Chapter 12. 뷰(View)의 작성

이번 장에서는 뷰(View)를 작성하고, 뷰를 통하여 데이터를 검색, 입력, 변경, 삭제하는 방법을 알아본다. 또한, FROM 절 뒤에 기술되는 서브쿼리인 인라인(Inline) 뷰를 작성하고 TOP-N 분석을 수행하는 방법을 설명한다.

뷰의 용도 및 종류

테이블에 생성된 뷰는 데이터의 논리적 부분집합으로 표현 할 수 있으며, 기존 테이블 또는 뷰에 기초한 논리적 테이블이다. 뷰는 내부에 데이터를 저장하고 있지 않지만 기존 테이블에 대한 윈도우와 같아서 데이터를 검색하거나 변경 할 수 있다. 뷰에서 참조하는 테이블을 기본 테이블이라고 하며, 뷰에는 SELECT 문장이 저장되어 있다.

뷰를 사용하는 목적은 다음과 같다.
- 뷰는 지정된 컬럼 및 행만을 보여주기 때문에 데이터의 접근을 제한 할 수 있다.
- 복잡한 쿼리 문장을 뷰에 저장함으로서 쿼리 문장을 단순하게 만들 수 있다. 예를 들어, 복잡한 조인 문장을 뷰에 저장함으로서 검색시 매번 조인 문장을 작성 할 필요가 없어진다.
- 뷰는 임의 사용자 및 애플리케이션 프로그램에 대하여 데이터 독립성을 제공한다. 한 개의 뷰는 여러 개의 테이블로부터 데이터를 검색하는데 사용 될 수 있다.

뷰는 뷰 내에 정의된 SELECT 문장이 검색하는 테이블의 개수에 따라 단순 뷰와 복합 뷰로 분류된다.

표 12-1. 뷰의 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

뷰의 작성 및 검색

뷰를 작성하는 CREATE VIEW 문장내에는 서브쿼리가 포함되며 문법은 다음과 같다.

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

뷰를 작성 할 때는 다음 사항에 주의하여야 한다.

- 뷰에 포함되는 서브쿼리는 조인, 집계 연산, 서브쿼리가 포함된 복잡한 SELECT 문장이 정의 될 수 있다.
- 뷰에 포함되는 서브쿼리에는 ORDER BY 절을 사용 할 수 없다. ORDER BY를 사용하려면 검색시 뷰에 기술하도록 한다.
- WITH CHECK OPTION으로 생성된 뷰에 제약조건 이름을 지정하지 않으면 SYS_C*n* 형식으로 저장된다.
- 뷰를 삭제한 후 다시 생성하는 대신 OR REPLACE 옵션으로 뷰를 변경하면, 기존의 객체 권한을 재부여 할 필요가 없다.

부서번호가 10번인 직원들의 정보를 검색하는 EMPVU10을 작성하면 다음과 같다.

```
SQL> CREATE VIEW EMPVU10
  2  AS SELECT EMPNO, ENAME, SAL
  3  FROM EMP
  4  WHERE DEPTNO = 10;

뷰가 생성되었습니다.

SQL> SELECT * FROM EMPVU10;

     EMPNO ENAME             SAL
---------- ---------- ----------
      7782 CLARK            2450
      7839 KING             5000
      7934 MILLER           1300
```

뷰의 구조를 확인하면 다음과 같다.

```
SQL> DESC EMPVU10
 이름                                      널?      유형
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(4)
 ENAME                                              VARCHAR2(10)
 SAL                                                NUMBER(7,2)
```

컬럼 별칭을 이용하여 뷰를 생성하면, 뷰를 검색 할 때 지정된 컬럼 별칭을 사용해야 한다.

```
SQL> CREATE VIEW SALVU10
  2  AS SELECT EMPNO ID, ENAME NAME, SAL*12 YEAR_SAL
  3  FROM EMP
  4  WHERE DEPTNO = 10;

뷰가 생성되었습니다.

SQL> SELECT * FROM SALVU10;

        ID NAME       YEAR_SAL
---------- ---------- ----------
      7782 CLARK           29400
      7839 KING            60000
      7934 MILLER          15600
```

위 문장은 아래 문장과 동일한 뷰를 생성한다.

```
CREATE VIEW SALVU10 (ID, NAME, YEAR_SAL)
AS SELECT EMPNO, ENAME, SAL*12
FROM EMP
WHERE DEPTNO = 10;

뷰가 생성되었습니다.
```

뷰를 검색하는 방법은 테이블을 검색하는 방법과 동일하며 내부적인 처리과정은 다음과 같다.

1. USER_VIEWS 딕셔너리 테이블로부터 뷰의 정의를 검색한다.
2. 뷰가 참조하는 테이블에 대한 접근 권한을 확인한다.
3. 뷰 쿼리가 기본 테이블을 참조하는 쿼리로 변환된다. 즉, 기본 테이블로부터 데이터가 검색, 변경이 이루어진다.

뷰를 이용한 검색은 다음과 같다.

```
SQL> SELECT * FROM SALVU10
  2  WHERE YEAR_SAL > 20000;

        ID NAME         YEAR_SAL
---------- ---------- ----------
      7782 CLARK           29400
      7839 KING            60000
```

뷰를 변경하려면 CREATE OR REPLACE VIEW 명령을 사용하며 컬럼 별칭을 추가할 수 있다.

```
SQL> CREATE OR REPLACE VIEW EMPVU10 (ID, NAME, JOB, SALARY)
  2  AS SELECT EMPNO, ENAME, JOB, SAL
  3  FROM EMP
  4  WHERE DEPTNO = 10;

뷰가 생성되었습니다.
```

집계 함수나 조인 문장이 포함된 복합 뷰를 생성하는 방법은 다음과 같다.

```
SQL> CREATE VIEW DEPT_AVG_VU (NAME, AVG_SAL)
  2  AS SELECT D.DNAME, AVG(E.SAL)
  3  FROM DEPT D, EMP E
  4  WHERE D.DEPTNO = E.DEPTNO
  5  GROUP BY D.DNAME;

뷰가 생성되었습니다.

SQL> SELECT * FROM DEPT_AVG_VU;

NAME              AVG_SAL
-------------- ----------
ACCOUNTING     2916.66667
RESEARCH             2175
SALES          2111.11111
```

뷰를 이용한 데이터 변경

단순 뷰에는 DML 작업이 가능하지만 아래와 같은 경우에는 DML 작업이 불가능하다.

■ 삭제가 불가능한 경우
서브쿼리에 다음과 같은 함수 또는 구문이 사용되면 뷰를 통하여 행의 삭제가 불가능하다.
· 집계 함수
· GROUP BY 구문
· DISTINCT 키워드
· ROWNUM과 같은 가상 컬럼

■ 변경이 불가능한 경우
서브쿼리에 다음과 같은 함수 또는 구문이 사용되면 뷰를 통하여 행의 변경이 불가능하다.
· 집계 함수
· GROUP BY 구문
· DISTINCT 키워드
· ROWNUM과 같은 가상 컬럼
· 표현식으로 정의된 컬럼 (예, SAL*12)

■ 입력이 불가능한 경우
서브쿼리에 다음과 같은 함수 또는 구문이 사용되면 뷰를 통하여 행의 변경이 불가능하다.
· 집계 함수
· GROUP BY 구문
· DISTINCT 키워드
· ROWNUM과 같은 가상 컬럼
· 표현식으로 정의된 컬럼
· 기본 테이블에 뷰에서 선택되지 않은 NOT NULL 컬럼이 있는 경우

WITH CHECK OPTION

뷰에 정의된 쿼리의 WHERE 절에 WITH CHECK OPTION 옵션을 사용하면 WHERE 조건에 만족하는 데이터만이 INSERT, UPDATE 작업을 수행 할 수 있다. 아래와 같이 변경하고자하는 데이터가 WHERE 조건에 만족하지 않으면 오류를 발생시킨다.

```
SQL> CREATE OR REPLACE VIEW EMPVU10
  2  AS SELECT *
  3  FROM EMP
  4  WHERE DEPTNO = 10
  5  WITH CHECK OPTION CONSTRAINT EMPVU10_CK;

뷰가 생성되었습니다.

SQL> UPDATE EMPVU10 SET DEPTNO = 20 WHERE ENAME = 'MILLER';
UPDATE EMPVU10 SET DEPTNO = 20 WHERE ENAME = 'MILLER'
       *
1행에 오류:
ORA-01402: 뷰의 WITH CHECK OPTION의 조건에 위배 됩니다
```

WITH READ ONLY

뷰에 정의된 쿼리의 WHERE 절에 WITH READ ONLY 옵션을 추가하면 뷰를 통한 DML 작업이 불가능하다. 만약, DML 작업을 수행하려고 하면 Oracle 서버는 오류를 발생시킨다.

```
SQL> CREATE OR REPLACE VIEW EMPVU10
  2  AS SELECT *
  3  FROM EMP
  4  WHERE DEPTNO = 10
  5  WITH READ ONLY;

뷰가 생성되었습니다.

SQL> DELETE FROM EMPVU10
  2  WHERE ENAME = 'SMITH';
DELETE FROM EMPVU10
            *
1행에 오류:
ORA-01752: 뷰으로 부터 정확하게 하나의 키-보전된 테이블 없이 삭제할 수 없습니다
```

뷰의 삭제

뷰를 삭제하는 방법은 다음과 같다.

```
DROP VIEW view
```

```
SQL> DROP VIEW EMPVU10;

뷰가 삭제되었습니다.
```

인라인 뷰

FROM 절의 서브쿼리를 인라인 뷰라고 하며 별칭을 부여 할 수 있으며, 서브쿼리의 결과는 메인 쿼리에서 참조된다. 다음은 인라인 뷰를 이용하여 부서별 최대 급여를 출력하는 SQL 문장이다.

```
SQL> SELECT D.DNAME, E.MAXSAL
  2  FROM DEPT D, (SELECT DEPTNO, MAX(SAL) MAXSAL
  3                FROM EMP
  4                GROUP BY DEPTNO) E
  5  WHERE D.DEPTNO = E.DEPTNO;

DNAME              MAXSAL
-------------- ----------
ACCOUNTING           5000
RESEARCH             3000
SALES                3500
```

TOP-N 분석

TOP-N 쿼리란 컬럼 값 중에서 가장 큰 값 또는 가장 작은 값 *n* 개를 질의하는 경우에 사용되는 SQL 문장이다. 예를 들면, 급여가 가장 높은 5명의 직원을 검색하거나 입사일이 가장 늦은 5명의 직원을 검색하는 경우이다. TOP-N 쿼리의 형식은 다음과 같다.

```
SELECT [column_list], ROWNUM
FROM (SELECT [column_list]
      FROM table
      ORDER BY Top-N_column)
WHERE ROWNUM <= N;
```

위에서 보듯이 인라인 뷰에 검색하고자 하는 컬럼을 ORDER BY 구문을 이용하여 정렬한 다음, 메인 쿼리의 결과에서 필요한 행의 개수 만큼 행을 출력하면 된다. 쿼리의 결과를 임의 개수만 출력하려면 가상 컬럼인 ROWNUM을 사용하면 되는데, ROWNUM 컬럼은 테이블에 실제 존재하지는 않지만 쿼리 실행 결과를 출력할 때 출력순서대로 행의 번호를 붙여주는 컬럼이다.

사원 중에서 급여를 가장 많이 받는 5명의 직원을 출력하면 다음과 같다.

```
SQL> SELECT ROWNUM, ENAME, SAL
  2  FROM (SELECT ENAME, SAL
  3        FROM EMP
  4        ORDER BY SAL DESC)
  5  WHERE ROWNUM <= 5;

    ROWNUM ENAME             SAL
---------- ---------- ----------
         1 KING             5000
         2 SCOTT            3000
         3 FORD             3000
         4 JONES            2975
         5 BLAKE            2850
```

복습

1. 부서명과 사원명을 출력하는 뷰 DNAME_ENAME_VU를 작성하시오.

2. 사원명과 사수명을 출력하는 뷰 WORKER_MANAGER_VU를 작성하시오.

3. 사원 테이블에서 사번, 사원명, 입사일을 입사일이 늦은 사원 순으로 정렬하시오.

4. 사원 테이블에서 사번, 사원명, 입사일을 입사일이 늦은 사원 5명을 출력하시오.

5. 사원 테이블에서 사번, 사원명, 입사일을 입사일이 6번째로 늦은 사원부터 10번째 사원까지 출력하시오.