

PREGUNTAS	1	2	3	4	5	6	7	8			NOTA
PUNTOS	0,5	0,5	1	2	1	1	2	2			
PUNTUACIÓN											

## Tarea 6

### Base de datos Talleres Faber

1.- Crear un procedimiento que muestre los vehículos (marca, modelo y color) que no estén reparados y los datos de los clientes y vehículos que han entrado a reparar hoy. (En nuestro caso ninguno).

NOTA: Son 2 consultas distintas en el mismo procedimiento.

```
create or replace procedure vehiculos_no_reparados
as
    cursor cveh is
        select marca, modelo, color
        from vehiculos v join reparaciones r on (v.matricula=r.matricula)
        where fechasalida is null;
    fila_rec cveh%rowtype;

begin
    open cveh;
    dbms_output.put_line(rpad('MARCA',20) || rpad('MODELO',20) || rpad('COLOR',10) );
    dbms_output.put_line(rpad('-----',20) || rpad('-----',20) || rpad('-----',20) );
    loop
        fetch cveh into fila_rec;
        exit when cveh%notfound;
        dbms_output.put_line(rpad(fila_rec.marca,20) || rpad(fila_rec.modelo,20) ||
rpad(fila_rec.color,20));
    end loop;
    close cveh;

end vehiculos_no_reparados;
```

2.- Crear una función que actualice el estado de las reparaciones (poner estado reparado a 1) que estén finalizadas en una fecha que se indique y que devuelva cuantas reparaciones han finalizado en esa fecha.

```
Create Or Replace Function Actualizar_Reparaciones (
    Fecha In Date
) Return Int As

    Cursor Cact Is
    Select
        Idreparacion,
        Fechasalida,
        Reparado
    From
        Reparaciones
    Where
        Fechasalida = Fecha
        And Reparado = 0;

    Fila_Rec Cact%Rowtype;
    Num      Int(5);
Begin
    Num := 0;
    Open Cact;
    Loop
        Fetch Cact Into Fila_Rec;
        Exit When Cact%Notfound;
        Update Reparaciones
        Set
            Reparado = 1, Observaciones = 'Sin observaciones';
        Where
            Idreparacion = Fila_Rec.Idreparacion;

        Num := Num + 1;
    End Loop;

    Close Cact;
    Return Num;
End Actualizar_Reparaciones;
```

-comprobación

```
set serveroutput on
DECLARE
    num NUMBER;
BEGIN
    num := actualizar_reparaciones('08/01/2011');
    dbms_output.put_line('Num. Reparaciones-> '||num);
END;
```

3.-Un listado con dos columnas: en la primera, en mayúsculas apellidos y nombre de todos los clientes (entre los apellidos y el nombre incluir una coma como separador) y en la segunda, la ciudad en la que cada cliente tiene su domicilio (únicamente la ciudad, no la dirección).

4.- Creación de funciones:

a) Diseña una función que calcule el importe de los recambios sustituidos en una reparación.

```
create or replace function importe_recambios (  
    id in decimal  
) return int as  
    total_recambios decimal(5,2) default 0;  
begin  
  
    return total_recambios;  
End Importe_Recambios;
```

b) Crear una función que devuelva el importe de las actuaciones que se llevan a cabo en una reparación (para calcular el importe multiplica las horas por el importe de cada actuación).

En ambas funciones pasar como variable de entrada el Id de la reparación.

c) Hacer una consulta que calcule el importe total (mano de obra y recambios) de las reparaciones que se le hayan realizado al vehículo de matrícula '1313 DEF' pasada como parámetro.

```
select idreparacion, importe_recambios(idreparacion)+importe_mano_obra(idreparacion)  
from reparaciones  
where matricula='1313 DEF';
```

## Base de datos Historial Laboral

5.- Crear un procedimiento que realice un listado con los nombres y fechas, de todos los empleados y departamentos por los que ha pasado ordenado por fecha. Realizar la comprobación.

6.- Crear un procedimiento que actualice los estudios de un empleado (pasando como parámetros el dni del empleado, nombre de la universidad, año, grado y especialidad). En caso de error enviar un mensaje "raise\_application\_error".

7.- Tabla de cambios datos empleados y trigger:

- a) Crear una nueva tabla de nombre CambiosEmpleados en la que se almacenarán los datos antiguos de un empleado cada vez que se hagan cambios. Esta tabla debe guardar también la fecha de modificación.
- b) Crear un trigger que se dispare cada vez que se haga una actualización de los datos de un empleado (a excepción del salario), se deben copiar los datos antiguos a la tabla CambiosEmpleados.
- c) Realiza la comprobación.

#### 8.- Trigger y procedimiento:

- a) Crear un procedimiento para cambiar a un empleado de departamento (pasados como parámetros el DNI del empleado, código del puesto de trabajo, fecha de inicio, el nombre del departamento y el dni del supervisor).
- b) Crear un trigger para que cuando un empleado cambie de puesto actualice automáticamente su historial laboral (fecha de finalización en su anterior departamento). Realizar la comprobación con el procedimiento anterior.