Pune Vidyarthi Griha's

College of Engineering, Nashik

206, Behind Reliance Petrol Pump, Dindoriroad, Mhasrul, Nashik-422004
Phone: 0253-6480000 / 36 /44 Toll Free Number: 18002665330
Website: https://pvgcoenashik.org Email: admission@pvgcoenashik.org

Approved by AICTE, New Delhi, DTE, Mumbai and
Affiliated to Savitribai Phule Pune University, Pune.

DTE Code: EN5330

“ File Structure & Orgnization ”

By,

Prof. Anand N. Gharu

Asst. Professor

Computer Department,

PVG COE, Nashik.

Syllabus of Unit-I

File Structure and Organization

1.1 Introduction

1.2 Logical and Physical Files

1.2.1 File

1.2.2 File Structure

1.2.3 Logical and Physical Files Definitions

1.3 Basic File Operations

1.3.1 Opening Files

1.3.2 Closing Files

1.3.3 Reading and Writing

1.3.4 Seeking

Syllabus of Unit-I

- 1.4 File Organization
- 1.4.1 Field and Record structure in file
- 1.4.2 Record Types
- 1.4.3 Types of file organization
- 1.4.3.1 Sequential
- 1.4.3.2 Indexed
- 1.4.3.3 Hashed
- 1.5 Indexing
- 1.5.1 What is an Index?
- 1.5.2 When to use Indexes?
- 1.5.3 Types of Index
- 1.5.3.1 Dense Index
- 1.5.3.2 Sparse Index

Introduction

- Atabse : https://youtu.be/d11viALaCvA
- Intro : https://youtu.be/fSWAkJz_huQ
- https://youtu.be/r8u8JuM0450?list=RDCMUC6E97LDJTFJgzWU7G3CHILw
- https://youtu.be/FR4QIeZaPeM
- https://youtu.be/fFi_0HVgLrQ

Introduction

Data : Data is a collection of facts, such as numbers, words, measurements, observations or just descriptions of things.

Database : database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS)

Management : Management allows a person to organize, store and retrieve data from a computer.

System : system is a set of rules, an arrangement of things, or a group of related things that work toward a common goal.

Files : A file is a collection of data stored in one unit, identified by a filename. It can be a document, picture, audio or video stream, data library, application, or other collection of data.

Introduction

File Structure : File Structures is the Organization of Data in Secondary Storage Device in such a way that minimize the access time and the storage space.

Files Organization : File organization refers to the way records are physically arranged on a storage device.

Fields : A database field is a single piece of information from a record.

Record : A record is composed of fields and contains all the data about one particular person, company, or item in a database.

Physical Database : Physical database design translates the logical data model into a set of SQL statements that define the database.

Logical Database: logical database must be able to access and identify all files within the storage system to operate correctly.

Logical & Physical Files

Files in Database

Files :

“ A file is a collection of data stored in one unit, identified by a filename.”

It can be a document, picture, audio or video stream, data library, application, or other collection of data.”

- A field is a single piece of information
- a record is one complete set of fields
- a file is a collection of records.

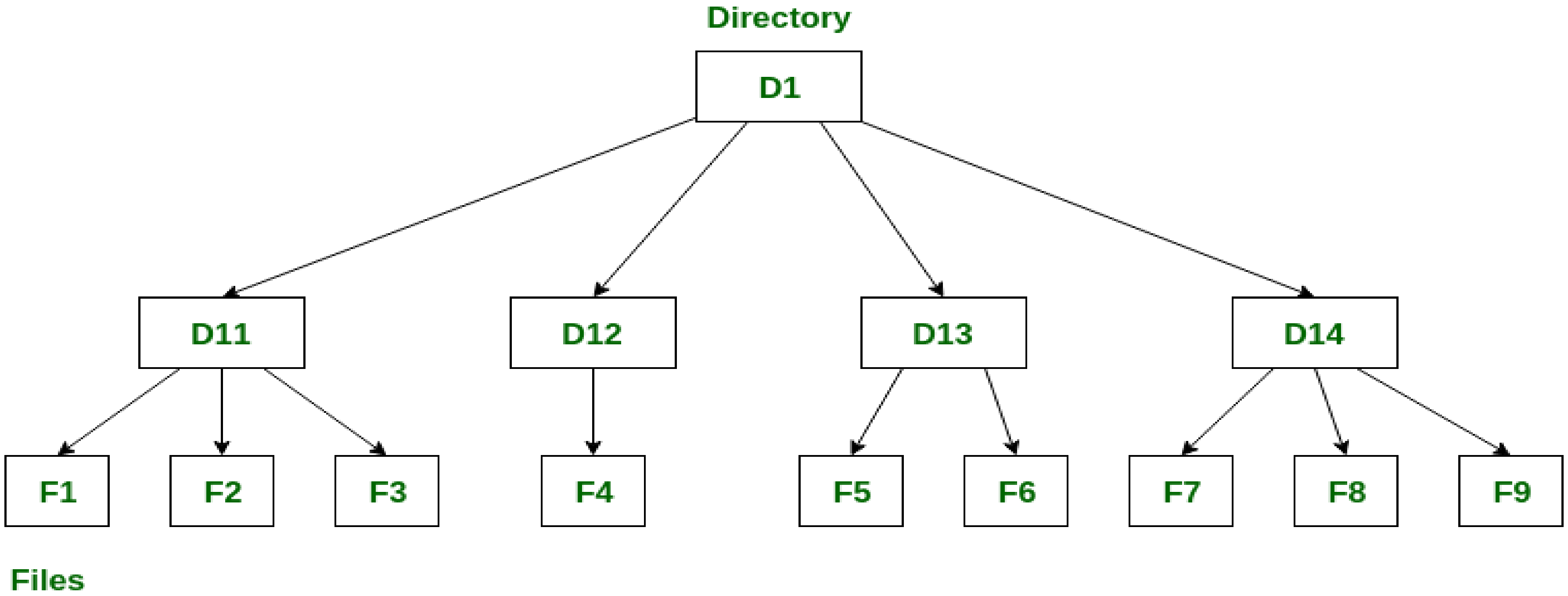
For example, a telephone book is analogous to a file. This is a collection of programs that enables you to enter, organize, and select data in a database.

File Structure in Database

File Structure :

“ File Structures is the Organization of Data in Secondary Storage Device in such a way that minimize the access time and the storage space.”

- A File Structure is a combination of representations for data in files and of operations for accessing the data.
- A File Structure allows applications to read, write and modify data.



Logical File and physical files

There are two types of files :

1. Physical files :

- Physical files contain the actual data that is stored on the system, and a description of how data is to be presented to or received from a program.
- They contain only one record format, and one or more members.
- Records in database files can be externally or program-described.
- Files can be viewed as logical or physical files
- Physical files is a files, viewed in term of how the data is stored on storage device and how the processing are made possible.

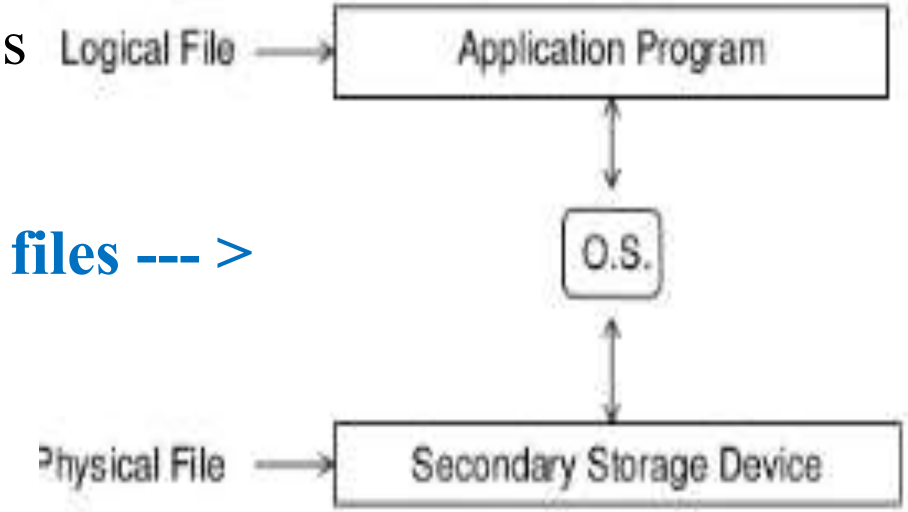
Logical File and physical files

2. Logical files :

“Logical files do not contain data. They contain a description of records that are found in one or more physical files”

- A logical file is a view or representation of one or more physical files.
- Logical files that contain more than one format are referred to as multi-format logical files.
- Logical files is a files, which viewed in term of how the data items contains its record and whats processing operation may be performed on the files
- .

Example of logical and physical files --- >



Basic File Operation

File Operations

There are various operation of files :

1. Open file operation
2. Close file operation
3. Read file operation
4. Write file operation
5. Seeking file operation

File Operations

1. **Open –** A file can be opened in one of the two modes, read mode or write mode. In read mode, the operating system does not allow anyone to alter data. In other words, data is read only. Files opened in read mode can be shared among several entities. Write mode allows data modification. Files opened in write mode can be read but cannot be shared.

2. **Close –** This is the most important operation from the operating system's point of view. When a request to close a file is generated, the operating system

- removes all the locks (if in shared mode),
- saves the data (if altered) to the secondary storage media, and
- releases all the buffers and file handlers associated with the file.

File Operations

3. Read – By default, when files are opened in read mode, the file pointer points to the beginning of the file. There are options where the user can tell the operating system where to locate the file pointer at the time of opening a file. The very next data to the file pointer is read.

4. Write – User can select to open a file in write mode, which enables them to edit its contents. It can be deletion, insertion, or modification. The file pointer can be located at the time of opening or can be dynamically changed if the operating system allows to do so.

File Operations

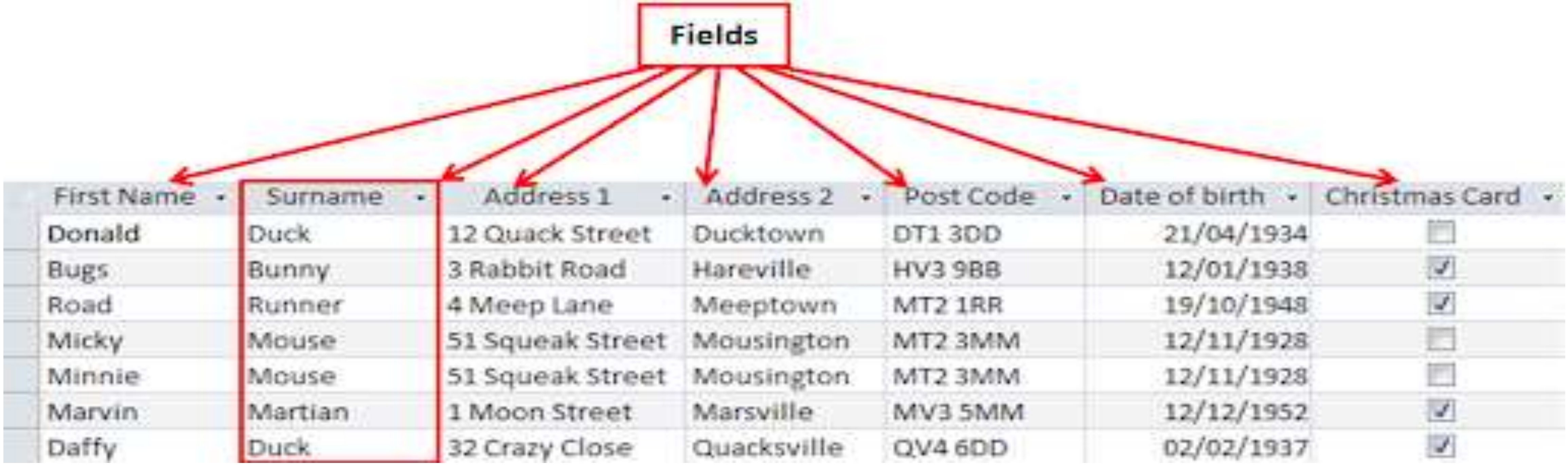
3. Locate(seeking) −

Every file has a file pointer, which tells the current position where the data is to be read or written. This pointer can be adjusted accordingly. Using find (seek) operation, it can be moved forward or backward.

File Organisation

Fields and Record structure in File

1. **Fields –** In computer science, data that has several parts, known as a record, can be divided into fields. Relational databases arrange data as sets of database records, so called rows. Each record consists of several fields; the fields of all records form the columns. Examples of fields: name, gender, Date of Birth.

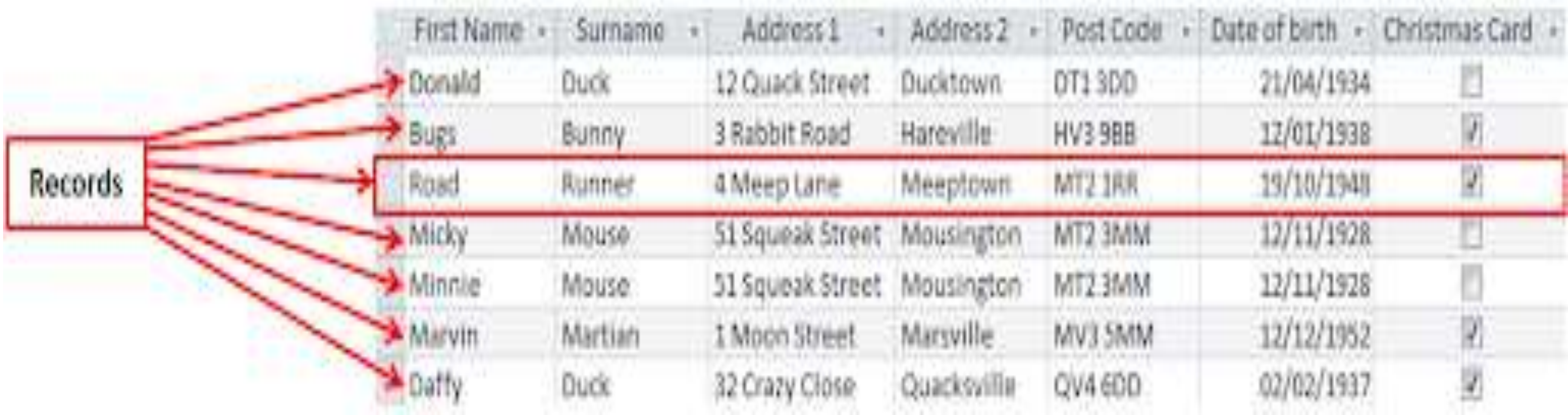


First Name	Surname	Address 1	Address 2	Post Code	Date of birth	Christmas Card
Donald	Duck	12 Quack Street	Ducktown	DT1 3DD	21/04/1934	☐
Bugs	Bunny	3 Rabbit Road	Hareville	HV3 9BB	12/01/1938	☑
Road	Runner	4 Meep Lane	Meeptown	MT2 1RR	19/10/1948	☑
Micky	Mouse	51 Squeak Street	Mousington	MT2 3MM	12/11/1928	☐
Minnie	Mouse	51 Squeak Street	Mousington	MT2 3MM	12/11/1928	☐
Marvin	Martian	1 Moon Street	Marsville	MV3 5MM	12/12/1952	☑
Daffy	Duck	32 Crazy Close	Quacksville	QV4 6DD	02/02/1937	☑

Fields and Record structure in File

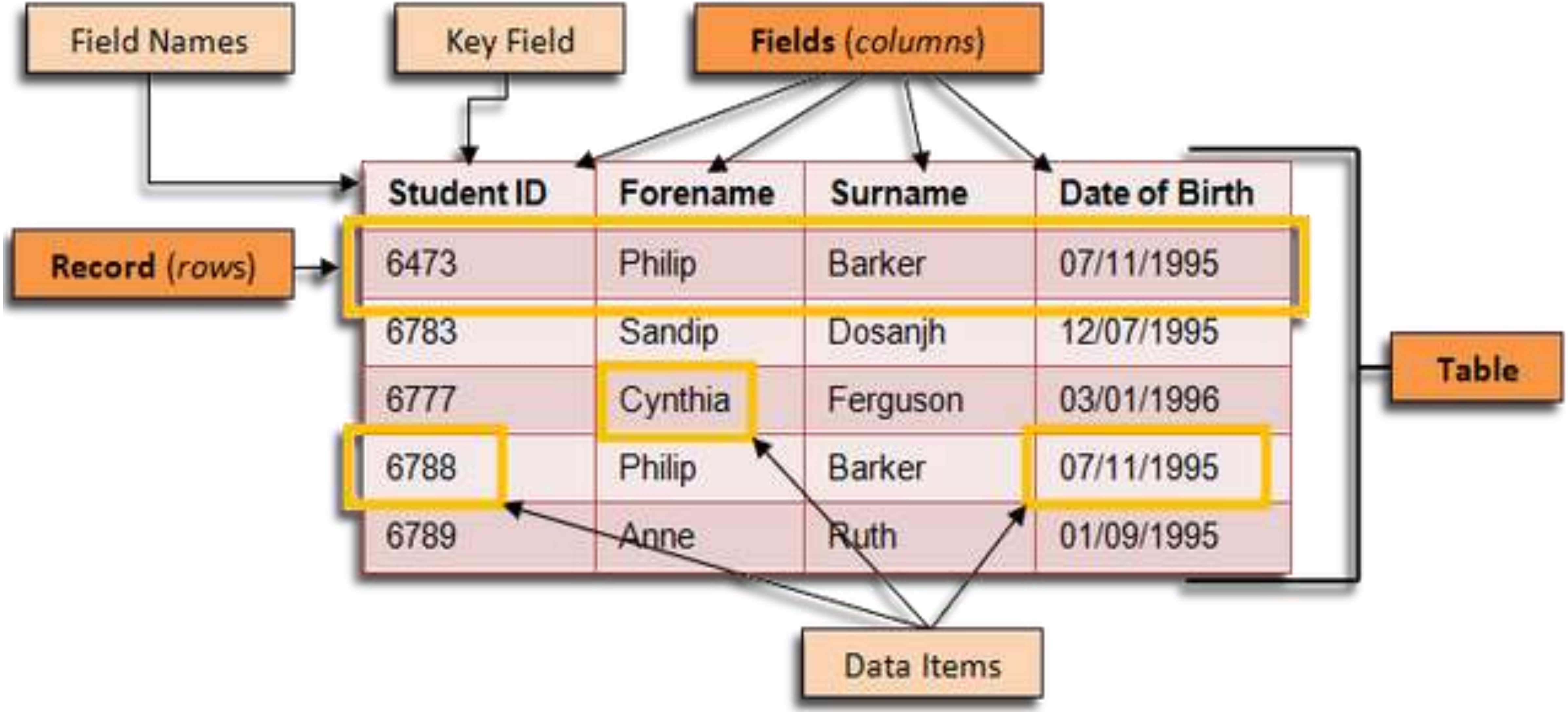
2. Record – In a database, a record (**sometimes called a row**) is a group of fields within a table that are relevant to a specific entity.

For example, in a table called customer contact information, a row would likely contain fields such as: ID number, name, street address, city, telephone number



First Name	Surname	Address 1	Address 2	Post Code	Date of birth	Christmas Card
Donald	Duck	12 Quack Street	Ducktown	DT1 3DD	21/04/1934	□
Bugs	Bunny	3 Rabbit Road	Hareville	HV3 9BB	12/01/1938	☑
Road	Runner	4 Meep Lane	Meeptown	MT2 1RR	19/10/1948	☑
Micky	Mouse	51 Squeak Street	Mousington	MT2 3MM	12/11/1928	□
Minnie	Mouse	51 Squeak Street	Mousington	MT2 3MM	12/11/1928	□
Marvin	Martian	1 Moon Street	Marsville	MV3 5MM	12/12/1952	☑
Daffy	Duck	32 Crazy Close	Quacksville	QV4 6DD	02/02/1937	☑

Fields and Record structure in File



Student ID	Forename	Surname	Date of Birth
6473	Philip	Barker	07/11/1995
6783	Sandip	Dosanjh	12/07/1995
6777	Cynthia	Ferguson	03/01/1996
6788	Philip	Barker	07/11/1995
6789	Anne	Ruth	01/09/1995

Types of File Organisation

Types File Organization

1. **Sequential File Organization**

2. **Indexed File Organization**

3. **Hashed File Organization**

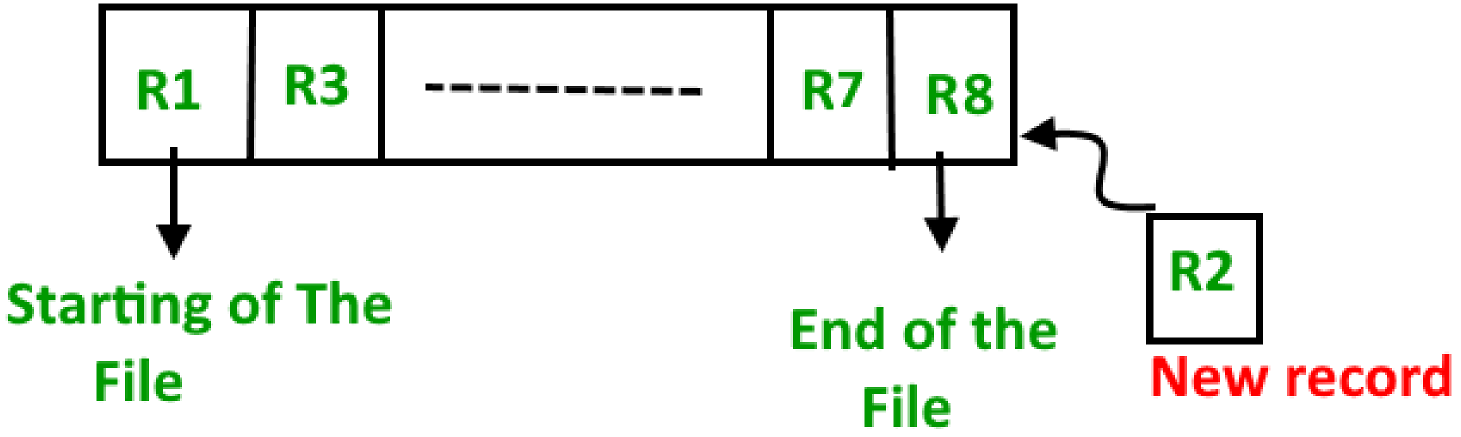
4. **Heap File Organization**

5. **B+ Tree File Organization**

6. **Clustered File Organization.**

Sequential File Organization

1. A sequential file is designed for efficient processing of records in sorted order on some search key.
2. Records are chained together by pointers to permit fast retrieval in search key order.
3. Pointer points to next record in order.
4. Records are stored physically in search key order
5. This minimizes number of block accesses.



Sequential File Organization

Advantages of Sequential File:

1. File design is simple.
2. Location of records requires only the record key.
3. Low-cost file media such as magnetic tapes can be used for storing data.
4. Retrieval of records become efficient if the query uses the sorting attribute as the search key.
5. Sorting of records on the ordering field is fast. [No sorting is required externally]

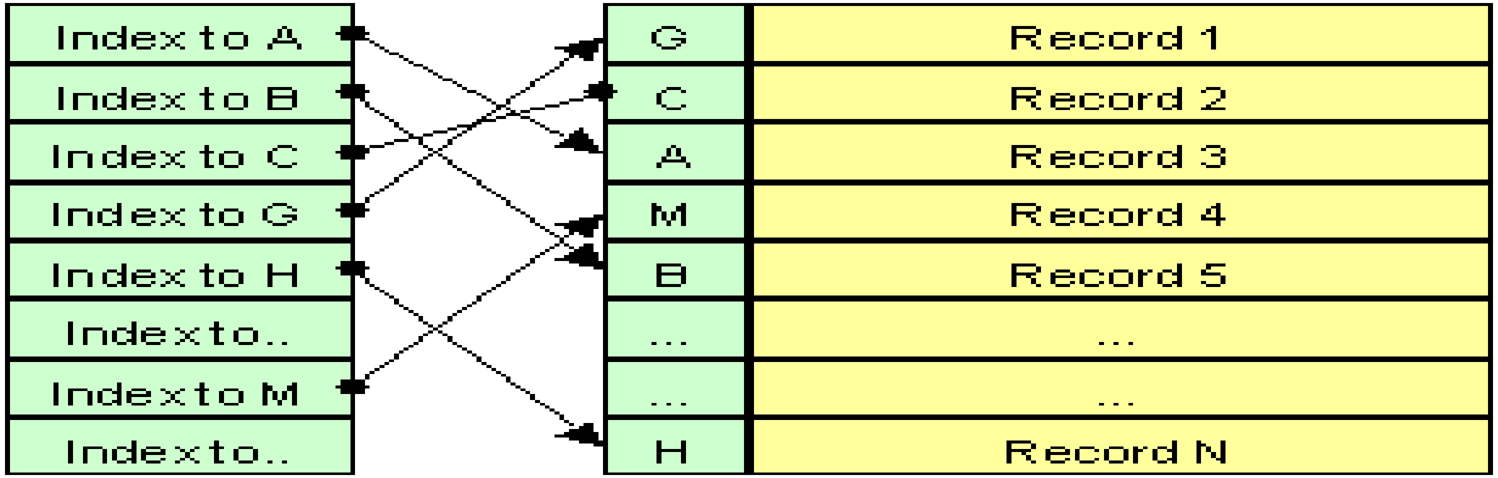
Sequential File Organization

Limitations/Draw-backs of sequential file:

1. Updating requires that all transaction records are stored in the record key sequence.
2. Insertion and deletion of records are expensive.
3. Updating the sorting attribute values of the records is also expensive.
4. Retrieval of records on the non-ordering attributes is not easy.
5. Searching is also difficult.

Indexed File Organization

1. An index is a data structure that organizes data records on disk to optimize certain file operation.
2. An index allows us to efficiently search or retrieve all records.
3. Using an index , we can achieve a fast search of data records.
4. An indexed file contains records ordered by a record key. A record key uniquely identifies a record and determines the sequence in which it is accessed with respect to other records.



Index to A
Index to B
Index to C
Index to G
Index to H
Index to..
Index to M
Index to..

G	Record 1
C	Record 2
A	Record 3
M	Record 4
B	Record 5
...	...
...	...
H	Record N

Indexed File Organisation

Indexed File Organization

Advantages of Indexed sequential access file organization

1. In indexed sequential access file, sequential file and random file access is possible.
2. It accesses the records very fast if the index table is properly organized.
3. Primary & Secondary index can be used to search the data
4. It provides quick access for sequential and direct processing.
5. Data maintained centrally and it kept up to date.

Indexed File Organization

Disadvantages of Indexed sequential access file organization

1. Indexed sequential access file requires unique keys and periodic reorganization.
2. If index value becomes high, then searching become slow
3. It requires more storage space.
4. It is expensive because it requires special software.
5. Backup should be taken regularly.
6. File is updated directly

Hashed File Organization

1. **Hashing :**

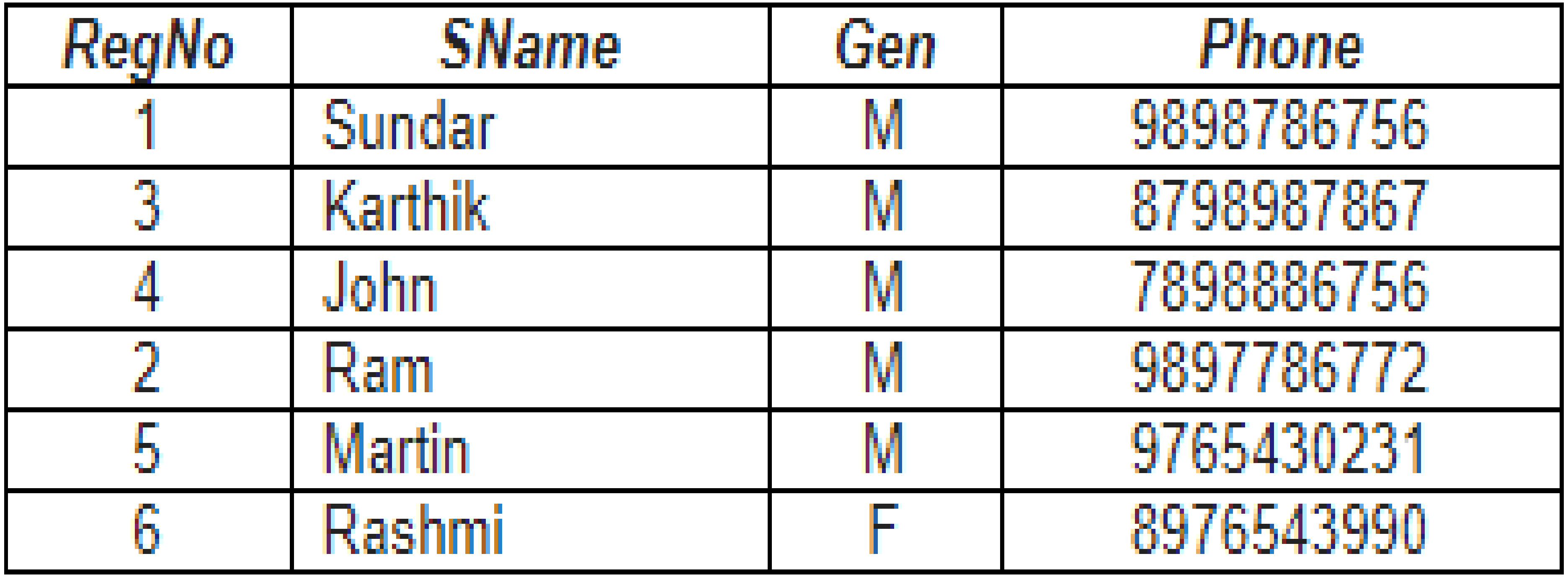
- **Hashing** is the process of converting a given key into another value.
- **A hash function** is used to generate the new value according to a mathematical algorithm.
- The result of a hash function is known as **a hash value**
- "It is a file organization technique where a hash function is used to compute the address of a record. It uses the value of an attribute or set of attributes as input and gives the location (page/block/bucket) where the record can be stored."
- **For example, let us consider the following table Student;**

Hashed File Organization

A hash function is a function which maps the large set of values into smaller set of files/locations/values. Let us organize the above table using the ***phone*** attribute value as input for the hash function.

Hash function :

h(phone mod 10)



RegNo	SName	Gen	Phone
1	Sundar	M	9898786756
3	Karthik	M	8798987867
4	John	M	7898886756
2	Ram	M	9897786772
5	Martin	M	9765430231
6	Rashmi	F	8976543990

Hashed File Organization

In the above hash function, phone is the phone attribute's value of each record. 10 is the number of buckets/pages where we want to store our table. [10 buckets means bucket0, bucket1, …, bucket9].

For our example,

For 1st record, h(9898786756 mod 10) = 6 ie., the first record has to be stored in 6th bucket.

For 2nd record, h(8798987867 mod 10) = 7 ie., the second record has be stored in 7th bucket.

…

For 4th record, h(7898886756 mod 10) = 6 ie., the fourth record has be stored in 6th bucket [like 1st]

For 5th record, h(9765430231 mod 10) = 1 ie., the 5th record has to be stored in 1st bucket.

For last record, h(8976543990 mod 10) = 0 ie., the last record has to be stored in 0th bucket.

Hashed File Organization

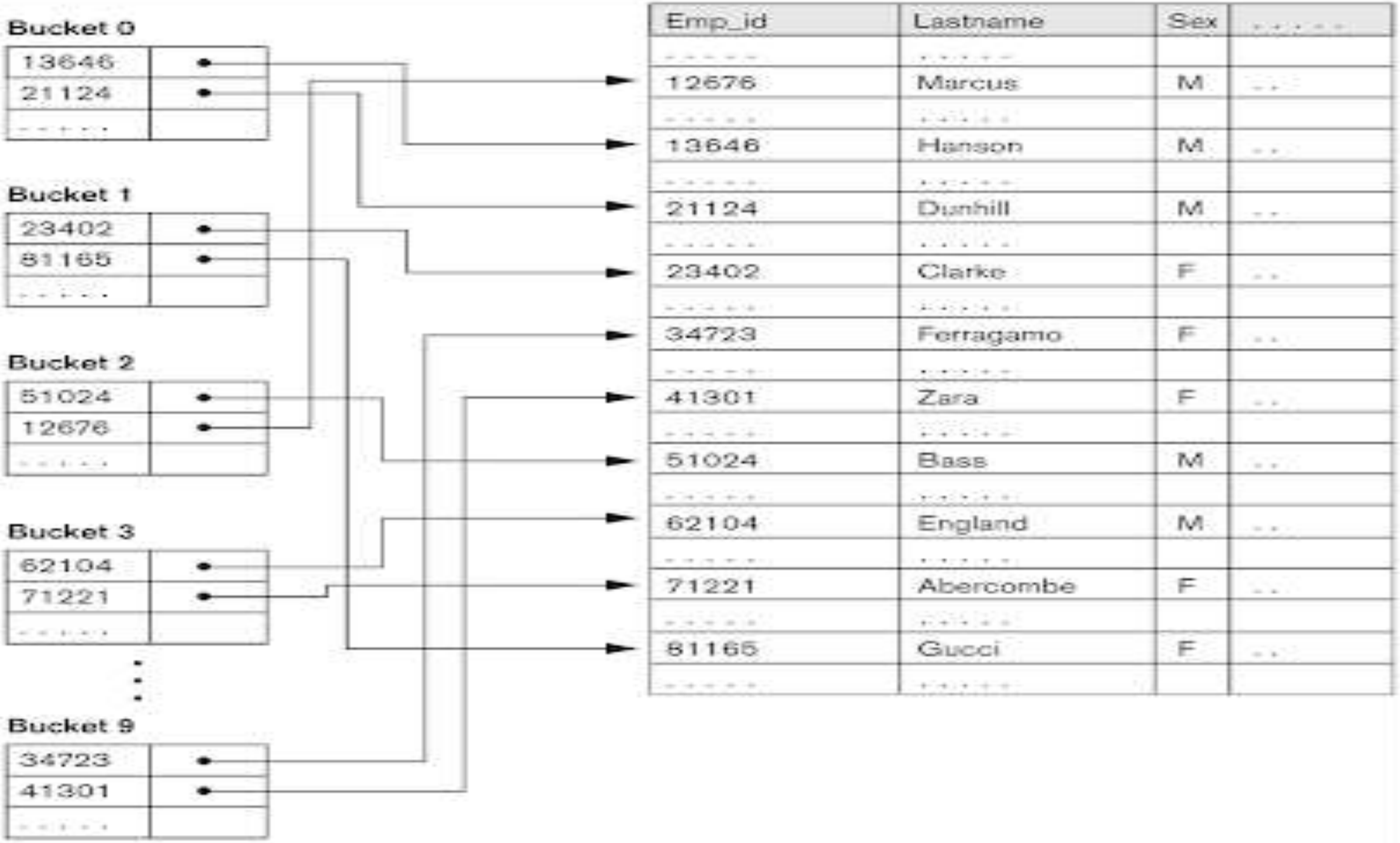
Advantages

1. accessing any record is very faster. Similarly updating/deleting a record is also very quick.
2. Easy to insert, delete, or update a record.
3. It is suitable for online transaction systems like online banking, ticket booking system etc.

Disadvantages

1. Records are randomly stored in scattered locations. May waste a lot of space in small files
2. If we are searching for range of data, then this method is not suitable.
3. Very difficult to sort the data files.
4. If there is a search on some columns which is not a hash column, then the search will not be efficient

Hashed File Organization



Emp_id	Lastname	Sex	......
......	......		
12676	Marcus	M	..
......	......		
13646	Hanson	M	..
......	......		
21124	Dunhill	M	..
......	......		
23402	Clarke	F	..
......	......		
34723	Ferragamo	F	..
......	......		
41301	Zara	F	..
......	......		
51024	Bass	M	..
......	......		
62104	England	M	..
......	......		
71221	Abercombe	F	..
......	......		
81165	Gucci	F	..
......	......		

Figure 18.15
Hash-based indexing.

Indexing

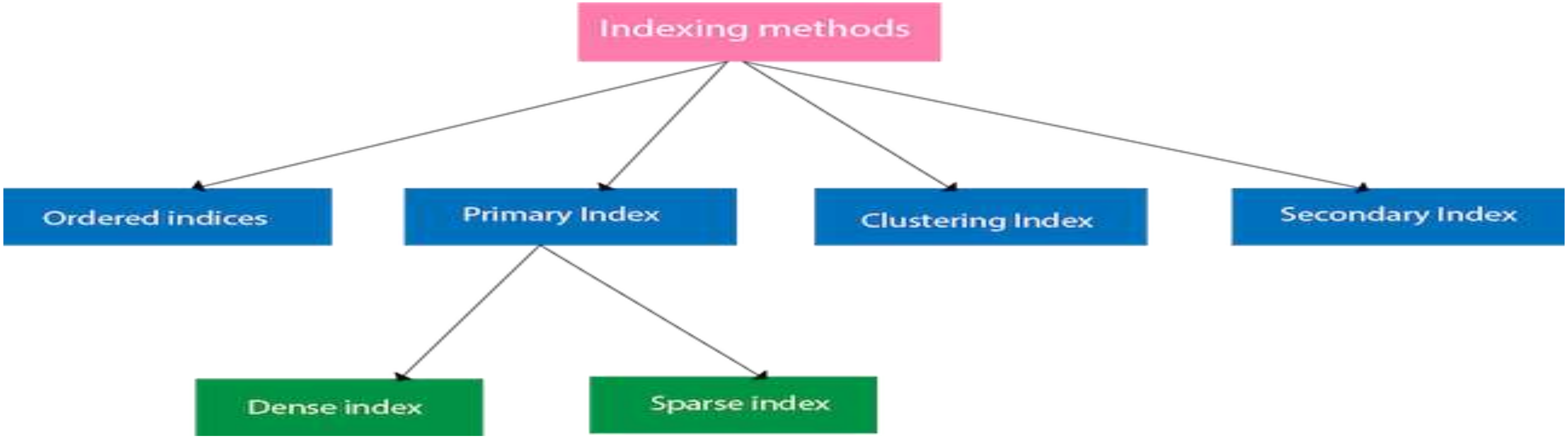
Indexing

Indexing is a data structure technique to efficiently retrieve records from the database files based on some attributes on which the indexing has been done.

Indexing in database systems is similar to what we see in books.

Indexing is defined based on its indexing attributes.

Indexes are used to improve performance of database system . E.g. catalogue of library



Indexing

Indexing is defined based on its indexing attributes. Indexing can be of the following types −

Primary Index − Primary index is defined on an ordered data file. The data file is ordered on a key field. The key field is generally the **primary key** of the relation.

Secondary Index − Secondary index may be generated from a field which is a **candidate key** and has a unique value in every record.

Clustering Index −

1. Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.
2. If the record file is physically ordered on non-key field which does not have different value for each records that field is called clustering field
3. Clustering is used to speed up retrieval of records that have the same value on clustering fields.

Indexing

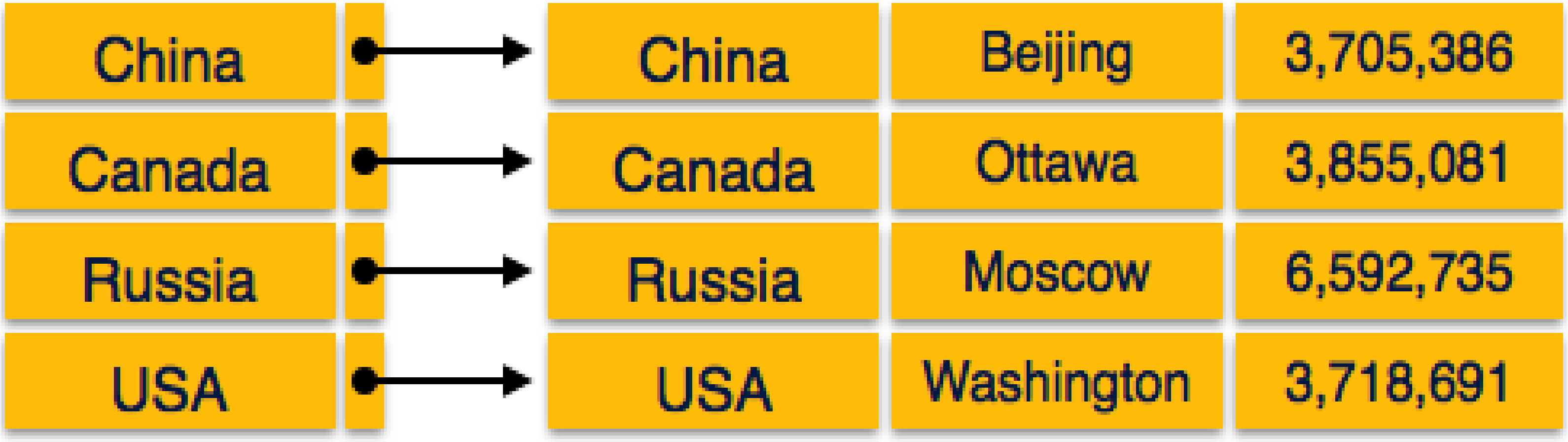
Ordered Indexing is of two types −

Dense Index

Sparse Index

Dense Index :

In dense index, there is **an index record for every search key value in the database.** This makes searching faster but requires more space to store index records itself. Index records contain search key value and a pointer to the actual record on the disk.

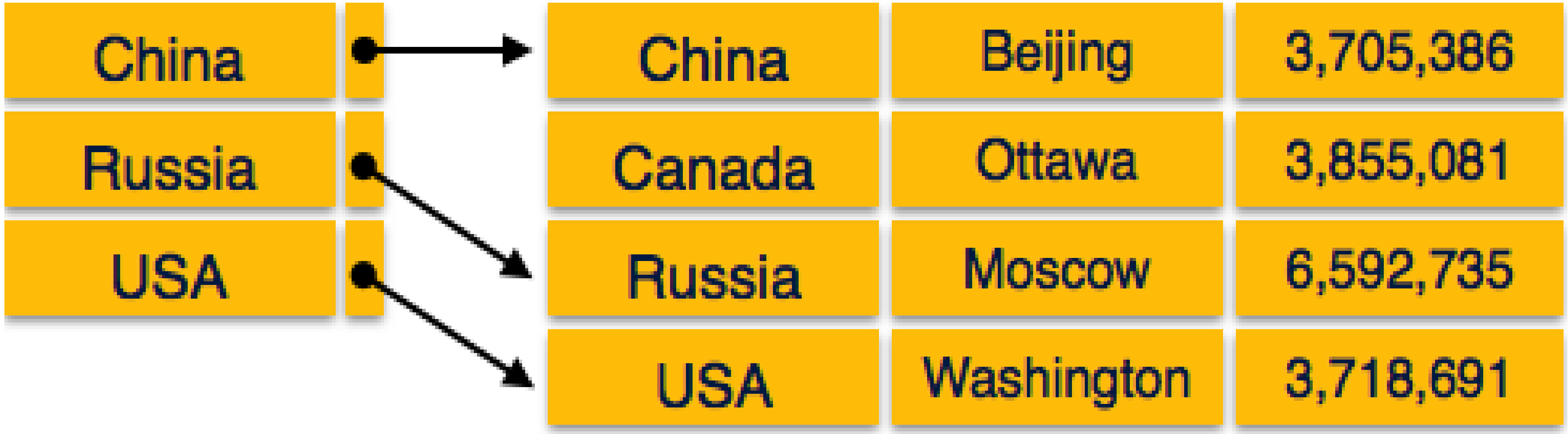


China	China	Beijing	3,705,386
Canada	Canada	Ottawa	3,855,081
Russia	Russia	Moscow	6,592,735
USA	USA	Washington	3,718,691

Sparse Index

Sparse Index :

In sparse index, **index records are not created for every search key**. An index record here contains a search key and an actual pointer to the data on the disk. To search a record, we first proceed by index record and reach at the actual location of the data. If the data we are looking for is not where we directly reach by following the index, then the system starts sequential search until the desired data is found.



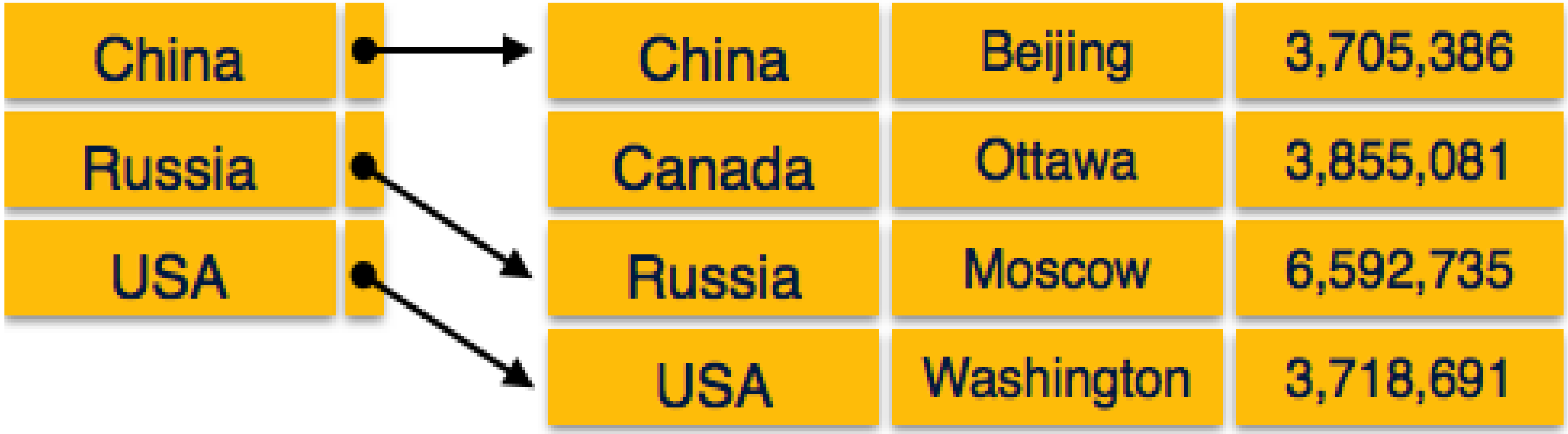
China
Russia
USA

China	Beijing	3,705,386
Canada	Ottawa	3,855,081
Russia	Moscow	6,592,735
USA	Washington	3,718,691

Multilevel Index

Multilevel Index :

Index records comprise search-key values and data pointers. **Multilevel index is stored on the disk along with the actual database files.** As the size of the database grows, so does the size of the indices. There is an immense need to keep the index records in the main memory so as to speed up the search operations. If single-level index is used, then a large size index cannot be kept in memory which leads to multiple disk accesses.



China	●
Russia	●
USA	●

China	Beijing	3,705,386
Canada	Ottawa	3,855,081
Russia	Moscow	6,592,735
USA	Washington	3,718,691

Primary Index Vs Secondary Index

SR. NO	PRIMARY INDEX	SECONDARY INDEX
1.	Primary Index is mandatory	Secondary Index is Optional
2.	There can be one primary index	There can be more than one Secondary index
3.	There can be no duplication in primary index	There can be dulicate entries in Secondary Index
4.	Can be Only one primary key for a table	Maximum 32 Secondary for a table
5.	Record can be stored and retrieved	Record can be retrieved only
6.	Primary Index is efficient	Secondary Index is Inefficient
7.	Primary Index contains uniques Primary Key	Secondary Index contains Candidate key

DENSE INDEX VS SPARSE INDEX

SR. NO	DENSE INDEX	SPARSE INDEX
1.	An index record appearsfor every search key value in file	An index record are only created for some records
2.	Indices are faster	Indices are slower
3.	Indices require more space	Indices require less space
4.	More maintainance for insertion and deletion	Less maintainance for insertion and deletion
5.	A Dense index in database is sequential file with pair of key & pointers for every record in data files	A sparse index in database is sequential file with index pointer pointing to the block of sorted data file
6.	The index record contain the search key and pointer	Each index contains value and pointer.

THANK YOU!!!

My Blog : https://anandgharu.wordpress.com/

Email : gharu.anand@gmail.com