**PROYECTO FICHAJE**

**Triggers**

1. **Trigger\_Eventos.sql**

CREATE OR REPLACE TRIGGER TriggerEventos

BEFORE INSERT OR UPDATE OR DELETE

ON empleados

FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO control VALUES (user, sysdate,'inserción');

END IF;

IF UPDATING THEN

INSERT INTO control VALUES (user, sysdate,'actualización');

END IF;

IF DELETING THEN

INSERT INTO control VALUES (user, sysdate,'borrado');

END IF;

END TriggerEventos;

La funcionalidad de este trigger es que cuando se inserta, se modifica o se borra en la tabla empleados, inserta en la tabla control el usuario, el día y el parámetro que se ha producido.

1. **Trigger\_Fichaje.sql**

CREATE OR REPLACE TRIGGER TriggerFichaje

BEFORE INSERT

ON salida\_empleados

FOR EACH ROW

DECLARE

v\_total NUMBER:=0;

v\_total2 NUMBER:=0;

BEGIN

select distinct hora\_entrada into v\_total from entrada\_empleados

where id\_empleado=:new.id\_empleado

and fecha=:new.fecha;

v\_total2:=:new.hora\_salida - v\_total;

INSERT INTO fichaje VALUES (null,:new.id\_empleado,:new.fecha,v\_total2);

END TriggerFichaje;

Lo que hace es que cuando se inserta en la tabla salida\_empleados, restamos hora\_salida con hora\_entrada para que nos salga el total de horas que trabaja un empleado en un solo día e inserte esa información en la tabla fichaje. Para ello las insertamos en una variable y trabajamos con ellas.

1. **Trigger\_Notificacion.sql**

CREATE OR REPLACE TRIGGER TriggerNotificacion

BEFORE INSERT

ON semana

FOR EACH ROW

DECLARE

v\_horas\_extras NUMBER;

v\_falta\_horas NUMBER;

BEGIN

IF :new.horas\_semanales >35 THEN

v\_horas\_extras:=:new.horas\_semanales - 35;

insert into notificacion values (null,:new.id\_empleado,'Horas extras: '||v\_horas\_extras);

END IF;

IF :new.horas\_semanales <35 THEN

v\_falta\_horas:=35 - :new.horas\_semanales;

insert into notificacion values (null,:new.id\_empleado,'Horas restantes: '||v\_falta\_horas);

END IF;

END TriggerNotificacion;

Cuando se insertan datos en la tabla semana, éste se encarga de coger las horas semanales de cada empleado y si la jornada que es 35, se pasan o no llegan, se almacenará dependiendo de los dos casos. Si se pasa la jornada, mostrará las horas extras que ha hecho ese empleado, por el contrario, si no ha llegado a completar la jornada, éste mostrará las horas restantes.

**Procedimientos**

1. **Procedimiento\_Total\_Horas\_Fecha.sql**

CREATE OR REPLACE PROCEDURE Horas\_Semanales

AS

CURSOR Cursor\_Horas\_Semanales IS SELECT id\_empleado,sum(total\_horas) as Total from fichaje

group by id\_empleado;

BEGIN

FOR D1 in Cursor\_Horas\_Semanales LOOP

insert into semana values (D1.id\_empleado,sysdate,D1.Total);

DBMS\_OUTPUT.PUT\_LINE('El empleado ' || D1.id\_empleado || ' Total Horas: ' || D1.Total);

END LOOP;

END Horas\_Semanales;

Este procedimiento se encarga de sumar las horas que hace cada empleado en toda la semana e inserta esos datos en la tabla semana.

1. **Procedimiento\_Borrado.sql**

CREATE OR REPLACE PROCEDURE Borrado\_Fichaje

IS

BEGIN

execute immediate 'truncate table fichaje';

END;

Lo que hace es que borra todo el contenido de la tabla fichaje, sin borrar la misma tabla.

**Jobs**

1. **Job\_Horas\_Semanales.sql**

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => 'Job\_Horas\_Semanales',

job\_type => 'STORED\_PROCEDURE',

job\_action => 'Horas\_Semanales',

start\_date => SYSTIMESTAMP,

repeat\_interval => 'FREQ=WEEKLY; BYDAY=SAT',

comments => 'Todos los viernes');

END;

/

Este job se encarga de ejecutar el procedimiento “Horas\_Semanales” cada semana los sábados.

1. **Job\_Excel.sql**

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => 'Job\_Excel',

job\_type => 'STORED\_PROCEDURE',

job\_action => 'Creacion\_Excel',

start\_date => SYSTIMESTAMP,

repeat\_interval => 'FREQ=WEEKLY; BYDAY=SAT',

comments => 'Todos los sabados');

END;

/

Este job se encarga de ejecutar el programa “Creacion\_Excel” cada semana, los sábados.

1. **Job\_Borrado.sql**

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => 'Borrado\_Fichaje\_Job',

job\_type => 'STORED\_PROCEDURE',

job\_action => 'Borrado\_Fichaje',

start\_date => SYSTIMESTAMP,

repeat\_interval => 'FREQ=WEEKLY; BYDAY=SUN',

comments => 'Todos los domingos');

END;

/

Este job ejecuta el procedimiento “Borrado\_Fichaje” cada semana el domingo.

**Programas**

1. **Creacion\_Excel\_Horas.sql**

col fecha new\_value fecha

select to\_char(sysdate,'ddmmyy')fecha from dual;

spool c:../../../Horas\_mensuales/Horas\_Mensuales&fecha

select \* from fichaje;

spool off

Este programa crea un archivo Excel. Creamos la variable fecha para poder insertarla en el nombre del archivo y se escribirá la fecha actual. Seleccionamos la ruta de la creación y seleccionamos los datos a insertar en el archivo Excel.