

뷰(View)





뷰(View)

뷰 생성, 데이터 검색, 수정, 삭제

뷰를 이용한 DML 연산

인라인 뷰(Inline View)

- 뷰(View)는 테이블 또는 다른 뷰를 기초로 하는 논리적 테이블입니다.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	24000			90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000		100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000		102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000		103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4800		103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800		103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07	IT_PROG	4200		103	60
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	SALARY					
100	Steven	King	SKING	2003-06-17	24000	2002-08-17	FI_MGR	12008	101	100
101	Neena	Kochhar	NKOCHHAR	2005-09-21	17000	2002-08-16	FI_ACCOUNT	9000	108	100
102	Lex	De Haan	LDEHAAN	2001-01-13	17000	2005-09-28	FI_ACCOUNT	8200	108	100
103	Alexander	Hunold	AHUNOLD	2006-01-03	9000	2005-09-30	FI_ACCOUNT	7700	108	100
104	Bruce	Ernst	BERNST	2007-05-21	6000
105	David	Austin	DAUSTIN	2005-06-25	4800	2006-03-24	SA_REP	8600	0.2	149
106	Valli	Pataballa	VPATABAL	2006-02-05	4800	2006-04-23	SA_REP	8400	0.2	149
107	Diana	Lorentz	DLORENTZ	2007-02-07	4200	2007-05-24	SA_REP	7000	0.15	149
108	Nancy	Greenberg	NGREENBE	2002-08-17	12008	2008-01-04	SA_REP	6200	0.1	149
109	Daniel	Faviet	DFAVIET	2002-08-16	9000	2006-01-24	SH_CLERK	3200		120
110	John	Chen	JCHEN	2005-09-28	8200	2006-02-23	SH_CLERK	3100		120
111	Ismael	Sciarra	ISCIARRA	2005-09-30	7700	2007-06-21	SH_CLERK	2500		120
...	2008-02-03	SH_CLERK	2800		120
...	2004-01-27	SH_CLERK	4200		121
...
204	Hermann	Baer	HBAER	515.123.8888		2002-06-07	HR_REP	6500	101	40
205	Shelley	Higgins	SHIGGINS	515.123.8080		2002-06-07	PR_REP	10000	101	70
206	William	Gietz	WGIEZT	515.123.8181		2002-06-07	AC_MGR	12008	101	110
						2002-06-07	AC_ACCOUNT	8300	205	110

- 접근제어를 통한 자동 보안이 제공되므로 데이터베이스 액세스를 제한하기 위해 사용합니다.
- 복잡한 질의를 쉽게 만들어주기 때문에 사용자의 데이터 관리를 간단하게 해 줍니다.
- 논리적 데이터 독립성을 제공하기 때문에 데이터의 독립성을 허용합니다.
- 동일한 데이터의 다른 뷰를 나타낼 수 있기 때문에 동일 데이터에 대해 동시에 여러 사용자의 상이한 응용이나 요구를 지원해 줍니다.

View의 정의

가

하나 이상의 테이블에서 원하는 모든 데이터를 선택하여, 그들을 사용자 정의하여 나타낸 것이다.

단순 뷰 - 한 개의 원본테이블을 사용해서 생성

복합 뷰 - 두 개 이상의 이본 테이블에 의해 생성(조인을 통해 생성된..)

뷰는 단순 뷰와 복합 뷰가 있습니다. 근본적인 차이점은 DML(삽입, 갱신, 삭제) 작업에 관련되어 있습니다.

특징	단순 뷰	복합 뷰
테이블 수	하나	둘 이상
함수 포함	없음	있음
데이터 그룹 포함	없음	있음
뷰를 통한 DML	있음	없음

뷰 정보 확인은...

SELECT * FROM USER_VIEWS;

뷰를 만들기 위해서 뷰를 생성할 수 있는 권한이 있어야 합니다. 현재 접속한 사용자에게 부여된 롤(ROLE)을 확인하거나, 권한을 확인하여 뷰 생성권한이 있는지 조사할 수 있습니다.

다음 구문은 현재 사용자에게 주어진 롤(ROLE)을 출력합니다.

```
SQL> SELECT * FROM USER_ROLE_PRIVS;
```

	USERNAME	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE	OS_GRANTED
1	HR	CONNECT	NO	YES	NO
2	HR	RESOURCE	NO	YES	NO

다음 구문은 현재 사용자에게 주어진 권한을 출력합니다.

```
SQL> SELECT * FROM USER_SYS_PRIVS;
```

	USERNAME	PRIVILEGE	ADMIN_OPTION
1	HR	CREATE VIEW	NO
2	HR	UNLIMITED TABLESPACE	NO
3	HR	CREATE DATABASE LINK	NO
4	HR	CREATE SEQUENCE	NO
5	HR	CREATE SESSION	NO
6	HR	ALTER SESSION	NO
7	HR	CREATE SYNONYM	NO

- CREATE VIEW 문장 내에서 서브쿼리를 작성합니다.
- 서브쿼리는 복합 SELECT 구문을 포함할 수 있습니다.

CREATE VIEW 문장에서 서브쿼리를 내장하여 뷰를 생성할 수 있습니다. 뷰를 정의하는 서브쿼리는 조인, 그룹, 서브쿼리를 포함하는 복잡한 SELECT 구문을 포함할 수 있습니다.

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

다음 구문은 10번 부서의 모든 사원에 대한 세부사항을 포함하는 뷰를 생성합니다. 뷰 이름은 EMP_VIEW_DEPT10입니다.

```
SQL> CREATE VIEW emp_view_dept60
  2 AS SELECT  employee_id, first_name, last_name, job_id, salary
  3     FROM    employees
  4     WHERE   department_id=60;
```

View EMP_VIEW_DEPT10이(가) 생성되었습니다.

SQL Plus DESCRIBE 명령어를 사용하여 뷰의 구조를 디스플레이 할 수 있습니다.

```
SQL> DESC emp_view_dept60;
```

이름	널?	유형

EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)

서브쿼리에서 열 별칭을 포함함으로써 열 이름을 제어할 수 있습니다. 서브쿼리의 SELECT 절에서 수식을 사용할 경우에는 SELECT 절에 열 별칭을 사용하거나 CREATE VIEW 절에 열 이름을 포함해줘야 합니다.

다음 구문은 부서 60에 대해 직원 번호는 EMPNO, 이름은 NAME, 급여는 MONTHLY_SALARY 별칭을 갖는 뷰를 생성합니다.

```
SQL> CREATE VIEW emp_dept60_salary
2  AS SELECT
3      employee_id AS empno,
4      first_name || ' ' || last_name AS name,
5      salary AS monthly_salary
6  FROM    employees
7  WHERE   department_id=60;
```


두 개 이상 테이블로부터 값을 디스플레이 하는 뷰를 생성합니다.

다음 구문은 모든 사원의 아이디와 이름 그리고 부서이름과 직무이름을 출력합니다. 부서이름은 DEPARTMENTS 테이블과 조인을 통해서 알아내며, 직무이름은 JOBS 테이블과 조인을 통해서 알 수 있습니다.

```
SQL> CREATE VIEW emp_view
2 AS SELECT
3     e.employee_id AS id,
4     e.first_name AS name,
5     d.department_name AS department,
6     j.job_title AS job
7 FROM   employees e
8 JOIN   departments d ON e.department_id = d.department_id
9 JOIN   jobs j ON e.job_id = j.job_id;
```

View EMP_VIEW이(가) 생성되었습니다.

- CREATE OR REPLACE VIEW 절을 사용하여 뷰를 수정합니다.
- CREATE VIEW 절에서 열 별칭들은 서브쿼리에서의 열과 동일한 순서로 나열됩니다.

```
CREATE OR REPLACE VIEW view_name ...
```

OR REPLACE 옵션은 비록 이 이름이 이미 존재할지라도 뷰가 생성될 수 있도록 해주므로, 그 소유자에 대한 이전 뷰 버전을 새로운 뷰 구문으로 대체합니다.

다음 구문은 EMP_DEPT60_SALARY 뷰를 수정합니다.

```
SQL> CREATE OR REPLACE VIEW emp_dept60_salary
2 AS SELECT
3     employee_id AS empno,
4     first_name || ' ' || last_name AS name,
5     job_id AS job,
6     salary
7 FROM     employees
8 WHERE    department_id=60;
```

- 뷰는 데이터베이스에서 기본 테이블을 기반으로 하기 때문에 데이터 손실 없이 뷰를 삭제합니다.

뷰를 제거하기 위해 DROP VIEW 문장을 사용합니다. 이 문장은 데이터베이스에서 뷰 정의를 제거합니다. 뷰 삭제는 뷰가 만들어진 기본 테이블에는 영향을 미치지 않습니다. 그 뷰에 기초하여 만들어진 뷰 또는 다른 어플리케이션은 무효화 됩니다.

뷰를 만든 사람 또는 DROP ANY VIEW 권한을 가진 사람만 뷰를 제거할 수 있습니다.

```
DROP VIEW view_name;
```

다음 구문은 뷰를 삭제합니다.

```
SQL> DROP VIEW emp_dept60_salary;
```



뷰의 규칙 정리

- 단순 뷰에서 DML연산을 수행할 수 있습니다.
- 뷰가 다음을 포함한다면 행을 제거할 수 없습니다.
 - 그룹 함수
 - GROUP BY 절
 - DISTINCT 키워드
- 뷰가 다음을 포함한다면 데이터를 수정할 수 없습니다.
 - 위의 행 제거 할 수 없는 조건
 - 표현식으로 정의된 열
 - ROWNUM 의사열
- 뷰가 다음을 포함한다면 데이터를 추가할 수 없습니다.
 - 위의 행 제거와 수정 할 수 없는 조건
 - 뷰에 의해 선택되지 않은 NOT NULL 열이 기본 테이블에 있을 때



뷰(View)

뷰 생성, 데이터 검색, 수정, 삭제

뷰를 이용한 DML 연산

인라인 뷰(Inline View)

- 뷰에 대한 DML 연산이 뷰의 조건을 만족할 때만 수행되도록 합니다.
- 뷰에서 부서번호를 변경하려고 하면 WITH CHECK OPTION 제약조건에 위배 (ORA-01402)되기 때문에 실패하게 됩니다.

View에 값이 수정되면 실제 테이블에도 반영이 되기 때문에 뷰를 읽기 용도로만 사용할 수 있습니다.

with check option -조건 컬럼 값을 변경하지 못하게 하는 옵션.

with read only - select만 허용하는 옵션

다음 구문은 실습을 위한 뷰를 생성합니다.

```
SQL> CREATE OR REPLACE VIEW emp_dept60
  2 AS SELECT
  3     employee_id, first_name, hire_date, salary, department_id
  4   FROM   emps
  5   WHERE  department_id=60
  6   WITH CHECK OPTION CONSTRAINT emp_dept60_ck;
```

View EMP_DEPT60이(가) 생성되었습니다.

다음은 위에서 생성된 뷰를 이용해 105번 사원의 부서를 10으로 변경을 시도하는 구문입니다.

```
SQL> UPDATE emp_dept60
  2 SET    department_id=10
  3 WHERE  employee_id=105;
```

오류 보고 -

ORA-01402: view WITH CHECK OPTION where-clause violation

- 뷰의 정의에 WITH READ ONLY 옵션을 추가하여 DML 연산이 수행될 수 없게 합니다.
- 뷰 임의 행에서 DML을 수행하면 에러가 발생합니다.

WITH READ ONLY 옵션으로 뷰를 생성하여 뷰에서 DML 연산이 수행될 수 없게 합니다.

다음 코드는 뷰에서 임의의 DML 연산을 하지 못하도록 EMP_DEPT60 뷰를 수정합니다.

```
SQL> CREATE OR REPLACE VIEW emp_dept60
2 AS SELECT
3     employee_id, first_name, hire_date, salary, department_id
4 FROM   emps
5 WHERE  department_id=60
6 WITH READ ONLY;
```

View EMP_DEPT60이(가) 생성되었습니다.

뷰에서 행을 제어하려고 하면 에러가 발생합니다.

```
SQL> DELETE FROM emp_dept60
2 WHERE employee_id=105;

DELETE FROM emp_dept60
WHERE employee_id=105
오류 발생 명령행: 1 열: 13
오류 보고 -
SQL 오류: ORA-42399: cannot perform a DML operation on a read-only view
42399.0000 - "cannot perform a DML operation on a read-only view"
```

- 인라인 뷰는 SQL문에서 사용 가능한 별칭(상관 이름)을 사용하는 서브쿼리입니다.
- 인라인 뷰는 기본질의 FROM절에 명명된 서브쿼리를 사용하는 것과 유사합니다.
- 인라인 뷰는 스키마 객체가 아닙니다.

인라인 뷰는 FROM 절에 서브쿼리가 온 것을 말합니다. FROM 절에는 테이블 또는 뷰가 올 수 있습니다. 그런데 서브쿼리를 FROM 절에 사용해 하나의 테이블 또는 뷰처럼 사용할 수 있습니다. 뷰도 하나의 독립적인 SELECT문이므로 FROM 절에 사용하는 서브쿼리도 하나의 뷰로 볼 수 있습니다. 그래서 FROM 절에 오는 뷰를 인라인 뷰라고 부릅니다.

```
SQL> SELECT row_number, first_name, salary
2  FROM  (SELECT first_name, salary,
3           row_number() OVER (ORDER BY salary DESC) AS row_number
4           FROM employees
5           ORDER BY salary DESC)
6  WHERE row_number BETWEEN 1 AND 10;
```




Chapter 11

수고하셨습니다