Oracle Database

DB Objects

Oracle Main DB Objects

Object	용도
VIEW	하나 혹은 복수개의 테이블 혹은 뷰를 기반으로 한 논리적 테이블
INDEX	테이블 혹은 클러스터의 색인된 컬럼에 나타나는 각 값에 대한 항목을 포함하고 열에 대한 직접적이고 빠른 접근을 제공하는 객체
SEQUENCE	유일한 순차 값을 생성하는 스키마 객체
SYNONYM	테이블, 뷰, 시퀀스 혹은 프로그램 유닛의 별칭(alias)

View
: Create View

- Syntax

```
CREATE [OR REPLACE] [FORCE|NO FORCE] VIEW view_name
    [(alias[,alias]…)]
    AS Subquery
    [WITH READ ONLY]
    [WITH CHECK OPTION [CONSTRAINT constraint];
```

- View의 종류
  - Simple View
    - 단일 테이블을 기반으로 하며 함수나 expression으로 정의된 컬럼을 포함하지 않은 뷰
    - 몇 가지 제약 조건만 피하면 Update, Insert, Delete 등 DML 작업 수행이 가능
  - Complex View
    - 다수의 테이블을 기반으로 하며 함수나 데이터 그룹 등을 포함하고 있는 뷰
    - Update, Insert, Delete 등 DML 작업 수행 불가
  - Updatable Join View

View
: Create View

- Simple View 작성 예

```
CREATE OR REPLACE VIEW emp_80
    AS SELECT employee_id, first_name, last_name, manager_id, salary
        FROM employees
        WHERE department_id = 80
    WITH READ ONLY;
```

- 이렇게 작성된 뷰는 일반 테이블처럼 SELECT 할 수 있다

```
SELECT first_name || ' ' || last_name as name,
        salary
    FROM emp_80;

DESC emp_80;  # View 구조의 확인
```

View
: Create View / Drop View

- Complex View 작성 예

```
CREATE OR REPLACE VIEW emp_detail
    (employee_id, employee_name, manager_name, department_name)
    AS SELECT
            emp.employee_id,
            emp.first_name || ' ' || emp.last_name,
            man.first_name || ' ' || man.last_name,
            department_name
        FROM employees emp, employees man, departments dept
        WHERE emp.department_id = dept.department_id
            AND emp.manager_id = man.employee_id;
```

- View 를 삭제하고자 하면 DROP VIEW 문을 이용한다
  - VIEW에 대한 정의만 삭제되며 기반이 되는 테이블은 삭제되지 않는다

```
DROP VIEW emp_detail;
```

View
: Dictionary for Views

- USER_VIEWS
- USER_OBJECTS

```
SELECT view_name, text
    FROM USER_VIEWS;
```

```
SELECT object_name, created, status
    FROM user_objects
    WHERE object_type='VIEW';
```

- USER_OBJECTS의 status는 Base Table의 상태에 따라 변화한다
  Base Table에 변화가 있으면 그 테이블에 의존하는 View는 INVALID 상태로 바뀌고
  View에 대한 질의가 수행되는 순간 Oracle은 View를 다시 컴파일하고 VALID한 경우
  status를 변경한 후 질의를 수행한다

Index
: Create Index

- 인덱스는 데이터에 대한 조회 속도를 높이기 위해 만든다
- 데이터에 대한 처리 요청시 Oracle은 효율적인 처리를 위해 필요한 인덱스들을 찾아 이용
- 인덱스는 특정 row를 찾거나 일정한 range의 row를 찾을 때 유용
- 인덱스는 만들고 나면 Oracle 서버가 자동으로 사용하고 유지보수한다
  - 데이터에 대한 Update, Insert, Delete를 요청하면 연관 있는 인덱스에 자동으로 반영해준다
- 인덱스는 테이블과 독립적이므로 인덱스를 Drop 해도 테이블의 데이터에는 영향을 주지 않는다. 다만 해당 테이블의 DML 처리 속도가 달라질 수 있다

- Syntax

```
CREATE [UNIQUE] INDEX index_name
    ON table_name (Column|Expr[, Column|Expr]…);
```

Index
: Create Index / Drop Index

- hr.employees 테이블을 복사하여 새 테이블 s_emp를 만들고 employee_id 컬럼에 UNIQUE INDEX를 만들어 봅니다

```
CREATE TABLE s_emp
    AS SELECT * FROM employees;

CREATE UNIQUE INDEX s_emp_id_pk
    ON s_emp (employee_id);
```

- 인덱스를 삭제하고자 하면 DROP INDEX 문을 실행
  - Syntax

```
DROP INDEX schema.index_name;
```

  - Example

```
DROP INDEX s_emp_id_pk;
```

Index
: Dictionary for Indexes

- USER_INDEXES
- USER_IND_COLUMNS

- Example

```
SELECT  t.index_name, t.uniqueness, c.column_name, c.column_position
    FROM user_indexes t, user_ind_columns c
    WHERE c.index_name = t.index_name
        AND t.table_name = 'EMP';
```

- Index가 필요한 경우
  - WHERE 절이나 JOIN 조건에 빈번하게 사용되는 컬럼
  - 많은 데이터를 담고 있으나 적은 수의 row들만 받아오는 쿼리 수행
  - 넓은 범위의 값을 포함하거나 null 값이 많은 컬럼들
  - 자주 업데이트 되지 않는 테이블

Sequence
: Create Sequence

- 시퀀스는 유일한 정수값을 발생시키는 객체
- 주로 Primary Key 값을 자동으로 발생시키고자 할 때 생성한다

- Syntax

```
CREATE SEQUENCE sequence_name
      [INCREMENT BY n]    #   증가값
      [START WITH n]      #   시작값
      [{MAXVALUE n|NOMAXVALUE}] #   최대값
      [{MINVALUE n|NOMINVALUE}] # 최소값
      [{CYCLE|NOCYCLE}]
      [{CACHE n|NOCACHE}]    #   Oracle Server가 미리 캐시해 놓을 개수
```

  - NOMAXVALUE가 default : 10의 27승
  - NOMINVALUE가 default : 1
  - CACHE n : 기본값은 20

Sequence
: Create Sequence

- author 테이블을 만들고, PK에 사용할 Sequence를 만들어 봅니다

```
CREATE TABLE author (
  id          NUMBER(10),
  name        VARCHAR2(50) NOT NULL,
  bio         VARCHAR2(500),
  PRIMARY KEY(id)
);

CREATE SEQUENCE seq_author_id
         START WITH 1
         INCREMENT BY 1
         MAXVALUE 1000000;
```

Sequence
: Using Sequence

- Sequence 값을 SQL 문장에서 사용하기 위해서는 아래 pseudo 컬럼들을 이용
  - CURRVAL : 지정 시퀀스의 현재 값을 알아낸다
  - NEXTVAL : 지정 시퀀스의 값을 증가시키고 해당 값을 반환한다

- Example

```
INSERT INTO author (id, name)
       VALUES(seq_author_id.NEXTVAL, 'Steven King');

SELECT seq_author_id.CURRVAL FROM dual;
```

Sequence
: Alter Sequence / Drop Sequence

- Sequence의 증가값, 최소값, 최대값, 캐시 수 등을 변경하려면 ALTER SEQUENCE 문을 사용한다

- Syntax

```
ALTER SEQUENCE sequence_name
      [INCREMENT BY n]    #   증가값
      [{MAXVALUE n|NOMAXVALUE}] #   최대값
      [{MINVALUE n|NOMINVALUE}] # 최소값
      [{CYCLE|NOCYCLE}]
      [{CACHE n|NOCACHE}]    #   Oracle Server가 미리 캐시해 놓을 개수
```

- 시퀀스를 삭제하려면 DROP SEQUENCE 문을 이용한다

```
DROP SEQUENCE schema.index_name;
```

Sequence
: Dictionary for Sequence

- USER_SEQUENCES
- USER_OBJECTS

- Example

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

```
SELECT object_name
    FROM user_objects
    WHERE object_type='SEQUENCE';
```