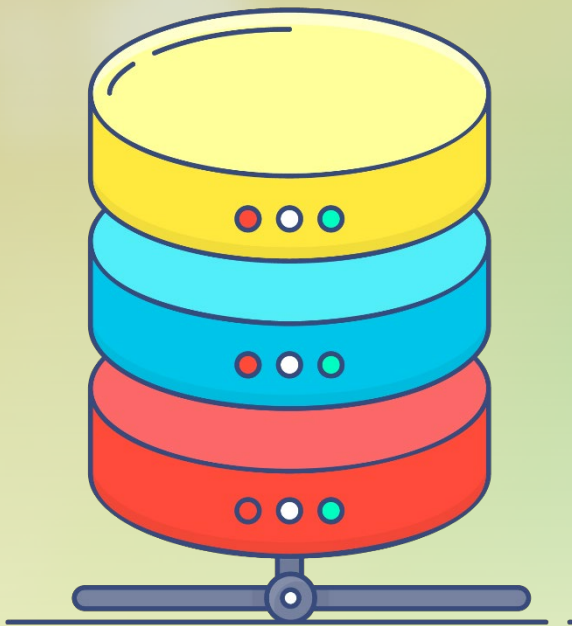


FUNCIONES VIII: MYSQL



**GESTIÓN DE BASES DE DATOS
JUAN CARLOS NAVIDAD GARCÍA**

1. Obtener los datos del propietario de una moto determinada. Comprobar que la moto está en la base de datos.

```
create function propietario(xmatricula varchar(8)) returns varchar(60) DETERMINISTIC
BEGIN
declare datos varchar(60);
declare cont int;
select count(*) into cont from motos where matricula=xmatricula;
if cont>0 then
select concat(nombre," ", tfno," ",fnac," ",fcarnet," ", tipocarnet) into datos from
clientes,motos where clientes.idcli=motos.idcliente and matricula=xmatricula;
ELSE
set datos=concat("La matrícula: ",xmatricula," no existe");
END IF;
return datos;
end //
```

```
mysql> select propietario("1137KNK") //
+-----+
| propietario("1137KNK") |
+-----+
| Samuel Ruiz 603665183 1997-07-04 2019-08-29 A |
+-----+
1 row in set (0.02 sec)
```

```
mysql> select propietario("1137KNU") //
+-----+
| propietario("1137KNU") |
+-----+
| La matrícula: 1137KNU, no existe |
+-----+
1 row in set (0.00 sec)
```

2. Obtener el precio del seguro de una moto determinada. Comprobar que la moto existe.

```
create function precio_seguro(xmatricula varchar(8)) returns varchar(50)
DETERMINISTIC
BEGIN
declare datos varchar(50);
declare cont int;
select count(*) into cont from motos where matricula=xmatricula;
if cont>0 then
select concat("El precio del seguro es: ",precio," Euros") into datos from
motos,polizas,tipopoliza where motos.matricula=polizas.matricula and
polizas.tipo=tipopoliza.idtipo and motos.matricula=xmatricula;
ELSE
set datos=concat("La matrícula: ",xmatricula," no existe");
END IF;
return datos;
end //
```

```
mysql> select precio_seguro("1137KNK") //
+-----+
| precio_seguro("1137KNK") |
+-----+
| El precio del seguro es: 400 Euros |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select precio_seguro("1137KNU") //
+-----+
| precio_seguro("1137KNU") |
+-----+
| La matrícula: 1137KNU, no existe |
+-----+
1 row in set (0.00 sec)
```

3. Crear una función para ver el total que ha facturado un agente determinado.

```
create function total_agente(xidagente int) returns varchar(50) DETERMINISTIC
BEGIN
declare datos varchar(50);
declare cuenta decimal;
select count(idagente)*precio into cuenta from polizas,tipopoliza where tipo=idtipo
and idagente=xidagente;
select concat("El agente ha facturado: ",cuenta," euros") into datos;
return datos;
end //
```

```
mysql> select total_agente(1) //
+-----+
| total_agente(1) |
+-----+
| El agente ha facturado: 800 euros |
+-----+
1 row in set (0.00 sec)
```

4. Crear una función para insertar una moto nueva. Utilizar la función y asignarle esa moto al cliente 2.

```
create function insertar_moto(xmatricula varchar(8), xmarca varchar(15), xmodelo
varchar(20), xpotencia int, xanio int) returns varchar(50) DETERMINISTIC
BEGIN
declare datos varchar(50);
insert into motos(matricula,marca,modelo,potencia,anio,idcliente) values(xmatricula,
xmarca, xmodelo, xpotencia, xanio, 2);
set datos=concat("La moto: ",xmatricula," ha sido añadida");
return datos;
end //
```

```
mysql> select insertar_moto("2187MFP","Honda","Hornet",92,2023) //
+-----+
| insertar_moto("2187MFP","Honda","Hornet",92,2023) |
+-----+
| La moto: 2187MFP ha sido añadida |
+-----+
1 row in set (0.00 sec)

mysql> select * from motos where marca="Honda" and modelo="Hornet" //
+-----+-----+-----+-----+-----+-----+
| matricula | marca | modelo | potencia | anio | idcliente |
+-----+-----+-----+-----+-----+-----+
| 2187MFP | Honda | Hornet | 92 | 2023 | 2 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

5. Crear una función para obtener la edad con que obtuvo su carnet un cliente en concreto.

```
create function edad_carnet(xidcli int) returns int DETERMINISTIC
BEGIN
declare datos int;
select TIMESTAMPDIF(year,fnac,fcarnet) into datos from clientes where
idcli=xidcli;
return datos;
end //
```

```
mysql> select edad_carnet(1) //
+-----+
| edad_carnet(1) |
+-----+
|                21 |
+-----+
1 row in set (0.00 sec)
```

6. Crear una función para añadir una póliza a la moto añadida en el punto 4. Utilízala con datos de tu elección.

```
create function crea_poliza(xmatricula varchar(8), xidagente int, xfinicio date,
xffin date, xtipo varchar(4)) returns varchar(50) DETERMINISTIC
BEGIN
declare datos varchar(50);
insert into polizas(matricula,idagente,finicio,ffin,tipo) values(xmatricula,
xidagente, xfinicio, xffin, xtipo);
set datos=concat("La moto: ",xmatricula," se ha añadido");
return datos;
end //
```

```
mysql> select crea_poliza("2187MFP",1,'2023-01-03','2024-01-03',"TRI") //
+-----+
| crea_poliza("2187MFP",1,'2023-01-03','2024-01-03',"TRI") |
+-----+
| La moto: 2187MFP se ha añadido |
+-----+
1 row in set (0.00 sec)

mysql> select * from polizas where matricula="2187MFP" //
+-----+-----+-----+-----+-----+-----+
| idpoliza | matricula | idagente | finicio   | ffin      | tipo |
+-----+-----+-----+-----+-----+-----+
| 11       | 2187MFP   | 1        | 2023-01-03 | 2024-01-03 | TRI  |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

7. Crear una tabla llamada **MOTOS_BORRADAS** con la misma estructura que **MOTOS**. A continuación, crear una función para eliminar una moto determinada, pero antes de hacerlo, tendrá que escribir el registro en **MOTOS_BORRADAS**. La función devolverá "Moto eliminada".

```
CREATE TABLE motosborradas AS (SELECT * FROM motos);

create function borrar_moto(xmatricula varchar(8)) returns varchar(50)
DETERMINISTIC
BEGIN
declare datos varchar(50);
DELETE FROM motosborradas WHERE matricula=xmatricula;
set datos=concat("La moto: ",xmatricula," se ha borrado");
return datos;
end //
```

```
mysql> select borrar_moto("0035LBB") //
+-----+
| borrar_moto("0035LBB") |
+-----+
| La moto: 0035LBB se ha borrado |
+-----+
1 row in set (0.01 sec)
```

8. Crear una función a la que se pasa el código de un cliente y devuelva el total de motos que tiene aseguradas, junto con el coste de todas ellas. Habrá que comprobar que dicho cliente existe.

```
create function motos_aseguradas(xidcli int) returns int DETERMINISTIC
BEGIN
declare datos int;
declare cont int;
select count(*) into cont from clientes where idcli=xidcli;
if cont>0 then
select count(*) into datos from polizas,motos where
motos.matricula=polizas.matricula and idcliente=xidcli;
ELSE
set datos=0;
end if;
return datos;
end //
```

```
mysql> select motos_aseguradas(1) //
+-----+
| motos_aseguradas(1) |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select motos_aseguradas(20) //
+-----+
| motos_aseguradas(20) |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

9. Crear una función a la que se le pasa un tipo de póliza y un porcentaje. Tendrá que actualizar el precio incrementándolo en el porcentaje indicado. Si el tipo de póliza no existe, habrá que indicarlo.

```
create function actualizar_poliza(xtipo varchar(4),porcentaje int) returns
varchar(30) DETERMINISTIC
BEGIN
declare datos varchar(30);
declare cont int;
select count(*) into cont from tipopoliza where idtipo=xtipo;
if cont>0 then
update tipopoliza set precio=precio+((precio*porcentaje)/100) where idtipo=xtipo;
set datos=concat(xtipo," ha sido actualizado");
ELSE
set datos=concat(xtipo," no existe");
end if;
return datos;
end //
```

```
mysql> select actualizar_poliza("TRI",20) //
+-----+
| actualizar_poliza("TRI",20) |
+-----+
| TRI ha sido actualizado    |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select actualizar_poliza("TRO",20) //
+-----+
| actualizar_poliza("TRO",20) |
+-----+
| TRO no existe              |
+-----+
1 row in set (0.00 sec)
```

10. Crear una función para hacer una copia de seguridad de todas las tablas. Se nombrarán igual pero añadiendo a cada tabla el sufijo _BK. La función devolverá el número total de registros que han sido copiados.

```
mysql> create function copia() returns varchar(30) DETERMINISTIC
-> BEGIN
-> declare datos varchar(30);
-> declare cuenta int;
-> select count(*) into cuenta from motos,agentes,clientes,polizas,tipopoliza;
-> CREATE TABLE motos_BK AS (SELECT * FROM motos);
-> CREATE TABLE agentes_BK AS (SELECT * FROM agentes);
-> CREATE TABLE clientes_BK AS (SELECT * FROM clientes);
-> CREATE TABLE polizas_BK AS (SELECT * FROM polizas);
-> CREATE TABLE tipopoliza_BK AS (SELECT * FROM tipopoliza);
-> select concat("Se han copiado: ",cuenta," registros.") into datos;
-> return datos;
-> end //
ERROR 1422 (HY000): Explicit or implicit commit is not allowed in stored function or trigger.
```