

다양한 예제로 쉽게 배우는

오라클 SQL 과 PL/SQL

10장 view 를 배웁니다

10. view 를 배웁니다

- View 란 가상의 테이블이다!

사원 테이블

사번	이름	성별	연락처	지역	연봉
1001	홍길동	남	111-1111	서울	3200
1002	유관순	여	222-2222	부산	3500
1003	신사임당	여	333-3333	광주	2900
1004	강감찬	남	444-4444	대전	3650
1005	김구	남	555-5555	간도	4200

V_사원

사번	이름	연락처
1001	홍길동	111-1111
1002	유관순	222-2222
1003	신사임당	333-3333
1004	강감찬	444-4444
1005	김구	555-5555

10. view 를 배웁니다

1. 단순 View (Simple View)

```
SCOTT>CONN / AS SYSDBA;  
SYS>GRANT CREATE VIEW TO scott ;
```

```
CREATE [OR REPLACE] [ FORCE | NOFORCE] VIEW view [ (alias, alias,.....)]  
AS sub-query  
[ WITH CHECK OPTION [CONSTRAINT 제약조건] ]  
[ WITH READ ONLY ]
```

- * OR REPLACE : 같은 이름의 View가 있을 경우 삭제 후 다시 생성합니다.
- * FORCE : 기본 테이블의 존재 여부에 상관없이 View 생성
- * NOFORCE : 기본 테이블이 존재할 경우에만 View 생성, 기본 값입니다
- * ALIAS : 기본 테이블의 칼럼 이름과 다르게 지정한 View의 칼럼 이름을 지정합니다.
- * WITH CHECK OPTION : 주어진 제약조건에 맞는 데이터만 입력 및 수정을 허용합니다.
- * WITH READ ONLY : SELECT 만 가능한 읽기 전용 뷰를 생성합니다.

10. view 를 배웁니다

-생성 예제 1:

professor 테이블의 profno, name, email, hpage 컬럼만 사용하는 View 를 생성하세요. View 이름은 v_prof 로 하세요.

```
SCOTT>CREATE OR REPLACE VIEW v_prof  
2 AS  
3 SELECT profno, name, email, hpage  
4 FROM professor ;
```

```
SCOTT>SELECT * FROM v_prof ;
```

10. view 를 배웁니다

```
SCOTT>CREATE INDEX idx_v_prof_name  
2  ON v_prof(name);  
ON v_prof(name)  
*
```

ERROR at line 2:

ORA-01702: a view is not appropriate here

View 에는 데이터가 없어서
인덱스를 생성 할 수 없습니다.
만약 View 가 느리다면
원본 테이블에 인덱스를 점검
해 보세요.

10. view 를 배웁니다

2. 복합 View (Complex View)

- 생성 예제 2:

Professor 테이블과 department 테이블을 조인하여 교수번호와 교수이름과 소속 학과이름을 조회하는 view 를 생성하세요. View 이름은 v_prof_dept2 로 하세요.

```
SCOTT>CREATE OR REPLACE VIEW v_prof_dept
2 AS
3   SELECT p.profno "교수번호"
4      ,   p.name "교수명"
5      ,   d.dname "소속학과명"
6   FROM professor p , department d
7   WHERE p.deptno = d.deptno ;
```

10. view 를 배웁니다

3. INLINE View (인라인 뷰) – 1회용 뷰

- 생성 예제1 :

Student 테이블과 department 테이블을 사용하여 학과별로 학생들의 최대 키와 최대 몸무게, 학과이름을 출력하세요.

```
SCOTT> SELECT d.dname "학과명"
2   ,    s.max_height "최대키"
3   ,    s.max_weight "최대몸무게"
4 FROM ( SELECT deptno1, MAX(height) max_height, MAX(weight) max_weight
5         FROM student
6         GROUP BY deptno1) s , department d
7 WHERE s.deptno1 = d.deptno ;
```


10. view 를 배웁니다

- Inline View 연습문제 1:

Student 테이블과 department 테이블을 사용하여 학과별로 가장 키가 큰 학생들의 이름과 키, 학과이름을 Inline View 를 사용하여 아래와 같이 출력하세요.

학과이름	최대키	학생이름	키
소프트웨어공학과	168	이미경	168
전자공학과	177	김재수	177
기계공학과	182	박동호	182
컴퓨터공학과	182	일지매	182
문헌정보학과	184	노정호	184
멀티미디어공학과	179	김주현	179

10. view 를 배웁니다

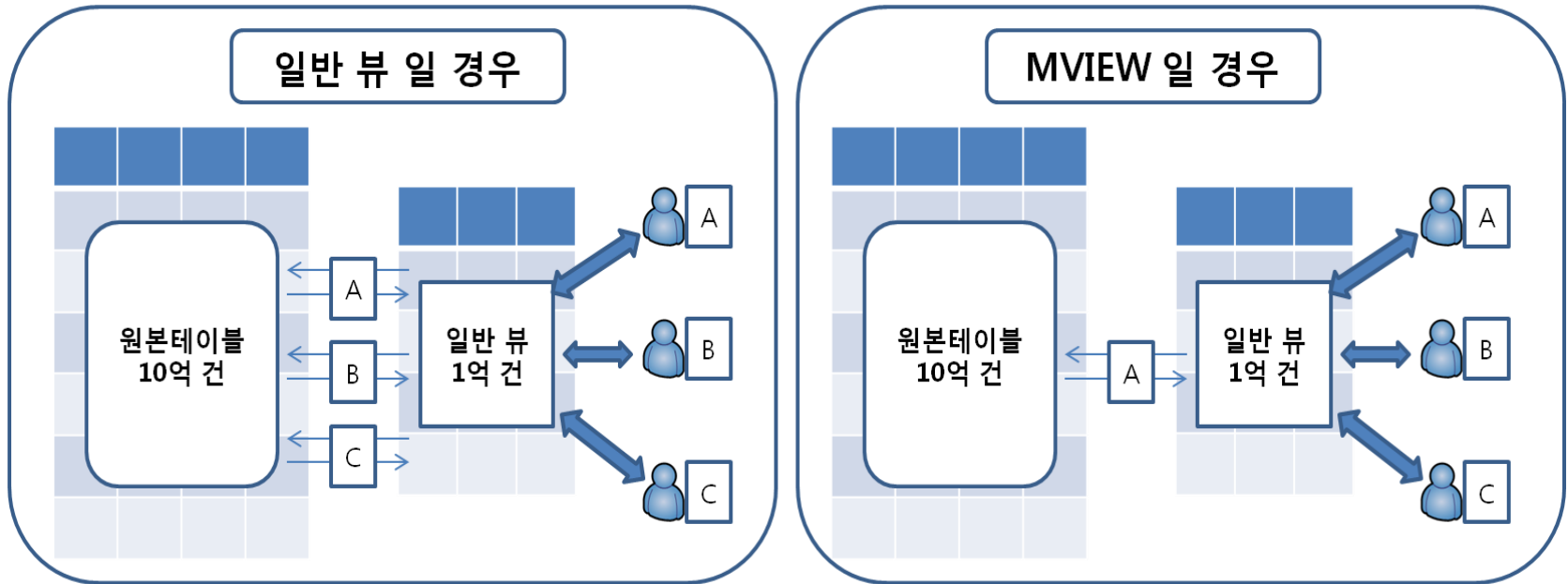
-Inline View 연습문제 2:

Student 테이블에서 학생의 키가 동일 학년의 평균 키 보다 큰 학생들의 학년과 이름과 키, 해당 학년의 평균 키를 출력하되 Inline View 를 사용해서 아래와 같이 출력 하세요.(학년 컬럼으로 오름차순 정렬해서 출력하세요)

학년	이름	키	평균키
1	안은수	175	170.4
1	인영민	173	170.4
1	김주현	179	170.4
2	일지매	182	175.6
2	노정호	184	175.6
3	오나라	177	166.6
3	임세현	171	166.6
4	서진수	180	175.8
4	김재수	177	175.8
4	박동호	182	175.8

10. view 를 배웁니다

4. Materialized View (MVIEW)



10. view 를 배웁니다

2) Mview 생성하기

```
SCOTT>CONN / AS SYSDBA ;  
SYS>GRANT query rewrite TO scott ;  
SYS>GRANT create materialized view TO scott ;  
SYS>CONN scott/tiger ;
```

```
SCOTT> CREATE MATERIALIZED VIEW mv_prof  
2 BUILD IMMEDIATE  
3 REFRESH  
4 ON DEMAND  
5 COMPLETE  
6 ENABLE QUERY REWRITE  
7 AS  
8 SELECT profno , name , pay  
9 FROM professor  
10 WHERE deptno in (101,102,103) ;
```

10. view 를 배웁니다

-생성 문법 설명 :

* 2행: Mview 를 생성하면서 서브쿼리 부분을 수행해서 데이터를 가져 오라는 뜻입니다.

* 3행, 4행: 원본 테이블에 데이터가 변경 되었을 경우 MView 와 언제 어떻게 동기화를 시킬 건지에 대한 옵션입니다. 4 행의 ON DEMAND 옵션은 사용자가 수동으로 동기화 명령을 수행해서 동기화 시키는 것이고 ON COMMIT 옵션도 쓸 수 있는데 이것은 원본테이블에 데이터 변경 후 Commit 이 발생하면 자동으로 동기화 시키라는 의미입니다. 그런데 ON COMMIT 옵션은 원본테이블에 데이터 변경이 많을 경우 동기화 시키느라 많은 부하를 발생 시킬 수 있기 때문에 원본테이블에 그룹함수를 사용하거나 Mview 에 조인이 되는 SQL만 있거나 또는 Group by 절에 사용된 컬럼에 COUNT 함수가 사용되는 경우에만 사용이 가능합니다.

* 5행 : REFRESH 를 하는 방법도 4가지가 있습니다.

- **COMPLETE** : MVIEW 내의 데이터 전체가 원본 테이블과 동기화 되는 방법입니다.

이 옵션을 사용하려면 ATOMIC_REFRESH=TRUE 와 COMPLETE 로 설정이 되어야 합니다. 데이터가 많을 경우 시간이 많이 소요됩니다.

- **FAST** : 원본 테이블에 새로운 데이터가 입력될 경우 그 부분만 Mview 로 동기화 하는 방법입니다. 이 방법은 Direct Path 나 Mview log 파일 을 사용하여 동기화 하게 됩니다.

- **FORCE** : FAST 방법이 가능한지 살펴보고 불가능하면 COMPLETE 방법을 사용하여 동기화 하게 됩니다.

- **NEVER** : 동기화를 하지 않습니다.

10. view 를 배웁니다

Mview 에는 데이터가 존재하므로 인덱스 생성도 가능합니다.

```
SCOTT>CREATE INDEX idx_mv_prof_pay
2 ON mv_prof(pay);
```

3) MView 관리하기

- 수동으로 원본 테이블과 Mview 데이터 동기화 하기

```
SCOTT>INSERT INTO professor(profno,name,id,position,pay,hiredate,deptno)
2 VALUES(5000,'나교수','improf','조교수',320,SYSDATE,101) ;
```

```
SCOTT> COMMIT ;
```

테스트 위해 이 데이터를 추가하세요

10. view 를 배웁니다

- 동기화 전 데이터 건 수 확인하기

```
SCOTT>SELECT COUNT(*) FROM professor WHERE deptno IN(101,102,103) ;
```

COUNT(*)
10

```
SCOTT>SELECT COUNT(*) FROM mv_prof ;
```

COUNT(*)
9

10. view 를 배웁니다

- DBMS_MVIEW 패키지로 동기화를 수행합니다

```
SCOTT>BEGIN
2  DBMS_MVIEW.REFRESH('MV_PROF') ;
3  END ;
4  /
```

PL/SQL procedure successfully completed.

```
SCOTT>SELECT COUNT(*) FROM mv_prof;
```

```
COUNT(*)
```

```
-----
```

```
10    <- 동기화가 완료되었습니다
```


10. view 를 배웁니다

- 다른 동기화 명령어들

* **DBMS_MVIEW.REFRESH_DEPENDENT('ABC') ;**

이 명령어는 ABC 라는 테이블을 사용하는 모든 MVIEW 를 찾아서 한꺼번에 동기화 하라는 의미입니다.

* **DBMS_MVIEW.REFRESH_ALL_MVIEWS ;**

이 명령어는 해당 사용자가 만든 모든 MVIEW를 동기화 하라는 의미입니다.

- Mview 조회하기

```
SCOTT>SELECT mview_name,query  
2 FROM user_mviews  
3 WHERE mview_name='MV_PROF';
```