



## Sesión 12

### CREATE TRIGGER II

Algunas observaciones sobre la sintaxis vista la semana pasada

### BEFORE – AFTER

HABITACIÓN(numero, categoria, superficie, piso)

Clave primaria: numero

Clave ajena: categoria → categoria

Las habitaciones tipo S únicamente están en la 5ª planta

número	categoría	superficie	piso
1	D	13	1
2	DT	14	1
3	DT	17	2

```
create or replace trigger tipo_suite
BEFORE insert or update of categoria, piso
on habitacion
for each row
when (new.categoria='S' and new.piso<>5)
begin
raise_application_error(-20601,'Las habitaciones tipo
                        SUITE sólo están en la quinta planta');
end;
```

**INSERT INTO habitacion values (1,'S',25,4);**

Error que empieza en la línea: 1 del comando :

INSERT INTO habitacion values (1,'S',25,4)

Informe de error -

Error SQL: ORA-20601: [Las habitaciones tipo SUITE sólo están en la quinta planta](#)

ORA-06512: en "DBD.TIPO\_SUITE", línea 2

ORA-04088: error durante la ejecución del disparador

```
create or replace trigger tipo_suite
AFTER insert or update of categoria, piso
on habitacion
for each row
when (new.categoria='S' and new.piso<>5)
begin
raise_application_error(-20601,'Las habitaciones tipo
                        SUITE sólo están en la quinta planta');
end;
```

**INSERT INTO habitacion values (1,'S',25,4);**

\*Cause: An UPDATE or INSERT [statement attempted to insert a duplicate key.](#)

For Trusted Oracle configured in DBMS MAC mode, you may see this message if a duplicate entry exists at a different level.

\*Action: Either remove the unique restriction or do not insert the key.

## TABLA MUTANTE:

Tabla que está siendo modificada por un INSERT, un UPDATE o un DELETE

*Los disparadores de fila presentan problemas a la hora de leer o modificar las filas de una tabla mutante.*

Supongamos que sólo se utiliza la tabla calendario para fechas del **año en curso** y que queremos controlar que un empleado no tenga demasiadas guardias. Para ello creamos un trigger.

```
create or replace trigger control_limpieza
after insert or update on calendario
for each row
declare
    cuantos number(2):=0;
    auxnombre empleado.nombre%type;
begin
    select count(*) into cuantos from calendario
    where emplimpieza=:new.emplimpieza and to_char(fecha,'MM')=to_char(:new.fecha,'MM');
    if (cuantos>5) then
        select nombre into auxnombre from empleado where nif=:new.emplimpieza;
        raise_application_error(-20601, auxnombre||' ya esta bastante cargado el mes de '||to_char(:new.fecha,'MM'));
    end if;
end;
```

Si probamos a hacer una modificación o una inserción sobre la tabla CALENDARIO, se muestra el siguiente error:

*Row 6: ORA-04091: la tabla CALENDARIO está mutando, puede que el disparador/la función no puedan verla*

*ORA-06512: en "dbd.control\_impieza"*

*ORA-04088: error durante la ejecución del disparador 'DBDcontrol\_impieza'*

*ORA-06512: en línea 1*

Vemos que no podemos, no siempre está permitido consultar la tabla que sufre la operación desde un disparador de fila.

Si fuese un disparador creado sólo para INSERT, definiéndolo para BEFORE en lugar de AFTER se solucionaría. Al ser también para UPDATE esto no ocurre. Existen situaciones cuando hay que trabajar con tablas mutantes que no se pueden resolver con un único disparador.

***En nuestros ejercicios no saldrán casos de disparadores con sentencias sobre tablas mutantes que no puedan ser resueltos de un modo tan sencillo como en el caso anterior, y que requieran de COMPOUND TRIGGER.***

## ORDEN de EJECUCIÓN de los disparadores definidos sobre una tabla:

Si hay varios disparadores para la misma sentencia, el orden de ejecución es:

1. BEFORE de sentencia
2. BEFORE de fila
3. AFTER de fila
4. AFTER de sentencia

## ACTIVAR/DESACTIVAR DISPARADORES

```
ALTER TRIGGER nombre_disparador ENABLE;  
ALTER TRIGGER nombre_disparador DISABLE;
```