

# 데이터베이스

- 순서 : Ch8-1(뷰)
- 학기 : 2018학년도 2학기
- 학과 : 가천대학교 컴퓨터공학과 2학년
- 교수 : 박양재

# 데이터베이스

- 목차

8.1 뷰의 기본 다루기

8.2 뷰의 고급 다루기

8.3 뷰의 생성에 사용되는 다양한 옵션

8.4 뷰 활용하여 TOP-N 구하기

## 8장. 뷰

### □ 뷰(VIEW)

#### 01.뷰의 기본 테이블

뷰는 ‘보다’라는 의미를 갖고 있는 점으로 알 수 있듯이 실제 테이블에 저장된 데이터를 뷰를 통해서 볼 수 있다. 사용자에게 주어진 뷰를 통해 기본 테이블을 제한적으로 사용하게 된다.

- 뷰는 **물리적 테이블에 근거한 논리적 가상 테이블**이다. **가상**은 실질적으로 데이터를 저장하고 있지 않기 때문에 붙인 것이고, **테이블**은 실질적으로 데이터를 저장하고 있지 않더라도 사용자는 마치 테이블을 사용하는 것과 동일하게 뷰를 사용할 수 있기 때문이다.

## 8장. 뷰

### □ 뷰(VIEW)

#### 01.뷰의 기본 테이블

실습 : 뷰의 기본 테이블 생성하기

```
SQL> CREATE TABLE DEPT_COPY  
2 AS  
3 SELECT * FROM DEPT ;
```

테이블이 생성되었습니다.

```
SQL> CREATE TABLE EMP_COPY  
2 AS  
3 SELECT * FROM EMP ;
```

테이블이 생성되었습니다.

```
SQL>  
SQL> SELECT * FROM DEPT_COPY ;
```

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

## 8장. 뷰

### □ 뷰(VIEW)

#### 01.뷰의 기본 테이블

실습 : 뷰의 기본 테이블 생성하기

```
SQL> SELECT * FROM EMP_COPY ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

14 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰(VIEW)

#### 02.뷰 정의하기

뷰는 CREATE VIEW로 생성하되 AS 다음에는 자주 사용되는 SELECT 문을 서브 쿼리문 형태로 기술한다.

[형식] CREATE [OR REPLACE] [{FORCE|NOFORCE}]VIEW view\_name

                            ①                            ②  
          [alias, alias, alias, ...)]  
          AS subquery  
          [WITH CHECK OPTION] .....③  
          [WITH READ ONLY] ; .....④

① CREATE OR REPLACE VIEW : CREATE VIEW를 통해서 만들어진 뷰의 구조를 변경하려면 삭제하고 다시 만들어야 한다. 그러나 CREATE OR REPLACE VIEW로 만들면 새로운 뷰도 만들고, 구조변경(REPLACE)도 가능하다.

② FORCE|NOFORCE : FORCE를 사용하면 기본 테이블의 존재 여부에 관계없이 뷰를 생성

③ WITH CHECK OPTION : 해당 뷰를 통해서 볼 수 있는 범위 내에서만 UPDATE 또는 INSERT가 가능

④ WITH READ ONLY : 해당 뷰를 통해서만 SELECT만 가능하며, INSERT/UPDATE/DELETE할 수 없다. 생략시에는 뷰를 사용하여 추가, 수정, 삭제(INSERT, UPDATE, DELETE)가 모두 가능하다.

## 8장. 뷰

### □ 뷰(VIEW)

#### 02.뷰 정의하기

뷰의 필요성 예제 : 만일, 30번 부서에 소속된 직원들의 사번과 이름, 부서번호를 자주 검색한다면 다음과 같이 SELECT문을 여러 번 입력해야 한다.

```
SQL> SELECT EMPNO, ENAME, DEPTNO  
2 FROM EMP_COPY  
3 WHERE DEPTNO = 30 ;
```

EMPNO	ENAME	DEPTNO
7499	ALLEN	30
7521	WARD	30
7654	MARTIN	30
7698	BLAKE	30
7844	TURNER	30
7900	JAMES	30

6 개의 행이 선택되었습니다.

위와 같이 결과를 출력하기 위해서 매번 SELECT문을 입력하는 것은 번거로운 일이다. 매번 SELECT문을 입력하는 대신에 간단하게 뷰를 정의해서 사용하면 원하는 결과를 얻을 수 있다.

## 8장. 뷰

### □ 뷰(VIEW)

#### 02.뷰 정의하기

뷰의 필요성 예제 : 만일, 30번 부서에 소속된 직원들의 사번과 이름, 부서번호를 자주 검색한다면 다음과 같이 SELECT문을 여러 번 입력해야 한다.

```
SQL> CREATE VIEW EMP_VIEW30
  2 AS
  3 SELECT EMPNO, ENAME, SAL, DEPTNO
  4 FROM EMP_COPY
  5 WHERE DEPTNO = 30 ;
CREATE VIEW EMP_VIEW30
      *
```

1행에 오류:

ORA-01031: 권한이 불충분합니다

```
SQL> CONN system/DBAdmin2015
```

연결되었습니다.

```
SQL>
```

```
SQL>
```

```
SQL> GRANT CREATE VIEW TO scott ;
```

권한이 부여되었습니다.

```
SQL> CONN scott/tiger
```

연결되었습니다.

```
SQL> CREATE VIEW EMP_VIEW30
  2 AS
  3 SELECT EMPNO, ENAME, SAL, DEPTNO
  4 FROM EMP_COPY
  5 WHERE DEPTNO = 30 ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM EMP_VIEW30 ;
```

EMPNO	ENAME	SAL	DEPTNO
7499	ALLEN	1600	30
7521	WARD	1250	30
7654	MARTIN	1250	30
7698	BLAKE	2850	30
7844	TURNER	1500	30
7900	JAMES	950	30

6개의 행이 선택되었습니다.



## 8장. 뷰

### □ 뷰(VIEW)

#### 02.뷰 정의하기

뷰의 필요성 예제 : 만일, 30번 부서에 소속된 직원들의 사번과 이름, 부서번호를 자주 검색한다면 다음과 같이 SELECT문을 여러 번 입력해야 한다.

뷰의 구조는 기본 테이블의 구조를 그대로 상속 받았음.

```
SQL> DESC EMP_VIEW30 ;
```

이름	널?	유형
EMPNO		NUMBER(4)
ENAME		VARCHAR2(10)
SAL		NUMBER(7,2)
DEPTNO		NUMBER(2)

## 8장. 뷰

### □ 뷰(VIEW)

#### 02.뷰 정의하기

실습 : 20번 부서에 소속된 직원들의 사번과 이름, 부서번호와 관리자의 사번을 출력하기 위한 뷰를 정의 하세요.

```
SQL> CREATE VIEW EMP_VIEW20
  2 AS
  3 SELECT EMPNO, ENAME, DEPTNO, MGR
  4 FROM EMP_COPY
  5 WHERE DEPTNO = 20 ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM EMP_VIEW20 ;
```

EMPNO	ENAME	DEPTNO	MGR
7369	SMITH	20	7902
7566	JONES	20	7839
7788	SCOTT	20	7566
7876	ADAMS	20	7788
7902	FORD	20	7566

## 8장. 뷰

### □ 뷰의 고급 다루기

뷰의 동작원리를 살펴보기 위해서 뷰의 저장된 내부 구조를 살펴보고 다양한 형태(단순 뷰, 복합 뷰)에 대한 개념을 이해하고 작성하는 방법을 살펴보자.

뷰는 물리적으로 데이터를 저장하지 않는데 다음과 같은 질의문을 수행할 수 있는 이유는 무엇인가?

```
SQL> SELECT * FROM EMP_VIEW30 ;
```

CREATE VIEW 명령어로 뷰를 정의 할 때 AS절 다음에 기술한 쿼리문장 자체를 저장한다. 뷰를 정의할 때 기술한 쿼리문은 데이터 디렉터리 USER\_VIEWS 테이블에 TEXT 컬럼 값을 보면 된다.

```
SQL> DESC USER_VIEWS
```

이름	널?	유형
VIEW_NAME	NOT NULL	VARCHAR2(30)
TEXT_LENGTH		NUMBER
TEXT		LONG
TYPE_TEXT_LENGTH		NUMBER
TYPE_TEXT		VARCHAR2(4000)
OID_TEXT_LENGTH		NUMBER

## 8장. 뷰

### □ 뷰의 고급 다루기

뷰는 물리적으로 데이터를 저장하지 않는데 다음과 같은 질의문을 수행할 수 있는 이유는 무엇인가?

```
SQL> COLUMN VIEW_NAME FORMAT A15
SQL> COLUMN TEXT FORMAT A50
SQL> SELECT VIEW_NAME, TEXT
  2  FROM USER_VIEWS ;
```

VIEW_NAME	TEXT
EMP_VIEW20	SELECT EMPNO, ENAME, DEPTNO, MGR FROM EMP_COPY WHERE DEPTNO = 20
EMP_VIEW30	SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP_COPY WHERE DEPTNO = 30

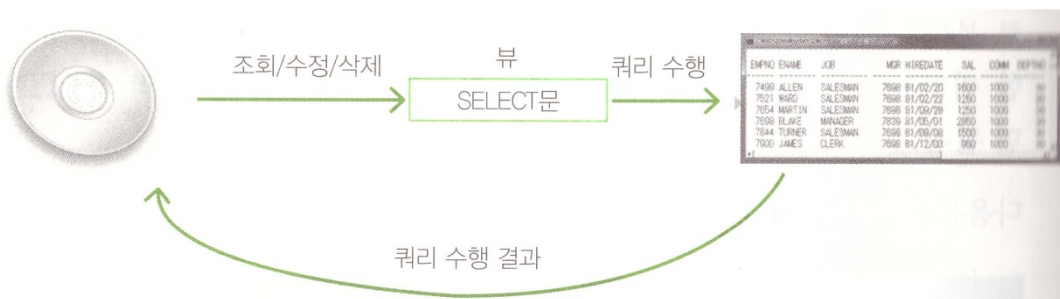
USER\_VIEWS를 보면 뷰는 SELECT문이 이름을 갖도록 한다. 기본 테이블은 디스크 공간을 할당 받아서 실질적으로 데이터를 저장하고 있지만, 뷰는 데이터 덱서너리 USER\_VIEWS 테이블에 사용자가 뷰를 정의 할 때 기술한 서브쿼리문(SELECT문)만을 문자열 형태로 저장하고 있다.

## 8장. 뷰

### □ 뷰의 고급 다루기

뷰는 물리적으로 데이터를 저장하지 않는데 다음과 같은 질의문을 수행할 수 있는 이유는 무엇인가?

\*.뷰에 대한 질의가 내부적으로 어떻게 처리되는가?



- ① 사용자가 뷰에 대해서 질의하면 USER\_VIEWS에서 뷰에 대한 정의를 조회한다.
- ② 기본 테이블에 대한 뷰의 접근 권한을 조회한다.
- ③ 뷰에 대한 질의를 기본 테이블에 대한 질의로 변환한다.
- ④ 기본 테이블에 대한 질의를 통해 데이터를 검색한다.
- ⑤ 검색된 결과를 출력한다.

## 8장. 뷰

### □ 뷰의 고급 다루기

뷰는 물리적으로 데이터를 저장하지 않는데 다음과 같은 질의문을 수행할 수 있는 이유는 무엇인가?

SELECT문의 FROM절 다음에 생성한 뷰인 EMP\_VIEW30을 기술하여 질의하면 오라클 서버는 USER\_VIEWS30을 찾는다. 그리고 뷰를 정의할 때 기술했던 서브 쿼리문이 저장된 TEXT값을 EMP\_VIEW30 위치로 가져간다.

질의는 기본 테이블인 EMP\_COPY테이블을 통해 일어난다. 즉, 기본 테이블인 EMP\_COPY에 대해서 서브 쿼리문을 수행한다. 이러한 동작으로 실질적으로 뷰에는 데이터가 없어도 데이터 검색을 할 수 있다.

뷰를 통해서 SELECT문으로 검색은 물론이고, DML(INSERT, DELETE, UPDATE)명령어를 수행할 수 있다. **뷰는 실질적으로 데이터를 저장하지 않고 쿼리문만 저장하는 형태이기 때문에 뷰에 DML로 내용을 조작하면 기본 테이블의 내용이 변경된다.**

## 8장. 뷰

### □ 뷰의 고급 다루기

실습 : 뷰에 INSERT문으로 데이터 추가하기

```
SQL> SELECT * FROM EMP_VIEW30;
```

EMPNO	ENAME	SAL	DEPTNO
7499	ALLEN	1600	30
7521	WARD	1250	30
7654	MARTIN	1250	30
7698	BLAKE	2850	30
7844	TURNER	1500	30
7900	JAMES	950	30

6 개의 행이 선택되었습니다.

```
SQL> INSERT INTO EMP_VIEW30  
2 VALUES(8888, 'GACHON', 1000, 30) ;
```

1 개의 행이 만들어졌습니다.

```
SQL> SELECT * FROM EMP_VIEW30 ;
```

EMPNO	ENAME	SAL	DEPTNO
7499	ALLEN	1600	30
7521	WARD	1250	30
7654	MARTIN	1250	30
7698	BLAKE	2850	30
7844	TURNER	1500	30
7900	JAMES	950	30
8888	GACHON	1000	30

7 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰의 고급 다루기

실습 : 뷰에 INSERT문으로 데이터 추가하기

뷰를 만든 기본 테이블인 EMP\_COPY 테이블에도 뷰에서 입력한 내용이 추가 되었다.

```
SQL> SELECT * FROM EMP_COPY ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10
8888	GACHON				1000		30

15 개의 행이 선택되었습니다.



## 8장. 뷰

### □ 뷰의 고급 다루기

실습 : 뷰에 INSERT문으로 데이터 추가하기

뷰를 만든 원본 테이블인 EMP 테이블에는 뷰에서 입력한 내용이 추가 되지 않는다.

```
SQL> SELECT * FROM EMP ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

14 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰의 고급 다루기

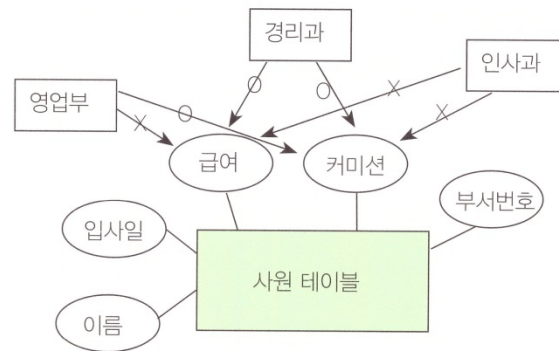
01. 뷰를 사용하는 이유 : 복잡하고 긴 쿼리문을 뷰로 정의하면 접근을 단순화 할 수 있고, 보안에 유리하기 때문이다.

뷰는 **사용자에게 특정 객체만 조회 할 수 있도록 권한을 부여 할 수 있다.** 그렇기 때문에 동일한 테이블에 접근하는 사용자들마다 서로 다른 뷰에 접근하도록 여러 개의 뷰를 정의해 놓고 **특정 사용자만 해당 뷰에 접근하게 할 수 있어 보안 목적으로 데이터를 보호**할 수 있다.

## 8장. 뷰

### □ 뷰의 고급 다루기

01. 뷰를 사용하는 이유 : 보안 목적으로 데이터를 보호(사원 테이블의 다양한 접근)



```
SQL> SELECT * FROM EMP_COPY ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30

개인정보인 급여와 커미션은 부서에 따라서 접근 제한해야하고, 인사과는 급여나 커미션 모두 조회하지 못하게하고, 경리과는 모두 조회 할 수 있도록하지만 영업부에서는 경쟁심을 유발하기 위해 다른 사원의 커미션을 조회 할 수 있게 한다.

## 8장. 뷰

### □ 뷰의 고급 다루기

01. 뷰를 사용하는 이유 : 보안 목적으로 데이터를 보호(사원 테이블의 다양한 접근)

이와 같이 뷰는 데이터베이스 실무에서 전체 데이터의 일부분만 접근 할 수 있도록 하여 중요한 데이터 유출을 방지하여 데이터를 보호하기 위한 보안 목적으로 사용할 수 있다.

급여와 커미션과 같은 개인적인 정보에는 인사과 직원이 접근하지 못하도록 제한하여 중요한 정보를 보호한다.

```
SQL> CREATE VIEW EMP_INSA
2 AS
3 SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, DEPTNO
4 FROM EMP_COPY ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM EMP_INSA ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	20
7499	ALLEN	SALESMAN	7698	81/02/20	30
7521	WARD	SALESMAN	7698	81/02/22	30
7566	JONES	MANAGER	7839	81/04/02	20
7654	MARTIN	SALESMAN	7698	81/09/28	30
7698	BLAKE	MANAGER	7839	81/05/01	30
7782	CLARK	MANAGER	7839	81/06/09	10
7788	SCOTT	ANALYST	7566	87/04/19	20
7839	KING	PRESIDENT		81/11/17	10
7844	TURNER	SALESMAN	7698	81/09/08	30
7876	ADAMS	CLERK	7788	87/05/23	20
7900	JAMES	CLERK	7698	81/12/03	30
7902	FORD	ANALYST	7566	81/12/03	20
7934	MILLER	CLERK	7782	82/01/23	10
8888	GACHON				30

15 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 02. 단순 뷰와 복합 뷰

뷰를 정의하기 위해 사용되는 기본 테이블의 수에 따라 단순 뷰(SIMPLE VIEW)와 복합 뷰(COMPLEX VIEW)로 분류된다.

단순 뷰	복합 뷰
하나의 테이블로 생성	여러 개의 테이블로 생성
그룹함수의 사용이 불가능	그룹함수의 사용이 가능
DISTINCT 사용이 불가능	DISTINCT 사용이 가능
DML 사용이 가능	DML 사용이 불가능

단순 뷰의 특징은 DML, 즉 INSERT, UPDATE, DELETE문을 사용할 수 있으며, 기본 테이블의 내용도 변경된다. 뷰를 마치 테이블 처럼 사용하지만 데이터를 저장하지 않고 기본 테이블을 근거로 동작하기 때문에 뷰를 가상(가짜)테이블이라고 한다.

# 8장. 뷰

## □ 뷰의 고급 다루기

### 02. 단순 뷰와 복합 뷰

실습 : 단순 뷰의 컬럼에 별칭 부여하기-기본 테이블(EMP\_COPY)의 컬럼명을 상속 받지 않고 한글화하여 컬럼명을 사원번호, 급여, 부서번호로 구성하는 뷰를 작성해 보세요.

```
SQL> CREATE OR REPLACE
  2 VIEW EMP_INSA<사원번호, 사원명, 급여, 부서번호>
  3 AS
  4 SELECT EMPNO, ENAME, SAL, DEPTNO
  5 FROM EMP_COPY ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM EMP_INSA
  2 WHERE 부서번호=30 ;
```

뷰 생성시 사용한  
한글 별칭 사용

사원번호	사원명	급여	부서번호
7499	ALLEN	1600	30
7521	WARD	1250	30
7654	MARTIN	1250	30
7698	BLAKE	2850	30
7844	TURNER	1500	30
7900	JAMES	950	30
8888	GACHON	1000	30

7 개의 행이 선택되었습니다.

```
SQL> SELECT * FROM EMP_INSA ;
```

사원번호	사원명	급여	부서번호
7369	SMITH	800	20
7499	ALLEN	1600	30
7521	WARD	1250	30
7566	JONES	2975	20
7654	MARTIN	1250	30
7698	BLAKE	2850	30
7782	CLARK	2450	10
7788	SCOTT	3000	20
7839	KING	5000	10
7844	TURNER	1500	30
7876	ADAMS	1100	20
7900	JAMES	950	30
7902	FORD	3000	20
7934	MILLER	1300	10
8888	GACHON	1000	30

15 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 02. 단순 뷰와 복합 뷰

실습 : 단순 뷰의 컬럼에 별칭 부여하기-기본 테이블(EMP\_COPY)의 컬럼명을 상속 받지 않고 한글화하여 컬럼명을 사원번호, 급여, 부서번호로 구성하는 뷰를 작성해 보세요.

```
SQL> SELECT * FROM EMP_INSA  
2  WHERE DEPTNO=30 ;  
WHERE DEPTNO=30
```

뷰 생성시 사용하지  
않은 컬럼명 사용

\*

2행에 오류:

ORA-00904: "DEPTNO": 부적합한 식별자

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 03. 그룹 함수를 사용한 단순 뷰

실습 : 그룹 함수 SUM과 AVG를 사용해서 각각 부서별 급여의 총액과 평균을 구하는 뷰를 작성하세요.

```
SQL> CREATE VIEW VIEW_SAL
 2 AS
 3 SELECT DEPTNO, SUM(SAL) AS "SalSum", AVG(SAL) AS "SalAvg"
 4 FROM EMP_COPY
 5 GROUP BY DEPTNO ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM VIEW_SAL ;
```

DEPTNO	SalSum	SalAvg
30	10400	1485.71429
20	10875	2175
10	8750	2916.66667



## 8장. 뷰

### □ 뷰의 고급 다루기

#### 03.그룹 함수를 사용한 단순 뷰

실습 : 그룹 함수 SUM과 AVG를 사용해서 각각 부서별 급여의 총액과 평균을 구하는 뷰를 작성하세요.(VIEW\_SAL1을 생성 할 때 급여의 총액과 평균을 구하는 컬럼에 반드시 별칭을 주어야 오류가 발생하지 않는다)뷰를 생성할 때 가상 컬럼을 사용하려면 사용자는 반드시 이름을 따로 설정해야 한다.,

```
SQL> CREATE VIEW VIEW_SAL1
  2 AS
  3 SELECT DEPTNO, SUM(SAL), AVG(SAL)
  4 FROM EMP_COPY
  5 GROUP BY DEPTNO ;
SELECT DEPTNO, SUM(SAL), AVG(SAL)
      *
```

3행에 오류:

ORA-00998: 이 식은 열의 별명과 함께 지정해야 합니다

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 03. 그룹 함수를 사용한 단순 뷰

단순 뷰는 DML명령어를 사용하여 조작이 가능하다. 그러나 몇가지 경우에는 조작이 불가능하다.

① 뷰 정의에 포함되어 있지 않은 기본 테이블 컬럼에 NOT NULL 제약조건이 지정되어 있는 경우는 INSERT문 사용이 불가능하다. 뷰에 INSERT문은 기본 테이블에 NULL값을 입력하는 형태가 되기 때문이다.

② 뷰에 SAL\*12 같이 산술 표현식으로 정의된 가상 컬럼이 정의되면 INSERT나 UPDATE가 불가능하다.

③ DISTINCT를 포함한 경우에도 DML명령어를 사용할 수 없다.

④ 그룹함수나 GROUP BY절을 포함한 경우에도 DML명령을 사용할 수 없다.

```
SQL> SELECT * FROM VIEW_SAL ;
```

DEPTNO	SalSum	SalAvg
30	10400	1485.71429
20	10875	2175
10	8750	2916.66667

```
SQL> INSERT INTO VIEW_SAL  
2 VALUES(40, 30000, 4000) ;  
INSERT INTO VIEW_SAL
```

\*

1행에 오류:

ORA-01733: 가상 열은 사용할 수 없습니다

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 04. 복합 뷰란 무엇인가?

두 개 이상의 기본 테이블에 의해 정의된 뷰를 말한다.

예제: **사원 테이블과 부서 테이블이 자주 조인한다고 가정하면** 매번 복잡한 SEELCT문을 **작성해야** 하는 문제점을 뷰를 작성함으로써 단순 하게 해결 하도록 뷰를 작성하세요.

```
SQL> SELECT E.EMPNO, E.ENAME, E.SAL, E.DEPTNO, D.DNAME, D.LOC  
2 FROM EMP E, DEPT D  
3 WHERE E.DEPTNO=D.DEPTNO  
4 ORDER BY EMPNO DESC ;
```

EMPNO	ENAME	SAL	DEPTNO	DNAME	LOC
7934	MILLER	1300	10	ACCOUNTING	NEW YORK
7902	FORD	3000	20	RESEARCH	DALLAS
7900	JAMES	950	30	SALES	CHICAGO
7876	ADAMS	1100	20	RESEARCH	DALLAS
7844	TURNER	1500	30	SALES	CHICAGO
7839	KING	5000	10	ACCOUNTING	NEW YORK
7788	SCOTT	3000	20	RESEARCH	DALLAS
7782	CLARK	2450	10	ACCOUNTING	NEW YORK
7698	BLAKE	2850	30	SALES	CHICAGO
7654	MARTIN	1250	30	SALES	CHICAGO
7566	JONES	2975	20	RESEARCH	DALLAS
7521	WARD	1250	30	SALES	CHICAGO
7499	ALLEN	1600	30	SALES	CHICAGO
7369	SMITH	800	20	RESEARCH	DALLAS

14 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 04.복합 뷰란 무엇인가?

두 개 이상의 기본 테이블에 의해 정의된 뷰를 말한다.

예제: 사원 테이블과 부서 테이블이 자주 보인다고 가정하면 매번 복잡한 SEELCT문을 작성해야 하는 문제점을 **뷰를 작성함으로서 단순 하게 해결 하도록** 뷰를 작성하세요.

```
SQL> CREATE VIEW EMP_VIEW_DEPT
  2 AS
  3 SELECT E.EMPNO, E.ENAME, E.SAL, E.DEPTNO, D.DNAME, D.LOC
  4 FROM EMP E, DEPT D
  5 WHERE E.DEPTNO=D.DEPTNO
  6 ORDER BY EMPNO DESC ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM EMP_VIEW_DEPT ;
```

EMPNO	ENAME	SAL	DEPTNO	DNAME	LOC
7934	MILLER	1300	10	ACCOUNTING	NEW YORK
7902	FORD	3000	20	RESEARCH	DALLAS
7900	JAMES	950	30	SALES	CHICAGO
7876	ADAMS	1100	20	RESEARCH	DALLAS
7844	TURNER	1500	30	SALES	CHICAGO
7839	KING	5000	10	ACCOUNTING	NEW YORK
7788	SCOTT	3000	20	RESEARCH	DALLAS

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 04. 복합 뷰란 무엇인가?

실습 : 각 부서별 최대 급여와 최소 급여를 출력하는 뷰를 SAL\_VIEW라는 이름을 작성하세요.

```
SQL> CREATE VIEW SAL_VIEW
  2 AS
  3 SELECT D.DNAME, MAX(E.SAL) AS MAX_SAL, MIN(E.SAL) AS MIN_SAL
  4 FROM EMP E, DEPT D
  5 WHERE E.DEPTNO=D.DEPTNO
  6 GROUP BY D.DNAME ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM SAL_VIEW ;
```

DNAME	MAX_SAL	MIN_SAL
ACCOUNTING	5000	1300
RESEARCH	3000	800
SALES	2850	950

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 05.뷰 삭제하기

[형식] DROP VIEW view\_name ;

뷰는 실체가 없는 가상 테이블이기 때문에 뷰를 삭제한다는 것은 USER\_VIEWS 데이터 디렉터리에 저장된 뷰의 정의를 삭제하는 것을 의미하며, 따라서 뷰를 삭제해도 뷰를 정의한 기본 테이블의 구조나 데이터에는 전혀 영향을 주지 않는다.

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 05. 뷰 삭제하기(지금까지 생성한 뷰를 조회하는 명령)

```
SQL> SELECT VIEW_NAME, TEXT  
2 FROM USER_VIEWS ;
```

VIEW_NAME	TEXT
EMP_INSA	SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP_COPY
EMP_VIEW20	SELECT EMPNO, ENAME, DEPTNO, MGR FROM EMP_COPY WHERE DEPTNO = 20
EMP_VIEW30	SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP_COPY WHERE DEPTNO = 30
EMP_VIEW_DEPT	SELECT E.EMPNO, E.ENAME, E.SAL, E.DEPTNO, D.DNAME, D.LOC FROM EMP E, DEPT D WHERE
SAL_VIEW	SELECT D.DNAME, MAX(E.SAL) AS MAX_SAL, MIN(E.SAL) AS MIN_SAL FROM EMP E, DEPT D
VIEW_SAL	SELECT DEPTNO, SUM(SAL) AS "SalSum", AVG(SAL) AS " SalAvg" FROM EMP_COPY GROUP BY

6 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰의 고급 다루기

#### 05.뷰 삭제하기(VIEW\_SAL 뷰 삭제하기)

```
SQL> DROP VIEW VIEW_SAL ;
```

뷰가 삭제되었습니다.

VIEW_NAME	TEXT
EMP_INSA	SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP_COPY
EMP_VIEW20	SELECT EMPNO, ENAME, DEPTNO, MGR FROM EMP_COPY WHERE DEPTNO = 20
EMP_VIEW30	SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP_COPY WHERE DEPTNO = 30
EMP_VIEW_DEPT	SELECT E.EMPNO, E.ENAME, E.SAL, E.DEPTNO, D.DNAME, D.LOC FROM EMP E, DEPT D WHERE
SAL_VIEW	SELECT D.DNAME, MAX(E.SAL) AS MAX_SAL, MIN(E.SAL) AS MIN_SAL FROM EMP E, DEPT D

```
SQL> SELECT * FROM VIEW_SAL ;  
SELECT * FROM VIEW_SAL
```

\*

1 행에 오류:

ORA-00942: 테이블 또는 뷰가 존재하지 않습니다



## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

#### 01.뷰 수정을 위한 OR REPLACE 옵션

[형식] CREATE [OR REPLACE] [{FORCE|NOFORCE}]VIEW view\_name

                                  ①                                  ②  
          [alias, alias, alias, ...]  
          AS subquery  
          [WITH CHECK OPTION] .....③  
          [WITH READ ONLY] ; .....④

① CREATE OR REPLACE VIEW : CREATE VIEW를 통해서 만들어진 뷰의 구조를 변경하려면 삭제하고 다시 만들어야 한다. 그러나 CREATE OR REPLACE VIEW로 만들면 새로운 뷰도 만들고, 구조변경(REPLACE)도 가능하다.

② FORCE|NOFORCE : FORCE를 사용하면 기본 테이블의 존재 여부에 관계없이 뷰를 생성

③ WITH CHECK OPTION : 해당 뷰를 통해서 볼 수 있는 범위 내에서만 UPDATE 또는 INSERT가 가능

④ WITH READ ONLY : 해당 뷰를 통해서만 SELECT만 가능하며, INSERT/UPDATE/DELETE할 수 없다. 생략시에는 뷰를 사용하여 추가, 수정, 삭제(INSERT, UPDATE, DELETE)가 모두 가능하다.

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

#### 01.뷰 수정을 위한 OR REPLACE 옵션

뷰를 생성할 때 OR REPLACE를 사용하면 존재하지 않는 뷰이면 새로운 뷰를 생성하고 존재하는 뷰라면 그 내용을 변경한다.

예제:기존에 작성된 EMP\_VIEW30 뷰에는 EMPNO, ENAME, SAL, DEPTNO 컬럼을 출력하는 형태였다.만일 EMP\_VIEW30뷰에 커미션 컬럼을 추가하려면 어떻게 해야 할까?

#### ① 새롭게 EMP\_VIEW30 뷰를 만들면

```
SQL> CREATE VIEW EMP_VIEW30
  2  AS
  3  SELECT EMPNO, ENAME, SAL, DEPTNO
  4  FROM EMP_COPY
  5  WHERE DEPTNO=30 ;
CREATE VIEW EMP_VIEW30
```

\*

1행에 오류:

ORA-00955: 기존의 객체가 이름을 사용하고 있습니다.

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

#### 01.뷰 수정을 위한 OR REPLACE 옵션

뷰를 생성할 때 OR REPLACE를 사용하면 존재하지 않는 뷰이면 새로운 뷰를 생성하고 존재하는 뷰라면 그 내용을 변경한다.

예제:기존에 작성된 EMP\_VIEW30 뷰에는 EMPNO, ENAME, SAL, DEPTNO 컬럼을 출력하는 형태였다.만일 EMP\_VIEW30뷰에 커미션 컬럼을 추가하려면 어떻게 해야 할까?

#### ② CREATE OR REPLACE VIEW 옵션을 사용하여 EMP\_VIEW30 뷰에 커미션 추가하기

```
SQL> CREATE OR REPLACE VIEW EMP_VIEW30  
2 AS  
3 SELECT EMPNO, ENAME, SAL, COMM, DEPTNO  
4 FROM EMP_COPY  
5 WHERE DEPTNO=30 ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM EMP_VIEW30 ;
```

EMPNO	ENAME	SAL	COMM	DEPTNO
7499	ALLEN	1600	300	30
7521	WARD	1250	500	30
7654	MARTIN	1250	1400	30
7698	BLAKE	2850		30
7844	TURNER	1500	0	30
7900	JAMES	950		30
8888	GACHON	1000		30

7 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

02.기본 테이블 없이 뷰를 생성하기 위한 FORCE 옵션(NOFORCE 옵션은 반드시 기본 테이블이 존재해야 뷰가 생성)디폴트는 NOFORCE이다.

예제: EMLPOYEEES\_VIEW 만들기

#### ①기본 테이블 EMPLOYEES 존재여부 확인

```
SQL> DESC EMPLOYEES ;  
ERROR:  
ORA-04043: EMPLOYEES 객체는 존재하지 않습니다.
```

#### ② 존재하지 않는 EMPLOYEES를 기본 테이블로 생성하면 오류 발생

```
SQL> CREATE OR REPLACE VIEW EMPLOYEES_VIEW  
2 AS  
3 SELECT EMPNO, ENAME, DEPTNO  
4 FROM EMPLOYEES  
5 WHERE DEPTNO=30;  
FROM EMPLOYEES  
*  
4행에 오류:  
ORA-00942: 테이블 또는 뷰가 존재하지 않습니다
```

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

02.기본 테이블 없이 뷰를 생성하기 위한 FORCE 옵션(NOFORCE 옵션은 반드시 기본 테이블이 존재해야 뷰가 생성)디폴트는 NOFORCE이다.

예제: EMLPOYEES\_VIEW 만들기

#### ③ FROCE 옵션으로 강제로 뷰 만들기

```
SQL> CREATE OR REPLACE FORCE VIEW NOTABLE_VIEWE
2 AS
3 SELECT EMPNO, ENAME, DEPTNO
4 FROM EMPLOYEES
5 WHERE DEPTNO=30;
```

경고: 컴파일 오류와 함께 뷰가 생성되었습니다.

#### ④ 뷰 생성 확인하기

```
SQL> SELECT VIEW_NAME, TEXT
2 FROM USER_VIEWS ;
```

VIEW_NAME	TEXT
EMP_INSA	SELECT EMPNO, ENAME, SAL, DEPTNO FROM EMP_COPY
EMP_VIEW20	SELECT EMPNO, ENAME, DEPTNO, MGR FROM EMP_COPY WHERE DEPTNO = 20
EMP_VIEW30	SELECT EMPNO, ENAME, SAL, COMM, DEPTNO FROM EMP_COPY WHERE DEPTNO=30
EMP_VIEW_DEPT	SELECT E.EMPNO, E.ENAME, E.SAL, E.DEPTNO, D.DNAME, D.LOC FROM EMP E, DEPT D WHER
NOTABLE_VIEWE	SELECT EMPNO, ENAME, DEPTNO FROM EMPLOYEES WHERE DEPTNO=30
SAL_VIEW	SELECT D.DNAME, MAX(E.SAL) AS MAX_SAL, MIN(E.SAL) AS MIN_SAL FROM EMP E, DEPT D

6 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

03.조건 컬럼값을 변경하지 못하게 하는 WITH CHECK OPTION

WITH CHECK OPTION을 사용하면 뷰를 생성할 때, 조건 제시에 사용된 컬럼 값을 변경하지 못하게하는 기능

예제: 뷰를 정의하는 서브 쿼리문에 WHERE절을 추가하여 기본 테이블 중 특정 조건을 만족하는 행들로 구성된 뷰를 생성할 수 있다.

① 30번 부서 소속의 사원정보만으로 구성된 뷰=EMP\_VIEW30

```
SQL> SELECT * FROM EMP_VIEW30 ;
```

EMPNO	ENAME	SAL	COMM	DEPTNO
7499	ALLEN	1600	300	30
7521	WARD	1250	500	30
7654	MARTIN	1250	1400	30
7698	BLAKE	2850		30
7844	TURNER	1500	0	30
7900	JAMES	950		30
8888	GACHON	1000		30

7 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

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

예제: 뷰를 정의하는 서브 쿼리문에 WHERE절을 추가하여 기본 테이블 중 특정 조건을 만족하는 행들로 구성된 뷰를 생성할 수 있다.(뷰는 마치 테이블처럼 SELECT문으로 조회할 수 있는 것은 물론 DML문으로 내용을 조작 할 수 있다)

② 30번 부서에 소속된 사원 중 급여가 1200 이상인 사원을 20번 부서로 이동 시켜보자

```
SQL> SELECT * FROM EMP_VIEW30 ;
```

EMPNO	ENAME	SAL	COMM	DEPTNO
7499	ALLEN	1600	300	30
7521	WARD	1250	500	30
7654	MARTIN	1250	1400	30
7698	BLAKE	2850		30
7844	TURNER	1500	0	30
7900	JAMES	950		30
8888	GACHON	1000		30

7 개의 행이 선택되었습니다.

```
SQL> UPDATE EMP_VIEW30
```

```
2 SET DEPTNO=20
```

```
3 WHERE SAL >=1200 ;
```

```
5 행이 갱신되었습니다.
```

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

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

② 30번 부서에 소속된 사원 중 급여가 1200 이상인 사원을 20번 부서로 이동 시켜보자  
 행의 개수가 이동 전보다 대폭 줄어들었다 - 뷰를 대상으로 UPDATE문을 수행하면 뷰를 생성할 때  
 사용했던 기본 테이블(EMP\_COPY)의 내용이 변경된다. 그런데 뷰를 SELECT문으로 조회하면 뷰를 정의  
 할 때 주어진 쿼리문인 부서번호 30인 사원만 조회하기 때문이다.

SQL> SELECT \* FROM EMP\_COPY ;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	20
7521	WARD	SALESMAN	7698	81/02/22	1250	500	20
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	20
7698	BLAKE	MANAGER	7839	81/05/01	2850		20
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	20
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10
8888	GACHON				1000		30

SQL> SELECT \* FROM EMP\_VIEW30 ;

EMPNO	ENAME	SAL	COMM	DEPTNO
7900	JAMES	950		30
8888	GACHON	1000		30

15 개의 행이 선택되었습니다.



## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

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

③ 이와 같은 혼란을 방지하기 위하여 뷰를 생성 할 때 WHERE절에 WITH CHECK OPTION을 기술하여 조건 제시를 위해 사용한 컬럼값이 뷰를 통해서 변경되지 못하게 한다.

```
SQL> SELECT * FROM EMP_COPY ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	20
7521	WARD	SALESMAN	7698	81/02/22	1250	500	20
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	20
7698	BLAKE	MANAGER	7839	81/05/01	2850		20
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	20
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10
8888	GACHON				1000		30

15 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

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

③ 이와 같은 혼란을 방지하기 위하여 뷰를 생성 할 때 WHERE절에 WITH CHECK OPTION을 기술하여 조건 제시를 위해 사용한 컬럼값이 뷰를 통해서 변경되지 못하게 한다.

```
SQL> CREATE OR REPLACE VIEW VIEW_CHK30
 2 AS
 3 SELECT EMPNO, ENAME, SAL, COMM, DEPTNO
 4 FROM EMP_COPY
 5 WHERE DEPTNO=30 WITH CHECK OPTION ;
```

뷰가 생성되었습니다.

```
SQL> UPDATE VIEW_CHK30
 2 SET DEPTNO=20
 3 WHERE SAL >=950 ;
UPDATE VIEW_CHK30
```

\*

1행에 오류:

ORA-01402: 뷰의 WITH CHECK OPTION의 조건에 위배 됩니다

```
SQL> SELECT * FROM VIEW_CHK30 ;
```

EMPNO	ENAME	SAL	COMM	DEPTNO
7900	JAMES	950		30
8888	GACHON	1000		30

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

04.기본 테이블 변경을 방지하는 WITH READ ONLY 옵션과 WITH CHECK OPTION

WITH READ ONLY 옵션은 기본 테이블의 어떤 컬럼에 대해서도 뷰를 통한 내용 수정을 불가능하게 만드는 기능

① VIEW\_CHK30 뷰의 COMM 컬럼을 1000으로 변경하기

```
SQL> SELECT * FROM VIEW_CHK30 ;
```

EMPNO	ENAME	SAL	COMM	DEPTNO
7900	JAMES	950		30
8888	GACHON	1000		30

```
SQL> UPDATE VIEW_CHK30  
2 SET COMM=1000 ;
```

2 행이 갱신되었습니다.

```
SQL> SELECT * FROM VIEW_CHK30 ;
```

EMPNO	ENAME	SAL	COMM	DEPTNO
7900	JAMES	950	1000	30
8888	GACHON	1000	1000	30

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

04. 기본 테이블 변경을 방지하는 WITH READ ONLY 옵션과 WITH CHECK OPTION

WITH READ ONLY 옵션은 기본 테이블의 어떤 컬럼에 대해서도 뷰를 통한 내용 수정을 불가능하게 만드는 기능

② WITH CHECK OPTION은 뷰를 설정 할 때 조건으로 설정한 컬럼은 변경이 안되고 나머지 컬럼은 변경이 가능하기 때문에 COMM=1000으로 수정되었다.

③ WITH READ ONLY 옵션을 지정한 뷰

```
SQL> CREATE OR REPLACE VIEW VIEW_READ30  
2 AS  
3 SELECT EMPNO, ENAME, SAL, COMM, DEPTNO  
4 FROM EMP_COPY  
5 WHERE DEPTNO=30 WITH READ ONLY ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM VIEW_READ30 ;
```

EMPNO	ENAME	SAL	COMM	DEPTNO
7900	JAMES	950	1000	30
8888	GACHON	1000	1000	30

## 8장. 뷰

### □ 뷰 생성에 사용되는 다양한 옵션

04.기본 테이블 변경을 방지하는 WITH READ ONLY 옵션과 WITH CHECK OPTION

WITH READ ONLY 옵션은 기본 테이블의 어떤 컬럼에 대해서도 뷰를 통한 내용 수정을 불가능하게 만드는 기능

④ WITH READ ONLY 옵션을 기술한 VIEW\_READ30 뷰의 COMM을 모두 2000으로 수정시도(실패)

```
SQL> UPDATE VIEW_READ30 SET COMM=2000 ;  
UPDATE VIEW_READ30 SET COMM=2000
```

\*

1행에 오류:

ORA-42399: 읽기 전용 뷰에서는 DML 작업을 수행할 수 없습니다.

⑤ WITH READ ONLY 옵션은 뷰를 설정할 때 조건으로 설정한 컬럼뿐만 아니라 다른 컬럼에 대해서도 변경이 불가능(기본 테이블의 모든 컬럼 값 수정 불가)

```
SQL> UPDATE VIEW_READ30 SET DEPTNO=20 ;  
UPDATE VIEW_READ30 SET DEPTNO=20
```

\*

1행에 오류:

ORA-42399: 읽기 전용 뷰에서는 DML 작업을 수행할 수 없습니다.

## 8장. 뷰

### □ 뷰 를 활용하여 Top-N 구하기

Top-N이란 사원 중에서 입사일이 가장 빠른 사람 5명이 누구인가?를 뷰를 통해서 검색

#### 01.ROWNUM 컬럼 이해하기

사원 중에서 입사일이 빠른 사람 5명(TOP-5)만 얻어오는 방법은 무엇일까?

일련의 데이터를 임의의 순서로 정렬한 후 그 중 일부의 데이터만 출력할 수 있도록하여 구한다. TOP-N을 구하기 위해서는 ROWNUM과 인라인 뷰가 사용된다.

```
SQL> SELECT ROWNUM, ROWID, EMPNO, ENAME, HIREDATE  
2 FROM EMP ;
```

ROWNUM	ROWID	EMPNO	ENAME	HIREDATE
1	AAAR3sAAEAAAAACXAAA	7369	SMITH	80/12/17
2	AAAR3sAAEAAAAACXAAB	7499	ALLEN	81/02/20
3	AAAR3sAAEAAAAACXAAC	7521	WARD	81/02/22
4	AAAR3sAAEAAAAACXAAD	7566	JONES	81/04/02
5	AAAR3sAAEAAAAACXAAE	7654	MARTIN	81/09/28
6	AAAR3sAAEAAAAACXAAF	7698	BLAKE	81/05/01
7	AAAR3sAAEAAAAACXAAG	7782	CLARK	81/06/09
8	AAAR3sAAEAAAAACXAAH	7788	SCOTT	87/04/19
9	AAAR3sAAEAAAAACXAAI	7839	KING	81/11/17
10	AAAR3sAAEAAAAACXAAJ	7844	TURNER	81/09/08
11	AAAR3sAAEAAAAACXAAK	7876	ADAMS	87/05/23
12	AAAR3sAAEAAAAACXAAL	7900	JAMES	81/12/03
13	AAAR3sAAEAAAAACXAAM	7902	FORD	81/12/03
14	AAAR3sAAEAAAAACXAAN	7934	MILLER	82/01/23

14 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰 를 활용하여 Top-N 구하기

Top-N이란 사원 중에서 입사일이 가장 빠른 사람 5명이 누구인가?를 뷰를 통해서 검색

#### 01.ROWNUM 컬럼 이해하기

ROWID : 테이블의 특정 레코드를 랜덤하게 접근하기 위한 논리적 주소값

ROWNUM : 각 행의 일련번호

① 입사일이 빠른 사람 5명을 얻으려면 출력 데이터를 입사일의 오름차순으로 정렬한다.

```
SQL> SELECT EMPNO, ENAME, HIREDATE  
2 FROM EMP  
3 ORDER BY HIREDATE ;
```

EMPNO	ENAME	HIREDATE
7369	SMITH	80/12/17
7499	ALLEN	81/02/20
7521	WARD	81/02/22
7566	JONES	81/04/02
7698	BLAKE	81/05/01
7782	CLARK	81/06/09
7844	TURNER	81/09/08
7654	MARTIN	81/09/28
7839	KING	81/11/17
7900	JAMES	81/12/03
7902	FORD	81/12/03
7934	MILLER	82/01/23
7788	SCOTT	87/04/19
7876	ADAMS	87/05/23

14 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰 를 활용하여 Top-N 구하기

Top-N이란 사원 중에서 입사일이 가장 빠른 사람 5명이 누구인가?를 뷰를 통해서 검색

01.ROWNUM 컬럼 이해하기

② 입사일 오름차순으로 정렬하는 쿼리문에 ROWNUM컬럼을 출력하여 순위를 확인한다.

```
SQL> SELECT ROWNUM, EMPNO, ENAME, HIREDATE  
2 FROM EMP  
3 ORDER BY HIREDATE ;
```

ROWNUM	EMPNO	ENAME	HIREDATE
1	7369	SMITH	80/12/17
2	7499	ALLEN	81/02/20
3	7521	WARD	81/02/22
4	7566	JONES	81/04/02
6	7698	BLAKE	81/05/01
7	7782	CLARK	81/06/09
10	7844	TURNER	81/09/08
5	7654	MARTIN	81/09/28
9	7839	KING	81/11/17
12	7900	JAMES	81/12/03
13	7902	FORD	81/12/03
14	7934	MILLER	82/01/23
8	7788	SCOTT	87/04/19
11	7876	ADAMS	87/05/23

14 개의 행이 선택되었습니다.

. 5번 위치



## 8장. 뷰

### □ 뷰를 활용하여 Top-N 구하기

실습 : 뷰와 ROWNUM 컬럼으로 TOP-N 구하기(입사일 기준 오름차순으로 정렬하는  
쿼리문으로 뷰를 만들기)

```
SQL> CREATE OR REPLACE VIEW VIEW_HIRE
2 AS
3 SELECT EMPNO, ENAME, HIREDATE
4 FROM EMP
5 ORDER BY HIREDATE ;
```

뷰가 생성되었습니다.

```
SQL> SELECT ROWNUM, EMPNO, ENAME, HIREDATE
2 FROM VIEW_HIRE ;
```

ROWNUM	EMPNO	ENAME	HIREDATE
1	7369	SMITH	80/12/17
2	7499	ALLEN	81/02/20
3	7521	WARD	81/02/22
4	7566	JONES	81/04/02
5	7698	BLAKE	81/05/01
6	7782	CLARK	81/06/09
7	7844	TURNER	81/09/08
8	7654	MARTIN	81/09/28
9	7839	KING	81/11/17
10	7900	JAMES	81/12/03
11	7902	FORD	81/12/03
12	7934	MILLER	82/01/23
13	7788	SCOTT	87/04/19
14	7876	ADAMS	87/05/23

14 개의 행이 선택되었습니다.

## 8장. 뷰

### □ 뷰를 활용하여 Top-N 구하기

실습 : 뷰와 ROWNUM 컬럼으로 TOP-N 구하기(입사일 기준 오름차순으로 정렬하는 쿼리문으로 뷰를 만들기)

① 뷰에서 입사일이 빠른 순으로 5명만 출력하기

```
SQL> SELECT ROWNUM, EMPNO, ENAME, HIREDATE  
2  FROM VIEW_HIRE  
3  WHERE ROWNUM <= 5 ;
```

ROWNUM	EMPNO	ENAME	HIREDATE
1	7369	SMITH	80/12/17
2	7499	ALLEN	81/02/20
3	7521	WARD	81/02/22
4	7566	JONES	81/04/02
5	7698	BLAKE	81/05/01

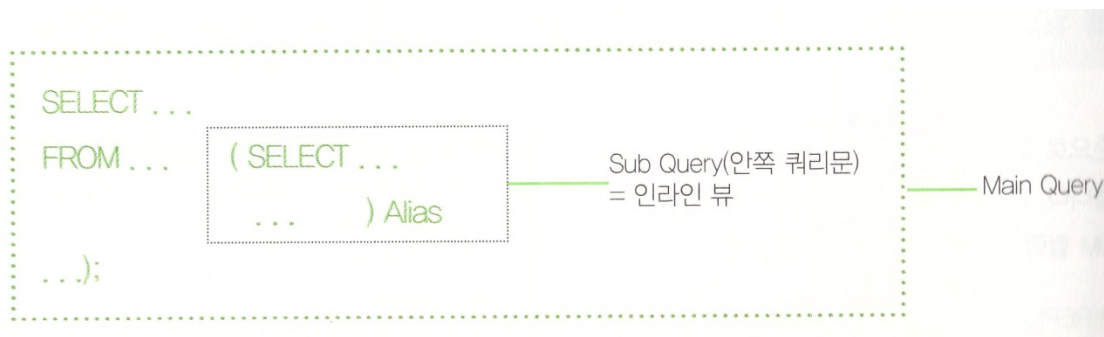
## 8장. 뷰

### □ 뷰 를 활용하여 Top-N 구하기

#### 02.인라인 뷰로 구하는 TOP-N

인라인 뷰는 SQL문에서 사용하는 서브 쿼리의 일종으로 보통 FROM절에 위치하여 테이블처럼 사용한다.

[형식]



## 8장. 뷰

### □ 뷰를 활용하여 Top-N 구하기

#### 02. 인라인 뷰로 구하는 TOP-N

인라인 뷰는 SQL문에서 사용하는 서브 쿼리의 일종으로 보통 FROM절에 위치하여 테이블 처럼 사용한다.

예제: 인라인 뷰를 사용해서 입사일이 빠른 5명을 출력하세요.

```
SQL> SELECT ROWNUM, EMPNO, ENAME, HIREDATE  
2 FROM< SELECT EMPNO, ENAME, HIREDATE  
3 FROM EMP  
4 ORDER BY HIREDATE>  
5 WHERE ROWNUM <= 5 ;
```

ROWNUM	EMPNO	ENAME	HIREDATE
1	7369	SMITH	80/12/17
2	7499	ALLEN	81/02/20
3	7521	WARD	81/02/22
4	7566	JONES	81/04/02
5	7698	BLAKE	81/05/01

## 8장. 뷰

### □ 뷰를 활용하여 Top-N 구하기

#### 02. 인라인 뷰로 구하는 TOP-N

실습: 인라인 뷰를 사용해서 입사일이 빠른 5명을 출력하세요.

```
SQL> CREATE OR REPLACE VIEW SAL_TOP5_VIEW
2 AS
3 SELECT ROWNUM AS RANKING, EMPNO, ENAME, SAL
4 FROM (SELECT EMPNO, ENAME, SAL
5        FROM EMP_COPY
6        WHERE SAL IS NOT NULL
7        ORDER BY SAL DESC
8       )
9 WHERE ROWNUM <= 5 ;
```

뷰가 생성되었습니다.

```
SQL> SELECT * FROM SAL_TOP5_VIEW ;
```

RANKING	EMPNO	ENAME	SAL
1	7839	KING	5000
2	7788	SCOTT	3000
3	7902	FORD	3000
4	7566	JONES	2975
5	7698	BLAKE	2850