

Tema 5: Diseño lógico de datos

1. Diseño lógico de datos.
 2. Transformación del modelo conceptual al lógico.
 3. Análisis relacional de datos.
 4. Documentación.
1. Diseño lógico de datos.

Hoy en día, prácticamente todos los sistemas de información almacenan y organizan los datos en DDBB. Para llevar a cabo la implementación de la DDBB que necesita el sistema habrá que tener en cuenta todas las fases de diseño de esta:

- Diseño conceptual: Este diseño es independiente del modelo de DDBB usado, del ordenador, del sistema gestor de bases de datos, etc. Simplemente se estudia el problema y se seleccionan los elementos del mundo real que vamos a modelar. Este diseño es al que corresponde el diagrama E/R
- Diseño lógico: Partiendo del diseño conceptual obtenido en la fase anterior, llegamos a un diseño lógico. Transformamos las entidades y relaciones obtenidas del modelo anterior en tablas. Para ello usamos la normalización.
- Diseño físico: Este diseño si depende del ordenador, del sistema gestor de DDBB, etc. En este caso, empleando el gestor de la DDBB, se implementan las tablas de las DDBB con sus características, organización y estructuras de almacenamiento interno.

Para evitar la gran dependencia que existía antes entre los ficheros y las aplicaciones que los utilizaban (cualquier cambio en la estructura física o lógica de los datos afectaba a las aplicaciones), el instituto ANSI publicó un informe en el que definía una arquitectura de tres niveles para ser utilizada en el diseño de DDBB, con objeto de minorizar el impacto producido por los cambios haciendo énfasis en la independencia que debe existir entre las referencias externas a los datos y la forma física de almacenamiento y organización de los mismos. Los tres niveles definidos son:

- Nivel externo: Constituye un nivel con el que interactúa el usuario. Este nivel representa una visión parcial de los datos, de manera que usuarios diferentes tendrán una visión distinta de los mismos, mostrando solo aquella parte que interesa al usuario.
- Nivel conceptual: Este nivel representa el esquema lógico de los datos, reflejando su estructura y relaciones, sin entrar en detalles físicos. Este nivel se construye mediante un modelo en el que se define en primer lugar aquella parte del mundo real que deseamos modelar, excluyendo los datos que no son necesarios. En este punto debemos decidir que modelo lógico se va a utilizar, existiendo varias alternativas como puede ser el modelo relacional, el jerárquico, orientado a objetos, etc
- Nivel físico: Este nivel debe ser transparente para el usuario. En este nivel se especifica la estructura de los datos así como el modo de almacenamiento empleado. Este apartado va a depender de varios factores tanto HW como Software, entre los que se puede señalar: S.O., Sistema de ficheros del sistema gestor de bases de datos, Unidades de almacenamiento externos, etc

2. Transformación del modelo conceptual al lógico.

El diseño de las DDBB del sistema se llevara a cabo aplicando la arquitectura ANSI de tres niveles, por tanto debemos partir del modelo conceptual y llegar hasta el esquema físico o interno.

El esquema conceptual representa los recursos del sistema y se define sin tener en consideración cuestiones físicas.

Para la definición de este esquema nos podemos ayudar de herramientas de modelado como los diagramas.

El modelo E/R (entidad relación) fue propuesto por Chen y posteriormente algunas aportaciones de han dado lugar E/R extendido.

Los componentes del modelo E/R son:

- Entidades: representan un objeto real o abstracto sobre el que queremos almacenar información.
- Relación: define una asociación entre entidades.
- Grado de una relación: número de entidades que participan en una relación, pudiendo ser reflexivas (una entidad se relaciona con ella misma), binaria (participan 2 entidades) y n-aria (participan n entidades).
- Cardinalidad: define el número máximo de ocurrencias de una entidad que participan en una relación. Puede ser de uno a uno, de uno a muchos y de muchos a muchos.
- Atributos: representan propiedades o características de una entidad o relación.

Dentro del modelo E/R extendido aparecen además otros conceptos:

- Jerarquía: una entidad puede mantener una relación de supertipo con otras entidades. Es el caso de la generalización y especialización.
- Agregación: conversión de una relación junto con sus entidades participantes, en una entidad para poder relacionarse con otra entidad.
- Exclusividad: es un tipo especial de relación en la que una entidad se asocia con varias entidades. La exclusividad relaciona una entidad con otra de entre varias posibles.

Una vez obtenido el modelo conceptual (representado en el diagrama E/R) debe ser transformado a un modelo lógico. La secuencia de pasos a aplicar para dicha transformación son:

1. Cada entidad se transforma en una tabla y los atributos de dicha entidad en atributos de la tabla.

2. Las relaciones de muchos a muchos se transforman en tablas cuya clave estará formada por la clave primaria de las entidades relacionadas. Las relaciones de uno a muchos propagan la clave principal de la entidad cuya cardinalidad es uno a la entidad de cardinalidad n.

Otra herramienta empleada para el diseño lógico de datos es el diagrama de estructura de datos (DED), en la que se establecen los diferentes registros que forman la base de datos y las relaciones entre ellos. La forma tridente apunta a la entidad que actúa como muchos.

Este tipo de esquema solo admite relaciones entre dos entidades, en caso de modelar relaciones en las que

intervengan más de dos entidades debemos redefinir el esquema reduciéndolo a relaciones binarias.

Se puede establecer una correspondencia entre diagramas E/R y diagramas de estructura de datos, pudiendo pasar de un tipo de diagrama al otro.

3. Análisis relacional de datos.

El análisis relacional de datos se centra en el estudio de la características del modelo E/R y de la estructura del modelo lógico (E/R tabla).

El diseño de bases de datos relacionales implica la creación de un conjunto de tablas relacionadas entre si. El modelo así obtenido puede presentar una serie de problemas como son la redundancia de datos y ambigüedades a medida que trabajamos con ellas. Debido a este problema, se ha definido una teoría que indica una serie de restricciones que hemos de aplicar sobre el modelo para así obtener un modelo normalizado, que evita los problemas planteados.

Para poder realizar el proceso de normalización hay que tener claros algunos conceptos:

- Se dice que atributo es atómico cuando no puede tener más de un valor, en caso contrario, cuando un atributo puede tener varios valores se dice que es un atributo multivaluado.

- Las claves son atributos que identifican de forma unívoca a las entidades. Existen varios tipos de clave:

1. Superclave: estará formada por uno o más atributos que identificarán de forma única a una entidad.

2. Clave candidata: es una superclave mínima, es decir, una superclave que si se le quita un atributo deja de ser superclave.

3. Clave primaria: es la clave candidata que el diseñador de la base de datos ha elegido para diferenciar las entidades.

- Otro concepto que hay que tener en cuenta es el de dependencia funcional. Existen varios tipos:

1. Dependencia funcional: dado un conjunto B decimos que dicho conjunto depende funcionalmente de otro conjunto A si para cualquier valor de A le corresponde un único valor de B. Se denota $A \rightarrow B$.

2. Dependencia funcional completa: decimos que un conjunto B tiene dependencia funcional completa respecto a otro conjunto A, si depende de dicho conjunto en su totalidad y no de una de sus partes. Se denota $A \twoheadrightarrow B$.

3. Dependencia transitiva: se dice que un conjunto B depende de forma transitiva de otro conjunto A, si existe un conjunto Z que depende funcionalmente de A y B depende funcionalmente de Z. Se denota $A \twoheadrightarrow Z \twoheadrightarrow B$.

Las principales formas normales de la teoría de normalización son:

- Primera forma normal (1FN): una tabla está en 1FN si todos los atributos no clave, dependen funcionalmente de la clave (es decir, si dada una clave se puede obtener el valor de todos sus atributos), o lo que es lo mismo, no existen grupos repetitivos para un valor de clave.

Pasos a seguir:

- A. Se crea a partir de la tabla inicial una nueva tabla con los atributos que dependen funcionalmente de la

clave (Que tendrán la misma clave que la tabla inicial). Esta tabla ya esta en primera forma normal.

B. Se crea una nueva tabla con los atributos restantes, eligiendo de entre estos uno como clave de la tabla (o mas de uno). Los criterios de elección de clave serán los mismos que se expusieron para los tipos de clave.

C. Se comprueba si esta tabla esta en primera forma normal. Si es así, la tabla inicial ya esta normalizada y finaliza el proceso. Si no, tomamos esta tabla como tabla inicial y volvemos a realizar el apartado A.

– Segunda Forma Normal (2FN): Una tabla esta en segunda forma normal si esta en primera forma normal y además todos los atributos que no pertenecen a la clave dependen funcionalmente de forma completa de ella. De esta definición se desprende que de una tabla en primera forma normal y cuya clave esta compuesta por un único atributo esta en segunda forma normal.

El proceso de normalización es como sigue:

A. Se crea a partir de la tabla inicial una nueva tabla con los atributos que dependen funcionalmente de forma completa de la clave (Que tendrán la misma clave que la tabla inicial) . Esta tabla ya esta en segunda forma normal.

B. Se crea una nueva tabla con los atributos restantes, siendo su clave el subconjunto de atributos de la clave inicial de los que dependen de forma completa.

C. Se comprueba si esta tabla esta en segunda forma normal. Si es así, la tabla inicial ya esta normalizada y finaliza el proceso. Si no, tomamos esta tabla como tabla inicial y volvemos a realizar el apartado A.

– Una tabla esta en tercera forma normal esta en tercera forma normal y además no existen atributos no claves que dependan transitivamente de la clave, es decir, no debe haber atributos no clave que dependan de otros atributos no primos (que no pertenecen a la clave).

El proceso de normalización es como sigue:

A. Se crea a partir de la tabla inicial una nueva tabla con los atributos que no poseen dependencias transitivas .(Que tendrán la misma clave que la tabla inicial) . Esta tabla ya esta en segunda forma normal.

B. Se crea una nueva tabla con los atributos restantes, siendo su clave el subconjunto de atributos de la clave inicial de los que dependen de forma completa.

C. Se crea una nueva tabla con los dos atributos no clave, que intervienen en la dependencia transitiva, seleccionando entre ambos a aquel que cumpla los requerimientos de clave. Esta nueva tabla esta ya en tercera forma normal

Existen mas formas normales: Forma normal Óbice Codd (FNBC), (4FN), (5FN), . Pero algunos autores opinan que a partir de todas estas formas normales se puede producir perdida de dependencias

4. Documentación.

Durante el desarrollo de un proyecto software se genera un importante volumen de documentación en las diferentes fases que componen su análisis y su diseño. Durante la fase de diseño lógico de datos se generara información en forma de diagrama entidad relación y Diagrama de estructuras de datos según la metodología empleada. Dichos diagramas deben formar parte de la documentación del proyecto. Pero además de dichos diagramas necesitaremos alguna herramienta que sirva para recopilar los elementos de datos con los que trabajaremos en el proyecto, así como una descripción detallada de dichos elementos. La herramienta que nos

sirve para este propósito es el Diccionario de datos.

El diccionario de datos nos sirve para tomar nota de todos los elementos a los que hacemos referencia en los diagramas empleados para modelar el sistema que queremos construir. En el tomaremos nota de los datos, objetos, entidades, almacenes y elementos de control a los que hacemos referencia E/R, DED, DFD, DFC, .

(Mirar tema 3)

Ce Fini.