

## MERGE 문의 이해

- MERGE문은 조건에 따라서 데이터의 삽입,갱신,삭제 작업을 한번에 할 수 있다.
- 해당 행이 존재하는 경우 UPDATE(DELETE포함 할 수 있음)를 수행하고, 새로운 행일경우 INSERT를 수행한다.
- 대상 테이블에 대한 UPDATE/INSERT 조건은 ON절에 의해 결정된다.
- MERGE문에서 CLOB 사용시 업데이트 할 내용이 2000bytes가 넘을때 ORA-00600 오류가 발생하며, patch set 11.2.0.2 버전으로 해결할 수 있다.

### MERGE Syntax

```
1  MERGE [ hint ]
2      INTO [ schema. ] { table | view } [ t_alias ]
3      USING { [ schema. ] { table | view }
4              | subquery
5              } [ t_alias ]
6      ON ( condition )
7      WHEN MATCHED THEN
8      UPDATE SET column = { expr | DEFAULT }
9              [, column = { expr | DEFAULT } ]...
10     [ DELETE where_clause ]
11     WHEN NOT MATCHED THEN
12     INSERT [ (column [, column ]...) ]
13     VALUES ({ expr [, expr ]... | DEFAULT })
```

- INTO : DATA가 UPDATE되거나 INSERT 될 테이블 또는 뷰를 지정.
- USING : 비교할 SOURCE 테이블 또는 뷰나 서브쿼리를 지정, INTO절의 테이블과 동일하거나 다를 수 있다.
- ON : UPDATE나 INSERT를 하게 될 조건으로, 해당 조건을 만족하는 DATA가 있으면 WHEN MATCHED 절을 실행하게 되고, 없으면 WHEN NOT MATCHED 이하를 실행하게 된다.
- WHEN MATCHED : ON 조건절이 TRUE인 ROW에 수행 할 내용 (UPDATE, DELETE포함 될 수 있음)
- WHEN NOT MATCHED : ON 조건절에 맞는 ROW가 없을 때 수행할 내용 (INSERT)

## MERGE 예제1

아래 예제는 부서번호 20,30의 사원이 존재하면 급여를 10% 인상하고, 존재하지 않으면 급여가 1000보다 큰 사원정보를 등록하는 예이다. (INSERT, UPDATE 예제)

```

1  -- 부서번호 10, 20의 사원정보를 가지는 테스트 테이블을 생성하자
2  CREATE TABLE emp_merge_test
3  AS SELECT empno, deptno, sal FROM emp WHERE deptno IN (10, 20);
4
5
6  -- 데이터를 확인해 보자
7  SELECT * FROM emp_merge_test;
8
9
10 -- 사원이 존재하면 급여를 10% 인상하고, 없으면 INSERT 한다.
11 MERGE INTO emp_merge_test m
12 USING ( SELECT empno, deptno, sal      -- USING절에 뷰가 올수 있다.
13         FROM emp
14         WHERE deptno IN (20,30)) e
15 ON ( m.empno = e.empno)
16 WHEN MATCHED THEN
17     UPDATE SET m.sal = ROUND(m.sal*1.1)
18 WHEN NOT MATCHED THEN
19     INSERT (m.empno, m.deptno, m.sal)
20     VALUES (e.empno, e.deptno, e.sal)
21     WHERE e.sal > 1000      -- INSERT 절의 조건절도 지정이 가능하다
22 COMMIT;
23
24
25 -- 20부서의 급여가 10%증가했고, 30부서는 등록되었는지 확인해 보자
26 SELECT * FROM emp_merge_test;
27
28
29 -- 다음 테스트를 위해서 emp_merge_test 테이블에 30부서 데이터를 삭제하자
30 -- 부서번호 10과 20의 사원정보만 남을 것이다.
31 DELETE FROM emp_merge_test WHERE deptno = 30;
32 COMMIT;

```

## MERGE 예제2

아래는 부서번호 10의 사원 급여를 10% 인상하고, 부서번호 20의 사원정보는 삭제하며, 부서번호 30의 사원 급여를 20% 인상하는 예이다. (INSERT, UPDATE, DELETE 예제)

```

1  -- 30 부서가 삭제되었는지 확인한다.
2  SELECT * FROM emp_merge_test;
3
4
5  -- MERGE 문 작성
6  MERGE INTO emp_merge_test m
7  USING emp e
8  ON (m.empno = e.empno)
9  WHEN MATCHED THEN
10     UPDATE SET m.sal = ROUND(m.sal*1.1)
11     DELETE WHERE (m.deptno = 20)      -- 부서번호 20의 사원정보는 삭제.
12 WHEN NOT MATCHED THEN
13     INSERT (m.empno, m.deptno, m.sal)
14     VALUES (e.empno, e.deptno, ROUND(e.sal*1.2));
15 COMMIT;
16
17
18 -- 정상적으로 변경되었는지 확인해 보자
19 SELECT * FROM emp_merge_test;
20
21
22 -- 테스트 테이블은 삭제하자
23 DROP TABLE emp_merge_test;

```