# **EXAMEN Tema 4**

Acceso a Almacenes de Datos

### Instalación de los datos

Para la instalación de los datos usaremos el archivo filmdb.sql que creará la base de datos y las tablas necesarias. Además añadirá datos de prueba.

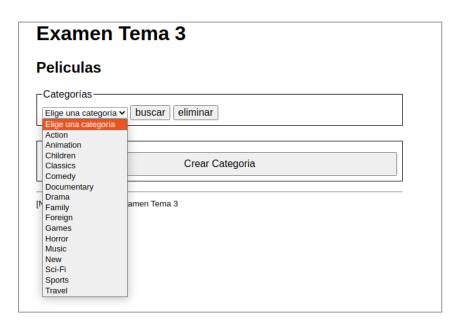
Los datos de prueba son 16 categorías, 1000 películas y 1000 relaciones película-categoría.

Para la instalación podemos usar:

- phpmyadmin
- HeidiSQL
- Extensión MySQL de VS Code
- o cualquier otra aplicación que permita ejecutar el sql.

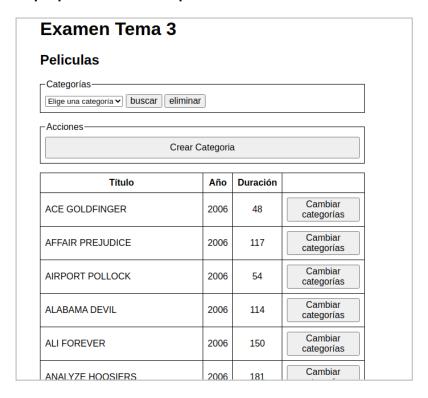
# **Ejercicios**

Página film.php: desplegable



 En un principio se debe mostrar un desplegable con todas las categorías (se obtiene de la tabla category). No se debe ver la tabla con las películas si no se elige una categoría.

### Página film.php: listado de películas



- 2. Una vez seleccionada la categoría, se debe mostrar igualmente el desplegable con todas las categorías y la tabla con todas las películas de dicha categoría.
- 3. Si no existen películas de esa categoría, debe indicar que no hay películas.



Ejemplo consulta de película por categoría "Animation":

SELECT film.film\_id, film.title, film.release\_year, film.length FROM film, film\_category

WHERE film.film\_id = film\_category.film\_id AND film\_category.category\_id = 2;

# Página create.php : crear una nueva categoría

4. Crea una categoría con el nombre que le indiquemos.

5. Se inserta en la columna 'last\_update' directamente la fecha obtenida en mysql con el comando CURRENT\_TIMESTAMP.

# Examen Tema 3 Nueva categoría Volver ¡Usuario creado satisfactoriamente! [Nombre] [Apellidos] | Examen Tema 3

### Ejemplo consulta:

INSERT INTO category (category\_id, name, last\_update) VALUES (NULL, 'kkfu', CURRENT\_TIMESTAMP)

## Página film.php: borrar una categoría.

- 6. Cuando se pulsa el botón eliminar, se debe eliminar la categoría indicada en el desplegable.
  - Debes tener en cuenta que si la categoría tiene películas asociadas, MySQL no te va a dejar. Dará un error. No muestres dicho error e indica de forma correcta al usuario que no se puede borrar hasta que no se quiten las películas de la categoría.

Página category\_film.php: Modificar las categorías de una película.

Examen Tema 3
<u>Volver</u>
Categorías actualizadas
Categorías de la pelicula: CANDIDATE PERDITION
<ul> <li>Action</li> <li>Animation</li> <li>Children</li> <li>Classics</li> <li>Comedy</li> <li>Documentary</li> <li>Drama</li> <li>Family</li> </ul>
<ul> <li>Foreign</li> <li>Games</li> <li>Horror</li> <li>Music</li> <li>New</li> <li>Sci-Fi</li> <li>Sports</li> <li>Travel</li> </ul>
Programación     Actualizar  [Nombre] [Apellidos]   Examen Tema 3

- 7. En la página film.php, cuando se pulsa el botón "cambiar categorías" de una película, se debe ir a la página *category\_film.php* donde se muestra un listado con todas las categorías y estarán marcadas a las que pertenece dicha película.
- 8. El usuario podrá modificar dicha selección añadiendo o eliminando categorías. Al pulsar el botón actualizar, se debe actualizar la base de datos.
- 9. Usa procedimientos almacenados cuando sea necesario.
- 10. Usa transacciones para que se modifique todo o nada.

### Ayuda:

En la tabla film\_category, si existe un registro con el id de la película y el id de la categoría es que dicha película pertenece a esa categoría. Si no existe es que no pertenece.

Ejemplo de crear vinculación entre película y categoría: INSERT INTO film\_category (film\_id, category\_id, last\_update) VALUES (35, 2, CURRENT\_TIMESTAMP)

IDEA: Aunque no es lo más eficiente, para actualizar, podemos eliminar todas las categorías de una película y luego ir añadiendo las que se han marcado (independientemente de si estaban antes o no).

Para realizar el examen puedes usar tanto **mysqli** como **PDO**, a elegir.