

APARTADO A

En esta tarea tenéis que crear 3 disparadores que se activarán con las tablas TEMPLE, TDEPTO, TCENTR

1.- El primer disparador se llama alta_empleado que se tiene que activar después de insertar un empleado en temple. Cuando se produzca este evento el disparador debe insertar en la tabla ALTAS (que tendremos que haber creado previamente) una fila con el número del nuevo empleado, el nombre del nuevo empleado, su salario y la fecha de alta.

```
SQL> CREATE TABLE ALTAS (
  2 numero_nuevo_empleado integer,
  3 nombre_nuevo_empleado varchar2(20),
  4 salario integer,
  5 fecha_alta date
  6 );
Tabla creada.

SQL> CREATE OR REPLACE TRIGGER alta_empleado
  2 AFTER INSERT ON TEMPLE
  3
  4 FOR EACH ROW
  5 BEGIN
  6 INSERT INTO ALTAS (numero_nuevo_empleado, nombre_nuevo_empleado, salario, fecha_alta) VALUES (:new.numem, :new.nomem, :new.salar, CURRENT_DATE);
  7 END;
  8 /
Disparador creado.

SQL> INSERT INTO TEMPLE (NUMEM, NUMDE, EXTEL, FECHA, FECIN, SALAR, COMIS, NUMPI, NOPIEM) VALUES (999, 111, 211, '12/03/1988', '16/05/2024', 1800, 0, 3, 'RUIZ, PAUL');
1 fila creada.

SQL> SELECT * FROM ALTAS;
NUMERO NUEVO EMPLEADO  NOMBRE NUEVO EMPLEAD  SALARIO  FECHA_AL
-----
          999  RUIZ, PAUL              1800  16/05/24
SQL>
```

2.- El segundo disparador se llama cambio_presupuesto que se activa después de modificar el presupuesto de un departamento. Cuando se produzca este evento el disparador debe insertar en la tabla CAMBIOS (que tendremos que haber creado previamente) una fila con el número del departamento que cambia de presupuesto, el presupuesto antiguo, el presupuesto nuevo y la fecha de modificación del presupuesto.

```
SQL> SQL Plus (Oracle y PL/SQL)
SQL*Plus: Release 21.0.0.0.0 - Production on Tue May 16 16:52:23 2024
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle. All rights reserved.
Introduzca el nombre de usuario: system
Introduzca la contraseña:
Hora de última Conexión Correcta: Tue May 16 2024 14:14:39 +02:00
Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

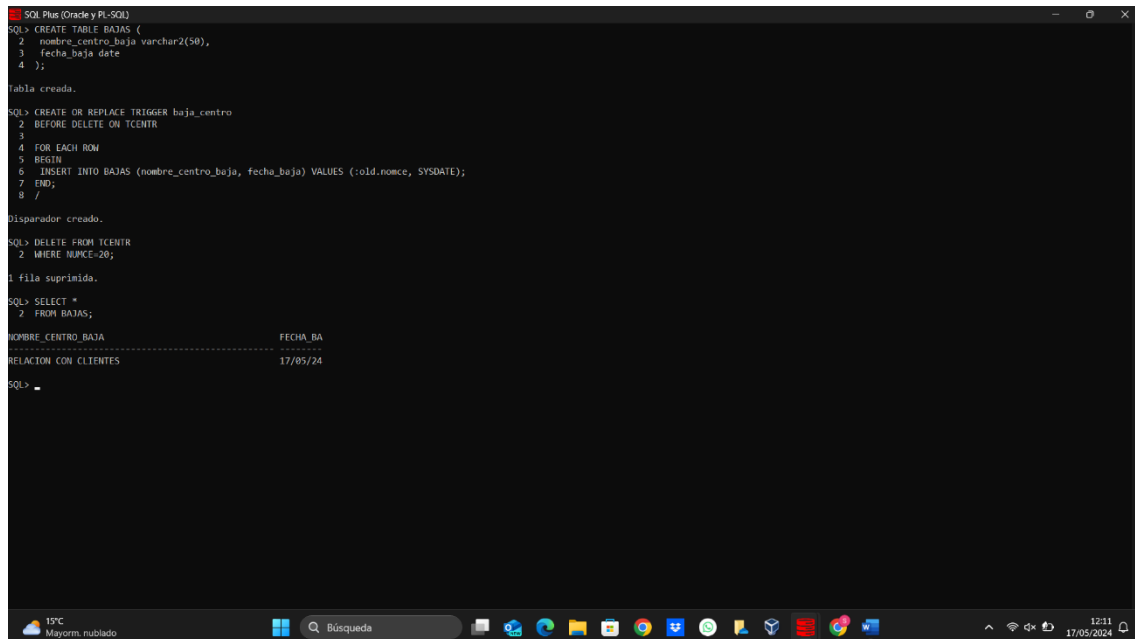
SQL> CREATE TABLE CAMBIOS (
  2 num_dep integer,
  3 presu_antigo integer,
  4 presu_nuevo integer,
  5 fecha_mod date
  6 );
Tabla creada.

SQL> CREATE OR REPLACE TRIGGER cambio_presupuesto
  2 AFTER UPDATE OF presu ON TDEPTO
  3
  4 FOR EACH ROW
  5 BEGIN
  6 INSERT INTO CAMBIOS (num_dep, presu_antigo, presu_nuevo, fecha_mod) VALUES (:new.numde, :old.presu, :new.presu, SYSDATE);
  7 END;
  8 /
Disparador creado.

SQL> UPDATE TDEPTO
  2 SET PRESU = 100
  3 WHERE NUMDE=121;
1 fila actualizada.

SQL> SELECT * FROM CAMBIOS;
NUM_DEP  PRESU_ANTIQUO  PRESU_NUEVO  FECHA_MOD
-----
       121         20         100  16/05/24
SQL>
```

3.-El tercer disparador se llama baja_centro que se activa antes de borrar un centro de trabajo. Cuando se produzca este evento el disparador debe insertar en la tabla BAJAS (que tendremos que haber creado previamente) una fila con el nombre del centro de trabajo que se dá de baja y la fecha de baja.



```
SQL> CREATE TABLE BAJAS (  
2 nombre_centro_baja varchar2(50),  
3 fecha_baja date  
4 );  
  
Tabla creada.  
  
SQL> CREATE OR REPLACE TRIGGER baja_centro  
2 BEFORE DELETE ON TCENTR  
3  
4 FOR EACH ROW  
5 BEGIN  
6 INSERT INTO BAJAS (nombre_centro_baja, fecha_baja) VALUES (:old.nombre, SYSDATE);  
7 END;  
8 /  
  
Disparador creado.  
  
SQL> DELETE FROM TCENTR  
2 WHERE NUMCE=20;  
  
1 fila suprimida.  
  
SQL> SELECT *  
2 FROM BAJAS;  
  
NOMBRE_CENTRO_BAJA          FECHA_BA  
-----  
RELACION CON CLIENTES      17/05/24  
  
SQL>
```

La tarea debe mostrar (mediante capturas) la creación de los disparadores y la prueba de cada uno, insertando, actualizando o borrando en la tabla correspondiente y comprobando después si el disparador funciona viendo las tablas ALTAS, CAMBIOS y BAJAS.

APARTADO B

Crear un disparador que se active después de insertar, borrar o actualizar una fila en TEMPLE, la actualización será del atributo salario. La acción del disparador será insertar una fila en la tabla ACTUALIZACIONES (se tendrá que crear) con la fecha de inserción, borrado o modificación, el nombre del empleado que se ha insertado, borrado o modificado y cuál ha sido la actualización (inserción, borrado o modificación). Si se ha modificado el salario también incluímos en esta tabla el salario antiguo y el nuevo.

```
SQL Plus (Oracle y PL-SQL)
SQL> CREATE TABLE ACTUALIZACIONES (
  1 Numero_empleado integer,
  2 Nombre_empleado varchar2(20),
  3 fecha date,
  4 Evento varchar2(20) CHECK (Evento IN ('Alta', 'Baja', 'Actualización')),
  5 Salar_V integer,
  6 Salar_N integer
  7 );
  8 ;

Tabla creada.

SQL> CREATE OR REPLACE TRIGGER actualizaciones
  1 AFTER INSERT OR DELETE OR UPDATE ON TEMPLE
  2
  3 FOR EACH ROW
  4 BEGIN
  5 IF UPDATING('salar') THEN
  6 INSERT INTO ACTUALIZACIONES (Numero_empleado, Nombre_empleado, fecha, Evento, Salar_V, Salar_N) VALUES (:new.numero, :new.nombre, SYSDATE, 'Actualización', :old.salar, :new.salar);
  7 END IF;
  8
  9 IF UPDATING THEN
  10 INSERT INTO ACTUALIZACIONES (Numero_empleado, Nombre_empleado, fecha, Evento) VALUES (:new.numero, :new.nombre, SYSDATE, 'Actualización');
  11 END IF;
  12
  13 IF DELETING THEN
  14 INSERT INTO ACTUALIZACIONES (Numero_empleado, Nombre_empleado, fecha, Evento) VALUES (:old.numero, :old.nombre, SYSDATE, 'Baja');
  15 END IF;
  16
  17 IF INSERTING THEN
  18 INSERT INTO ACTUALIZACIONES (Numero_empleado, Nombre_empleado, fecha, Evento) VALUES (:new.numero, :new.nombre, SYSDATE, 'Alta');
  19 END IF;
  20
  21 END;
  22 /
  23

Disparador creado.

SQL> INSERT INTO TEMPLE (NUMEM, NUMDE, EXTTEL, FECHA, FECCIN, SALAR, COMIS, NUMPHI, NOPEM) VALUES (125, 110, 545, '08/08/1999', '08/08/2019', 8000, 0, 0, 'RON, PABLO');
El empleado RON, PABLO ha sido dado de alta en el 16/05/24

1 fila creada.

SQL> UPDATE TEMPLE
  1 SET SALAR = 2000
  2 WHERE NUMEM = 125;
Nombre del empleado: RON, PABLO , Salario nuevo: 2000 , Salario viejo: 8000
Nombre del empleado: RON, PABLO , Salario nuevo: 2000 , Salario viejo: 8000
Los datos del empleado RON, PABLO han sido actualizados
```

```
SQL Plus (Oracle y PL-SQL)
8 END IF;
9
10 IF UPDATING THEN
11 INSERT INTO ACTUALIZACIONES (Numero_empleado, Nombre_empleado, fecha, Evento) VALUES (:new.numero, :new.nombre, SYSDATE, 'Actualización');
12 END IF;
13
14 IF DELETING THEN
15 INSERT INTO ACTUALIZACIONES (Numero_empleado, Nombre_empleado, fecha, Evento) VALUES (:old.numero, :old.nombre, SYSDATE, 'Baja');
16 END IF;
17
18 IF INSERTING THEN
19 INSERT INTO ACTUALIZACIONES (Numero_empleado, Nombre_empleado, fecha, Evento) VALUES (:new.numero, :new.nombre, SYSDATE, 'Alta');
20 END IF;
21
22 END;
23 /

Disparador creado.

SQL> INSERT INTO TEMPLE (NUMEM, NUMDE, EXTTEL, FECHA, FECCIN, SALAR, COMIS, NUMPHI, NOPEM) VALUES (125, 110, 545, '08/08/1999', '08/08/2019', 8000, 0, 0, 'RON, PABLO');
El empleado RON, PABLO ha sido dado de alta en el 16/05/24

1 fila creada.

SQL> UPDATE TEMPLE
  1 SET SALAR = 2000
  2 WHERE NUMEM = 125;
Nombre del empleado: RON, PABLO , Salario nuevo: 2000 , Salario viejo: 8000
Nombre del empleado: RON, PABLO , Salario nuevo: 2000 , Salario viejo: 8000
Los datos del empleado RON, PABLO han sido actualizados

1 fila actualizada.

SQL> DELETE FROM TEMPLE
  1 WHERE NOPEM = 'RON, PABLO';
El empleado RON, PABLO ha sido dado de baja en el 16/05/24

1 fila suprimida.

SQL> SELECT * FROM ACTUALIZACIONES;

NUMERO_EMPLEADO  NOMBRE_EMPLEADO  FECHA  EVENTO  SALAR_V  SALAR_N
-----
125 RON, PABLO    16/05/24 Alta
125 RON, PABLO    16/05/24 Actualización
125 RON, PABLO    16/05/24 Actualización
125 RON, PABLO    16/05/24 Baja
```