



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;

```
MariaDB [ROCA_MARCO_Practica3]> 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;
```

store_id	Manager	CANTIDAD
1	Mike Hillyer	326
2	Jon Stephens	273

2 rows in set (0.001 sec)

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 salario <=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;

```
MariaDB [ROCA_MARCO_Practica3]> 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;
```

Codigo	Empleado	Salario	Mensual	Su_salario_es
10001	Facello, Georgi	88958	6842.9231	Medio
10002	Simmel, Bezalel	72527	5579.0000	Medio
10003	Bamford, Parto	43699	3361.4615	Bajo
10004	Koblick, Chirstian	74057	5696.6923	Medio
10005	Maliniak, Kyoichi	94692	7284.0000	Elevado
10006	Preusig, Anneke	60098	4622.9231	Medio
10007	Zielinski, Tzvetan	88070	6774.6154	Medio
10008	Kalloufi, Saniya	52668	4051.3846	Medio
10009	Peac, Sumant	94443	7264.8462	Elevado
10010	Piveteau, Duangkiew	80324	6178.7692	Medio
10011	Sluis, Mary	56753	4365.6154	Medio
10012	Bridgland, Patricio	54794	4214.9231	Medio
10013	Terkki, Eberhardt	68901	5300.0769	Medio
10014	Genin, Berni	60598	4661.3846	Medio
10015	Nooteboom, Guoxiang	40000	3076.9231	Bajo
10016	Cappelletti, Kazuhito	77935	5995.0000	Medio
10017	Bouloucos, Cristinel	99651	7665.4615	Elevado
10018	Peha, Kazuhide	84672	6513.2308	Medio
10019	Haddadi, Lillian	50032	3848.6154	Bajo
10020	Warwick, Mayuko	47017	3616.6923	Bajo

20 rows in set (9.896 sec)

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;

```
MariaDB [ROCA_MARCO_Practica3]> select miFactorial(7) as Resultado;
+-----+
| Resultado |
+-----+
|      5040 |
+-----+
1 row in set (0.001 sec)

MariaDB [ROCA_MARCO_Practica3]> select miFactorial(5) as Resultado;
+-----+
| Resultado |
+-----+
|       120 |
+-----+
1 row in set (0.000 sec)
```

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);
```

```

MariaDB [ROCA_MARCO_Practica3]> insert into notas values(9,'roca','marco','canto',70);
Query OK, 1 row affected (0.001 sec)

MariaDB [ROCA_MARCO_Practica3]> insert into notas values(10,'roca','marco','musica',90);
Query OK, 1 row affected (0.001 sec)

MariaDB [ROCA_MARCO_Practica3]>
MariaDB [ROCA_MARCO_Practica3]> insert into notas values(11,'roca','marco','fisica',-120);
ERROR 1442 (HY000): Can't update table 'notas' in stored function/trigger because it is already used by statement which invoked this stored function/trigger
MariaDB [ROCA_MARCO_Practica3]> insert into notas values(12,'roca','marco','cultura',-20);
ERROR 1442 (HY000): Can't update table 'notas' in stored function/trigger because it is already used by statement which invoked this stored function/trigger
MariaDB [ROCA_MARCO_Practica3]> insert into notas values(13,'roca','marco','deporte',0);
Query OK, 1 row affected (0.001 sec)

MariaDB [ROCA_MARCO_Practica3]> insert into notas values(14,'roca','marco','canto',120);
ERROR 1442 (HY000): Can't update table 'notas' in stored function/trigger because it is already used by statement which invoked this stored function/trigger
MariaDB [ROCA_MARCO_Practica3]> insert into notas values(15,'roca','marco','musica',190);
ERROR 1442 (HY000): Can't update table 'notas' in stored function/trigger because it is already used by statement which invoked this stored function/trigger
MariaDB [ROCA_MARCO_Practica3]> select * from Notas;
+----+-----+-----+-----+-----+
| ID_NOTA | apellido | nombre | materia | nota |
+----+-----+-----+-----+-----+
| 1 | roca | marco | psicla | 90 |
| 2 | roca | luis | mate | 50 |
| 3 | roca | vito | cultura | 70 |
| 4 | roca | juana | aventura | 100 |
| 5 | roca | piter | biologia | 90 |
| 6 | roca | marco | fisica | 40 |
| 7 | roca | marco | cultura | 60 |
| 8 | roca | marco | deporte | 10 |
| 9 | roca | marco | canto | 70 |
| 10 | roca | marco | musica | 90 |
| 11 | roca | marco | deporte | 0 |
| 12 | roca | piter | biologia | 90 |
| 13 | roca | marco | fisica | 40 |
| 14 | roca | marco | cultura | 60 |
| 15 | roca | marco | deporte | 10 |
| 16 | roca | marco | canto | 70 |
| 17 | roca | marco | musica | 90 |
| 18 | roca | marco | deporte | 0 |
+----+-----+-----+-----+-----+
18 rows in set (0.000 sec)

MariaDB [ROCA_MARCO_Practica3]>

```