

Actividad 3.2. Diseño lógico relacional

OCW VJ1220 Bases de datos

Objetivo

Obtener un esquema lógico relacional a partir de un esquema conceptual que ha sido elaborado usando el modelo entidad-relación a partir de una descripción de requisitos.

Qué hacer

El diseño lógico se trata en el **capítulo 7** del libro. Los conceptos sobre diseño lógico que trabajaremos en esta actividad son:

- Cómo se transforman las **entidades** del esquema conceptual en tablas del esquema lógico relacional, teniendo en cuenta que:
 - una entidad puede ser **fuerte** o **débil**,
 - los atributos tienen una **cardinalidad**,
 - los atributos pueden ser **simples** o **compuestos**,
 - una entidad puede tener varios **identificadores**.
- Las tres formas que hay para representar las **relaciones uno a uno** en el esquema relacional:
 - **integrando** las dos entidades en una sola, dando lugar a una única tabla,
 - añadiendo una **clave ajena** en una de las tablas correspondientes a las entidades participantes en la relación,
 - añadiendo una **nueva tabla** para la relación.
- Las dos formas que hay para representar las **relaciones uno a muchos** en el esquema relacional:
 - añadiendo una **clave ajena** en una de las tablas correspondientes a las entidades participantes en la relación,
 - añadiendo una **nueva tabla** para la relación.
- La única forma que hay para representar las **relaciones muchos a muchos** en el esquema relacional:
 - añadiendo una **nueva tabla** para la relación.

Una vez realizada la lectura haz el ejercicio de diseño lógico correspondiente al esquema conceptual adjunto. Este esquema conceptual corresponde a la siguiente descripción de requisitos.

Descripción de requisitos

Nos encontramos en una empresa que diseña y fabrica tarjetas de proximidad. Este tipo de tarjetas permite realizar un control de accesos inteligente a las instalaciones de los edificios. Para ello, se instala un lector junto a cada puerta cuyo acceso se quiere controlar y la apertura de ésta depende de que el lector obtenga la autorización pertinente. Cuando una tarjeta se aproxima a un lector, éste la identifica y averigua los privilegios de acceso de dicha tarjeta consultando la base de datos. Si la tarjeta tiene permiso para acceder, la puerta se abre. En la base de datos que se va a diseñar también se registrará la información relativa a los accesos realizados a determinadas ubicaciones.

A continuación se explica el funcionamiento del sistema. Cuando se entrega una tarjeta a un usuario se vincula su *login* al número de la tarjeta. Un usuario sólo podrá tener una tarjeta vinculada y ésta es personal. Mediante el *login* del usuario la aplicación podrá acceder a otra base de datos y conseguir información sobre él, por lo que no necesitamos almacenar otros datos sobre usuarios.

Cuando se vincula un usuario a una tarjeta se guarda también la fecha en la que se ha vinculado y la fecha en que caduca la tarjeta (en principio es 30 días después de la fecha de vinculación, aunque esta fecha se va actualizando conforme pasa el tiempo). A cada tarjeta se le asignan uno o varios roles. Los roles se establecerán previamente desde la aplicación y se guardarán en la base de datos, pudiendo ser modificados en cualquier momento. Un rol tiene un nombre identificativo y una descripción asociada. Por ejemplo, el rol LDTI tiene la descripción *personal de limpieza de despachos edificio TI*.

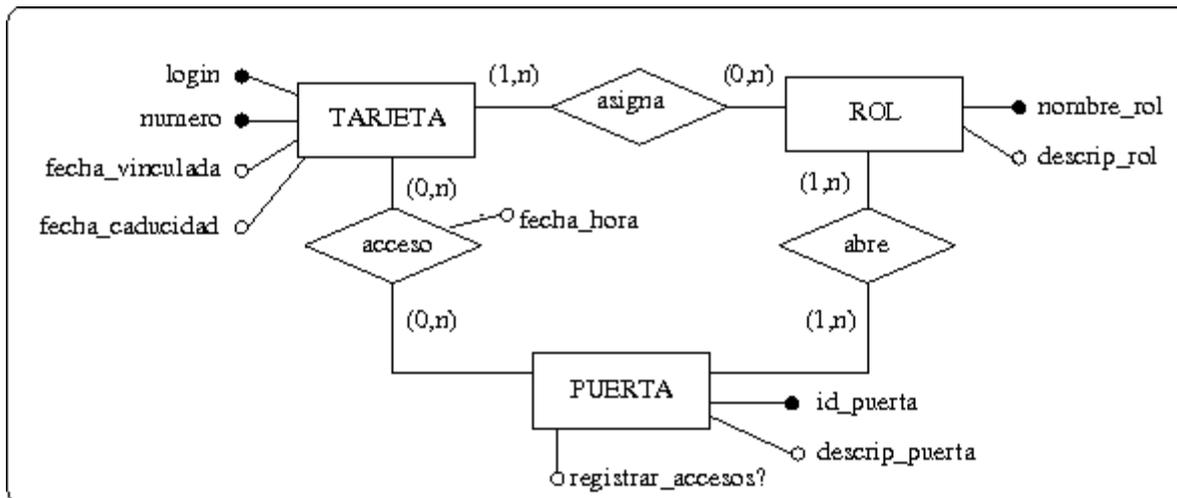
A cada rol se le permite acceder a una serie de puertas; esto también se establece mediante la aplicación y se debe guardar. De este modo, cuando un lector detecte una tarjeta podrá determinar si la tarjeta tiene permiso para acceder. De cada puerta se guarda su número de identificación (por ejemplo, TI2117DD) y una descripción (por ejemplo, *despacho de profesorado*).

Además de realizar el control de accesos, la aplicación registrará todos los accesos realizados a través de determinadas puertas. Para ello, en cada lector se marcará si se debe realizar el registro de accesos o no. Cuando la aplicación permita el acceso a una de estas puertas, se registrará dicho acceso anotando también la fecha y la hora. De este modo se puede saber siempre qué personas han accedido a los lugares sobre los que se desea tener un mayor control.

Como medida de seguridad, las tarjetas caducan cada 30 días y dejan de funcionar. Cuando esto sucede, se deben aproximar a un lector especial que es el que puede validar la tarjeta. Si se consigue la validación (el modo de conseguirla estará programado en la aplicación, no importa para este ejercicio), su fecha de caducidad se actualiza sumándole 30 días a la fecha actual, y la tarjeta vuelve ser útil.

Esquema conceptual

El esquema conceptual que se obtiene a partir de la descripción de requisitos que se ha detallado en el apartado anterior es el que se muestra a continuación:



Nótese que:

- La entidad TARJETA tiene dos identificadores posibles: el número que identifica a la tarjeta y el *login* del usuario al que se ha vinculado ya que nos dicen que solamente lo estará a un usuario (es personal).
- La relación entre ROL y PUERTA se ha denominado **abre** para referirse a que el ROL permite acceder a la PUERTA.
- El atributo **registrar_accesos?** será un booleano de manera que si el valor es verdadero, la aplicación deberá registrar en la base de datos todos los accesos realizados a esa puerta.

Ejercicio

Obtén el **esquema lógico relacional** correspondiente al esquema conceptual mostrado en el apartado anterior. Para obtener el esquema lógico sigue los pasos indicados en el libro. Como resultado deberás obtener un conjunto de tablas y para cada una debes indicar cuál es la clave primaria. Además, debes indicar qué columnas son claves ajenas y a qué tabla hacen referencia. Al final deberás dibujar un diagrama como los que aparecen en las figuras 7.7 y 7.8 del libro donde se muestran los esquemas lógicos de los ejercicios resueltos (páginas 151 y 152).