

<보충 설명> View에서 쓰이는 DML 문장들에 대한 보충 설명

-> 뷰를 통한 데이터 조작

1) 단순 뷰에서는 대개 DML 작업을 수행할 수 있다.

2) 뷰에 다음 항목이 포함되어 있으면 행을 제거할 수 없다.

- 그룹 함수

- GROUP BY 절

- DISTINCT 키워드

3) 뷰에 다음 항목이 포함되어 있으면 뷰의 데이터를 수정할 수 없다.

- 그룹 함수

- GROUP BY 절

- DISTINCT 키워드

- 표현식으로 정의된 열

4) 뷰에 다음 항목이 포함되어 있으면 뷰를 통해 데이터를 추가할 수 없다.

- 그룹 함수

- GROUP BY 절

- DISTINCT 키워드

- 표현식으로 정의된 열

- 뷰에서 선택되지 않은 기본 테이블의 NOT NULL 열

위 개념들을 쉽게 설명하기 위해서 다음과 같은 테이블을 새로 만들었다.

Emp_ID	Emp_Name	Dep_ID	Salary
101	철수	10	56000
102	유리	20	45000
103	맹구	20	34000
104	짱구	30	23000
105	훈이	10	12000
106	흰둥이	30	10000
107	유리	10	15000

[EMPTEST] Number는 Primary Key, Name은 Not Null, Dep_ID는 Group By 함수를 이
용하기 위해 Salary로 통계를 내기 위해 삽입하였음.

우선 이 테이블 내에서 부서 이름, 직원 이름, 봉급에 대해서 뷰로 만들겠다.

이 뷰의 이름은 EMP_VU_GEN으로 하겠다.

Create Or Replace View EMP_VU_GEN

AS

```
Select Dep_ID, Emp_Name, Salary  
from EMPTEST;
```

1)번에서 유리의 봉급을 46000으로 변경해 보겠다.

Update EMP_VU_GEN

Set Salary=46000

where Emp_Name='유리';

결과는 EMPTEST 테이블의 값과 뷰의 값을 통해서 확인을 해보면 다음과 같다.

102 유리 20 46000 -> EMPTEST 테이블의 값

20 유리 46000 -> EMP_VU_GEN 테이블의 값

이처럼 뷰의 값이 변경하게 된다면, 차후에 본래 테이블의 값도 변경이 된다.

이 뷰와 같은 경우에는 Update/Delete 문장에만 적용이 된다. Insert를 적용하기 위해서는 4번의 개념 설명에서 참조하면 되겠다.

2)번부터 설명하기 위해서 하나는 직원 이름들에 대해서 Distinct를 해보겠고, 하나는 Group By 함수를 이용하여 부서 번호 별 봉급 합계를 내는 뷰를 만들어 보겠다. 또한 표현식으로 정의된 열에 대해서는 직원 이름과 연봉에 대해서 뷰로 만들 것이고, 마지막으로 데이터 추가에 대해서 설명하는 것으로 유인물을 마치겠다.

1. Distinct를 이용한 뷰에서 DML을 적용 못하는 이유

Create Or Replace View EMP_VU_DIST

AS

Select Distinct Emp_Name

from EMPTEST;

이처럼 작성을 하게 되면 유리가 2명이더라도 결과는 6개만 나오게 된다. 하나 Distinct를 이용하게 되면 왜 DML 자체를 못 쓰게 되는 걸까? 비록 짱구의 이름을 짱9로 고치려도 해도 오류가 걸린다. 그리고 철수라는 이름을 없애려도 오류가 걸리게 된다. 물론 봉미선의 데이터를 삽입하는 것도 불가능하다. 이를 쉽게 설명하기 위해 지난번에 배웠던 인덱스의 개념 중 ROWID 개념을 이용해서 설명하겠다. 주소는 어찌됐든 이해를 위해 적는 것이니 참조만 하길 바란다.

EMP_NAME	RowID
철수	0000000001
유리	0000000002
맹구	0000000003
짱구	0000000004
훈이	0000000005
흰둥이	0000000006
유리	0000000007

-> Distinct ->

EMP_NAME	RowID
철수	0000000001
유리	0000000006
맹구	0000000003
짱구	0000000004
짱구	0000000005
훈이	0000000006
흰둥이	0000000007

View를 Distinct를 하게 되더라도 View에 있는 RowID의 값들은 아이러니하게도 7개로 그대로 저장이 된다. 하나 Distinct를 쓰게 된다면 RowID는 무슨 소용인가... 라는 개념이 되어 버린다. 결국 Distinct 내부에 있는 값들에 대해서 RowID는 무슨 값을 가리키는지에 대해서는 그런 거 없기 때문에 유리가 2개라서 수정/삭제에 제한을 주는 것도 당연하고, 철수나 흰둥이처럼 유일한 값에도 DML로 터치를 못하는 이유가 바로 그것이다.

2. Group By를 이용한 뷰에서 DML을 적용 못하는 이유

Create Or Replace View EMP_VU_Group

AS

Select dep_id, SUM(Salary) "Sum"

```
from EMPTEST
```

```
Group By dep_ID;
```

상식적으로 써보면 그룹 함수 내에 있는 값들에 대해서 정산하는 경우에 돈에 빵구가 생기면 곤란해지기 마련이다. 테이블의 예를 들어서 40번이라는 부서가 존재하는 직원이 본래 테이블에는 아직 없는데 이에 대해서 추가하려면 뷰에 저장된 값들에서 참조되지 않는 값이 지정되었기 때문에 추가가 되지 않고, 맹구의 봉급이 변경되지 않는 이상 합은 그대로 유지되기 때문에 합에 대한 값을 제어할 수 없는 건 당연하고, 실제로 저장된 부서 번호가 없어지지 않는 이상 삭제도 불가능하다. 이처럼 뷰에서 Group By, Group 함수가 이미 적용된 View에서 DML 문장을 쓰게 된다면 오류가 걸리는 것은 테이블 내에 세부 정보의 내용이 변경되지 않는 이상 DML를 절대 쓸 수 없게 된다.

3. 표현식으로 정의된 열

```
Create Or Replace View EMP_VU_ExpFunc
```

```
AS
```

```
Select emp_name, salary*12 "Annual"
```

```
from EMPTEST;
```

본래 테이블에서 지정된 값들을 통해서 표현식(연산자로 표현한 식, 물론 함수도 될 수 있다.)으로 된 값들에 대해서는 뷰로 저장할 때 함수를 거쳐서 넘어감으로서 값이 참조가 되지만, 실질적으로 표현식으로 된 값에 함수를 적용하지 않고 값을 추가/변경을 하게 된다면 테이블과 뷰와의 함수 규정을 어기기 때문에 Insert, Update 문을 쓸 수 없는 것이다. 같은 함수를 이용해서 추가/변경을 해도 마찬가지이다.

4. 데이터 추가할 때 살펴봐야 할 점

```
Create Or Replace View EMP_VU_InsertTest
```

```
AS
```

```
Select Emp_ID, EMP_NAME, Salary
```

```
from EMPTEST;
```

```
Insert Into EMP_VU_InsertTest
```

```
Values(108, '봉미선', 78000);
```

본래 Emp_ID와 EMP_NAME의 값은 애초에 Not Null이라는 제약조건(Primary Key는 Not Null, Unique 2가지 짚뽕한 거)이 걸려져 있기 때문에 이러한 값들이 뷰에 포함되어 있지 않으면 뷰를 통한 값을 추가하기 힘들다. 비록 108번 직원인 봉미선의 값을 추가하게 될지 언정 DEP_ID는 Null이게 된다. 그래도 DEP_ID는 Not Null이란 제약 조건이 없기 때문에 무사히 저장된다. 이처럼 View를 통해서 값을 추가하기 위해서는 꼭 Not Null이란 제약조건이 있는 컬럼을 꼭 넣길 바란다. 필수이다.