

Garantía de Excelencia

PRACTICA3 - Funciones & Triggers

Nombre: Marco Antonio Roca Montenegro.

Institución: Universidad Privada Boliviana.

Materia: Bases de datos Avanzadas.

Docente: Msc. Carlos Aníbarro.

Fecha: 12/06/2023.

Contenido

INTRODUCCIÓN	. 3
Función "clientesSucursal"	. 3
Función "eval_salario"	. 4
Función "miFactorial"	. 5
Trigger "datosNotas"	. 6

INTRODUCCIÓN

El presente documento contiene las pruebas realizadas con respecto a los procedimientos y funciones implementados dentro de la base de datos ROCA_MARCO_Practica3, realizado en MariaDB – Versión: 15.1.

FUNCIÓN "CLIENTESSUCURSAL"

La función "clientesSucursal" tiene como objetivo el mostrar la cantidad de clientes por sucursal de la tabla "sakila.store" (únicamente 2 tiendas). La base de datos "sakila" cuenta con una tabla "customer", por lo cual para saber la cantidad la función hace uso de un "count" tomando en cuenta el id de la tienda como parámetro de igualdad. En conclusión, nos retorna una respuesta con tipo de dato *INT*.

Para su uso: select s.store_id, concat(f.first_name, '', f.last_name) as Manager,
 clientesSucursal(s.store_id) as CANTIDAD from sakila.store s, sakila.staff f where
 s.manager_staff_id=f.staff_id;

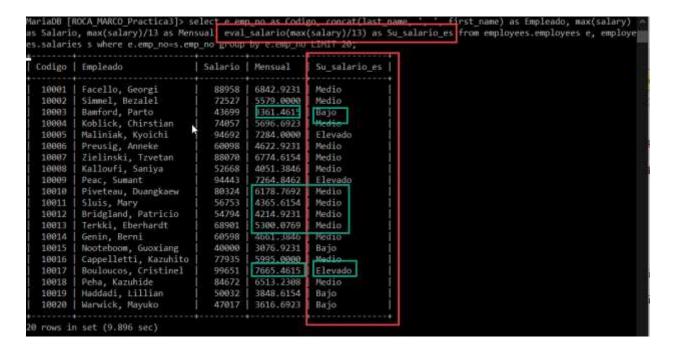
FUNCIÓN "EVAL SALARIO"

Recibe como parámetro un dato de tipo *float*, el cual es una cantidad la cual en la práctica denomina la evaluación de un salario (evaluamos el salario mensual). Lo que nos debe retornar es un mensaje el cual responde a:

Si el salario es < 4000 ? 'Bajo' : salario >=4000 AND salairo <=7000 ? 'Medio' : 'Elevado'.

Adicionalmente se establece un "**LIMIT 20**", para controlar que se muestre en consola únicamente 20 resultados.

Para su uso: select e.emp_no as Codigo, concat(last_name, ', ', first_name) as
 Empleado, max(salary) as Salario, max(salary)/13 as Mensual,
 eval_salario(max(salary)/13) as Su_salario_es from employees.employees e,
 employees.salaries s where e.emp_no=s.emp_no group by e.emp_no LIMIT 20;



FUNCIÓN "MIFACTORIAL"

Calcula el factorial del parámetro el cual propiciemos (tipo de entrada y salida *INT*). Internamente se maneja una variable de **iteración** = al parámetro de entrada y una variable denominada **resultado** (definida con valor 1). Luego se realiza un bucle *WHILE*, multiplica resultado = 1 * el valor de iteración, luego restamos iteración - 1 siempre y cuando iteración > 1.

Con esto, una vez iteración sea igual a 1 se rompe el ciclo WHILE (Considerando que el primer paso es multiplicar por un valor 1).

- **Para su uso:** select miFactorial(7) as Resultado;

TRIGGER "DATOSNOTAS"

El siguiente trigger realiza un control de inserción de notas. Entonces el disparador se ejecuta **DESPUES** de haber hecho un insert. Si la nota es menor a 0 y mayor a 100, El disparador (*AFTER INSERT*) realiza un *DELETE* de los datos ingresados.

Datos de prueba:

```
insert into notas values(5,'roca','piter','biologia',90);
insert into notas values(6,'roca','marco','fisica',40);
insert into notas values(7,'roca','marco','cultura',60);
insert into notas values(8,'roca','marco','deporte',10);
insert into notas values(9,'roca','marco','canto',70);
insert into notas values(10,'roca', 'marco', 'musica', 90);
insert into notas values(11,'roca','marco','fisica',-120);
insert into notas values(12,'roca','marco','cultura',-20);
insert into notas values(13,'roca','marco','deporte',0);
insert into notas values(14,'roca','marco','canto',120);
insert into notas values(15,'roca','marco','musica',190);
```

```
Number | Number | Practice3|> | Desert into motes embes(9, "roca", "marco", "casto", 78);

2007 (M., 1 rea effected (0.00) sec)

Nursian | Number | Practicas| | Desert into motes values(10, "roca", "marco", "masica", 90);

Nursian | Number | Practicas| | Desert into motes values(10, "roca", "marco", "fisica", -120);

Nursian | Number | Number | Practicas| | Desert into motes values(13, "roca", "marco", "fisica", -120);

Nursian | Number | Nursian | Desert into motes values(13, "roca", "marco", "fisica", -120);

Nursian | Nursian | Nursian | Desert into motes values(13, "roca", "marco", "casto", "fisica", -120);

Nursian | Nursian | Nursian | Desert into motes values(12, "roca", "marco", "casto", "fisica", -120);

Nursian | Nursian | Nursian | Desert into motes values(12, "roca", "marco", "casto", "marco", "marco", "marco", "marco", "casto", "marco", "marc
```