Universidad Nacional Autónoma de Honduras

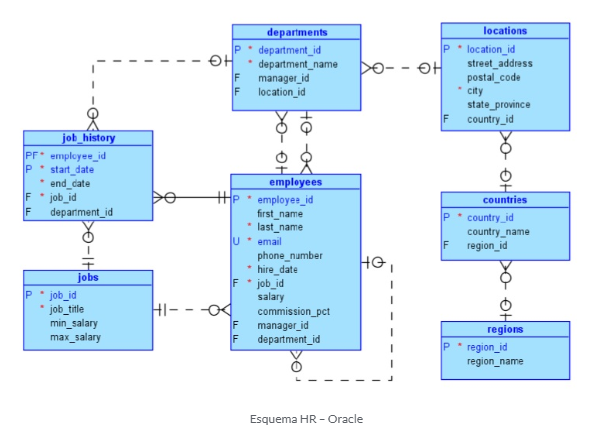
Facultad de Ciencias Económicas, Administrativas y Contables

Departamento de Informática Administrativa

Base de Datos II.

Estimados estudiantes,

Por este medio les remito la práctica a realizar.



1. Crear una función que reciba como parámetro de entrada el nombre del departamento (Ejemplo “Sales”) y calcule el salario promedio del departamento. La función deberá devolver un mensaje como el siguiente: “El departamento Sales tiene un salario promedio de 8955”

UTILIZAR **ROUND** Las funciones numéricas estándar más comunes del **PLSQL** y el SQL de **Oracle** son: **ROUND** (m, n): la función **ROUND** acepta como entrada un número y devuelve como salida otro número redondeado a un número específico de decimales (que se indica en el segundo parámetro de la función)

CREATE OR REPLACE FUNCTION DEPARTAMENT\_AVG(P\_DEPARTMENT\_NAME HR.DEPARTMENTS.DEPARTMENT\_NAME%TYPE) RETURN VARCHAR IS

--ESTA VARIABLES ES CON LA QUE VOY A SACAR MIS CALCULOS

V\_AVG NUMBER;

-- VARIBLE CON LA QUE RECIBO EL MENSAJE

V\_MESS VARCHAR(255);

BEGIN

--

SELECT ROUND(AVG(SALARY),2)

INTO V\_AVG

FROM HR.EMPLOYEES E, HR.DEPARTMENTS D

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

AND D.DEPARTMENT\_NAME = P\_DEPARTMENT\_NAME;

V\_MESS := ('El departamento' || P\_DEPARTMENT\_NAME ||' tiene un salario promedio de: ' || to\_char(v\_avg));

return v\_mess;

end;

/

1. Crear una función que recibirá como parámetro el Código del empleado y retornara un valor Boolean deberá comprobar, la existencia de historia de un empleado en la tabla Job\_History; la función devolverá TRUE si encuentra registros en la tabla y FALSE si no encuentra registros.

/\*

2. Crear una función que recibirá como parámetro el Código del empleado y retornara un valor Boolean deberá comprobar,

la existencia de historia de un empleado en la tabla Job\_History; la función devolverá TRUE si encuentra registros en la

tabla y FALSE si no encuentra registros.

/\*

2. Crear una función que recibirá como parámetro el Código del empleado y retornara un valor Boolean deberá comprobar,

la existencia de historia de un empleado en la tabla Job\_History; la función devolverá TRUE si encuentra registros en la

tabla y FALSE si no encuentra registros.

\*/

CREATE OR REPLACE FUNCTION empleado\_tiene\_historial(p\_employee\_id hr.employees.employee\_id%type) return boolean is

v\_contador number; --variable de contador para contar los numeros de registros

begin

--aqui le digo que cuento todoo lo que tenga

/\*Seleccion y cuenta ingresa en la variable v\_contador desde la tabla hr,job\_history con valor denominado j y cuando en la tabla de hr.job history haiga un valor que sea igual al parametro que me paso el usuario\*/

select count(\*)

into v\_contador

from hr.job\_history j

where j.employee\_id = p\_employee\_id;

if v\_contador = 0 then

return false; -- si en el contador no existen registros va retornar un false

else

return true; -- cualquier otro valor diferente de 0 vas a retornar un true

end if;

end;

/

/\*EJECUTAR LA FUNCION\*/

declare

v\_resultado boolean;

begin

v\_resultado := empleado\_tiene\_historial(101);

if v\_resultado then

DBMS\_OUTPUT.PUT\_LINE('El Empleado tiene historial');

else

DBMS\_OUTPUT.PUT\_LINE('El Empleado no tiene historial');

end if;

end;

/

1. Crear una función que reciba como parámetro el Employee\_id, y retorne la cantidad de subordinados o personal a cargo que este tiene (El campo manager\_id indica el jefe de cada empleado)
2. Crear un procedimiento para insertar registros en la tabla Employees, recibirá como parámetros todos los campos existentes, deberá controlar por medio de excepción si el Employee\_id ya existe en la tabla de empleados y desplegar mensaje indicándolo.

1. Crear un procedimiento para eliminar registro de la tabla de Employees, recibirá como parámetro el Employee\_id, y controlara con excepción si el Employee\_id no existe, enviara mensaje indicándolo.
2. Crear un procedimiento para actualizar todos los valores de la tabla Employees, recibirá como parámetro todas las columnas de la tabla, y actualizará las columnas con los nuevos valores que se envíen. Controlar por medio de Excepción si el registro no existe, y desplegar un mensaje indicándolo.
3. Crear una Función que recibirá como parámetro el Employee\_id y retornara el nombre completo del empleado, deberá controlar por medio de excepción si el Employee\_id no existe en la tabla de empleados y desplegar mensaje indicándolo.
4. Crear una función que deberá de actualizar los salarios de todos los empleados de un departamento ‘X’ por un porcentaje de valor ingresado. Retornará verdadero si la actualización fue exitosa y falso si se generó algún error durante la misma.
5. Crear un procedimiento que recibirá como parámetros el número de Departamento, y retornara como parámetro, el salario más alto de ese departamento, el salario más bajo del departamento, el promedio de Salario y el total de empleados del departamento.