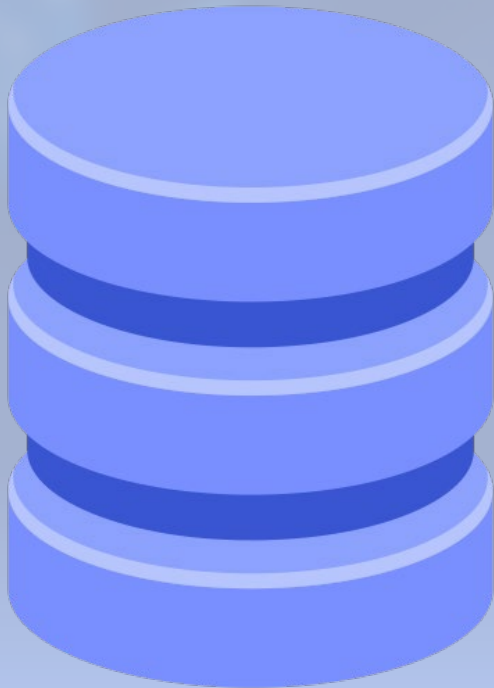


# **PROCEDIMIENTOS III: MYSQL**

---



---

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

1. Crear un procedimiento para obtener los coches alquilados de un determinado tipo.

```
CREATE Procedure tipo_coche (IN xtipo VARCHAR(2))
BEGIN
select coches.matricula, marca, modelo from coches, alquiler where
coches.matricula=alquiler.matricula and cod_tipo=xtipo;
END //
```

```
mysql> call tipo_coche ("TU");
+-----+-----+-----+
| matricula | marca      | modelo |
+-----+-----+-----+
| 2550KLP   | Nissan     | Leaf   |
| 8348JFT   | Mazda      | 3       |
| 5385JYD   | Mercedes-Benz | Clase A |
| 5385JYD   | Mercedes-Benz | Clase A |
+-----+-----+-----+
4 rows in set (0.01 sec)
```

2. Crear una función llamada días que calcule la diferencia en días de dos fechas dadas.

```
CREATE FUNCTION dias (fecha date, fechas date) RETURNS int DETERMINISTIC
BEGIN
DECLARE resultado int;
IF fechas < fecha THEN
SELECT TIMESTAMPDIF(day,fecha,fechas) into resultado;
ELSE
SELECT TIMESTAMPDIF(day,fechas,fecha) into resultado;
END IF;
RETURN resultado;
END //
```

```
mysql> SELECT dias('2023-2-10',curdate());
+-----+
| dias('2023-2-10',curdate()) |
+-----+
| 14 |
+-----+
1 row in set (0.00 sec)
```

### 3. Crear una función para obtener el precio de un alquiler (utilizando la función anterior). Aplicar esa función a todos los coches alquilados.

```
create function alquiler(xmatricula varchar(7),f_alq date, f_dev date) returns
decimal(8,2) deterministic
begin
declare total decimal(8,2);
select preciodia*dias(f_alq,f_dev) into total from coches,alquiler where
coches.matricula=alquiler.matricula and alquiler.matricula=xmatricula and
date(f_alquiler)=f_alq and date(f_devolucion)=f_dev;
return total;
end//
```

```
mysql> select alquiler("3859JTF",'2021-09-10','2021-09-11');
+-----+
| alquiler("3859JTF",'2021-09-10','2021-09-11') |
+-----+
| 108.55 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> select alquiler(matricula,f_alquiler,f_devolucion) from alquiler;
+-----+
| alquiler(matricula,f_alquiler,f_devolucion) |
+-----+
| 108.55 |
| 2517.68 |
| 434.20 |
| NULL |
| 108.81 |
| 123.60 |
| 21939.26 |
| 9710.82 |
| NULL |
| NULL |
| NULL |
| NULL |
| NULL |
| NULL |
+-----+
```

### 4. Crear un procedimiento que actualice el precio de cada coche de un tipo determinado incrementándolo en un porcentaje concreto.

```
CREATE PROCEDURE aumento (IN xtipo varchar(2),IN xpor INT)
BEGIN
UPDATE coches set preciodia=preciodia+((preciodia*xpor)/100) WHERE cod_tipo=xtipo;
SELECT "Se han actualizado los precios";
END //
```

```
mysql> call aumento("TU", 10);
+-----+
| Se han actualizado los precios |
+-----+
| Se han actualizado los precios |
+-----+
1 row in set (0.00 sec)
```

5. Crear un procedimiento llamado **totalpagar (fecha)** que liste los coches devueltos en la fecha que se pasa como parámetro, calculando el precio total a pagar por el alquiler de esos coches. Se pueden reutilizar funciones creadas en los puntos anteriores.

```
CREATE PROCEDURE totalpagar(xfecha date)
BEGIN
SELECT coches.matricula, marca, modelo,
alquiler(coches.matricula,f_alquiler,f_devolucion) from coches, alquiler where
date(f_devolucion)=xfecha and alquiler(coches.matricula,f_alquiler,f_devolucion) is
not NULL;
END //
```

```
mysql> call totalpagar('2021-09-11');
+-----+-----+-----+-----+
| matricula | marca | modelo | alquiler(coches.matricula,f_alquiler,f_devolucion) |
+-----+-----+-----+-----+
| 3859JTF | Toyota | Prius | 108.55 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

6. Crear un procedimiento que me devuelva los datos del coche más potente.

```
CREATE PROCEDURE mas_pot()
BEGIN
select * from coches order by potencia desc LIMIT 1;
END //
```

```
mysql> call mas_pot;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| matricula | marca | modelo | Cod_color | f_matricula | cod_tipo | cod_combustible | kms | potencia | consumo | vmax | preciodia |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 8345JGY | Chrysler | 300 Luxury | NE | 2017-09-14 | LI | D | 2034 | 218 | 11.03 | 158 | 198.22 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## 7. Crear un procedimiento para listar los coches que no se han alquilado en la última semana.

```
CREATE PROCEDURE no_alq()
BEGIN
select matricula,marca,modelo from coches where matricula not in (select matricula
from alquiler where timestampdiff(day,f_alquiler,curdate())>7);
END //
```

```
mysql> call no_alq();
+-----+-----+-----+
| matricula | marca | modelo |
+-----+-----+-----+
| 1234HVD | Kia | Cee'd |
| 2550KLT | Nissan | Qashqai |
| 3070KZZ | Mini | Cooper S |
| 6839HGL | Volkswagen | Polo |
| 8345JGY | Chrysler | 300 Luxury |
| 4843MXD | Renault | Megane |
| 1234HKD | Ford | Mustang |
| 1234HKY | Ford | Mustang |
+-----+-----+-----+
8 rows in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)
```

## 8. Crear un procedimiento al que se le pasan dos fechas y liste los coches matriculados entre esas fechas.

```
CREATE PROCEDURE fecha_matricula(IN fecha date, IN fechas date)
BEGIN
SELECT matricula, marca, modelo from coches where f_matricula between fecha and
fechas;
END //
```

```
mysql> call fecha_matricula('2018-01-01','2019-01-01');
+-----+-----+-----+
| matricula | marca | modelo |
+-----+-----+-----+
| 2550KLP | Nissan | Leaf |
| 2550KLT | Nissan | Qashqai |
| 3661JJR | Seat | Ibiza |
| 3859JTF | Toyota | Prius |
| 3945JZF | Citroën | Berlingo |
| 5385JYD | Mercedes-Benz | Clase A |
| 6930JVS | Ford | Fiesta |
| 8348JFT | Mazda | 3 |
+-----+-----+-----+
8 rows in set (0.00 sec)
```

**9. Crear un procedimiento para añadir un coche nuevo. Comprobar que la matrícula que se introduce no exista previamente en la base de datos.**

```
CREATE PROCEDURE coche_nuevo(IN xmatricula varchar(7), IN xmarca varchar(25), IN
xmodelo varchar(50), IN xcolor varchar(2), IN xfmatr date, IN xtipo varchar(2), IN xcomb
varchar(1), IN xkms int, IN xpotencia smallint, IN xcons decimal, IN xvmax smallint, IN
xpreciodia decimal)
BEGIN
declare cont int;
select count(*) into cont from coches where matricula=xmatricula;
if cont>0 THEN
select "La matricula ya se encuentra en la base de datos";
ELSE
INSERT INTO coches
values(xmatricula,xmarca,xmodelo,xcolor,xfmatr,xtipo,xcomb,xkms,xpotencia,xcons,xvmax,
xpreciodia);
END IF;
END //
```

```
mysql> call coche_nuevo("4843MXD", "Renault", "Twingo", "R0", '2022-02-03', "TU", "G", 101000, 90, 6.4, 180, 50.35);
+-----+
| La matricula ya se encuentra en la base de datos |
+-----+
| La matricula ya se encuentra en la base de datos |
+-----+
1 row in set (0.00 sec)
```

```
mysql> call coche_nuevo("7239MFP", "Renault", "Twingo", "R0", '2022-02-03', "TU", "G", 101000, 90, 6.4, 180, 50.35);
Query OK, 1 row affected, 2 warnings (0.01 sec)
```

**10. Crear un procedimiento para realizar copias de seguridad de las tablas coches, clientes y alquileres.**

```
CREATE PROCEDURE copia()
BEGIN
DROP TABLE IF EXISTS coches_bk, alquiler_bk, clientes_bk;
CREATE TABLE coches_bk AS (SELECT * FROM coches);
CREATE TABLE alquiler_bk AS (SELECT * FROM alquiler);
CREATE TABLE clientes_bk AS (SELECT * FROM clientes);
END //
```

```
mysql> call copia();
Query OK, 12 rows affected (0.05 sec)
```