

Introduction to Oracle9i : SQL

Chapter9. 뷰의 생성 및 관리

- ▶ 뷰의 개념을 이해하고 생성하는 문법을 익힌다.
- ▶ 뷰를 통해 데이터를 검색, 조작한다.
- ▶ TOP-N 분석 기능을 숙지한다.

※ 데이터베이스 객체

객체	의미
테이블	저장구조의 기본 단위, 행과 열로 구성
뷰	하나 이상의 테이블로부터 구성한 데이터들의 논리적인 집합
시퀀스	순차적인 숫자값 생성
인덱스	검색 속도를 향상
시노님	객체의 별도 이름

1. 뷰의 정의

(1) 뷰의 개념

- 테이블이나 다른 VIEW를 기초로 한 가상의 테이블
- 데이터를 저장하지 않고 질의 문장을 저장

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7934	MILLER	CLERK	7782	23-JAN-82	1300		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

1. 뷰의 정의

(2) 뷰의 용도

- 데이터 접근 제한
- 복잡한 쿼리문을 간단하게 수행 가능
- 데이터 독립성 제공
- 같은 데이터에 대한 다양한 관점 제공

1. 뷰의 정의

(3) 뷰의 종류

1) 단순 뷰

- 오직 하나의 테이블에서 데이터가 유래
- 데이터 그룹 또는 함수를 불포함
- 뷰를 통해 DML 수행 가능

2) 복합 뷰

- 다중 테이블에서 데이터가 유래
- 데이터 그룹 또는 함수를 포함
- 뷰를 통한 DML을 항상 허용하지 않음

2. 뷰의 생성

(1) 뷰의 생성

- **CREATE VIEW**문 내에 서브쿼리 내장
- 서브쿼리는 복합 **SELECT** 구문을 포함할 수 있고 **ORDER BY** 절은 포함 불가

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW 뷰이름  
[(별칭[,별칭,...])]  
AS 서브쿼리  
[WITH CHECK OPTION [CONSTRAINT 제약조건명 ]]  
[WITH READ ONLY] ;
```

```
CREATE VIEW emp_20  
AS SELECT *  
FROM emp  
WHERE deptno = 20 ;
```

2. 뷰의 생성

(2) 별칭을 사용한 뷰의 생성

- 별칭을 사용하여 뷰를 생성한 경우 별칭으로 검색

```
CREATE VIEW emp_30
  AS SELECT empno emp_no, ename name, sal salary
  FROM      emp
  WHERE      deptno = 30 ;
```

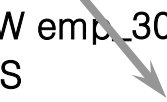


```
SELECT empno emp_no, ename name, sal salary
FROM    emp_30 ;
```


2. 뷰의 생성

(3) 뷰를 통한 데이터 검색

```
SELECT *  
FROM emp_30 ;
```



```
CREATE VIEW emp_30  
AS  
SELECT empno emp_no,ename name, sal salary  
FROM emp  
WHERE deptno = 30 ;
```

2. 뷰의 생성

(4) 뷰의 구조 및 이름 확인

```
SELECT *  
FROM   user_views ;
```

3. 뷰의 수정

(1) CREATE OR REPLACE문

```
CREATE OR REPLACE VIEW emp_10 (employee_no, employee_name, job_title)
AS SELECT empno, ename, job
FROM emp
WHERE deptno = 10 ;
```

- ALTER VIEW문은 사용 불가능

4. 복합 뷰의 생성

(1) 복합 뷰 생성구문

```
CREATE VIEW dept_sum (name, minsal, maxsal, avgsal)
  AS SELECT d.dname, MIN(e.sal), MAX(e.sal), AVG(e.sal)
  FROM dept d, emp e
  WHERE d.deptno = e.deptno
  GROUP BY d.dname ;
```

5. 뷰에서의 DML 수행

(1) 단순 VIEW에서는 DML연산 수행

(2) DELETE를 수행할 수 없는 경우

- 그룹 함수
- GROUP BY절
- DISTINCT키워드
- ROWNUM 의사열

5. 뷰에서의 DML 수행

(3) UPDATE를 수행할 수 없는 경우

- 그룹 함수
- GROUP BY절
- DISTINCT키워드
- ROWNUM 의사열
- 표현식으로 정의된 열

(4) INSERT를 수행할 수 없는 경우

- 그룹 함수
- GROUP BY절
- DISTINCT키워드
- ROWNUM 의사열
- 표현식으로 정의된 열
- 선택되지 않은 NOT NULL열이 기본 테이블에 있는 경우

6. WITH CHECK OPTION

- 뷰의 생성 시 제시된 조건에 만족하는 값만 입력받을 수 있도록 뷰 생성 시 제시하는 옵션

```
CREATE OR REPLACE VIEW emp_20
  AS SELECT *
  FROM emp
  WHERE deptno = 20
  WITH CHECK OPTION CONSTRAINT emp_20_ck ;
```



```
UPDATE emp_20
SET deptno = 30
WHERE empno = 7566;
```

```
UPDATE emp_20
```

```
*
```

```
ERROR at line 1:
```

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

7. DML연산의 부정

- **WITH READ ONLY** 옵션으로 뷰를 생성하면 뷰에 대해 **DML연산을 수행 불가능**
- **READ ONLY** 뷰에 **DML연산 수행 시 에러 발생**

```
CREATE OR REPLACE VIEW emp_10  
AS SELECT empno, ename, job  
FROM emp  
WHERE deptno = 10  
WITH READ ONLY ;
```


8. 뷰의 제거

```
DROP VIEW 뷰이름 ;
```

```
DROP VIEW emp_10 ;
```

9. 인라인 뷰

- 1회성 뷰로 FROM 절에 작성하는 서브쿼리
- 인라인 뷰는 정식 스키마 객체는 아님

```
SELECT a.last_name, a.salary, a.department_id, b.maxsal
FROM   employees e, (SELECT department_id, MAX (salary) maxsal
                     FROM   employees
                     GROUP BY department_id) b
WHERE  a.department_id = b.department_id
AND    a.salary < b.maxsal ;
```

10. TOP-N 분석

- 가장 크거나 가장 작은 n개의 값을 도출하는 기능
 - 가장 잘 팔리는 상품 품목 10개
 - 가장 월급을 적게 받는 사원 10명

(1) TOP-N 분석 구문

```
SELECT [컬럼명] , ROWNUM  
FROM    (SELECT [컬럼명]  
          FROM    테이블명  
          ORDER BY Top-N 컬럼명 )  
WHERE   ROWNUM <= N ;
```

10. TOP-N 분석

(2) TOP-N 분석 수행

월급을 가장 많이 받는 사원 3명 검색

```
SELECT ROWNUM as RANK, last_name, salary
FROM   (SELECT last_name, salary
        FROM   employees
        ORDER BY salary DESC )
WHERE  ROWNUM <= 3 ;
```