

---

## PRÁCTICA 21: PROCEDIMIENTOS ALMACENADOS

---

1. Crea la tabla ***Categorías***:

idCategoria: Numérico entero, clave principal  
nombre: Alfanumérico (50)  
descripción: Alfanumérico (100)

2. Crea un procedimiento que inserte una categoría en la tabla ***Categorías***, a partir de los datos que recibe como parámetros de entrada, controlando que tanto el id como el nombre no se repitan.
3. Crea un procedimiento almacenado que permita buscar una categoría por su nombre.
4. Crea un procedimiento almacenado que permita eliminar una categoría a partir de su id, comprobando que dicho id existe y mostrando un mensaje de error en caso contrario.
5. Crea un procedimiento almacenado que permita modificar el nombre y la descripción de una categoría, verificando que no se repite el nombre de la categoría.

---

## PRÁCTICA 21: PROCEDIMIENTOS ALMACENADOS

---

Realiza los siguientes ejercicios para las bases de datos de ejemplo que vienen instaladas por defecto con MySQL, *Shakila* y *World*

6. Crea un procedimiento que visualice todas las películas cuyo coste de reemplazo sea superior a un valor que se pasará como parámetro de entrada. ¿Cuántas películas tienen un coste de reemplazo superior a 20€?
7. Crea un procedimiento que visualice todas las películas cuyo coste de reemplazo esté comprendido entre dos cantidades que se pasarán como parámetros de entrada. ¿Cuántas películas tienen un coste de reemplazo superior a 20€ e inferior a 21,99?
8. Crea un procedimiento que reciba como parámetros de entrada el continente y la lengua y obtenga todos los países de ese continente que hablen esa lengua, ¿qué países de Asia tienen como lengua entre otras el inglés? Nota: El tipo *enum* es compatible con el tipo *varchar*.
9. Crea una función que calcule el volumen de una esfera cuyo radio de tipo *FLOAT* se pasará como parámetro. Realiza una consulta después para calcular el volumen de una esfera de radio 5.
10. Crea un procedimiento almacenado (SP) que cambie el mail de un cliente, tabla *costumer*, por otro que se pasará como parámetro. El SP recibirá dos parámetros, el identificador del cliente y el nuevo mail. Ejecuta el SP con valores concretos.
11. Crea un procedimiento almacenado (SP) que tenga como parámetros el nombre, apellidos y el nuevo mail de un cliente de la tabla *costumer*. Utilizar el procedimiento del ejercicio 5 para cambiar el mail.
12. Crea una función que devuelva el número de actor de la tabla *actor*, pasando como parámetros el apellido y nombre.
13. Crea un procedimiento que visualice las películas cuya categoría (comedia, drama, ...) se pasa como parámetro). Llama después a este procedimiento para obtener todas las películas de la categoría drama y de la categoría comedia. ¿Qué ventaja le encuentras a realizar esta consulta de esta forma a realizarla de forma directa a través de sentencias SQL?
14. Crea un procedimiento que pase dos parámetros de entrada, identificador de categoría e identificador de actor y obtenga los datos de las películas sobre esa categoría en las que ha trabajado ese actor. Prueba el ejemplo con el actor 182 y la categoría 2.
15. Modifica el procedimiento para que tenga 3 parámetros de entrada, nombre, apellidos del actor y nombre de la categoría y visualice para un determinado actor y una determinada categoría, las películas en las que ha trabajado. Realiza una versión sin llamar al procedimiento del ejercicio 14 y otra versión llamándolo. Encuentra todas las películas de animación en las que ha trabajado el actor DEBBIE AKROYD. Busca todas las películas Documentales en las que ha trabajado el actor ED CHASE.

---

## PRÁCTICA 21: PROCEDIMIENTOS ALMACENADOS

---

16. Crea un procedimiento que tenga un parámetro de entrada que será el nombre de la categoría y un parámetro de salida que contendrá el número de películas para esa categoría.
17. Crea un procedimiento que reciba una cadena que puede contener letras y números y sustituya los números por ' \* '. Por ejemplo, si introducimos la cadena '12abcd23rts' devolverá la cadena '\*\*abcd\*\*rts'. Utiliza las funciones de manejo de cadenas que incorpora MySQL:  
<https://dev.mysql.com/doc/refman/5.7/en/string-functions.html>
18. Desarrolla una función que devuelva el número de años completos que hay entre dos fechas que se pasan como parámetros. Utiliza la función DATEDIFF. Tienes más funciones de manejo de fecha y hora para MySQL en:  
<https://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html>
19. Escribe una función que, haciendo uso de la función anterior, devuelva los trienios que hay entre dos fechas.
20. Escribe un procedimiento que permita borrar un actor cuyo identificador se pasará como parámetro. Si el actor cuyo número se ha pasado como parámetro no existe, aparecerá un mensaje diciendo "Este actor no existe". Comprueba el funcionamiento del procedimiento. ¿Qué ocurre cuando se intenta borrar un actor que ya existe?
21. Escribe un procedimiento que añada una nueva entrada a la tabla film\_category que guarda las categorías a las que pertenece cada película. El procedimiento recibirá como parámetros el identificador de película y el nombre de la categoría. Deberían tenerse en cuenta las siguientes situaciones:
  - a) Si no existe la película correspondiente al número pasado como parámetro, se mostrará un mensaje diciendo 'El film con nº x no existe'. X es el número de película pasado como parámetro y se abandonará el procedimiento en este caso.
  - b) Si no existe la categoría pasada como parámetro, se mostrará un mensaje diciendo 'La categoría x no existe', donde x es el nombre de categoría pasado como parámetro y se abandonará el procedimiento.
  - c) Si ya existe la entrada o fila que se pretende añadir a film\_category, aparecerá un mensaje diciendo 'la película x ya está registrada en esa categoría'. En caso contrario se procede al alta de dicha fila en la tabla film\_category.
  - d) Comprobar el procedimiento en las distintas situaciones.

REALIZA DOS VERSIONES DEL EJERCICIO, UNA SIN UTILIZAR HANDLERS Y OTRA USÁNDOLOS.