GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY COLLEGE
(Autonomous)

Reaccredited by NAAC with A+

DATABASE MANAGEMENT SYSTEM

By

Mr N Kartik

Assistant Professor, Dept of Computer Applications,

PRESIDENCY COLLEGE, BANGALORE-24

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

PRESIDENCY COLLEGE
(AUTONOMOUS)
AFFILIATED TO BENGALURU CITY UNIVERSITY, APPROVED BY AICTE, DELHI & RECOGNISED BY THE GOVT. OF KARNATAKA
RE-ACCREDITED BY NAAC WITH 'A+' GRADE

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY COLLEGE
(Autonomous)

UNIT 2

Data Modeling Using Entity-Relationship Model:

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

PRESIDENCY COLLEGE
(AUTONOMOUS)
AFFILIATED TO BENGALURU CITY UNIVERSITY, APPROVED BY AICTE, DELHI & RECOGNISED BY THE GOVT. OF KARNATAKA
RE-ACCREDITED BY NAAC WITH 'A+' GRADE
GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Syllabus

- Data Modeling
- Using Entity-Relationship Model
- Using High-Level Conceptual Data Models for Database Design
- An Example Database Application
- Entity Types, Entity Sets, Attributes and Keys
- Relationship Types, Relationship Sets
- Roles and Structural Constraints
- Weak Entity Types
- Refining the ER Design Company Database Diagrams
- Naming Conventions and Design. Issues
- File organization and indexing
- Type of single level ordered index
- Multi-level indexes, indexes on multiple keys

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Entity-Relationship[ER]model

- ER-Model is used to represent objects in the real world and of relationship among these objects, which represents the overall logical structure of a database
- ER-model is a high-level conceptual data model developed by **chen** in 1976 to facilitate database design

High level Conceptual data models for database design

The database design consists of the following steps

1. ***Requirement collection & analysis***
2. ***Conceptual design***
3. ***Logical design (Data model mapping)***
4. ***Physical design***

1.Requirement collection & analysis

- ***During this step , database designers interview prospective database users to understand & document their data requirement***

2.Conceptual design

- **In this step a conceptual schema for the database , using a high-level conceptual data model is created. Hence, this step is known as " Conceptual Design"**
- **This conceptual schema explains description of the entity types , relationship & constraints**

3. Logical design (Data model mapping)

- It involves the actual implementation of database, using a commercial database
- In this step, there exists transformation between high-level data model to implement data model. Hence, called " Data model mapping"

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College (Autonomous)

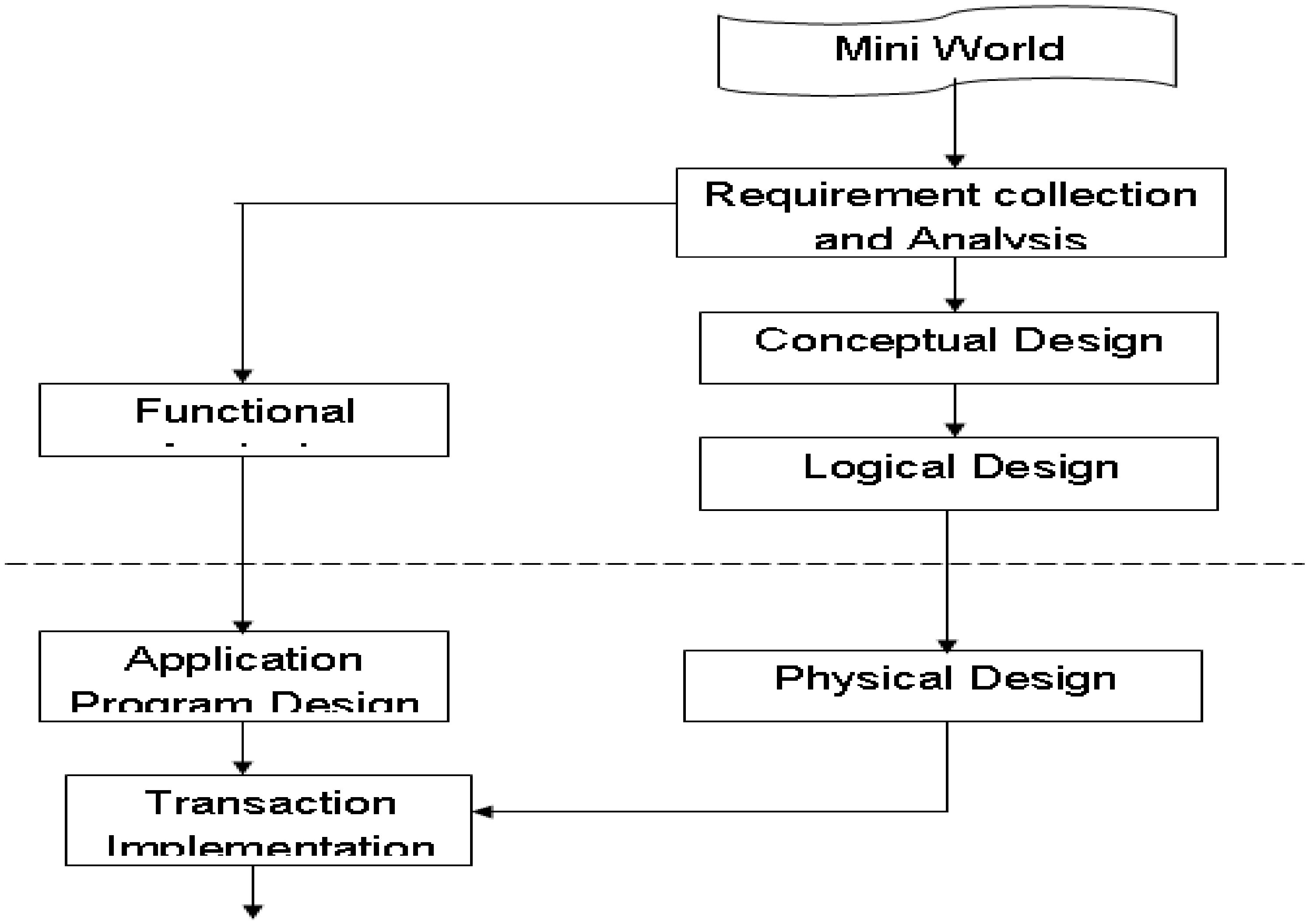
Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

4.Physical design

During this step, the internal storage structures, indexes, access paths and file organization for the database files are specified.


Mini World
Requirement collection and Analysis
Conceptual Design
Logical Design
Functional
Physical Design
Application Program Design
Transaction Implementation

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

1. Entity
2. Attributes
3. Entity sets
4. Entity types
5. Key attributes of an entity type
6. Value sets (Domain) of attributes

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College (Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Examples of entity and attributes

- 1.An **entity** is a thing in the real world.
- **2.Attributes :**

An **attribute** is a property that describes an entity.

Entity	Attributes	Values
Car	Color	Red
	Make	Volkswagen
	Model	Bora
	Year	2014

- An **entity type** defines a collection of entities that have the same attributes.

Entity Type Example:

***Entity Type*:**

Student

***Entity Attributes*:**

StudentID,

Name,

Surname,

Date of Birth,

Department

Entity set:

- The collection of all entities of a particular entity type in the database at any point in time is called an **entity set.**

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Key attributes of an entity type: A key attribute is a minimal set of attributes of an entity set, which uniquely identifies an entity in an entity set.

Example:

For the company entity, name can be the key attribute because no two companies can have the same name.

For the student entity, regno can be the key attribute.

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Different Types of attributes.

1. **Simple or atomic attributes**
2. **Composite attributes**
3. **Single valued attributes**
4. **Multi valued attributes**
5. **Derived attributes**
6. **Stored attributes**
7. **Null attributes**
8. **Key attributes**

1.Simple or atomic attributes:

These attributes can't be subdivided into further.

Example: Regno, ID, age, Deptno

2.Composite attributes: Composite attributes can be divided into smaller subparts.

Address

Street No. Area City State Pin code

3.Single valued attributes: An attribute that can take only one value at a time is called single valued attributes.

Example: The age attribute will have a single value.

4.**Multi valued attributes:** Some attribute have more than one value for the same attribute and are called multi valued attributes.

Example: A **college degree** attribute can have multiple values.

Degree [BCA,B,Sc, MCA, PhD]

Admin [Bangalore, Mumbai]

Carcolor [Red, Black].

5.Derived attributes: If the value of an attribute can be derived from some other attributes, then such attributes are called derived attributes.

Example, gross pay of an employee this can be derived by knowing basic pay, allowances, and deduction for employee.

6.**Stored attributes:** The value of certain attributes cannot be obtained or derived from some other attributes, that is, they are not derived from any other attributes.

Example: Birth date, Book_ID

7.**Null attributes:** A null attributes used when an attributes does not have any value.

- Email: All employees in an employee database may not have e-mail address.

- **8. Key attributes:** An entity type usually has an individual whose values are distinct for each individual entity. Such an attribute is called a key attribute.

An Example – Company Database

The company consists of several departments and every department has a manager. It is also required to record when he has become the manger.

Several employees' works for a department.

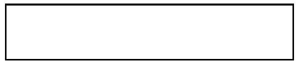
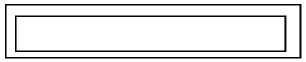
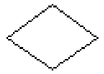
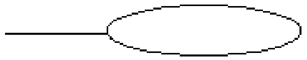
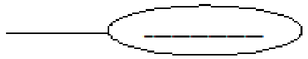
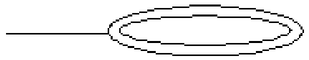
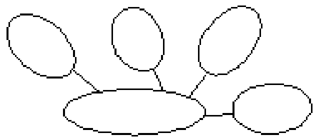
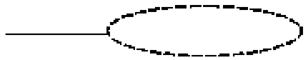
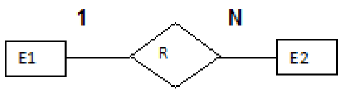
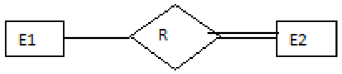
A department have several locations.

Each department controls several projects.

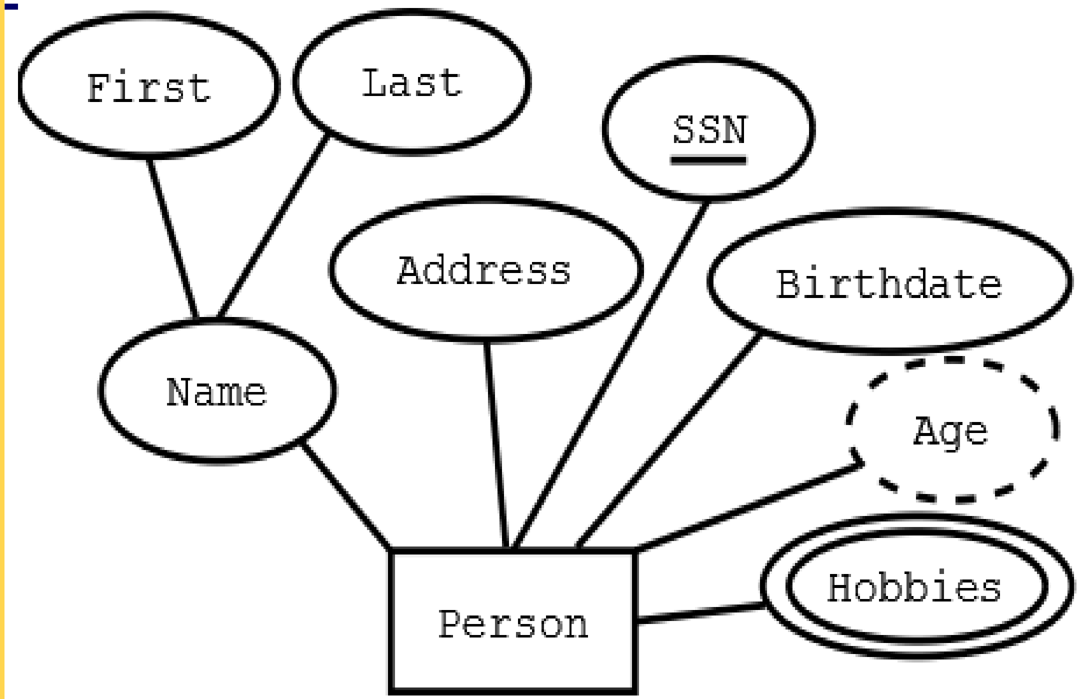
Employee details like Social Security Number (SSN), name, the department he works for etc are to be stored.

An employee works for only one department, but he/she can work on more than one project.

We need to know the employee dependent details for some specific purpose like PF, Insurance and etc.

ER Symbols:	Meaning
	Entity
	Weak Entity
	Relation ships
	Attributes
	Key attribute
	Multi valued attribute
	Composite attributes
	Derived attribute
	Cardinality ratio 1:N E1, E2 in R
	Total participation E2 in R

Graphical Representation in E-R diagram



Rectangle -- Entity

Ellipses -- Attribute (underlined attributes are [part of] the primary key)

Double ellipses -- multi-valued attribute

Dashed ellipses-- derived attribute, e.g. age is derivable from birthdate and current date.

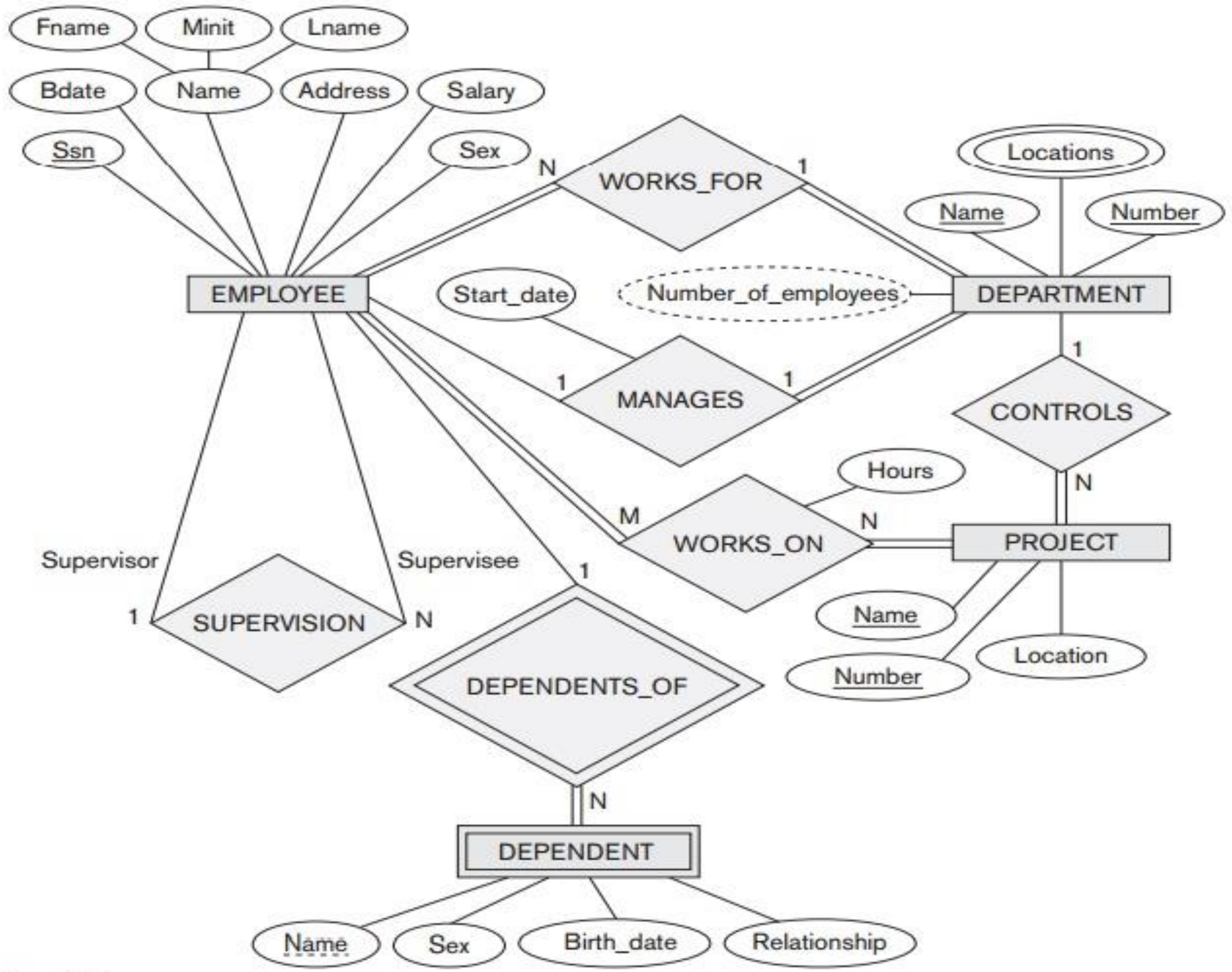


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

- **<u>What is Relationship?</u>**

Definition:

A relationship is an association among two or more entities. A relationship captures how two or more entities are related to one another.

- **1.Relationship Set:** A collection of relationships of same type is called the relationship set.

Degree of Relationship:

The degree of relationship is the number of entities participating in a relation.

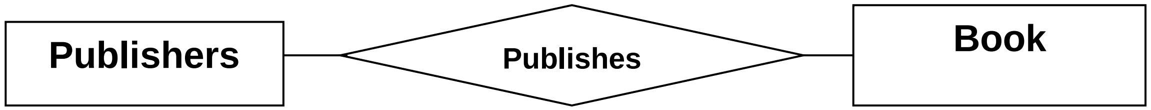
Relationship types

***1.Binary relationship : A relationship of degree 2 is known
as Binary relationship***

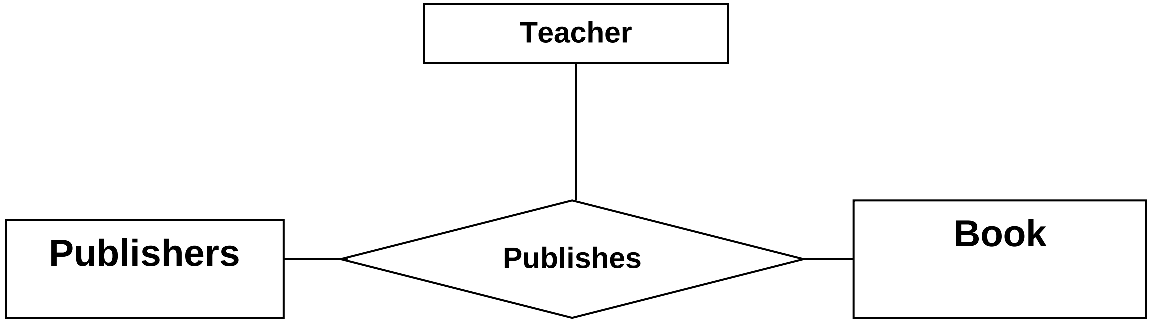
2.Ternary relationship : A relationship of degree 3 is

known as Ternary relationship

1. **Binary relationship:** A relationship of degree 2 is called binary relationship. Example, the figure shows an example of binary relationships.



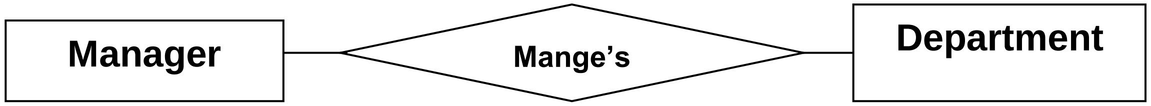
2.Ternary relationship: Relations of degree three are called ternary relationship.



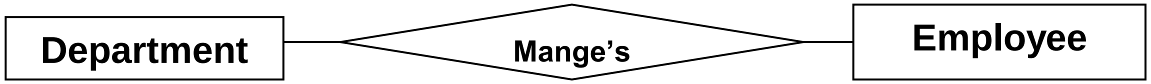
Different Types of relationship or Explain cardinality in DBMS

1. One to One (1:1)
2. One to Many (1 : M)
3. Many to One (M:1)
4. Many to Many (M : M)

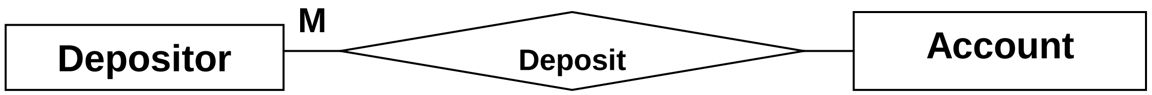
<u>1.One to One relationship (1:1) relationships:</u> An entity in A is associated with at most one entity in B and vice versa.



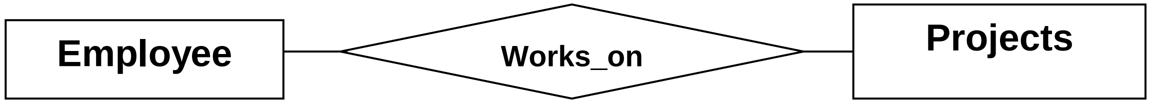
2.One to Many relationship (1:M) relationships: An entity in A is associated with any number in B, an entity in B however can be associated with at most one entity in A.



3. **Many to One relationship (M:1) relationships:** An entity in A is associated with at most one entity in B. An entity in B however it can be associated with any number of entities in A. many Depositors deposit to single account.



4. **Many to Many relationship (M:N) relationships:** An entity in A is associated with any number of entities in B and, an entity in B is associated with any number of entities in A.



GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College (Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Relationship type vs. relationship set (1)

- Relationship Type:
  - Is the schema description of a relationship
  - Identifies the relationship name and the participating entity types
  - Also identifies certain relationship constraints
- Relationship Set:
  - The current set of relationship instances represented in the database
  - The current *state* of a relationship type

Relationships of Higher Degree

- Relationship types of degree 2 are called binary
- Relationship types of degree 3 are called ternary and of degree n are called n-ary
- In general, an n-ary relationship is not equivalent to n binary relationships
- Constraints are harder to specify for higher-degree relationships ($n > 2$) than for binary relationships

Discussion of n-ary relationships (n > 2)

- If a particular binary relationship can be derived from a higher-degree relationship at all times, then it is redundant
- For example, the TAUGHT_DURING binary relationship in Figure 3.18 (see next slide) can be derived from the ternary relationship OFFERS (based on the meaning of the relationships)

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College (Autonomous)

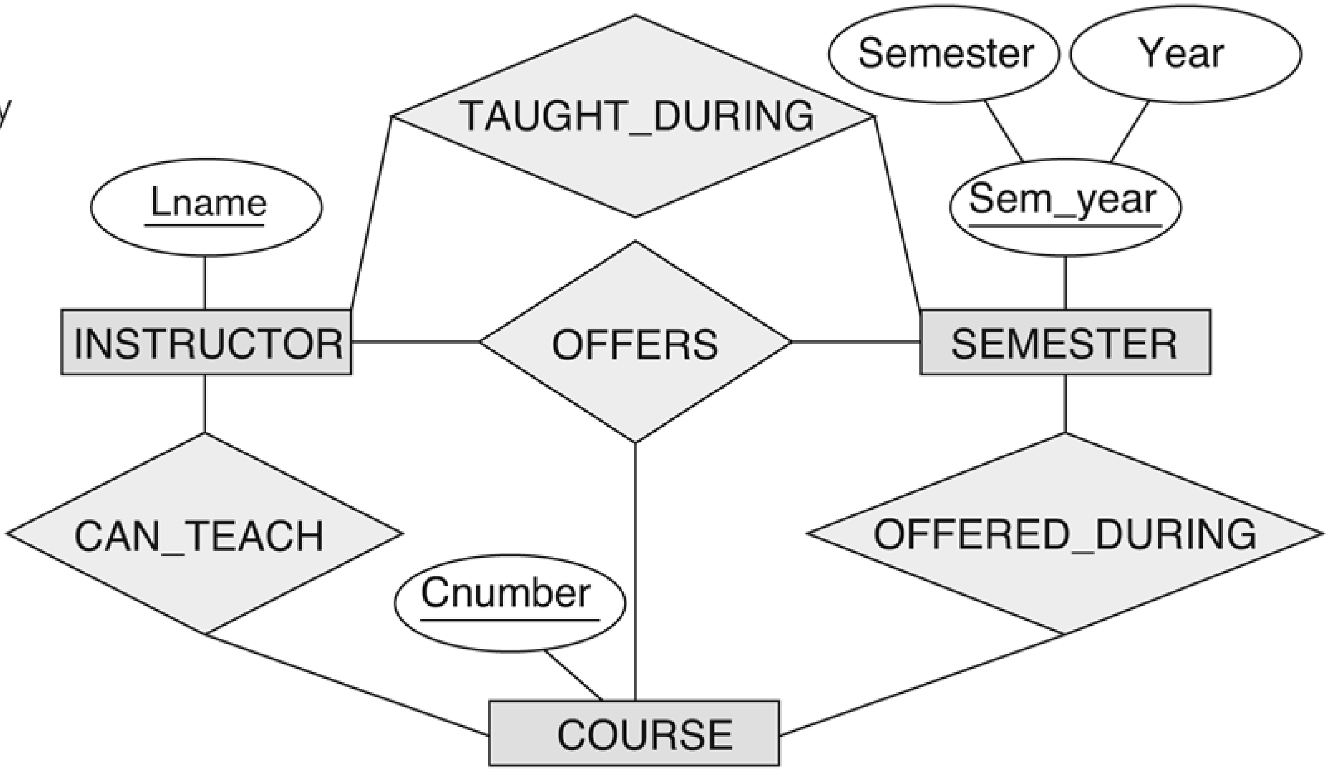
Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

example of ternary vs binary a ternary relationship

Figure 3.18
Another example of ternary versus binary relationship types.



GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College (Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

ER DIAGRAM – company database

WORKS_FOR, MANAGES, WORKS_ON, CONTROLS, SUPERVISION, DEPENDENTS_OF

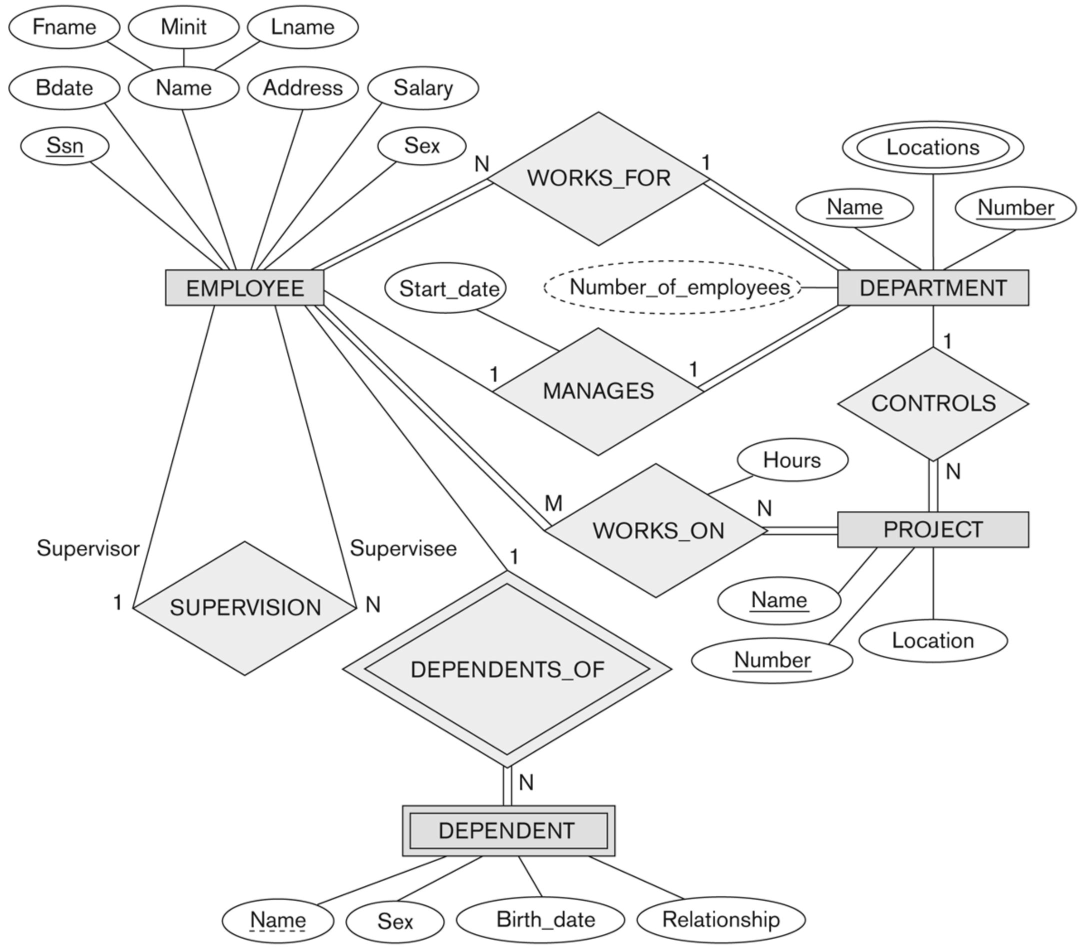


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

KEYS

- Keys play an important role in the relational database.
- It is used to **uniquely identify any record or row of data from the table.** It is also used to establish and identify relationships between tables.

KEYS Example

STUDENT
ID
Name
Address
Course

PERSON
Name
DOB
Passport_Number
License_Number
SSN

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

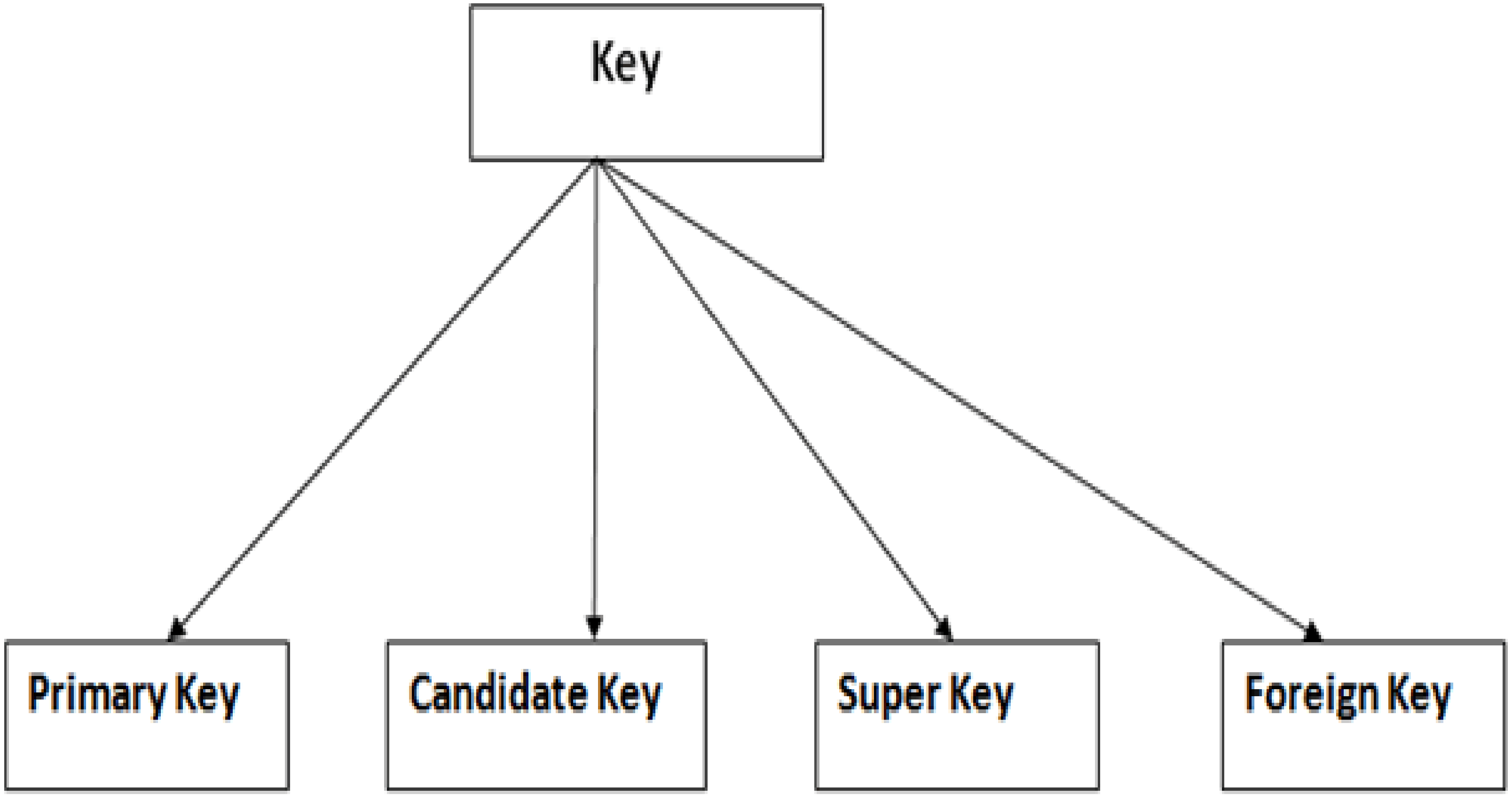
Presidency College (Autonomous)

Reaccredited by NAAC with A+

Presidency Group

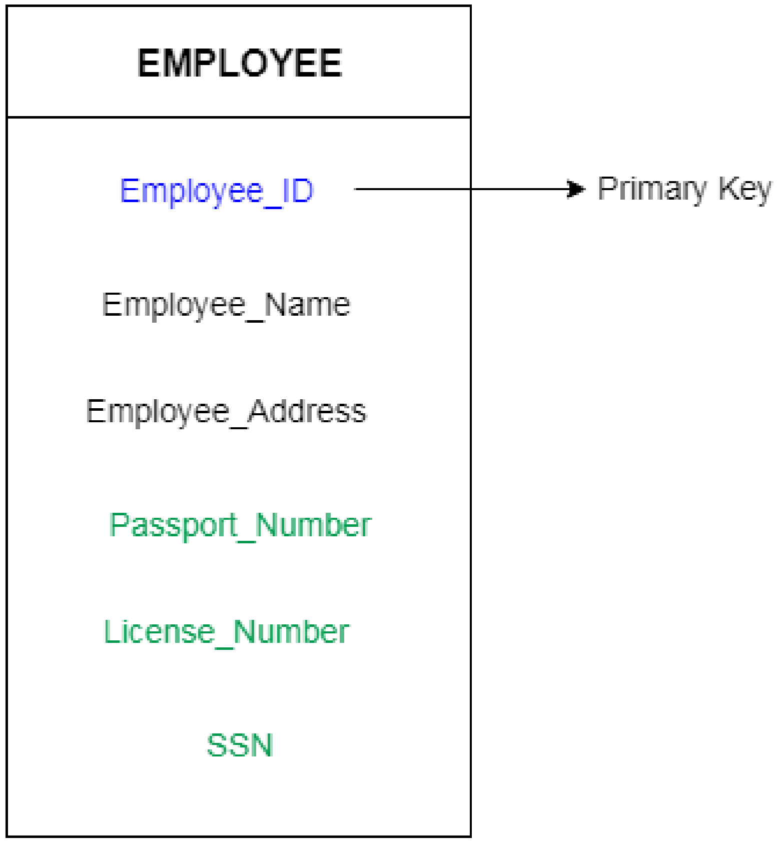
OVER 40 YEARS OF ACADEMIC WISDOM

Types of KEYS



Primary key

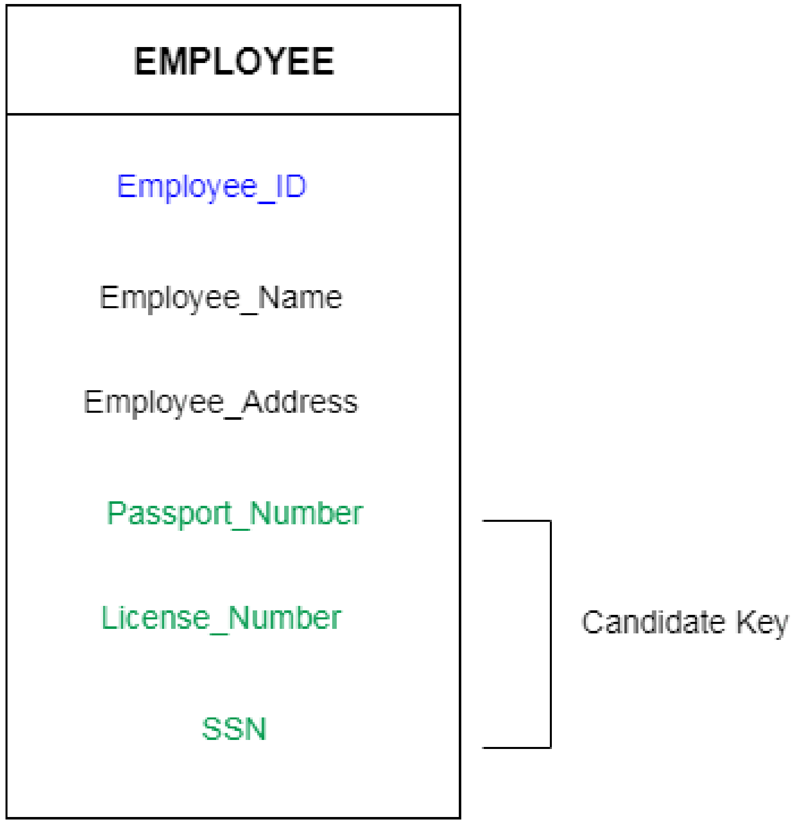
- It is the first key which is used to identify one and only one instance of an entity uniquely. The key which is most suitable from those lists become a primary key.



Candidate Key

- A candidate key is an attribute or set of an attribute which can uniquely identify a tuple.

- The remaining attributes except for primary key are considered as a candidate key. The candidate keys are as strong as the primary key. minimal super key is called candidate key.



GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

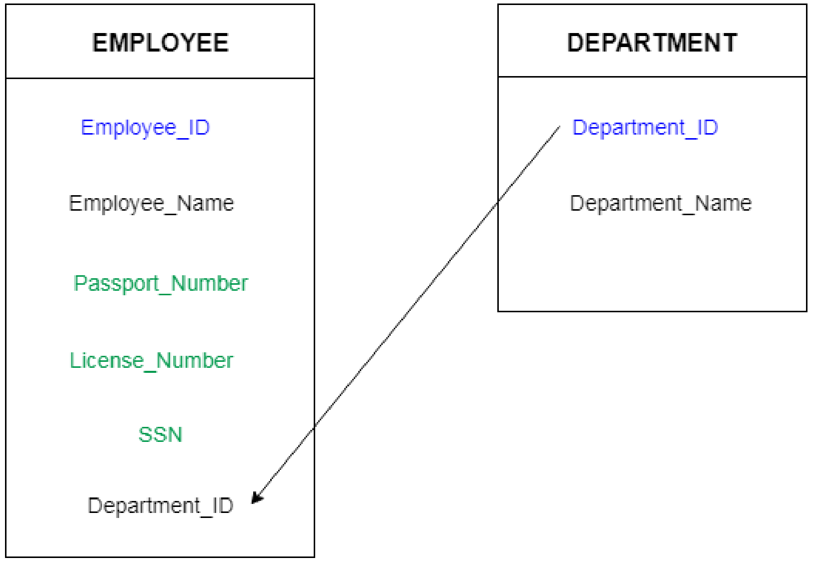
Super Key

- Super key is a set of an attribute which can uniquely identify a tuple. Super key is a superset of a candidate key.
- **For example:** In the above EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME) the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.
- The super key would be EMPLOYEE-ID, (EMPLOYEE_ID, EMPLOYEE-NAME), etc.

Foreign key

- Foreign keys are the column of the table which is used to point to the primary key of another table.

- In a company, every employee works in a specific department, and employee and department are two different entities. So we can't store the information of the department in the employee table. That's why we link these two tables through the primary key of one table.



File Organization

- A database consist of a huge amount of data. The data is grouped within a table in RDBMS, and each table have related records.
- A user can see that the data is stored in form of tables, but in actual this huge amount of data is stored in physical memory in form of files.
- **File –** A file is named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Types of File Organization

- Various methods have been introduced to Organize files. These methods have advantages and disadvantages based on access or selection
- Sequential File Organization(SORTED)
  - Pile File Method
  - Sorted File Method
- Heap File Organization (UNSORTED)
- Hash File Organization
  - External
  - Internal

Sequential File Organization(SORTED)

- Sequential File Organization: (SORTED) This method is the easiest method for file organization. In this method, files are stored sequentially.
- This method can be implemented in two ways:
  - Pile File Method
  - Sorted File Method

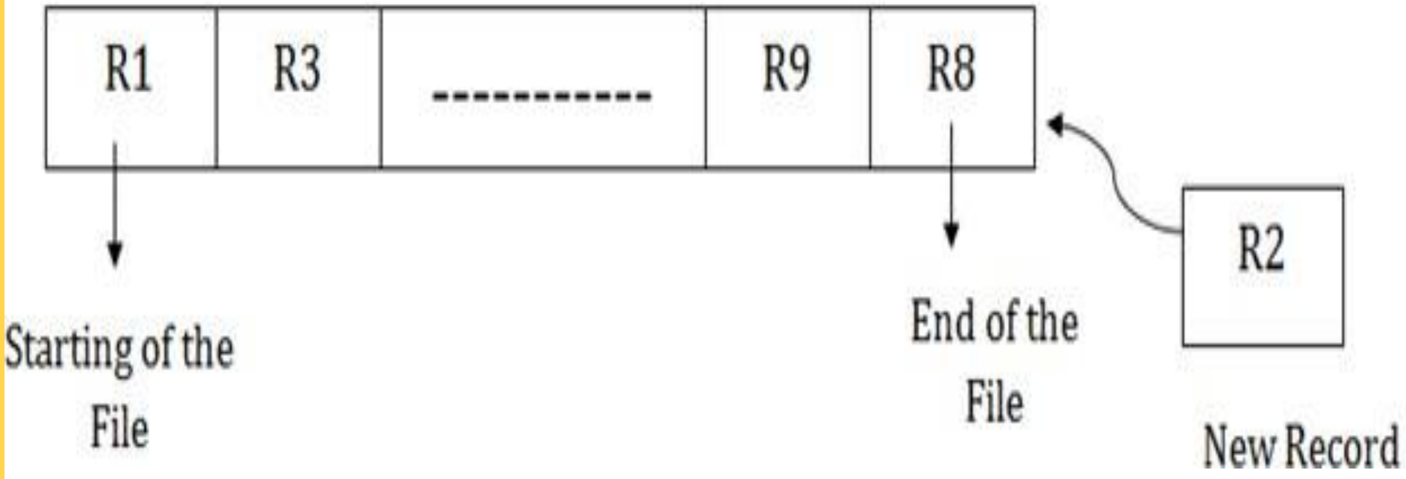
Pile File Method

- In this method, we store the record in a sequence, i.e., **one after another. Here, the record will be inserted in the order in which they are inserted into tables**.
- In case of updating or deleting of any record, the record will be searched in the memory blocks. When it is found, then it will be marked for deleting, and the new record is inserted.

Pile File Method

Insertion of the new record:

Suppose we have four records R1, R3 and so on upto R9 and R8 in a sequence. Hence, records are nothing but a row in the table. Suppose we want to insert a new record R2 in the sequence, then it will be placed at the end of the file. Here, records are nothing but a row in any table.

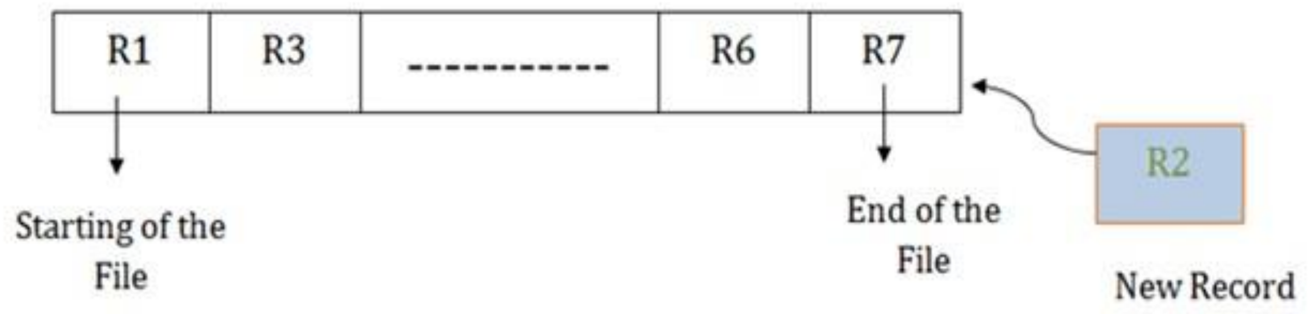


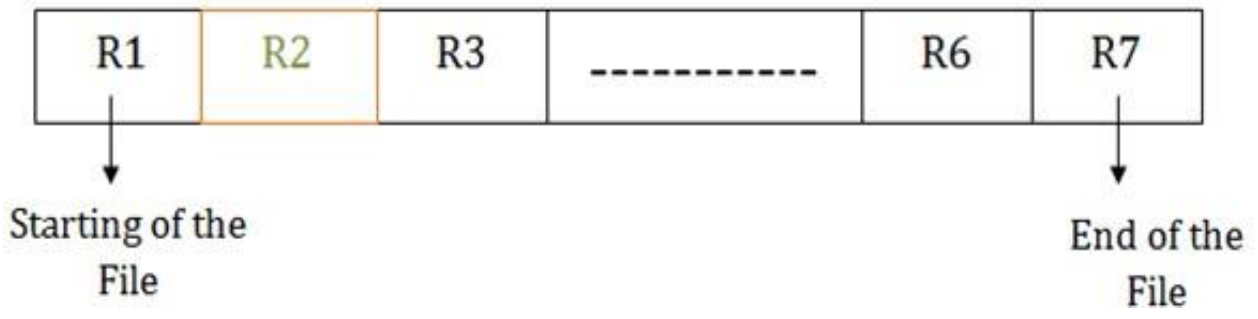
Sorted File Method

- In this method, the new record is always inserted at the file's end, **and then it will sort the sequence in ascending or descending order**. Sorting of records is based on any primary key or any other key.
- In the case of modification of any record, it will update the record and then sort the file, and lastly, the updated record is placed in the right place.

Sorted File Method

Suppose there is a preexisting sorted sequence of four records R1, R3 and so on upto R6 and R7. Suppose a new record R2 has to be inserted in the sequence, then it will be inserted at the end of the file, and then it will sort the sequence.





GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Advantages of sequential file organization

  - It contains a fast and efficient method for the huge amount of data.
  - In this method, files can be easily stored in cheaper storage mechanism like magnetic tapes.
- It is simple in design. It requires no much effort to store the data.
- This method is used when most of the records must be accessed like grade calculation of a student, generating the salary slip, etc.
- This method is used for report generation or statistical calculations.

Disadvantages of sequential file organization

- We cannot jump on a particular record that is required but we must move sequentially which takes time.
- Sorted file method takes more time and space for sorting the records

Hashing Technique

- Hashing method provides very fast access to records on certain search conditions.
- The search condition must be an equality condition on a single field called the hash field of the file.
- The hash field is also a key field of the file, in which case it is called hash key.
- **The basic terms associated with the hashing technique are:**

1. Hash table: It is simply an array that is having address of records.

2. Hash function: It is the transformation of a key into the corresponding location or address in the hash table. (it can be defined as function that takes key as input and transforms it into a hash table index)

3. Hash key: Let R be a record and its key hashes into a key called hash key.

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Hashing Technique

Assume a table with 8 slots:

Hash key = key % table size

4	=	36	%	8	
2	=	18	%	8	
0	=	72	%	8	
3	=	43	%	8	
6	=	6	%	8	

[0]	72
[1]	
[2]	18
[3]	43
[4]	36
[5]	
[6]	6
[7]	

Types of Hashing

1. Internal Hashing: The hashing technique employed for internal files is known as Internal Hashing. Internal files are the files stored in primary memory
2. External Hashing: The hashing technique employed for disk files is known as External hashing

Internal Hashing

- **Internal hashing:** For internal files, hash table is an array of records, having array index ranges from 0 to M-1, let us consider a hash function H(K) such that **H(K) = key MOD M** which produce a remainder between 0 and M-1 depending on the value of key, this value then used for the record
- Hashing is typically implemented as a **hash table** through the use of an array of records. Let the array index range is from 0 to M–1, as shown in Figure(a); then we have *M* **slots** whose addresses correspond to the array indexes. We choose a hash function that transforms the hash field value into an integer between 0 and M-1. One common hash function is the *h*(*K*) = *K* **mod** ***M*** function, which returns the remainder of an integer hash field value *K* after division by *M*; this value is then used for the record address

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College (Autonomous)

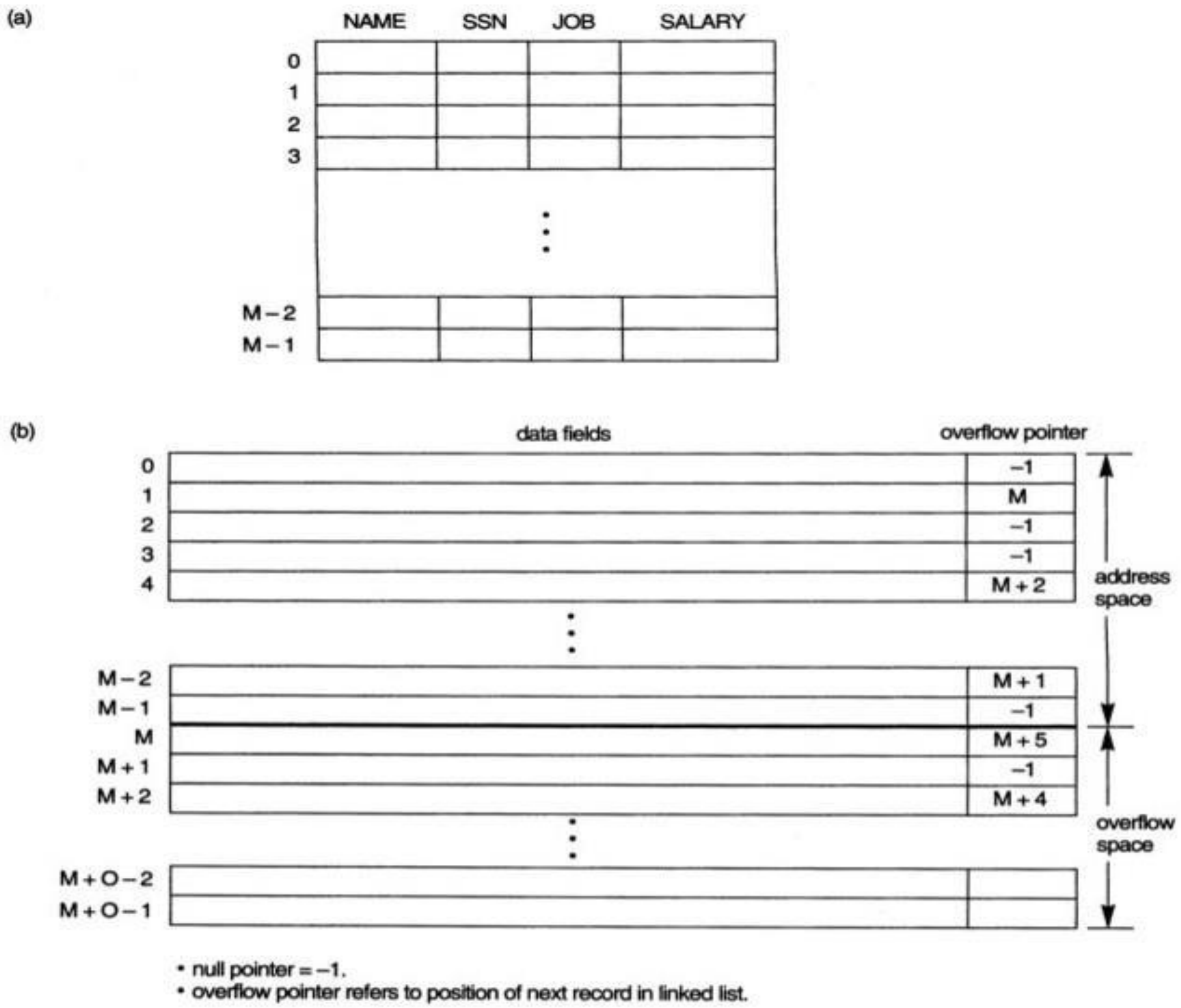
Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Internal Hashing

- Internal hashing data structures.
  - Array of *M* positions for use in internal hashing.
  - (f) Collision resolution by chaining records.



• null pointer = −1.
• overflow pointer refers to position of next record in linked list.

Collision

- A **collision** occurs when the hash field value of a record that is being inserted hashes to an address that already contains a different record.
- In this situation, we must insert the new record in some other position. The process of finding another position is called **collision resolution**.

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

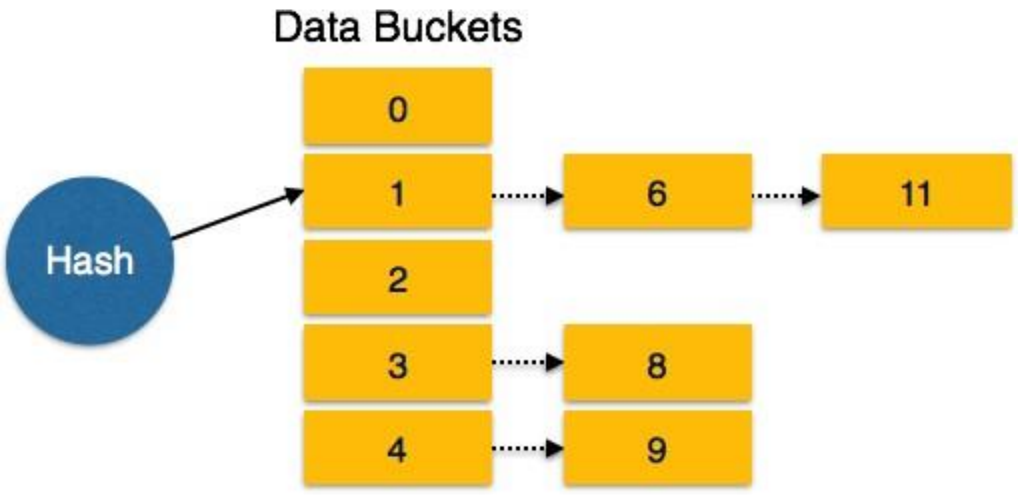
OVER 40 YEARS OF ACADEMIC WISDOM

Collision

- There are numerous methods for collision resolution, including the following:
- **Open addressing.** once a position specified by the hash address is found to be occupied, the program checks the subsequent positions in order until an unused (empty) position is found. Algorithm (b) may be used for this purpose.

Collision

- **Chaining. In** this method, various overflow locations are kept, usually by extending the array with several overflow positions. Additionally, a pointer field is added to each record location. A collision is resolved by placing the new record in an unused overflow location and setting the pointer of the occupied hash address location to the address of that overflow location. A linked list of overflow records for each hash address is thus maintained, as shown:



Collision

- **Multiple hashing.** The program applies a second hash function if the first results in a collision. If another collision results, the program uses open addressing or applies a third hash function and then uses open addressing if necessary.

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Indexing

- **Indexing** is a data structure technique which allows you to quickly retrieve records from a database file.
- An Index is a small table having only two columns.
- The first column comprises a copy of the primary or candidate key of a table.
- Its second column contains a set of pointers for holding the address of the disk block where that specific key value stored.

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

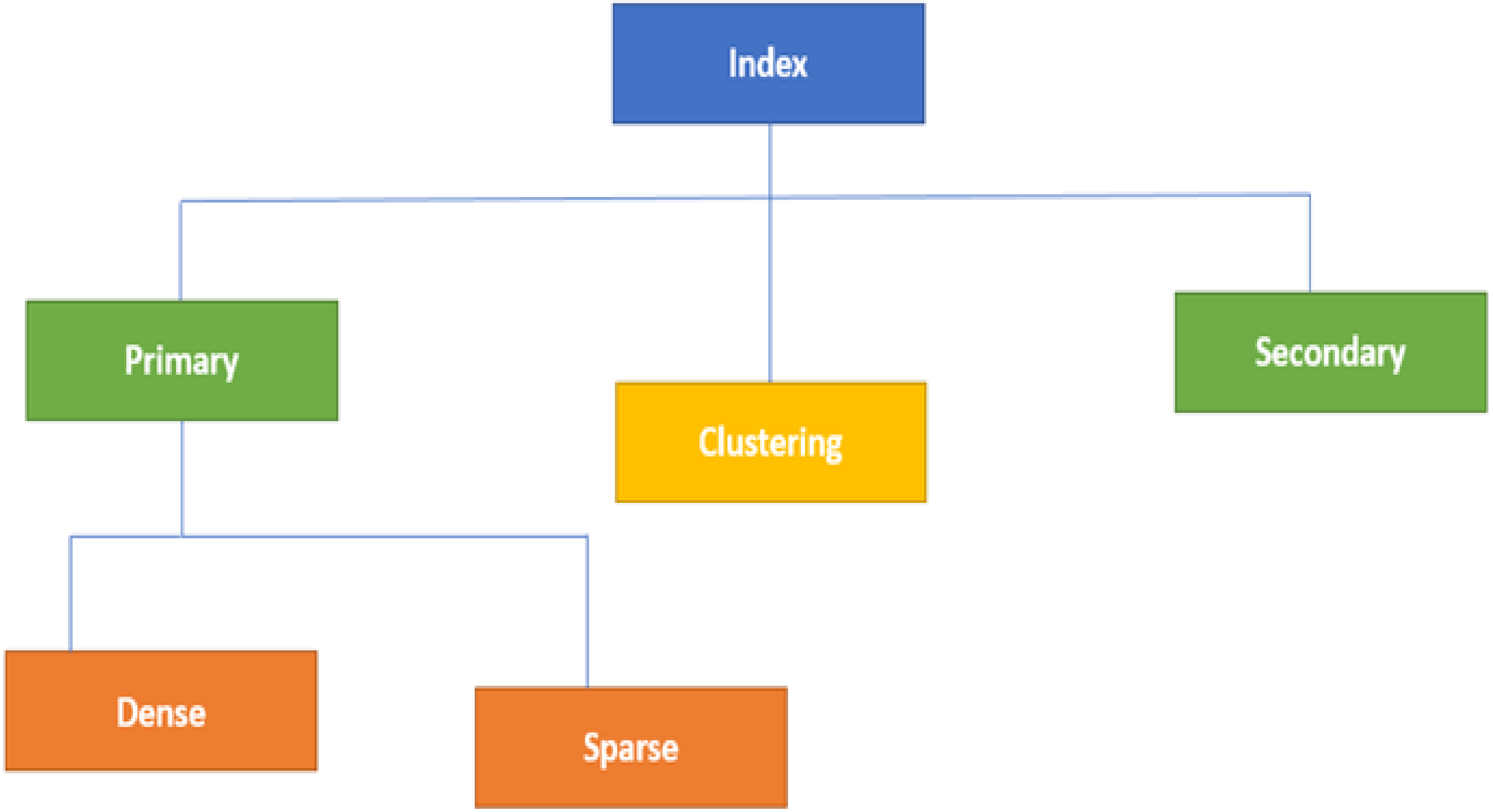
Presidency College (Autonomous)

Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Types of single level ordered indexes



Single level ordered indexes

- **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, or a non-key with duplicate values.
- **Clustering Index** − Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.

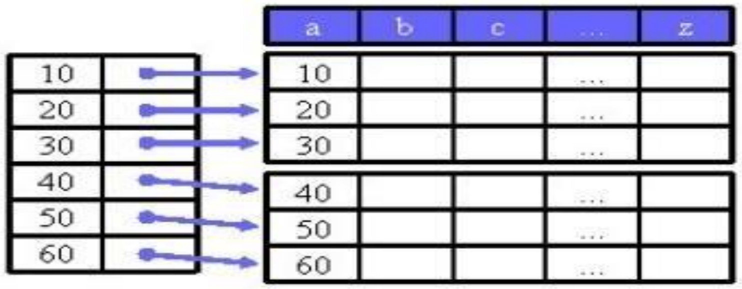
Primary Index

- Primary Index is an ordered file which is fixed length size with two fields.
  - The first field is the same a primary key and second,
  - filed is pointed to that specific data block.
  - In the primary Index, there is always one to one relationship between the entries in the index table.
- The primary Indexing in DBMS is also further divided into two types.
  - Dense Index
  - Sparse Index

Dense Index

- In a dense index, a record is created for every search key valued in the database. This helps you to search faster but needs more space to store index records
- In this Indexing, method records contain search key value and points to the real record on the disk

✓ A dense index has one index entry for every search key value



- every data record is represented in the index
- an existence search of a record may be done via index only
- the record is directly found in a block using the pointer, i.e., no search within the block

Dense Index

- It is an index record that appears for only some of the values in the file. Sparse Index helps you to resolve the issues of dense Indexing in DBMS.
- In this method of indexing technique, a range of index columns stores the same data block address, and when data needs to be retrieved, the block address will be fetched.

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency College
(Autonomous)

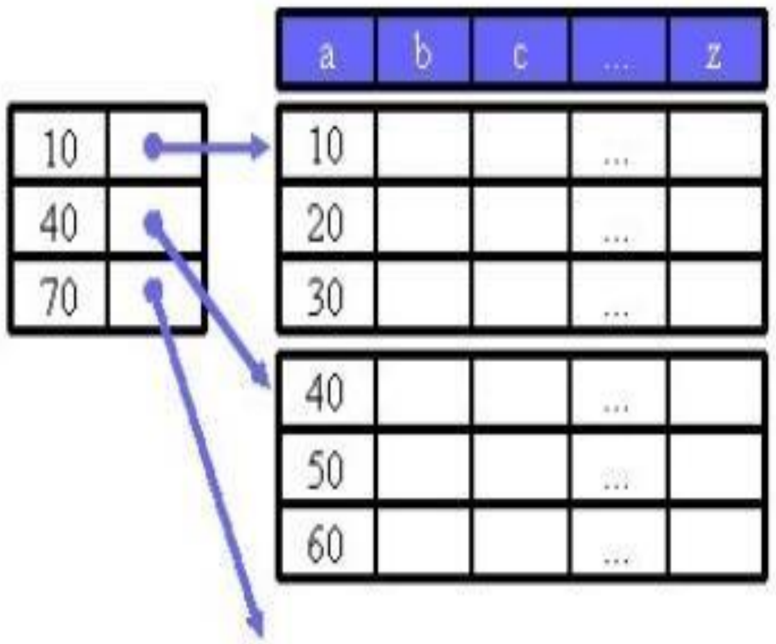
Reaccredited by NAAC with A+

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

Sparse Index

A sparse index has one index entry for every data block, i.e., the key of the first record



- only one index field per block, i.e., less index data
- cannot find out if a record exists only using the index

Other Index

- **Multilevel Indexing in Database** is created when a primary index does not fit in memory.
- In this type of indexing method, you can reduce the number of disk accesses to short any record and kept on a disk as a sequential file and create a sparse base on that file.

Multi-key indexes

- Multi-key indexes are **used to index all the elements of an array, or all the elements and/or all the keys of a map**. As a result, for each table row, the index contains as many entries as the number of elements/entries in the array/map that is being indexed (modulo duplicate elimination).