

## 14장 가상 테이블인 뷰

### 14.1. 뷰의 기본 다루기

- 뷰(View)는 한마디로 물리적인 테이블을 근거한 논리적인 가상 테이블이라고 정의할 수 있다.
- 가상이란 단어는 실질적으로 데이터를 저장하고 있지 않기 때문에 붙인 것이고, 테이블이란 단어는 실질적으로 데이터를 저장하고 있지 않더라도 사용자는 마치 테이블을 사용하는 것과 동일하게 뷰를 사용할 수 있기 때문에 붙인 것이다.
- 뷰는 기본 테이블에서 파생된 객체로서 기본 테이블에 대한 하나의 쿼리문이다.
- 뷰(View)란 '보다'란 의미를 갖고 있는 점을 감안해 보면 알 수 있듯이 실제 테이블에 저장된 데이터를 뷰를 통해서 볼 수 있도록 한다.
- 사용자에게 주어진 뷰를 통해서 기본 테이블을 제한적으로 사용하게 된다.

#### 14.1.1. 뷰의 기본 테이블

- 뷰는 이미 존재하고 있는 테이블에 제한적으로 접근하도록 한다.
- 뷰를 생성하기 위해서는 실질적으로 데이터를 저장하고 있는 물리적인 테이블이 존재해야 하는데 이 테이블을 기본 테이블이라고 한다.

```
-- 뷰의 기본 테이블 생성하기
DROP TABLE DEPT_COPY;
CREATE TABLE DEPT_COPY
AS
SELECT * FROM DEPT;

DROP TABLE EMP_COPY;
CREATE TABLE EMP_COPY
AS
SELECT * FROM EMP;

SELECT * FROM DEPT_COPY;
SELECT * FROM EMP_COPY;
```

#### 14.1.2. 뷰 정의하기

- 테이블을 생성하기 위해서 CREATE TABLE 로 시작하지만, 뷰를 생성하기 위해서는 CREATE VIEW로 시작한다. AS 다음은 마치 서브 쿼리문과 유사한다.

```
-- 형식
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW view_name
[(alias, alias, alias, ...)]
AS subquery
[WITH CHECK OPTION]
[WITH READ ONLY];

-- 뷰를 생성할 권한이 불충분한 경우
CREATE OR REPLACE VIEW EMP_VIEW30
AS
SELECT EMPNO, ENAME, SAL, DEPTNO
FROM EMP_COPY
WHERE DEPTNO=30; -- SQL Error: ORA-01031: insufficient privileges
```

```

CONN system/sys
GRANT CREATE VIEW TO scott; -- system계정으로 로그인하여 뷰를 생성할 수 있는 권한을 부여한다.
CONN scott/tiger

SELECT * FROM EMP_VIEW30;

-- 뷰 정의하기
DROP VIEW EMP_VIEW30;
CREATE VIEW EMP_VIEW30
AS
SELECT EMPNO, ENAME, DEPTNO
FROM EMP_COPY
WHERE DEPTNO=30;

DESC EMP_VIEW30

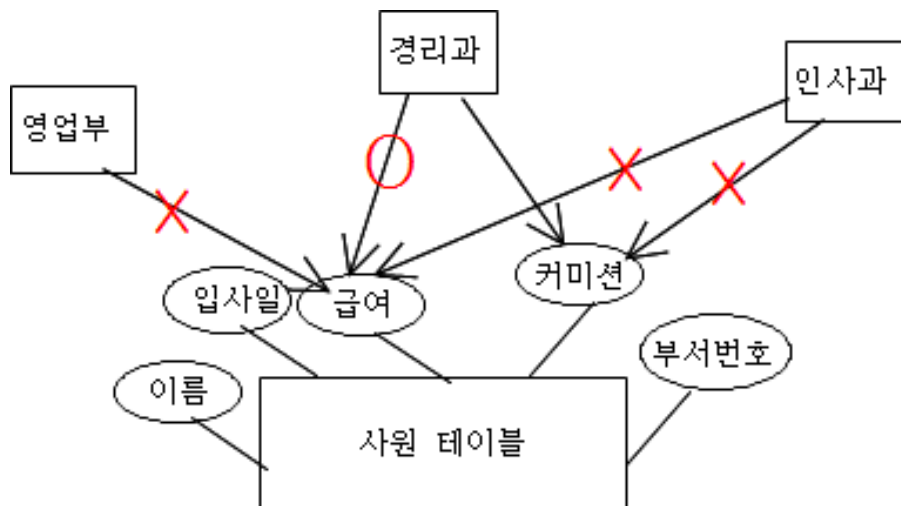
SELECT * FROM EMP_VIEW30;

```

## 14.2. 뷰 고급 다루기

### 14.2.1. 뷰를 사용하는 이유

- 뷰를 사용하는 이유는 두 가지로 설명할 수 있다.
  - 복잡하고 긴 쿼리문을 뷰로 정의하면 접근을 단순화시킬 수 있다.
  - 보안에 유리하다.
- 사용자마다 특정 객체만 조회할 수 있도록 권한을 부여를 할 수 있기에 동일한 테이블을 접근하는 사용자마다에 따라 서로 다르게 보도록 여러 개의 뷰를 정의해 놓고 특정 사용자만이 해당 뷰에 접근할 수 있도록 한다.



### 14.2.2. 단순뷰와 복합뷰

- 뷰는 뷰를 정의하기 위해서 사용되는 기본 테이블의 수에 따라 단순 뷰(Simple View)와 복합 뷰(Complex View)로 나뉜다.

단순 뷰	복합 뷰
하나의 테이블로 생성	여러 개의 테이블로 생성
그룹 함수의 사용이 불가능	그룹 함수의 사용이 가능

DISTINCT 사용이 불가능	DISTINCT 사용이 가능
DML 사용 가능	DML 사용 불가능

```
-- 단순 뷰의 컬럼에 별칭 부여하기
DESC EMP_VIEW30;
CREATE OR REPLACE
VIEW EMP_VIEW(사원번호, 사원명, 부서번호)
AS
SELECT EMPNO, ENAME, DEPTNO
FROM EMP_COPY;

SELECT * FROM EMP_VIEW
WHERE 부서번호=30;

-- 복합 뷰 만들기
CREATE OR REPLACE
VIEW EMP_VIEW_DEPT
AS
SELECT E.EMPNO, E.ENAME, E.SAL, E.DEPTNO, D.DNAME, D.LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO
ORDER BY EMPNO DESC;

SELECT * FROM EMP_VIEW_DEPT;
```

### 14.2.3. 뷰 삭제 알아보기

```
-- 뷰 삭제하기
SELECT VIEW_NAME, TEXT
FROM USER_VIEWS;

SELECT * FROM EMP_VIEW;
DROP VIEW EMP_VIEW;
SELECT * FROM EMP_VIEW;
```

## 14.3. 뷰 생성에 사용되는 다양한 옵션

### 14.3.1. 뷰 수정을 위한 OR REPLACE 옵션

- CREATE OR REPLACE VIEW 를 사용하면 존재하지 않은 뷰이면 새로운 뷰를 생성하고 기존에 존재하는 뷰이면 그 내용을 변경한다.

```
CREATE OR REPLACE VIEW EMP_VIEW30
AS
SELECT EMPNO, ENAME, SAL, COMM, DEPTNO
FROM EMP_COPY
WHERE DEPTNO=30;
```

### 14.3.2. 기본 테이블 없이 뷰를 생성하기 위한 FORCE 옵션

- 뷰를 생성하는 경우에 일반적으로 기본 테이블이 존재한다는 가정 하에서 쿼리문을 작성한다.
- 극히 드물기는 하지만, 기본 테이블이 존재하지 않는 경우에도 뷰를 생성해야 할 경우가 있습니다. 이럴 경우에 사용하는 것이 FORCE 옵션이다.
- FORCE 옵션과 반대로 동작하는 것으로서 NOFORCE 옵션이 있다.
- NOFORCE 옵션은 반드시 기본 테이블이 존재해야 할 경우에만 뷰가 생성된다.
- 특별한 설정이 없으면 디폴트로 NOFORCE 옵션이 지정된 것이므로 간주한다.

```
-- FORCE 옵션으로 기본 테이블 없이 뷰 생성하기
DESC EMPLOYEES

CREATE OR REPLACE VIEW EMPLOYEES_VIEW
AS
SELECT EMPNO, ENAME, DEPTNO
FROM EMPLOYEES
WHERE DEPTNO=30; -- SQL Error: ORA-00942: table or view does not exist

CREATE OR REPLACE FORCE VIEW NOTABLE_VIEW
AS
SELECT EMPNO, ENAME, DEPTNO
FROM EMPLOYEES
WHERE DEPTNO=30;

SELECT VIEW_NAME, TEXT
FROM USER_VIEWS;

-- NOFORCE는 뷰 생성의 디폴트값으로 FORCE의 반대 기능을 가진 옵션이다.
CREATE OR REPLACE NOFORCE VIEW EXISTTABLE_VIEW
AS
SELECT EMPNO, ENAME, DEPTNO
FROM EMPLOYEES
WHERE DEPTNO=30; -- SQL Error: ORA-00942: table or view does not exist

SELECT VIEW_NAME, TEXT
FROM USER_VIEWS;

CREATE OR REPLACE VIEW EXISTTABLE_VIEW
AS
SELECT EMPNO, ENAME, DEPTNO
FROM EMPLOYEES
WHERE DEPTNO=30;
```

### 14.3.3. 조건 컬럼값을 변경하지 못하게 하는 WITH CHECK OPTION

- 뷰를 생성할 때 조건 제시에 사용된 컬럼 값을 변경 못하도록 하는 기능을 제공한다.

```
CREATE OR REPLACE VIEW VIEW_CHK30
AS
SELECT EMPNO, ENAME, SAL, COMM, DEPTNO
FROM EMP_COPY
WHERE DEPTNO=30 WITH CHECK OPTION;

UPDATE VIEW_CHK30 SET DEPTNO=20
WHERE SAL>=1200;
SELECT * FROM VIEW_CHK30;
```

#### 14.3.4. 기본 테이블 변경을 막는 WITH READ ONLY 옵션 WITH CHECK OPTION 비교

- WITH READ ONLY 옵션은 뷰를 통해서는 기본 테이블의 어떤 컬럼에 대해서도 내용을 절대 변경할 수 없도록 하는 것이다.

```
UPDATE VIEW_CHK30 SET COMM=1000;
SELECT * FROM VIEW_CHK30;

CREATE OR REPLACE VIEW VIEW_READ30
AS
SELECT EMPNO, ENAME, SAL, COMM, DEPTNO
FROM EMP_COPY
WHERE DEPTNO=30 WITH READ ONLY;

UPDATE VIEW_READ30 SET COMM=1000;
SELECT * FROM VIEW_READ30;
```

### 14.4. 뷰 활용하여 Top-N 구하기

- TOP-N을 구하기 위해서는 ROWNUM과 인라인 뷰가 사용된다.

#### 14.4.1. ROWNUM 컬럼 성격 파악하기

```
SELECT ROWNUM, EMPNO, ENAME, HIREDATE
FROM EMP;

SELECT EMPNO, ENAME, HIREDATE
FROM EMP
ORDER BY HIREDATE;

SELECT ROWNUM, EMPNO, ENAME, HIREDATE
FROM EMP
ORDER BY HIREDATE;
```

#### 14.4.2. 인라인 뷰로 구하는 TOP-N의 개념

- 인라인 뷰란 메인 쿼리의 SELECT 문의 FROM 절 내부에 사용된 서브 쿼리문을 말한다.
- 우리가 지금까지 생성한 뷰는 CREATE 명령어로 뷰를 생성했지만, 인라인 뷰는 SQL 문 내부에 뷰를 정의하고 이를 테이블처럼 사용한다.

```
-- 급여(SAL)를 많이 받는 6~10째 사원을 출력하기 위해서는 인라인 뷰안에 또 다른 인라인 뷰를 사용해야 한다.
-- 또한 ROWNUM 컬럼에 별칭을 부여해야 검색이 가능하다.
SELECT ROWNUM, RNUM, ENAME, SAL FROM
(SELECT ROWNUM RNUM, ENAME, SAL FROM
(SELECT * FROM EMP ORDER BY SAL DESC))
WHERE RNUM BETWEEN 6 AND 10;
```

### [꿀팁] MySQL에서 ROW COUNT 방법

```
SELECT @no:=@no+1 AS no, article_no, writer_id, title, regdate  
FROM article, (SELECT @no:=0) article  
ORDER BY article_no DESC;
```

### [과제] 과제-14-01.TXT

SQL> CONN SCOTT/TIGER 로 접속하여 SQL문을 작성하세요.

1. 사원 테이블(EMP)에서 가장 최근에 입사한 사원들중에 3~5번째의 사번과 사원명을 출력하는 SQL문을 작성하세요?

<정답>