

Problema de diseño de una base de datos

El diseño de una base de datos es el proceso de desarrollo de las estructuras de una base de datos a partir de la utilización de los requisitos de los datos. Su objetivo es obtener bases de datos que representen finalmente la realidad u sean capaces de adaptarse fácilmente a los posibles cambios en los requisitos. Criterios para un diseño eficaz:

- Los datos de entrada deben ser completos, consistentes y utilizables
- La metodología de diseño debe ser correcta y utilizar efectivamente los datos de entrada
- El personal técnico y los usuarios deben estar comunicados interactivamente
- Es necesario utilizar un modelo conceptual (entidad/relación) para conseguir una comunicación efectiva entre desarrolladores y usuarios ![[Pasted image 20250916175115.png]]

Etapas de diseño

1. Formulación de requisitos y análisis

Su objetivo es identificar y describir los datos necesarios para la organización, su uso y reglas semánticas. Se utilizarán como entrada entrevistas con usuario, documentos de la empresa... lo que dará lugar a:

- Requisitos de información del usuario (datos usados y sus asociaciones)
- Requisitos de procesamiento (frecuencia de informes, tiempo de respuesta...)
- Algunas posibles cuestiones:
 - ¿Qué vistas de usuario se necesitan?
 - ¿Qué elementos de datos necesitan esas vistas?
- Claves para identificar las entidades
- Relaciones entre los datos
- Necesidades operacionales: seguridad, integridad, tiempo de respuesta A partir de todo ello, se elaborará un esquema descriptivo de la realidad formado por unas especificaciones de requisitos para el diseño conceptual Pasos a seguir:

1. Identificar el alcance del esfuerzo de la definición de la base de datos
2. Desarrollar una colección de metadatos estándar
3. Identificar las vistas de usuario y requisitos de datos
4. Construir un diccionario de datos y sus interrelaciones
 1. N° de elementos de datos
 2. Nombre
 3. Tipo

4. Longitud
 5. Rango
 6. Frecuencia de uso
 7. Autorización de actualización
5. Identificar el volumen de datos y los patrones de uso
 - Estimación del volumen de datos y su crecimiento
 - Uso de los datos: definir transacciones y accesos necesarios para transacción
 6. Establecer los requisitos operacionales
 - Seguridad
 - Integridad
 - Tiempo de respuesta
 - Copias de seguridad y restauración
 - Archivo (Históricos, tiempo de vida)
 - Crecimiento
 7. Diagramas de Flujo de Datos (DFD)

2. Diseño conceptual

En este paso se transforma el esquema descriptivo, refinándolo y estructurándolo adecuadamente. Esta etapa responde a la pregunta: "¿Cómo representar?" Es necesario buscar una representación basada en un modelo de datos que cumpla determinadas propiedades: coherencia, completitud, no redundancia, simplicidad, fidelidad..., para llegar así al denominado esquema conceptual. Para la representación del esquema conceptual, usaremos el Modelo Entidad/Relación Extendido (E/R), Además de unas plantillas que sirven de soporte documental junto al diagrama E/R

Un esquema conceptual comprende una descripción central única de los distintos contenidos de información que pueden coexistir en una base de datos. Debe de estar basado en:

1. El número de componentes básicos debe ser tan reducido como sea posible
2. Debe separar claramente conceptos distintos
3. Debe preservar la simetría, es decir, no destruir las simetrías naturales
4. La redundancia tiene que ser cuidadosamente controlada

Para elaborar este esquema conceptual se han utilizado diferentes metodologías:

- Descendentes (top-down): el esquema conceptual refleja directamente la visión de la empresa que se intenta modelar en la BDD. Se parte de un modelo general de la empresa para elaborar el esquema conceptual y, luego, sobre él se definen las vistas de usuario como subconjuntos de este esquema conceptual (Recomendado para BDD pequeñas)

- Ascendentes (bottom-up): el resultado de la integración de las vistas de los grupos de usuarios. Se empieza construyendo estas vistas y, teniendo en cuenta las restricciones entre dichas vistas, se elabora el esquema conceptual mediante un proceso de integración de vistas (Recomendado para BDD medianas y grandes)

3. Diseño lógico*

Objetivo: Transformar el modelo de datos conceptual en un modelo lógico conforme a un SGBD concreto y definir las especificaciones para los programas de acceso a la BDD

Entradas: - Esquema conceptual - Cuantificación de los requisitos operacionales - Volumen y cuantificación de uso - Especificaciones de los programas de alto nivel - Características del SGBD

Salidas: - Esquema procesable para el SGBD - Vistas externas y restricciones de seguridad para éstas - Especificaciones para el diseño físico - Guías para el diseño de programas - Guías para las operaciones sobre la base de datos (requisitos, restricciones y operaciones)

Pasos a seguir: 1. Correspondencia al modelo lógico 2. Diseño de los subesquemas 3. Diseño de programas 4. Revisión del diseño

4. Diseño físico

Esta es la última etapa, en la cual, teniendo presentes los requisitos de los procesos, características de los SGBD, del SO y el hardware, se pretenden entre otros, los siguientes objetivos:

- Disminuir los tiempos de respuesta
- Minimizar el espacio de almacenamiento
- Evitar las reorganizaciones periódicas
- Proporcionar la máxima seguridad
- Optimizar el consumo de recursos

En definitiva, cumplir los objetivos del sistema y conseguir la optimización de la relación coste/beneficio

Entradas: - Lista de objetivos del diseño físico con sus correspondientes prioridades y cuantificación (a ser posible) - Esquema lógico específico - Recursos de la máquina disponibles - Recursos software disponibles (S.O, ...) - Información sobre las aplicaciones que utilizarás la base de datos - Políticas de seguridad de datos

Salidas: - Estructura interna (esquema interno) - Especificaciones para el ajuste de la base de datos - Normas de seguridad

En general, existen 3 estrategias de los fabricantes para el diseño físico

a) El SGBD impone una estructura interna y deja muy poca flexibilidad al diseñador. Esto suele suponer una mayor independencia físico/lógica a costa de menor eficiencia

b) el DBA(Administrador de la Base de Datos) diseña la estructura interna. Esto supone más trabajo y un perjuicio para la independencia de datos, aunque puede mejorar la eficiencia

c) El SGBD proporciona una estructura interna inicial a partir de algunos parámetros dados por el diseñador, el cual, posteriormente, puede irlos modificando a fin de realizar su ajuste y optimizar, así, el rendimiento de la BDD