

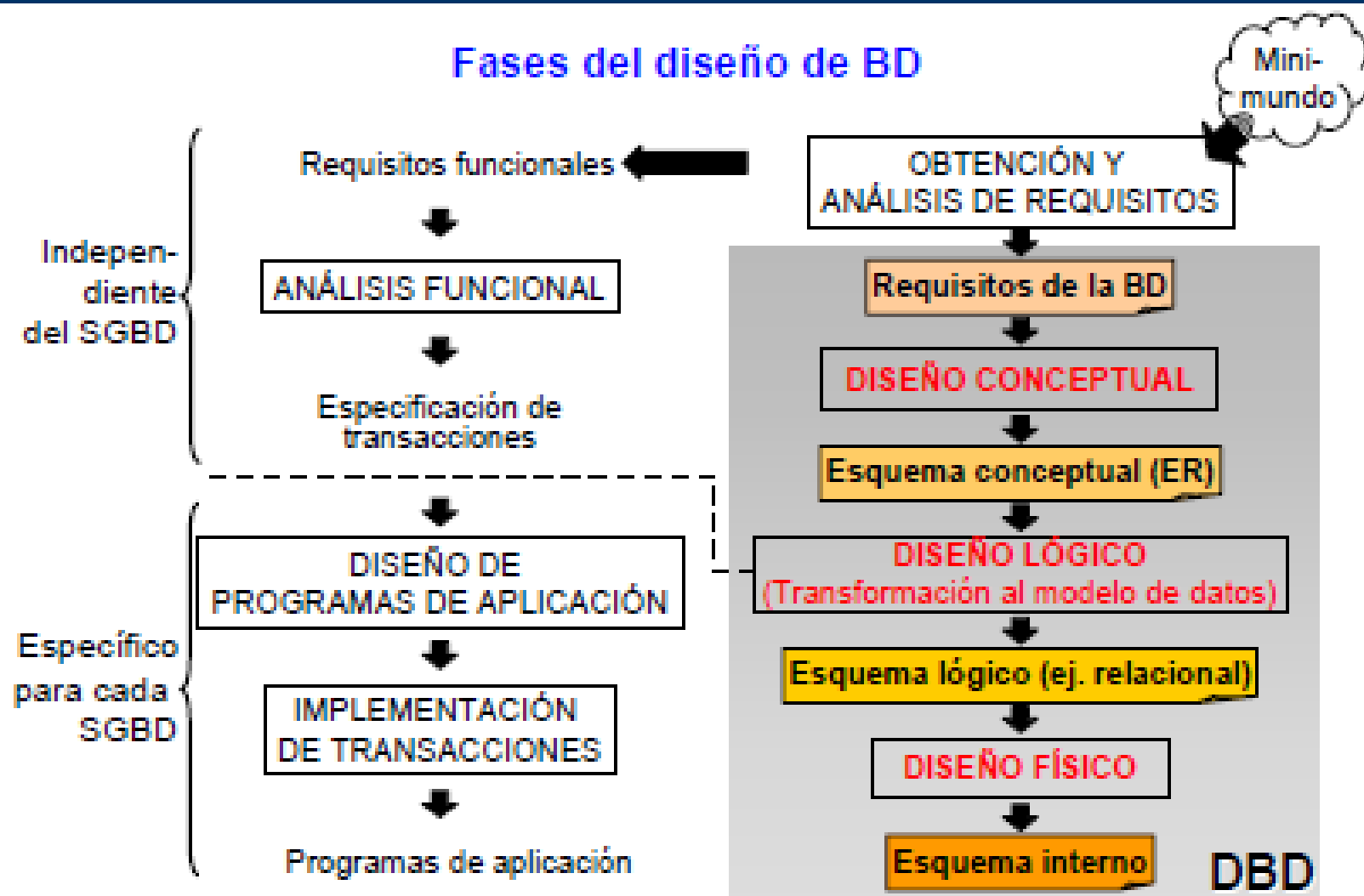
Tema 4

Diseño Físico de Bases de Datos

DAM/DAW - Bases de datos

José Micolau Lombarte

Fases del Diseño de BD



4.1 Introducción al Diseño Físico

- ◆ Conceptualmente, podemos definir el diseño físico de una base de datos como un proceso que, a partir de su diseño lógico y de información sobre su uso esperado, creará una configuración física de la base de datos adaptada al entorno donde se alojará y que permita el almacenamiento y la explotación de los datos de la base de datos con un rendimiento adecuado.
- ◆ Este proceso tendrá como elementos de diseño:
 - Entradas:
 - El esquema lógico de nuestra BD (elaborado previamente).
 - Información sobre el uso esperado de la BD.
 - Salidas
 - Decisiones sobre las estructuras físicas más adecuadas a usar.
 - Elementos como: tablas, índices, vistas, particiones, otros métodos.
 - Adaptación al entorno: SGBD de ejecución y HW donde se alojará el SGBD
 - Que permita el almacenamiento y explotación de los datos con un rendimiento adecuado.
- ◆ **OBJETIVO : definir como será y se implementará nuestra base de datos en el soporte/entorno donde se va a ejecutar.**

4.1 Introducción al Diseño Físico

- ◆ Desde un punto de vista teórico, en el modelo relacional, los datos se almacenan en relaciones. Cada relación contiene tuplas, que se corresponden con las instancias (u ocurrencias) de un concepto. Las tuplas están compuestas por atributos, que son los que permiten representar las características (o propiedades) de los conceptos. Al implementar el modelo anterior en un SGBD relacional, las relaciones se representan mediante tablas, que contienen filas, que corresponden a las tuplas de la relación. Cada tabla tiene un conjunto de columnas, que representan a los atributos del concepto a representar. Al implementar las tablas en ficheros, las filas se representan mediante registros y las columnas mediante campos. La siguiente tabla muestra un resumen de esta nomenclatura y la relación entre términos. A lo largo del módulo usaremos principalmente los términos correspondientes a las dos últimas filas.

Nivel	Concepto	Instancia	Característica
Modelo relacional	Relación	Tupla	Atributo
SQL	Tabla	Fila	Columna
Físico	Fichero	Registro	Campo

4.2 Diseño Físico

- ◆ Los datos se almacenan en soportes no volátiles gestionados por el sistema operativo (SO) de la máquina donde se alojan. Es el SO el encargado de efectuar las operaciones de lectura y escritura física de los datos. Los SGBD aprovechan las rutinas existentes de los SO para leer y escribir los datos de las bases de datos.
- ◆ Los SO gestionan los datos a partir de unas estructuras globales llamadas ficheros. Normalmente el SO no reserva una gran cantidad de espacio destinada a satisfacer todas las necesidades futuras de almacenamiento en el dispositivo de almacenamiento no volátil, sino que realiza una asignación inicial y va adquiriendo más espacio a medida que lo necesita. La unidad de adquisición se denomina extensión. Una extensión es una agrupación de páginas consecutivas en el dispositivo de almacenamiento no volátil. A su vez, las páginas son el componente físico más pequeño. Las páginas son los elementos que finalmente contienen y almacenan los datos del nivel lógico.
- ◆ La página es el componente más importante del nivel físico ya que es la unidad mínima de acceso y de transporte del sistema de entrada y salida (E/S) de un SGBD. Eso quiere decir que:
 - La página es la unidad mínima de transferencia entre la memoria externa (no volátil) y la memoria interna (o memoria principal).
 - En los SGBD el espacio en el dispositivo de almacenamiento no volátil siempre se asigna en un número múltiplo de páginas.
 - Cada página puede ser direccionada individualmente.

4.2 Metodología del Diseño Físico

El objetivo de esta etapa es producir una descripción de la implementación de la base de datos en memoria secundaria. Esta descripción incluye las estructuras de almacenamiento y los métodos de acceso que se utilizarán para conseguir un acceso eficiente a los datos.

El diseño físico se divide de cuatro fases, cada una de ellas compuesta por una serie de pasos:

- Traducir el esquema lógico global para el SGBD específico.
 1. Diseñar las relaciones base para el SGBD específico.
 2. Diseñar las reglas de negocio para el SGBD específico.
- Diseñar la representación física.
 1. Analizar las transacciones.
 2. Escoger las organizaciones de ficheros.
 3. Escoger los índices secundarios.
 4. Considerar la introducción de redundancias controladas.
 5. Estimar la necesidad de espacio en disco.
- Diseñar los mecanismos de seguridad.
 1. Diseñar las vistas de los usuarios.
 2. Diseñar las reglas de acceso.
- Monitorizar y afinar el sistema.

4.2 Metodología del Diseño Físico

2.1 Traducir el esquema lógico global

La primera fase del diseño lógico consiste en traducir el esquema lógico global en un esquema que se pueda implementar en el SGBD escogido. Para ello, es necesario conocer toda la funcionalidad que éste ofrece. Por ejemplo, el diseñador deberá saber:

- Si el sistema soporta la definición de claves primarias, claves ajenas y claves alternativas.
- Si el sistema soporta la definición de datos requeridos (es decir, si se pueden definir atributos como no nulos).
- Si el sistema soporta la definición de dominios.
- Si el sistema soporta la definición de reglas de negocio.
- Cómo se crean las relaciones base.

4.2 Metodología del Diseño Físico

2.1.1. Diseñar las relaciones base para el SGBD específico

Las relaciones base se definen mediante el lenguaje de definición de datos del SGBD. Para ello, se utiliza la información producida durante el diseño lógico: el esquema lógico global y el diccionario de datos. El esquema lógico consta de un conjunto de relaciones y, para cada una de ellas, se tiene:

- El nombre de la relación.
- La lista de atributos entre paréntesis.
- La clave primaria y las claves ajenas, si las tiene.
- Las reglas de integridad de las claves ajenas.

En el diccionario de datos se describen los atributos y, para cada uno de ellos, se tiene:

- Su dominio: tipo de datos, longitud y restricciones de dominio.
- El valor por defecto, que es opcional.
- Si admite nulos.
- Si es derivado y, en caso de serlo, cómo se calcula su valor.

4.2 Metodología del Diseño Físico

2.1.2. Diseñar las reglas de negocio para el SGBD específico

Las actualizaciones que se realizan sobre las relaciones de la base de datos deben observar ciertas restricciones que imponen las reglas de negocio de la empresa. Algunos SGBD proporcionan mecanismos que permiten definir estas restricciones y vigilan que no se violen.

4.2 Metodología del Diseño Físico

2.2 Diseñar la representación física

Uno de los objetivos principales del diseño físico es almacenar los datos de modo eficiente. Para medir la eficiencia hay varios factores que se deben tener en cuenta:

- *Productividad de transacciones.* Es el número de transacciones que se quiere procesar en un intervalo de tiempo.
- *Tiempo de respuesta.* Es el tiempo que tarda en ejecutarse una transacción. Desde el punto de vista del usuario, este tiempo debería ser el mínimo posible.
- *Espacio en disco.* Es la cantidad de espacio en disco que hace falta para los ficheros de la base de datos. Normalmente, el diseñador querrá minimizar este espacio.

4.2 Metodología del Diseño Físico

2.2.1 Analizar las transacciones

Una transacción es una unidad de trabajo sobre la Base de Datos atómica y totalmente recuperable. Todas las acciones de una transacción son totalmente realizadas a deshechas. Para realizar un buen diseño físico es necesario conocer las consultas y las transacciones que se van a ejecutar sobre la base de datos. Esto incluye tanto información cualitativa, como cuantitativa. Para cada transacción, hay que especificar:

- La frecuencia con que se va a ejecutar.
- Las relaciones y los atributos a los que accede la transacción, y el tipo de acceso: consulta, inserción, modificación o eliminación. Los atributos que se modifican no son buenos candidatos para construir estructuras de acceso.
- Los atributos que se utilizan en los predicados del WHERE de las sentencias SQL. Estos atributos pueden ser candidatos para construir estructuras de acceso dependiendo del tipo de predicado que se utilice.
- Si es una consulta, los atributos involucrados en el join de dos o más relaciones. Estos atributos pueden ser candidatos para construir estructuras de acceso.
- Las restricciones temporales impuestas sobre la transacción. Los atributos utilizados en los predicados de la transacción pueden ser candidatos para construir estructuras de acceso.

4.2 Metodología del Diseño Físico

Para cada **consulta** se debe especificar lo siguiente:

- Ficheros a los que tendrá acceso la consulta.
- Campo(s) sobre los que se especificará las condiciones de selección de la consulta.
- Campo(s) sobre los que se especificarán condiciones de concatenación, o condiciones para enlazar registros.
- Campos cuyos valores obtendrá la consulta. Los campos mencionados en los apartados 2 y 3 son candidatos para definir estructuras de acceso.

Para cada **transacción de actualización** se debe especificar lo siguiente:

- Ficheros que se actualizarán.
- Tipo de actualización en cada fichero.
- Campos sobre los que se especificarán condiciones de selección para una operación de modificación o de borrado.
- Campos cuyos valores alterará una operación de modificación.

4.2 Metodología del Diseño Físico

2.2.2 Escoger las organizaciones de ficheros

El objetivo de este paso es escoger la organización de ficheros óptima para cada relación. Por ejemplo, un fichero desordenado es una buena estructura cuando se va a cargar gran cantidad de datos en una relación al inicializarla, cuando la relación tiene pocas tuplas, también cuando en cada acceso se deben obtener todas las tuplas de la relación, o cuando la relación tiene una estructura de acceso adicional, como puede ser un índice. Por otra parte, los ficheros dispersos (hashing) son apropiados cuando se accede a las tuplas a través de los valores exactos de alguno de sus campos (condición de igualdad en el WHERE). Si la condición de búsqueda es distinta de la igualdad (búsqueda por rango, por patrón, etc.), la dispersión no es una buena opción. Hay otras organizaciones, como la ISAM o los árboles B+.

4.2 Metodología del Diseño Físico

2.2.3 Escoger los índices secundarios

Los índices secundarios permiten especificar caminos de acceso adicionales para las relaciones base. Por ejemplo, la relación INMUEBLE se puede haber almacenado en un fichero disperso a través del atributo inum. Si se accede a menudo a esta relación a través del atributo alquiler, se puede plantear la creación de un índice sobre dicho atributo para favorecer estos accesos. Pero hay que tener en cuenta que estos índices conllevan un coste de mantenimiento que hay que sopesar frente a la ganancia en prestaciones. A la hora de seleccionar los índices, se pueden seguir las siguientes indicaciones:

- Construir un índice sobre la clave primaria de cada relación base.
- No crear índices sobre relaciones pequeñas.
- Añadir un índice sobre los atributos que se utilizan para acceder con mucha frecuencia.
- Añadir un índice sobre las claves ajenas que se utilicen con frecuencia para hacer joins.
- Evitar los índices sobre atributos que se modifican a menudo.
- Evitar los índices sobre atributos poco selectivos (aquellos en los que la consulta selecciona una porción significativa de la relación).
- Evitar los índices sobre atributos formados por tiras de caracteres largas.

Los índices creados se deben documentar, explicando las razones de su elección.

4.2 Metodología del Diseño Físico

2.2.4. Considerar la introducción de redundancias controladas

En ocasiones puede ser conveniente relajar las reglas de normalización introduciendo redundancias de forma controlada, con objeto de mejorar las prestaciones del sistema. En la etapa del diseño lógico se recomienda llegar, al menos, hasta la tercera forma normal para obtener un esquema con una estructura consistente y sin redundancias. Pero, a menudo, sucede que las bases de datos así normalizadas no proporcionan la máxima eficiencia, con lo que es necesario volver atrás y desnormalizar algunas relaciones, sacrificando los beneficios de la normalización para mejorar las prestaciones. Es importante hacer notar que la desnormalización sólo debe realizarse cuando se estime que el sistema no puede alcanzar las prestaciones deseadas. Y, desde luego, la necesidad de desnormalizar en ocasiones no implica eliminar la normalización del diseño lógico: la normalización obliga al diseñador a entender completamente cada uno de los atributos que se han de representar en la base de datos. Por lo tanto, hay que tener en cuenta los siguientes factores:

- La desnormalización hace que la implementación sea más compleja.
- La desnormalización hace que se sacrifique la flexibilidad.
- La desnormalización puede hacer que los accesos a datos sean más rápidos, pero ralentiza las actualizaciones.

4.2 Metodología del Diseño Físico

2.2.5. Estimar la necesidad de espacio en disco

Debe ser un requisito que la implementación de la Base de Datos Física pueda ser manejada por la configuración del Hardware corriente. Si este no es el caso, el diseñador todavía tiene que estimar la cantidad del espacio en disco que es necesaria para almacenar la Base de Datos. En el suceso que los nuevos Hardware tienen que ser obtenidos. El objetivo de estos pasos es para estimar la cantidad de espacio del disco que es necesario para el soporte de la implementación de la Base de Datos en almacenamiento secundario. Como con los pasos previos, estimar el disco usado es muy dependiente de la tarjeta DBMS y el Hardware usado para soportar la Base de Datos. En general, el estimar está basado en la cantidad de cada tupla y el número de tuplas en la relación. La última estimación debería ser un número máximo, pero debería también valer en consideración a como la relación aumentará, y modificando el resultado de la cantidad de disco por este factor de desarrollo para determinar la cantidad de potencial de la Base de Datos en el futuro.

4.2 Metodología del Diseño Físico

2.3 Diseñar los mecanismos de seguridad

Los datos constituyen un recurso esencial para la empresa, por lo tanto su seguridad es de vital importancia. Durante el diseño lógico se habrán especificado los requerimientos en cuanto a seguridad que en esta fase se deben implementar. Para llevar a cabo esta implementación, el diseñador debe conocer las posibilidades que ofrece el SGBD que se vaya a utilizar.

2.3.1. Diseñar las vistas de los usuarios

El objetivo de este paso es diseñar las vistas de los usuarios correspondientes a los esquemas lógicos locales. Las vistas, además de preservar la seguridad, mejoran la independencia de datos, reducen la complejidad y permiten que los usuarios vean los datos en el formato deseado.

2.3.2. Diseñar las reglas de acceso

El administrador de la base de datos asigna a cada usuario un identificador que tendrá una palabra secreta asociada por motivos de seguridad. Para cada usuario o grupo de usuarios se otorgarán permisos para realizar determinadas acciones sobre determinados objetos de la base de datos. Por ejemplo, los usuarios de un determinado grupo pueden tener permiso para consultar los datos de una relación base concreta y no tener permiso para actualizarlos.

4.2 Metodología del Diseño Físico

2.4 Monitorizar y afinar el sistema

Una vez implementado el esquema físico de la base de datos, se debe poner en marcha para observar sus prestaciones. Si éstas no son las deseadas, el esquema deberá cambiar para intentar satisfacerlas. Una vez afinado el esquema, no permanecerá estático, ya que tendrá que ir cambiando conforme lo requieran los nuevos requisitos de los usuarios. Los SGBD proporcionan herramientas para monitorizar el sistema mientras está en funcionamiento.