Nguyen Tat Thanh Institute of International Education (NIIE)

DATABASE MANAGEMENT SYSTEMS

(Creadits 3)

MSc. Luong Tran Ngoc Khiet

May - 2021

Course content

Chap 2. Data storage management

MSc. Luong Tran Ngoc Khiet

NTT Institute of International Education (NIIE)

Chap 2. Data Storage management

1. **Database organization**

2. **Index**

3. **Triggers***

1. Database organization

1.1 File organization

1.2 Organize news items

1.3 File organization includes unordered records (Heap File)

1.4 File organization consists of ordered records (Sorted File)

1.5 SAN (Storage Area Network)

1.6 RAID (Redundant Arrays of Independent Disks)

1.1 File organization

- Storage types:
  - Primary storage
  - Secondary storage

1.1 File organization

❑ Primary storage

- A form of storage that the CPU can manipulate directly. For example:
  - *main memory of the computer,*
  - *memory is used for cache*
- Access speed is fast but has limited storage capacity and high price

1.1 File organization

❑ Secondary storage

- A form of storage that the CPU cannot manipulate directly (data must be transferred to primary storage). For example: magnetic disk, optical disk, magnetic tape
- Access speed is slower than primary storage
- Higher storage capacity, lower cost

Types of memory organization - Primary storage

static RAM (Random Access Memory)

- Random access memory (time to read/write memory cells is the same)
- Memory that allows reading and writing (data being changed or in use)
- Data on RAM will be lost when there is a power outage

Types of memory organization - Primary storage

Cache memory

- It is RAM but stores data from previous reads
- When the program needs to read data, it can read it in the cache => program execution will be fast.

Types of memory organization - Primary storage

DRAM (dynamic RAM)

- Is the main working area for the CPU (main memory)
- Stores programs and data
- Access speed is slower than RAM but the price is cheaper

1.1 File organization

Types of disk organization - **Secondary storage**

Includes all types

- CD-ROM: Compact Disk Read Only
- Đĩa quang: Optical disk
- Đĩa từ: Magnetic disk
- Băng từ: Magnetic tape

1.1 File organization

- Designer, installer and administrator:
  - Must understand storage organization techniques
  - Know the pros and cons of these techniques
- Databases are physically organized
  - Files of records
  - Each piece of information is considered an entity
  - For example, each record is a student, with attributes such as code, full name, address...

1.2 Record organization

- What are records and types of records?
  - A record is an entity and has data fields.
  - Each field has a data type
  - Basic data types such as string, number, date, logic
  - Special data types such as images, sounds, movies, etc.
  - The collection of all field names along with their data types is called the record type.

1.2 Record organization

Sample a record student

student Tran Son Hai	
Id	A001
fullname	Tran Son Hai
dob	09/04/1981
gender	Male
address	46 Tân Hải
mobile phone	0903080808
scholarship	10.000.000

1.2 Record organization

Sample a record student

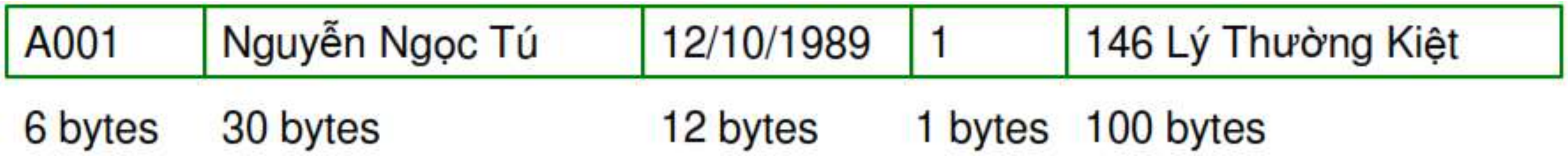
```
structure STUDENT
{
    string  id;
    string  fullname;
    date    dob;
    boolean gender;
    string  address;
    string  mobile;
    real    scholarship;
}
```

1.2 Record organization

- In a file, if the size of all records is the same, it is called a fixed-length record.
- Otherwise, it is called a variable-length record

1.2 Record organization

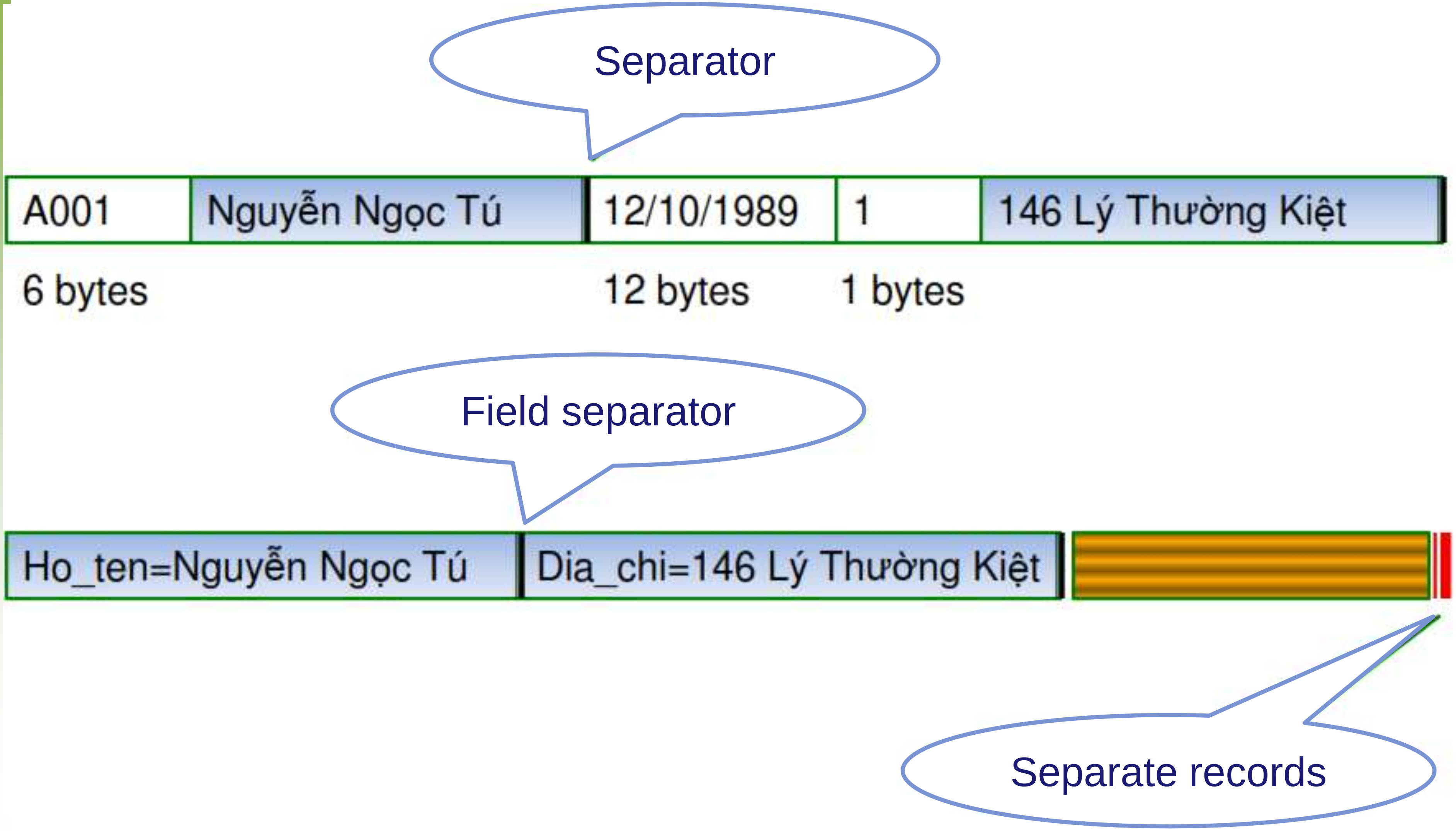
- For example, a record has a fixed length



A001	Nguyễn Ngọc Tú	12/10/1989	1	146 Lý Thường Kiệt
6 bytes	30 bytes	12 bytes	1 bytes	100 bytes

1.2 Record organization

- For example, the record has a variable length



STORAGE TEMPLATES IN BLOCKS

- Block
  - A data unit used to transfer data between disk and memory. For example, the block is 8 Kbytes in size
  - The pieces of information in the file will be stored in disk blocks.
  - A block can store many records (when the size of the block is larger than the size of the record).

1.2 Record organization

STORAGE TEMPLATES IN DIVISIONED BLOCKS

- Records can be stored on multiple blocks
- If B is the block size (in bytes) and R is the record size (fixed record length). And B>=R then:
- The block allows to contain up to B/R pieces of information. If B/R has a non-zero remainder, additional space must be allocated for this remainder (a new block must be allocated).
- If the length of the record changes, it can be stored divided or undivided.

1.2 Record organization

STORAGE TEMPLATES IN DIVISIONED BLOCKS

- The piece of information is stored in one block and the rest is stored in another block
- At the end of the first block there will be a pointer pointing to the address of the next block
- If the size of the records is large, partitioned storage will save block space usage.

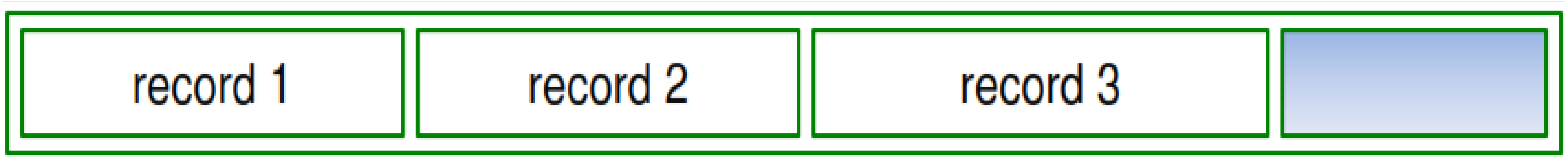
STORAGE TEMPLATES IN DIVISIONED BLOCKS

- When records do not allow division across blocks
- If the record length is fixed and B>R then the records will be stored in a block and have a starting address.
- This starting address is calculated as follows:

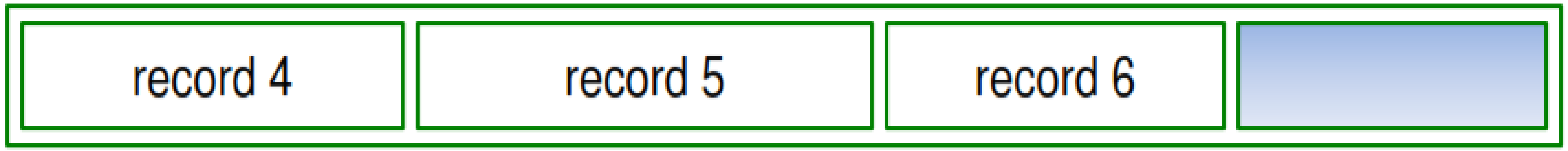
Let B be the size of the block
Let R be the size of the record
So N = B/R is the number of records contained in the block
If F is the total number of pieces of information, then the number of blocks needed M = F/N
The blocks are numbered from 0 to M-1
The I record will have the address: I div N + I mod N

1.2 Record organization

For example, storing records in undivided blocks



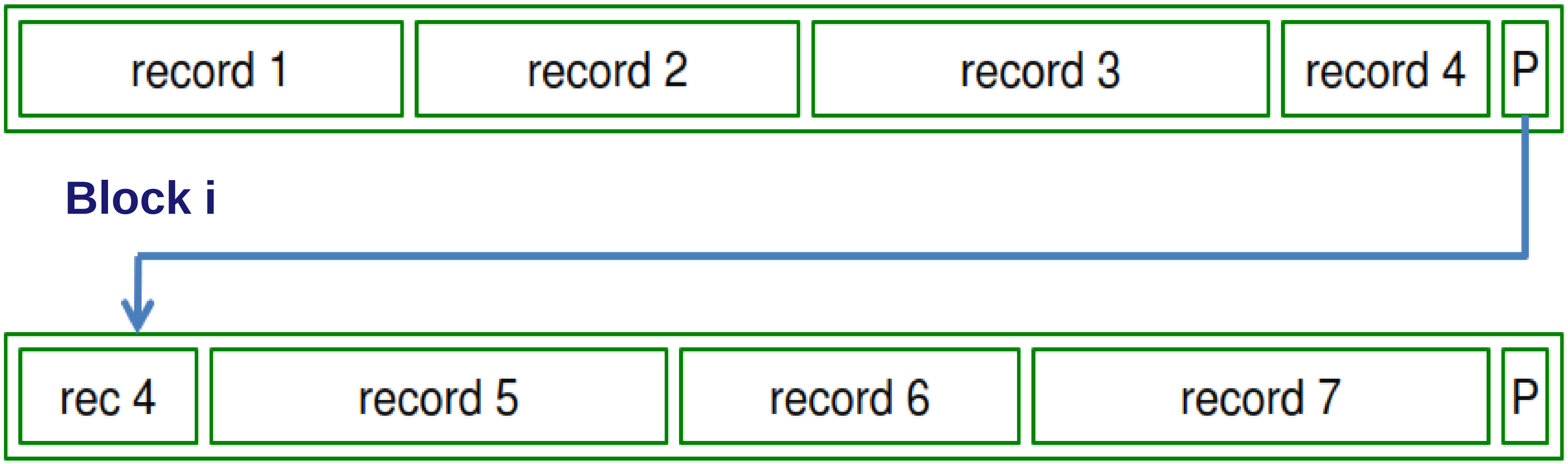
Block i



Block i+1

1.2 Record organization

For example, storing records in divided blocks



Block i

Block i

1.2 Record organization

❑ Types of block allocation

Continuous dispensing -Cấp phát liên tục

- Blocks on disk will be allocated contiguously
- Advantage: when reading the entire file, it will be read very quickly
- Disadvantage: difficulty increasing file size

1.2 Record organization

❑ Types of block allocation

Link allocation - Cấp phát liên kết

- Each block will have a pointer, pointing to the next block
- Advantages: easy to increase file size
- Disadvantage: reading the entire file will be slow

1.2 Record organization

❑ **Types of block allocation**

Cluster - Cấp phát xâu

- Is a sequence of blocks, also known as a segment file or an extension file
- Allocation blocks can be adjacent or linked

For example: Storage organization - SQL Server

- File manager
  - .mdf : meta data file
  - .ldf : log data file
  - .bak : bakup data file
- Disk management
- Department of physical data management

For example: Physical Organization - SQL Server

- Data in the database is organized into logical components for users to use such as: Table, View...
- Physical data can be stored on multiple files or multiple drives.
- Users (except DBAs) work only on logical components of SQL Server.

1.2 Organize news items

❑ File header:

Is descriptive information for the file. Include:

- Addresses of storage blocks on disk.
- Describes record formats such as field sizes and order of fields.

1.2 Organize news items

❑ File header:

When searching for news:

- One or more blocks are read into the main memory buffer
- The program performs searches on main memory
- If the record is still not found, the blocks continue to be read into main memory
- The search will end when a record is found or no record is found in the file
- The larger the file, the longer it will take to search

1.2 Organize news items

- A relation is a set of records. With a set of records, the problem is how to organize them in a file.
  - Heap File Organization: A record is stored in any file. There is no order of the records. Usually a file stores a relationship.
  - Sorted File Organization: Records are organized sequentially according to the value of a search key in each record.
  - Hash file organization: Use a hash function that calculates on some properties of the record and uses the results to determine the disk block in which the record is stored.

1.3 File organization includes unordered records

- Heap File:

- Is the simplest type of file organization
- The records are not sorted.
- Adding a new record is simple:
  - Find the last disk block of the file, copy the disk block to the buffer, add a new record and then write it back to disk.
  - The address of the last disk block is updated back into the header file.

1.3 File organization includes unordered records

❑ Heap File:

- Edit record: delete old record and add new record.
- Delete record: locate the block containing the record you want to delete, read into the buffer, delete this record in the block, write the block back to disk.

If you delete many records, it may become fragmented

⇨ the solution is to just mark them as deleted

1.3 File organization includes unordered records

❑ Heap File

- Because the records in the file are not arranged in order, when searching, you must use the sequential search method: Read each disk block into main memory one at a time and then search for the records.

➭ Thus, if the file consists of b disk blocks, the average search time will be n/2.

1.4 File organization includes ordered records

- Sorted File

- Records are ordered by the value of one or more fields.
- If the ordered field is a key field, it is called an ordered key (the value must be unique).
- Organizing files in order is also called sequential files

1.4 File organization includes ordered records

- **Sorted File**
  - Convenient for searching, can find information quickly using binary search method.
  - When adding records, an appropriate location for the new piece of information must be determined.
    - Method 1: declare excess free space
    - Method 2: add new record to sub file

1.4 File organization includes ordered records

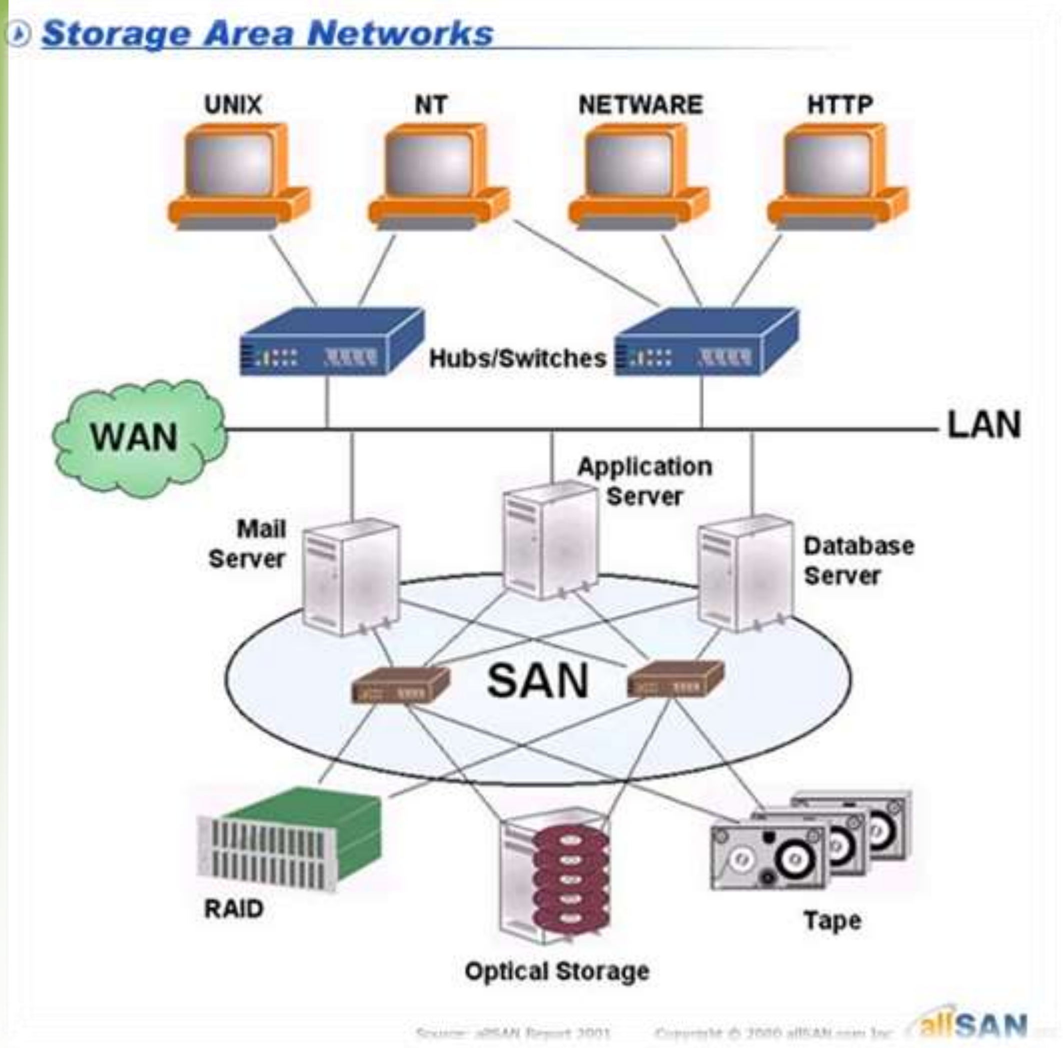
❑ Sorted File

- When editing a record: If you edit the values of the key field, there are two operations: delete the old record and add a new record.
  - ⇨ Restrict editing of key field values
- When deleting records: Deleting many records can lead to fragmentation
  - ⇨ Solution is to just mark delete

1.5 SAN (Storage Area Network)

- SANs is a network designed to make adding storage devices to servers as easy as Disk Array Controllers or Tape Libraries.
- SANs are designed to easily utilize storage features, allowing multiple servers to share a single storage device..

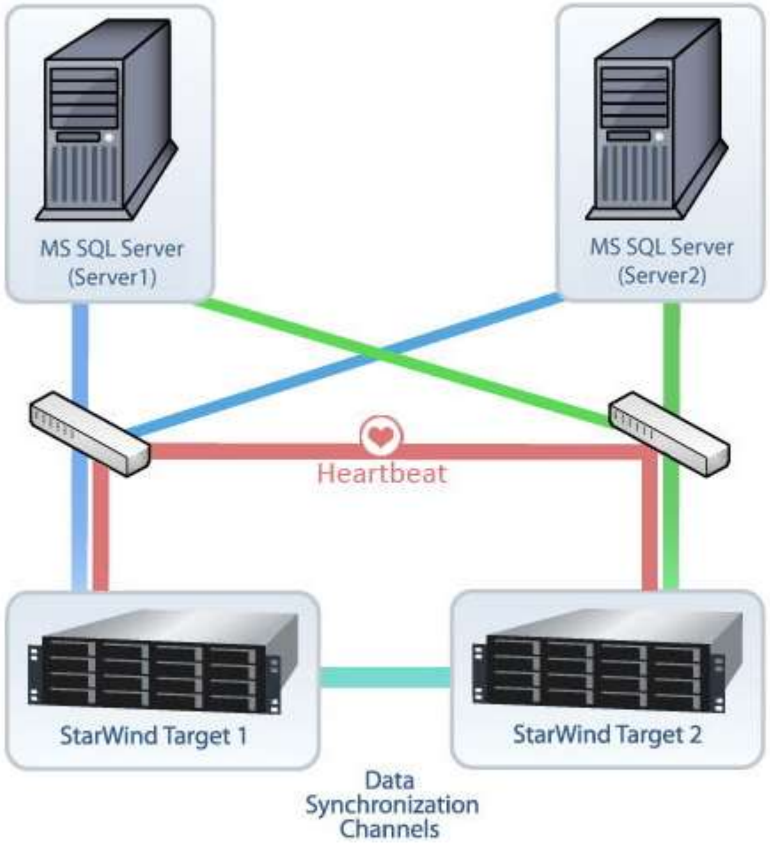
1.5 SAN (Storage Area Network)



For example:

StarWind iSCSI SAN Solution

for Microsoft SQL Server



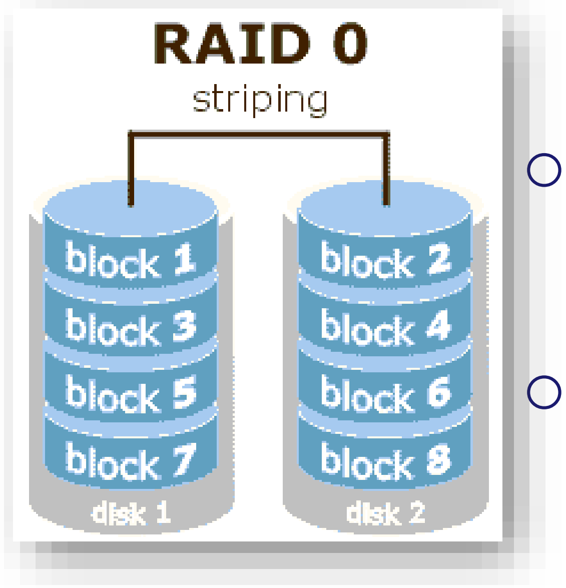
1.5 SAN (Storage Area Network)

- **Benefits of using SANs**
  - Easily manage, share, and expand capabilities through the process of adding storage devices to the network without having to change servers or storage devices.
  - SANs allow multiple servers to share a single storage device.
  - SANs allow servers to be replaced while in use without affecting data.
  - SANs provide quick data recovery solutions.

1.6 RAID (Redundant Arrays of Independent Disks)

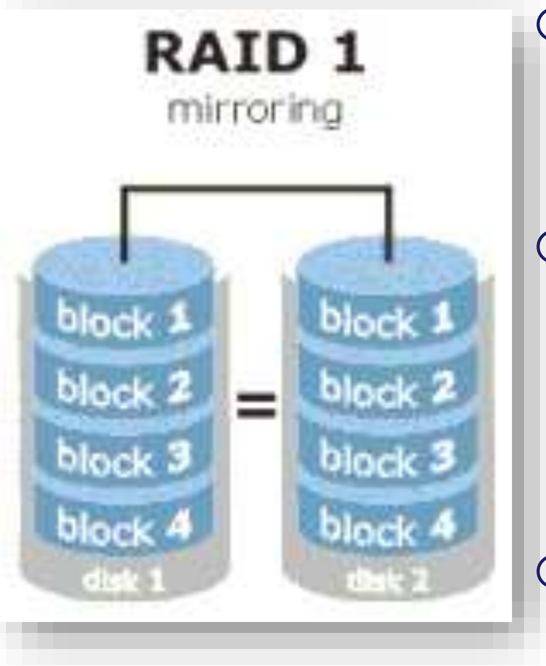
- Initially, RAID was used as a hedging solution. It allows recording data to multiple hard disks at the same time.
- Later, RAID had many variations: ensuring data security, helping to significantly increase data retrieval speed from hard disks.

1.6 RAID (Redundant Arrays of Independent Disks)



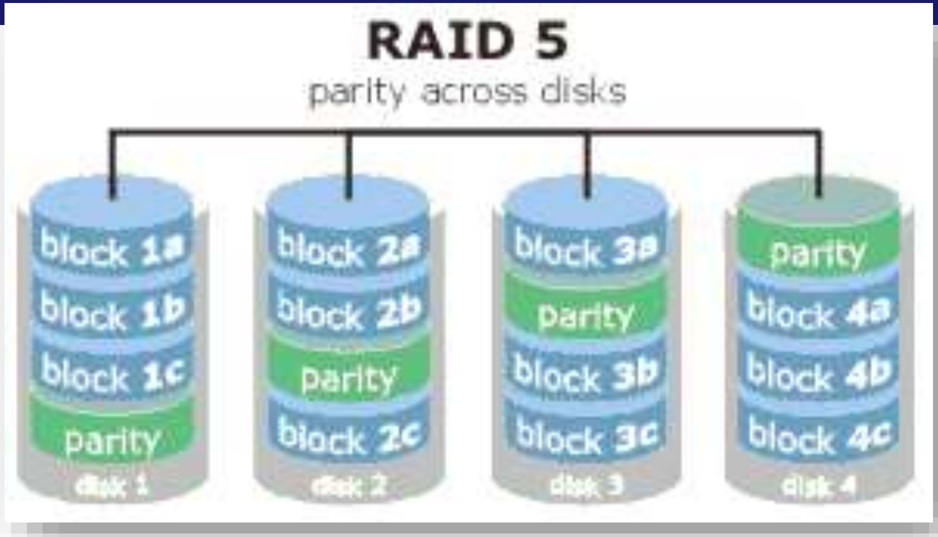
- Requires at least two hard drives.
- Data is written in parts across multiple drives (Striping)
- Advantage: increased read/write speed.
- Disadvantages: less safe, if a disk is damaged, data cannot be recovered.

1.6 RAID (Redundant Arrays of Independent Disks)



- Requires at least 2 hard disks
- Data is recorded the same on both discs (mirroring).
- Advantages: high safety, if one disk is damaged, data will still exist on the remaining disk
- Disadvantages: speed performance is not the top factor

1.6 RAID (Redundant Arrays of Independent Disks)



- Need at least 3 hard drives or more
- Data and backups are divided across all hard drives.
- Data is recorded in sections on disks (striping) (-1 disk records parity)
- RAID 5 both ensures improved speed and maintains safety.

2. Index

- **What is an index?**
  - Organizes pairs, including keys and addresses of data.
  - Address points to a disk block or record
- **Categorize the index**
  - A single level index is an ordered file index.
  - A multi-level index is an index of tree-structured data

2. Index

- **Why creating Index**
  - Increase data retrieval speed
  - Continuity across lines is not required
- **When should you not create an Index?**
  - Takes up disk space to store Index. When the user updates data on the Index column, SQL Server also updates the index
  - Managing an Index will take time and resources, so if the Index is not often used, there is no need to create it..

2. Index

- **Primary/Secondary**
  - For example: Sort by name, same name, sort by age (name is primary and age is secondary)
- **Clustered/Non clustered**
  - Clustered: the order of physical storage records is in the order of the index
  - Non clustered: the order of physical storage records is not in the order of the index
- **Dense/sparse**
  - Dense = Index all records
  - Sparse = Only type some records

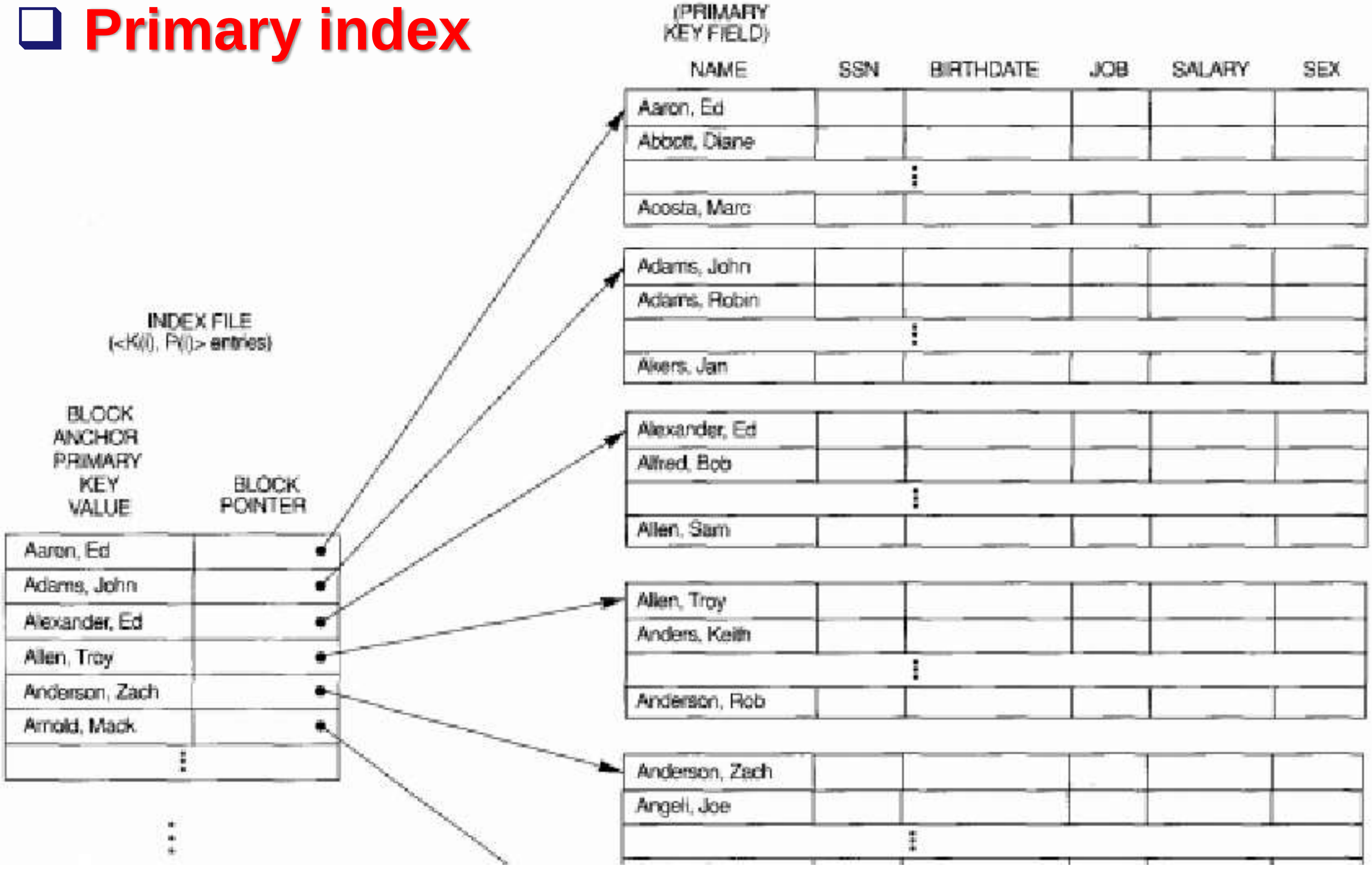
2. Index

- **Primary index**
  - The index key is the key of the ordered file, key values must be unique.
  - An index file is an ordered file consisting of two fields with fixed-length records.
  - A record in the index file, including 2 parts **[Ki, Pi]** :
    - **Ki** is the value corresponding to the value in the ordered key field
    - **Pi** contains the block address of the record with key **Ki** on the data file.

Each index entry corresponds to the first record of the block.

2. Index

❑ Primary index

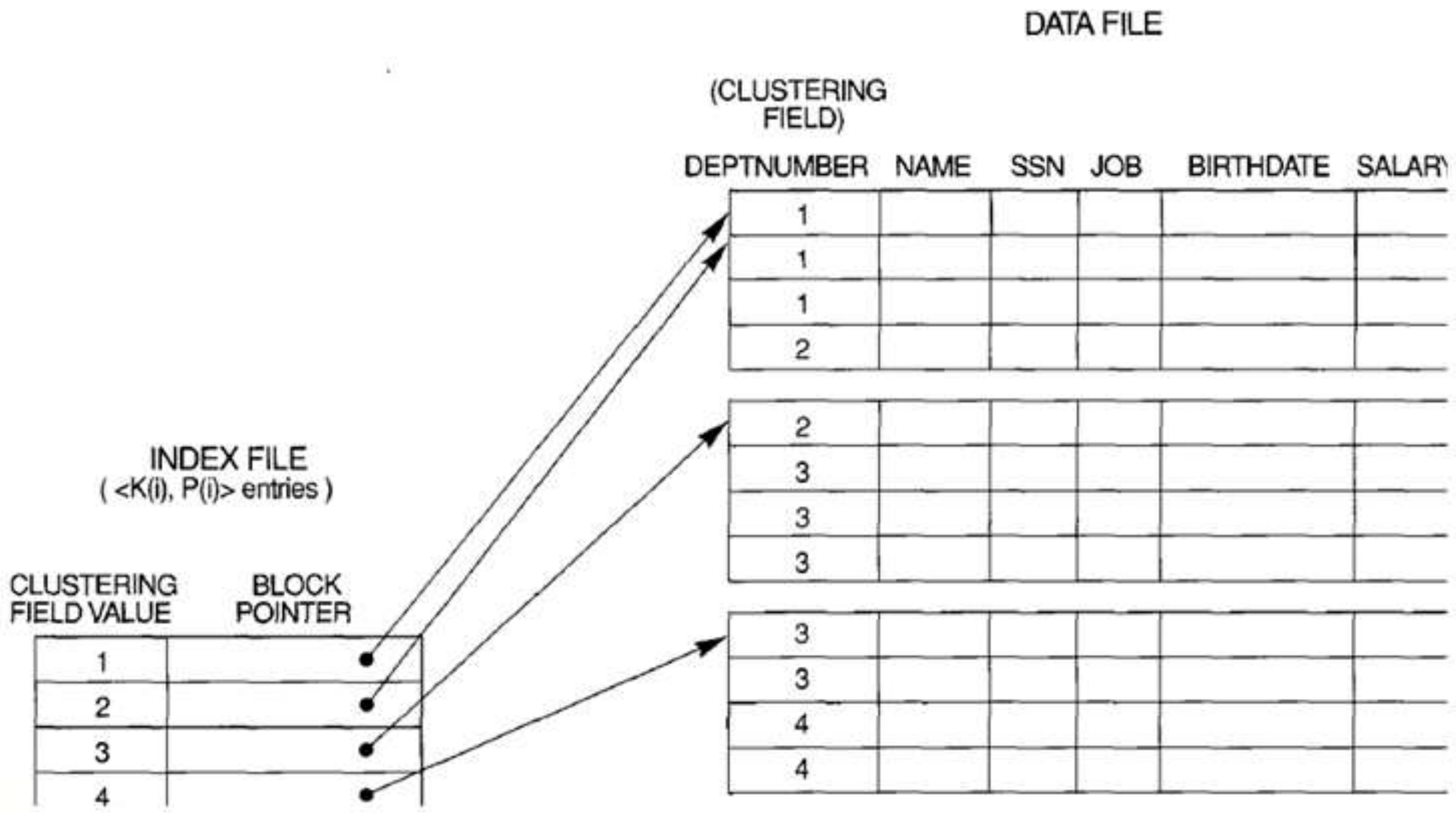


2. Index

- **Clustered**

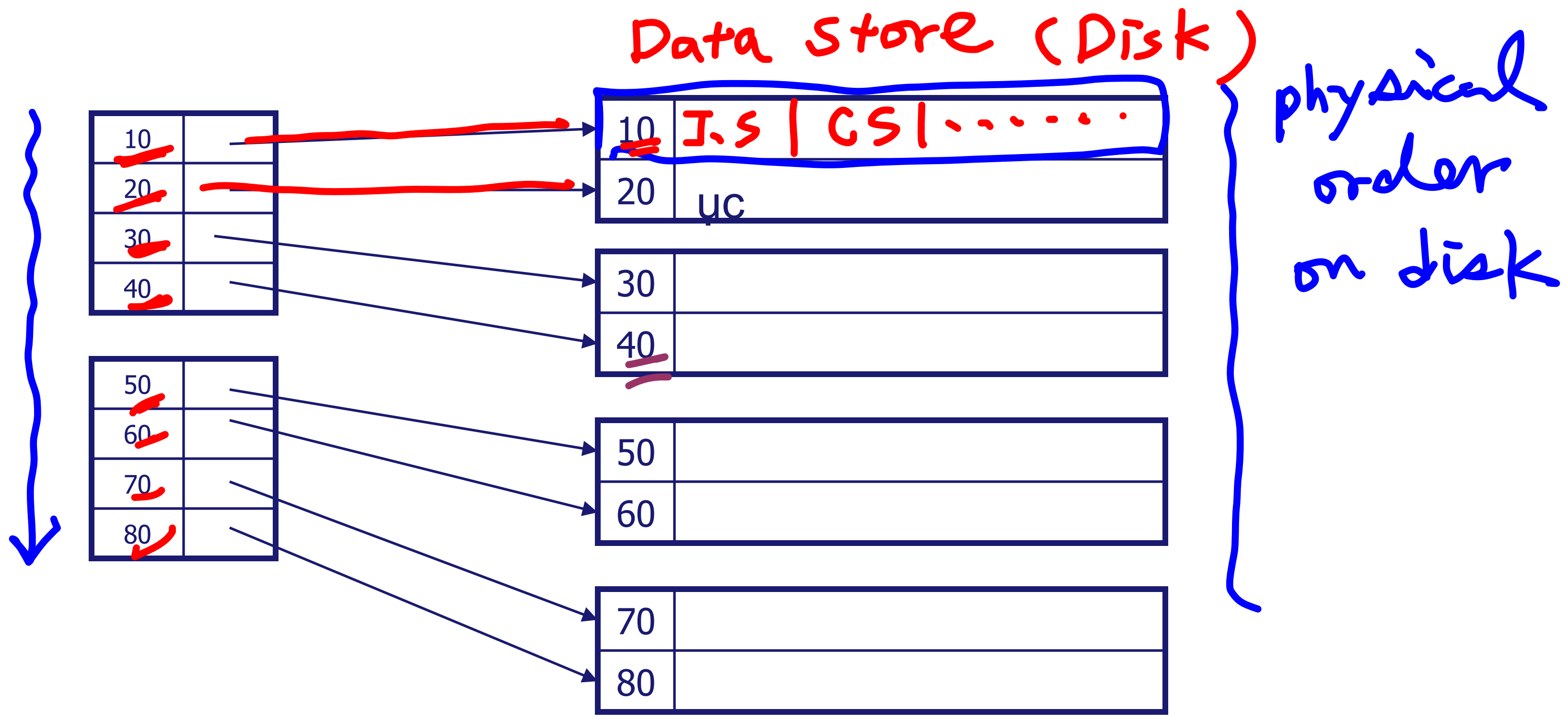
- The index key is not an ordered file key, key values may repeat.
- In Index entries:
  - The first element is the key (representative value only)
  - The second element is the address of the disk block.
- Each index entry corresponds to the first record of the block..

❑ Clustered Index



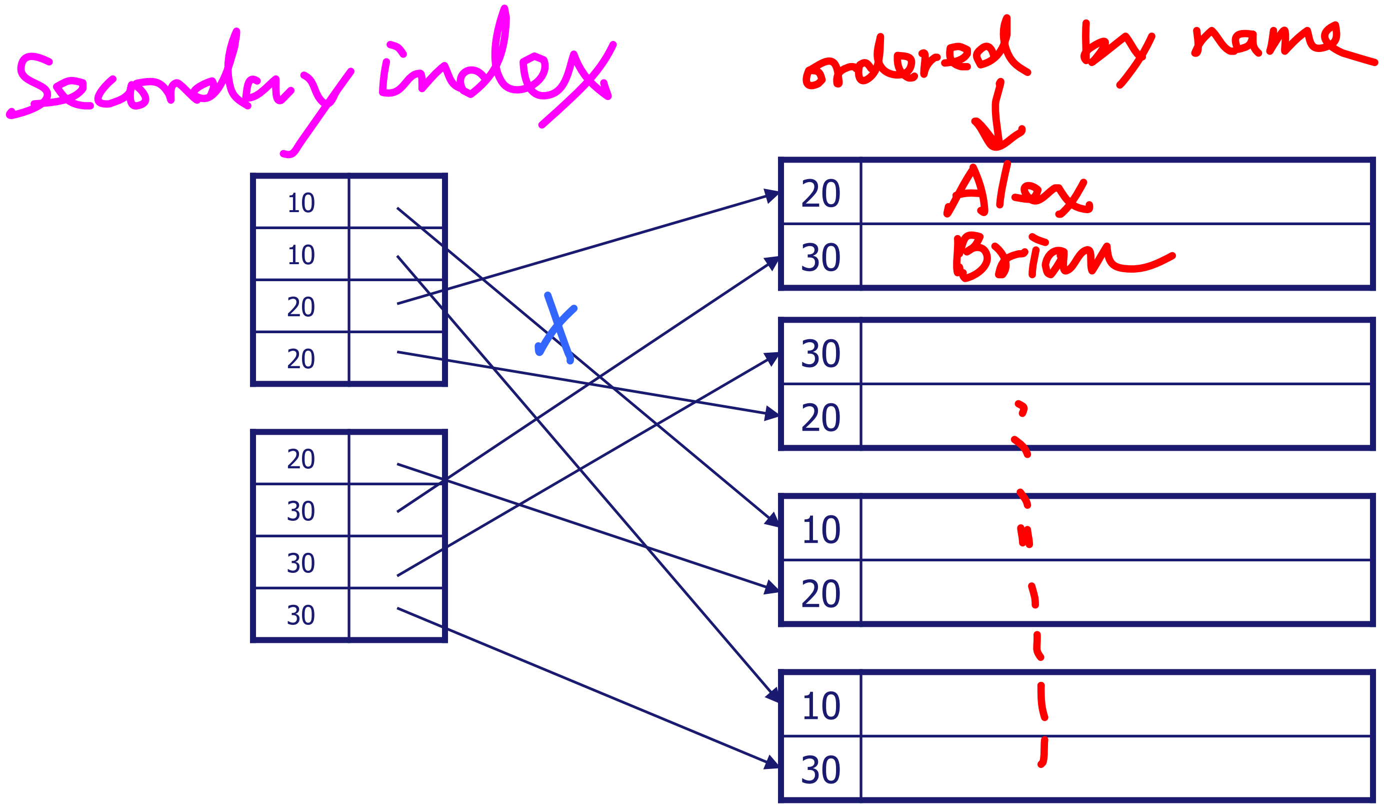
❑ Clustered Index in SQL

Used to index attributes that are primary keys

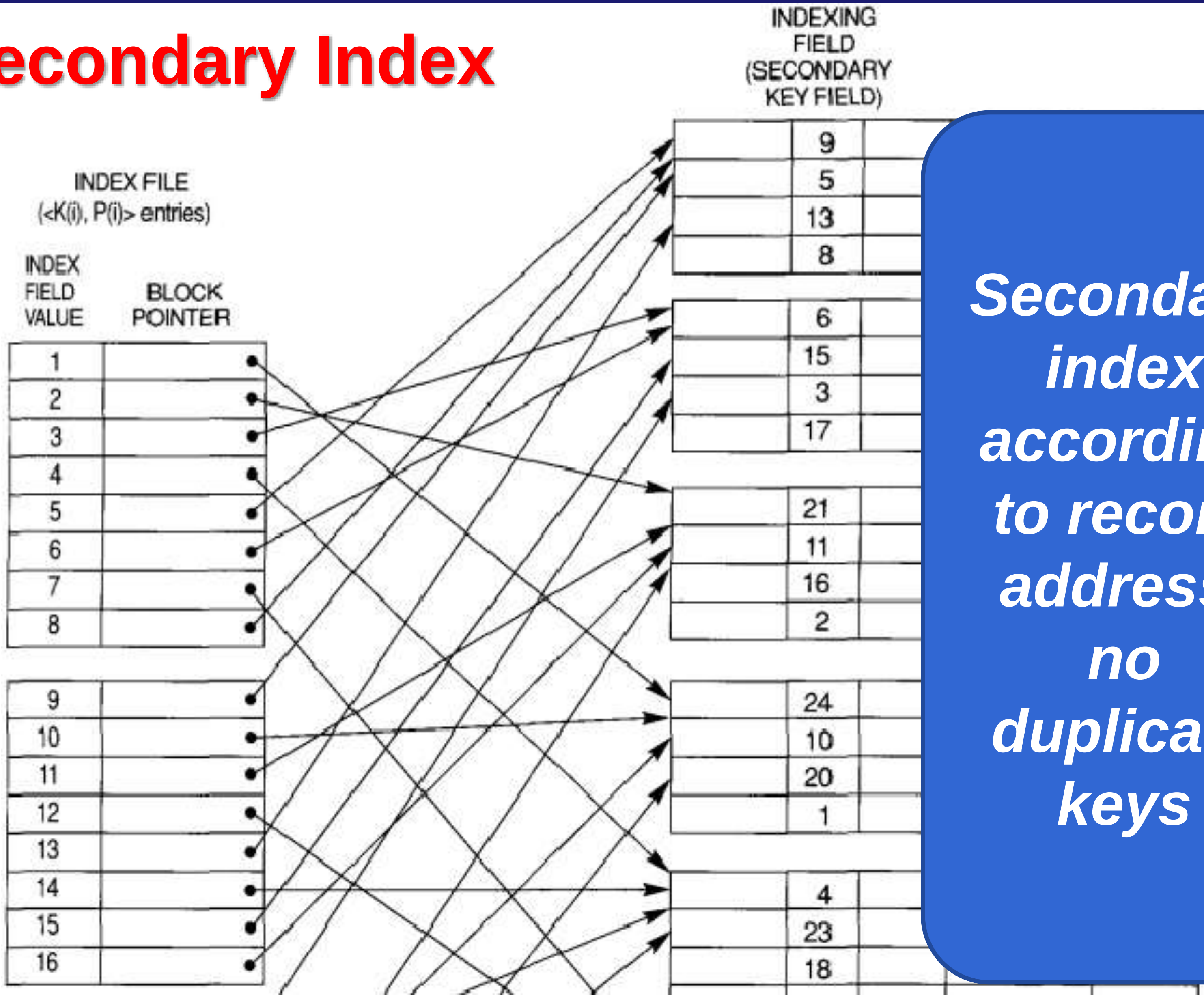


❑ Non clustered Index in SQL

Used to index attributes that are not primary keys



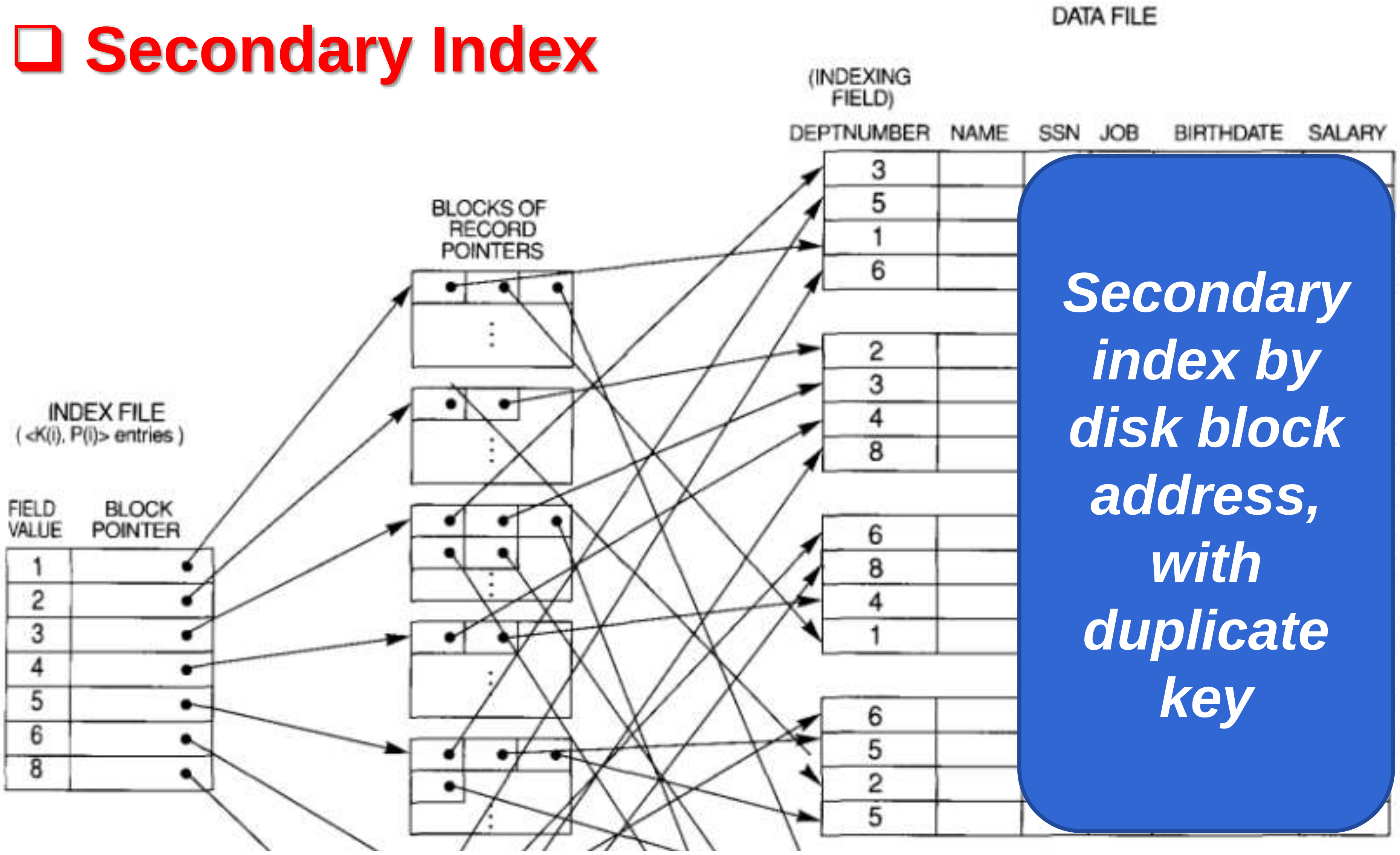
❑ Secondary Index



Secondary index according to record address, no duplicate keys

2. Index

❑ Secondary Index



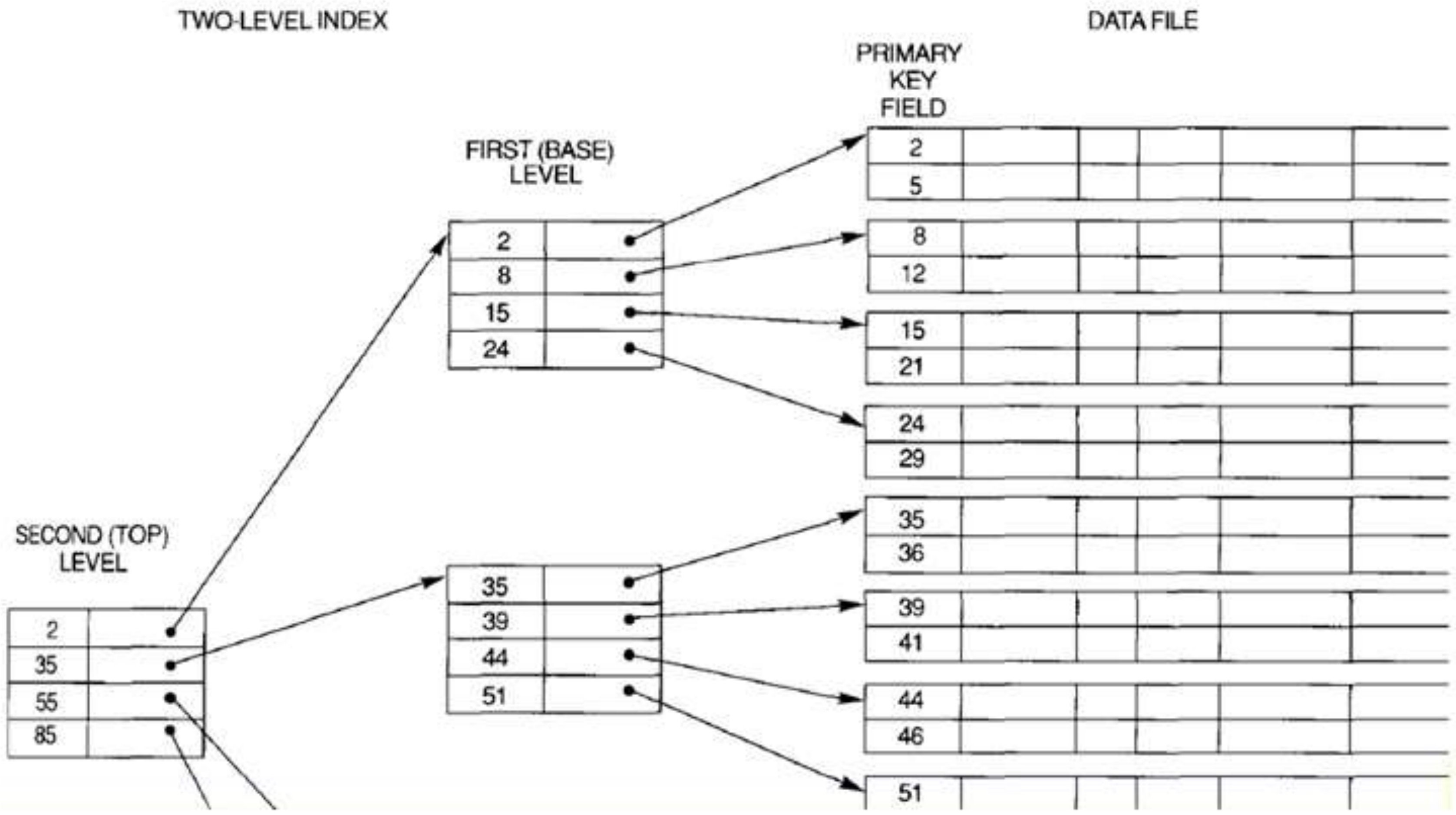
Secondary index by disk block address, with duplicate key

2. Index

- **Multi level index**
  - Multi-level organization to further improve search speed
  - First-level (base) indexes are index entries that have unique key values. This index is referred to as the index file
  - On this index file, create a primary index: a second level index
  - The second level index has index entries that are the first record of the disk block

2. Index

❑ Multi level index



Create Index in SQL Server

- SQL supports 2 types of Index:
  - Cluster Index
  - Non Cluster Index

Create Index in SQL Server

- ❑ Cluster Index: Only a single cluster index can be created for a data table.
  - By default the primary key will be the cluster index
  - The table data is arranged in the order of the cluster index

Create Index in SQL Server

- **Non Cluster Index:** can create 249 non-cluster indexes for a data table.
  - The table data is not arranged in the order of the non-cluster index.
  - Often create indexes for data columns used for joins or where conditions or the value of this column frequently changes.

Create Index in SQL Server

- Syntax:
  - CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name

    ON table_name

    (column_name[,column_name]…)

Create Index in SQL Server

- Sample create Index:
  - ***CREATE NONCLUSTERED INDEX idxExternalCandidate ON ExternalCandidate(cAgencyCode)***
  - ***CREATE CLUSTERED INDEX idxRecruitment ON RecruitmentAgencies(cAgencyCode)***

Exercises on creating Index on SQL Server

Suppose your database has the following table:SinhVien(MaSV, TenSV, TuoiSV, DiaChi)

- In which MaSV is the primary key, often used to join other tables; name (TenSV) often appears in WHERE conditions in informational queries.Requirement: Determine Cluster and non cluster index for SinhVien table. Write SQL statements to create tables and create corresponding Index statements..

Exercises on creating Index on SQL Server

SinhVien(MaSV, TenSV, TuoiSV, DiaChi)

In which MaSV is the primary key, often used to join other tables; name (TenSV) often appears in WHERE conditions in informational queries.

- CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name ON table_name (column_name[,column_name]…)
- CREATE CLUSTERED INDEX idx_MASV ON SINHVIEN(MASV)
- CREATE NONCLUSTERED INDEX idx_TENSV ON SINHVIEN(TENSV)

Index on SQL Server

- ❑Primary key → Cluster Index
- ❑Frequently accessed columns should create non clustered index → increase database access speed.

Trigger

- Trigger will take effect when we change the data on a particular data table, or processing will changes the data of command insert, update, delete.
- Triggers can contain queries from other tables or include complex SQL commands

Trigger

- Advantages of using triggers:
  - Trigger runs automatically: when there is a change in the data on the data table and Triggers can cascade when related tables are affected.
  - Triggers have less restrictive effects than value constraints (which can constrain references to columns of other data tables).

Trigger

- Syntex create new Trigger:

```
Create Trigger trigger_name
on table_name
for  [insert,update,delete]
As
Begin
    {Declare processing variables}
    {Transact-SQL}
End
```

- Syntex delete Trigger:

```
 Drop Trigger trigger_Name
```

Handling Errors in Triggers

- ❖ Message error in trigger:

Raiserror(‘Error message string’, Severity level,Error string parameter)

  - Froms 17 from 25 indicate hardware and software failures and are usually only addressed and tracked by the system administrator.
  - From 11 to 16 are created by the user and can be corrected by the user.
  - Below 10, the error is usually of a notification nature and is often caused by a mistake when typing the command.

- ❖ Do not allow data to be changed: Rollback Tran

For example using Trigger

When adding a new player, check that the angler's position on the field belongs to only one of the following positions:

Thủ môn, Tiền đạo,

Tiền vệ, Trung vệ, Hậu vệ.

For example using Trigger

```
Create trigger tg_Cauthu_vitri on Cauthu
FOR INSERT
AS
BEGIN
        declare @vitri nvarchar(20)
        select  @vitri=vitri from inserted
        if @vitri not in   (N'Thủ môn',N'Tiền đạo',N'Tiền vệ',
                            N'Trung vệ',N'Hậu vệ')
        begin
                raiserror ('Vi tri khong hop le',15,1)
                rolLback tran
                return
        end
END
```

For example using Trigger

```
Create trigger tg_Cauthu_vitri on Cauthu   FOR INSERT
AS
BEGIN
          declare @vitri nvarchar(20)
          select   @vitri=vitri from inserted
          if @vitri not in (N'Thủ môn',N'Tiền đạo',N'Tiền vệ', N'Trung
vệ',N'Hậu vệ')
          begin
                    raiserror ('Vi tri khong hop le',15,1)
                    rolLback tran
                    return
          end
END
```

```
Select * from Cauthu
```

```
Insert into Cauthu values (N'Nguyễn văn Tèo em', N'Hậu vệ',
'2/14/2000’, Null, 'BBD', 'VN', 21)
```

For example using Trigger

When adding new players, check that the shirt numbers of players belonging to the same club must be different.

For example using Trigger

```
Create trigger tg_Cauthu_soao on Cauthu
FOR INSERT
AS
BEGIN
        declare @so int, @maclb varchar(5)
        select @so=so,  @maclb=maclb   from Inserted
        if ((select count(*) from cauthu
            where maclb=@maclb and so=@so) >1)
        begin
                raiserror (N'Số áo bị trùng',15,1)
                rolLback tran
                return
        end
END
```

For example using Trigger

```
Create trigger tg_Cauthu_soao on Cauthu for INSERT
AS
BEGIN
        declare @so int, @maclb varchar(5)
        select @so=so,  @maclb=maclb   from Inserted
        if ((select count(*) from cauthu       where maclb=@maclb and so=@so) >1)
        begin
                raiserror (N'Số áo bị trùng',15,1)
                rolLback tran
                return
        end
END
select * from cauthu

insert into Cauthu  values(N'Nguyễn văn Tèo em', N'Hậu vệ',
'2/14/2000', Null, 'BBD', 'VN', 21)
```

Inserted and Deleted Table

- When adding data to a table, the added data will be placed in the Inserted clipboard
- When deleting data from the table, the deleted data will be placed in the Deleted clipboard
- The update operation is handled:
  - Delete old data rows (old data will be put into the Deleted table)
  - Add a new data row (new data will be put into the Inserted table)
- To get the newly updated data into the table, we access the Deleted or Inserted clipboard

Do homework!

Prepare:

1. Group 4 member (Leader get requirement for team member).

2. Collect database sample:

- QLBongDa: write script commands create database table and insert value (file sql).

- Mỗi phần truy vấn tự chọn 3 câu để viết SQL.
- Khi nộp bài sẽ chạy thử trên máy của nhóm.

Discussion