

Índice

INTRODUCCIÓN.

1. DISEÑO LÓGICO DE DATOS.

2. TRANSFORMACIÓN DEL MODELO CONCEPTUAL A MODELOS LÓGICOS.

2.1. Transformación del modelo E/R al modelo relacional.

2.2. Transformación del modelo E/R al modelo jerárquico.

3. ANÁLISIS RELACIONAL DE DATOS.

3.1. Formas normales.

3.2. Dependencias funcionales.

3.3. Definición formal de las formas normales.

3.4. Descomposición de relaciones.

3.5. Consideraciones finales sobre la teoría de la normalización.

4. DOCUMENTACIÓN.

4.1. Grafo relacional.

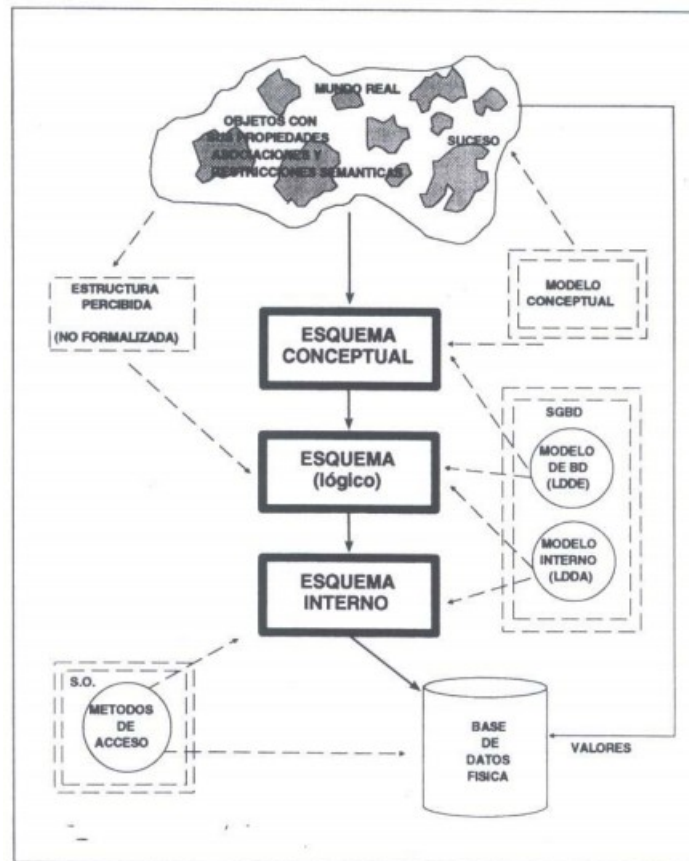
4.2. Grafo de dependencias funcionales.

CONCLUSIÓN.

INTRODUCCIÓN.

Los modelos de datos son un eficaz instrumento en el diseño de bases de datos.

En el diseño de bases de datos, se distingue la fase de modelado conceptual, que es la descripción del mundo real de acuerdo con un modelo altamente semántico e independiente del SGBD en el que posteriormente se vaya a hacer la implementación de la base de datos, y la fase de diseño lógico, en la cual se ha de obtener un esquema que responda a la estructura lógica específica del SGBD que se aplique en cada caso, por lo que dicho esquema está sometido a las restricciones que imponga el modelo del SGBD en concreto.



Detalle de la transformación del mundo real a la BD física

Con la ayuda del modelo conceptual, se obtiene el esquema conceptual, formalmente se suele representar mediante un modelo entidad interrelación (E-R).

Aplicando al esquema conceptual las reglas del modelo de datos propio del SGBD que se va a utilizar, se obtiene el esquema lógico (esquema de base de datos); de éste se pasa al esquema interno. Por último, se implementa la base de datos física en los soportes secundarios.

Las herramientas CASE proporcionan una importante ayuda en el diseño de bases de datos, al disponer de modelos de datos semánticos (en general basados en el modelo E/R, con algunas extensiones) que facilitan el diseño conceptual y realizan la transformación al modelo lógico relacional propio de los productos comerciales más extendidos.

1. DISEÑO LÓGICO DE DATOS.

Yourdon presenta tres perspectivas diferentes para analizar un sistema siguiendo las técnicas del análisis estructurado:

- **Función:** qué hace el sistema.
- **Información:** qué información utiliza el sistema.
- **Tiempo:** cuándo sucede algo en el sistema.

Es recomendable examinar un sistema bajo estas tres visiones. Para cada perspectiva hay un conjunto de técnicas que se utilizan con frecuencia:

Dimensión de la función: el diagrama de flujo de datos se utiliza para mostrar las funciones del sistema y sus interfaces.

Dimensión de la información: el diagrama entidad/interrelación se utiliza para señalar las entidades y las relaciones entre ellas.

Dimensión del tiempo: la lista de eventos se utiliza para mostrar cualquier cosa que ocurra y sobre la que el sistema debe responder.



Además de estudiar cada dimensión, es interesante ver las relaciones que existen entre ellas, se puede observar la existencia de tres planos entre las principales dimensiones: información-tiempo, información-función y función-tiempo.

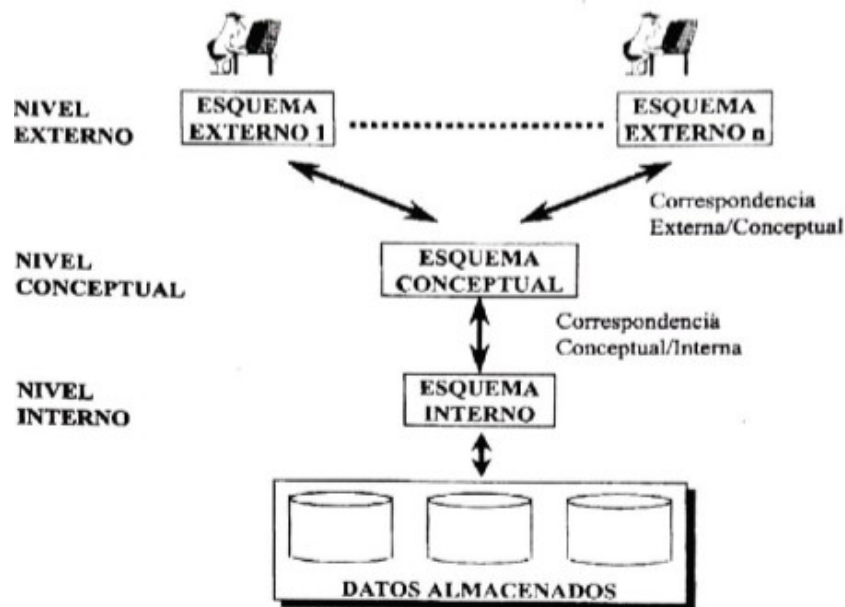
Existen técnicas definidas en estos planos (principalmente matriciales) que describen la relación entre los modelos y, por lo tanto, permiten hacer comprobaciones de consistencia.

En la dimensión de la información se pueden distinguir varios niveles:

Esquema externo. Es la visión que de la base de datos tiene un usuario en particular.

Esquema conceptual. Visión global de los datos.

Esquema interno. Es la forma en que se organizan los datos en el almacenamiento físico.



A nivel conceptual, existen dos tipos de modelos de datos:

Los **modelos de alto nivel**, que facilitan la descripción del conjunto de información de la empresa con independencia de la máquina (tanto del hardware como del SGBD concreto), por lo que sus conceptos son cercanos al mundo real (entidades, atributos, interrelaciones, etc.); son modelos de análisis, y no de implementación. Entre estos modelos podemos citar el entidad/interrelación.

Los **modelos lógicos**, que se encuentran soportados por los SGBD (de ahí que reciban también el nombre de modelos de bases de datos), por lo que sus conceptos son propios de cada SGBD.

En el proceso de diseño de una base de datos, se distinguen tres grandes fases:

Diseño conceptual: cuyo objetivo es obtener una buena representación de los recursos de información de la empresa.

Diseño lógico: transforma el esquema conceptual adaptándolo al modelo lógico de datos en el que se apoya el SGBD que se va a utilizar.

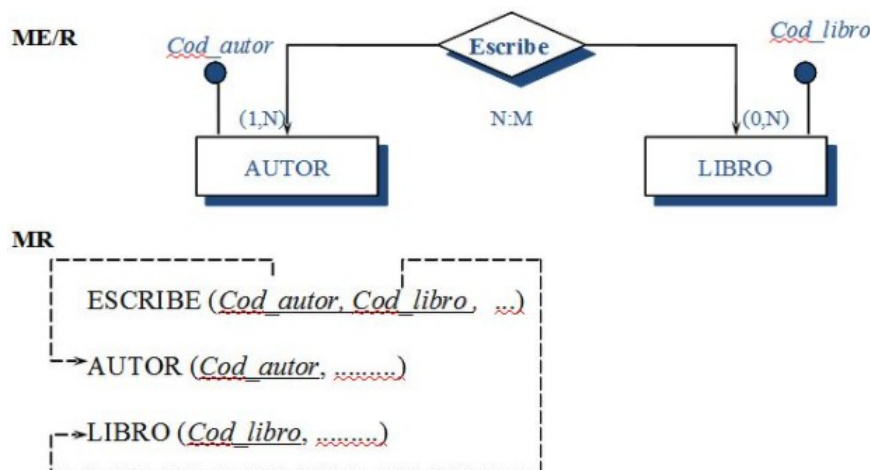
Diseño físico: construcción del esquema lógico.

2. TRANSFORMACIÓN DEL MODELO CONCEPTUAL A MODELOS LÓGICOS.

2.1. Transformación del modelo E/R al modelo relacional.

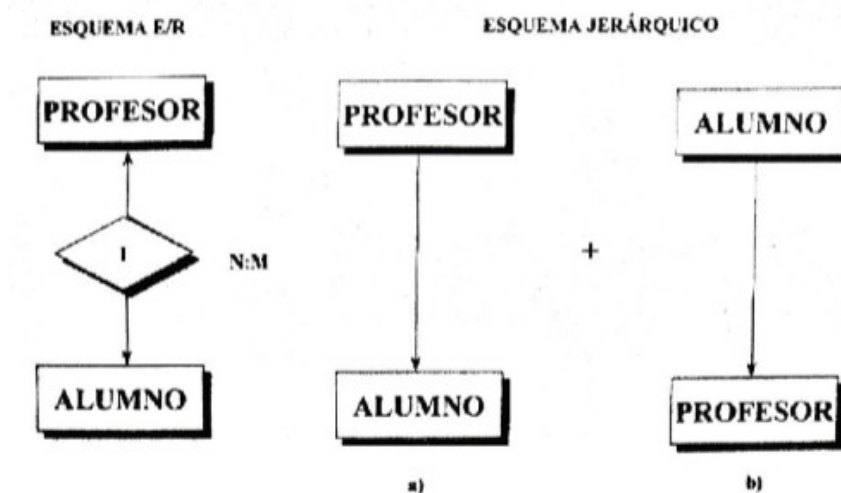
El paso de un esquema en el modelo E/R al relacional está basado en los tres principios siguientes:

- Todo tipo de entidad se convierte en una relación.
- Todo tipo de interrelación N:M se transforma en una relación.
- Todo tipo de interrelación 1:N se traduce en el fenómeno de propagación de clave del lado de entidad 1 al lado N o bien se crea una nueva relación.



2.2. Transformación del modelo E/R al modelo jerárquico.

El modelado directo del mundo real según esquemas jerárquicos está plagado de inconvenientes; hay técnicas que incluso sugieren la introducción de redundancias, con los problemas asociados de falta de control de la integridad de la base de datos y de pérdida de simetrías.



3. ANÁLISIS RELACIONAL DE DATOS.

El diseño de una base de datos relacional se puede realizar aplicando al mundo real, en una primera fase, un modelo semántico como el E/R, a fin de obtener un esquema conceptual; en una segunda fase, se transforma dicho esquema al modelo relacional mediante las correspondientes reglas de transformación. Si bien este enfoque es el más recomendable, existe otra posibilidad que es plasmar directamente en el modelo relacional nuestra percepción del mundo real, obteniendo el esquema relacional sin realizar ese paso intermedio que es el esquema conceptual. En esto consiste el análisis relacional.

Siempre es conveniente aplicar un conjunto de reglas, conocidas como teoría de la normalización, que permiten asegurar que un esquema relacional cumple unas ciertas propiedades.

La teoría de la normalización evita las redundancias y las anomalías de utilización, obteniendo relaciones más estructuradas que no presenten inconsistencias ni pérdidas de información.

3.1. Formas normales.

La teoría de la normalización se centra en lo que se conoce como formas normales. Se dice que un esquema de relación está en una determinada forma normal si satisface un conjunto específico de restricciones.

La teoría de la normalización se basa en ciertas restricciones definidas sobre los atributos de una relación, las cuales son conocidas con el nombre de dependencias. Existen varios tipos de dependencias.

3.2. Dependencias funcionales.

- **Dependencia funcional:** $A \rightarrow B$, representa que B es funcionalmente dependiente de A. Para un valor de A siempre aparece un valor de B.

Ejemplo: Si A es el D.N.I., y B el Nombre, está claro que para un número de D.N.I., siempre aparece el mismo nombre de titular.

- **Dependencia funcional completa:** $A \rightarrow B$, si B depende de A en su totalidad. Tiene sentido plantearse este tipo de dependencia cuando A está compuesto por más de un atributo.

Ejemplo: Supongamos que A corresponde al atributo compuesto: D.N.I._Empleado + Cod._Dpto. y B es Nombre_Dpto. En este caso B depende del Cod_Dpto., pero no del D.N.I._Empleado. Por tanto no habría dependencia funcional completa.

- **Dependencia transitiva:** $A \rightarrow B \rightarrow C$. Si $A \rightarrow B$ y $B \rightarrow C$, Entonces decimos que C depende de forma transitiva de A.

Ejemplo: Sea A el D.N.I. de un alumno, B la localidad en la que vive y C la provincia. Es un caso de dependencia transitiva $A \rightarrow B \rightarrow C$.

- **Determinante funcional:** todo atributo, o conjunto de ellos, de los que depende algún otro atributo.

Ejemplo: El D.N.I. es un determinante funcional pues atributos como nombre, dirección, localidad, etc, dependen de él.

- **Dependencia multievaluada:** $A \rightarrow \rightarrow B$. Son un tipo de dependencias en las que un determinante funcional no implica un único valor, sino un conjunto de ellos. Un valor de A siempre implica varios valores de B.

Ejemplo: CursoBachillerato $\rightarrow \rightarrow$ Modalidad. Para primer curso siempre va a aparecer en el campo Modalidad uno de los siguientes valores: Ciencias, Humanidades/Ciencias Sociales o Artes. Igual para segundo curso.

3.3. Definición formal de las formas normales.

a) Primera forma normal (1FN).

Se dice que una relación está en 1FN si cada atributo toma un único valor del dominio subyacente.

b) Segunda forma normal (2FN).

Se dice que una relación está en 2FN si:

- Está en 1FN.
- Cada atributo no principal tiene dependencia funcional completa respecto de cada una de las claves.

c) Tercera forma normal (3FN).

Se dice que una relación está en 3FN si:

- Está en 2FN.
- No existe ningún atributo no principal que dependa transitivamente de algo de las claves de relación.

d) Forma normal de Boyce y Codd (FNBC)

Una relación se encuentra en FNBC si, y sólo si, todo determinante es clave.

3.4. Descomposición de relaciones.

La transformación de una relación que se encuentra en una determinada forma normal, en otra relación cuya forma normal es superior se realiza por medio del operador de proyección del álgebra relacional.

3.5. Consideraciones finales sobre la teoría de la normalización.

La teoría de la normalización nos ayuda a estructurar mejor las relaciones, teniendo dos posibles redundancias y anomalías, y a representar mejor nuestro mundo real en un esquema relacional.

4. DOCUMENTACIÓN.

Como resultado de la fase de diseño lógico de datos, se obtienen una serie de grafos con la información del modelo lógico. Los más habituales son:

4.1. Grafo relacional.

Una forma sencilla de representar el esquema relacional es el denominado grafo relacional. Es un grafo compuesto de un conjunto de nodos multiparticionados, donde cada nodo representa un esquema de relación, es decir, una tabla de la base de datos.

4.2. Grafo de dependencias funcionales.

Una herramienta muy útil a la hora de explicitar las dependencias funcionales es el grafo o diagrama de dependencias funcionales, mediante el cual se representa un conjunto de atributos y las dependencias funcionales existentes entre ellos. En el grafo aparecen los nombres de los atributos unidos por flechas, las cuales indican las dependencias funcionales.

CONCLUSIÓN.

Bajo el paraguas de la denominación "herramientas CASE" (Computer Aided Software Engineering) hay todo un abanico de utilidades que ayudan de forma automática o semiautomática en todas o algunas de las etapas del ciclo de vida del software. Ejemplos de tales herramientas son Oracle Designer, Visual CASE, etc..

Entre las características distintivas de este tipo de herramientas cabe destacar, la posibilidad de describir el modelo conceptual (UML, entidad/interrelación E/R), según una notación específica y generar directamente un modelo lógico (relacional, objeto-relacional).