



Actividades UT 2. Bloque IV

TRABAJANDO CON LAS TABLAS DE U2 ALUMNOS, MATRICULAS, MODULOS, PROFESOR

Responder a las siguientes preguntas y si es necesario realizar las pruebas para ver que pasa:

- *Qué pasa si borro un registro de la tabla PROFESOR que tiene asignado un módulo*

Se produce una violación de restricción de la clave ajena

- *Probar si es posible borrar dicho profesor y el mensaje que da*

No es posible ya que existe una clave ajena con la opción RESTRICTED, con lo que no puedo eliminar un registro de una tabla que tiene asociado valores en otra tabla.

- *¿Si quiero poder ELIMINAR/BORRAR datos de un profesor de la tabla PROFESORES, independientemente de que tenga registros asociados en la tabla MODULOS, y sin que se elimine la información del módulo que imparte dicho profesor, ¿ qué podría hacer.....?*

Tendría que modificar dicha restricción de clave ajena poniéndola en vez de RESTRICTED hacerla ON DELETE SET NULL

- *Indicar como tendría que modificar la estructura de la tabla y concretamente dicha restricción.*

ALTER TABLE MODULOS DROP CONSTRAINT MODULOS_PROFESOR_FK1;

ALTER TABLE MODULOS

**ADD CONSTRAINT MODULOS_PROFESOR_FK1 FOREIGN KEY
(NIF_PROFESOR) REFERENCES PROFESOR (NIF)
ON DELETE SET NULL;**

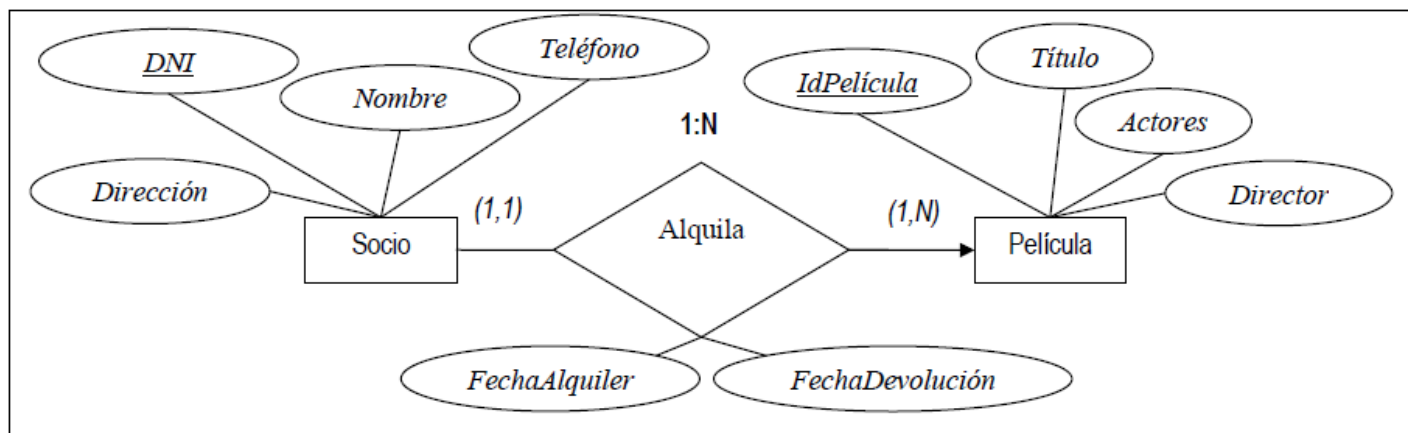


Realizar el esquema E-R del siguiente supuesto de VIDEOCLUB

- Supongamos que tenemos un videoclub, y lo queremos informatizar. Los socios del videoclub alquilan las películas. Un socio puede alquilar muchas películas cada vez, pero cada película sólo puede ser alquilada por un socio cada vez. La información de cada una de las entidades serán las necesarias, incluida la dirección y teléfono del socio
- En cada operación de alquiler debe poder registrarse la fecha de alquiler, y la fecha de devolución de la película alquilada, de forma que esta última se rellenará cuando sea devuelta la película.
- sólo nos interesa mantener información de las películas que están alquiladas en el presente

Solución:

- Los socios del videoclub alquilan las películas, por tanto se trata de una relación binaria (de grado 2) entre la entidad socio y la entidad película. Un socio puede alquilar muchas películas cada vez, pero cada película sólo puede ser alquilada por un socio cada vez. Por tanto la relación es del tipo 1:N, este factor determina el número de entidades asociadas dentro de la relación.
- La cardinalidad de una entidad identifica el número de ocurrencias máximo y mínimo que participan en una relación. La cardinalidad de la entidad Socio es (1,1) pues en un alquiler hay uno y sólo un socio, mientras que la cardinalidad de la entidad Película es (1,N) pues en un alquiler hay al menos una película y podrías ser más. En cada operación de alquiler debe poder registrarse la fecha de alquiler, y la fecha de devolución de la película alquilada.



Pasar el esquema E-R al modelo relacional

Transformación del modelo conceptual

Para ello, en principio basta con crear de cada entidad su tabla derivada. Acto seguido, de alguna forma tendremos que reflejar mediante tablas la relación que hemos identificado entre un socio y una película: alquiler.



Opción 1: Se generan las tablas derivadas de cada entidad, y el identificador de la entidad socio pasa a formar parte de la tabla derivada de la entidad película como atributo foráneo. Como la asociación está atribuida, tendremos que migrar también todos sus atributos.

SOCIOS(número de socio, nombre, dirección, teléfono)

PELICULAS(código película, título, director, reparto, **número de socio**, **fecha**)

Como se puede observar esta solución es factible debido a que no se viola el principio de clave (no se repite el atributo identificador), pero tiene el problema que para aquellas películas que no están alquiladas se generan celdas vacías en los campos número de socio y fecha.

nos el ejemplo:

SOCIOS			
numero de socio	nombre	dirección	teléfono
001	---	---	---
002	---	---	---

PELICULAS					
código	título	director	reparto	nº socio	fecha
A	---	---	---	001	27/01/07
B	---	---	---	001	27/01/07
C	---	---	---		

Esta sería la solución propuesta a partir de lo indicado en las diapositivas, es decir en una relación 1:N la clave de la entidad donde la relación es 1 migra a la tabla donde la relación es N, sin embargo analizando las tablas bien en la tabla de PELICULAS, los campos nº socio y fecha solo estarán cubiertos si la película esta prestada, y esto puede dar lugar a muchos registros con estos campos en blanco.

Opción 2: Se generan las tablas derivadas de cada entidad y una tercera tabla derivada de la asociación. Los atributos identificadores de cada entidad que participan en la asociación pasan a formar parte de esta tercera tabla como atributos foráneos.

SOCIOS(número de socio, nombre, dirección, teléfono)

PELICULAS(código película, título, director, reparto)

ALQUILERES(fecha, número de socio, **código de película**) → ya que una película no puede aparecer varias veces en la tabla al existir solo una película física y no existir copias, con lo cual al tener solo los alquileres presentes una película solo la tiene un socio, no puede aparecer en la tabla la misma película con dos socios

SOCIOS			
numero de socio	nombre	dirección	teléfono
001	---	---	---
002	---	---	---

PELICULAS			
código película	título	director	reparto
A	---	---	---
B	---	---	---
C	---	---	---

ALQUILERES		
fecha	numero de socio	código de película
27/01/07	001	A
27/01/07	001	B



Como se puede observar, esta opción es factible también, al tiempo que no genera valores de campo nulos. El único inconveniente que tiene es que genera más redundancia que la primera opción, pero esta redundancia es más que asumible y casi siempre es la óptima pese a que podamos tener un poco de redundancia.

Crear un documento WORD con las respuestas y comandos SQL a cada una de las cuestiones planteadas.

a. *Crear las tablas U2A a partir del modelo relacional*

Si elegimos la opción de 3 tablas, crearíamos:

1. primero la tabla de SOCIOS con su clave primaria
2. la tabla de PELICULAS con su clave primaria
3. por último crearía la tabla de préstamos, creando su clave primaria y el código de la película (suponiendo que solo hay una película para prestar) y con sus correspondientes claves ajenas (una a socios y otra a películas)

b. *Introducir algún dato en las tablas e indicar el orden en que se deben de introducir*

Usuario desde el que se realiza la acción: U2A

Instrucción SQL: Suponiendo que elegimos la opción de 3 tablas. Meteríamos primero datos de SOCIOS y PELICULAS no importa el orden en este caso. Teniendo datos de socios y películas ya podríamos insertar en la tabla préstamos.

Cada vez que insertamos una fila en una tabla al pulsar el ENTER y después ejecutar el COMMIT se genera una instrucción SQL INSERT que es la que permite insertar en una tabla.

c. *Realizar las acciones oportunas para que el usuario U2 pueda consultar las tablas anteriores*

Usuario desde el que se realiza la acción: SYSTEM o U2A

Instrucción SQL:

GRANT SELECT on U2A.SOCIOS to U2;

GRANT SELECT on U2A.PELICULAS to U2;

GRANT SELECT on U2A.PRESTAMOS to U2;

Indicar si asignamos privilegios de sistema o de objeto: OBJETO



- d. *Comprobar las instrucciones para poder consultar dichas tablas el usuario U2*

Usuario desde el que se realiza la acción: _____ U2 _____

Instrucción SQL:

```
SELECT * FROM U2A.SOCIOS;  
SELECT * FROM U2A.PELICULAS;  
SELECT * FROM U2A.PRESTAMOS;
```

Si lo hacemos desde el SQLdeveloper, entrando en la conexión de U2, nos vamos al objeto de su esquema OTROS USUARIOS, elegimos el usuario U2A y al abrirse su esquema deberíamos de poder ver sus tablas, y pinchando en ellas se nos muestran los datos

- e. *Realizar las acciones necesarias para que el usuario U2 pueda añadir en la tabla en la que se registran los prestamos u alquileres*

Usuario desde el que se realiza la acción: _____ U2A _____

Instrucción SQL: GRANT INSERT on U2A.PRESTAMOS to U2;

- f. *Comprobar que desde dicho usuario puede añadir un préstamo y consultar si se ha añadido el registro.*

Usuario desde el que se realiza la acción: _____ U2 _____

Instrucción SQL: acceder a la tabla PRESTAMO de U2A desde el esquema de U2 y probar a insertar un nuevo registro

- g. *Desde el usuario U2 modificar en la tabla SOCIOS el teléfono de uno de los socios. (Asegurarse de pulsar al finalizar la modificación el ENTER y después pulsar el botón de COMMIT para que la modificación sea efectiva y definitiva). ¿Qué sucede y por qué?*

Usuario desde el que se realiza la acción: _____ U2 _____

Instrucción SQL: no puede, ya que no tiene el privilegio de actualizar

- h. *Realizar los pasos necesarios para permitir que el usuario U2 pueda modificar en la tabla se SOCIOS solo los campos de dirección y teléfono*

Usuario desde el que se realiza la acción: _____ U2A _____

Instrucción SQL: GRANT UPDATE(teléfono) on U2A.SOCIOS to U2;

Indicar si asignamos privilegios de sistema o de objeto: ___ Objeto _____



- i. *Comprobar desde el usuario U2 que puede modificar el teléfono de un socio. (Asegurarse de pulsar al finalizar la modificación el intro y después pulsar el botón de COMMIT para que la modificación sea efectiva y definitiva). Comprobar al final desde el usuario U2A si aparece modificado*

Usuario desde el que se realiza la acción: _____ U2 _____

Instrucción SQL: acceder a la tabla SOCIO de U2A desde el esquema de U2 y probar a modificar el teléfono de un socio

- j. *Desde el usuario U2 añadir a un registro existente de la tabla donde se realizan los préstamos/alquiler la fecha de devolución*

Usuario desde el que se realiza la acción: _____ U2 _____

Instrucción SQL: no me dejaría ya que solo tengo privilegio de añadir (insert) pero no de modificar (update)

- k. *Que tendríamos que hacer para que pueda introducir a un registro de la tabla PRESTAMOS la fecha de devolución de la película*

Usuario desde el que se realiza la acción: _____ U2A _____

Instrucción SQL: GRANT UPDATE(fecha_devolución) on U2A.PRESTAMOS to U2;

Indicar si asignamos privilegios de sistema o de objeto: _____ OBJETO _____

- l. *Comprobar que el usuario U2 puede añadir la fecha de devolución de una película*

Usuario desde el que se realiza la acción: _____ U2 _____

Instrucción SQL: tendría que acceder desde el esquema de U2 al objeto OTROS USUARIOS y seleccionar U2A y en tablas de U2A abrir la tabla PRESTAMOS y modificar la fecha de devolución de alguno de los registros existentes

- m. *A partir de este momento el usuario U2 no podrá modificar ningún dato de la tabla SOCIOS. Realizar las acciones oportunas para comprobar los privilegios que tiene y eliminar aquellos que le permitían actualizar valores en dicha tabla.*

Usuario desde el que se realiza la acción: _____ U2A _____

Instrucción SQL: REVOKE UPDATE on U2A. SOCIOS to U2

Indicar si asignamos privilegios de sistema o de objeto: _____ OBJETO _____