Ejemplo paso a paso de la cláusula MERGE

Ejemplo de la funcionalidad MERGE que de manera general puede *actualizar*, *borrar* e *insertar* en una sola instrucción.

```
MERGE INTO TABLA1
   USING (TABLA2)
   ON (CONDICION DE UNION)
   WHEN MATCHED THEN UPDATE SET columna = expresion WHERE (CONDICION
WHEN-MATCHED)
     DELETE WHERE (CONDICION DELETE)
   WHEN NOT MATCHED THEN INSERT (columas)
     VALUES (valores)
     WHERE (CONDICION NOT-MATCHED)
```

- TABLA1 puede ser una tabla o una vista y se les conoce como tabla DESTINO
- TABLA2 puede ser una tabla, vista o un **subquery**, es llamada también ORIGEN, pues de ahí se van a tomar los datos a actualizar o insertar
- CONDICION DE UNION liga ambas tablas
- la sub-sentencia UPDATE, después del SET es muy similar a la tradicional, es decir, se pueden actualizar varias columnas, pero me parece que hay algunas columnas como la usada para la union que parece no pueden actualizarse, eso revisarlo en la documentación.
- El DELETE es una subclausula de WHEN MATCHED THEN UPDATE SET y en cualquier caso, los WHERE se marcan como opcionales.
- El INSERT que no tiene INTO, es similar a la sentencia original, con columnas separadas de los valores y separado por VALUES, y sólo va con la clausula WHEN NOT MATCHED THEN INSERT.

Para el ejemplo, se utilizarán las tablas del módelo HR que viene en la versión Express de Oracle 11g.

Primero se crea una tabla auxiliar llamada *bonuses*, se pone un truncate en la lista de las siguientes instrucciones por si desean repetir el ejemplo más de una vez, se llena y se le quitan algunos registros para que sea más fácil de observar y comprender el resultado:

```
CREATE TABLE bonuses (employee_id NUMBER, bonus NUMBER DEFAULT 100);

truncate table bonuses;

INSERT INTO bonuses(employee_id)
   (SELECT e.employee_id FROM employees e
   GROUP BY e.employee_id);

delete from bonuses where employee_id < 150
or employee_id between 156 and 165
or employee_id > 170;

SELECT * FROM bonuses;
```

Después de ejecutar el script anterior quedan los siguientes registros:

```
EMPLOYEE_ID      BONUS
----------- ----------
        150        100
        151        100
        152        100
        153        100
        154        100
        155        100
        166        100
        167        100
        168        100
        169        100
        180        100
        181        100
        182        100

 13 filas seleccionadas
```

Y ahora se va a ejecutar el siguiete MERGE y posteriormente se revisarán los registros que quedan en la tabla (se han remarcardo en rojo los que ya no aparecerán):

```
MERGE INTO bonuses D
   USING (SELECT employee_id, salary, department_id FROM employees
   WHERE department_id = 80) S
   ON (D.employee_id = S.employee_id)
   WHEN MATCHED THEN UPDATE SET D.bonus = D.bonus + S.salary*.01
     DELETE WHERE (S.salary > 8000)
   WHEN NOT MATCHED THEN INSERT (D.employee_id, D.bonus)
     VALUES (S.employee_id, S.salary*.5 )
     WHERE (S.salary <= 8000);
```

Que arroja:

```
19 filas fusionadas.
```

Pero si se vuelve a ver el listado de la tabla *bonues*, ya no hay 13 filas, ni 19, depués del MERGE hay 17. Y es que hizo las tres operaciónes antes descritas:

1. Se eliminaron los que están remarcados en rojo del listado anterior.
2. Se actualizaron los que están remarcados en amarillo del siguiente listado.
3. Se insertaron los remarcados en verde del siguiente listado
4. Y no se vieron afectados los que están de color azul.

```
EMPLOYEE_ID      BONUS
----------- ----------
        153        180
        154        175
        155        170
        159       4000
        160       3750
        161       3500
        164       3600
```

```
       165       3400
       166        164
       167        162
       171       3700
       172       3650
       173       3050
       179       3100
       180        100
       181        100
       182        100

17 filas seleccionadas
```

Se irá desglosando la sentencia, y en particular todo gira aledeor de los empleados del departamento 80.

```
MERGE INTO bonuses D
   USING (SELECT employee_id, salary, department_id FROM employees
   WHERE department_id = 80) S
   ON (D.employee_id = S.employee_id)
   WHEN MATCHED THEN UPDATE SET D.bonus = D.bonus + S.salary*.01
     DELETE WHERE (S.salary > 8000)
   WHEN NOT MATCHED THEN INSERT (D.employee_id, D.bonus)
     VALUES (S.employee_id, S.salary*.5 )
     WHERE (S.salary <= 8000);
```

Si son del departamento 80, hay dos clausulas que ejecutar, una, actualizar su bono con el uno porciento de su salario adicional y dos, borrar los que ganen más de 8,000.

```
MERGE INTO bonuses D
   USING (SELECT employee_id, salary, department_id FROM employees
   WHERE department_id = 80) S
   ON (D.employee_id = S.employee_id)
   WHEN MATCHED THEN UPDATE SET D.bonus = D.bonus + S.salary*.01
     DELETE WHERE (S.salary > 8000)
   WHEN NOT MATCHED THEN INSERT (D.employee_id, D.bonus)
     VALUES (S.employee_id, S.salary*.5 )
     WHERE (S.salary <= 8000);
```

Si para el caso de cuando los empleados no son del departamento 80, hay una clausula INSERT que ejecutar, misma que pondrá como bono la mitad del sueldo, siempre que sus salarios sean menores de 8,000.

```
MERGE INTO bonuses D
   USING (SELECT employee_id, salary, department_id FROM employees
   WHERE department_id = 80) S
   ON (D.employee_id = S.employee_id)
   WHEN MATCHED THEN UPDATE SET D.bonus = D.bonus + S.salary*.01
     DELETE WHERE (S.salary > 8000)
   WHEN NOT MATCHED THEN INSERT (D.employee_id, D.bonus)
     VALUES (S.employee_id, S.salary*.5 )
     WHERE (S.salary <= 8000);
```

Si deseas corroborar que los resultados son coherentes, podrías ayudarte de las siguientes sentencias:

1. registros eliminados

```
SELECT * FROM bonuses WHERE employee_id in (SELECT employee_id FROM
 employees WHERE department_id = 80 and salary> 8000); --registros
que fueron eliminados
```

2. registros que no cambiaron

```
SELECT * FROM bonuses WHERE employee_id in (SELECT employee_id FROM
 employees WHERE department_id != 80 and salary <= 8000); --
registros no modificados
```

3. registros que si cambiaron, algunos porque fueron insertados otros sólo cambió el valor.

4.
```
SELECT * FROM bonuses WHERE employee_id in (SELECT
employee_id FROM employees
   WHERE department_id = 80 and salary <= 8000) order by 1;
```