**Desarrollo de aplicaciones multiplataforma**

**Acceso a datos**

**Ejercicios obligatorios-Tema 1**

1. Realizar un programa que solicite un número de empleado determinado, el programa debe comprobar si el salario es mayor que 1000, y si no es así actualizar el salario con una subida del 20% y, en una tabla temporal, insertar el nombre del empleado y el mensaje "Sueldo actualizado". Si ya superaba los 1000, se inserta en la tabla temporal el nombre del empleado y el mensaje "No necesita actualización".

**RESPUESTA:**

1. Hacer un programa que solicite por teclado un código de cliente e inserte en la tabla temporal el número total de pedidos de ese cliente (tot\_ped), el precio total de esos pedidos (precio\_tot), el código del cliente (cod\_cli) y el nombre del cliente (nombre\_cli).

**RESPUESTA:**

SET SERVEROUTPUT ON;

DROP TABLE tabla\_temporal;

CREATE TABLE tabla\_temporal (

cod\_cli NUMBER(4),

nombre\_cli VARCHAR2(50),

precio\_tot NUMBER(8),

tot\_ped NUMBER(8)

);

DECLARE

codigo\_cliente NUMBER := &id\_cliente; -- Declaración y asignación de la variable código\_cliente.

nombre customer.name%TYPE;

pedido\_total sales\_order.total%TYPE; -- Debe ser sales\_order.total, ya que se obtiene de la tabla sales\_order.

precio\_total item.total%TYPE;

BEGIN

FOR cliente IN (SELECT

c.customer\_id,

c.name,

so.total AS pedido\_total,

i.total AS precio\_total

FROM

customer c

JOIN sales\_order so ON c.customer\_id = so.customer\_id

JOIN item i ON so.order\_id = i.order\_id

WHERE

c.customer\_id = codigo\_cliente)

LOOP

nombre := cliente.name;

pedido\_total := cliente.pedido\_total;

precio\_total := cliente.precio\_total;

insert into tabla\_temporal values(codigo\_cliente,nombre,precio\_total,pedido\_total);

END LOOP;

END;

/

1. Crearse un registro t\_reg\_emple con la siguiente estructura: código del empleado, nombre y el job. Insertar en una tabla temporal el nombre del empleado y job para el empleado 7782.

**RESPUESTA:**

**SET SERVEROUTPUT ON;**

**DROP TABLE tabla\_temporal;**

**CREATE TABLE tabla\_temporal (**

**codigo\_empleado NUMBER(4),**

**nombre VARCHAR2(50),**

**trabajo VARCHAR2(50)**

**);**

**DECLARE**

**TYPE t\_reg\_emple IS RECORD (**

**codigo\_empleado NUMBER(4),**

**nombre employee.first\_name%TYPE,**

**trabajo VARCHAR2(50)**

**);**

**empleado t\_reg\_emple;**

**BEGIN**

**SELECT**

**employee\_id,**

**last\_name,**

**function**

**INTO empleado**

**FROM**

**employee**

**JOIN job ON job.job\_id = employee.job\_id**

**WHERE**

**employee\_id = 7782;**

**INSERT INTO tabla\_temporal VALUES (**

**empleado.codigo\_empleado,**

**empleado.nombre,**

**empleado.trabajo**

**);**

**dbms\_output.put\_line('Fila insertada en tabla\_temporal.');**

**-- Mostrar los valores insertados.**

**dbms\_output.put\_line('Código: '**

**|| empleado.codigo\_empleado**

**|| ', Nombre: '**

**|| empleado.nombre**

**|| ', Trabajo: '**

**|| empleado.trabajo);**

**END;**

**/**

1. Insertar 50 filas en una tabla. Recorre todas las filas de la tabla y las inserta en una tabla temporal. Crear la tabla students e insertar algunos datos.

**RESPUESTA:**

SET SERVEROUTPUT ON;

DROP TABLE students;

CREATE TABLE students (

codigo\_alumno NUMBER(4),

nombre VARCHAR2(50),

apellido VARCHAR2(50)

);

BEGIN

--crear 50 estudiantes llamado loli pop

FOR i IN 1..50 LOOP

INSERT INTO students (codigo\_alumno, nombre, apellido)

VALUES (i, 'Loli', 'Pop');

END LOOP;

END;

/

drop TABLE tabla\_temporal;

CREATE TABLE tabla\_temporal (

codigo\_alumno NUMBER(4),

nombre VARCHAR2(50),

apellido VARCHAR2(50)

);

BEGIN

--inserta en una tabla temporal

INSERT INTO tabla\_temporal (codigo\_alumno, nombre, apellido)

SELECT codigo\_alumno, nombre, apellido FROM students;

END;

/

1. Llenar una tabla PL con los nombres de los clientes y cargarla en una tabla temporal de sql que contenga el índice de la tabla PL y el nombre del cliente, y visualizar ésta.

**RESPUESTA:**

1. Llenar una tabla temporal con los códigos de clientes, nombre del cliente y código de empleado de tres formas posibles
   1. usando bucle LOOP,
   2. usando bucle WHILE y
   3. usando bucle FOR.

**RESPUESTA:**

1. A los pedidos del cliente 2, añadir 10 unidades más a la cantidad e insertar en una tabla temporal el código de artículo y un mensaje que diga "Diez unidades más vendidas".

**RESPUESTA:**

1. Crear un procedimiento que añada nuevos pedidos a un cliente determinado. El procedimiento recibe el idped, idart, cantidad y codcli.

**RESPUESTA:**

1. Procedimiento que borra los pedidos del cliente especificado. Recibe el codcli.

**RESPUESTA:**

1. Procedimiento para cambiar el oficio de un empleado. Se pasa al codemp y el nuevo oficio. Insertará en TEMP oficio\_ant, nuevo, codemp.

**RESPUESTA:**

1. Crear una función "Anual" para devolver el salario anual cuando se pasa el salario mensual y la comisión de un empleado. Asegurarse que controla nulos. Utilizar una variable de acoplamiento para ver lo que devuelve.

**RESPUESTA:**

1. Crear un paquete "Actualiza" que tiene tres procedimientos y una tabla PL:
   1. Un procedimiento de alta de un pedido.
   2. Un procedimiento de baja de los pedidos y detalles de un cliente concreto.
   3. Una tabla PL que almacenará los códigos de los pedidos.
   4. Un procedimiento "Listar" que tiene como parámetro 'in' el código de cliente y como parámetro de salida una tabla PL con los códigos de los pedidos del cliente especificado, además se grabará en una tabla temporal el código del pedido y el código del artículo.

**RESPUESTA:**

1. Crear un trigger para ver como suceden los eventos de activación. Vamos a utilizar la tabla plantilla y vamos a cambiar a las "Enfermeras" por "Interinos".

**RESPUESTA:**

1. Crear un disparador a nivel de fila tal que después de insertar, modificar o borrar un detalle de la tabla detalles introduzca en la tabla temporal el usuario , la fecha del sistema el código de pedido que se ha modificado, insertado o borrado, así como un mensaje que diga "Detalle dado de alta", "Detalle borrado", "Detalle modificado" según corresponda.

**RESPUESTA:**

1. Disparador que inserta una fila en la tabla Temp con el texto ‘subida de salario empleado’ y el número del empleado al que se le ha subido el salario. El disparador se activará después de actualizar el salario de la tabla empleador.

**RESPUESTA:**

1. Trigger que se dispara cada vez que se borra un empleado, guardando el número empleado, apellido y departamento en la tabla TEMP.

**RESPUESTA:**

1. Limitar a 5 (0 al 4) el número de detalles de cada pedido.

**RESPUESTA:**

1. Solicitar un código de cliente por teclado; si existe se inserta su código en una tabla temporal y el mensaje 'EXISTE', y si no existe se dispara la excepción NO\_DATA\_FOUND y se inserta en la tabla temporal el mensaje 'NO EXISTE'.

**RESPUESTA:**

1. Se solicita un código de artículo por teclado y se inserta en una tabla temporal el precio y la descripción de este artículo si está pedido, es decir, si su código está en la tabla detalle. Si el artículo no está pedido, se genera una excepción de usuario con el mensaje "El artículo (código) no lo ha pedido ningún cliente"

**RESPUESTA:**

1. Modificar el ejercicio 2 añadiendo excepciones, de tal forma que en el gestor de excepciones controlemos que select ha fallado, insertando en una tabla temporal cliente no existe, o bien articulo no existe, según haya fallado uno u otro.

**RESPUESTA:**

1. Crear un trigger que se ejecutará automáticamente cuando se elimine algún empleado en la tabla correspondiente visualizando el número y el nombre de los empleados borrados.

**RESPUESTA:**

1. Crear un procedimiento que reciba como parámetro un código de empleado, y modificar el salario de un empleado en función del número de empleados que tiene a su cargo:
   1. si no tiene ningún empleado a su cargo subirle 50 euros.
   2. si tiene 1 empleado a su cargo subirle 80 euros.
   3. si tiene 2 empleados a su cargo subirle 100 euros.
   4. si tiene más de tres empleados a su cargo subirle 110 euros.
   5. si es el PRESIDENTE su salario se incrementa en 30 euros.

**RESPUESTA:**

1. Hacer dos bloques PL/SQL uno de los cuales escribe ‘Hola mundo’ al derecho y otro del revés.

**RESPUESTA:**

1. Hacer un bloque o proceso que muestre por pantalla el nombre y el jefe de cada empleado, con WHILE.

**RESPUESTA:**

1. Visualizar los apellidos de los empleados pertenecientes al departamento pasado por parámetro, numerándolos secuencialmente.

**RESPUESTA:**

1. Escribir un procedimiento que reciba todos los datos de un nuevo empleado y procese la transacción de alta, gestionando los siguientes errores:
   1. no\_existe\_departamento
   2. no\_existe\_director
   3. numero\_empleado\_duplicado
   4. salario nulo con RAISE\_APPLICATION\_ERROR
   5. Otros posibles errores de Oracle visualizando código y mensaje de error.

**RESPUESTA:**

1. Escribe un procedimiento (recibe como parámetro el oficio) el procedimiento subirá el sueldo de todos los empleados que ganen menos del salario medio de su oficio. La subida será del 50% de la diferencia entre el salario de los empleados y la media de su oficio. Se deberá gestionar los posibles errores.

**RESPUESTA:**

1. Escribir un programa que incremente el salario de los empleados de un determinado departamento que se pasará como primer parámetro. El incremento será una cantidad en euros que se pasará como segundo parámetro en la llamada. El programa deberá informar del número de filas afectadas por la actualización. Los salarios se han de actualizar individualmente.

**RESPUESTA:**

1. Crear un programa que reciba un número de empleado y una cantidad que se incrementará al salario del empleado correspondiente. Utilizar una excepción definida por el usuario denominada salario\_nulo y la predefinida when no\_data\_found

**RESPUESTA:**