

Clase práctica: Ejemplos de Modelización

Fernando Bugni

Nota:

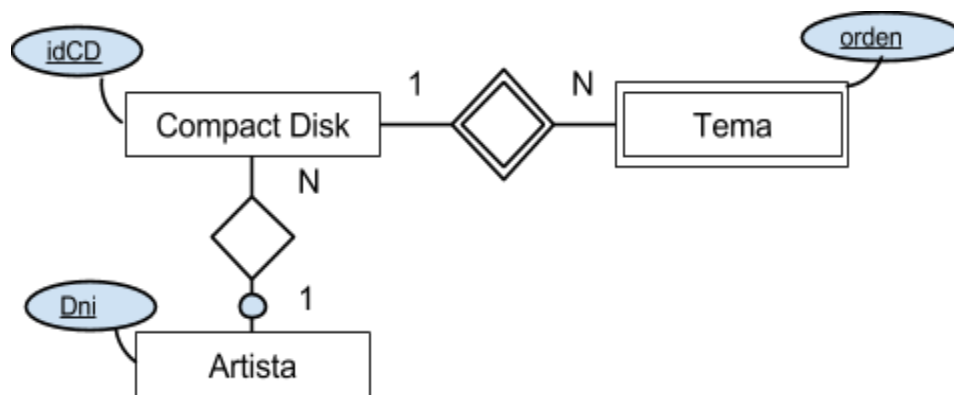
Si al leer este apunte tiene alguna sugerencia o encuentra algún error no dude en informarlo por mail a fernando.bugni@gmail.com con el fin de mejorar esta clase.

En esta clase vamos a ver distintos ejemplos de como realizar un modelado correcto en base a un enunciado. También aclararemos puntos importantes en el diseño que por lo general fueron errores en otros cuatrimestres.

Ejemplo 1:

Una pequeña disquería necesita una base de datos para organizar su mercadería. El dueño necesita guardar los Compact Disk que posee en el catálogo, así como el orden de los temas que posee. También se quiere saber cuales son los CDs que lanzó cada artista. Puede pasar que haya artistas que todavía no hayan lanzado su trabajo discográfico pero se los necesita guardados para saber que existen y que, eventualmente, van a llegar a stock. Cada tema esta en un único CD, o sea no hay covers; y además la disquería no acepta realizar Compact Disk de varios artistas, sino que deben ser todos solistas.

Solución:



Detalles a tener en cuenta:

- Notar que Tema (o Pista) es una entidad débil de Compact Disk. Un Tema que no tenga asociado un compact disk no debería existir. Además el tema necesita fuertemente el idCD para identificarse, o sea necesita la relación de una entidad fuerte. Notar también que para identificar un tema vamos a necesitar <idCD, orden>. Algo importante es que en las entidades débiles sólo existen relaciones 1:N.
- Puede pasar que el artista no haya lanzado ningún CD todavía. Es por eso que la participación en la relación con CD es parcial.
- Puede suceder que el atributo identificador orden sea creado con cualquier número e

inclusive no siendo sucesivos con respecto a la última inserción. Es por eso que se debe agregar la restricción adicional:

“El atributo identificador orden, en Tema, se debe ir incrementando de a un valor empezando por el valor 1”

Ejemplo 2:

La Asociación de Fútbol Argentino necesita guardar información sobre la historia de los equipos asociados. Necesita saber en que división se desempeñaba cada uno de los equipos en cada año. Hay un único campeonato por año para cada división. Se desea saber también para esa división y ese año en que puesto terminó el equipo. Además se desea guardar las canchas donde hace de local cada equipo.

Solución:



Detalles a tener en cuenta:

- Relacionando dos entidades con la cardinalidad M:N podemos guardar historia de lo que fue sucediendo entre ellas. Es importante notar que se debe agregar un atributo temporal para guardar las distintas instancias en el tiempo. Este atributo debe ser utilizado en la clave.

Puede suceder que para un equipo y una fecha dada haya varias divisiones. Esto no queremos que pase. Es por eso que debemos agregar una restricción adicional:

“Un equipo sólo puede participar de una división para un año dado”

Además se le puede agregar otros atributos, como por ejemplo puesto en que terminó el campeonato.

- Notar que al agregar el atributo puesto se debe agregar alguna restricción adicional para evitar tener dos equipos en el mismo puesto para el mismo campeonato. Esto se logra escribiendo en lenguaje coloquial abajo del diagrama:

“Todos los equipos que están relacionados a una división para un determinado año deben tener un valor distinto en el atributo puesto, y este debe ir desde 1 hasta la cantidad de equipos”

relacionados.”

- Imagínense que queremos obtener las canchas que jugaron en una división específica. ¿Utilizando este modelo se podría resolver esa pregunta? Sí, se podrían relacionar las 4 relaciones que surgen del diagrama y filtrar por una división en particular. ¿Podríamos tener una relación entre Cancha y División, así de esta forma contabilizar mejor las consultas? Sí pero estaríamos agregando redundancia de información al modelo.

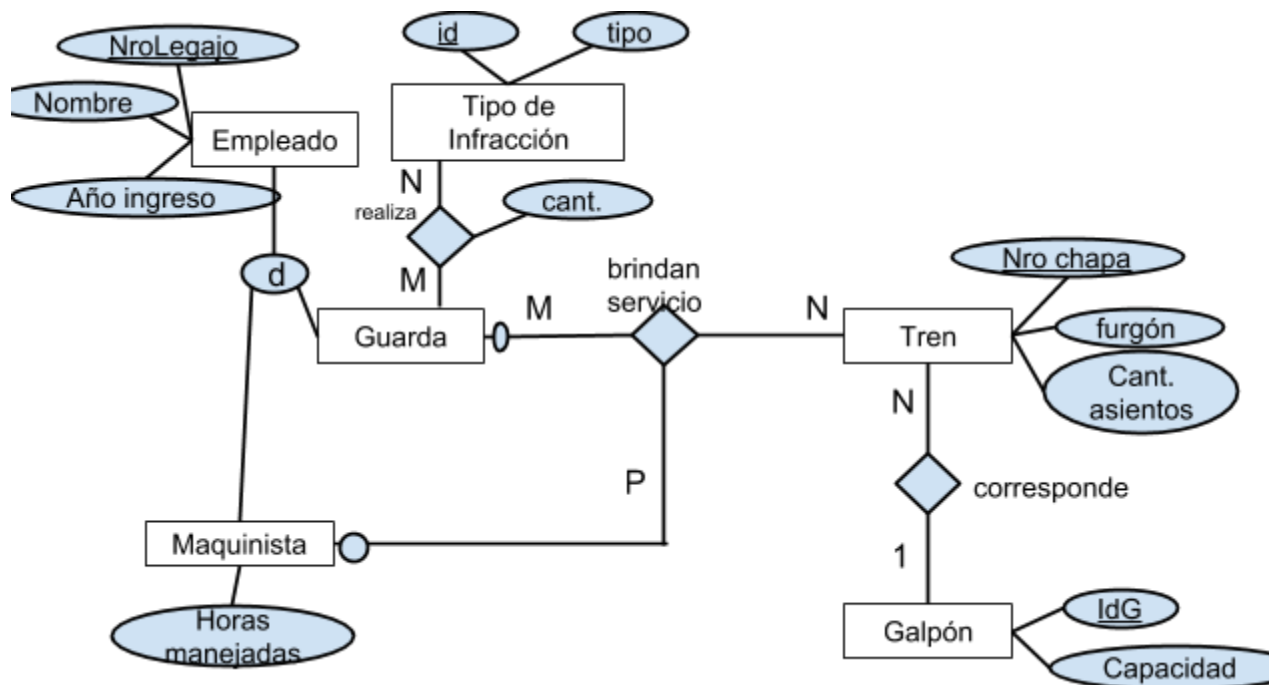
Ejemplo 3:

Una empresa de transporte ferroviario desea guardar la información de sus empleados. De cada uno de ellos desea saber su número de legajo, nombre, apellido y su año de ingreso. Sobre los maquinistas se desea saber la cantidad de horas manejadas y sobre los guardatren (también llamados guardas) la cantidad de infracciones y de que tipo pudo encontrar.

También se desea guardar cuales operarios trabajan en cada formación. Una formación se compone por un maquinista, un guarda y el tren. Se necesita de los tres para que se pueda prestar servicio. Por política de la empresa, se desea que la conformación de los servicios se realice con la mayor cantidad de combinaciones posible para que todos los empleados se conozcan. Puede suceder que haya maquinistas sin servicio asignado, idem para los guardas pero los trenes siempre están en servicio.

Además cada tren se guarda en un galpón asignado. Se desea guardar esa correspondencia. Los galpones tienen capacidad máxima, mientras que los trenes se desea saber la cantidad de asientos que posee y si prestan servicio de furgón.

Solución:



Detalles a tener en cuenta:

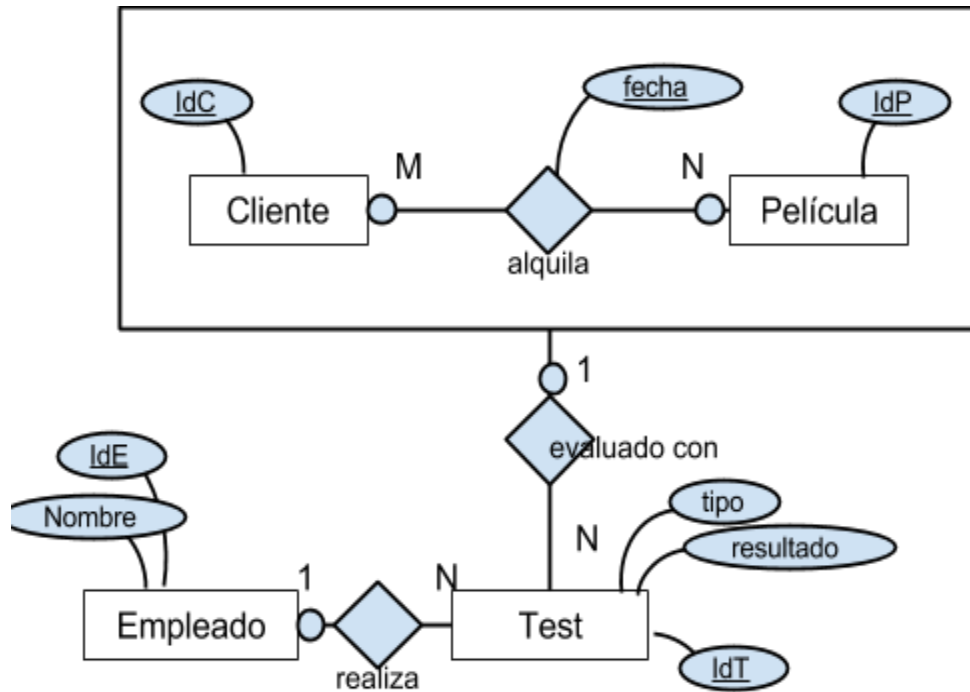
- Guardamos los datos de cada empleado en su entidad. Para cada categorización, se guarda sus datos específicos.
- Guardamos los tipos de infracción existente en 'Tipo de Infracción'. Esta se relaciona con Guarda en 'realiza' que posee cardinalidad M:N. La cantidad de infracciones es un atributo de dicha relación.
- Notar que 'brindan servicio' va a tener dos referencias a NroLegajo pero una corresponderá a Maquinista y otra a Guarda.
- Las relaciones que se forman entre Maquinista, Guarda y Tren (en brindan servicio) deben tener a los tres participantes. No se puede evitar ninguno. Pueden existir maquinistas que no se encuentren en la relación forman, idem para Guarda pero no para Tren. Notar que la cardinalidad permite realizar todas las combinaciones.
- Para ver la cardinalidad de la relación se debe fijar dos extremos y calcular la cardinalidad del que queda libre. Por ejemplo: dado un Maquinista y un Guarda se le asigna N trenes.
- Cada tren se guarda en un galpón específico. Notar que se desprende una restricción adicional de la capacidad: *La cantidad de trenes relacionados a un galpón tiene que ser menor a capacidad máxima.*

Ejemplo 4:

Un videoclub barrial necesita guardar en su base de datos las películas alquiladas por sus

clientes. En algunas ocasiones, los empleados deben verificar que en ese alquiler la película no fue dañada. Para ello el empleado debe realizar varios testeos de la calidad de la película (algunos de ellos: test de calidad de la imagen, calidad de sonido etc...). Se desea guardar el resultado de cada test y también que empleado verificó que alquiler, para que luego si falla la película se puede saber quien fue el que la alquiló y quien verificó su estado.

Solución:



Detalles a tener en cuenta:

- Para representar la historia de alquileres de los clientes se utiliza una relación nueva llamada "alquila". Esta va a guardar una referencia a Cliente, a Película y además una fecha. Los tres atributos se componen para ser clave primaria. De esta forma, puede haber una instancia del mismo cliente alquilando la misma película en fechas distintas.
- La relación entre Cliente y Película se va armando acorde a los alquileres que se van efectuando. Cuando se quiere testear una película se toma un par <Cliente, Película, fecha> y se lo relaciona con N cantidad de tests. Cada test es de un tipo determinado y nos da un resultado. Estos tests los realiza un Empleado.
- Notar la participación de "evaluado con" es parcial entonces no es necesario realizar tests para todos los alquileres.
- Un empleado cualquiera realiza cada uno de los tests que van surgiendo. Puede haber empleados que no realicen ningún test.
- Puede haber clientes que no realicen ningún alquiler. Idem para película.