13.1 기타 스키마 객체

- 뷰 (VIEW) 정의

: 테이블 또는 다른 뷰를 기초로 하는 논리적 테이블.
: 뷰는 그 자체로서 소유하는 데이터는 없지만, 창문처럼 창문을 통해 어떤 데이터를 보거나 변경할 수 있다.
: 뷰에서 참조하는 테이블을 기본 테이블(Base Table) 이라고 한다.

- 뷰 (VIEW) 사용 목적

: 데이터베이스에서 선택적으로 데이터를 보여줄 수 있기 때문에, 데이터베이스에 대한 접근을 제한 할 수 있다.
: 복잡한 질의로부터 결과를 검색하기 위한 단순한 질의를 만들 수 있다.
: 사용자와 애플리케이션 프로그램에 대한 데이터 독립성을 제공한다.
: 하나의 뷰는 여러 개의 테이블로부터 데이터를 검색하는데 사용 가능하다.

- 뷰 (VIEW) 종류

특징	단순 뷰	복합 뷰
테이블 갯수	1개	1개 이상
함수 포함 여부	포함하지 않음	포함
그룹 연산 포함 여부	포함하지 않음	포함
뷰를 통한 DML 작업 가능 여부	가능	부분적으로 가능

13.2 기타 스키마 객체

- 뷰 (VIEW) 작성법

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
      [(alias[, alias] ...)]
AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

- 뷰 (VIEW) 작성시 주의할 점.

  : subquery는 조인, set 연산, 서브쿼리가 포함된 복잡한 SELECT 문이 정의 가능하다.
  : subquery에는 ORDER BY 절을 사용할 수 없다. ORDER BY를 사용하려면 검색시 뷰에 기술한다.
  : 뷰를 수정하기 위해서는 CREATE OR REPLACE 을 이용한다.
  : CREATE VIEW 권한을 가져야 뷰를 생성할 수 있다.

- CREATE VIEW 권한 할당

```
SQL> conn / as sysdba
연결되었습니다.
SQL> GRANT CREATE VIEW TO SCOTT;

권한이 부여되었습니다.

SQL> conn scott/tiger
연결되었습니다.
```

13.3 기타 스키마 객체

- 뷰 (VIEW) 생성

```
SQL> CREATE VIEW EMP_VIEW
  2  AS
  3  SELECT EMPNO, ENAME, SAL, HIREDATE
  4  FROM EMP
  5  WHERE DEPTNO = 10;

뷰가 생성되었습니다.
```

```
SQL> SELECT *
  2  FROM EMP_VIEW;

     EMPNO ENAME                       SAL HIREDATE
---------- -------------------- ---------- --------
      7782 CLARK                      2450 81/06/09
      7839 KING                       5000 81/11/17
      7934 MILLER                     1300 82/01/23
```

```
SQL> CREATE VIEW DEPT_VIEW
  2  AS
  3  SELECT DEPTNO NO , DNAME name
  4  FROM DEPT;

뷰가 생성되었습니다.

SQL> DESC DEPT_VIEW;
 이름                                      널?      유형
 ----------------------------------------- -------- ------------
 NO                                        NOT NULL NUMBER(2)
 NAME                                               VARCHAR2(14)
```

```
SQL> SELECT VIEW_NAME
  2  FROM USER_VIEWS;

VIEW_NAME
------------------------
EMP_VIEW
DEPT_VIEW
```

13.4 기타 스키마 객체

- 뷰 (VIEW) 수정
  : CREATE OR REPLACE 명령 이용한다.

```
SQL> CREATE OR REPLACE VIEW DEPT_VIEW
  2  AS
  3  SELECT DEPTNO NO, DNAME NAME, LOC
  4  FROM DEPT;

뷰가 생성되었습니다.

SQL> DESC DEPT_VIEW;
 이름                                      널?      유형
 ----------------------------------------- -------- ------------

 NO                                        NOT NULL NUMBER(2)
 NAME                                               VARCHAR2(14)
 LOC                                                VARCHAR2(13)
```

- 복합 뷰 (VIEW) 생성

```
SQL> CREATE VIEW EMP_DEPT_VIEW
  2  AS
  3  SELECT EMPNO, ENAME , DNAME
  4  FROM EMP , DEPT
  5  WHERE EMP.DEPTNO = DEPT.DEPTNO
  6  AND DEPT.DEPTNO = 30;

뷰가 생성되었습니다.
```

```
SQL> SELECT * FROM EMP_DEPT_VIEW;

     EMPNO ENAME                DNAME
---------- -------------------- ------
      7499 ALLEN                SALES
      7521 WARD                 SALES
      7654 MARTIN               SALES
      7698 BLAKE                SALES
      7844 TURNER               SALES
      7900 JAMES                SALES
```

13.5 기타 스키마 객체

- 뷰 (VIEW) 에서 DML 작업
  : 단순 뷰에서 DML 연산을 수행 가능 하다.
  :뷰가 다음을 포함한다면 행을 제거할 수 없다.
  - 그룹함수
  - GROUP BY 절
  - DISTINCT 키워드

  : 뷰가 다음을 포함한다면 데이터를 수정할 수 없다.
  - 위의 임의의 조건
  - ROWNUM 의사열
  - 표현식으로 정의된 열 (예: SAL*12)

  : 뷰가 다음을 포함한다면 데이터를 추가할 수 없다.
  - 위의 임의의 조건
  - 뷰에 의해 선택되지 않은 NOT NULL 열이 기본테이블에 있을때

```
SQL> SELECT * FROM EMP_VIEW;

     EMPNO ENAME                          SAL HIREDATE
---------- ------------------ ---------- --------
      7782 CLARK                      2450 81/06/09
      7839 KING                       5000 81/11/17
      7934 MILLER                     1300 82/01/23
```

```
SQL> DELETE FROM EMP_VIEW
  2  WHERE EMPNO = 7782;

1 행이 삭제되었습니다.
```

```
SQL> SELECT * FROM EMP_VIEW;

     EMPNO ENAME                          SAL HIREDATE
---------- ------------------ ---------- --------
      7839 KING                       5000 81/11/17
      7934 MILLER                     1300 82/01/23
```

13.6 기타 스키마 객체

- 뷰 (VIEW) 의 제약 조건
  : WITH CHECK OPTION
  - WHERE 조건에 만족하는 데이터만이 INSERT , UPDATE 작업을 수행할 수 있다.

```
SQL> CREATE OR REPLACE VIEW EMP_VIEW
  2  AS
  3  SELECT * FROM EMP
  4  WHERE DEPTNO =10
  5  WITH CHECK OPTION CONSTRAINT EMP_VIEW10_CHCEK;
```

```
SQL> UPDATE EMP_VIEW
  2  SET DEPTNO = 20 WHERE ENAME = 'KING';
UPDATE EMP_VIEW
       *
1행에 오류:
ORA-01402: 뷰의 WITH CHECK OPTION의 조건에 위배 됩니다
```

  : WITH READ ONLY
  - 뷰를 통한 DML 작업은 불가능하다.

```
SQL> CREATE OR REPLACE VIEW  DEPT_VIEW
  2  AS
  3  SELECT * FROM DEPT
  4  WHERE DEPTNO = 10
  5  WITH READ ONLY;
```

```
SQL> DELETE FROM DEPT_VIEW
  2  WHERE DEPTNO  = 10;
DELETE FROM DEPT_VIEW
            *
1행에 오류:
ORA-01752: 뷰으로 부터 정확하게 하나의 키-보전된 테이블 없이 삭제할 수 없습니다
```

13.7 기타 스키마 객체

- 뷰 (VIEW) 제거
  : 기본 테이블을 기반으로 하기 때문에 데이터 손실 없이 뷰를 삭제한다.
  : 뷰 삭제는 뷰가 만들어진 기본 테이블에는 영향을 미치지 않는다.

```
DROP VIEW view
```

```
SQL> DROP VIEW DEPT_VIEW;

뷰가 삭제되었습니다.
```

13.8 기타 스키마 객체

- 시퀀스 (SEQUENCE) 정의

: 여러 사용자들이 공유하는 데이터베이스 객체로서, 호출 될 때마다 중복되지 않은 고유한 숫자를 리턴하는 객체이다.

: 중복되지 않는 기본키 컬럼에 사용할 값을 발생시키는데 주로 사용한다.

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

```
SQL> CREATE SEQUENCE  EMP_SEQ
  2  INCREMENT BY 1
  3  START WITH 100
  4  MAXVALUE 9999
  5  NOCACHE
  6  NOCYCLE;

시퀀스가 생성되었습니다.
```

```
SQL> SELECT SEQUENCE_NAME , MIN_VALUE, MAX_VALUE, INCREMENT_BY
  2  FROM USER_SEQUENCES;

SEQUENCE_NAME          MIN_VALUE  MAX_VALUE INCREMENT_BY
-------------------- ---------- ---------- ------------
EMP_SEQ                        1       9999            1
```

13.9 기타 스키마 객체

- 시퀀스 (SEQUENCE) 사용

: NEXTVAL
- 지정된 시퀀스에서 순차적인 시퀀스 번호를 추출할 때 사용.
- 시퀀스명.NEXTVAL

: CURRVAL
- 사용자가 방금 추출한 시퀀스 번호를 참조할 때 사용.
- 시퀀스명.CURRVAL
- 반드시 NEXTVAL에 의해서 번호를 추출한 후에 사용해야 한다.

```
SQL> INSERT INTO EMP
  2  VALUES ( EMP_SEQ.NEXTVAL , '홍길동','인사', NULL , SYSDATE,
  3   2500, 300, 40 );

1 개의 행이 만들어졌습니다.

SQL> SELECT EMPNO, ENAME, JOB
  2  FROM EMP
  3  WHERE DEPTNO = 40;

     EMPNO ENAME                JOB
---------- -------------------- ---------
       100 홍길동               인사
```

```
SQL> SELECT EMP_SEQ.CURRVAL FROM DUAL;

   CURRVAL
----------
       100
```

13.10 기타 스키마 객체

- 시퀀스 (SEQUENCE) 변경
  : 증분, 최대값, 최소값, 순환여부 , 캐시여부를 변경할 수 있다.
  : 시퀀스가 변경되면 다음번 시퀀스 번호 추출부터 변경사항이 적용된다.
  : START WITH 옵션은 변경 불가이며 , 시퀀스를 삭제하고 재생성해야 한다.
  : MAXVALUE 값은 현재 시퀀스 번호보다 큰 번호로 지정해야 한다.

```
SQL> ALTER SEQUENCE EMP_SEQ
  2   INCREMENT BY 2
  3   MAXVALUE 10000
  4   NOCACHE
  5   NOCYCLE;

시퀀스가 변경되었습니다.
```

- 시퀀스 (SEQUENCE) 삭제

```
SQL> DROP SEQUENCE EMP_SEQ;

시퀀스가 삭제되었습니다.
```

13.11 기타 스키마 객체

- 인덱스 (INDEX) 정의

: 테이블에서 행을 검색할 때 검색 속도를 높이기 위해 Oracle 서버가 사용하는 스키마 객체이다.
: 인덱스 없이 데이터를 검색하면 테이블의 모든 데이터를 읽어 데이터를 선별한다.
: 인덱스를 사용하면 디스크의 I/O 를 감소시킬 수 있다.
: 해당 테이블과 논리적으로 독립적이다.
: Oracle 서버에 의해 자동으로 사용 및 관리된다.
반면에 테이블을 삭제하면 관련 인덱스는 자동으로 삭제된다.

- 인덱스 (INDEX) 생성

1. 자동 생성
: PRIMARY KEY , UNIQUE 제약 조건 지정 시 UNIQUE INDEX 가 자동 생성 된다.

2. 수동 생성
: 한 개 컬럼 또는 여러 컬럼을 이용하여 인덱스 생성 가능하다.

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

```
SQL> CREATE INDEX EMP_ENAME_IDX
  2  ON EMP( ENAME );

인덱스가 생성되었습니다.
```

13.12 기타 스키마 객체

- 인덱스(INDEX)를 사용할 컬럼 선정

: 값의 범위가 넓은 컬럼 (즉, 컬럼내의 값이 다양할수록 좋다)
: NULL 값이 많은 컬럼 (NULL 값은 인덱스에 포함되지 않기 때문에 인덱스 크기가 감소)
: WHERE절 또는 JOIN 조건에 사용되는 컬럼
: 테이블이 크고 대부분의 쿼리 문장이 테이블내 전체 데이터의 약 2 ~ 4% 이내를 검색하는 경우

* 인덱스가 반드시 성능을 향상시키는 것은 아니다. 테이블에 DML 작업을 수행하면 관련 인덱스도 변경되어야 하므로 오히려 속도가 저하될 수도 있다.

- 인덱스(INDEX) 작성할 필요 없는 경우

: 테이블이 작은 경우
: 쿼리 문장의 조건에 자주 사용되지 않는 컬럼.
: 대부분의 쿼리문장이 테이블내 전체 데이터의 약 2 ~ 4 % 이상을 검색하는 경우
: 테이블이 자주 변경되는 경우
: 인덱스가 작성된 컬럼이 쿼리문장의 조건에서 표현식에 포함된 경우

13.13 기타 스키마 객체

- 인덱스(INDEX) 확인

```
SQL> SELECT IC.INDEX_NAME, IC.COLUMN_NAME,
  2  IC.COLUMN_POSITION, IX.UNIQUENESS
  3  FROM USER_INDEXES IX, USER_IND_COLUMNS IC
  4  WHERE IC.INDEX_NAME = IX.INDEX_NAME
  5  AND IC.TABLE_NAME = 'EMP';

INDEX_NAME           COLUMN_NAME                    COLUMN_POSITION UNIQUENES
-------------------- ------------------------------ --------------- ---------
PK_EMP               EMPNO                                        1 UNIQUE
EMP_ENAME_IDX        ENAME                                        1 NONUNIQUE
```

- 인덱스(INDEX) 삭제

```
SQL> DROP INDEX EMP_ENAME_IDX;

인덱스가 삭제되었습니다.
```

13.14 기타 스키마 객체

- 동의어 (SYNONYM) 정의 및 특징

: 동의어는 객체에 대한 별칭이다.
: 객체에 대한 접근방법을 단순화 할 수 있다.
: 객체이름의 길이 단축

```
CREATE [PUBLIC] SYNONYM synonym
FOR object;
```

```
SQL> conn / as sysdba
연결되었습니다.
SQL> GRANT CREATE SYNONYM TO SCOTT;

권한이 부여되었습니다.
```

```
SQL> CREATE SYNONYM D_SYN
  2  FOR DEPT;

동의어가 생성되었습니다.

SQL> SELECT * FROM D_SYN;

    DEPTNO DNAME                        LOC
---------- ---------------------------- ----------
        10 ACCOUNTING                   NEW YORK
        20 RESEARCH                     DALLAS
        30 SALES                        CHICAGO
        40 OPERATIONS                   BOSTON
```

```
SQL> DROP SYNONYM D_SYN;

동의어가 삭제되었습니다.
```

13.15 기타 스키마 객체

- CREATE PUBLIC 정의 및 특징

  : 다른 사용자들도 같이 사용 할 수 있는 공용 동의어가 생성된다.
  : DBA 권한을 갖는 사용자만이 공용 동의어를 생성할 수 있다.
  : 또한, 다른 사용자들은 공용 동의어에 의해 참조되는 테이블에 접근 할 수 있는 권한을 부여 받아야 한다.

```
SQL> CONN / as sysdba
연결되었습니다.
SQL> CREATE PUBLIC SYNONYM HR_EMP
  2  FOR HR.EMPLOYEES;

동의어가 생성되었습니다.

SQL> GRANT SELECT ON HR.EMPLOYEES TO SCOTT;

권한이 부여되었습니다.
```

```
SQL> CONN SCOTT/TIGER
연결되었습니다.
SQL>
SQL> SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME
  2  FROM HR_EMP
  3  WHERE FIRST_NAME = 'Steven';

EMPLOYEE_ID FIRST_NAME LAST_NAME
----------- ---------- -------------------------
        100 Steven     King
        128 Steven     Markle
```