
KEEP
CALM
AND REMEMBER
YOUR
TRIGGERS!

Usando la misma base de datos de la actividad anterior:

Crea un trigger que inserta un registro en una tabla nueva llamada EMP_AUDIT cada vez que modificamos el salario de un empleado. Sólo se realizará la operación si el salario que se va a modificar difiere del nuevo.

La tabla EMP_AUDIT tendrá los siguientes campos:

- Identificador del empleado que se está actualizando.
- El momento en que se hace la actualización.
- Un mensaje que contenga el salario anterior y el nuevo.

Una vez realizada la actividad súbela a la plataforma para su evaluación.

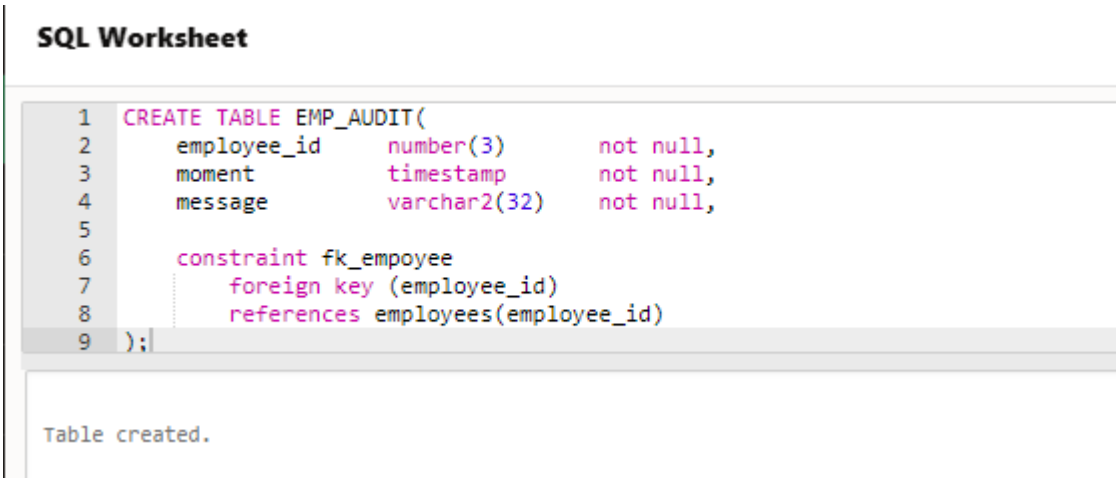
```
CREATE TABLE EMP_AUDIT(
    employee_id     number(3)       not null,     -- integer
    moment          timestamp       not null,     -- interval | date
    message         varchar2(32)    not null,     -- varchar3

    constraint fk_empoyee
        foreign key (employee_id)
        references employees(employee_id)
);
```

Primero creamos la tabla donde guardar todos los cambios que no sean neutros respecto a una futura auditoría sobre los salarios de los empleados de la compañía. He programado la condición NOT NULL para hacer un poco más complicado alterar un registro que en teoría es generado automáticamente por el disparador.

livesql.oracle.com da error cuando he declarado **employees.employee_id%type**
ORA-00911: invalid character

Si está creada la tabla dará el error:
ORA-00955: name is already used by an existing object



```
CREATE OR REPLACE TRIGGER EMP_SALARY_TRIGGER
    BEFORE
        UPDATE OF salary ON employees
         -- Cuando el evento que ha disparado el trigger es UPDATE:
                    -- :old ≠ null
                    -- :new ≠ null
    FOR EACH ROW
        BEGIN
            -- Evita registros que mencionan cambios neutros
            IF :old.salary != :new.salary THEN

                -- Registra el cambio
                INSERT INTO EMP_AUDIT
                    VALUES (
                        :old.employee_id,
                        systimestamp,
                        :old.salary || ' to ' || :new.salary
                    );

                 -- Mensage opcional
                DBMS_OUTPUT.PUT
                  ('New row inserted into EMP_AUDIT table');

            END IF;
END EMP_SALARY_TRIGGER;
```

```
1  CREATE OR REPLACE TRIGGER EMP_SALARY_TRIGGER
2      BEFORE
3          UPDATE OF salary ON employees
4          -- Cuando el evento que dispara el trigger es UPDATE
5              -- :old ≠ null
6              -- :new ≠ null
7
8      FOR EACH ROW
9          BEGIN
10             -- Evita registros que mencionan cambios neutros
11             IF :old.salary != :new.salary THEN
12
13                 -- Registra el cambio
14                 INSERT INTO EMP_AUDIT
15                     VALUES (
16                         :old.employee_id,
17                         systimestamp,
18                         :old.salary || ' to ' || :new.salary
19                     );
20
21                 -- Mensage opcional
22                 DBMS_OUTPUT.PUT
23                   ('New row inserted into EMP_AUDIT table');
24
25             END IF;
26
27 END EMP_SALARY_TRIGGER;
28 /
```

```
Trigger created.
```

```
BEGIN
    UPDATE employees SET salary = salary + 64;
    UPDATE employees SET salary = salary - 32;
    UPDATE employees SET salary = salary * 25;
    UPDATE employees SET salary = salary / 8;
END;
```

```
1  BEGIN
2      UPDATE employees SET salary = salary + 100;
3      UPDATE employees SET salary = salary - 50;
4      UPDATE employees SET salary = salary * 10;
5      UPDATE employees SET salary = salary / 100;
6  END;
```

Statement processed.

```
1  select * from emp_audit;
2
```

136	10-MAR-22 10.29.54.220058 PM	2200 to 2300
137	10-MAR-22 10.29.54.220098 PM	3600 to 3700
138	10-MAR-22 10.29.54.220136 PM	3200 to 3300
139	10-MAR-22 10.29.54.220172 PM	2700 to 2800
140	10-MAR-22 10.29.54.220211 PM	2500 to 2600
141	10-MAR-22 10.29.54.220249 PM	3500 to 3600
142	10-MAR-22 10.29.54.220303 PM	3100 to 3200
143	10-MAR-22 10.29.54.220373 PM	2600 to 2700
144	10-MAR-22 10.29.54.220415 PM	2500 to 2600
145	10-MAR-22 10.29.54.220452 PM	14000 to 14100
146	10-MAR-22 10.29.54.220490 PM	13500 to 13600
147	10-MAR-22 10.29.54.220527 PM	12000 to 12100
148	10-MAR-22 10.29.54.220563 PM	11000 to 11100
149	10-MAR-22 10.29.54.220600 PM	10500 to 10600

Download CSV

Rows 1 - 50. More rows exist.

Existen muchos más registros pero para verlos tendríamos que utilizar un cursor estático (si no esperamos que entren datos en la tabla emp_audit) con un for optimizado (...)

Gracias por tu tiempo.