

PHP y SQLite

Diseño de bases de datos



Diseño de base de datos

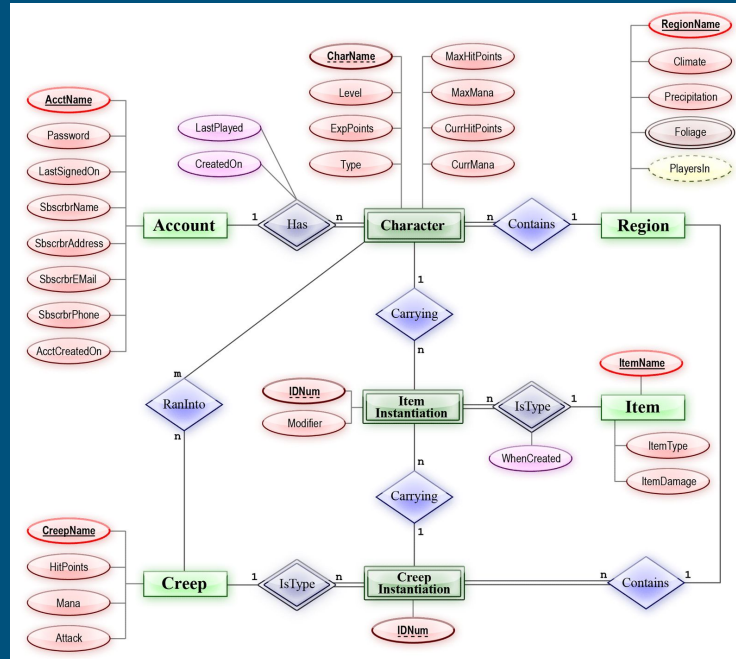
- 1) Etapa del diseño conceptual
- 2) Etapa del diseño lógico
- 3) Etapa del diseño físico

Etapa del diseño conceptual

- En esta etapa no importa la tecnología ni la plataforma en la que implementaremos nuestra base de datos.
- El resultado de esta etapa es un modelo de alto nivel.
- El modelo más utilizado es el modelo Entidad-Relación (Entity–relationship model) o simplemente modelo ER.

Etapa del diseño conceptual

Fuente: wikipedia.



Etapa del diseño lógico

- En la etapa del diseño lógico, adaptamos el modelo Entidad-Relación a una Sistema de Gestión de Bases de Datos (SGBD) seleccionada.
- Si se trata de un SGBD relacional, esta etapa obtendrá un conjunto de relaciones con sus atributos, claves primarias y claves foráneas.
- En esta etapa nos concentramos en las cuestiones tecnológicas relacionadas con el modelo de base de datos.

Etapa del diseño físico

- En la etapa de diseño físico “optimizamos” el diseño lógico dependiendo de las herramientas que nos proporciona el SGBD.
- Por ejemplo, podemos hacer una tabla intermedia para relaciones muy altas, crear índices secundarios, añadir una columna a una base de datos con cálculos intermedios para acelerar el acceso a los datos, etc.
- Dentro del diseño físico seleccionamos las estructuras físicas de implementación, tamaño de buffers o memorias intermedias, entre otros factores.

PHP y SQLite

Etapa del diseño conceptual



Etapa del diseño conceptual

- El modelo más usado en esta etapa es el de Entidad-Relación o ER.
- En este modelo tendremos las entidades (donde se almacenan de manera lógica la información) y la relación entre las diversas entidades.
- Los cimientos del modelo ER se le atribuyen al Dr. Peter Chen en 1976.



Fuente: wikipedia

Francisco Arce
www.pacoarce.com

Etapa del diseño conceptual

- Las partes básicas de un modelo ER son:
 - Entidades
 - Atributos
 - Relaciones

PHP y SQLite

Entidades



Etapa del diseño conceptual

- Una “entidad” es una parte diferenciada del mundo físico, la cual posee información que necesitamos para nuestro sistema.
- Ejemplo de entidades:
 - Sistema comercio electrónico: artículos, clientes, pedidos, facturas.
 - Sistema escolar: alumnos, profesores, materias, salones.
 - Sistema de almacén: artículos, facturas, entradas, salidas, clientes, proveedores.
 - Biblioteca: libros, autores, editoriales, lectores.
 - Sistema contable: cuentas, movimientos, usuarios, préstamos, ingresos.
 - Sistema médico: Pacientes, doctores, medicinas, recetas, facturas.

Etapa del diseño conceptual

- Una entidad en realidad es un conjunto de “objetos” similares, por ejemplo, la entidad “alumnos” contará con la información de muchos individuos pertenecientes a una escuela.
- La entidad “facturas” será la colección de una o muchas facturas en un sistema de almacenaje.

PHP y SQLite

Atributos



Etapa del diseño conceptual

- Los atributos son partes de la información (o propiedades) de la entidad.
- Hay propiedades que entran al modelo (nos interesan) y hay propiedades o atributos que no nos interesan.
- Por ejemplo de una entidad alumno:
 - Matrícula
 - Nombre
 - Edad
 - Género
 - Grado
 - Salón

Etapa del diseño conceptual

- Las entidades se representan por un rectángulo, y los atributos o propiedades se escriben dentro de él.



PHP y SQLite

Llaves



Etapa del diseño conceptual

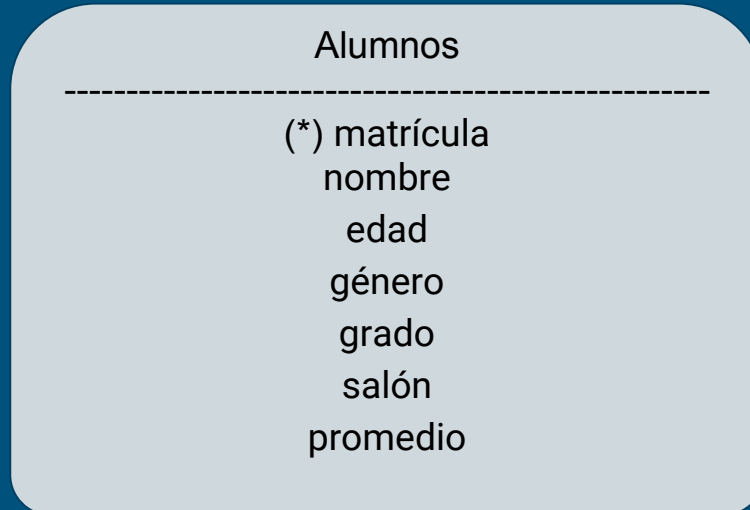
- Un atributo o propiedad lo consideraremos como “llave” si es diferente para cada elementos de la entidad.
- Un atributo candidatos a ser llave en una entidad “alumno” será la matrícula, puesto son diferentes para cada alumno y no se repiten.
- El atributo “nombre” no es candidato, ya que suelen duplicarse, incluso nombre y apellido.
- Para la entidad “paciente” será un “atributo candidato” a ser llave su número de seguridad social.

Etapa del diseño conceptual

- En una entidad puede haber más de un atributo a ser candidato a ser llave, por ejemplo, en un sistema de control automotriz se pueden considerar a las placas del vehículo, al número de motor o en algunos países existe un número de control vehicular.
- El diseñador de la base de datos seleccionará una propiedad o atributo candidato como “llave principal”.
- Por lo tanto, una llave principal es una propiedades que distingue a un objeto de una entidad en forma unívoca, y por lo tanto, no se repite.

Etapa del diseño conceptual

- Dentro del modelo ER, distinguimos a la llave principal subrayándola, por medio de un asterisco, un ícono de llave, etc.



PHP y SQLite

Relaciones



Etapa del diseño conceptual

- En el modelo ER, una relación entre entidades se representa por medio de un rombo.
- En algunas versiones del modelo ER sólo se representa con una línea entre las entidades.

Etapa del diseño conceptual



PHP y SQLite

Cardinalidad



Etapa del diseño conceptual

Existen tres tipos de cardinalidad:

Relación uno a uno (1:1)

Relación uno a muchos (1:N)

Relación muchos a muchos (M:N)

PHP y SQLite

Relación 1 a 1



Etapa del diseño conceptual

- Las relaciones uno a uno (1:1) tiene una permanencia fuerte, por ejemplo, una mascota sólo tiene un dueño, y un dueño sólo tiene una mascota.
- Una computadora sólo tiene un usuario, y un usuario sólo tiene una computadora.
- Son raras este tipo de relaciones: por lo general una persona puede tener muchas mascotas, y un usuario puede tener más de una computadora.

Etapa del diseño conceptual



PHP y SQLite

Relación 1 a muchos



Etapa del diseño conceptual

Las relaciones uno a muchos (1:N) son más comunes. A un elemento de la entidad “A” tiene muchos elementos de la entidad “B”.

Por ejemplo:

- Un dueño tiene muchas mascotas.
- Una escuela tiene muchos alumnos.
- Un jefe tiene muchos subalternos.
- Un doctor tiene muchos pacientes.

Etapa del diseño conceptual



PHP y SQLite

Relación muchos a muchos



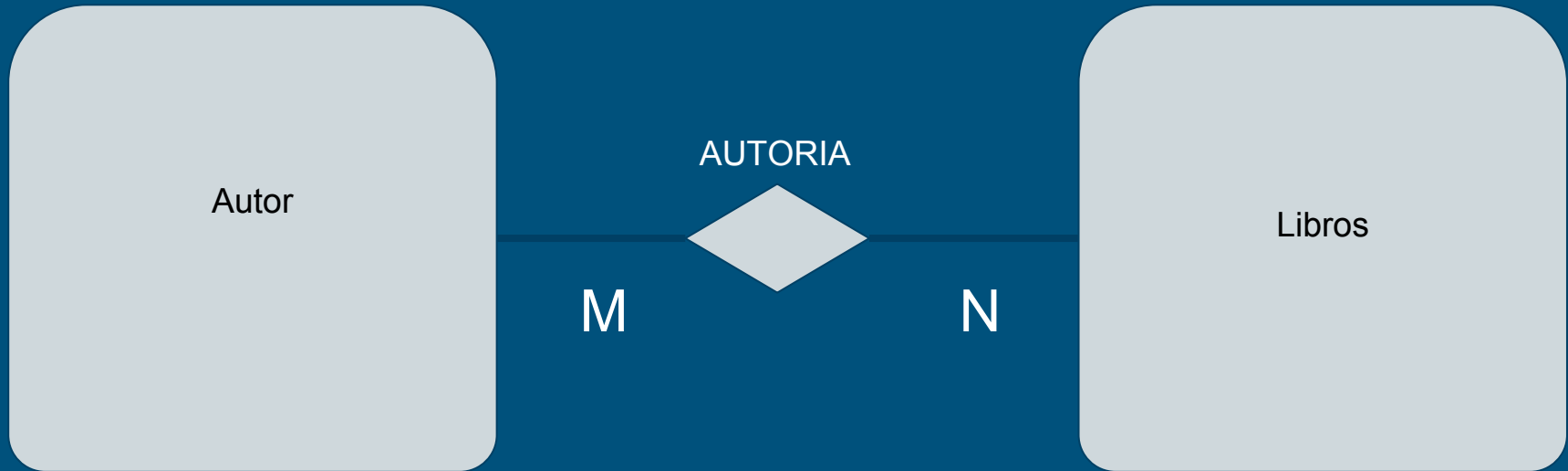
Etapa del diseño conceptual

Las relaciones muchos a muchos (M:N) son más complejas. A un elemento de la entidad "A" tiene muchos elementos de la entidad "B", pero un elemento de la entidad "B" tiene muchos elementos de la entidad "A".

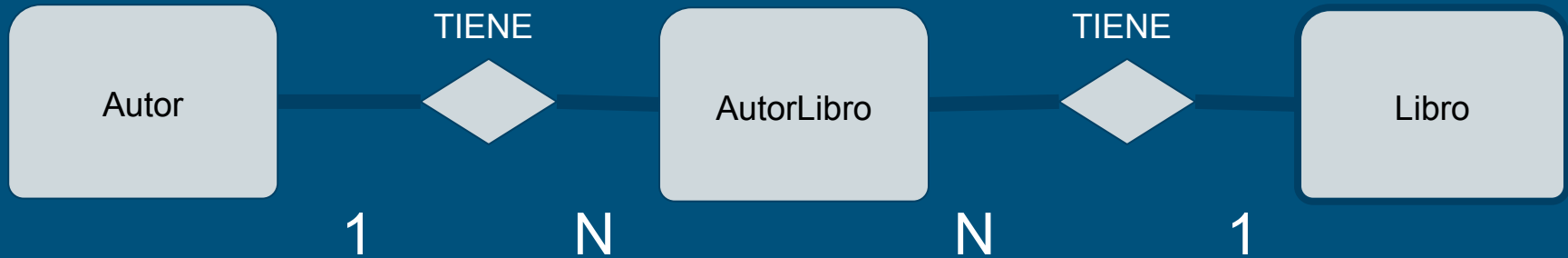
Por ejemplo:

- Un estudiante tiene muchas materias y una materia tiene muchos estudiantes inscritos.
- Una estudiante tiene muchos salones y un salón tiene muchos alumnos.
- Una entrada de almacén tiene muchos artículos y un artículo tiene muchas entradas de almacén.

Etapa del diseño conceptual



Etapa del diseño conceptual



PHP y SQLite

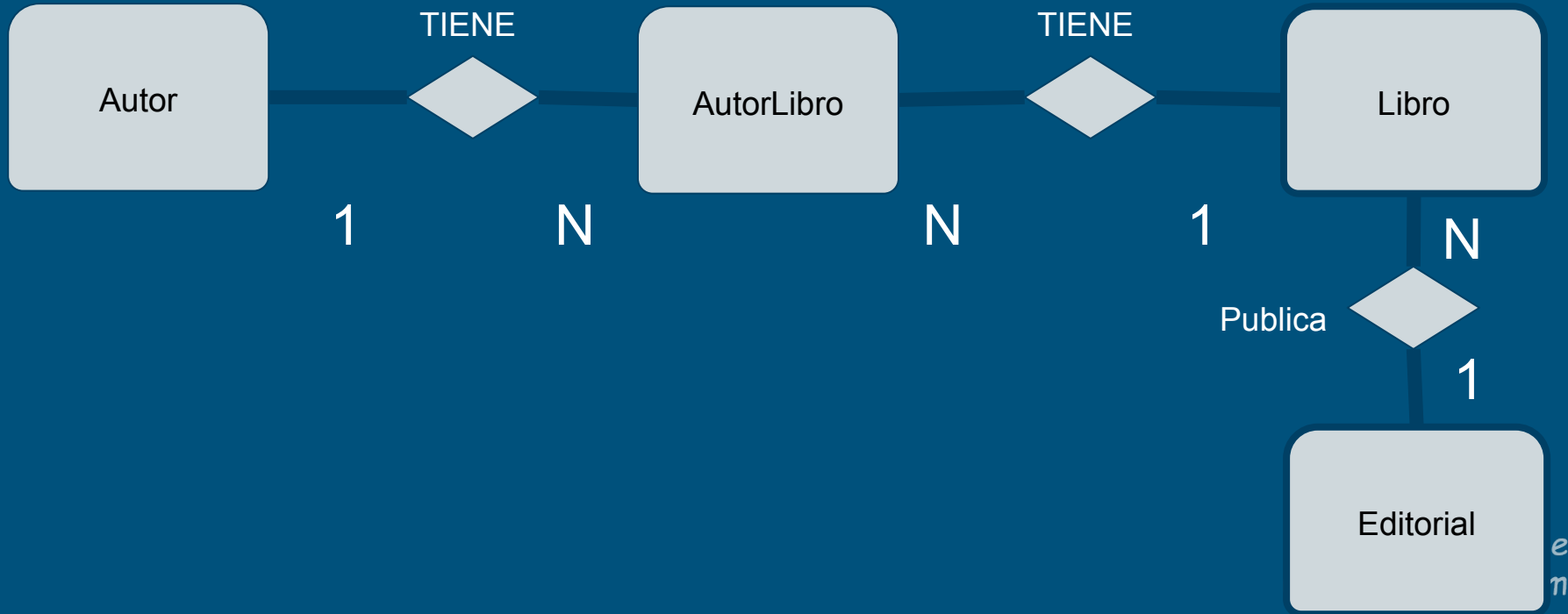
Llaves foráneas



Etapa del diseño conceptual

- Una “llave foránea” es un campo en la tabla que relaciona a la llave primaria de otra tabla.

Etapa del diseño conceptual



PHP y SQLite

Diseño lógico



Etapa del diseño conceptual

Autores

idAutor
Nombre
Nacionalidad
Nacimiento
Novel

AutoresLibros

id
idAutor
idLibro

Libro

idLibro
Titulo
Paginas
Fecha
idEditorial

Editorial

idEditorial
Nombre
País

PHP y SQLite

Normalización de bases de datos



Etapa del diseño conceptual

- La normalización de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Primera Forma Normal (1FN)

- Las bases de datos relacionales se normalizan para:
 - Evitar la redundancia de los datos.
 - Disminuir problemas de actualización de los datos en las tablas.
 - Proteger la integridad de los datos.

Etapa del diseño conceptual

- En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:
 - Cada tabla debe tener su nombre único.
 - No puede haber dos filas iguales. No se permiten los duplicados.
 - Todos los datos en una columna deben ser del mismo tipo.

Primera Forma Normal (1FN)

Una tabla está en Primera Forma Normal si:

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.
- La tabla contiene una clave primaria única.
- La clave primaria no contiene atributos nulos.
- No debe existir variación en el número de columnas.
- Los Campos no clave deben identificarse por la clave (Dependencia Funcional)

Primera Forma Normal (1FN)

Una tabla está en Primera Forma Normal si:

- Debe Existir una independencia del orden tanto de las filas como de las columnas, es decir, si los datos cambian de orden no deben cambiar sus significados
- Esta forma normal elimina los valores repetidos dentro de una Base de Datos.

PHP y SQLite

Normalización de bases de datos



Segunda Forma Normal (2FN)

Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales. (Todos los atributos que no son clave principal deben depender únicamente de la clave principal).

Segunda Forma Normal (2FN)

En otras palabras podríamos decir que la segunda forma normal está basada en el concepto de dependencia completamente funcional. Una dependencia funcional $\{x \rightarrow y\}$ es completamente funcional si al eliminar los atributos A de X significa que la dependencia no es mantenida, esto es que $\{A \in X, X - \{A\} \nrightarrow Y\}$. Una dependencia funcional $\{x \rightarrow y\}$ es una dependencia parcial si hay algunos atributos $\{A \in X\}$ que pueden ser eliminados de X y la dependencia todavía se mantiene, esto es $\{A \in X, X - \{A\} \rightarrow Y\}$.

Segunda Forma Normal (2FN)

Por ejemplo $\{DNI, ID_PROYECTO\} \rightarrow HORAS_TRABAJO$ (con el DNI de un empleado y el ID de un proyecto sabemos cuántas horas de trabajo por semana trabaja un empleado en dicho proyecto) es completamente funcional dado que ni $DNI \rightarrow HORAS_TRABAJO$ ni $ID_PROYECTO \rightarrow HORAS_TRABAJO$ mantienen la dependencia. Sin embargo $\{DNI, ID_PROYECTO\} \rightarrow NOMBRE_EMPLEADO$ es parcialmente dependiente dado que $DNI \rightarrow NOMBRE_EMPLEADO$ mantiene la dependencia.

PHP y SQLite

Normalización de bases de datos



Tercera Forma Normal (3FN)

La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

Un ejemplo de este concepto sería que, una dependencia funcional $X \rightarrow Y$ en un esquema de relación R es una dependencia transitiva si hay un conjunto de atributos Z que no es un subconjunto de alguna clave de R , donde se mantiene $X \rightarrow Z$ y $Z \rightarrow Y$.

Tercera Forma Normal (3FN)

Por ejemplo, la dependencia $SSN \rightarrow DMGRSSN$ es una dependencia transitiva en EMP_DEPT de la siguiente figura. Decimos que la dependencia de DMGRSSN el atributo clave SSN es transitiva vía DNUMBER porque las dependencias $SSN \rightarrow DNUMBER$ y $DNUMBER \rightarrow DMGRSSN$ son mantenidas, y DNUMBER no es un subconjunto de la clave de EMP_DEPT. Intuitivamente, podemos ver que la dependencia de DMGRSSN sobre DNUMBER es indeseable en EMP_DEPT dado que DNUMBER no es una clave de EMP_DEPT.

Formalmente, un esquema de relación $\{R\}$ está en 3 Forma Normal Elmasri-Navathe,² si para toda dependencia funcional $X \rightarrow A$, se cumple al menos una de las siguientes condiciones:

Tercera Forma Normal (3FN)

Formalmente, un esquema de relación R está en 3 Forma Normal Elmasri-Navathe,² si para toda dependencia funcional $X \rightarrow A$, se cumple al menos una de las siguientes condiciones:

X es superllave o clave.

A es atributo primo de R ; esto es, si es miembro de alguna clave en R .

Además el esquema debe cumplir necesariamente, con las condiciones de segunda forma normal.

PHP y SQLite

Diseño lógico



Etapa del diseño conceptual

https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

http://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm

http://www.tutorialspoint.com/dbms/er_diagram_representation.htm

https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n