이블에서 대부분의 질의에 의해 검색되는 행이 2%~4% 미만인 경우

■ 인덱스를 생성하지 않아야 하는 경우

- 테이블이 작을 때
- 열이 질의의 조건으로 자주 사용되지 않는 경우
- 대부분의 질의가 테이블에 있는 행의 2%~4%이상을 검색할 경우
- 테이블이 자주 갱신되는 경우
- 인덱스화된 열이 표현식의 일부로 참조되는 경우

인덱스 확인

- USER_INDEXES 데이터 디렉터리 뷰는 인덱스 이름 및 고유성을 포함
- USER_IND_COLUMNS 뷰는 인덱스 이름, 테이블 이름 및 열 이름을 포함

```
SELECT ic.index_name, ic.column_name, ic.column_position  
       ix.uniqueness  
FROM   USER_INDEXES ix, USER_IND_COLUMNS ic  
WHERE  ic.index_name = ix.index_name  
AND    ic.table_name = 'EMPLOYEES' ;
```

인덱스 제거

- DROP INDEX 명령을 사용하여 데이터 디렉터리에서 제거

DROP INDEX index;

- 인덱스를 삭제하려면 인덱스 소유자 이거나 DROP ANY INDEX 권한이 있어야 한다.

동의어 (SYNONYM)

- 동의어(객체의 다른 이름)를 생성하여 객체 액세스를 단순화 한다
동의어를 사용하여 다음을 수행할 수 있다.
 - 다른 사용자가 소유한 테이블을 쉽게 참조
 - 긴 객체 이름을 짧게 한다.

CREATE [PUBLIC] SYNONYM synonym FOR object;

dept_sum_vu뷰의 간략한 이름을 생성

CREATE SYNONYM d_sum FOR dept_sum_vu;

동의어 삭제

DROP SYNONYM d_sum;

뷰 (view)

- 하나 이상의 테이블 (또는 다른 뷰)에서 원하는 데이터를 선택하여, 그들을 사용자 정의하여 나타낸 것

- 뷰 생성

CREATE [OR REPLACE] VIEW view_name [(column_aliases)] AS
defining-query

[WITH READ ONLY]

[WITH CHECK OPTION]

- WITH CHECK OPTION이 지정된 경우

- 뷰를 통해 삽입되거나 갱신되는 모든 행은 뷰 정의에 따라야 함 (WITH CHECK OPTION절을 사용하지 않고 뷰가 작성된 경우, 뷰에 대해 수행되는 삽입 및 갱신 조작이 뷰 정의를 따르는지 검사하지 않음)
- 뷰가 읽기 전용인 경우에는 이 옵션을 지정할 수 없음
- 뷰 정의는 subquery를 포함하지 않아야 함

뷰 (view)

Job01 SA_REP (Sales Representative)인 사원들의 뷰를 생성한다.

```
CREATE VIEW salesman AS  
SELECT *  
FROM EMPLOYEES  
WHERE JOB_ID = 'SA_REP';
```

■ 뷰 삭제

```
DROP VIEW view_name  
[CASCADE CONSTRAINT];
```

```
DROP VIEW salesman;
```



- DEPT 테이블의 기본 키 열에 사용할 시퀀스를 생성, 시퀀스 값은 200부터 시작하여 10씩 증가하며 최대 1000까지 가능. 시퀀스의 이름은 DEPT_ID_SEQ 로 지정하시오.
- 시퀀스 이름, 최대값 , 증가 값, 마지막 번호와 같은 시퀀스 정보를 표시하는 질의를 수행하시오.

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPARTMENTS_SEQ	9990	10	280
DEPT_ID_SEQ	1000	10	200
EMPLOYEES_SEQ	1.0000E+27	1	207
LOCATIONS_SEQ	9900	100	3300

- EMP 테이블의 외래 키 열(DEPT_ID)에 대해 비고유 인덱스를 생성
- 데이터 디렉터리에서 있는 DEP 테이블의 인덱스 및 고유성을 표시하는 질의를 수행하시오.



- EMPLOYEES 테이블에서 일부 데이터를 숨기려고 한다.
EMPLOYEES 테이블에서 직원 번호, 직원 이름 및 부서 번호를 가져와 EMPLOYEES_VU를 만드시오. 직원 이름의 컬럼명은 EMPLOYEE로 지정한다.
- EMPLOYEES_VU의 모든 데이터를 출력하시오.
- EMPLOYEES_VU를 사용하여 직원 이름 및 부서 번호를 구하는 쿼리를 작성하시오.
- 부서 50은 해당 부서에 속한 직원 데이터에 액세스해야 한다. 직원 번호, 직원 이름 및 부서 번호를 포함하는 DEPT50이라는 뷰를 생성한다. 보안상의 이유로 뷰를 통해 직원이 다른 부서로 이동하는 것을 허용하지 않도록 한다.
- DEPT50 뷰를 통해 Bell의 부서를 80으로 변경하도록 시도해본다.