1. Carga la base de datos Sakila
2. Crea una tabla LOG\_FILM en la para almacenar todos los cambios realizados a la tabla *film* los campos de la tabla son los siguientes:
   1. Id. número consecutivo, único y que no permite valores nulos
   2. Tipo. Tipo de operación realizada (siempre es update)
   3. Film\_id. id de el film que se modificó
   4. Last\_value. Valor anterior (no importa lo que se modifiqué, todo se castea a texto)
   5. New\_value. Valor modificado (no importa lo que se modifiqué, todo se castea a texto)
   6. Timestamp. Almacena la información de fecha y hora del cambio

create table LOG\_FILM(

id int auto\_increment not null primary key

Tipo varchar(7)

Film\_id int

Last\_value varchar(15)

New\_value varchar(15)

timestamp Timestamp);

1. Crea un stored procedure que reciba los datos necesarios para almacenar la información de la tabla LOG\_FILM

delimiter //

CREATE PROCEDURE `log\_film\_edit`(in tipoo varchar(7),in film int, in last varchar(15), in new varchar(15))

begin

insert into log\_film (Tipo,Film\_id,last\_value,new\_value) values (tipoo,film,last,new);

end//

1. Crea un trigger que se active cada vez que realizamos un update en la tabla film. Este trigger guarda la información requerida en la tabla LOG\_FILM utilizando el stored procedure del inciso 3

delimiter //

create trigger on\_up

after update

on log\_film

for each row

begin

call log\_film\_edit('update',new.film\_id,'', new.original\_language\_id);

end//

1. Realiza un procedimiento almacenado (y cursores) que permita agregar el idioma original (*original\_language*) a la tabla *film*. Por el momento no es posible saber el lenguaje original de todas las películas, pero sí se sabe la siguiente información
   1. Todas las películas con categoría “Documentary” están en italiano
   2. Todas las películas con categoría “Foreign” están en japonés
   3. Todas las película en las que actúa SISSI SOBIESKI están en alemán
   4. Todas las películas en las que actúa ED CHASE están en mandarín
   5. Todas las películas en las que actúa AUDREY OLIVIER están en francés
   6. El resto de películas está en inglés

DELIMITER //

CREATE PROCEDURE curdemo()

BEGIN

alter table films add column original\_language varchar(10);

DECLARE done INT DEFAULT FALSE;

DECLARE fname, lname,genere,name varCHAR(20);

DECLARE cur1 CURSOR FOR SELECT name,first\_name,last\_name FROM film

join film\_category on film.film\_id = film\_category.film\_id

join category on film\_category.category\_id = category.category\_id

join film\_actor on film.film\_id = film\_actor.film\_id

join actor on actor.actor\_id =film\_actor.actor\_id;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur1;

read\_loop: LOOP

IF done THEN

LEAVE read\_loop;

END IF;

FETCH cur1 INTO genere,fname,lname;

if(CONCAT((fname)," ",(lname)='SISSI SOBIESKI')

{

update film SET original\_language='aleman';

}

FETCH cur1 INTO genere,fname,lname;

elseif(CONCAT((fname)," ",(lname)='ED CHASE')

{

update film SET original\_language='mandarín';

}

FETCH cur1 INTO genere,fname,lname;

elseif(CONCAT((fname)," ",(lname)='AUDREY OLIVIER')

{

update film SET original\_language='mandarín';

}

FETCH cur1 INTO genere,fname,lname;

elseif(genere='Foreign')

{

update film SET original\_language='japonés';

}

FETCH cur1 INTO genere,fname,lname;

elseif(genere='Documentary')

{

update film SET original\_language='italiano';

}

elseif

{

update film SET original\_language='inglés';

}

END LOOP;

CLOSE cur1;

END;

//

1. Utiliza db2 para modelar el siguiente caso práctico

La empresa Mary Gomitas S.A. es una empresa de retail dedicada a la venta de gomitas de todos los sabores. La empresa desea que le diseñes una base de datos que le permita lo siguiente:

1. Almacenar toda la información relacionada con sus gomitas
2. Registrar las variaciones de precios de las gomitas
   1. Las gomitas cambian de precio el día 1o de febrero, 25 de abril y 25 de octubre. Se incrementa el precio es un 45% sobre su valor
   2. Las gomitas cambian de precio el 15 de febrero, 5 de mayo y 5 de noviembre. El nuevo precio es igual al precio anterior al 1 de febrero más un incremento de 10%

CREATE TABLE gomitas

(

id int not null,

precio decimal(6,2) not null,

empiza date not null,

termina date not null,

term business\_time(empiza, termina),

PRIMARY key(id, term without overlaps));

1. En enero se surte la tienda. Realiza al menos 12 inserciones de diferentes productos y con diferentes preciosa cada uno en la tabla o tablas que hayas creado en el punto 1
   1. El precio de los productos está considera de manera inicial para ser el mismo durante todo el año (1 enero del 2018 al 1 enero de 2019)

insert into gomitas(1, 12, '2018-01-01', '2019-01-01')

insert into gomitas(2, 16, '2018-01-01', '2019-01-01')

insert into gomitas(3, 19, '2018-01-01', '2019-01-01')

insert into gomitas(4, 23, '2018-01-01', '2019-01-01')

insert into gomitas(5, 26, '2018-01-01', '2019-01-01')

insert into gomitas(6, 29, '2018-01-01', '2019-01-01')

insert into gomitas(7, 31, '2018-01-01', '2019-01-01')

insert into gomitas(8, 18, '2018-01-01', '2019-01-01')

insert into gomitas(9, 36, '2018-01-01', '2019-01-01')

insert into gomitas(10,33, '2018-01-01', '2019-01-01')

insert into gomitas(11,13, '2018-01-01', '2019-01-01')

insert into gomitas(12,42, '2018-01-01', '2019-01-01')

insert into gomitas(13,41 , '2018-01-01', '2019-01-01'));

1. Realiza las actualizaciones de precios según el punto número 2.

update gomitas

for portion of business\_time FROM '2018-2-1' to '2019-1-1'

set precio = precio \* 1.45;

update gomitas

for portion of business\_time FROM '2018-2-15' to '2019-1-1'

set precio = (precio / 1.45) \* 1.1;

update gomitas

for portion of business\_time FROM '2018-4-25' to '2019-1-1'

set precio = precio \* 1.45;

update gomitas

for portion of business\_time FROM '2018-5-5' to '2019-1-1'

set precio = (precio / 1.45)\*1.1;

update gomitas

for portion of business\_time FROM '2018-10-25' to '2019-1-1'

set precio = precio \* 1.45;

update gomitas

for portion of business\_time FROM '2018-11-5' to '2019-1-1'

set precio= (precio / 1.45) \* 1.1;

1. Realiza las siguientes consultas:
   1. Consulta que muestre el precio promedio de cada producto durante todo el año

select SUM(precio)/COUNT(\*) as promedio from gomitas;

* 1. Consulta que los precios más altos de cada producto

select MAX(precio) as MAX from gomitas;

* 1. Consulta que los precios más bajos de cada producto

select MIN(precio) as MAX from gomitas;