#STORED PROCEDURE 1 - ´ordena\_tabla´

creamos un stored procedure que ordena la tabla corp\_d en forma ascendente si el parámetro de entrada es 1 o de forma descendente si el parmetro de entrada es 2;

Tablas con las que interactúa: corp\_d.

#STORED PROCEDURE 2 - ´inserta\_film\_playlist´

creamos un stored procedure que inserta un film en una playlist, para ello hay que insertar en 2 tablas primero en rep\_mov\_tab y luego en mod\_playlist, y luego modificar en la tabla rep\_list los campos de cantidad de peliculas y de minutos totales de la playlist

Tablas con las que interactúa: rep\_list, rep\_mov\_tab, mod\_playlist.

#STORED PROCEDURE 3 - ´elimina\_film\_playlist´

creamos un stored procedure que elimina un film en una playlist, para ello hay que eliminar los films en la tabla rep\_mov\_tab y luego insertar en mod\_playlist, para finalmente modificar en la tabla rep\_list los campos de cantidad de peliculas y de minutos totales de la playlist.

Tablas con las que interactúa: rep\_list, rep\_mov\_tab, mod\_playlist.

#STORED PROCEDURE 4 - ´c\_account\_user´

#Creamos un stored procedure que cuando se cree una cuenta de usuario, nos cree la cantidad de perfiles que el usuario nos proporcione con un nombre estándar

Tablas con las que interactúa: user\_account, id\_acc\_user, user\_id.

#STORED PROCEDURE 5 - ´c\_replist´

#Creamos un stored procedure que crea una lista de reproduccion en funcion de un id de usuario y un nombre para la lista.

Tablas con las que interactúa: rep\_list, crea\_playlist.

#STORED PROCEDURE 6 - ´r\_replist´

#Creamos un stored procedure que elimina una lista de reproduccion en funcion de un id de lista

Tablas con las que interactúa: rep\_list, crea\_playlist, mod\_playlist, del\_playlist, rep\_mov\_tab.

#STORED PROCEDURE 7 - ´r\_user´

#Creamos un stored procedure que elimina un usuario

Tablas con las que interactúa: rep\_list, crea\_playlist, mod\_playlist, del\_playlist, rep\_mov\_tab, user\_account, id\_acc\_user, user\_id..