

Design of Geographic Information Systems

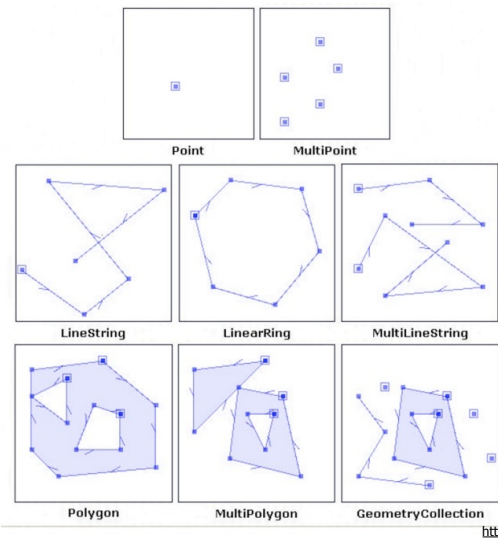
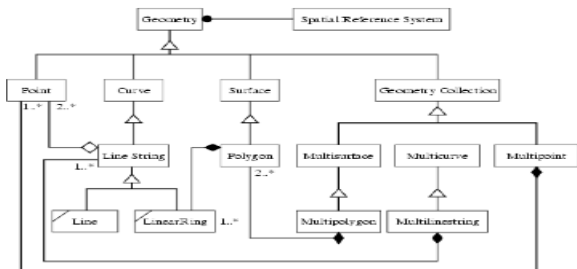
class 7

Thomas Bittner
bittner3@buffalo.edu

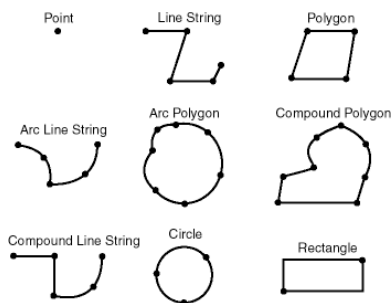
Recap: Spatial data types

OGIS Spatial Data Model

- Consists of base-class Geometry and four sub-classes:
 - Point, Curve, Surface and GeometryCollection
 - spatial data types in OGIS

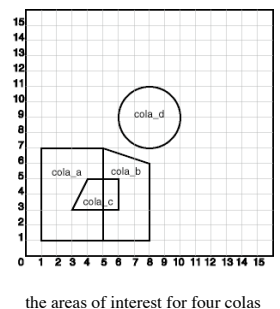


Geometry Types in Oracle SQL



Example: Oracle commands for the following operations

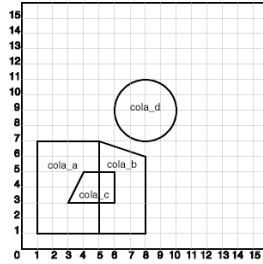
- Create a table (COLA_MARKETS) to hold the spatial data
- Insert rows for four areas of interest (cola_a, cola_b, cola_c, cola_d)
- Create a spatial index (COLA_SPATIAL_IDX)
- Perform some spatial queries



Example: Oracle commands for the following operations

- Create a table (COLA_MARKETS) to hold the spatial data

```
CREATE TABLE cola_markets (
  mkt_id NUMBER PRIMARY KEY,
  name VARCHAR2(32),
  shape SDO_GEOMETRY);
```



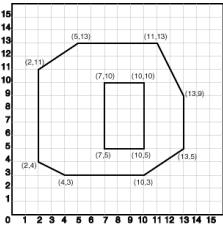
the areas of interest for four colas

SDO_GEOMETRY Object Type

```
SDO_GEOMETRY(
  2003, -- two-dimensional polygon
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,3), -- one rectangle (1003 = exterior)
  SDO_ORDINATE_ARRAY(1,1, 5,7), -- only 2 points needed to
  -- define rectangle (lower left and upper right) with
  -- Cartesian-coordinate data
)
```

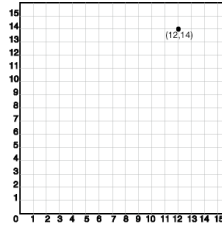
```
CREATE TYPE sdo_geometry AS OBJECT
(
  SDO_GTYPE NUMBER,
  SDO_SRID NUMBER,
  SDO_POINT SDO_POINT_TYPE,
  SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,
  SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

GEOMETRY examples: Polygon with a Hole



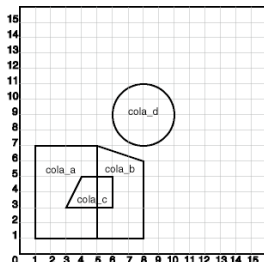
```
INSERT INTO cola_markets VALUES(
  10,
  'polygon with hole',
  SDO_GEOMETRY(
    2003, -- two-dimensional polygon
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1, 19,2003,1), -- polygon with hole
    SDO_ORDINATE_ARRAY(2,4, 4,3, 10,3, 13,5, 13,9, 11,13, 5,13, 2,11, 2,4,
      7,5, 7,10, 10,10, 10,5, 7,5)
  ));
```

GEOMETRY examples: Points



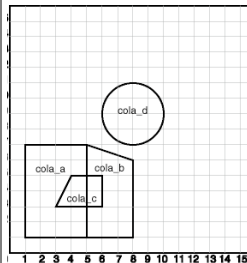
```
INSERT INTO cola_markets VALUES(
  90,
  'point_only',
  SDO_GEOMETRY(
    2001,
    NULL,
    SDO_POINT_TYPE(12, 14, NULL),
    NULL));
```

Exercise: specify the insert statements for the areas cola_b, cola_c, and cola_d



the areas of interest for four colas

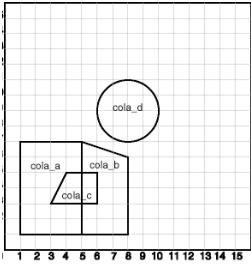
Exercise: specify the insert statements for the areas cola_b, cola_c, and cola_d



the areas of interest for four colas

```
INSERT INTO cola_markets VALUES(
  1,
  'cola_a',
  SDO_GEOMETRY(
    2003, -- two-dimensional polygon
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,3), -- one rectangle (1003 = exterior)
    SDO_ORDINATE_ARRAY(1,1, 5,7) -- only 2 points needed to
    -- define rectangle (lower left and upper right) with
    -- Cartesian-coordinate data
  )
);
```

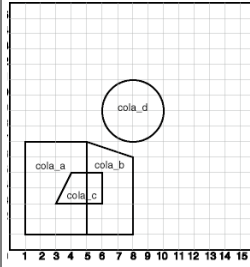
Exercise: specify the insert statements for the areas cola_b, cola_c, and cola_d



```
INSERT INTO cola_markets VALUES(
  2,
  'cola_b',
  SDO_GEOMETRY(
    2003, -- two-dimensional polygon
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1), -- one polygon (exterior polygon ring)
    SDO_ORDINATE_ARRAY(5,1, 8,1, 8,6, 5,7, 5,1)
  )
);
```

the areas of interest for four colas

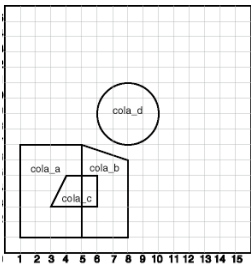
Exercise: specify the insert statements for the areas cola_b, cola_c, and cola_d



```
INSERT INTO cola_markets VALUES(
  3,
  'cola_c',
  SDO_GEOMETRY(
    2003, -- two-dimensional polygon
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1), -- one polygon (exterior polygon ring)
    SDO_ORDINATE_ARRAY(3,3, 6,3, 6,5, 4,5, 3,3)
  )
);
```

the areas of interest for four colas

Exercise: specify the insert statements for the areas cola_b, cola_c, and cola_d

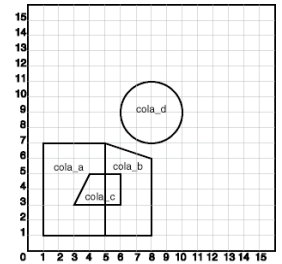


```
INSERT INTO cola_markets VALUES(
  4,
  'cola_d',
  SDO_GEOMETRY(
    2003, -- two-dimensional polygon
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,4), -- one circle
    SDO_ORDINATE_ARRAY(8,7, 10,9, 8,11)
  )
);
```

the areas of interest for four colas

Example: Oracle commands for the following operations

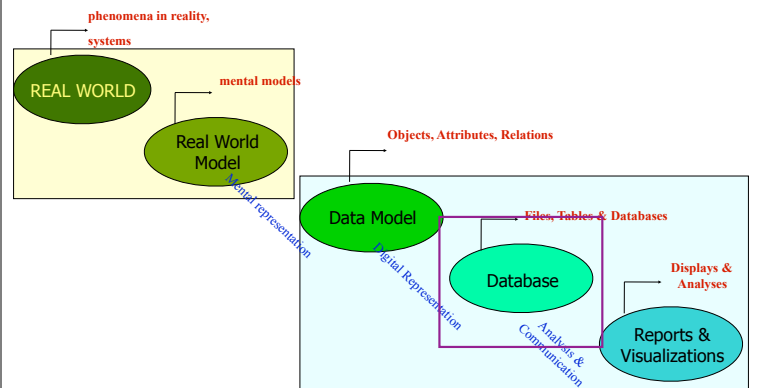
- Create a table (COLA_MARKETS) to hold the spatial data
- Insert rows for four areas of interest (cola_a, cola_b, cola_c, cola_d)
- Create a spatial index (COLA_SPATIAL_IDX)
- Perform some spatial queries



the areas of interest for four colas

Data storage and access

Components of computerized information systems



Storage Hierarchy in Computers

- Computers have several components
 - Central Processing Unit (CPU)
 - Input, output devices, e.g. mouse, keyboard, monitors, printers
 - Communication mechanisms, e.g. internal bus, network card, modem
 - Storage Hierarchy
- Types of storage Devices
 - Main memories - fast but content is lost when power is off
 - Secondary storage - slower, retains content without power
 - Tertiary storage - very slow, retains content, very large capacity
- DBMS usually manage data
 - on secondary storage, e.g. disks
 - Use main memory to improve performance
 - User tertiary storage (e.g. tapes) for backup, archival etc.

How storage devices work

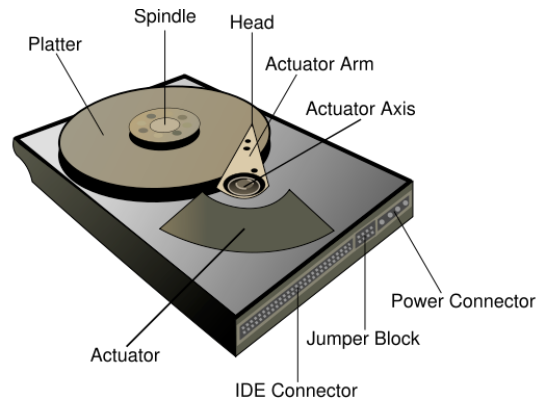
<http://computer.howstuffworks.com/hard-disk.htm>

Secondary Storage Hardware: Disk Drives



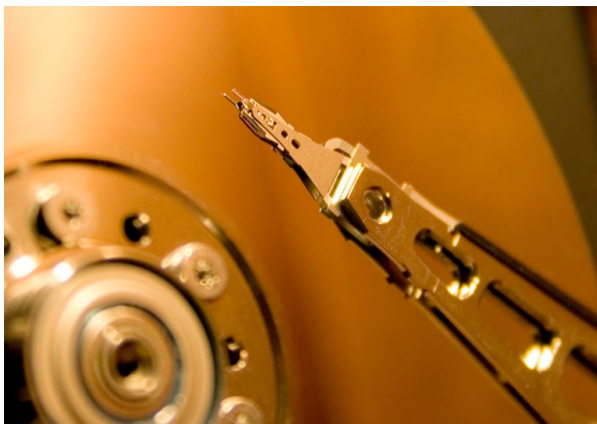
<http://computer.howstuffworks.com/hard-disk.htm>

Secondary Storage Hardware: Disk Drives

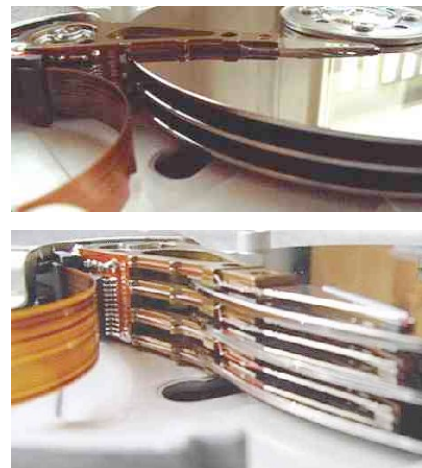


<http://computer.howstuffworks.com/hard-disk.htm>

Secondary Storage Hardware: Disk Drives - head



<http://computer.howstuffworks.com/hard-disk.htm>

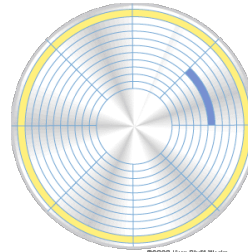




The mechanism that moves the arms on a hard disk has to be incredibly fast and precise. It can be constructed using a high-speed linear motor.

Storing the data

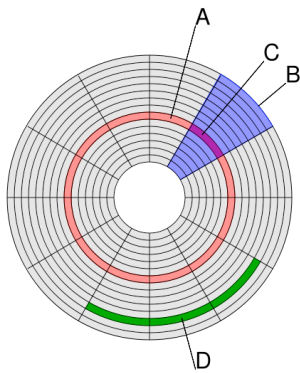
Data is stored on the surface of a platter in **sectors** and **tracks**. Tracks are concentric circles, and sectors are pie-shaped wedges on a track:



A typical track is shown in yellow; a typical sector is shown in blue. A sector contains a fixed number of bytes -- for example, 256 or 512. Either at the drive or the [operating system](#) level, sectors are often grouped together into **clusters**.

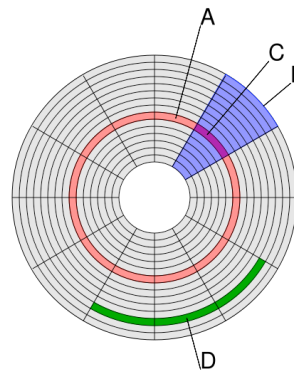
The process of **low-level formatting** a drive establishes the tracks and sectors on the platter. The starting and ending points of each sector are written onto the platter. This process prepares the drive to hold blocks of bytes.

High-level formatting then writes the file-storage structures, like the file-allocation table, into the sectors. This process prepares the drive to hold files.

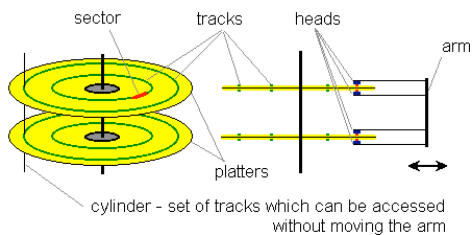


Disk structure:
(A) track
(B) geometrical sector
(C) [track sector](#)
(D) **cluster**

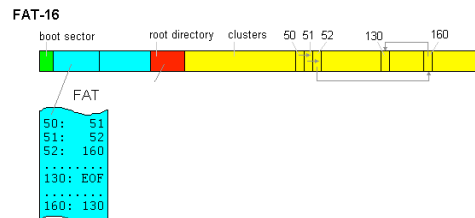
A cluster need not be physically contiguous on the disk; it may span more than one [track](#) or, if [sector interleaving](#) is used, may even be discontinuous within a track.



Storing the data

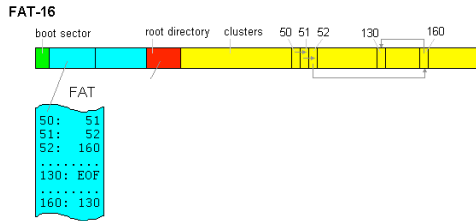


File allocation table



The FAT table has one entry for each cluster in the partition - it contains the number of the next cluster in the chain.

File allocation table

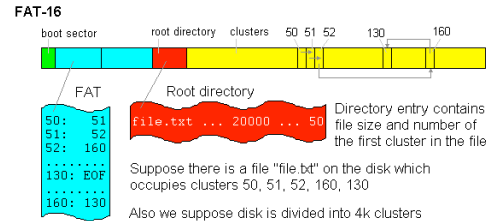


The FAT table has one entry for each cluster in the partition - it contains the number of the next cluster in the chain.

There is one chain of clusters for each file.

31

File allocation table



Directory entry contains file size and number of the first cluster in the file.

Suppose there is a file "file.txt" on the disk which occupies clusters 50, 51, 52, 160, 130

Also we suppose disk is divided into 4k clusters

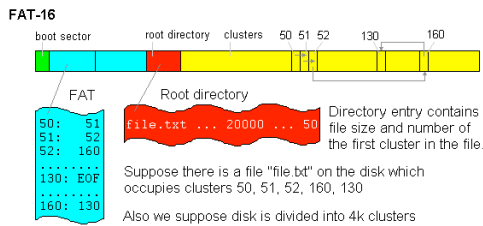
The FAT table has one entry for each cluster in the partition - it contains the number of the next cluster in the chain.

There is one chain of clusters for each file.

The number of the first cluster of the file is stored in the directory entry for that file, along with the file size, some attributes, and the last modification date.

32

File allocation table



Suppose there is a file "file.txt" on the disk which occupies clusters 50, 51, 52, 160, 130

Also we suppose disk is divided into 4k clusters

The FAT table has one entry for each cluster in the partition - it contains the number of the next cluster in the chain.

There is one chain of clusters for each file.

The number of the first cluster of the file is stored in the directory entry for that file, along with the file size, some attributes, and the last modification date.

Space for the directories other than root is allocated among the data clusters, just as if they were ordinary files.

Only the root directory has a special location.

33

The lecture notes are based on slides by Shashi Shekhar

Secondary Storage Hardware: Disk Drives

Disk hardware and organization

- Circular platters with magnetic storage medium
- Multiple platters are mounted on a spindle
- Platters are divided into concentric tracks
- A cylinder is a collection of tracks across platters with common radius
- Tracks are divided into sectors
- A sector size may a few kilo-Bytes

Writing data on a disc

- Disk heads to read and write
- There is disk head for each platter (recording surface)
- A head assembly moves all the heads together in radial direction
- Spindle rotates at a high speed, e.g. thousands revolution per minute

Accessing a sector has three major steps:

- Seek: Move head assembly to relevant track
- Latency: Wait for spindle to rotate relevant sector under disk head
- Transfer: Read or write the sector
- Other steps involve communication between disk controller and CPU

The lecture notes are based on slides by Shashi Shekhar

Using Disk Hardware Efficiently

- Disk access cost are affected by
 - Placement of data on the disk
 - seek cost > latency cost > transfer
 - A few common observations follow
- Size of sectors
 - Larger sector provide faster transfer of large data sets
 - But waste storage space inside sectors for small data sets
- Placement of most frequently accessed data items
 - On middle tracks rather than innermost or outermost tracks
 - Reason: minimize average seek time
- Placement of items in a large data set requiring many sectors
 - Choose sectors from a single cylinder
 - Reason: Minimize seek cost in scanning the entire data set.

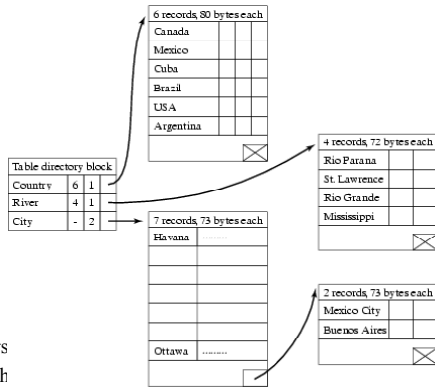
The lecture notes are based on slides by Shashi Shekhar

Software view of Disks: Fields, Records and File

- Views of secondary storage (e.g. disks)
 - Hardware views - discussed in last few slides
 - Software views - Data on disks is organized into fields, records, files
- Software view:
 - Field presents a property or attribute of a relation or an entity
 - Records represent a row in a relational table
 - Files are collections of records

Mapping Records and files to Disk

- Records
 - Often smaller than a sector
 - Many records in a sector
- Files with many records
 - Many sectors per file
- File system
 - Collection of files
 - Organized into directories
- Mapping tables to disk
 - City table takes 2 sectors
 - Others take 1 sector each



4.1.2 Buffer Management

- Motivation
 - Accessing a sector on disk is much slower than accessing main memory
 - Idea: Keep repeatedly accessed data in main memory buffers
 - To improve the completion time of queries
 - Reducing load on disk drive
- Buffer Manager software module decides
 - Which sectors stay in main memory buffers?
 - Which sector is moved out if we run out of memory buffer space?
 - When to pre-fetch sector before access request from users?
 - These decision are based on the disk access patterns of queries!

How to structure data files

4.1.4 File Structures

- What is a file structure?
 - A method of organizing records in a file
 - For efficient implementation of common file operations on disks
- Measure of efficiency
 - I/O cost: Number of disk sectors retrieved from secondary storage
 - CPU cost: Number of CPU instruction used
 - Total cost = sum of I/O cost and CPU cost

4.1.4 File Structures - selected file operations

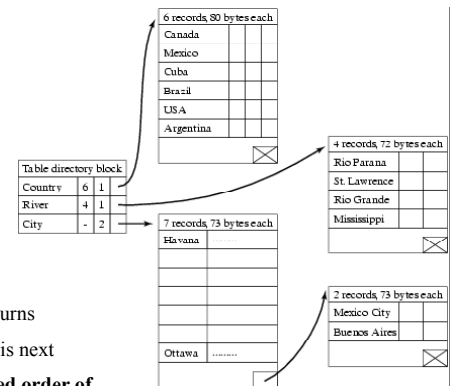
- Common file operations
 - Find: key value --> record matching key values
 - Findnext --> Return next record after find if records were sorted
 - Insert --> Add a new record to file without changing file-structure
 - Nearest neighbor of a object in a spatial dataset

4.1.4 File Structures - selected file operations

find(Name = Canada) on Country table returns record about Canada

findnext() on Country table returns ???

findnext() on Country table returns record about Cuba since Cuba is next value after Canada **in the sorted order of Name**

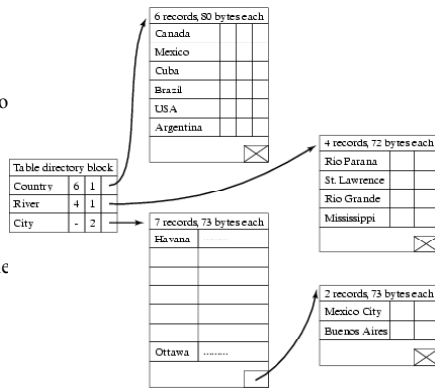


4.1.4 File Structures - selected file operations

insert(record about Panama) into Country table

adds a new record

location of record in Country file depends on file-structure



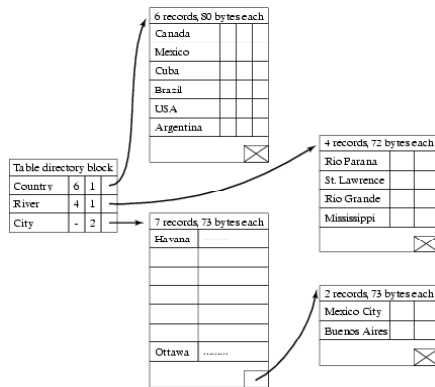
4.1.4 Common File Structures

- Common file structures
 - Heap or unordered or unstructured
 - Ordered
 - Hashed
 - Clustered
- Basic Comparison of Common File Structures
 - Heap file is efficient for inserts and used for logfiles
 - But find, findnext, etc. are very slow
 - Hashed files are efficient for find, insert, delete etc.
 - But findnext is very slow
 - Ordered file organization are very fast for findnext
 - and pretty competent for find, insert, etc.

4.1.4 File Structures: Heap

•Heap

- Records are in no particular order
- insert can simple add record to the last sector
- find, findnext, nearest neighbor scan the entire files



File Structure : Hash

•Components of a Hash file structure

- A set of buckets (sectors)
- Hash function : key value --> bucket
- Hash directory: bucket --> sector

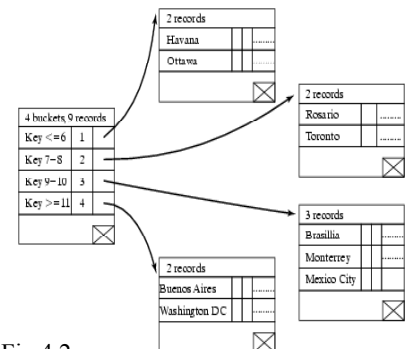


Fig 4.2

File Structure : Hash

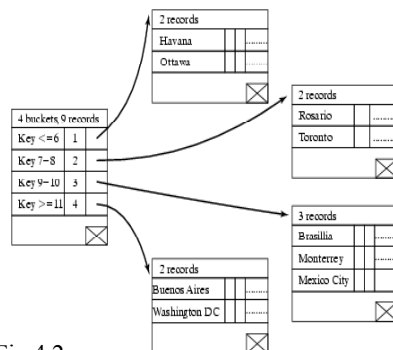
• Operations

- find, insert, delete are fast
 - compute hash function
 - lookup directory
 - fetch relevant sector

• findnext, nearest neighbor are slow

- no order among records

Fig 4.2



4.1.4 File Structures: Ordered

•Ordered

- Records are sorted by a selected field (Example Fig. 4.3 below)
- findnext can simply pick up physically next record
- find, insert, delete may use binary search, is is very efficient

Ordered file storing City table(ordered)

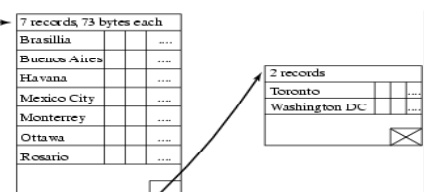


Figure 4.3

4.1.5 Spatial File Structures: Clustering

- Motivation:
 - Ordered files are not natural for spatial data
 - Clustering records in sector by space filling curve is an alternative
 - In general, clustering groups records
 - accessed by common queries
 - into common disk sectors
 - to reduce I/O costs for selected queries
- Clustering using Space filling curves
 - Z-curve
 - Hilbert-curve

Z-Curve

• What is a Z-curve?

- A space filling curve
- Connecting points by z-order
 - curves look like combinations of Ns or Zs

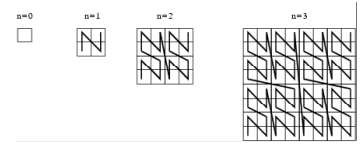
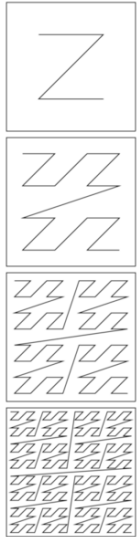


Fig 4.4



Binary numbers

decimal	2^3	2^2	2^1	2^0
---------	-------	-------	-------	-------

Binary numbers

decimal	2^3	2^2	2^1	2^0
0	0	0	0	0

Binary numbers

decimal	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1

Binary numbers

decimal	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0

Binary numbers

decimal	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1

Binary numbers

decimal	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4				
5				
6				
7				
8				
...				

what are the binary reps. for 4-8?

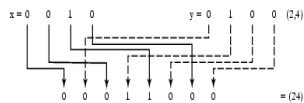
Binary numbers

decimal	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
...				

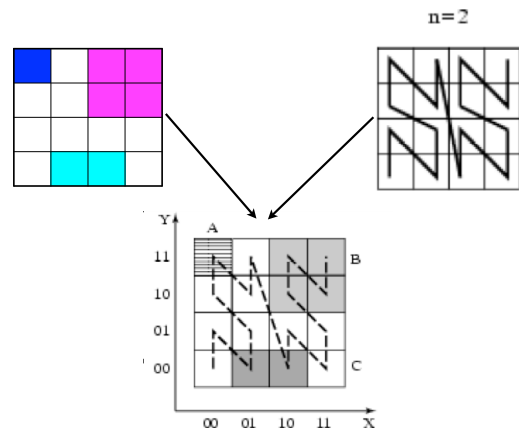
Z-Curve

binary rep. of 2: 0010
 binary rep. of 4: 0100

a Z-curve
 Generated from interleaving bits
 binary representations
 of x and y coordinates

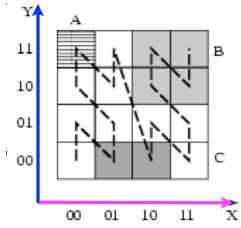


Example of Z-values



The lecture notes are based on slides by Shashi Shekhar

Example of Z-values



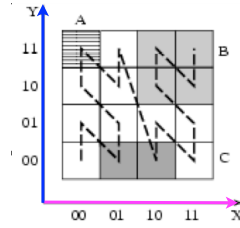
the right-most y-bit

2^3	2^2	2^1	2^0	Obj	Z
0	1	0	1	A	5

the right-most x-bit

The lecture notes are based on slides by Shashi Shekhar

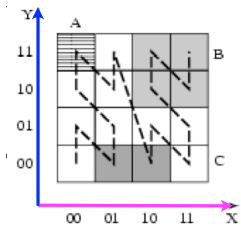
Example of Z-values



2^3	2^2	2^1	2^0	Obj	Z
0	1	0	1	A	5
				C1	
				C2	
				B1	
				...	

The lecture notes are based on slides by Shashi Shekhar

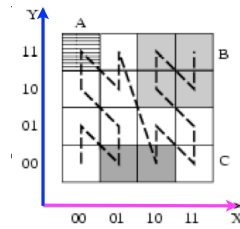
Example of Z-values



2^3	2^2	2^1	2^0	Obj	Z
0	1	0	1	A	5
0	0	1	0	C1	2

The lecture notes are based on slides by Shashi Shekhar

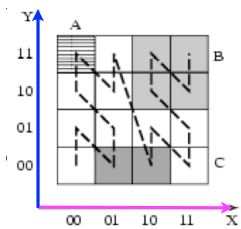
Example of Z-values



2^3	2^2	2^1	2^0	Obj	Z
0	1	0	1	A	5
0	0	1	0	C1	2
1	0	0	0	C2	8

The lecture notes are based on slides by Shashi Shekhar

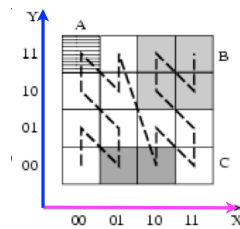
Example of Z-values



2^3	2^2	2^1	2^0	Obj	Z
0	1	0	1	A	5
0	0	1	0	C1	2
1	0	0	0	C2	8
1	1	0	0	B1	12
...

The lecture notes are based on slides by Shashi Shekhar

Example of Z-values

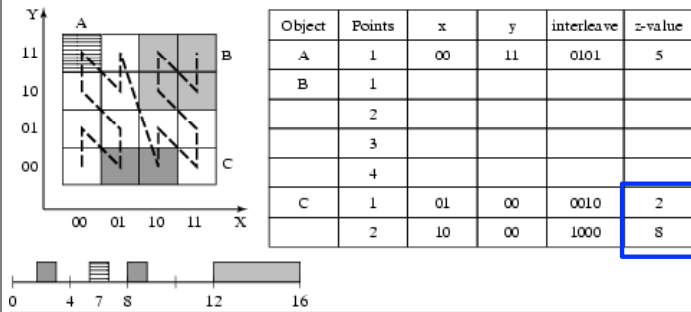


2^3	2^2	2^1	2^0	Obj	Z
0	1	0	1	A	5
0	0	1	0	C1	2
1	0	0	0	C2	8
1	1	0	0	B1	12
...



Example of Z-values

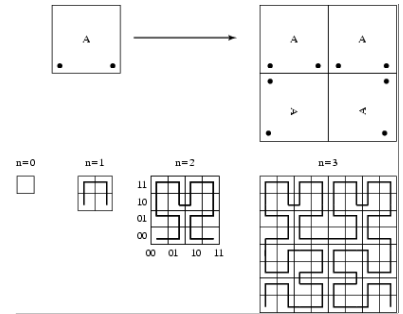
Note: C gets z-values of 2 and 8, which are **not close**



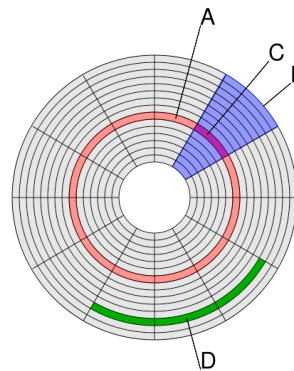
Hilbert Curve

Fig 4.5

- A space filling curve
 - Example: Fig. 4.5
- More complex to generate
 - due to rotations
 - See details on pp. 92-93
 - Illustration on next slide!
- Implementing file operations
 - similar to ordered files



Summary

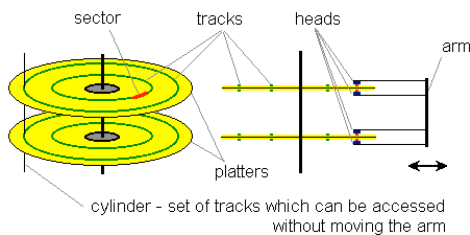


Disk structure:

- (A) track
- (B) geometrical sector
- (C) track sector
- (D) cluster

A cluster need not be physically contiguous on the disk; it may span more than one track or, if sector interleaving is used, may even be discontinuous within a track.

Storing the data

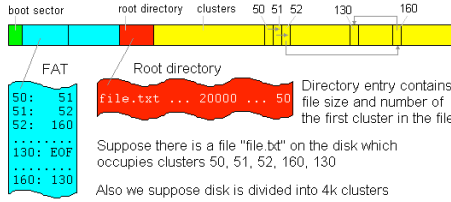


Secondary Storage Hardware: Disk Drives

- Disk hardware and organization
 - Circular platters with magnetic storage medium
 - Multiple platters are mounted on a spindle
 - Platters are divided into concentric tracks
 - A cylinder is a collection of tracks across platters with common radius
 - Tracks are divided into sectors
 - A sector size may a few kilo-Bytes
- Writing data on a disc
 - Disk heads to read and write
 - There is disk head for each platter (recording surface)
 - A head assembly moves all the heads together in radial direction
 - Spindle rotates at a high speed, e.g. thousands revolution per minute
- Accessing a sector has three major steps:
 - Seek: Move head assembly to relevant track
 - Latency: Wait for spindle to rotate relevant sector under disk head
 - Transfer: Read or write the sector
 - Other steps involve communication between disk controller and CPU

File storage

FAT-16



Directory entry contains file size and number of the first cluster in the file.

Suppose there is a file "file.txt" on the disk which occupies clusters 50, 51, 52, 160, 130

Also we suppose disk is divided into 4k clusters

The FAT table has one entry for each cluster in the partition - it contains the number of the next cluster in the chain.

There is one chain of clusters for each file.

The number of the first cluster of the file is stored in the directory entry for that file, along with the file size, some attributes, and the last modification date.

Space for the directories other than root is allocated among the data clusters, just as if they were ordinary files.

Only the root directory has a special location.

73

The lecture notes are based on slides by Shashi Shekhar

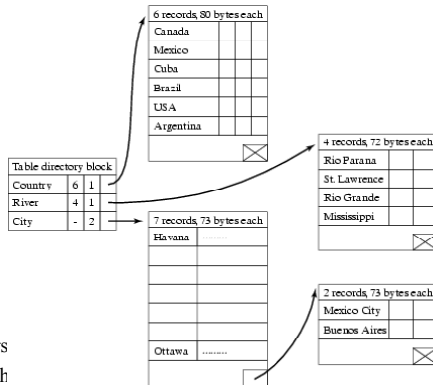
Using Disk Hardware Efficiently

- Disk access cost are affected by
 - Placement of data one the disk
 - Fact that seek cost > latency cost > transfer (See Table 4.2, pp. 86)
 - A few common observations follow
- Size of sectors
 - Larger sector provide faster transfer of large data sets
 - But waste storage space inside sectors for small data sets
- Placement of most frequently accessed data items
 - On middle tracks rather than innermost or outermost tracks
 - Reason: minimize average seek time
- Placement of items in a large data set requiring many sectors
 - Choose sectors from a single cylinder
 - Reason: Minimize seek cost in scanning the entire data set.

The lecture notes are based on slides by Shashi Shekhar

Mapping Records and files to Disk

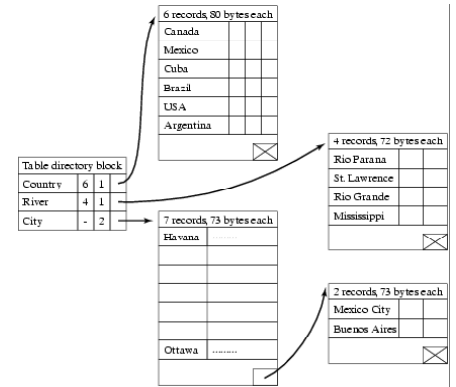
- Records
 - Often smaller than a sector
 - Many records in a sector
- Files with many records
 - Many sectors per file
- File system
 - Collection of files
 - Organized into directories
- Mapping tables to disk
 - City table takes 2 sectors
 - Others take 1 sector each



The lecture notes are based on slides by Shashi Shekhar

File operations (1)

- Find: key value --> record matching key values
- Findnext --> Return next record after find if records were sorted
- Insert --> Add a new record to file without changing file-structure
- Nearest neighbor of a object in a spatial dataset



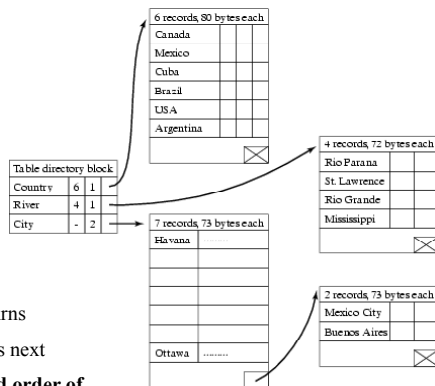
The lecture notes are based on slides by Shashi Shekhar

File operations (2)

find(Name = Canada) on Country table returns record about Canada

findnext() on Country table returns ???

findnext() on Country table returns record about Cuba since Cuba is next value after Canada in the sorted order of Name



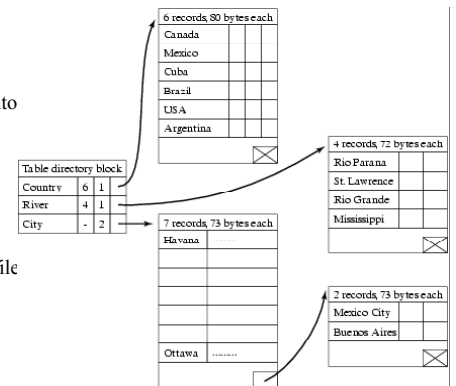
The lecture notes are based on slides by Shashi Shekhar

File operations (3)

insert(record about Panama) into Country table

adds a new record

location of record in Country file depends on file-structure

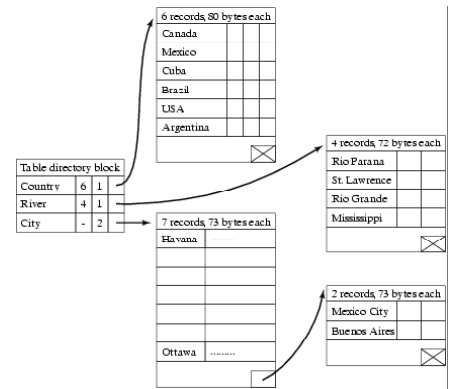


Common File Structures and how they support file operations

- Common file structures
 - Heap or unordered or unstructured
 - Ordered
 - Hashed
 - Clustered
- Basic Comparison of Common File Structures
 - Heap file is efficient for inserts and used for logfiles
 - But find, findnext, etc. are very slow
 - Hashed files are efficient for find, insert, delete etc.
 - But findnext is very slow
 - Ordered file organization are very fast for findnext
 - and pretty competent for find, insert, etc.

4.1.4 File Structures: Heap

- Heap
 - Records are in no particular order
 - insert can simple add record to the last sector
 - find, findnext, nearest neighbor scan the entire files



File Structure : Hash

•Components of a Hash file structure

- A set of buckets (sectors)
- Hash function : key value --> bucket
- Hash directory: bucket --> sector

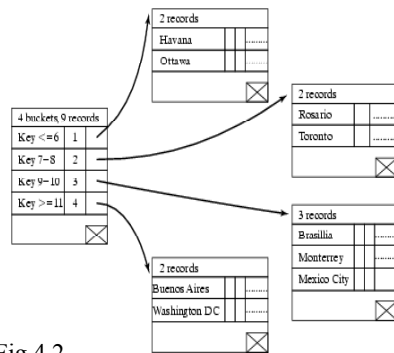


Fig 4.2

File Structures: Ordered

- Ordered
 - Records are sorted by a selected field (Example Fig. 4.3 below)
 - findnext can simply pick up physically next record
 - find, insert, delete may use binary search, is is very efficient

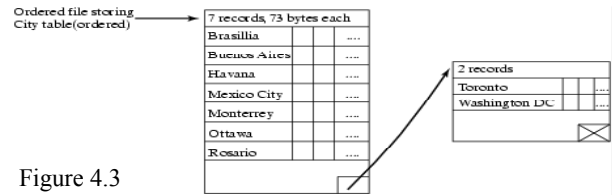


Figure 4.3

Spatial File Structures: Clustering

- Motivation:
 - Ordered files are not natural for spatial data
- Clustering records in sector by space filling curve is an alternative
- In general, clustering groups records
 - accessed by common queries
 - into common disk sectors
 - to reduce I/O costs for selected queries
- Clustering using Space filling curves
 - Z-curve
 - Hilbert-curve

Z-Curve

- What is a Z-curve?
 - A space filling curve
 - Connecting points by z-order
 - curves look like combinations of Ns or Zs

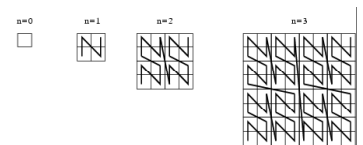
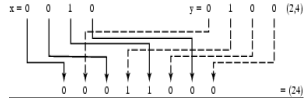


Fig 4.4

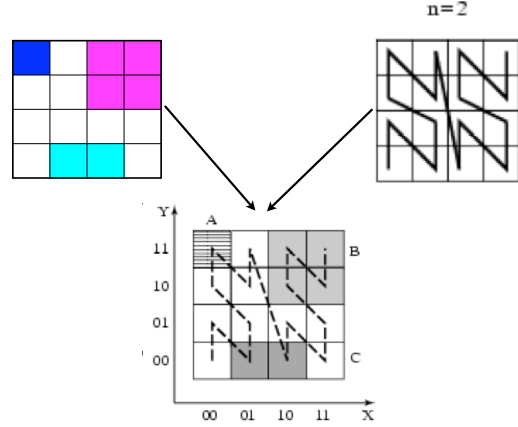
Z-Curve

binary rep. of 2: 0010
 binary rep. of 4: 0010

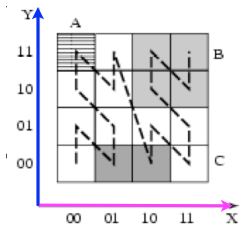
a Z-curve
 Generated from interleaving bits
 binary representations
 of x and y coordinates



Example of Z-values



Example of Z-values



2^3	2^2	2^1	2^0	Obj	Z
0	1	0	1	A	5
0	0	1	0	C1	2
1	0	0	0	C2	8
1	1	0	0	B1	12
...

