

Bases de Datos II

Almacenamiento y Estructura de archivos

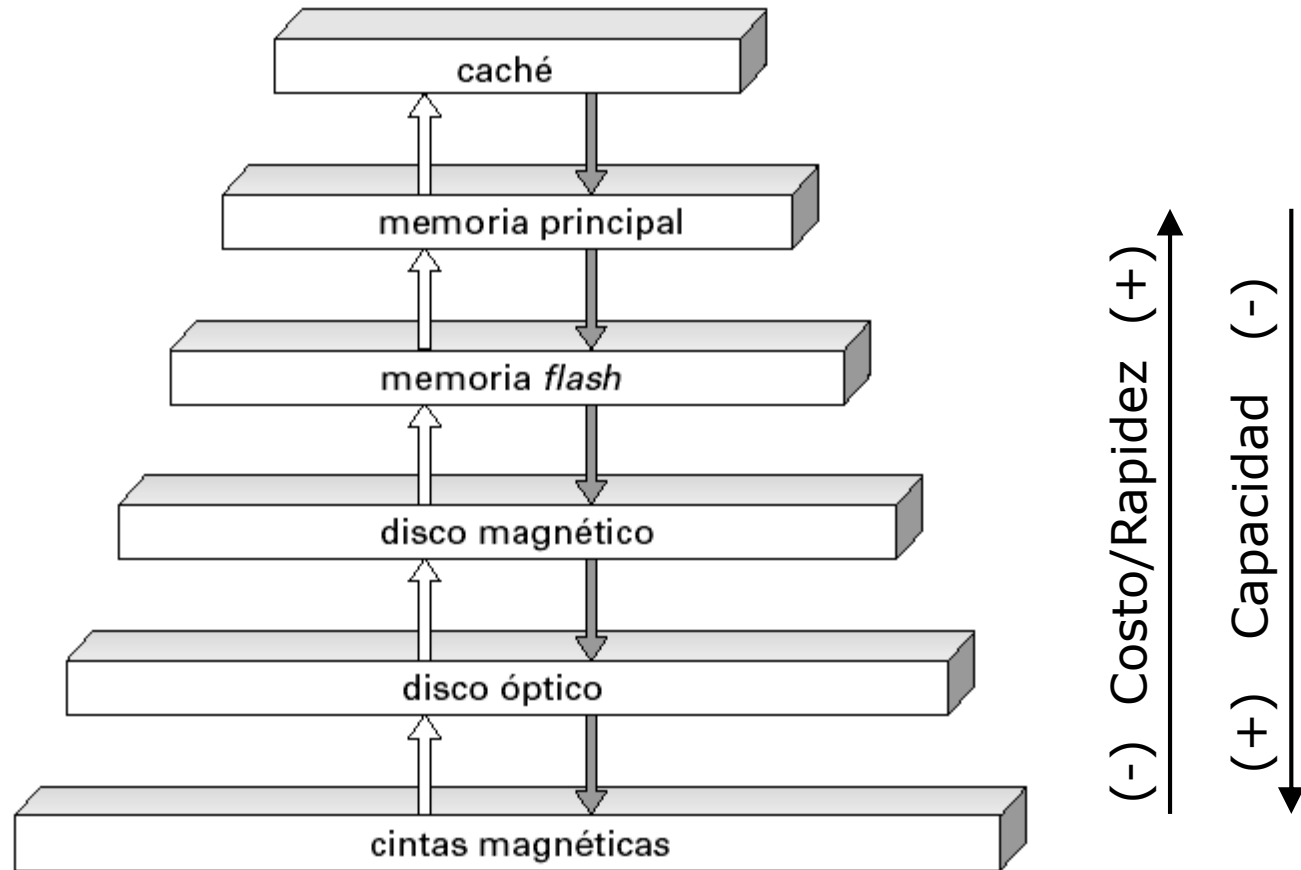


Ingeniería en Informática
Facultad Politécnica – UNA
Ing. Joaquín Lima

Clasificación de los Medios de Almacenamiento

- ❑ Velocidad de acceso a los datos
- ❑ Costo por unidad de datos
- ❑ Confiabilidad
 - ❑ Pérdida de datos por fallo de energía o del sistema
 - ❑ Pérdida de datos por fallos del dispositivo
- ❑ Tiempo de almacenamiento:
 - ❑ Volátil.
 - ❑ Los datos se pierden cuando se corta la energía
 - ❑ No Volátil:
 - ❑ Los datos persisten aun cuando se corta la energía.
- ❑ Capacidad

Medios de Almacenamiento



Medios de Almacenamiento

□ **Cache**

- Memoria muy costosa de poca capacidad
- Muy Rápido Acceso.
- \sim Velocidad del Procesador.
- Volátil

□ **Memoria RAM**

- Rápido Acceso
- Muy pequeña como para almacenar una BD de propósito general.
- Volátil
- Costo elevado

Medios de Almacenamiento

❑ Memoria Flash

- No Volátil. Tecnología EEPROM
- Lectura casi tan rápida como las memorias RAM o caché.
- Escritura lenta, para escribir hay que rescribir todo un bloque.
- Costo relativamente barato con respecto a las anteriores (gracias a la demanda).
- Vida útil reducida, entre 1 y 5 millones de ciclos de lectura-escritura.
- **Unidad de Estado Sólido / SSD**
 - ❑ **Capacidad Limitada. Actualmente 128Gb max.**

Medios de Almacenamiento

❑ Discos Magnéticos

- Ampliamente utilizados
- Baratos. Mejor relación costo/capacidad.
- Gran Capacidad. Actualmente rondando varios TB
- Acceso Directo. Lectura y escritura en cualquier orden.
- No Volátil. Capaz de sobrevivir a fallas de energía y errores del Sistema
- Lectura y escritura considerablemente inferior a los anteriores.

Medios de Almacenamiento

❑ Discos Ópticos

- No Volátiles
- Baratos. Formas populares son el CD y el DVD. Escritura única, Múltiples lecturas (WORM).
- Existen versiones re-escribibles CD-RW y DVD-RW/+RW/RAM.
- Lecturas y escrituras mas lentas que los discos magnéticos.
- Sistemas Juke-box. Permiten tener múltiples discos cambiables automáticamente para guardar grandes volúmenes de datos.

❑ Cintas Magnéticas

- Muy Lentas. Solo de Acceso Secuencial.
- Útiles como medio de Backup en sistemas Juke-box de hasta 1 PB

Jerarquía de Medios de Almacenamiento

□ Primario

- Rápidos pero volátiles
- Cache, RAM
- Utilizado como medios de almacenamiento temporal

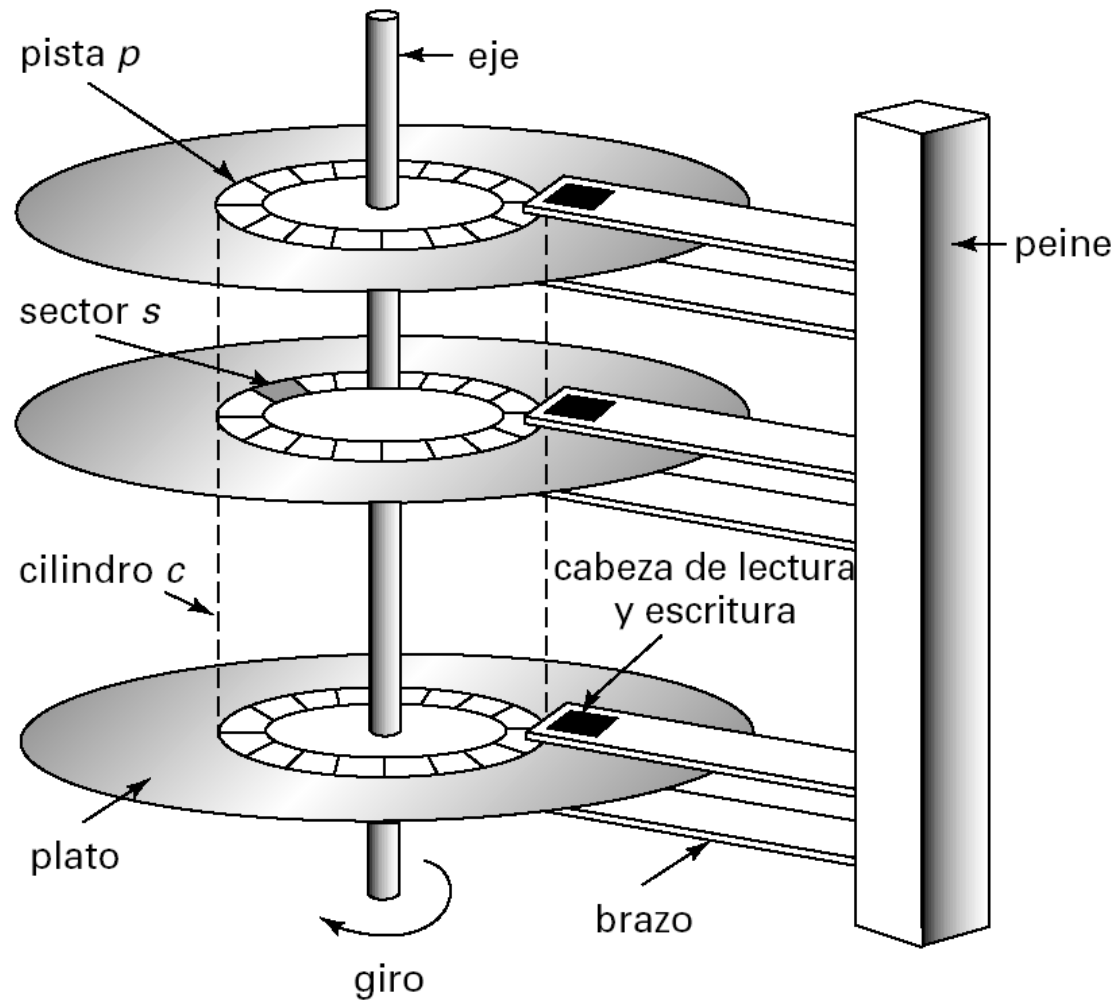
□ Secundario

- Medianamente rápidos y no volátiles
- Memorias Flash, Discos Magnéticos y Ópticos.
- Utilizados como medio de almacenamiento permanente en línea (on line)

□ Terciario

- No Volátiles pero de muy lento acceso
- Utilizados para backup (off line).

Discos Magnéticos



Discos Magnéticos

- ❑ Platos
 - Consisten en piezas de vidrio impregnadas con material magnético que almacena la información. Pueden haber varios por disco montados en un solo eje, típicamente de 1 a 5
- ❑ Cabezas de Lectura/Escritura
 - Leen y escriben información en forma de codificación magnética
- ❑ Pistas
 - La superficies de los discos están divididas en pistas, en la cuales se almacena la información.
- ❑ Sector
 - Consiste en la unidad de almacenamiento más atómica en un disco.
 - Su tamaño por lo general es de 512 B.
- ❑ Cilindro
 - Es el conjunto de todas las pista i en los platos del disco.
- ❑ Brazos
 - Soportes que mantienen los cabezales sobre los discos.
- ❑ Peine
 - Soporte sobre el cual van ensamblados los brazos.

Discos Magnéticos

- ❑ Controladores de disco.
 - (P)ATA (IDE), SATA, SCSI
 - HW que implementa la lógica de comunicación entre el sistema y el disco.
 - Por lo general se encarga de:
 - ❑ Aceptar comando de lectura y escritura de sectores o bloques
 - ❑ Controlar la mecánica del disco. Iniciar el movimiento del disco, ubicar y accionar los cabezales, iniciar la escritura o lectura de datos
 - ❑ Computar sumas de comprobación de manera a verificar si la información se ha leído correctamente.
 - ❑ Asegurar la correcta escritura releiendo los datos escritos
 - ❑ Realizar el mapeamiento de sectores defectuosos.

Discos Magnéticos

■ Medidas de Rendimiento:

■ Tiempo de Acceso:

- Tiempo que toma leer o escribir datos desde su petición hasta su confirmación.
- Tiempo de búsqueda: Tiempo que se tarda en ubicar el peine en la pista correcta. 4 a 10 ms.
- Latencia Rotacional: Tiempo que tarda en aparecer el sector a escribir/leer debajo de los cabezales. 4 a 15 ms.

■ Taza de Transferencia

- Volumen de datos por unidad de tiempo que se pueden guardar o recuperar del disco.
- Típicamente de 30 a 100 MB/s para lectura
- 20 a 85 MB/s para escritura.

■ Tiempo medio de fallo (TMDF):

- Lapso de tiempo que un solo disco puede trabajar continuamente sin fallas. De 3 a 5 años.

Optimización del Acceso al Disco

□ Bloques

- Mínima cantidad de sectores transferidos entre el disco y la memoria del computador
- Bloque pequeños provocan mayor cantidad de operaciones de transferencia.
- Bloques grandes desperdician memoria del computador.
- Tamaños típicos entre 4 a 16 KB.

□ Planificación de Brazos del Disco.

- Algoritmos que se encargan de planificar el movimiento del peine para maximizar la cantidad de solicitudes atendidas por unidad de tiempo.
- Basados en el algoritmo del ascensor: mover el peine en una dirección hasta atender todas la solicitudes en dicha dirección y luego repetir la operación en dirección inversa.
- Se implementan en el controlador de disco.

Optimización del Acceso al Disco

❑ Organización de Archivos

- Reorganizar los bloques correspondientes a un archivo para mejorar el tiempo de acceso.
- Guardar los datos de un archivo en el mismo cilindro o en cilindros contiguos.
- Problema: Fragmentación de los archivos.
 - ❑ El archivo cambia continuamente provocando que sus datos queden esparcidos por el disco
 - ❑ Nuevos archivos deben ser almacenados en bloques muy separados por la fragmentación existente.
 - ❑ Acceso a estos archivos provoca un incremento del movimiento del peine
- Solución: Utilizar herramientas que desfragmentan el disco. Su uso debe realizarse cuando el disco esté desocupado.

Optimización del Acceso al Disco

- ❑ Memoria Intermedia de Escritura
 - Utiliza un memoria No Volátil: RAM o Flash
 - Permite almacenar cierta cantidad de operaciones de escritura al disco.
 - Las operaciones pueden ser reordenadas para minimizar el movimiento del peine.
 - Tolerancia ante fallas de energía. Una vez restaurada la alimentación el sistema puede escribir los datos pendientes.
 - Las operaciones no necesitan esperar por la confirmación de la escritura.
- ❑ Disco intermedio de escritura
 - Utilizado exactamente como las memorias de escritura. Mas Lento/Mayor Capacidad.
 - No se requiere de planificación para el disco auxiliar.
- ❑ Sistemas de archivos con planificación
 - Sistemas de archivos a nivel de SO que reordenan las operaciones de escritura para mejorar el rendimiento del disco

Redundancia de discos

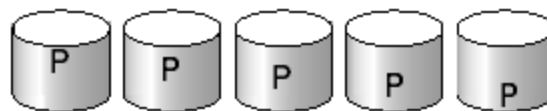
- ❑ Si el TMDF de fallo de un disco es 100.000 hs (~ 11 años), el tiempo TMDF de 100 discos trabajando juntos es de 1.000hs (~ 41 días)
- ❑ Si un disco falla los datos se pierden.
- ❑ Es posible utilizar un discos espejados. Todas las operaciones se realizan sobre los discos espejados.
- ❑ El Tiempo Promedio de Perdida de Dato (TMPD) es dependiente del TMDF y el Tiempo Medio de Reparación (TMDR)
 - Si se tiene un disco espejado con TMDF de 100.000 hs y TMDR de 10 hs, entonces el $\text{TMPD} = 100.000^2 / 2 / 10 = \sim 57000$ años. (teniendo en cuenta solo fallos independientes)

RAID

- ❑ RAID: Redundant Arrays of Independent Disks.
 - Técnica de organización de múltiples discos que provee la visión de un único disco con:
 - ❑ Alta capacidad.
 - ❑ Alta velocidad.
 - ❑ Alta confiabilidad.
- ❑ Existen 6 niveles de RAID (Tarea).
- ❑ Los mas utilizados son RAID 1 y RAID 5



(b) RAID 1: Discos con imagen



(f) RAID 5: Paridad distribuida con bloques entrelazados

Acceso al Almacenamiento

- ❑ Los archivos de una Base de Datos son almacenados en Bloques que son transferidos del disco a la memoria. El SGBD busca maximizar el uso de los bloques transferidos de manera a minimizar el tiempo empleado en acceso al disco
- ❑ Se utilizan Buffers para almacenar los bloques recuperados del disco. Existe un Administrador de Buffers que se encarga de mantener los bloques en memoria.
- ❑ El Administrador de Buffers es invocado cuando se necesita un bloque del disco. Este trabaja de la siguiente manera:
 - Si el bloque esta en memoria, retorna la dirección del bloque en memoria
 - Si el bloque no esta en memoria, reserva espacio en memoria para leer el bloque del disco
 - ❑ Si es necesario, se reemplaza un bloque en memoria para obtener espacio. El bloque a reemplazar es escrito en el disco si ha cambiado.
 - Lee el bloques del disco, lo almacena en memoria y retorna su dirección en la memoria.

Acceso al Almacenamiento

- ❑ El Administrador de Buffers debe utilizar una estrategia de reemplazo de bloques que minimice la necesidad de acceso al disco.
- ❑ La mayoría de los SO utilizan la estrategia de reemplazo LRU (último recientemente utilizado), la cual es mala para los Sistemas de Bases de Datos.
- ❑ La estrategia de reemplazo adecuada para los Sistemas de BD es MRU.
 - Se reemplaza el bloque más recientemente utilizado.
 - Cuando un SGBD termina de utilizar un bloque, este se convierte en el bloque candidato a ser reemplazado si otro bloque es requerido.
 - Existen casos para los cuales esta estrategia MRU no debe ser utilizada, que son el diccionario de datos y los índices.
 - El Administrador de Buffers puede llevar un registro estadístico combinado con una técnica heurística que determine bloques frecuentemente utilizados deben de ser mantenidos en memoria.

Organización de Archivos

- ❑ La BD puede ser almacenada en una colección de archivos, los cuales son utilizados para almacenar:
 - Tablas de usuario y sistema
 - Índices de usuario y sistema
 - Funciones y procedimientos
 - Objetos binarios
 - Etc...
- ❑ Cada archivo representa a una Tabla y esta compuesto de registros que representan las filas.
- ❑ Como los archivos están divididos en bloques en el disco, los bloques pueden contener varios registros del archivo.
- ❑ Además puede haber casos en que un registro esté ubicado en dos o más bloques.
- ❑ Los registros pueden ser de longitud fija o variable.

Organización de Archivos

□ Registros de Longitud fija

- Cada registro empieza en el byte $n * i$ del bloque, donde n es la longitud de los registros e i es el número del registro.
- Los registros son siempre agregados al final del archivo
- Se puede permitir que los registros desborden el bloque

■ Las estrategias de borrado de un registro i pueden ser puede ser:

- Mover los registros $i+1$ a n una posición atrás.
- Mover el registro n a i .
- Dejar espacios vacíos, pero manteniéndolos enlazados en una *lista libre*.

registro 0
registro 1
registro 2
registro 3
registro 4
registro 5
registro 6
registro 7
registro 8

C-102	Navacerrada	400
C-305	Collado Mediano	350
C-215	Becerril	700
C-101	Centro	500
C-222	Moralzarzal	700
C-201	Navacerrada	900
C-217	Galapagar	750
C-110	Centro	600
C-218	Navacerrada	700

Organización de Archivos

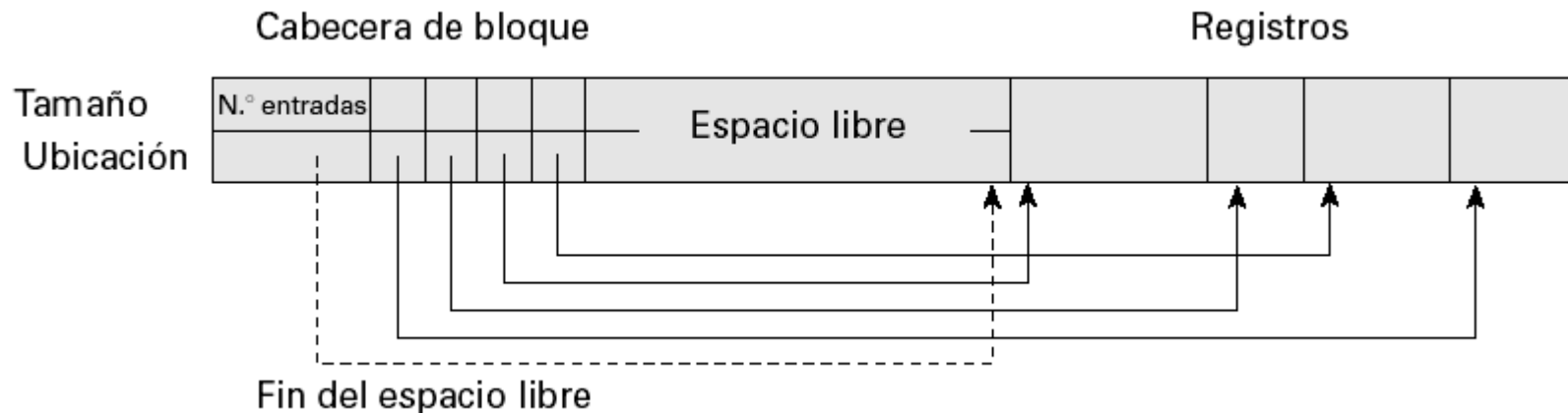
▣ Lista Libre

- Se utiliza el primer registro del bloque como cabecera del mismo. Allí se guarda entre otros datos el primer registro libre del archivo.
- Cada registro libre tiene un puntero que apunta al siguiente registro libre

cabecera				
registro 0	C-102	Navacerrada	400	
registro 1				
registro 2	C-215	Becerril	700	
registro 3	C-101	Centro	500	
registro 4				
registro 5	C-201	Navacerrada	900	
registro 6				
registro 7	C-110	Centro	600	
registro 8	C-218	Navacerrada	700	

Organización de Archivos

- ❑ Registros de longitud variable:
 - Son utilizados en BD en diferentes casos:
 - ❑ Cuando se requiere guardar múltiples tipos de registros en una misma tabla.
 - ❑ Cuando los registros contienen uno o más campos cuya longitud es variable.
 - ❑ Cuando la tabla permite campos repetitivos.



Organización de Archivos

- Considerar registros de longitud variables del siguiente tipo:

- **type** *cuentas* = **record**
 sucursal : char (22);
 cuentas_suc : **array** [*] **of record**
 nro_cuenta : char(10);
 saldo : real;
 end
end

- Los Archivos de registro de longitud variable pueden ser implementados como archivos de registros de longitud fija mediante los siguientes esquemas:
 - Espacio reservado.
 - Representación con listas
 - Estructura de bloques anclas y de desbordamiento.

Organización de Archivos

Espacio Reservado

0	Navacerrada	C-102	400	C-201	900	C-218	700
1	Collado Mediano	C-305	350	⊥	⊥	⊥	⊥
2	Becerril	C-215	700	⊥	⊥	⊥	⊥
3	Centro	C-101	500	C-110	600	⊥	⊥
4	Moralzarzal	C-222	700	⊥	⊥	⊥	⊥
5	Galapagar	C-217	750	⊥	⊥	⊥	⊥

Representación con Listas

0	Navacerrada	C-102	80.000
1	Collado Mediano	C-305	70.000
2	Becerril	C-215	140.000
3	Centro	C-101	100.000
4	Moralzarzal	C-222	140.000
5		C-201	180.000
6	Galapagar	C-217	150.000
7		C-110	120.000
8		C-218	140.000




Bloques ancha y de desbordamiento

bloque
ancla

Navacerrada	C-102	400	
Collado Mediano	C-305	350	
Becerril	C-215	700	
Centro	C-101	500	
Moralzarzal	C-222	700	
Galapagar	C-217	750	

bloque de
desbordamiento

C-201	900	
C-218	700	
C-110	600	



Organización de Archivos


- ❑ Los registros de un archivo pueden ser organizados internamente de las siguientes formas:
 - Heap
 - ❑ Los registros son guardados en cualquier lugar del archivo en donde exista espacio suficiente.
 - Secuencial
 - ❑ Los registros se guardan en orden secuencial de acuerdo a una clave búsquedas.
 - Hash
 - ❑ Los registros son almacenados en un bloque especificado por una función hash que se aplica sobre el valor de un campo del registro.
 - Organización en agrupación
 - ❑ Los datos de diferentes relaciones se guardan en el mismo archivo.

Organización de Archivos

❑ Archivos Secuenciales

- Los registros se insertan secuencialmente
- Si los registros se insertan ordenados de acuerdo al valor de un campo del registro se tiene un **archivo secuencial ordenado**
 - ❑ Permiten una rápida recuperación de los registros a partir de un valor de búsqueda en el campo de ordenamiento.
 - ❑ Ayuda a minimizar la cantidad de accesos al disco para obtener datos en una búsqueda basada en el orden.

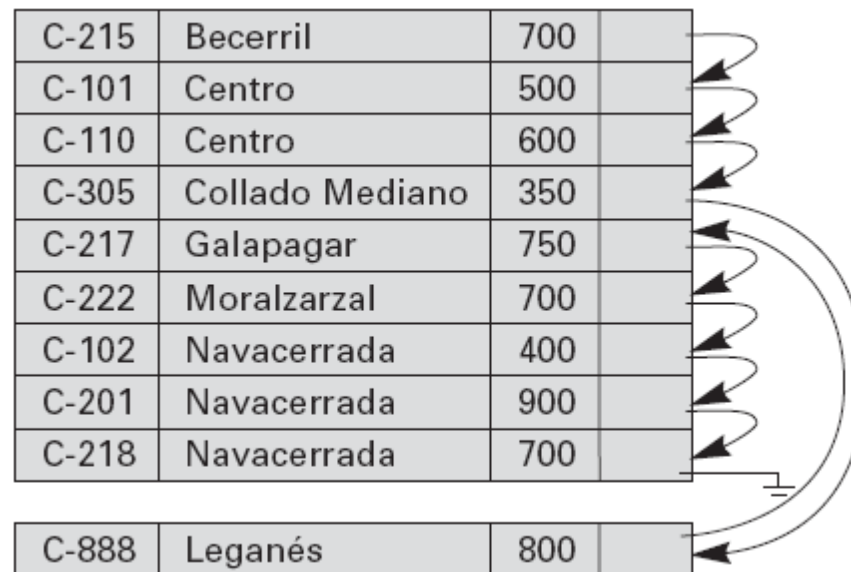
C-215	Becerril	700	
C-101	Centro	500	
C-110	Centro	600	
C-305	Collado Mediano	350	
C-217	Galapagar	750	
C-222	Moralzarzal	700	
C-102	Navacerrada	400	
C-201	Navacerrada	900	
C-218	Navacerrada	700	



Organización de Archivos

■ Archivos Secuenciales

- Puede ser costoso mantener el orden físico de los registros cuando ocurren inserciones o borrados, ya que pueden requerirse el desplazamiento de muchos registros
- Para insertar registros se siguen las siguientes reglas:
 - Localizar el registro que antecede en al orden al que se va insertar
 - Si existe espacio libre contiguo el nuevo registro se inserta en dicho lugar, caso contrario se inserta en un bloque de desbordamiento
 - Se ajustan los punteros para mantener el orden secuencial lógico.



Organización de Archivos

❑ Organización en agrupación

- Varias relaciones se pueden guardar un solo archivo de manera a optimizar la recuperación en operaciones que requieran datos de ambas relaciones

IMPOSITOR

<i>nombre-cliente</i>	<i>número-cuenta</i>
López	C-102
López	C-220
López	C-503
Abril	C-305

CLIENTE

<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
López	Principal	Arganzuela
Abril	Preciados	Valsaín


López	Mayor	Arganzuela
López	C-102	
López	C-220	
López	C-503	
Abril	Preciados	Valsaín
Abril	C-305	

Organización de Archivos

❑ Organización en agrupación

- Esta organización es buena para consultas que involucren ambas tablas
 - ❑ `select número-cuenta, nombre-cliente, calle-cliente, ciudad-cliente`
`from impositor, cliente`
`where impositor.nombre-cliente = cliente.nombrecliente`
- Malo para consultas sobre una sola relación
 - ❑ `select * from cliente`
- Es necesario encadenar los datos de la relación más dispersa.
- Solo debe ser utilizada cuando primen la operaciones que requieren datos de ambas relaciones

López	Mayor	Arganzuela	
López	C-102		
López	C-220		
López	C-503		
Abril	Preciados	Valsaín	
Abril	C-305		



Diccionario de Datos

- ❑ También denominado Catálogo del Sistema
- ❑ Guarda información acerca del sistema y sus datos, tales como:
 - Información acerca de la relaciones
 - ❑ Nombre de las relaciones
 - ❑ Nombre y tipo de los atributos de cada relación
 - ❑ Nombre y definiciones de las vistas
 - ❑ Restricciones de integridad
 - Cuentas de usuario y incluyendo nombre y contraseña
 - Información estadística y descriptiva
 - ❑ Número de tuplas de cada relación
 - ❑ Método de almacenamiento de cada relación
 - Información de la organización física
 - ❑ Tipo de almacenamiento para cada relación
 - ❑ Localización física de cada relación
 - Información de los índices
 - ❑ Nombre del índice y de su relación
 - ❑ Tipo de índice
 - ❑ Atributos sobre los cuales se crea el índice

Diccionario de datos

- ❑ El catalogo por lo general se guarda en la misma base de datos como estructura de relación.
- ❑ Debe mantener en memoria para permitir un acceso eficiente al mismo.
- ❑ La siguiente es una representación posible del catálogo del sistema.
 - *Metadatos-catálogo-sistema* = (nombre-relación, número-atributos)
 - *Metadatos-atributos* = (nombre-atributo, nombrerelación, tipo-dominio, posición, longitud)
 - *Metadatos-usuarios* = (nombre-usuario, contraseñacifrada, grupo)
 - *Metadatos-índices* = (nombre-índice, nombre-relación, tipo-índice, atributos-índice)
 - *Metadatos-vistas* = (nombre-vista, definición)

Tarea

- ❑ Leer capítulo 11 del libro de Silberschatz.
- ❑ Ver ejercicios del capítulo.
- ❑ Unirse al grupo:
 - http://groups.google.com/group/db2_2009