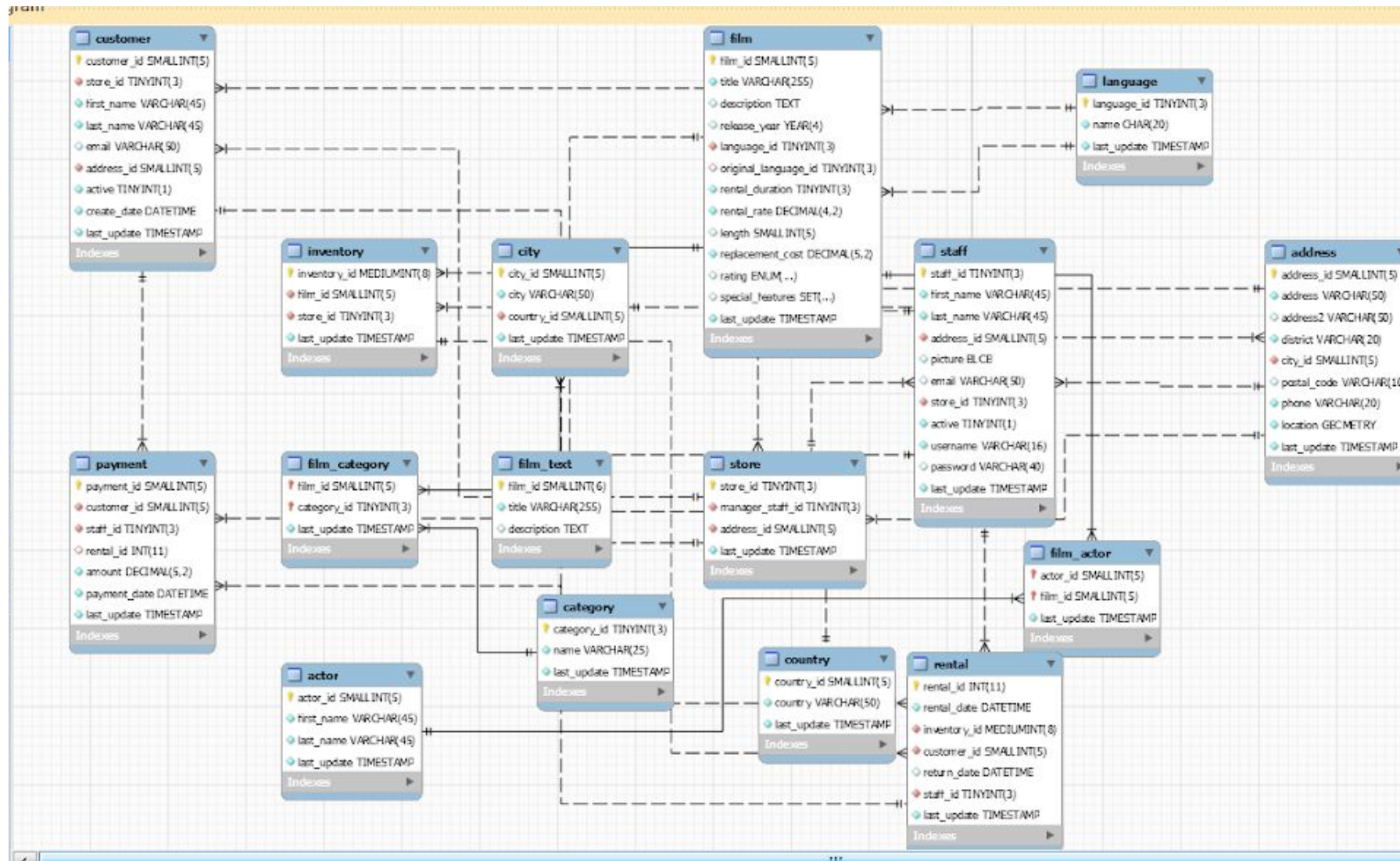


EJERCICIOS TEMA 9: PROGRAMACIÓN EN BASES DE DATOS



EJERCICIOS TEMA 9: PROGRAMACIÓN EN BASES DE DATOS

1. Crea un procedimiento que visualice todas las películas cuyo coste de reemplazo (replacement_cost en la tabla FILM) sea superior a un valor que se pasará como parámetro de entrada. ¿Cuántas películas tienen un costo de reemplazo superior a 20€?

NOTA: puedes ver la estructura de la tabla con la orden DESC FILM

2. Crea un procedimiento que visualice todas las películas cuyo coste de reemplazo (replacement_cost en la tabla FILM) esté comprendido entre dos cantidades que se pasarán como parámetros de entrada. ¿Cuántas películas tienen un coste de reemplazo entre a 20€ y 21,99 €?

3. Crear un PA que usando la tabla costumer, cambie el mail de un cliente por otro que se pasará como parámetro. El PA recibirá dos parámetros, el identificador del cliente y el nuevo mail (en la tabla son los campos customer_id, email). Ejecutar el PA para cambiar:

2,'patricia@gmail.com

customer_id	store_id	first_name	last_name	email	address_id	active	create_date
2	1	PATRICIA	JOHNSON	patricia@gmail.com	6	1	2006-02-14 22:04:36

4. Crear un procedure que tenga como parámetros de entrada el nombre, apellidos y el nuevo mail de un cliente de la tabla COSTUMER, (NOTA: utiliza el procedure del ejercicio anterior para cambiar el mail.)

call cambio_mail2('MARY','SMITH','mary1@gmail.com')

customer_id	EMAIL
1	mary1@gmail.com

5. Crea un procedimiento que muestre los campos TITLE, DESCRIPTION de las películas cuya categoría (comedia, drama,... se pasa como parámetro). Llama después a este procedimiento para obtener todas las películas de la categoría "Comedy"

6. 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. Utiliza las tablas FILM, FILM_ACTOR, FILM_CATEGORY (utiliza el comando DESC table para ver su estructura).

Prueba el ejemplo con el actor 180 y la categoría 2.

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate
820	SONS INTERVIEW	A Taut Character Study ...	2006	1	NULL	3	2.99
963	WATCH TRACY	A Fast-Paced Yarn of a ...	2006	1	NULL	5	0.99

7. 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 “EL ACTOR: N° DEL ACTOR NO EXISTE”.. Comprueba el funcionamiento del procedimiento intentado borrar un actor que no exista.

MENSAJE
EL ACTOR: 340 NO EXISTE

¿Qué ocurre cuando tratas de borrar un actor que ya existe? ¿Porqué?

CALL BORRAR_ACTOR(3)

FUNCIONES:

1. Crear una función que retorne el número de actor de la tabla **ACTOR**, pasando como parámetros el apellido y nombre. Ejecuta la función para ('CHASE', 'ED').

NUMERO
3

2. Crea una función que devuelva el número de años completos que hay entre dos fechas que se pasan como parámetros. Utiliza la función DATEDIFF para calcular los días transcurridos entre las dos fechas, divide los días obtenidos entre 365 y usa la función TRUNCATE para quedarte con el número de años completos.

AÑOS_COMPLETOS
('2018/05/25','1971/02/18')
47

3. Crear una función que, utilizando la función anterior, devuelve los trienios que hay entre dos fechas que le pasamos por parámetro.

NUMERO_TRIENIOS
15

TRIGGERS

- Visualiza los triggers de la BD sakila. Hay dos formas diferentes.
 - SHOW TRIGGERS
 - select * from information_schema.triggers where trigger_Schema='sakila'
- Construir un disparador que permita auditar las operaciones de borrado de datos que se realicen en la tabla actor según las siguientes especificaciones:
 - Crear la tabla AUDITARACTOR, con una sola columna **col1 VARCHAR2(200)**. Cuando se produzca cualquier borrado de datos en esta tabla se insertará una fila que contendrá: Fecha y hora, Número de actor, Apellido de actor, con el formato:

El actor XX de apellido XXXX fue borrado el día XXXXX

Para probar el funcionamiento del trigger, realiza las siguientes operaciones:

Inserta las siguientes filas en la tabla actor (444, 'Mar', 'Lopez') (445, 'Mar', 'Lopez') (446, 'Mar', 'Lopez')

Borra la primera fila que has insertado (delete from actor where actor_id=444) y comprueba los cambios en la tabla

auditar_actor:

COL1
EL ACTOR 444 DE APELLIDO Lopez FUE BORRADO EL DIA 2018-05-29

Borra las dos siguientes filas (delete from actor where first_name='Mar') y comprueba los cambios en la tabla auditar_actor.

EJERCICIOS TEMA 9: PROGRAMACIÓN EN BASES DE DATOS

COL1
EL ACTOR 444 DE APELLIDO Lopez FUE BORRADO EL DIA 2018-05-29
EL ACTOR 445 DE APELLIDO Lopez FUE BORRADO EL DIA 2018-05-29
EL ACTOR 446 DE APELLIDO Lopez FUE BORRADO EL DIA 2018-05-29

3. Escribir un trigger para auditar las modificaciones de nombre y apellidos en la tabla actores insertado en la tabla auditar_actor los siguientes datos: - Fecha - Número de actor - La operación de actualización: MODIFICACIÓN. - El valor anterior y el valor nuevo del nombre y el apellido. Inserta de nuevo alguna de las filas del ejercicio 2 y luego realiza una operación de actualización sobre esa fila para comprobar los cambios en la tabla AUDITAR_ACTOR.

```
INSERT INTO ACTOR (ACTOR_ID, FIRST_NAME, LAST_NAME) VALUES (444, 'Mar', 'Lopez');
UPDATE ACTOR SET FIRST_NAME='MARINA' WHERE ACTOR_ID=444;
SELECT * FROM AUDITAR_ACTOR;
```

2018-05-29	MODIFICACIÓN	Mar	MARINA
------------	--------------	-----	--------

4. Crea un trigger para la tabla film para que cuando se realice una actualización al campo replacement_cost no pueda sufrir un incremento mayor del 10% (es decir, si la operación de actualización va a suponer un aumento mayor del 10%, se pondrá por defecto una subida del 10%.) Comprueba el funcionamiento del trigger actualizando por ejemplo el replacement_cost de la película con identificador 1 a un valor de 60 ¿Qué coste de reemplazo tiene ahora el film ?

225	SELECT REPLACEMENT_COST FROM FILM WHERE FILM_ID=1
226	
227	
Result Grid Filter Rows: Export: Wrap Cell Content:	
	REPLACEMENT_COST
	20.99

```
UPDATE FILM SET REPLACEMENT_COST=60 WHERE FILM_ID=1;
```

Result Grid	Filter Rows:
REPLACEMENT_COST	
23.09	

5. Crear un trigger AFTER UPDATE que solo audite las modificaciones de título o descripción de la tabla film, de modo que si se ha producido una modificación de alguna de las dos columnas, se inserte en una nueva tabla el valor antiguo y el nuevo del título y/o la descripción. Realiza comprobaciones modificando alguno de los dos campos y también haciendo actualizaciones que no afecten a uno de esos campos (en este caso el trigger no debe dispararse)

```
UPDATE FILM SET TITLE='PROBANDO' WHERE FILM_ID=1
```

MENSAJE
ACADEMY DINOSAUR....PROBANDO