

Asignatura: **Administración de Bases de Datos**

Tema 2:

Desarrollo e Implantación de una Base de Datos

Pedro P. Alarcón Cavero
pedrop.alarcon@eui.upm.es

Juan Garbajosa Sopeña
jgs@eui.upm.es

Febrero 2010

Contenido

- Proceso de diseño de bases de datos
- Diseño conceptual
- Diseño lógico
- Estructura física de los SGBD
- Diseño físico
- Ejemplo SGBD
- Puesta en marcha de una base de datos

Contenido

Proceso Diseño
D. Conceptual
D. Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Proceso de Diseño de Bases de Datos

- Obtener la estructura lógica y física que contenga
 - la *información* útil para un grupo de usuarios de acuerdo con unos *requisitos* pre-establecidos
- Objetivos (difíciles de conseguir y de medir)
 - Requisitos funcionales
 - Requisitos de rendimiento
 - ...

Contenido

Proceso Diseño

D. Conceptual

D. Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
3

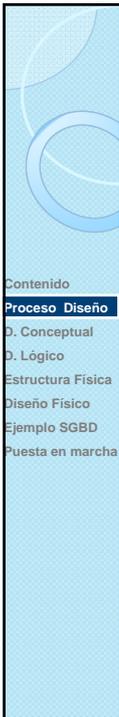
Proceso de Diseño de Bases de Datos

```

    graph TD
      subgraph Step1 [1. Análisis de Requisitos]
        RD[Requisitos de Datos]
        RP[Requisitos de Procesos]
      end
      subgraph Step2 [2. Diseño Conceptual]
        DEC[Diseño de Esquemas Conceptual y Externos  
(independ. del SGBD)]
        DP[Diseño de Procesos (transac. y diferidos)  
(independ. del SGBD)]
      end
      subgraph Step4 [4. Diseño Lógico]
        DLEC[Diseño de Esquemas Conceptual y Externos  
(dependiente del SGBD)]
      end
      subgraph Step5 [5. Diseño Físico]
        DEI[Diseño del Esquema Interno  
(dependiente del SGBD)]
      end
      subgraph Step6 [6. Implementación]
        SLD[Sentencias LDD  
Sentencias LDDA]
        IT[Implementación de Transacciones]
      end
      RD --> DEC
      RP --> DP
      DEC --> DLEC
      DLEC --> DEI
      DP --> IT
      DP --> DEI
      SLD --- C[Catálogo]
      SLD --- B[(BD)]
  
```

1. Análisis de Requisitos
2. Diseño Conceptual
3. Elección de SGBD
4. Diseño Lógico (*estándar y específico*)
5. Diseño Físico
6. Implementación
7. Carga

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
4



Contenido

Proceso Diseño

D. Conceptual

D. Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Análisis y Especificación de Requisitos

- Se refiere a la aplicación o sistema
- Se especifican requisitos referidos a datos y procesos
- Metodologías: OMT, UML, SADT, Warnier-Orr, DFD's, Ágiles, etc.
- Esta fase es costosa y de gran importancia

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 5



Contenido

Proceso Diseño

D. Conceptual

D. Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Elección de un SGBD

- Adecuación a la aplicación
 - modelo (relacional, red, OO...)
 - transacciones...
- Coste de adquisición del software
- Coste de adquisición del hardware
- Coste de mantenimiento
- Coste de creación y conversión de la BD
- Coste de personal
- Coste de formación
- Servicio venta y post-venta

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 6



2. Diseño Conceptual

- Compuesto por:
 - Diseño del esquema conceptual
 - Diseño de procesos (transacciones y procesos diferidos)
- Enfoques para diseño del esquema conceptual:
 - Centralizado
 - Integración de vistas
- Estrategias para diseño del esquema conceptual:
 - Hacia-abajo (Top-down)
 - Hacia-arriba (Bottom-up)

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
7



Diseño de Procesos

- En paralelo con diseño del esquema conceptual: realimentación
- Independiente del SGBD y en un estadio temprano dentro del proceso
- Diseño de procesos: transacciones y procesos diferidos
- La BD debe contener toda la información para soportar los procesos
- Se presta principalmente atención a las transacciones
- Los procesos diferidos pueden ser críticos en BD voluminosas

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
8

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

3. Diseño Lógico

- Definir el EC y EE en el SGBD elegido, a partir del EC de la fase anterior
- Fases
 - Diseño lógico estándar
 - Diseño lógico específico
- Resultado
 - Conjunto de sentencias LDD del SGBD elegido
 - Detalles de diseño físico, si ha lugar

Las sentencias se completan al finalizar el diseño físico

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
9

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Diseño Lógico específico

- LDD en SQL2
 - CREATE SCHEMA <nombrebd> AUTHORIZATION <usuario>;
 - CREATE DOMAIN <nombre> AS <tipodato> [DEFAULT <valor>] [CHECK <condic>];
 - CREATE TABLE [<esquema>.]<nombre_tabla>
 - (<atrib> <tipo> | <dominio> [NOT NULL] [DEFAULT <valor>] [CHECK <condic>], ...
 - PRIMARY KEY (<atrib>,...)
 - [UNIQUE (<atrib>,...)]
 - [FOREIGN KEY (<atribo>) REFERENCES tabla(<atrib>)
 - [ON {DELETE | UPDATE} {SET NULL | SET DEFAULT | CASCADE}]]
 - [CHECK (<condicion>)]);
 - CREATE VIEW <nombre_vista> [(<lista_atributos>)] AS <consulta> [WITH CHECK OPTION];

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
10



Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

4. Estructura Física de los SGBD

- El diseño físico es claramente dependiente del SGBD
- No obstante muchos gestores poseen rasgos comunes
- Presentaremos algunos
- Gestión de la información persistente
 - Discos
 - Gestión del espacio en disco
 - Índices
 - RAID

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 11



Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Ficheros y Discos

- EL SGBD almacena la información en disco duro
- Esto repercute en el diseño del SGBD
 - READ
 - transfiere los datos desde el disco a la memoria principal (RAM)
 - WRITE
 - transfiere los datos desde la RAM al disco
 - Ambas operaciones son costosas, y deben planificarse cuidadosamente

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 12



Ficheros y Discos

- ¿Por qué no almacenar todo en la memoria principal?
 - *Coste elevado, superior al de memoria secundaria*
 - *Es volátil, y queremos que los datos sean persistentes, entre diferentes ejecuciones*
 - Almacenamiento jerárquico típico:
 - Memoria principal (RAM) para los datos usados en cada momento
 - Discos para la base de datos (almacenamiento secundario)
 - Cintas, CD, DVD, etc. para almacenar versiones de los datos (almacenamiento terciario)

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
13



Discos

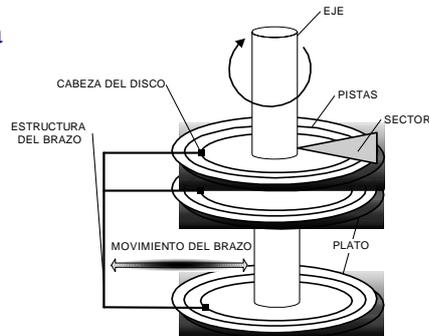
- Dispositivo de almacenamiento secundario
- Principal ventaja sobre las cintas: acceso aleatorio vs. *secuencial*
- Los datos se almacenan y recuperan en unidades llamadas bloques de disco o páginas
- Al contrario que en la RAM, el tiempo para recuperar un bloque de disco depende de su ubicación en el disco

Por tanto, la ubicación relativa de las páginas en el disco es un factor muy importante en el rendimiento del SGBD

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
14

Componentes de un disco

- Discos giratorios (ej. 90rps)
- El “arm assembly” se mueve hacia dentro o hacia fuera para situar la cabeza en la pista deseada. Las pistas bajo una misma cabeza forman un cilindro (imaginario)
- Solo una cabeza lee/escribe al mismo tiempo y en un plato
- El tamaño del sector es un múltiplo del tamaño del bloque/página (que es fijo)



Acceso a las Páginas de Disco

- Tiempo de acceso (read/write) a un bloque de disco:
 - *Tiempo de posicionamiento:*
 - movimiento del brazo para posicionar la cabeza lectora en la pista
 - varía entre 1 y 20 mseg
 - *Tiempo de rotación*
 - espera a que el bloque rote justo debajo de la cabeza lectora
 - varía entre 0 y 10 mseg
 - *Tiempo de transferencia*
 - transferencia de los datos a/desde disco
 - está en torno a 1mseg para páginas de 4KB
- Clave para bajar el coste de I/O:
 - Reducir las esperas de posicionamiento/rotación



Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Colocación de las páginas en disco

- Concepto de ‘Siguiete’ bloque:
 - bloques en la misma pista, seguido por
 - bloques en el mismo cilindro, seguido por
 - bloques en cilindros adyacentes
- Los bloques de un fichero deben colocarse secuencialmente en el disco (por ‘Siguiete’), para minimizar la esperas (posicionamiento y rotación)
- Para una búsqueda secuencial, la precarga (*pre-fetching*) de varias páginas al tiempo es un avance importante

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 17



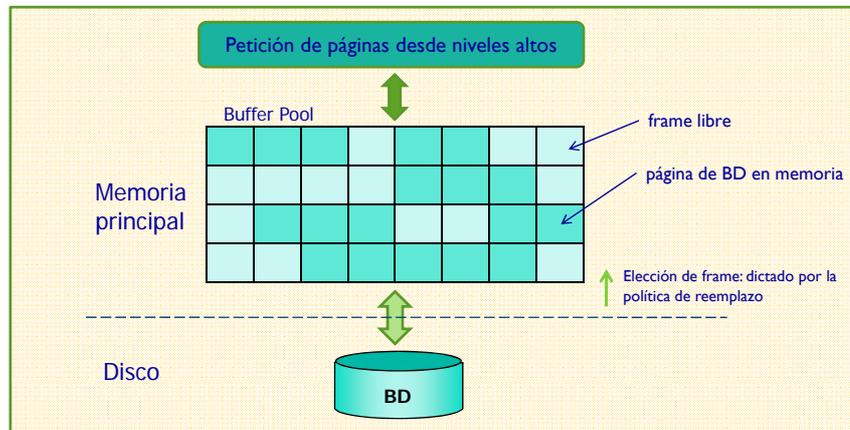
Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Gestión del espacio en disco

- El nivel mas bajo del SGBD gestiona el espacio en disco
- Los niveles mas altos llaman a esta capa para:
 - cargar/descargar una página
 - leer/escribir una página
- Una petición para una secuencia de páginas debe realizarse colocando las páginas secuencialmente (contiguas) en el disco
- Los niveles altos no necesitan conocer como se hace esto, o como se gestiona el espacio libre

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 18

Gestión del Buffer en el SGBD



- Los datos deben estar en la RAM para que el SGBD pueda operar con ellos
 - Se gestiona una tabla de correspondencia: <frame#, pageld>
- ↑ Memoria principal ↑ Disco

Solicitud de página

- Si el SGBD solicita una página que no está en el Buffer Pool:
 - Elegir un frame para reemplazarlo. Si el frame fue modificado desde su carga (estado *dirty*), se escribe en el disco
 - Cargar la página requerida en el frame elegido
 - Fijar la página y devolver su dirección
- El solicitante de una página debe liberarla, e indicar si la página fue modificada (bit de marcado *dirty*)
- Las páginas del pool pueden ser solicitadas varias veces,
 - Se utiliza un contador (“pin count”)
 - Una página es candidata para ser reemplazada si: “pin count” = 0
- Si se puede predecir la petición (p.e., recorrido secuencial) puede realizarse una precarga (pre-fetched) de varias páginas al tiempo

SGBD vs. Sistema de Ficheros del SO

- Diferencias entre los SO
 - Aspectos de portabilidad
- Un fichero no es habitual que ocupe varios discos
- La gestión de los buffers en SGBD requiere forzar dónde va una página en el disco
- Las páginas deben estar contiguas no importando desperdiciar espacio
- Tipos de ficheros
 - Registros de longitud fija
 - Registros de longitud variable

Contenido
 Proceso Diseño
 D. Conceptual
 Diseño Lógico
Estructura Física
 Diseño Fisico
 Ejemplo SGBD
 Puesta en marcha

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
21

Registros de Longitud Fija

- La información sobre los tipos de campos (y longitud en bytes) está almacenada en el *catálogo del sistema*
- Encontrar el campo “n” supone recorrer el registro

Tabla del catálogo SYSCOLUMNS:

colname	char(18)	nombre de la columna
tabid	integer	id. de la tabla
colno	smallint	nº de columna
coltype	smallint	tipo de datos de la columna
collength	smallint	longitud (física) de la columna

➔

colname	tabid	colno	coltype	collength
IdCliente	2	1	I	2
Nombre	2	2	0	15
FechaAlta	2	3	7	4

Tabla de la base de datos:

```
CREATE TABLE Clientes
(IdCliente Smallint Not Null,
Nombre Char(15) Not Null,
FechaAlta Date,
PRIMARY KEY (IdCliente));
```

➔

Fichero de datos

101Ana Tomia	05052005
102Nestor Tilla	09092009
103Carmelo Cotón	03032010
104Leandro Gao	03032010

➔

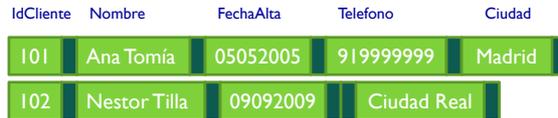
102 (2 bytes) = 00000000 00110111
 Nestor Tilla (15b): 1 caracter=1 byte
 09092009 (4b): almacena número
 Dirección Campo 3 (Registro2) = 21 + 2 + 15

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
22

Registros de longitud variable

- Formatos

- Separador de campos



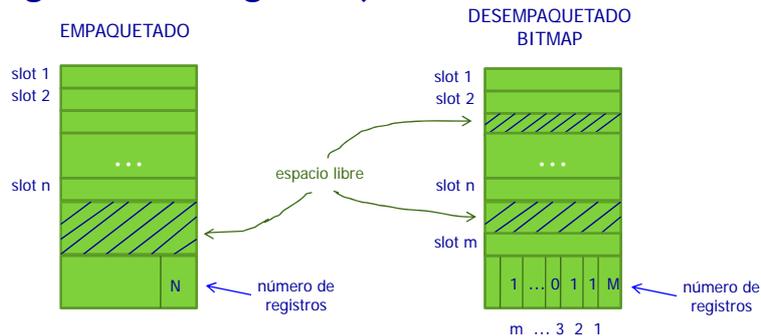
- Directorio de registro

- Acceso directo al campo "i"
- Eficiencia al almacenar valores nulos
- Sobrecarga por gestión del directorio de registro



Formato de Página

- Registro de longitud fija

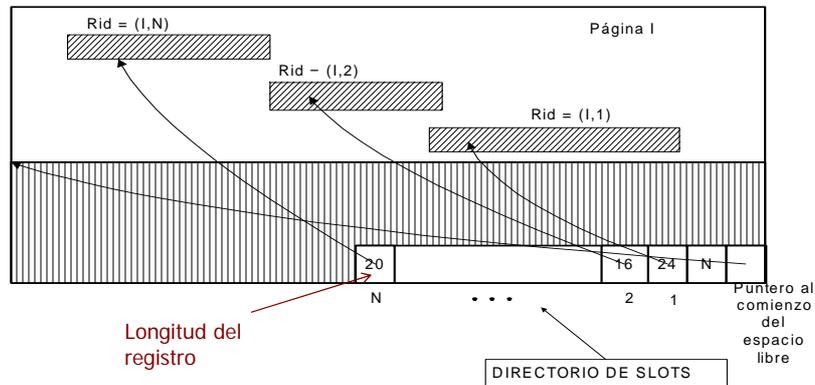


- RID: Record id = <page id, slot #>

- En la primera alternativa, el movimiento o empaquetamiento de los registros para gestionar el espacio libre de la página puede variar el "RID", y esto en algunos casos puede no ser aceptable

Formato de Página

- Registro de longitud variable



- Se pueden mover registros en la página sin cambiar su “Rid”
- Esto también es interesante para registros de longitud fija

Ficheros de Registros

- Una página o bloque es aceptable en operaciones de entrada/salida, pero los niveles mas altos del SGBD operan con registros, y ficheros de registros
- Fichero
 - colección de páginas, conteniendo cada una, una colección de registros
 - Debe permitir:
 - Inserción, modificación y borrado de registros
 - Leer un registro concreto (con su identificador de registro: rid)
 - Recorrer todos los registros (con posibilidad de utilizar condiciones en los registros a ser recuperados)

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Índices

- Un fichero (pila) permite recuperar registros
 - Especificando el *rid* (registro Id), o
 - Recorriendo los registros secuencialmente
- Algunas veces, queremos recuperar registros especificando valores de uno o mas campos, p.e.,
 - Obtener los estudiantes del departamento 'OEI'
 - Obtener los estudiantes con curso > 2
- Los índices son estructuras de fichero que permiten resolver estas consultas basadas en valores, de forma eficiente

2010 © Pedro P. Alarcón, Juan Garbajosa

Administración de Bases de Datos – EU informática (UPM)

27

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Estructura de índices ISAM

- Operación
 - Creación del Fichero
 - La páginas hoja (de datos) se colocan secuencialmente ordenadas por la clave de búsqueda; se asignan las páginas de índice, y el espacio para páginas de overflow
 - Entradas del índice
 - <valor clave de búsqueda, pagina id>
 - Permiten la búsqueda "directa" de las entradas de datos, que están en las páginas hoja
 - Búsqueda
 - Empieza en root, usa comparación de claves hasta llegar a la hoja
 - Inserción
 - Encontrar la hoja correspondiente, y ponerlo ahí
 - Borrado
 - Encontrar y borrar la hoja; si la página de overflow está vacía, desasignarla
 - Estructura de árbol estática
 - Las inserciones y borrados afectan únicamente a páginas hoja
- El interés del ABD radica en el número de E/S que implica cada operación: por ejemplo, estimar tiempos (medios) de cada operación

Páginas de índices

Páginas de datos

Páginas de overflow

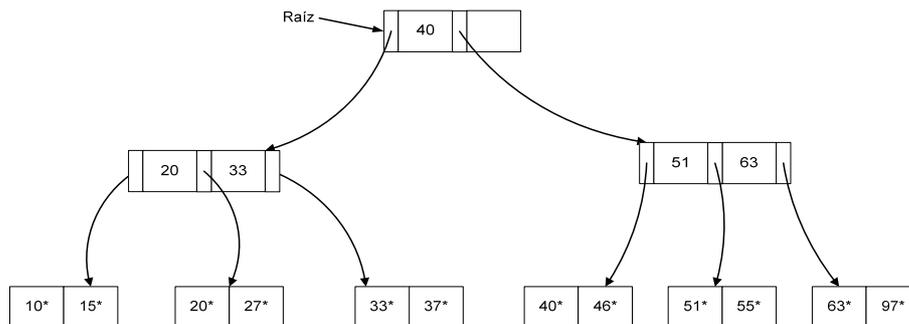
2010 © Pedro P. Alarcón, Juan Garbajosa

Administración de Bases de Datos – EU informática (UPM)

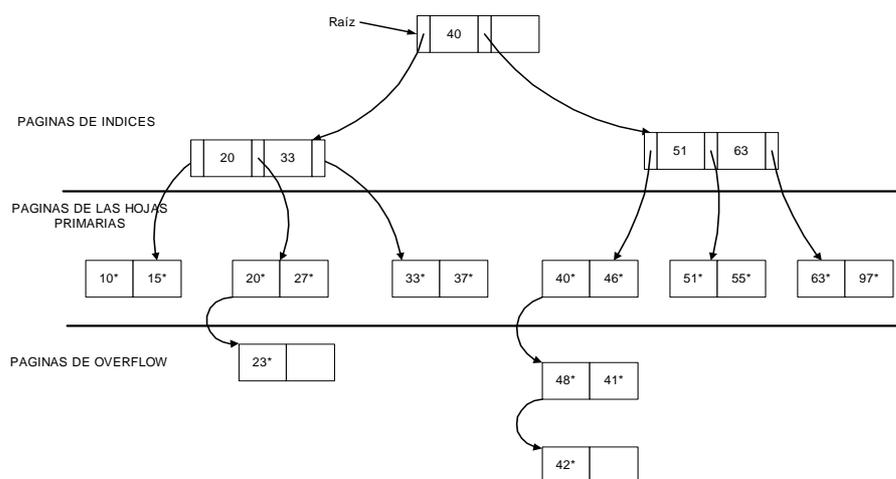
28

Ejemplo de árbol ISAM

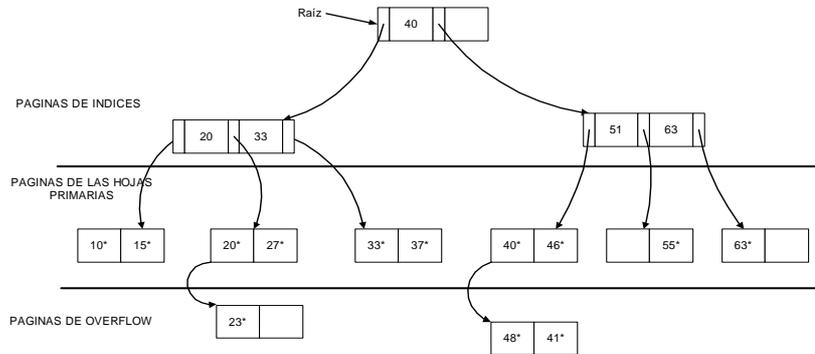
- Cada nodo puede tener 2 entradas



Tras insertar 23*, 48*, 41*, 42* ...



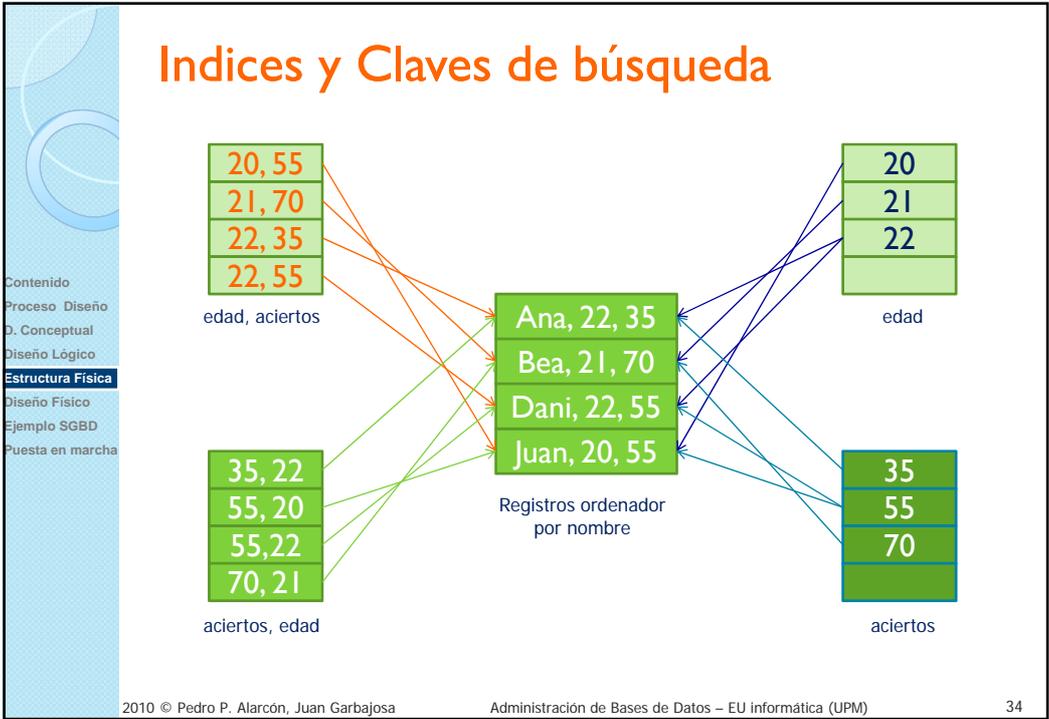
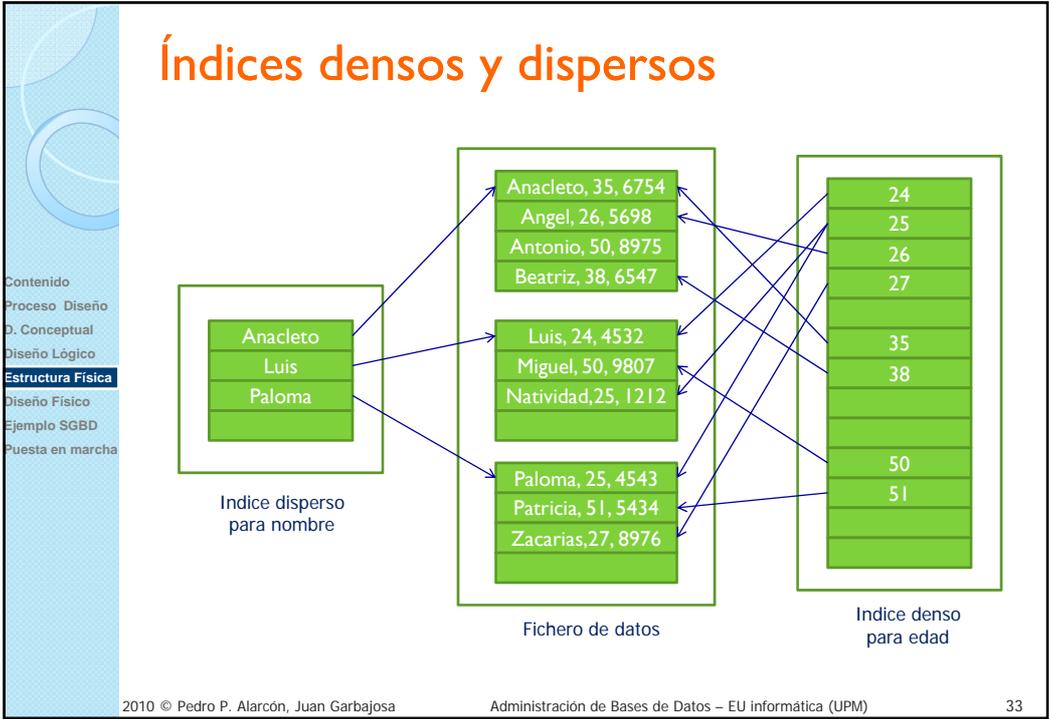
....Al borrar 42*, 51*, 97*



- el 51* aparece en nodo intermedio pero no en nodo hoja

Índices densos y dispersos

- *Denso vs. Disperso*
 - Si hay por lo menos una entrada para el valor de clave de búsqueda (en algún registro de datos), entonces Denso
 - Cada índice Disperso está agrupado
 - Los índices Dispersos son mas pequeños; sin embargo, algunas optimizaciones útiles se basan en índices Densos
- Disperso (direcciona páginas)
 - Ordenación por claves principales de una tabla, si las tuplas están físicamente ordenadas por ese campo
- Denso (direcciona registros)
 - Ordenación por otro campo, p.ej. índices secundarios



Almacenamiento RAID

- RAID:
 - Redundant Arrays of Inexpensive/Independent Disks
- Características
 - Combina varios discos duros para formar una única unidad lógica en la que almacenar los datos de forma conjunta
 - Mayor tolerancia a fallos y mayor rendimiento que un solo disco o un grupo de discos independientes
- Objetivos
 - Aumentar el rendimiento y la fiabilidad
 - MTTF (Mean Time To Failure)
 - 1 disco = 200.000 horas <> 22,8 años
 - Banco de 100 discos = 2.000 horas <> 83,3 días
- Dos técnicas principales
 - Data striping
 - Los datos están particionados. El tamaño de una partición se llama "striping unit".
 - Las particiones son distribuidas entre varios discos.
 - Mejora el rendimiento
 - Redundancia
 - Mas discos -> mas fallos.
 - La información redundante permite reconstruir los datos si un disco falla.
 - Mejora la fiabilidad

Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Niveles RAID

- Nivel 0: Sin redundancia
- Nivel 1: *Mirrored* (dos copias idénticas)
 - Cada disco tiene una imagen espejo (*check disk*)
 - Lecturas paralelas. Una escritura involucra dos discos
 - Máxima tasa de transferencia = tasa de transferencia de un disco
- Nivel 0+1: Striping and Mirroring
 - Lecturas paralelas. Una escritura involucra dos discos
 - Máxima tasa de transferencia = ancho de banda conjunto (*aggregate bandwidth*)

Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Niveles RAID

- Nivel 3: Paridad de bit intercalado
 - Un byte se “rompe” en bits
 - Cada lectura y escritura involucra todos los discos;
 - la matriz de discos puede procesar una petición cada vez
- Nivel 4: Paridad de bloque [de fichero] intercalado
 - Un bloque de fichero se “rompe” en varios discos
 - Un disco espejo (comprobación)
 - Lecturas paralelas para pequeñas peticiones; las grandes utilizan todo el ancho de banda
 - La escritura incluye bloques modificados y disco espejo
- Nivel 5: Paridad de bloque intercalado y distribuidos
 - Similar al nivel 4 de RAID, pero los bloques de paridad están distribuidos entre todos los discos

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
37

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Niveles RAID

- Cada nivel de RAID ofrece una combinación específica de tolerancia a fallos (redundancia), rendimiento y coste

El diagrama muestra un triángulo con vértices etiquetados como 'rendimiento' (arriba), 'disponibilidad' (derecha) y 'coste' (abajo). Dentro del triángulo se sitúan los niveles RAID: RAID 0+1 en la parte superior; RAID 0 en el centro; RAID 3 y RAID 5 en la parte inferior izquierda; RAID 1 en la parte inferior derecha; y JBOD en la base izquierda.

JBOD (Just a Bunch Of Disks):
Grupo de discos en un volumen lógico, sin mejora de rendimiento ni disponibilidad

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
38



Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Diseño Físico

- Objetivos
- Determinación del espacio de almacenamiento
- Técnicas para acelerar operaciones de unión natural (sistemas relacionales)
- Elección de caminos de acceso (índices)
- Producción del Esquema Interno
- Monitorizaciones

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 39



Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Objetivos del Diseño Físico

- Optimizar tiempos de respuesta
 - Definición
 - Factores que influyen
- Minimizar espacio de almacenamiento para los ficheros físicos de la BD
- Optimizar rendimiento de transacciones (throughput)
- Proporcionar procedimientos óptimos de recuperación e integridad de la BD
- Asegurarse que los requisitos y criterios de seguridad y confidencialidad se cumplen

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 40



Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Determinar espacio de almacenamiento

- Determinar los factores propios de cada fichero
- Establecer tamaño (Kb o Mb) de las estructuras lógicas
- Definir ocupaciones iniciales
- Definir modo de gestionar e incrementar el espacio
- Establecer correspondencia entre estructuras lógicas y ficheros físicos

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 41



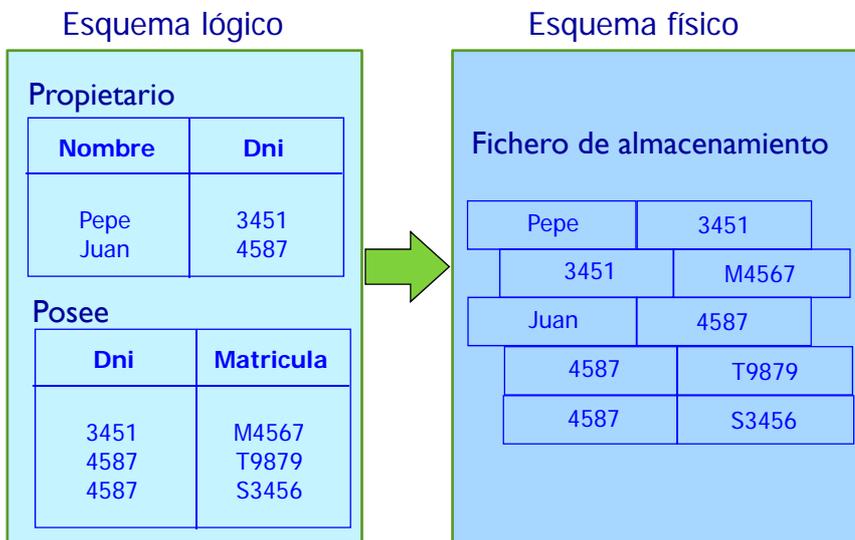
Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

Técnicas para acelerar la unión natural

- Agrupamiento (clustering)
 - Proporciona eficiencia en uniones naturales para relaciones 1:N
- Desnormalizar el esquema lógico
 - Se compensa con transacciones que mantengan la consistencia de la BD o con procesos off-line
- Particionamiento
 - Vertical
 - Horizontal

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 42

Ejemplo de clustering



Elección de caminos de acceso (índices)

- Consideraciones
 - Para cada consulta, elegir campos candidatos para crear índices
 - Para cada transacción de actualización, elegir campos candidatos para crear índices
 - Considerar frecuencias de ejecución
 - Considerar restricciones de tiempo
- Normas generales para elegir caminos de acceso
 - Claves primarias
 - Claves foráneas
 - Atributos de uso frecuente
 - Frecuencia actualizaciones



Elección del tipo de índice

- Pilas
 - Por cada operación se cargan varias tuplas a la vez
 - Número relativamente bajo de tuplas
 - Hay que recuperar todas las tuplas para cada operación
 - Adecuada para una tabla cargada permanentemente en memoria
- Hash
 - Útil cuando la búsqueda se basa en clave completa
- Índices: patrón, valor exacto, rangos y patrones de clave
 - Índices de uno o varios niveles

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
45



Elección del tipo de índice

- ISAM
 - Estático
 - Creado al tiempo que el fichero de datos
 - Su rendimiento se deteriora conforme se actualiza la tabla
 - Las actualizaciones hacen perder la secuencia de acceso a la clave
- Árboles B⁺
 - El rendimiento no se deteriora con el uso
 - Mantiene el orden de las claves cuando se actualiza
 - Si las actualizaciones no son frecuentes el rendimiento puede ser peor por tener un nivel más de índice
- Clusters: Índices y Tablas
 - Recuperación de tuplas anidadas

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
46



Producción del Esquema Interno

- Sintaxis SQL para índices
 - Creación de índice
 - CREATE [UNIQUE] INDEX < index name >
ON < table name > (< column name > [< order >]
 { < column name > [< order >] })

[CLUSTER]
 - Borrado
 - DROP INDEX < index name >
- Sentencias específicas del SGBD para reserva de espacio

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
47



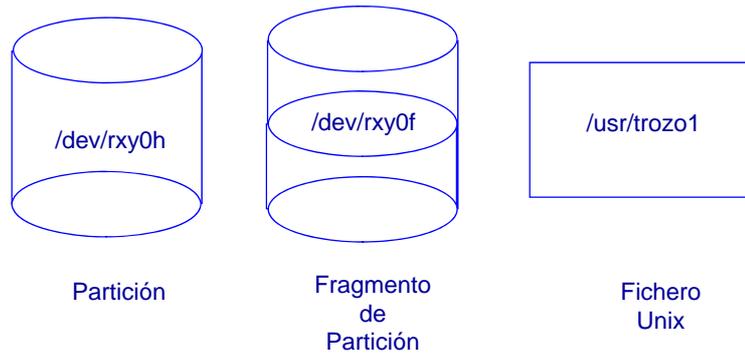
Monitorizaciones de la BD

- Permiten obtener estadísticas de utilización:
 - Se almacenan en el catálogo o DD
 - Incluyen, entre otras:
 - nº de invocaciones de transacciones predefinidas
 - actividades de E/S en ficheros
 - nº de páginas o bloques de ficheros de datos, índices u otros
 - frecuencia de utilización de índices
 - Con esta información se puede afinar el sistema

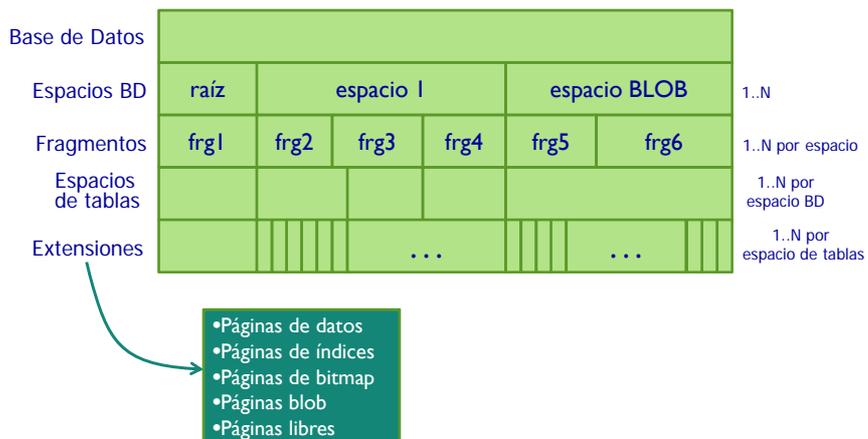
2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
48

6. Ejemplo de SGBD

- Espacio de almacenamiento



Esquema de una BD



Espacio físico y lógico

- Espacio Físico (contiguo)
 - Páginas
 - Extensiones
 - Fragmentos
 - Páginas especiales: BLOB, Bitmap ...
- Espacio Lógico (colección de espacios físicos)
 - Espacios de base de datos
 - formado por una colección de fragmentos, de 1 a N
 - Espacios de tablas
 - colección de todas las extensiones asignadas a una tabla
 - el espacio no es necesariamente continuo (en una extensión si)
 - Espacios especiales: BLOB, Bitmap...
- Cada tabla tiene su espacio de tabla con todos sus datos
- En el espacio raíz pueden estar los ficheros “logical logs” y “physical logs”

2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
51

Ejemplo de estructura de una página

Página #

The diagram illustrates the structure of a database page. At the top is the 'Cabecera de Página' (Page Header) containing '# bytes libres' (free bytes) and 'próximo slot libre' (next free slot), followed by a 'Timestamp'. Below the header are several 'Fila' (Row) blocks. A large green block represents 'espacio libre en la página' (free space in the page). At the bottom is the 'Pie (tabla de slots)' (Footer / Slot Table), which consists of a table of slots (4 bytes per slot, 1 slot per fila) and a 'Timestamp'. Below the footer, a legend defines the fields: 'página #' (3 bytes), 'slot #' (1 byte), 'offset (despl.) de fila' (2 bytes), and 'tamaño fila' (2 bytes). Arrows indicate the mapping between these fields and the corresponding parts of the page structure.

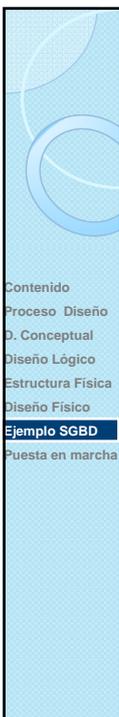
2010 © Pedro P. Alarcón, Juan Garbajosa
Administración de Bases de Datos – EU informática (UPM)
52



Ejemplo Sintaxis Creación Esquema BD

- **CREATE DATABASE nombreBD**
 [IN nombredbs]
 [WITH {[BUFFERED] LOG | LOG MODE ANSI}]
 - Por defecto se crea en el “rootdbs”, pero no es recomendable
 - nombredbs: dbspace que contendrá el catálogo del sistema para la BD
 - WITH [BUFFERED] LOG: para ejecución de transacciones (commit,rollback,..)
 - Sin BUFFERED se escribe directamente en disco (más rápido, menos seguro)
 - LOG MODE ANSI, funciona sin BUFFERED
- **DROP DATABASE nombre**

2010 © Pedro P. Alarcón
Administración de Bases de Datos – EU Informática (UPM)
53



Ejemplo Sintaxis Creación Esquema BD

- **CREATE [TEMP] TABLE (<especificacion_de_columna>)**
 [WITH NO LOG]
 [IN <nombredbpace>]
 [EXTEND SIZE <tamaño>]
 [NEXT SIZE <tamaño>]
 [LOCK MODE (PAGE|ROW)]

<especificacion_de_columna>:=
 (<nombrecolumna> {<tipodato>{BYTE|TEXT} [IN {TABLE|<blobpace>}}]
 [NOT NULL] [UNIQUE [CONSTRAINT <nombreconstr>]] [,]
 [UNIQUE (<listacolumnas>) [CONSTRAINT <nombreconstr >]] [, ...]

2010 © Pedro P. Alarcón
Administración de Bases de Datos – EU Informática (UPM)
54

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Ejemplo Sintaxis Creación Esquema BD

- ALTER TABLE <nombre_tabla>
 - [<añadir_modificar_eliminar_columnas>]
 - [<añadir_modificar_eliminar_restricciones>]
 - [MODIFY NEXT SIZE <tamaño>]
 - [LOCK MODE (PAGE|ROW)]

se reescriben todos los datos de la tabla, compactando espacios libres
- DROP TABLE <nombre_tabla>
- CREATE VIEW <nombre_vista> [(<lista_columnas>)]
 - AS <consulta>
 - [WITH CHECK OPTION]
- DROP VIEW <nombre_vista>

2010 © Pedro P. Alarcón Administración de Bases de Datos – EU Informática (UPM) 55

Contenido

Proceso Diseño

D. Conceptual

Diseño Lógico

Estructura Física

Diseño Físico

Ejemplo SGBD

Puesta en marcha

Ejemplo Sintaxis Creación Esquema BD

- CREATE [UNIQUE] [CLUSTER] INDEX <nombre_indice>
 - ON <nombre_tabla> (<columna I> [ASC|DESC],)
- Índices cluster:
 - solo se puede tener uno por tabla
 - almacena físicamente las filas en el orden de la clave del índice
 - no se actualiza (reorganiza) al insertar o eliminar filas en la tabla
 - para reorganizar, eliminar el índice y volver a crearlo
- DROP INDEX <nombre_indice> TO <nombre_tabla>

2010 © Pedro P. Alarcón, Juan Garbajosa Administración de Bases de Datos – EU informática (UPM) 56

Ejemplo Sintaxis Creación Esquema BD

- Tipos de datos:
 - CHAR [(n)], VARCHAR
 - SMALLINT
 - INTEGER
 - DECIMAL [(m[,n])]
 - MONEY [(m[,n])]
 - SMALL FLOAT, FLOAT [(n)]
 - SERIAL [(n)]
 - DATE, DATETIME : fecha y hora
 - TEXT contiene texto, sin límite
 - BYTE contiene objetos binarios

Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

7. Puesta en marcha de una BD

- Creación de la BD diseñada con la asignación de espacio definida
- Carga de datos
 - nueva creación: introducción de datos
 - ya existente: conversión de datos
- Creación de índices
 - únicos: se *pueden* crear antes para comprobar unicidad
 - duplicados: después de la carga
 - cluster: antes de la carga
- Comprobar validez del espacio asignado, y del crecimiento esperado
- Comprobar subsistema de copia/recuperación
- Ejecutar baterías de pruebas de aceptación
- Revisar existencia y validez de documentación y procedimientos

Contenido
Proceso Diseño
D. Conceptual
Diseño Lógico
Estructura Física
Diseño Físico
Ejemplo SGBD
Puesta en marcha

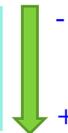
Ejemplo de estimación de espacio

- Una base de datos tiene un área con 1.000.000 de registros. Realizar una estimación del espacio necesario para crear la base de datos suponiendo un crecimiento del 15% anual y cinco años de vida. La cabecera de registro ocupará 30 bytes, y la cabecera y el pie de página 64 cada uno. Cada registro de datos tiene 275 bytes. Suponer páginas de diferentes tamaños (1024, 2048, 4096, 8192).
- Calcular el tiempo medio para acceder a un registro utilizando la clave primaria y la clave secundaria suponiendo que las páginas son de 2048 bytes. El índice principal tiene 20 bytes para el rango de claves y 10 para la dirección de la página. Un índice secundario tiene 10 bytes para la clave y 10 bytes para la dirección.
- Se debe suponer el tiempo de posicionamiento 30ms, y el tiempo de acceso secuencial en disco 5 ms. Para los cálculos el llenado es del 70%, y la base de datos está uniformemente distribuida en la superficie del disco.

Resumen

- Se ha analizado el papel y relevancia del ABD

- Análisis
- Diseño lógico y Diseño físico
- Implementación y Carga
- Puesta en marcha



- Se ha visto cómo

- Estimar parámetros de reserva de espacio
- Técnicas básicas para minimizar tiempo de acceso a los datos

- Queda por ver métodos y procedimientos para

- Garantizar seguridad (confidencialidad), integridad y recuperación
- Optimizar tiempos de respuesta y throughput de transacciones