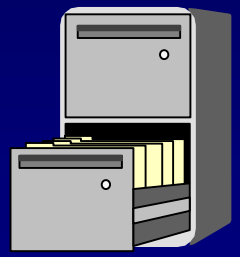
File Structures

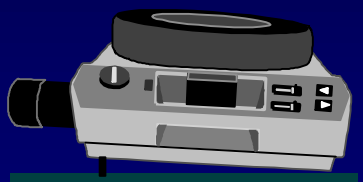
An Introduction



สมชาย ประสิทธิ์จูตระกูล

Outline

- Introduction
- Basic Concepts
- Secondary Storage
- Sequential Files
- Direct Files
- Indexed Files
- Tree-Based Files
- Multilist & Inverted Files



Managing Large Quantities of Data

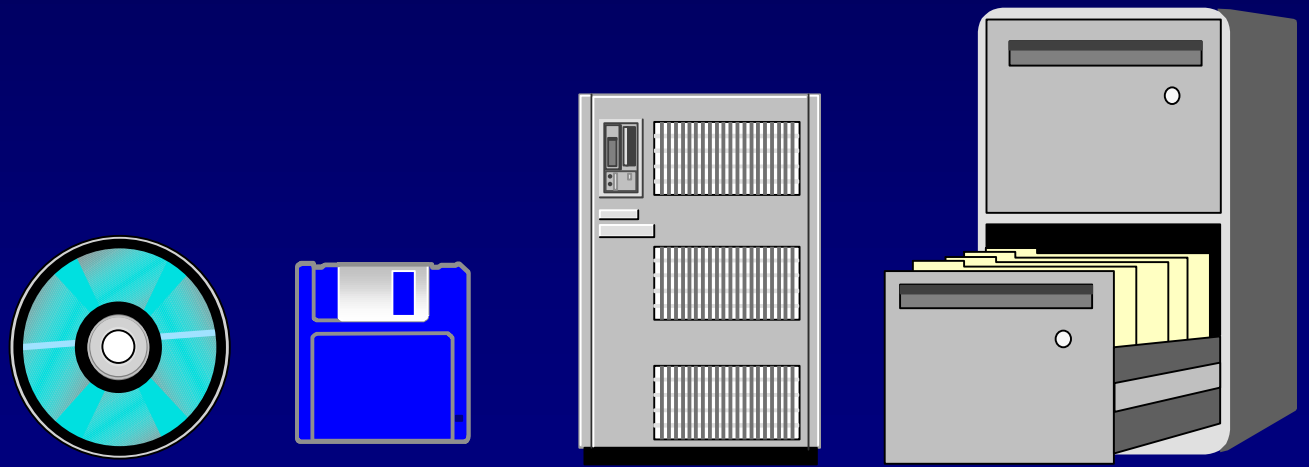
- Accessed by multiple people and programs
- Kept on external storage devices
- Always reliably available for processing
- Rapidly accessible when information is needed

Speed & Capacity

- Disks are slow.
  - RAM ≈100 ns
  - Disk ≈10 ms
- Disks provide enormous capacity.
  - RAM ≈10 MB (volatile)
  - Disk ≈1000 MB (nonvolatile)

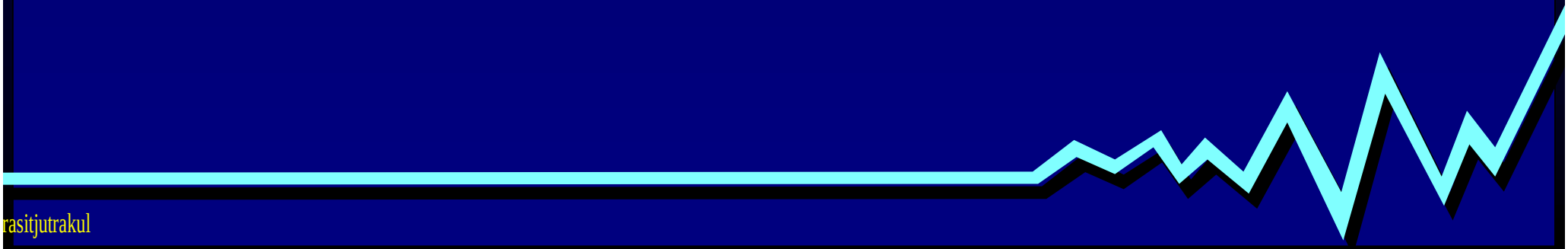
Design Goal

Minimizing disk accesses for files that keep changing in content and size.



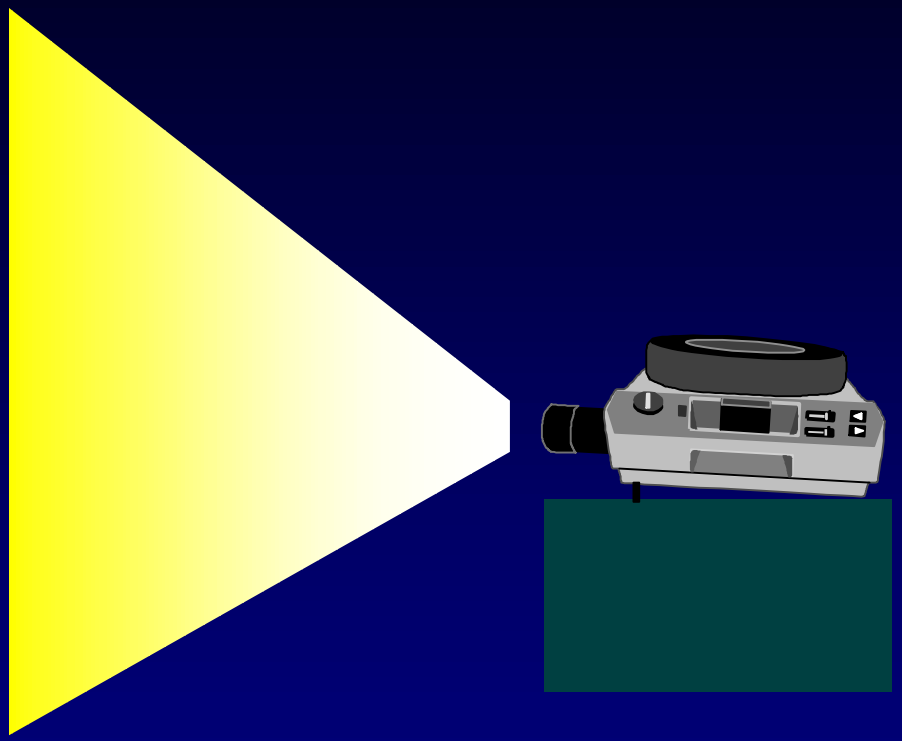
A Short History

- 1950s : Sequential access + indexes
- 1960s : Tree Structures
- 1970s : B-tree
- 1980s : Extendible Hashing

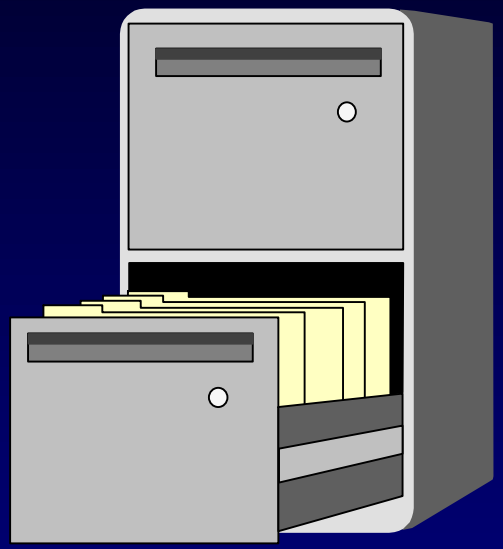


Basic Concept : Outline

- Files
- Records, Fields
- Keys
- Users
- File Processing
- File Design

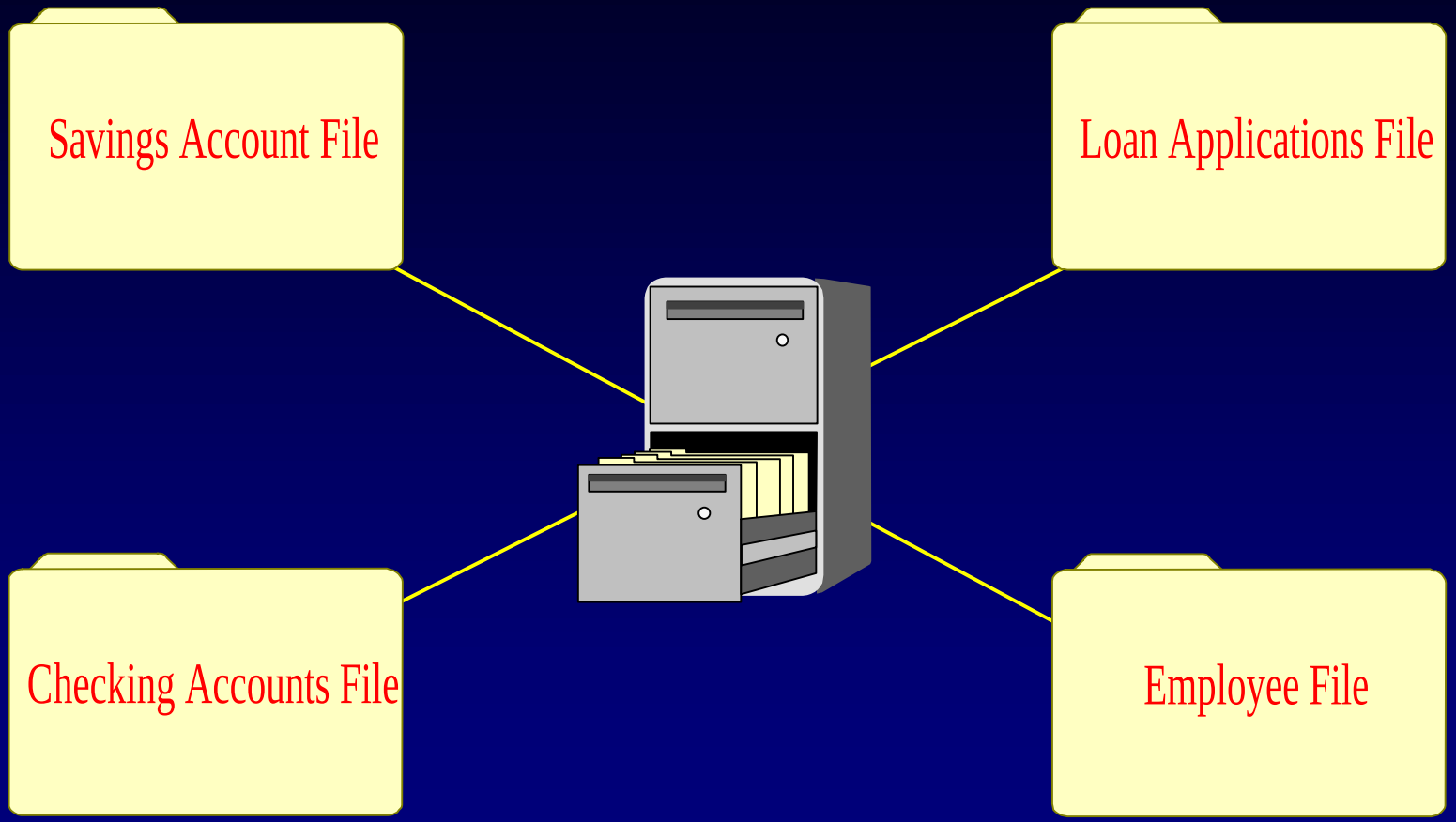


Filing System



- Size
- Persistence
- Sharability

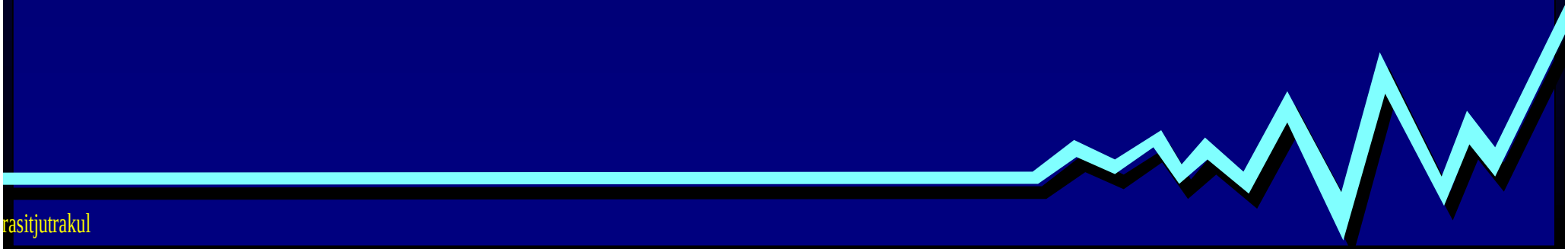
Files


Savings Account File
Loan Applications File
Checking Accounts File
Employee File

Records

Account	Name	Address	Balance
018-745-96	Thongdee	36 Sathon, 10600	25,250.93
108-964-09	Dundee	488 Rama 4, 10330	2,252.00
116-057-43	Yudee	56 Chareonkrung, 10210	99,768.25
248-922-88	Wangdee	102 Bantadthong, 10330	125,899.29
741-673-76	Dundee	77 Saphanluang, 10330	232.48

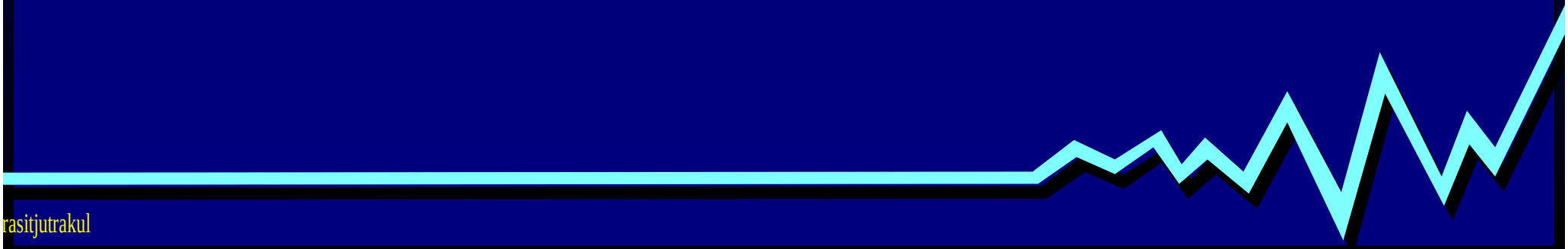
Checking Accounts File



Fields

Account	Name	Address	Balance
018-745-96	Thongdee	36 Sathon, 10600	25,250.93
108-964-09	Dundee	488 Rama 4, 10330	2,252.00
116-057-43	Yudee	56 Chareonkrung, 10210	99,768.25
248-922-88	Wangdee	102 Bantadthong, 10330	125,899.29
741-673-76	Dundee	77 Saphanluang, 10330	232.48

Checking Accounts File



Files & Records

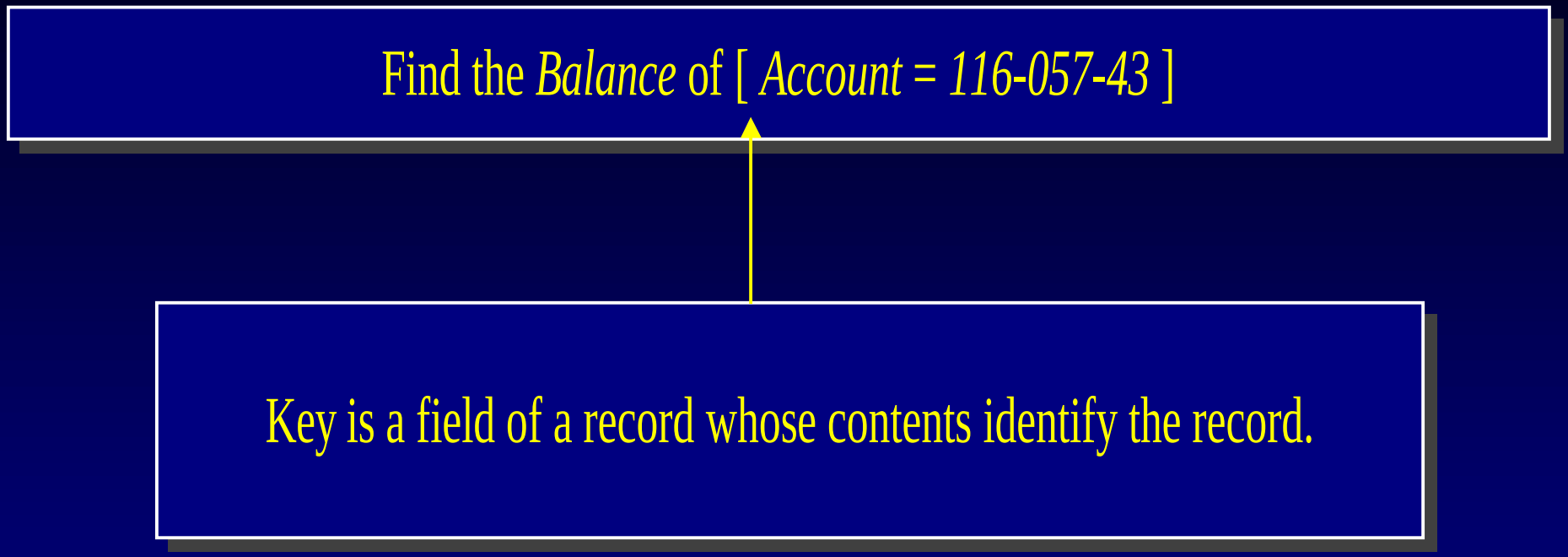
- A file is a collection of records of the same type.
- A record is a collection of related fields.

Keys

Find the *Balance* of [*Account = 116-057-43*]

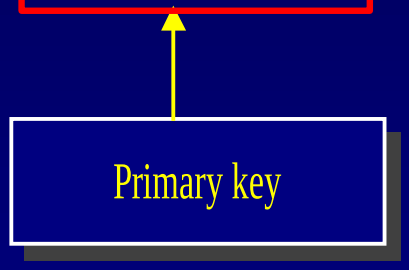
- Locate the Checking Account file.
- Access the record whose contents of the *Account* field = 116-057-43.
- Retrieve the record from the file.
- Examine the contents of the *Balance* field.

Keys


Find the *Balance* of [*Account = 116-057-43*]
Key is a field of a record whose contents identify the record.

Primary Keys

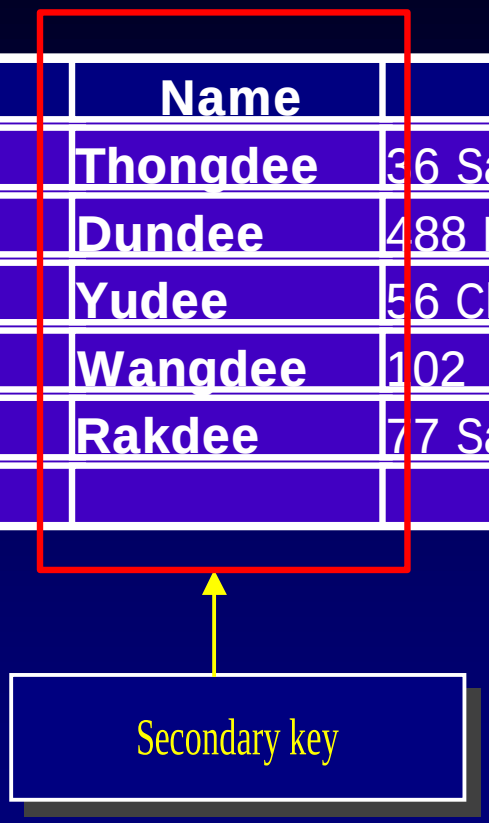
Account	Name	Address	Balance
018-745-96	Thongdee	36 Sathon, 10600	25,250.93
108-964-09	Dundee	488 Rama 4, 10330	2,252.00
116-057-43	Yudee	56 Chareonkrung, 10210	99,768.25
248-922-88	Wangdee	102 Bantadthong, 10330	125,899.29
741-673-76	Rakdee	77 Saphanluang, 10330	232.48



A primary key is a field that *uniquely* identify the record.

Secondary Keys

Account	Name	Address	Balance
018-745-96	**Thongdee**	36 Sathon, 10600	25,250.93
108-964-09	**Dundee**	488 Rama 4, 10330	2,252.00
116-057-43	**Yudee**	56 Chareonkrung, 10210	99,768.25
248-922-88	**Wangdee**	102 Bantadthong, 10330	125,899.29
741-673-76	**Rakdee**	77 Saphanluang, 10330	232.48

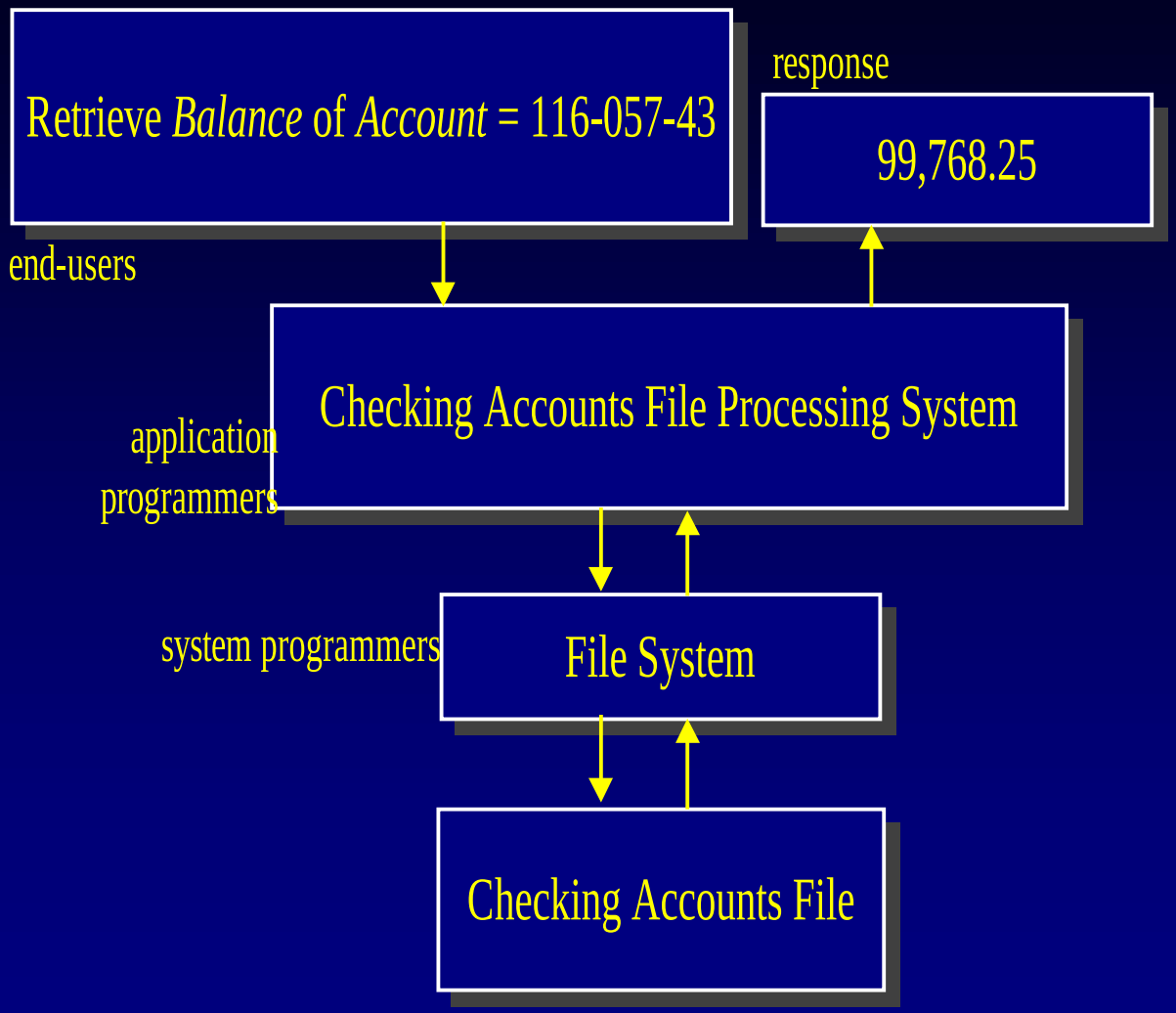


A secondary key is a field that does identify the record, but this identification is not unique.

Users

- End-users
- Application programmers
- System programmers

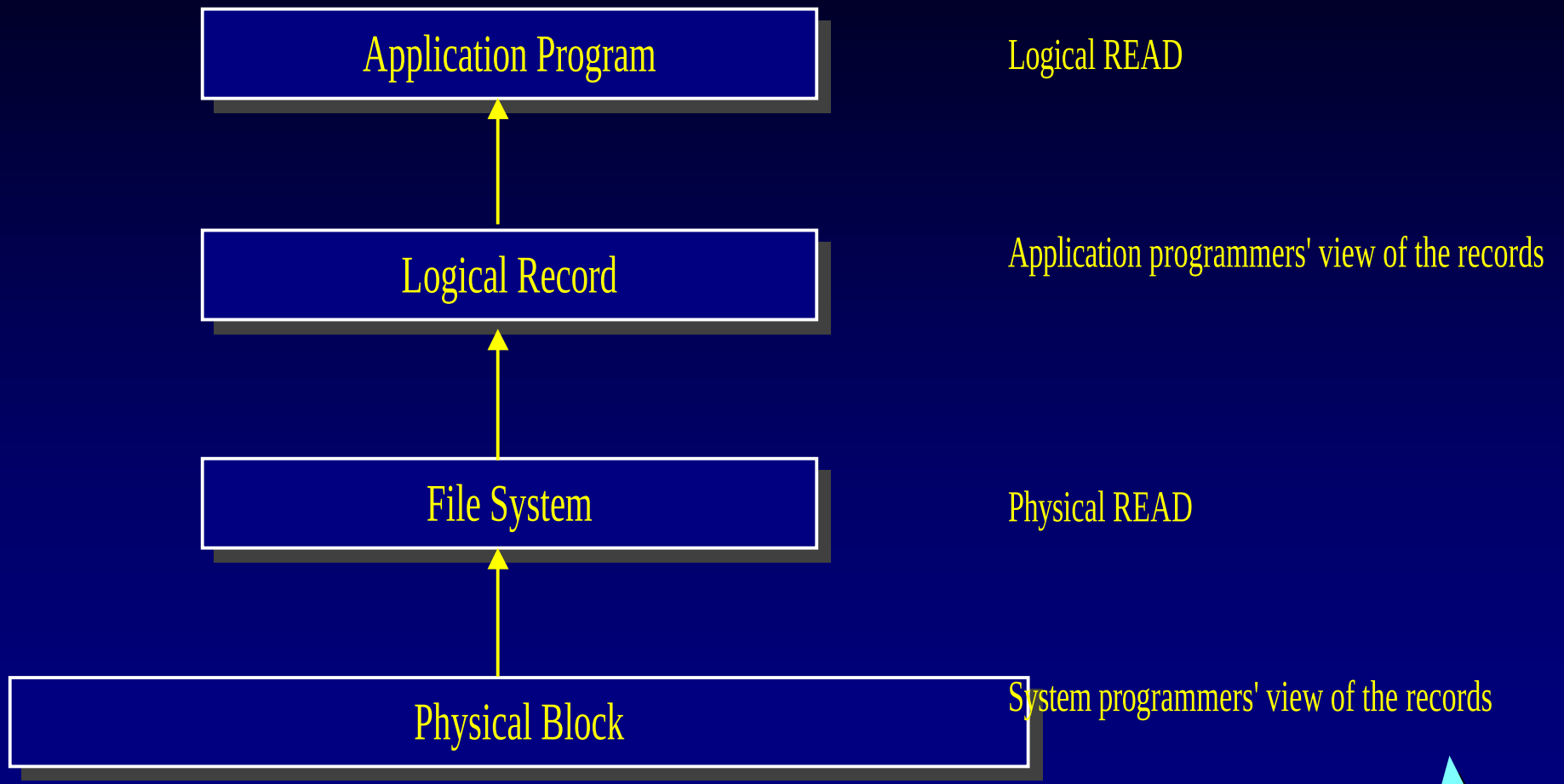
File Processing Systems



Users' Concerns

- End-users
  - receive accurate information.
- Application programmers
  - aware of the file organization, record structure, and access mechanisms.
- System programmers
  - aware of the available tools and resources to enhance the file system efficiency.

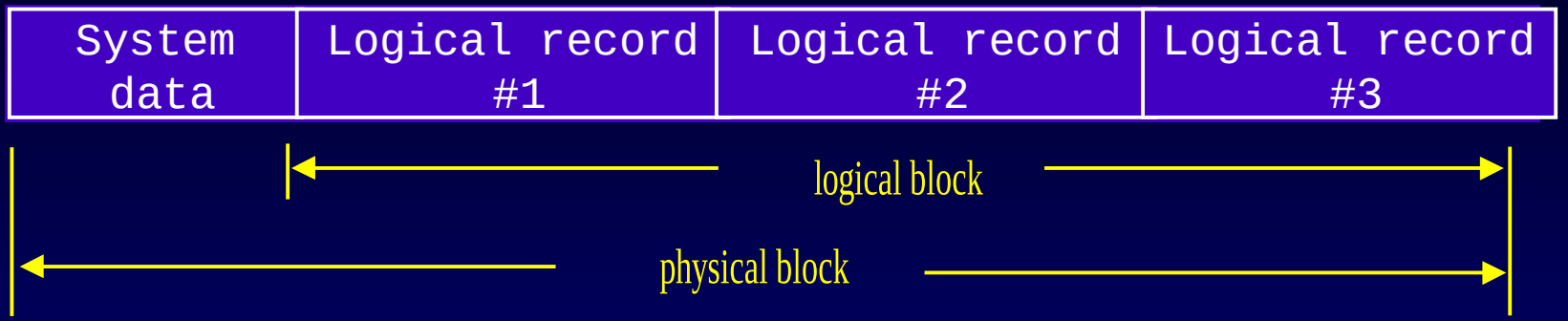
Data Transfer



Logical Records

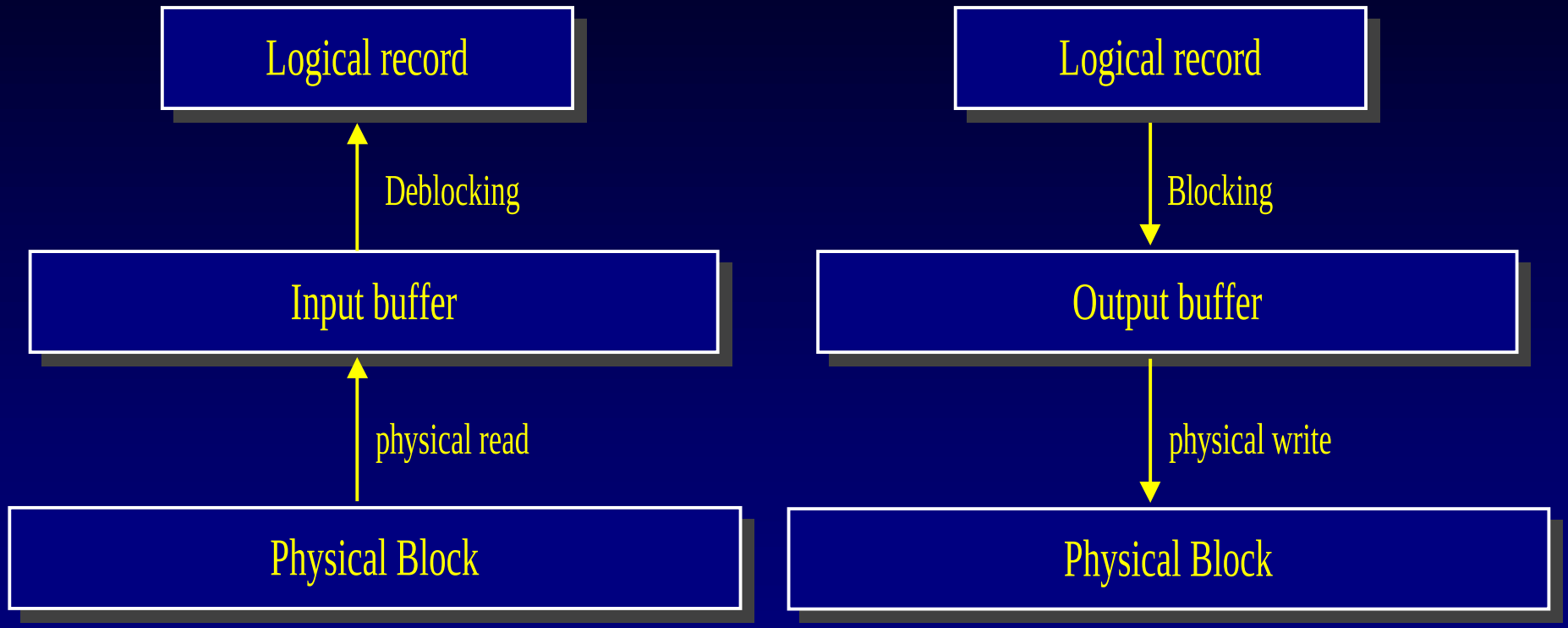
```
typedef struct customerTag {
        int     iAccount;
        char    szName[20];
        char    szAddress[50];
        float   fBalance;
} recCustomer;

recCustomer    CustomerRecord;
```

Physical Blocks

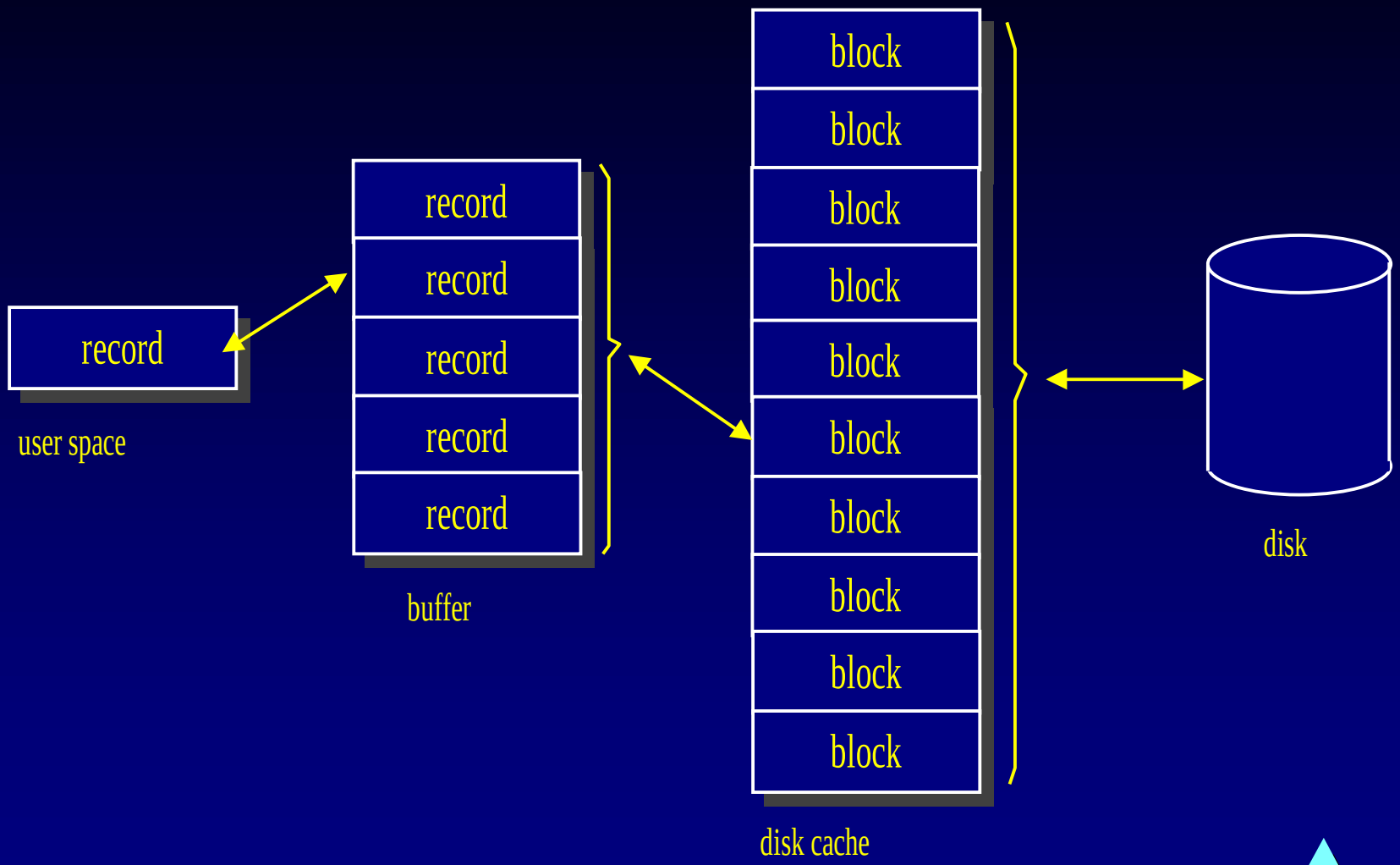


Blocking factor

Blocking & Deblocking



Disk Caching



Blocking Factor

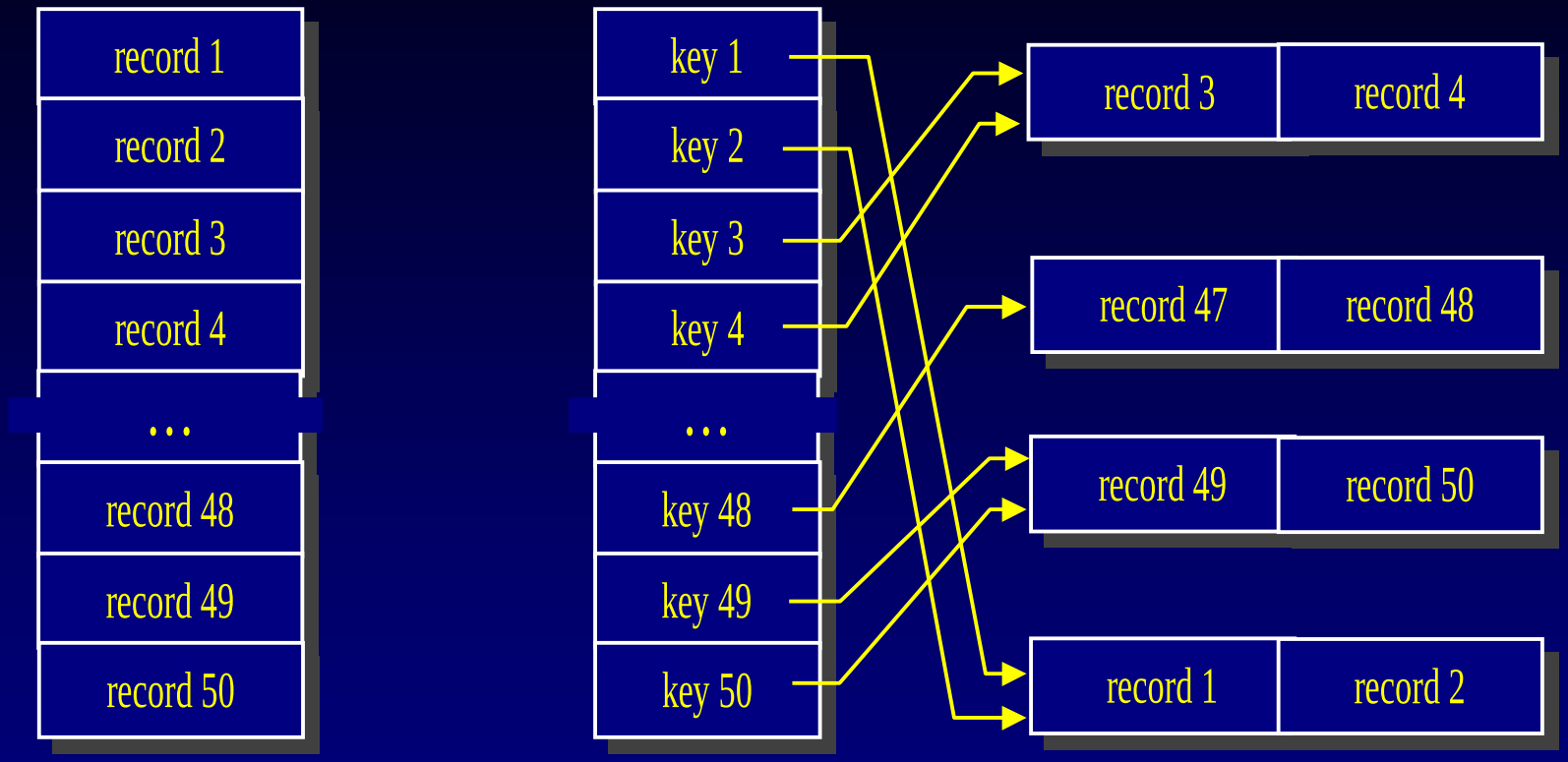
- Blocking factor vs # Block transfers
- Blocking factor vs Buffer size
- Optimal blocking factor

If the blocking factor were equal to the number of logical records then one could successfully argue that only one data transfer would be needed !!!

Logical & Physical File Structure

- Logical file structure
  - The organization of all logical records in the file.
- Physical file structure
  - The organization of all the physical blocks stored in secondary storage.

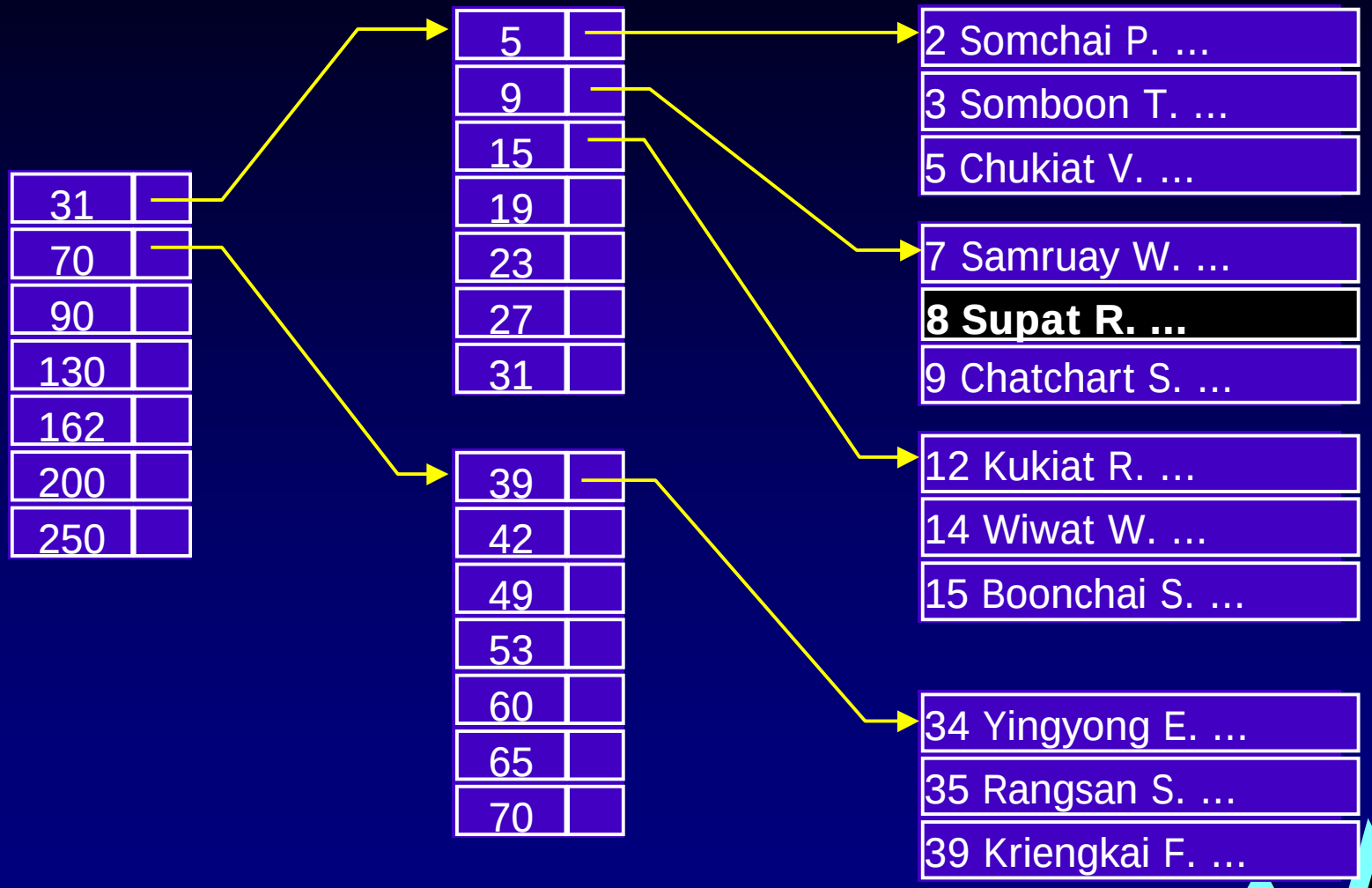
Logical & Physical File Structure



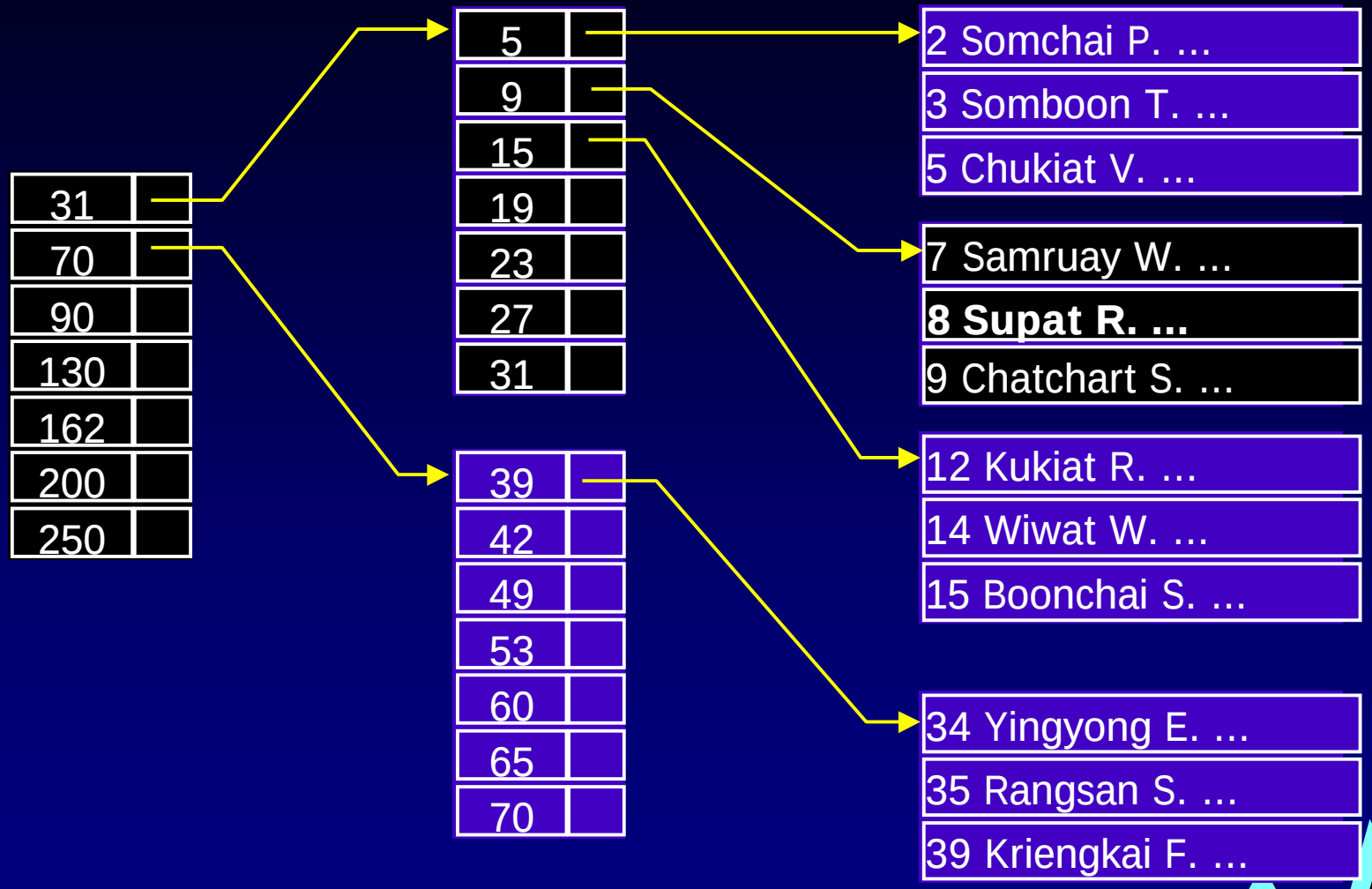
sequential file

physical linked sequential file

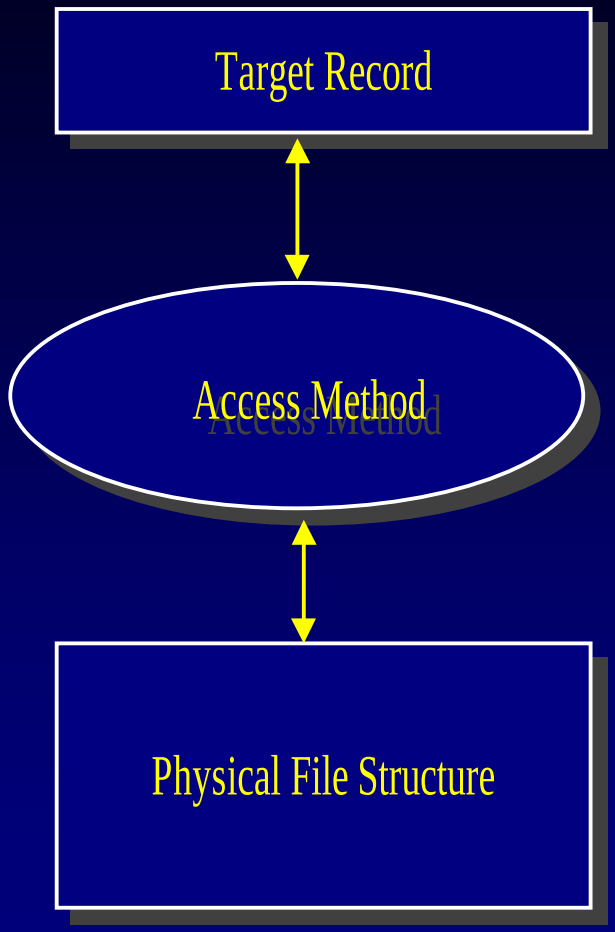
Access Path


31
70
90
130
162
200
250
5
9
15
19
23
27
31
39
42
49
53
60
65
70
2 Somchai P. ...
3 Somboon T. ...
5 Chukiat V. ...
7 Samruay W. ...
8 Supat R. ...
9 Chatchart S. ...
12 Kukiat R. ...
14 Wiwat W. ...
15 Boonchai S. ...
34 Yingyong E. ...
35 Rangsan S. ...
39 Kriengkai F. ...

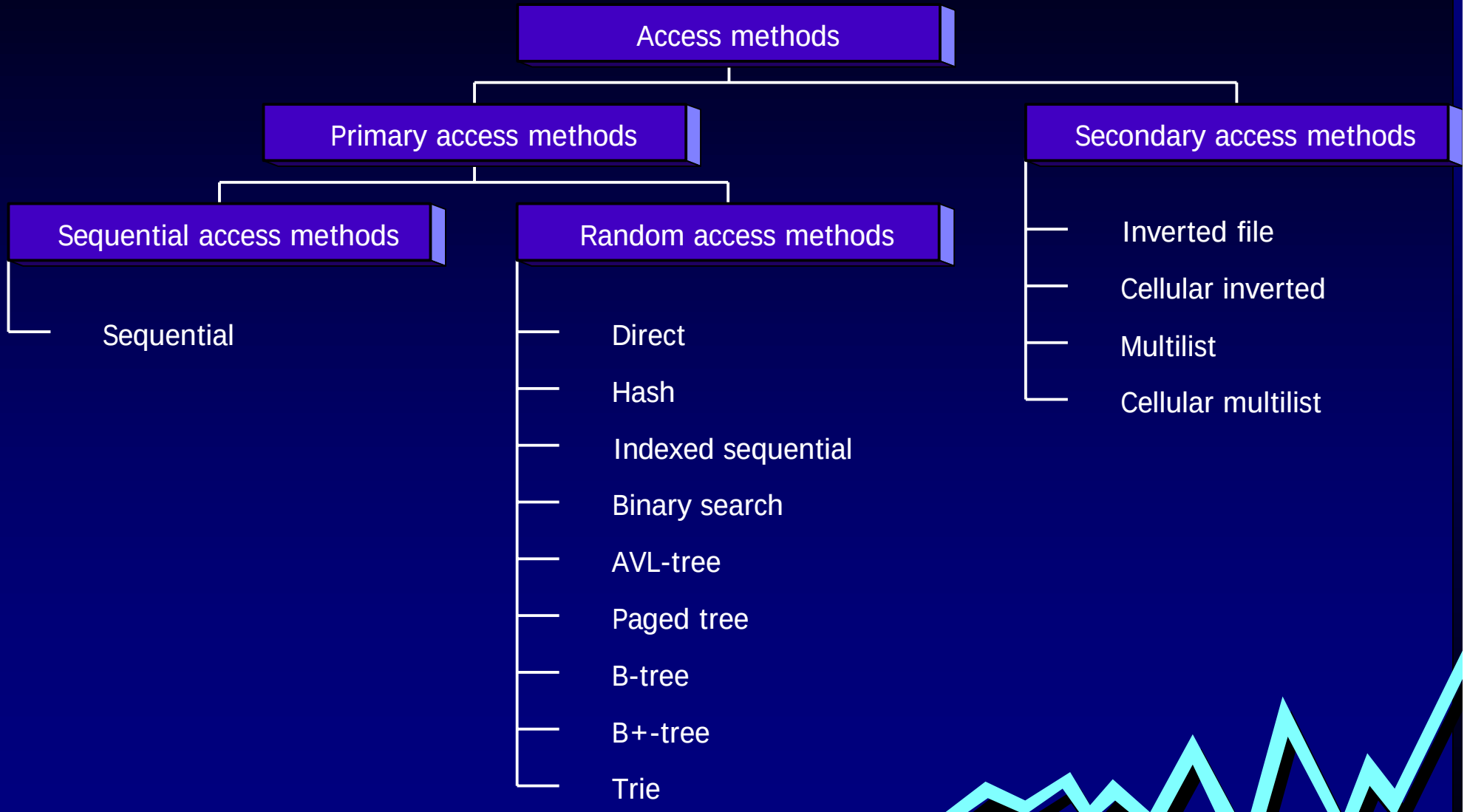
Access Path


31
70
90
130
162
200
250
5
9
15
19
23
27
31
39
42
49
53
60
65
70
2 Somchai P. ...
3 Somboon T. ...
5 Chukiat V. ...
7 Samruay W. ...
8 Supat R. ...
9 Chatchart S. ...
12 Kukiat R. ...
14 Wiwat W. ...
15 Boonchai S. ...
34 Yingyong E. ...
35 Rangsan S. ...
39 Kriengkai F. ...

Access Methods



Classification of Access Methods

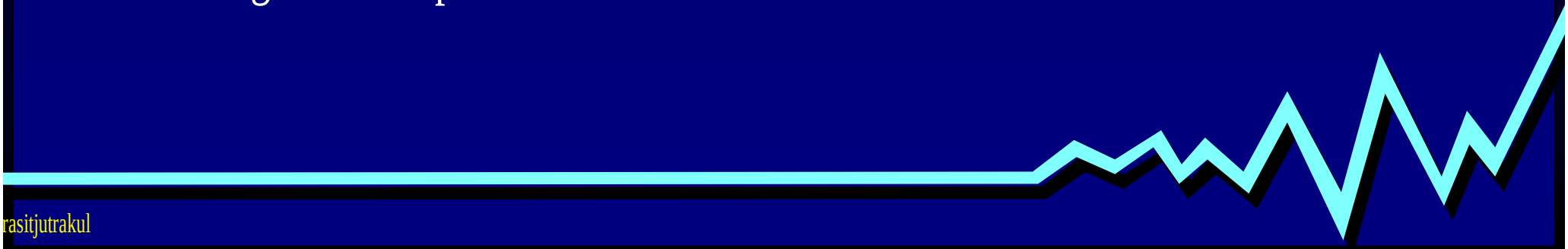


File Design

- Logical file design
  - select one of the available file organizations
  - design a new file organization
- Physical file design
  - design the physical file

File Design

- Selection of blocking factor
- Allocation of the I/O buffers
- Size of the physical file
- Organization of the physical blocks
- Design or selection of the access method
- Selection of the primary key
- File growth
- Reorganization point



File Operations

- *RetrieveAll*
- *Batch*
- *RetrieveOne*
- *RetrieveNext*
- *RetrievePrevious*
- *InsertOne*
- *DeleteOne*
- *UpdateOne*
- *RetrieveFew*

Performance

- Response time
  - The type of allowable operations.
  - The frequency of each type of operation.

Ex. 95% **`Retrieve_One`**

5% **`Batch`**

Random or Sequential ?

Search length