CI-5313 Arquitectura y Administración de Bases de Datos

Clase 2 – Estructuras de Almacenamiento Secundario

Prof. Edna Ruckhaus

Láminas Prof. José Tomás Cadenas

Jerarquía de Memoria y Dispositivos de Almacenamiento

- Nivel de almacenamiento principal (actuales)
  - Memoria caché, RAM estática
  - RAM dinámica (DRAM) o memoria principal
- Nivel de almacenamiento secundario (BD)
  - Memoria Flash
  - Discos Magnéticos
  - Discos ópticos (CD ó DVD)
- Nivel de almacenamiento Terciario (Histórico)
  - Tape Jukebox (Bancos de cintas), Optical Jukebox (Bancos de discos ópticos)

Data on External Storage

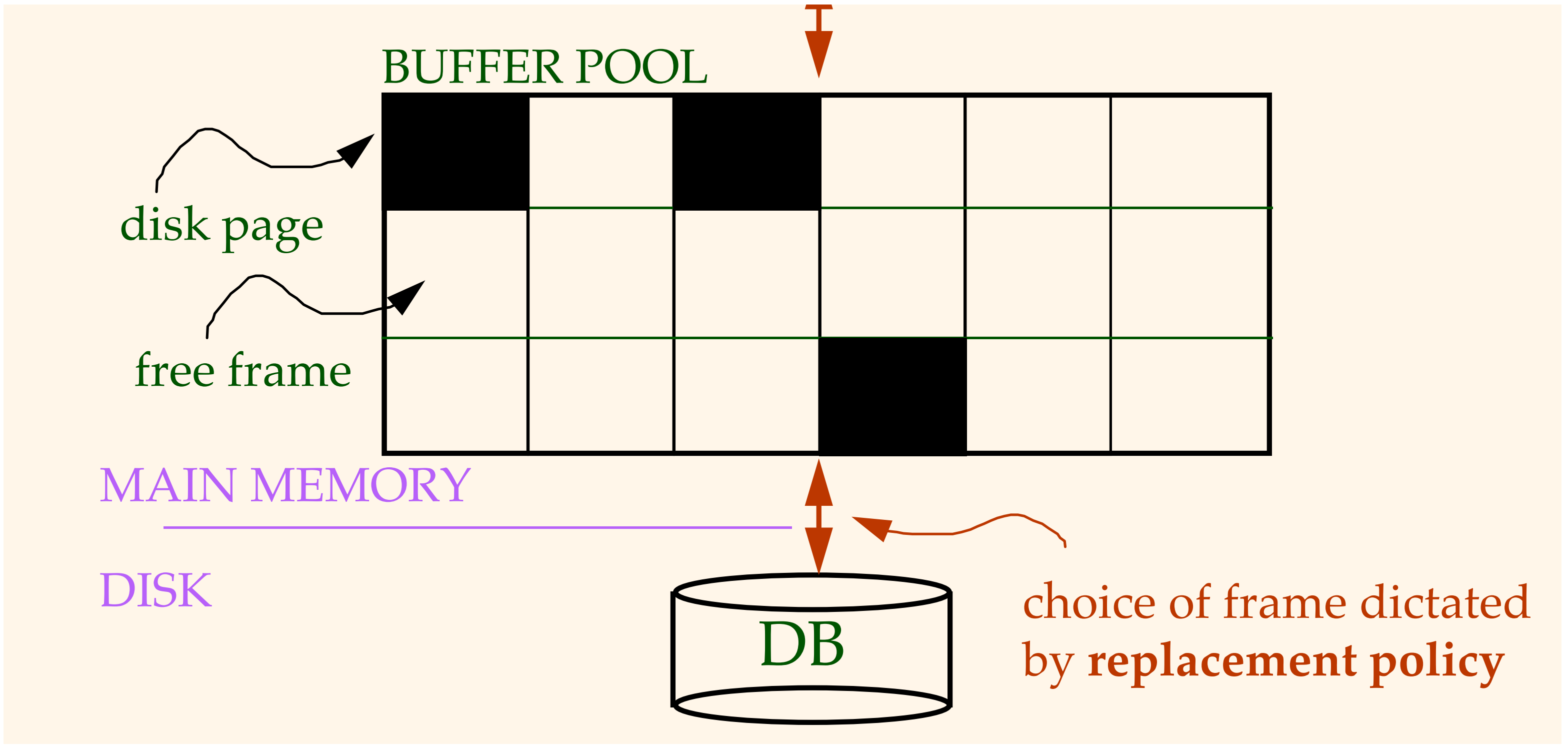
- READ: transfer data from disk to main memory (RAM).
- WRITE: transfer data from RAM to disk.
- Both are high-cost operations, relative to in-memory operations, so must be planned carefully!
- Time to retrieve a disk page varies depending upon location on disk. Therefore, relative placement of pages on disk has major impact on DBMS performance!
- File organization: Method of arranging a file of records on secondary storage.

Data on External Storage

- Page: Unit of transfer between main memory and secondary storage.
- Disks: Can retrieve random page at fixed cost
  But reading several consecutive pages is much cheaper than reading them in random order
- Record id (rid) is sufficient to physically locate record
  - Indexes are data structures that allow us to find the record ids of records with given values in index search key fields.

.

Data on External Storage



When a page is requested

- If requested page is not in pool: – Choose a frame for replacement
  - If frame is dirty, write it to disk
  - Read requested page into chosen frame
- Pin the page and return its address.

- If requests can be predicted (e.g.,sequential scans)
  - pages can be pre-fetched several pages at a time!

More on buffer management

- Requestor of page must unpin it, and indicate whether page has been modified:
  - dirty bit is used for this.
- Page in pool may be requested many times,
  - a pin count is used. A page is a candidate for replacement iff pin count = 0.

Frame replacement policy

- Frame is chosen for replacement by a replacement policy:
  - Least-recently-used (LRU), MRU etc.
- Policy can have big impact on # of I/O's; depends on the access pattern.
- Sequential flooding: Nasty situation caused by LRU + repeated sequential scans.
  - *# buffer frames < # pages in file* means each page request causes an I/O. MRU much better in this situation (but not in all situations, of course).

Disk Storage Devices (contd.)

- A track is divided into smaller sectors
  - because it usually contains a large amount of information
- The division of a track into sectors is hard-coded on the disk surface and cannot be changed.
  - One type of sector organization calls a portion of a track that subtends a fixed angle at the center as a sector.
- A track is divided into blocks (pages).
  - The block size B is fixed for each system.
    - Typical block sizes range from B=512 bytes to B=4096 bytes.
  - Whole blocks are transferred between disk and main memory for processing.
- Tracks under heads make a cylinder (imaginary!).

Disk Storage Devices (contd.)

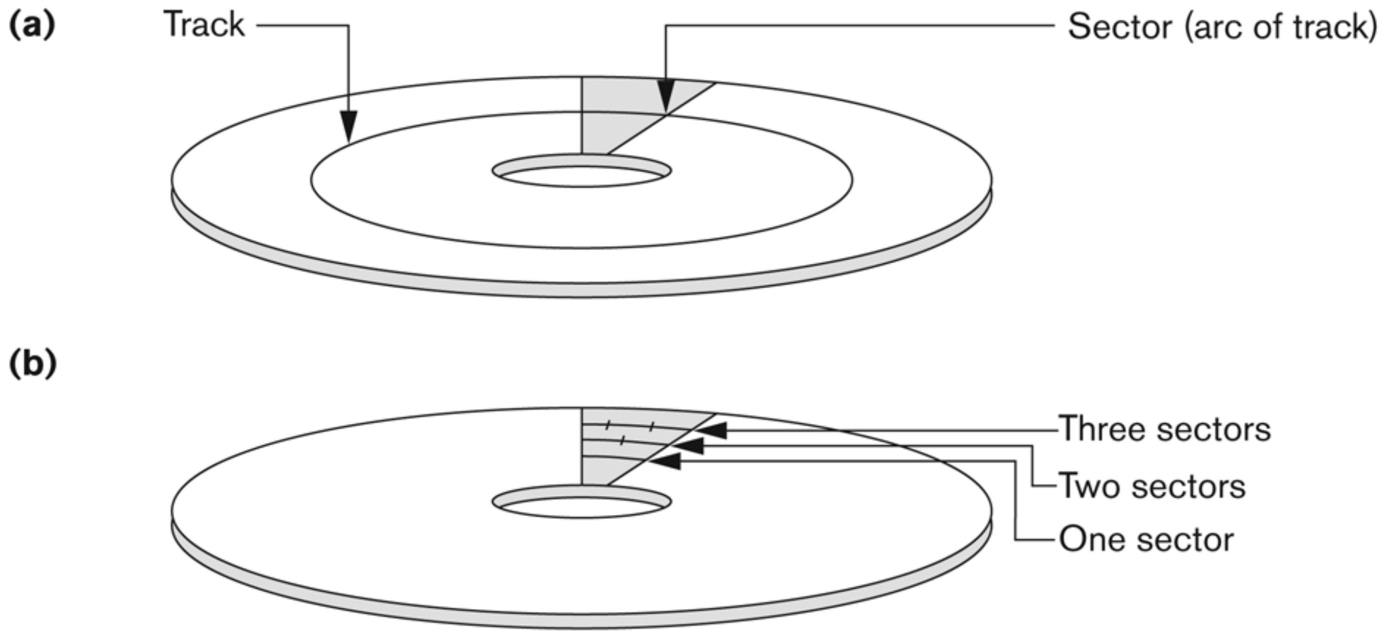
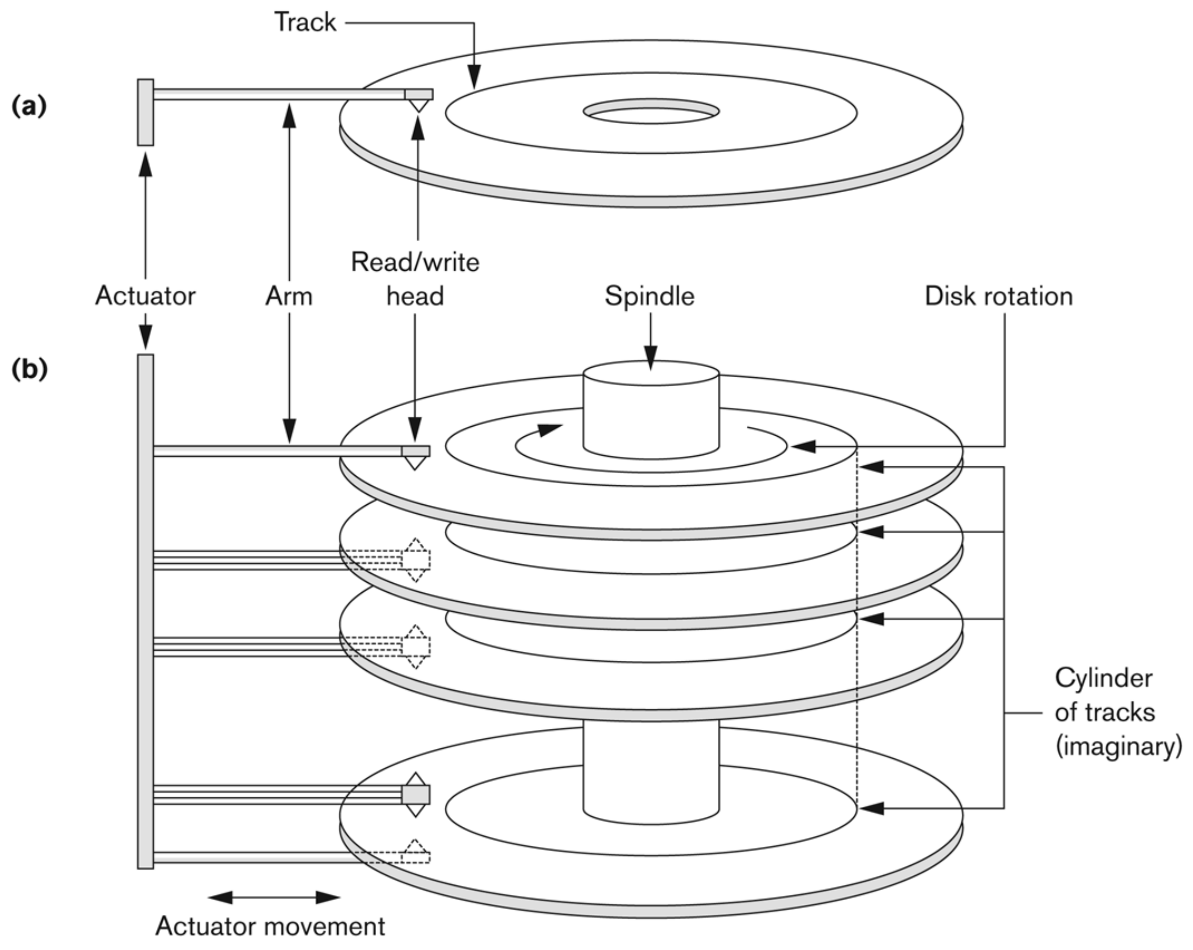


Figure 13.2
Different sector organizations on disk. (a) Sectors subtending a fixed angle. (b) Sectors maintaining a uniform recording density.

Disk Storage Devices (contd.)

Figure 13.1
(a) A single-sided disk with read/write hardware. (b) A disk pack with read/write hardware.



Disk Storage Devices (contd.)

- A read-write head moves to the track that contains the block to be transferred.
  - Disk rotation moves the block under the read-write head for reading or writing.
- A physical disk block (hardware) address consists of:
  - a cylinder number (imaginary collection of tracks of same radius from all recorded surfaces)
  - the track number or surface number (within the cylinder)
  - and block number (within track).
- Reading or writing a disk block is time consuming because of the seek time s and rotational delay (latency).

Disk Storage Devices (contd.)

- Time to access (read/write) a disk block:
  - seek time: moving arms to position disk head on track
  - rotational delay: waiting for block to rotate under head
  - transfer time: actually moving data to/from disk surface
- Seek time and rotational delay dominate.
- Key to lower I/O cost: reduce seek/rotation
- Delays! Hardware vs. software solutions?

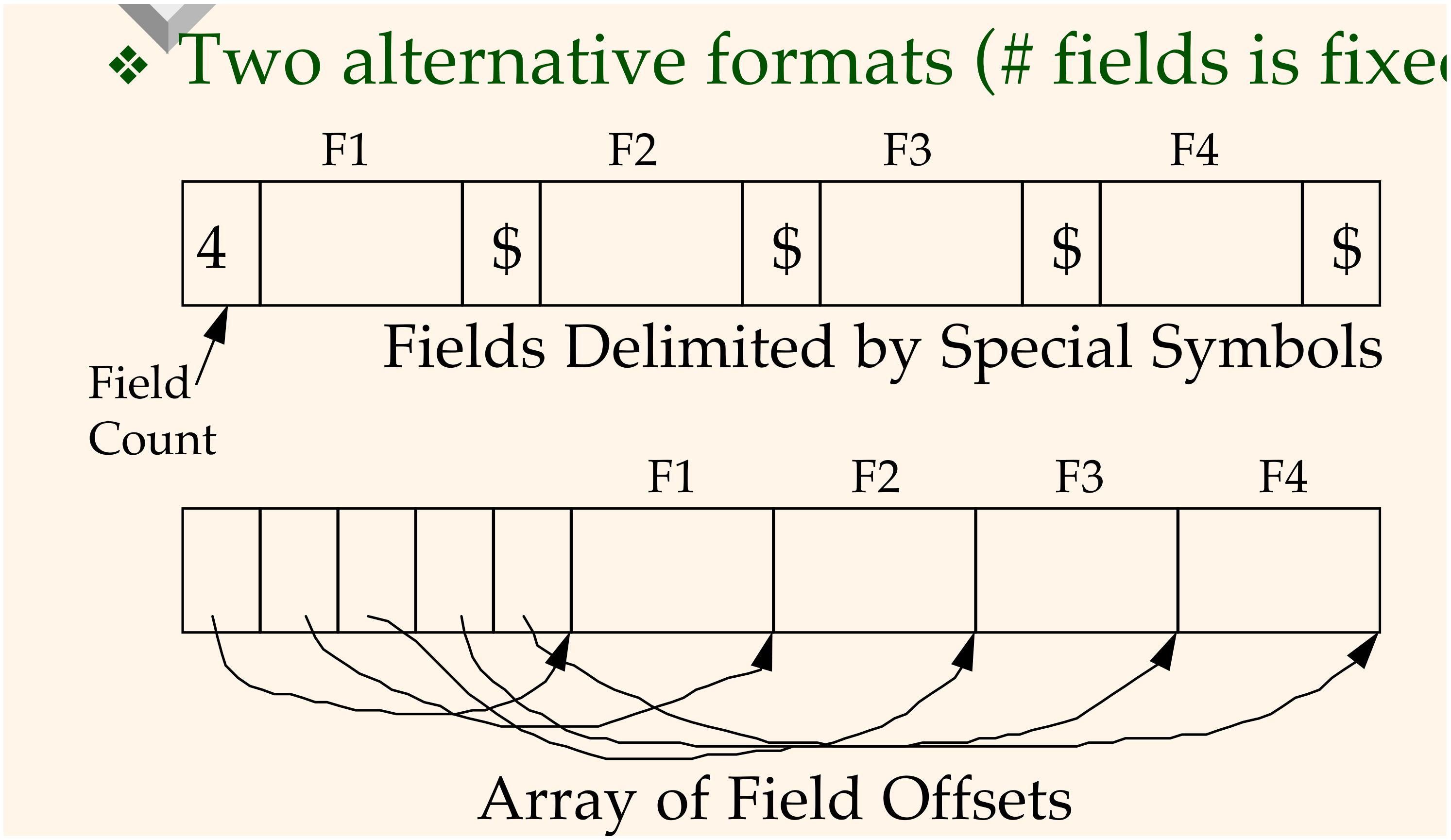
Arranging Pages on Disk

- `Next' block concept:
  - blocks on same track, followed by
  - blocks on same cylinder, followed by
  - blocks on adjacent cylinder
- Blocks in a file should be arranged sequentially on disk (by `next'), to minimize seek and rotational delay.
- For a sequential scan, pre-fetching several pages at a time is a big win!

Records

- Fixed and variable length records
- Records contain fields which have values of a particular type
  - E.g., amount, date, time, age
- Fields themselves may be fixed length or variable length
- Variable length fields can be mixed into one record:
  - Separator characters or length fields are needed so that the record can be “parsed.”

Records



❖ Two alternative formats (# fields is fixe

Fields Delimited by Special Symbols

Array of Field Offsets

Blocking

- Blocking:
  - Refers to storing a number of records in one block on the disk.
- Blocking factor (bfr) refers to the number of records per block.
- There may be empty space in a block if an integral number of records do not fit in one block.
- Spanned Records:
  - Refers to records that exceed the size of one or more blocks and hence span a number of blocks.

Files of Records

- Page or block is OK when doing I/O, but higher levels of DBMS operate on records, and files of records.
- FILE: A collection of pages, each containing a collection of records. Must support:
  - insert/delete/modify record
  - read a particular record (specified using record id)
  - scan all records (possibly with some conditions on the records to be retrieved)

Files of Records (contd.)

- The physical disk blocks that are allocated to hold the records of a file can be contiguous, linked, or indexed.
- In a file of fixed-length records, all records have the same format. Usually, unspanned blocking is used with such files.
- Files of variable-length records require additional information to be stored in each record, such as separator characters and field types.
  - Usually spanned blocking is used with such files.

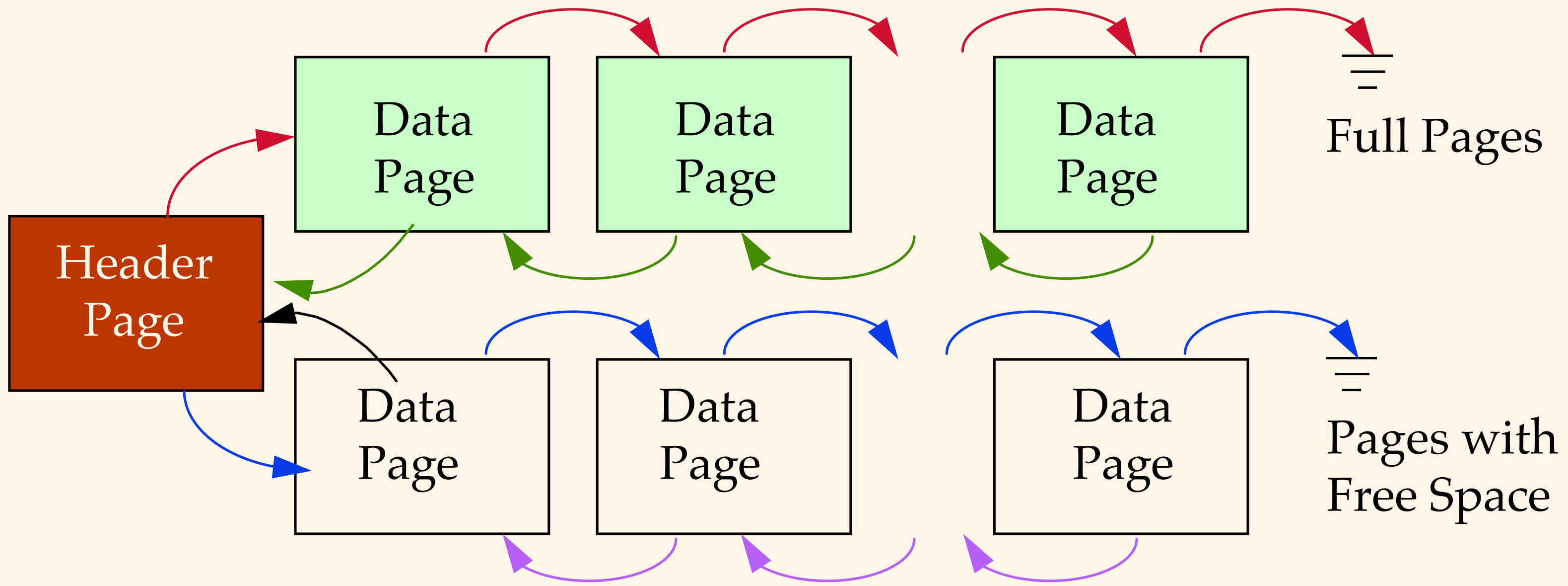
Operations on Files

- Typical file operations include:
  - OPEN: Readies the file for access, and associates a pointer that will refer to a current file record at each point in time.
  - FIND: Searches for the first file record that satisfies a certain condition, and makes it the current file record.
  - FINDNEXT: Searches for the next file record (from the current record) that satisfies a certain condition, and makes it the current file record.
  - READ: Reads the current file record into a program variable.
  - INSERT: Inserts a new record into the file & makes it the current file record.
  - DELETE: Removes the current file record from the file, usually by marking the record to indicate that it is no longer valid.
  - MODIFY: Changes the values of some fields of the current file record.
  - CLOSE: Terminates access to the file.
  - REORGANIZE: Reorganizes the file records.
    - For example, the records marked deleted are physically removed from the file or a new organization of the file records is created.
  - READ_ORDERED: Read the file blocks in order of a specific field of the file.

Unordered (Heap) Files

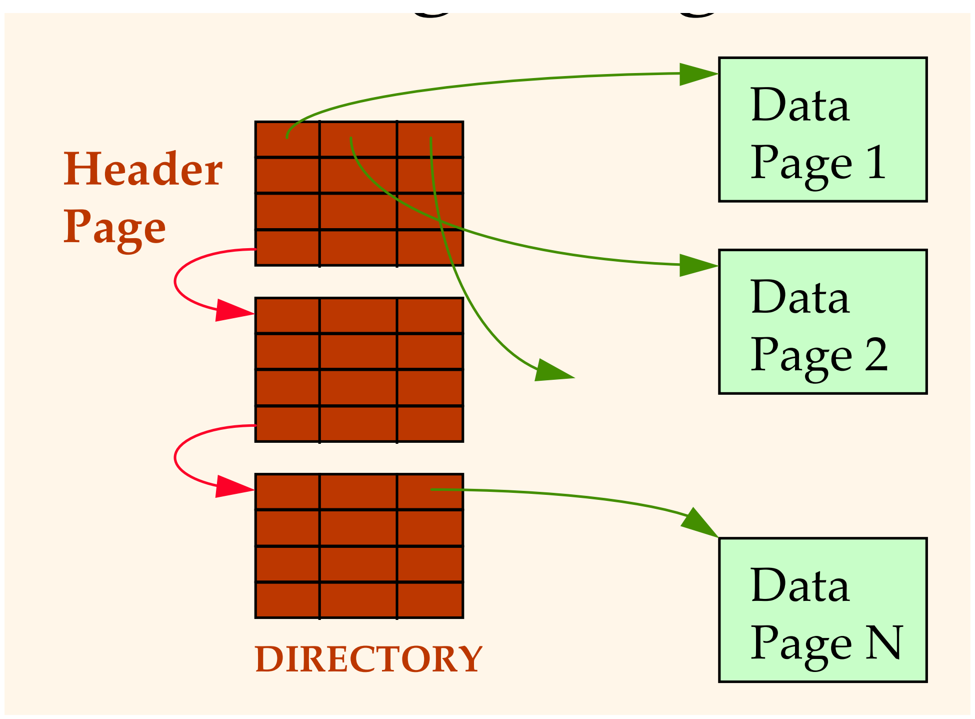
- Simplest file structure contains records in no particular order.
- As file grows and shrinks, disk pages are allocated and de-allocated. New records are inserted at the end of the file.
- To support record level operations, we must:
  - keep track of the pages in a file
  - keep track of free space on pages
  - keep track of the records on a page
- A linear search through the file records is necessary to search for a record.
  - This requires reading and searching half the file blocks on the average, and is hence quite expensive.
- Reading the records in order of a particular field requires sorting the file records.

Unordered (Heap) Files – Implemented as a List



- The header page id and Heap file name must be stored someplace.
- Each page contains 2 `pointers' plus data.

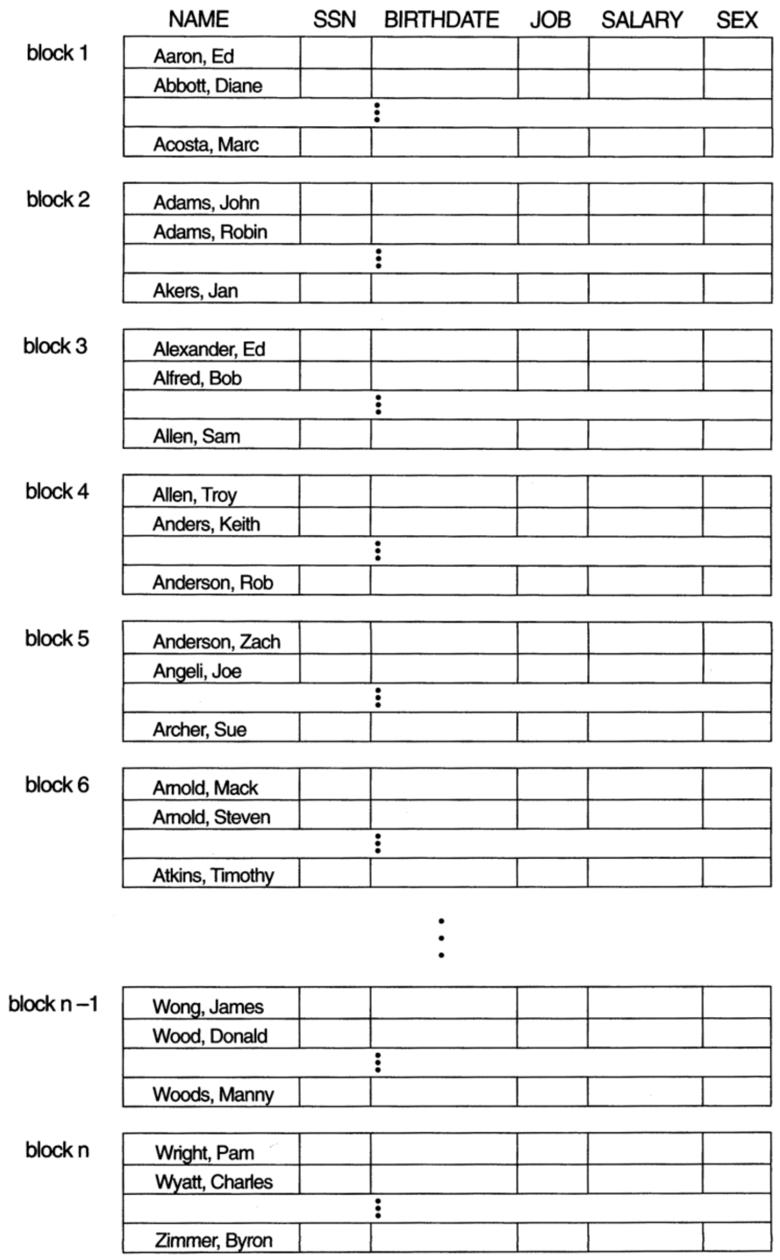
Unordered (Heap) Files – Using a Page Directory



Ordered Files

- Also called a sequential file.
- File records are kept sorted by the values of an ordering field.
- Insertion is expensive: records must be inserted in the correct order.
  - It is common to keep a separate unordered overflow (or transaction) file for new records to improve insertion efficiency; this is periodically merged with the main ordered file.
- A binary search can be used to search for a record on its ordering field value.
  - This requires reading and searching $\log_2$ of the file blocks on the average, an improvement over linear search.
- Reading the records in order of the ordering field is quite efficient.

Ordered Files (contd.)



	NAME	SSN	BIRTHDATE	JOB	SALARY	SEX
block 1	Aaron, Ed					
	Abbott, Diane					
	⋮					
	Acosta, Marc					
block 2	Adams, John					
	Adams, Robin					
	⋮					
	Akers, Jan					
block 3	Alexander, Ed					
	Alfred, Bob					
	⋮					
	Allen, Sam					
block 4	Allen, Troy					
	Anders, Keith					
	⋮					
	Anderson, Rob					
block 5	Anderson, Zach					
	Angeli, Joe					
	⋮					
	Archer, Sue					
block 6	Arnold, Mack					
	Arnold, Steven					
	⋮					
	Atkins, Timothy					
	⋮					
block n –1	Wong, James					
	Wood, Donald					
	⋮					
	Woods, Manny					
block n	Wright, Pam					
	Wyatt, Charles					
	⋮					
	Zimmer, Byron					

Modelo de Costos

bfr el factor de bloqueo del archivo

N número de registros del archivo

B número de bloques de disco necesarios para almacenar el archivo, se calcula como: techo(N/bfr)

D tiempo promedio para leer o escribir un bloque de disco

C tiempo promedio para procesar un registro del archivo

Análisis de casos promedios, basado en suposiciones simples

Archivo Heap

- Los registros del archivo se colocan en el orden en el cual se van insertando, los nuevos registros se insertan al final del archivo
- Eliminar registros:
  - Se borra el registro y se corren todos los registros siguientes al eliminado para no desperdiciar espacio
  - Se marca el registro como borrado y se reorganiza en algún momento para eliminarlo definitivamente.
  - El último registro pasa al espacio eliminado.
- Si se desea tener los registros en algún orden particular se debe ejecutar una rutina de ordenamiento externa a la organización del archivo (external sorting)
- Inserción es muy eficiente
- No facilita ningún tipo de búsqueda

Heap File (costos)

- Scan completo: B*(D+bfr*C)
- Búsqueda por igualdad (sobre clave): B/2*(D+bfr*C)
- Búsqueda por rango: B*(D+bfr*C)
- Insert: 2*D+C
- Delete: costo de la búsqueda del registro a eliminar +C+D

Archivo ordenado (costos)

- Scan completo: B*(D+bfr*C)
- Búsqueda por igualdad sobre el atributo “ordenado” (no único): D*log$_2$B+C*log$_2$bfr
- Búsqueda por rango: el costo total es el costo de realizar la búsqueda, mas el costo de traerse el primer bloque que contiene el primer registro del rango, mas el costo de acceder todos los otros registros del rango, lo cual puede implicar mas accesos a discos, si son muchos registros
- Insert: costo de buscar la posición donde debe ir el nuevo registro + 2*(B/2)*(D+bfr*C))=costo busqueda+B*(D +bfr*C)
- Delete: eliminación es el costo de la búsqueda del registro a eliminar + B*(D+bfr*C)

Ejercicio disco magnético

- ¿Cuál es la capacidad total de una pista y cuál es su capacidad útil (excluyendo gaps entre bloques)?
- ¿Cuántos cilindros tiene?
- ¿Cuál es la capacidad total de un cilindro y cuál es su capacidad útil (excluyendo gaps entre bloques)?
- ¿Cuál es la capacidad total del paquete de disco y cuál es su capacidad útil (excluyendo gaps entre bloques)?
- Dado que la tasa de transferencia en bytes/ms es 512 bytes/ms?
  - ¿Cuál es el tiempo de transferir un bloque en ms?
- Dado que el retraso promedio de rotación rd (latency) es 12.5 ms y dado que el tiempo de búsqueda promedio s=30 mseg ¿Tiempo promedio para localizar y transferir un bloque?
- Calcular el tiempo promedio que se necesitaría para transferir 20 bloques. Cual sería la diferencia si los bloques fuesen contiguos?

Ejercicio Archivos

- Dado un archivo de estudiantes que contiene N=20.000 registros de longitud fija. Cada registro tiene los siguientes campos: Nombre (40 bytes), CI (9 bytes), Dirección (40 bytes), Telf. (11 bytes), Fecha de Nacimiento (8 bytes), Sexo (1 byte) y Código Carrera (3 bytes), además se utiliza un campo de un byte como marca de eliminación. El archivo es almacenado en el paquete de disco con los parámetros del ejercicio anterior. C, tiempo procesar un registro = 0.1 ms.

- ¿Calcular el tamaño de R en bytes?
- ¿Calcular el factor de bloqueo bfr y el número de bloques suponiendo una organización no expandida (unspanned)?
- Calcular el tiempo promedio que toma encontrar un registro por CI mediante búsqueda lineal en el archivo (Heap)
- Suponga que el archivo es ordenado por CI; calcular el tiempo que toma buscar un registro dado su valor de CI mediante una búsqueda binaria