

Introducción al diseño de bases de datos

Jordi Casas Roma

PID_00213710



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació per a la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

Introducción.....	5
Objetivos.....	6
1. Proceso de diseño de una base de datos.....	7
2. Fases del diseño de una base de datos.....	10
2.1. Fase 1. Recogida y análisis de requisitos	10
2.1.1. Recogida de requisitos	10
2.1.2. Estructuración y refinamiento de los requisitos	11
2.1.3. Formalización de los requisitos	11
2.2. Fase 2. Diseño conceptual	12
2.2.1. El modelo ER	12
2.2.2. El lenguaje unificado de modelización	13
2.3. Fase 3. Diseño lógico	13
2.3.1. Reconsideraciones del modelo conceptual	14
2.3.2. Transformación del modelo conceptual en el modelo lógico	14
2.3.3. Normalización	14
2.4. Fase 4. Diseño físico	15
2.4.1. El nivel físico y el nivel virtual	15
2.4.2. Transformación del modelo lógico en el modelo físico	16
2.5. Fase 5. Implementación y optimización	16
2.5.1. Procesamiento y optimización de consultas	17
2.5.2. Procesamiento de vistas	18
2.5.3. Administración de la seguridad	19
Resumen.....	20
Glosario.....	21
Bibliografía.....	22

Introducción

El diseño de bases de datos es un proceso complejo que permite obtener una implementación de una base de datos a partir de los requisitos iniciales de los usuarios del sistema de información. Este proceso guía al diseñador de bases de datos por varias etapas con el objetivo de segmentar un problema de una complejidad considerable en diferentes subproblemas de menor complejidad.

Cada uno de los subproblemas identificados corresponde a una de las etapas del proceso de diseño de bases de datos. En estos materiales didácticos se describe el proceso global de diseño de bases de datos y las diferentes etapas que lo forman. Este módulo es solo un texto introductorio al diseño de bases de datos y será necesario profundizar en el estudio de cada una de sus etapas mediante los distintos módulos de la asignatura que corresponden a las diferentes etapas del proceso. Así pues, en este módulo se presenta una visión general de todo el proceso de diseño de bases de datos.

Objetivos

En estos materiales encontraréis las herramientas indispensables para lograr los objetivos siguientes:

- 1.** Entender en qué consiste el diseño de bases de datos y cuáles son sus objetivos.
- 2.** Conocer las distintas etapas que integran el proceso de diseño de una base de datos.

1. Proceso de diseño de una base de datos

El proceso de diseño de bases de datos consiste en definir la estructura lógica y física de una o más bases de datos para responder a las necesidades de los usuarios con respecto a la información y para un conjunto concreto de aplicaciones.

Mediante un proceso de diseño de bases de datos, se pueden decidir las tablas y relaciones que debe tener una base de datos determinada, los atributos de las diferentes tablas, las claves primarias y las claves foráneas que se deben declarar en cada tabla, etc. Todas estas tareas forman parte del proceso de diseño de bases de datos. Para poder tomar estas decisiones de la manera más correcta posible, hay que tener en cuenta las necesidades de información de los usuarios en relación con un conjunto concreto de aplicaciones.

Por lo tanto, el diseño de una base de datos es el proceso en el que se define la estructura de los datos que debe tener la base de datos de un sistema de información determinado.

Los requisitos que debe cumplir un sistema de información y la complejidad de la información que se presenta en él provocan que el diseño de una base de datos sea un proceso complicado. Para simplificar este proceso, es muy recomendable utilizar la estrategia de “divide y vencerás” (*divide and conquer*).

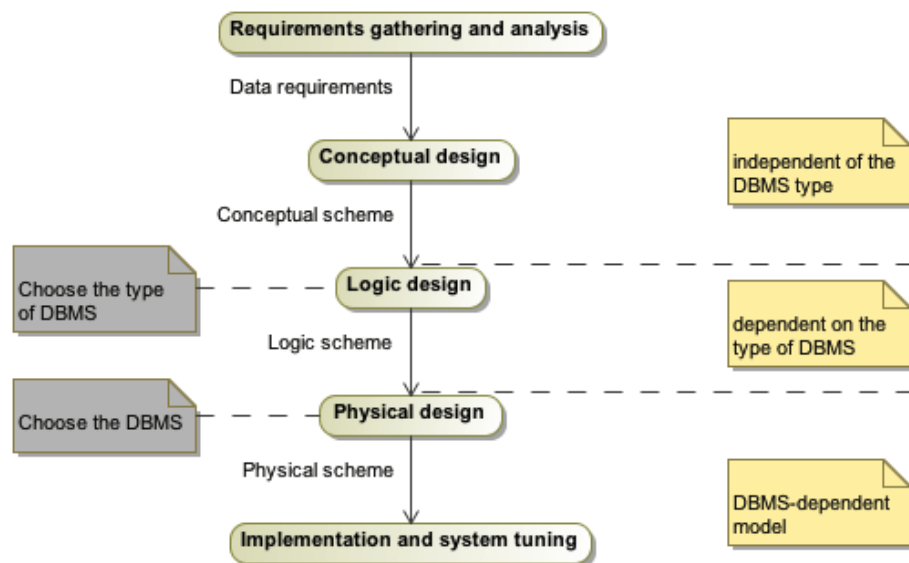
Si aplicamos este concepto, obtenemos las diferentes etapas del diseño de bases de datos. Estas etapas son secuenciales y el resultado de cada una sirve de punto de partida de la etapa siguiente. El resultado de la última etapa será el diseño final de nuestra base de datos. De este modo, un proceso de una cierta complejidad se descompone en diferentes procesos de menor complejidad. La figura 1 muestra las distintas etapas del diseño de bases de datos.

En primer lugar, tenemos la **recogida y análisis de requisitos**. Esta etapa debe permitir obtener los requisitos y las restricciones de los datos del problema. Para obtener esta información será necesario mantener conversaciones con los diferentes usuarios de la futura base de datos y de las aplicaciones que estén relacionadas con ésta. Sólo si se cruzan los requisitos de los diferentes perfiles de usuarios será posible establecer un marco completo de requisitos y las restricciones de los datos relacionados con la futura base de datos.

Dividir para vencer

La estrategia de “divide y vencerás” propone resolver un problema complejo mediante la subdivisión en un conjunto de problemas más sencillos donde la resolución de los diferentes subproblemas implica solucionar el problema inicial.

Figura 1. Etapas del diseño de bases de datos



Sistema gestor de bases de datos

Un sistema gestor de bases de datos (SGBD; en inglés *database management system*, DBMS) es un tipo de software específico que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

A continuación, se inicia el **diseño conceptual**. En esta etapa se crea un esquema conceptual de alto nivel a partir de las especificaciones y los requisitos obtenidos en la etapa anterior. En este proceso hay que extraer las necesidades y los requisitos de la problemática y sintetizarlos en un modelo visual de manera que permita representar los datos y las restricciones de los conceptos que se quieren modelizar en el sistema de información. Este modelo se denomina *esquema conceptual*.

Hasta esta etapa del diseño de bases de datos todavía no ha sido necesario elegir el tipo de bases de datos que se utilizará (relacional, orientada a objetos, documental, etc.) ni el sistema gestor de bases de datos (SGBD⁽¹⁾) que se utilizará o el lenguaje concreto con el que se implementará la base de datos.

⁽¹⁾ SGBD es la sigla de *sistema gestor de bases de datos*.

En el momento en el que se inicia la tercera etapa del proceso de diseño, el **diseño lógico**, hay que determinar el tipo de bases de datos que se utilizará. Es decir, no es necesario todavía escoger un SGBD concreto, pero sí el tipo de bases de datos que se quiere utilizar. En esta etapa el esquema conceptual se convierte en un esquema lógico adecuado al tipo de bases de datos que se pretende usar.

Por *tipos de bases de datos* entendemos los diferentes grupos de bases de datos según el modelo de datos que aplican. Actualmente hay varios tipos de SGBD, entre los cuales los más utilizados son las bases de datos relacionales, orientadas a objetos, documentales, geográficas o multidimensionales. Por ejemplo, las bases de datos relacionales son el conjunto de todos los SGBD que aplican modelos de datos relacionales.

En este punto, y antes de iniciar la etapa de **diseño físico**, hay que elegir un SGBD concreto sobre el que se pretende implementar la base de datos. La etapa de diseño físico adapta el esquema lógico a las necesidades específicas de un SGBD concreto y, posteriormente, ajusta algunos parámetros para el fun-

cionamiento correcto de la base de datos. Por *base de datos concreta* o *SGDB concreto* entendemos una aplicación concreta de bases de datos. En el caso de bases de datos relacionales, ejemplos de SGBD concretos son Oracle Database, Mysql, SQL Server o IBM Informix, entre otros.

Finalmente, la última etapa es la **implementación y optimización** de la base de datos. Esta etapa permite cargar los datos y posteriormente permite ajustar algunos parámetros del modelo físico y para optimizar el rendimiento de la base de datos.

Estas etapas del diseño no hay que seguirlas estrictamente de manera secuencial, y en muchos casos es habitual rehacer el diseño de la etapa anterior a partir de necesidades detectadas en fases posteriores. Estos **bucles de retroalimentación** son habituales y permiten afinar los diseños de las distintas etapas de una manera iterativa.

El proceso que muestra la figura 1 se basa en el modelo de **diseño orientado a datos**. Este modelo se centra en el diseño de los contenedores de la información y en la estructura de la base de datos. Paralelamente a este modelo existe el modelo de **diseño orientado a procesos**, que se centra en las aplicaciones de bases de datos para determinar los datos y el uso que de estas hacen las aplicaciones. Tradicionalmente, el diseño de aplicaciones se ha basado en este segundo modelo, pero cada vez resulta más claro que ambas actividades son paralelas y que están estrechamente interrelacionadas. Las herramientas de diseño de bases de datos y de aplicaciones se combinan cada vez con mayor frecuencia.

2. Fases del diseño de una base de datos

A continuación veremos con algo más de detalle las fases que forman el proceso de diseño de una base de datos.

2.1. Fase 1. Recogida y análisis de requisitos

La primera fase en el diseño de una base de datos consiste en conocer y analizar con detalle las expectativas, las necesidades y los objetivos de los futuros usuarios de la base de datos. Este proceso se denomina *recogida y análisis de requisitos*.

La fase de recogida y análisis de requisitos se puede dividir en tres subfases secuenciales: la recogida de requisitos; la estructuración y el refinamiento de los requisitos, y la formalización de los requisitos.

2.1.1. Recogida de requisitos

Para determinar los requisitos, en primer lugar hay que establecer los actores del sistema de información que interaccionarán con la base de datos. Esto incluye a los usuarios y las aplicaciones, tanto si son nuevos como si no lo son. Normalmente, un grupo de analistas se encarga de hacer el análisis de requisitos y, muy probablemente, este análisis suele ser informal, incompleto e incluso incoherente en algún punto. Por lo tanto, hay que dedicar muchos esfuerzos a trabajar esta información y convertirla en una especificación que los diseñadores de bases de datos puedan utilizar para modelizar e implementar el sistema de información.

En esta fase, no solo hay que recoger y analizar los requisitos referentes a la estructura o la forma de la información (tipos de datos y relaciones entre ítems de datos), sino que hay que capturar y analizar cualquier tipo de requisito, independientemente de qué tipo sea. Hay que recoger, analizar y documentar cualquier requisito que los usuarios esperen de la base de datos. Esto incluye los procesos que se deben ejecutar sobre la base de datos, las restricciones sobre los datos, las restricciones sobre el rendimiento del sistema de información, las restricciones relativas a la implementación (tanto en lo que respecta al hardware como en lo que se refiere al software), requisitos de seguridad o requisitos de rendimiento (por ejemplo, tiempo de respuesta), entre otros. Algunas de las actividades más habituales de esta fase son las siguientes:

- Identificar los grupos de usuarios y las principales áreas de aplicación que utilizarán la base de datos y que se verán directa o indirectamente afecta-

dos por ésta. Dentro de cada grupo hay que elegir usuarios clave y formar comités para llevar a cabo la recopilación y la especificación de requisitos.

- Estudiar y analizar la documentación existente relativa a las aplicaciones en uso.
- Estudiar el entorno actual y el uso que se quiere dar a la información. Esto incluye el estudio de las entradas, el flujo y las salidas de información, además de las frecuencias y los usos de las diferentes tareas dentro del sistema de información.
- Hacer entrevistas y encuestas a los futuros usuarios para que puedan manifestar su opinión y sus prioridades acerca del nuevo sistema de información.

2.1.2. Estructuración y refinamiento de los requisitos

Se debe tener en cuenta que algunos de estos requisitos, muy probablemente, cambiarán durante el proceso de diseño y que hay que estar atentos y en contacto permanente con los usuarios de la base de datos para detectar posibles problemas. Es una buena práctica incorporar los usuarios de la base de datos durante el proceso de desarrollo, puesto que así se incrementa su grado de implicación y satisfacción. Hay algunas propuestas de metodologías para la recogida y el análisis de requisitos basadas en el trabajo conjunto de los desarrolladores con los usuarios de la base de datos, como, por ejemplo, el diseño conjunto de aplicaciones (JAD²).

⁽²⁾JAD es la sigla de la expresión inglesa *joint application design*.

2.1.3. Formalización de los requisitos

El paso siguiente es convertir los requisitos a un formato estructurado mediante técnicas de especificación de requisitos como, por ejemplo, el análisis orientado a objetos (OOA³), diagramas de flujo de datos (DFD⁴) o la notación Z. Estas técnicas utilizan diferentes tipos de recursos (diagramas, texto, tablas, gráficos, diagramas de decisión, etc.) para organizar y representar los requisitos de manera clara.

⁽³⁾OOA es la sigla de la expresión inglesa *object oriented analysis*.

⁽⁴⁾DFD es la sigla de la expresión inglesa *data flow diagrams*.

La notación Z

La notación Z es un lenguaje formal utilizado en ingeniería del software.

Esta fase puede representar un coste importante dentro del proceso de diseño de una base de datos, pero es muy importante y puede ser determinante para el éxito o el fracaso del sistema de información. Detectar y corregir los errores o problemas en las fases iniciales del proyecto es mucho menos costoso que arrastrar los errores hasta las fases finales, cuando corregirlos tendrá unos costes mucho más importantes. La satisfacción del usuario final vendrá determinada por la capacidad de recoger y captar sus necesidades e implementarlas de manera correcta en la solución final.

2.2. Fase 2. Diseño conceptual

La fase de **diseño conceptual** tiene como objetivo crear un esquema conceptual de alto nivel e independiente de la tecnología a partir de los requisitos, las especificaciones y las restricciones que se han recogido en la fase anterior.

En esta fase se parte de la recogida y el análisis de requisitos obtenidos en la fase anterior y tiene como objetivo diseñar un esquema conceptual de la base de datos que sea consistente con los requisitos, las especificaciones y las restricciones impuestas por la problemática que hay que resolver.

Un **esquema conceptual** es una descripción concisa de los requisitos de datos que se expresa mediante conceptos proporcionados por un modelo de datos de alto nivel, fácil de entender y sin detalles de implementación. El esquema, además, debe servir de referencia para verificar que se han agrupado todos los requisitos y que no hay ningún conflicto entre ellos.

En esta fase del diseño todavía no se considera el tipo de base de datos que se utilizará. Y, por lo tanto, tampoco el SGBD ni el lenguaje concreto de implementación de la base de datos. En esta etapa nos concentraremos en la estructura de la información, sin resolver de momento cuestiones relacionadas con la tecnología.

2.2.1. El modelo ER

Hay varios modelos de datos de alto nivel que permiten modelizar los requisitos, las especificaciones y las restricciones que se han obtenido en la primera fase del diseño de una base de datos. Uno de los más conocidos y utilizados es el modelo entidad-interrelación, que abreviaremos como **modelo ER**. Este modelo es uno de los más utilizados en el diseño conceptual de las aplicaciones de bases de datos, principalmente, debido a su simplicidad y facilidad de uso.

Los principales elementos que incluye el modelo son los tipos de entidad, los atributos y los tipos de relaciones entre entidades. El objetivo principal del modelo ER es permitir a los diseñadores reflejar en un modelo conceptual los requisitos del mundo real que sean de interés para el problema. El modelo ER facilita el diseño conceptual de una base de datos y, como ya hemos comentado, es aplicable al diseño de cualquier tipo de bases de datos.

Modelo ER

En inglés se denomina *entity-relationship model*. Dada la ambigüedad de la traducción, en español encontramos autores que lo traducen como *modelo entidad-relación* y otros que lo traducen como *modelo entidad-interrelación*. Ambos conceptos se refieren al mismo modelo.

2.2.2. El lenguaje unificado de modelización

El lenguaje unificado de modelización (UML) es un lenguaje gráfico diseñado para especificar, visualizar, modificar, construir y documentar un sistema. El lenguaje UML incorpora una gran cantidad de diagramas que permiten representar el modelo de un sistema desde perspectivas diferentes. En relación con el diseño conceptual de bases de datos, nos interesa especialmente el diagrama de clases, que permite representar información del dominio de discurso. Los diagramas de clases son diagramas estáticos que describen la estructura de un sistema a partir de las clases o tipos de entidad del sistema, sus atributos y las asociaciones o tipos de relaciones que se establecen entre ellos. Estos diagramas han mostrado una capacidad excelente para la modelización de datos. Por este motivo, son cada vez más importantes en el diseño conceptual de bases de datos.

Lenguaje UML

El lenguaje unificado de modelización (en inglés, *unified modeling language*) es un lenguaje de propósito general para modelizar sistemas de software. Este estándar fue creado, y actualmente es mantenido, por el Object Management Group (OMG).

2.3. Fase 3. Diseño lógico

Previamente a la fase de diseño lógico, se debe elegir un tipo de base de datos. Es decir, no hay que escoger todavía un SGBD concreto, sino que simplemente hay que seleccionar el tipo de base de datos que se quiere implementar. Es importante que quede claro que el tipo de base de datos determina el esquema de diseño lógico. Una vez elegido el tipo de SGBD donde se quiere implementar la base de datos, ya se puede iniciar la fase del diseño lógico.

En la fase de **diseño lógico** se transforma el modelo conceptual, independiente del tipo de tecnología, en un modelo lógico dependiente del tipo de SGBD en el que se quiere implementar la base de datos.

En esta etapa se parte del diseño conceptual desarrollado en el paso anterior y se obtiene un diseño lógico de la futura base de datos. En esta transformación se ajusta el modelo considerando el tipo de SGBD en el que se quiere implementar la base de datos. Por ejemplo, si se quiere crear la base de datos en un sistema relacional, esta etapa obtendrá un conjunto de relaciones con sus atributos, sus claves primarias y sus claves foráneas correspondientes.

En esta etapa nos concentramos en las cuestiones tecnológicas relacionadas con el modelo de la base de datos, asumiendo que en la etapa anterior ya hemos resuelto la problemática de estructuración de la información desde el punto de vista conceptual.

Por lo tanto, el resultado de esta etapa será un modelo lógico de la estructura de la información. Generalmente, cuando este modelo lógico hace referencia a un SGBD relacional, se denomina *modelo relacional*. En este material nos centraremos en la transformación del modelo conceptual a un modelo relacional, es decir, a un modelo lógico para una base de datos relacional.

En el diseño lógico se pueden ver tres subfases o partes independientes, que se aplican de manera secuencial para transformar el modelo conceptual obtenido en la fase anterior en un modelo lógico que será el resultado de esta fase. En primer lugar, en la subfase llamada *reconsideraciones del modelo conceptual* revisamos el modelo conceptual para asegurarnos de que está libre de algunos errores tipificados e identificables. A continuación, se produce la transformación del modelo conceptual en el modelo lógico y, finalmente, aplicamos la teoría de la normalización al modelo lógico.

2.3.1. Reconsideraciones del modelo conceptual

En esta primera parte se realiza un análisis en profundidad del modelo conceptual obtenido en la fase anterior, con la intención de detectar y corregir algunos errores que se suelen producir en los modelos conceptuales y que conviene detectar y reparar lo antes posible para evitar que se propaguen en fases posteriores. Dichos errores se denominan *trampas de diseño*. Es conveniente conocer cada uno de estos errores y asegurarse de que el modelo conceptual que se quiere transformar está libre de ellos antes de continuar con el diseño lógico.

2.3.2. Transformación del modelo conceptual en el modelo lógico

En estos materiales nos centraremos en el diseño lógico de bases de datos relacionales. Por lo tanto, el modelo lógico generado será aplicable a cualquier base de datos relacional. Partiremos del resultado de la etapa del diseño conceptual expresado mediante un diagrama de clases UML y veremos cómo se puede transformar utilizando una estructura de datos del modelo relacional. Este proceso muestra la manera de llevar a cabo la transformación de los diferentes elementos que forman el modelo conceptual en elementos que constituyen el modelo lógico o, más concretamente, el modelo relacional.

El modelo relacional

El modelo relacional es el modelo lógico específico para bases de datos relacionales.

2.3.3. Normalización

La teoría de la normalización aplica la teoría de conjuntos, la lógica y el álgebra relacional para formalizar un conjunto de ideas simples, que guían un buen diseño de bases de datos relacionales.

La teoría de la normalización utiliza las formas normales (FN) para reconocer los casos en los que no se aplican buenos criterios de diseño. Una relación está en una forma normal determinada si satisface un conjunto de restriccio-

nes específicas que son propias de esta forma normal. La infracción de estas restricciones origina que la relación tenga un conjunto de anomalías y redundancias de actualización no deseables. Las formas normales son declarativas, es decir, cada forma normal indica las restricciones que se deben cumplir, pero no describe ningún procedimiento para conseguirlo.

2.4. Fase 4. Diseño físico

Previamente a la fase de diseño físico hay que elegir un SGBD concreto. Hay que estudiar los diferentes sistemas comerciales o libres que hay en el mercado y seleccionar un SGBD donde se pueda implementar el sistema de información que se ha ido gestando en las fases anteriores del proceso de diseño.

Los componentes físicos que forman cada SGBD son específicos. Los fabricantes utilizan estrategias y tecnologías diferentes para maximizar el rendimiento de sus sistemas gestores de bases de datos. En este nivel no existe ningún estándar y, por lo tanto, habrá que adaptar el esquema lógico obtenido en el paso anterior, teniendo presentes las características de cada sistema gestor. El diseñador debe considerar los aspectos de implementación física y de eficiencia que dependen específicamente del SGBD elegido.

El **diseño físico** es una fase del proceso de diseño de bases de datos que adapta el esquema lógico obtenido en la fase anterior al SGBD concreto, que utilizará el sistema de información.

2.4.1. El nivel físico y el nivel virtual

El estudio de los niveles físico y virtual de las bases de datos permite ver aspectos como las estructuras de almacenamiento y las rutas de acceso por los ficheros de la base de datos. Cada SGBD ofrece diferentes opciones para organizar ficheros y rutas de acceso, y es necesario que el diseñador conozca la implementación concreta, con el fin de implementar de una manera más eficiente el diseño físico del sistema de información.

También es importante conocer las características de los procesos que consultan y actualizan la base de datos, como por ejemplo las frecuencias de ejecución y los volúmenes que se espera tener de los diferentes datos que se quieren almacenar con el fin de conseguir un buen rendimiento de la base de datos.

Algunos criterios importantes que pueden ser útiles para elegir las opciones de diseño físicas de la base de datos son los siguientes:

- **Tiempo de respuesta.** Es el tiempo que transcurre desde que se envía una petición al SGBD hasta que éste devuelve los datos de la respuesta. Una parte importante de este tiempo está bajo el control del SGBD y hace refe-

rencia al tiempo de acceso por parte del SGBD a los datos requeridos para generar la respuesta. Otros aspectos no son controlados por el SGBD, como por ejemplo la planificación del sistema operativo o los tiempos de acceso a los medios físicos de almacenamiento de los datos.

- **Uso del espacio.** Es la cantidad de espacio de disco utilizado por los ficheros de la base de datos y las estructuras de rutas de acceso al disco, incluyendo índices y otras rutas de acceso.
- **Rendimiento.** Es la cantidad media de transacciones que se pueden procesar en un minuto de tiempo. Este factor puede ser crítico para sistemas transaccionales, como por ejemplo líneas aeronáuticas o entidades bancarias.

2.4.2. Transformación del modelo lógico en el modelo físico

En este paso se transforma el modelo lógico de una base de datos en un modelo físico, según el SGBD elegido para implementar el sistema de información, las características concretas del hardware utilizado y el sistema operativo y el software básico. Cada SGBD ha desarrollado un lenguaje propio, hecho a medida por el constructor mismo, para implementar el diseño físico de la base de datos, de acuerdo con las características del entorno, y para obtener el máximo rendimiento del hardware, del sistema operativo y del propio gestor. En realidad, se podría considerar una ampliación del lenguaje SQL estándar con las cláusulas propias que cada gestor necesita para definir los componentes del diseño físico. Sin embargo, existe un gran parecido o equivalencia entre una parte importante de los diferentes gestores.

El estándar SQL incorpora la definición de todos los componentes del diseño lógico de la base de datos. En cambio, no contiene ningún elemento del diseño físico.

Para transformar el diseño lógico de la base de datos en un SGBD concreto, partimos de las definiciones de tablas (con toda la información relacionada; es decir, atributos, claves primarias, claves foráneas y claves alternativas). A continuación se relaciona cada elemento con un espacio adecuado en el nivel virtual y finalmente se relaciona cada espacio virtual con un fichero físico, que constituye el nivel físico del sistema de información.

2.5. Fase 5. Implementación y optimización

La última etapa es la implementación y la optimización de la base de datos.

La **etapa de implementación y optimización** consiste en realizar la carga de los datos y posteriormente ajustar algunos parámetros relacionados con el modelo físico de la base de datos para optimizar el rendimiento.

El objetivo principal de esta etapa es optimizar el rendimiento de la base de datos. En primer lugar, hay que realizar la carga de los datos, puesto que no es posible optimizar el acceso a los datos sin poder determinar el tamaño de las tablas, los tipos de accesos y consultas, la frecuencia de éstos, etc.

Finalmente, también habrá que concretar los diferentes roles de usuarios y aplicaciones para poder determinar los permisos de los diferentes grupos. El componente de gestión de la seguridad y las vistas permiten limitar los accesos y de este modo reducir el riesgo de problemas derivados de accesos no autorizados.

2.5.1. Procesamiento y optimización de consultas

El lenguaje SQL empleado por las bases de datos relacionales es un lenguaje declarativo. Esto implica que se especifica el resultado que se quiere obtener a partir de una consulta realizada, en lugar de determinar el algoritmo o el método que hay que usar para obtener el resultado. Por lo tanto, es necesario entender el conjunto de tareas que realizan los SGBD relacionales para obtener la respuesta deseada.

Los SGBD relacionales evalúan sistemáticamente las posibles estrategias alternativas que se pueden presentar, con el objetivo de elegir la que se considera óptima. El procesamiento de consultas recoge todo el conjunto de actividades realizadas por el SGBD, que tienen como objetivo la extracción de información de la base de datos para lograr la estrategia más eficiente y proporcionar un mejor rendimiento del sistema de información.

El procesamiento de consultas se puede dividir en las cuatro etapas principales siguientes:

1) Descomposición de la consulta. Traducción de la consulta expresada en lenguaje SQL a una representación interna basada en el álgebra relacional.

2) Optimización de la consulta. Proceso de selección del plan de ejecución de la consulta más eficiente entre las muchas estrategias generalmente disponibles en el procesamiento de una consulta. La optimización de consultas se puede realizar sobre tres vertientes:

a) Optimización semántica basada en las restricciones especificadas en el esquema de la base de datos.

b) Optimización sintáctica, que consiste en transformar heurísticamente la expresión relacional original en otra equivalente pero que sea mucho más eficiente.

c) Optimización física, con el objetivo de elegir entre los distintos planes de evaluación –que pueden tener costes diferentes– el que sea más eficiente. Para determinar el coste más eficiente hay que conocer el coste de cada operación, que a menudo depende de diferentes parámetros.

3) Generación de código.

4) Ejecución de la consulta.

Es el proceso de optimización de consultas, es decir, el tratamiento que da un SGBD a las consultas hechas por los usuarios mediante SQL. Este punto es uno de los más importantes que se deben tener en cuenta cuando se diseña un SGBD relacional, puesto que la opción elegida afecta directamente al rendimiento global del sistema.

La optimización de consultas es un aspecto muy importante que hay que considerar cuando se diseña y se construye un SGBD relacional. Las técnicas que se utilizan para optimizar consultas condicionan el rendimiento global del sistema, puesto que determinan el tiempo que necesita el sistema gestor para resolver las consultas que han hecho los usuarios.

2.5.2. Procesamiento de vistas

Una vista es una tabla lógica que permite acceder a la información de una o varias tablas mediante una consulta predefinida. No contiene información por sí misma, sino que se basa en información de otras tablas. Las vistas proporcionan mecanismos de seguridad y permiten al diseñador de la base de datos personalizar la vista que tienen los diferentes usuarios de la base de datos.

Hacer un uso adecuado de las vistas es un aspecto clave para lograr un buen diseño de una base de datos, puesto que nos permite ocultar los detalles de las tablas y mantener la visión del usuario independientemente de la evolución que tenga la estructura de tablas.

Las vistas habían sido durante mucho tiempo un simple mecanismo de simplificación de consultas, pero actualmente tienen una importancia capital en varias áreas, como el diseño externo, los almacenes de datos⁵ o la informática distribuida.

⁽⁵⁾En inglés, *data warehouse*.

2.5.3. Administración de la seguridad

Por último, hay que tener en cuenta las técnicas que se emplean para proteger la base de datos contra accesos no autorizados y los mecanismos para asignar y revocar privilegios a los diferentes usuarios. De estas y otras acciones se encarga el componente de seguridad del SGBD. Este componente viene siendo cada día más importante, dado que en la actualidad una gran cantidad de ordenadores y otros tipos de dispositivos están interconectados y, por lo tanto, cualquier persona podría convertirse en usuario, y posible atacante, de una base de datos.

En muchas organizaciones, la información es un activo intangible y de naturaleza sensible, que tiene un valor muy importante. Para preservar esta información hay que proteger el sistema de información y conocer las obligaciones legales que hay que cumplir.

Resumen

En este módulo hemos visto, de una manera muy general, el proceso de diseño de una base de datos.

Hemos introducido, aunque muy brevemente, las distintas etapas que forman el proceso de diseño de una base de datos. Abordaremos este tema en detalle en el resto de los módulos de la asignatura.

El proceso de diseño de una base de datos se inicia con la fase de recogida y análisis de requisitos, lo cual permite recoger y centralizar las necesidades de los diferentes grupos de usuarios y aplicaciones. A partir de este análisis, en la segunda fase, se modeliza un esquema conceptual que permite describir el modelo de datos de una manera independiente de la tecnología.

La etapa siguiente en este proceso es el diseño lógico, que requiere haber elegido previamente el tipo de base de datos que se quiere utilizar en la implementación del sistema de información. El tipo de base de datos determina el modelo lógico que va a desarrollarse. Por ejemplo, en el caso de utilizar un tipo de base de datos relacional, se transforma el modelo conceptual en un modelo lógico específico para bases de datos relacionales denominado *modelo relacional*.

El diseño físico permitirá adaptar el modelo lógico a un sistema gestor de bases de datos (SGBD) concreto. Por lo tanto, previamente a este paso se deberá escoger el SGBD específico que se quiere utilizar para implementar el sistema de información. En esta etapa se crea la estructura física que almacenará los datos de la base de datos.

Para terminar, la última etapa permite la optimización de la base de datos y la gestión de la seguridad relacionada con los usuarios y las aplicaciones de la base de datos. Lógicamente, habrá que realizar la carga de los datos previamente, puesto que el tamaño de las tablas, los tipos de consultas y las frecuencias de estas son elementos importantes para optimizar el acceso a los datos por parte del sistema gestor.

Glosario

database management system *f* Véase **sistema gestor de bases de datos**.
sigla **DBMS**

divide and conquer *loc* Véase **divide y vencerás**.

divide y vencerás *loc* Estrategia que propone resolver un problema complejo mediante la subdivisión en un conjunto de problemas más sencillos cuya resolución implica resolver el problema inicial.
en divide and conquer

entity relationship model *m* Véase **modelo ER**.

lenguaje unificado de modelización *m* Lenguaje de propósito general para modelizar sistemas de software. El estándar fue creado, y actualmente es mantenido, por el Grupo de Gestión de Objetos.
en unified modeling language
sigla **UML**

modelo ER *m* Modelo entidad-interrelación de datos de alto nivel que permite modelizar los requisitos, las especificaciones y las restricciones.
en entity-relationship model

SGBD *m* Véase **sistema gestor de bases de datos**.

sistema gestor de bases de datos *m* Tipo de software específico que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
en database management system
sigla **SGBD**

UML *m* Véase **lenguaje unificado de modelización**.

Bibliografía

Elmasri, Ramez; Navathe, Shamkant, B. (2007). *Fundamentos de sistemas de bases de datos* (5.^a ed.). Madrid: Pearson Educación.

Ramakrishnan, Raghu; Gehrke, Johannes (2003). *Database management systems* (3.^a ed.). Boston: McGraw-Hill Higher Education.