K. K. Wagh Polytechnic, Nashik
Hirabai Haridas Vidyanagari, Amrutdham, Panchavati, Nashik-422003
Department of Computer Technology

Unit 01- Database System Concepts

Course Outcomes (COs) / Unit Outcomes (UOs):

Course Outcome (CO 304.1):Design Normalized database on given data

Unit Outcomes:

State the importance of DBMS over file processing in the given situation

Describe the overall structure of the given DBMS

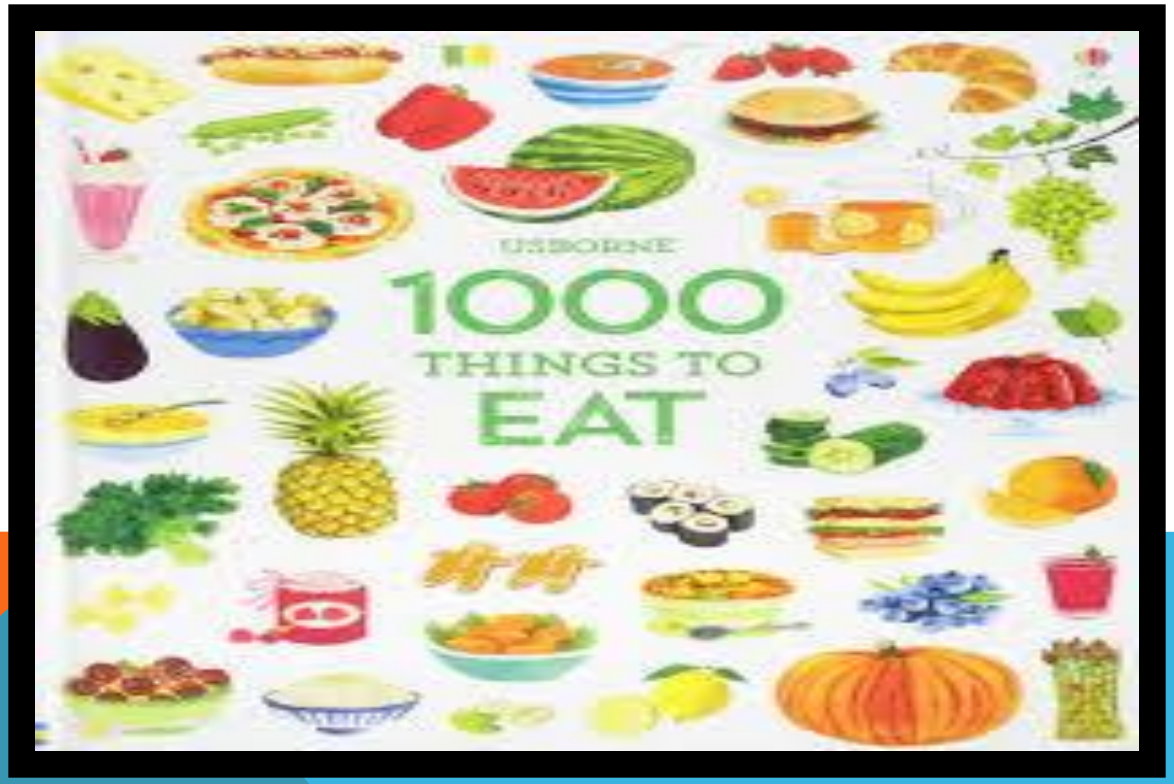
Identify the relevant database model in the given situation

Draw the E-R diagram of the given database & identify relationship between the entities

❖ Data

-Data is the collection of facts stored in database .

-The term data is define as “a set of isolated and unrelated raw facts with an implicit meaning”.



❖ DataBase

- **Database** is a collection of information that is organized so that it can be easily accessed, managed and updated.

❖ DBMS

- A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data.
- The collection of data, usually referred to as the database, contains information relevant to an enterprise.
- The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.

❖ DBMS

- Database systems are designed to manage large bodies of information.
- Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information.
- In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

❖ Database-System Applications

Databases are widely used. Here are some representative applications:

- **Banking:** For customer information, accounts, loans, and banking transactions.
- **Airlines**: For reservations and schedule information. Airlines were among the first to use databases in a geographically distributed manner.
- **Universities**: For student information/ course registrations, and grades.

❖ Database-System Applications

Databases are widely used. Here are some representative applications:

- **Credit card transactions :** For purchases on credit cards and generation of monthly statements.
- **Telecommunication :**For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.

❖Database-System Applications

- **Finance** : for storing information about holdings, sales and purchases of financial instruments such as stocks and bonds; also for storing real-time market data to enable on-line trading by customers and automated trading by the firm.

- **Sales** :For customer ,product, and purchase information.

- **On-line Retailers** : For sales data noted above plus on-line order tracking/ generation of recommendation lists, and maintenance of on-line product evaluations.

❖Database-System Applications

- **Manufacturing :** For management of the supply chain and for tracking production of items in factories, inventories of items in warehouses and stores, and orders for items.

- **Human Resource** : For information about employees, salaries, payroll taxes, benefits, and for generation of pay checks

❖ Disadvantages of File Processing System

➢ **Data redundancy and Inconsistency.**

Since different programmers create the **files and application** programs over a long period, the various files are likely to have different structures and the programs may be written in several programming languages. Moreover , the same information may be duplicated in several places (files).

❖ Disadvantages of File Processing System

➢Data redundancy and Inconsistency.

Example: customer may appear in a file that consists of **savings-account** records and in a file that consists of **checking-account** records. This redundancy leads to higher storage and access cost. In addition, it may lead to data inconsistency; For example, the address and telephone number of a particular

that is, the various copies of the same data may no longer agree. For example, a changed customer address may be reflected in **savings-account** records but not elsewhere in the system.

❖ Disadvantages of File Processing System

➢ **Data isolation :**

Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.

➢ **Integrity problems:**

The data values stored in the database must satisfy certain types of consistency constraints. For example, the balance of certain types of bank accounts may never fall below a prescribed amount (say, 500 Rs.). Developers enforce these constraints in the system by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.

❖ Disadvantages of File Processing System

➢ **Difficulty in Accessing Data**

-Suppose that one of the bank officers needs to find out the names of all customers who live within a particular postal-code area. The officer asks the data-processing department to generate such a list. Because the designers of the original system did not anticipate this request , there is no application program on hand to meet it. There is, however, an application program to generate the list of all customers.

➢ The bank officer has now two choices: either obtain the list of all customers and extract the needed information **manually** or ask a system programmer to **write the necessary application program**. Both alternatives are obviously unsatisfactory.

❖ Disadvantages of File Processing System

➢ Atomicity Problems

- A computer system, Like any other mechanical or electrical device, is subject to failure. In many applications, it is crucial that, if a failure occurs, the data be restored to the consistent state that existed prior to the failure.

- Consider a program to transfer 1000 RS from account A to account B. If a system failure occurs during the execution of the Program, it is possible that the 1000 Rs was removed from account A but was not credited to account B, resulting in an inconsistent database state.
- Clearly, it is essential to database consistency that either both the credit and debit occur, or that neither occur. That is, the funds transfer must be atomic-it must happen in its entirety or not at all.

❖ Disadvantages of File Processing System

➢ **Concurrent-Access Anomalies.**

For the sake of overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously. Indeed, today, the largest Internet retailers may have millions of accesses per day to their data by shoppers. In such an environment, interaction of concurrent updates is possible and may result in inconsistent data.

➢ **Security Problem**

❖ Data Abstraction:

- For the system to be usable, it must retrieve data efficiently. The need for efficiency has led designers to use complex data structures to represent data in the database.

- Since many database-system users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users' interactions with the system :

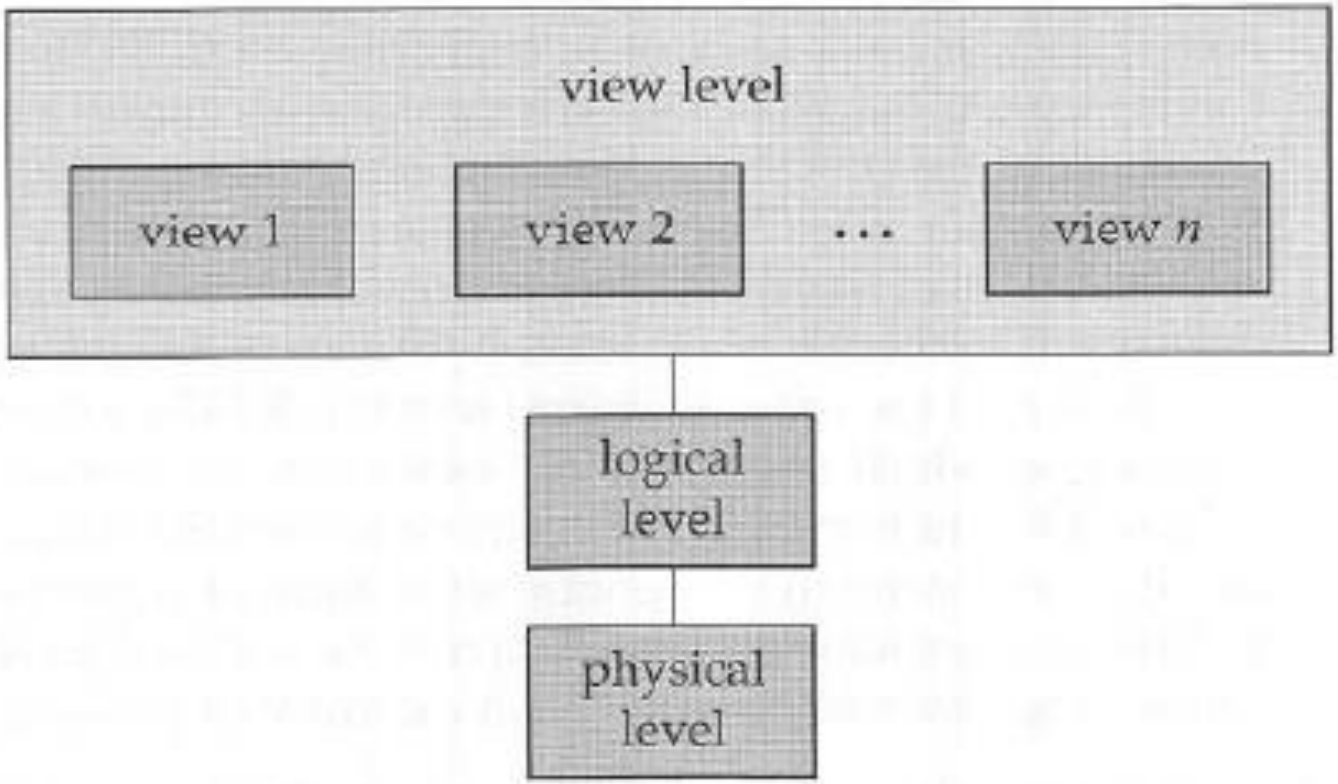


Figure 1.1 The three levels of data abstraction.

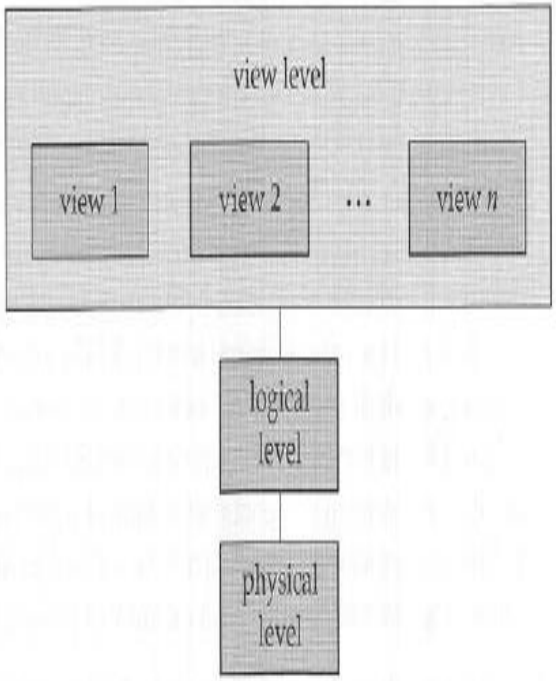


Figure 1.1 The three levels of data abstraction.

1. Physical Level

- The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in Detail.

2. Logical Level

- The next-higher level of abstraction describes What data are stored in the database, and what relationships exist among those data.

- The logical level thus describes the entire database in terms of a small number of relatively simple structures..

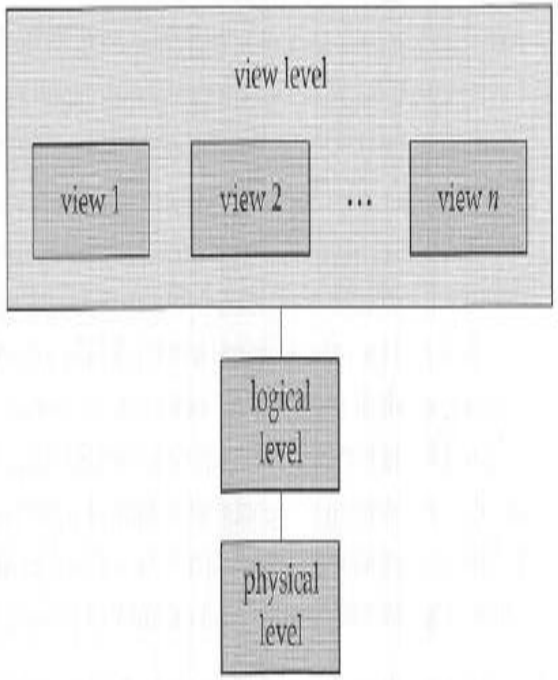


Figure 1.1 The three levels of data abstraction.

3. View Level

➢ The highest level of abstraction describes **only part of the entire database.**

➢ Even though the logical level uses simpler structures, complexity Remains because if the variety of information stored in a large database.

➢Many users of the database system do not need all this information; instead, they need to access only a part of the database.

➢The view level of abstraction exists to simplify their interaction with the system. The system may provide many views for the same database.

❖ Instances and Schema

- **Instances –** The collection of information stored in the database at a particular moment is called an instance of the database.

- **Schema –** The overall design of the database is called the database schema.

❖ Data Independence

- It is the ability of an application to change the storage structure & access strategy.
- This is a prime advantage of a database. In conventional systems applications are data-dependent.
- DBA must have the freedom to change storage structure or access strategy in response to changing requirements without having to modify existing applications.

❖ Data Independence

It is divided into 2 types:

1. **Logical Data Independence:** The ability to change the conceptual schema without having to change the external schemas and their application programs. It is easy to achieve this.

Ex. New fields can be added to database without disturbing old records.

2. **Physical Data Independence:** The ability to change the internal schema without having to change the conceptual schema. It is difficult to achieve this.

Ex. To achieve this, attributes of different tables are considered & changes are done. Then those changes are reflected to old one.2.

❖ Database Users

There are four different types of database-system users.

1) Naive Users

They are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.

consider a user who wishes to find her account balance over the World Wide Web. Such a user may access a form, where she enters her account number. An application program at the Web server then retrieves the account balance, using the given account number, and passes this information back to the user.

2) Application Programmers

Are computer professionals who write application programs. Application programmers can choose from many tools to develop user interfaces. **Rapid application development (RAD)** tools are tools that enable an application programmer to construct forms and reports with minimal programming effort.

3) Sophisticated Users

Interact with the system without writing programs. Instead, they form their requests in a database query language. They submit each such query to a query processor, whose function is to break down DML statements into instructions that the storage manager understands.

Analysts who submit queries to explore data in the database fall in this category.

4. Database Administrator

One of the main reasons for using DBMS is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA). The functions of a DBA include:

1)Schema Definition -The DBA creates the original database schema by executing a set of data definition statements in the DDL.

2)Storage Structure and Access-Method Definition.

4. Database Administrator

The functions of a DBA include:

3)Schema and Physical-Organization Modification. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

4) Granting of authorization for data access.

By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system

4. Database Administrator

The functions of a DBA include:

5) **Routine maintenance,** Examples of the database administrator's routine maintenance activities are:

- Periodically backing up the database, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.

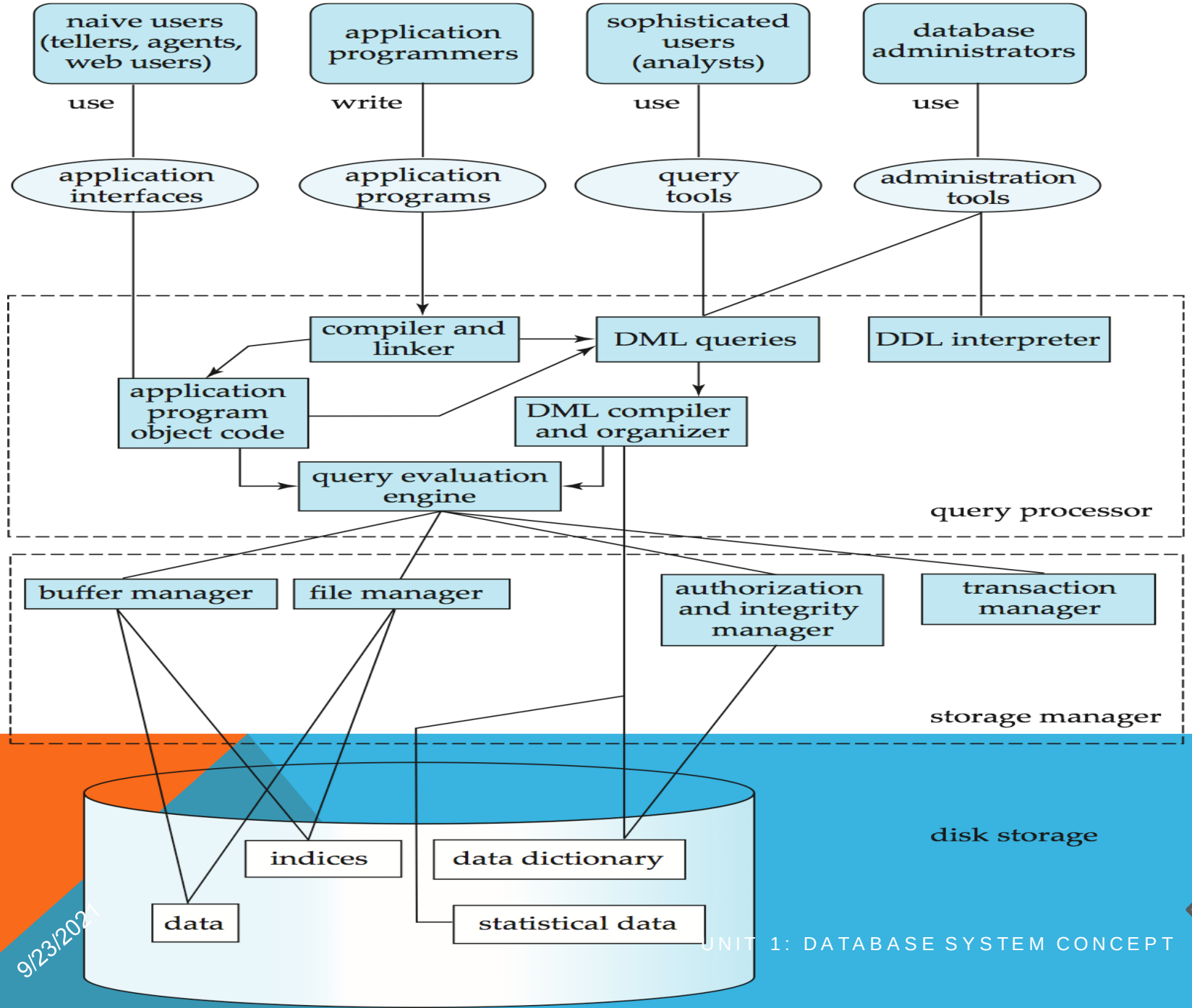
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.

- Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.

❖ Components of DBMS and Overall Structure of DBMS.

The functional components of a database system can be broadly divided into the

1. **Storage Manager,**
2. **Query Processor**
3. **Data Base Users**


naive users (tellers, agents, web users)
application programmers
sophisticated users (analysts)
database administrators
use
write
use
use
application interfaces
application programs
query tools
administration tools
compiler and linker
DML queries
DDL interpreter
application program object code
DML compiler and organizer
query evaluation engine
query processor
buffer manager
file manager
authorization and integrity manager
transaction manager
storage manager
indices
data dictionary
data
statistical data
disk storage

1. Storage Manager

➢The storage manager is important because databases typically require a large amount of storage space.

➢ Corporate databases range in size from hundreds of gigabytes to, for the largest databases ,Terabytes of data. A gigabyte is 1000 megabytes(1 billion bytes), and a terabyte is 1 million megabytes (1 trillion bytes).

➢ Since the main memory of computers cannot store this much information, the information is stored on disks.

1. Storage Manager

➢Data are moved between disk storage and main memory as needed.

➢Since the movement of data to and from disk is slow relative to the speed of the central processing unit, it is imperative that the database system structure the data so as to minimize the need to move data between disk and main memory.

A storage manager is a program module that provides the interface between the low level data stored in the database and the application programs and queries submitted to the system

1. Storage Manager

Thus, the **storage manager** is responsible for storing, retrieving, and updating data in the database.

The storage manager components include :

- **Authorization and Integrity Manager** - which tests for the satisfaction of integrity Constraints and checks the authority of users to access data.

- **Transaction Manager** - which ensures that the database remains in a consistent(correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.

1. Storage Manager

➢**File Manager-** which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

➢**Buffer Manager-** which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory.

The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.

1. Storage Manager

The storage manager implements several data structures as part of the physical system implementation:

- **Data files-** which store the database itself.
- **Data dictionary-** which stores *metadata* about the structure of the database, in particular the schema of the database.
- **Indices** -which can provide *fast access to data items.* Like the index in this textbook, a database index provides pointers to those data items that hold a particular value.

2.Query Processor

- The query processor is important because it helps the database system simplify and facilitate access to data.

- However, quick processing of updates and queries is important.

- It is the job of the database system to translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level.

2. Query Processor

The query processor components include :

- **DDL Interpreter** - which interprets DDL statements and records the definitions in the data dictionary.

- **DML Compiler** - which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

  A query can usually be translated into any of a number of alternative evaluation. plans that all give the same result. The DML compiler also performs

2. Query Processor

The query processor components include :

- **Query Optimization** - that is, it picks the lowest cost evaluation plan from among the alternatives.

- **Query Evaluation Engine** - which executes low-level instructions generated by the DML compiler.

❖ Advantages of DBMS over file Processing

1) Independence of data and program –

This is the prime advantage of database .both the user program and database can be altered independently of each other which save time and money which would be required to retain consistency .

2) Controlling redundancy – In DMBS there is no redundancy (duplicate data). If any type of duplicate data arise ,then DBA can control and arrange data in non –redundant way.

it store data on the basis of Primary key, which is always unique key and have non-redundant information.

for e.g. Roll no is the Primary key to store the student data.

❖ Advantages of DBMS over file Processing

3) Data shareability – in a database system data can easily shared by different user . for example student data can be shared by teacher ,department , administrative block , account branch and laboratory .

4) Data integrity –Enforcing data integrity ensure the quality of data in the database .

5) Centralized Control – with the central control the database ,the DBA can ensure the standard are followed in the representation of data.

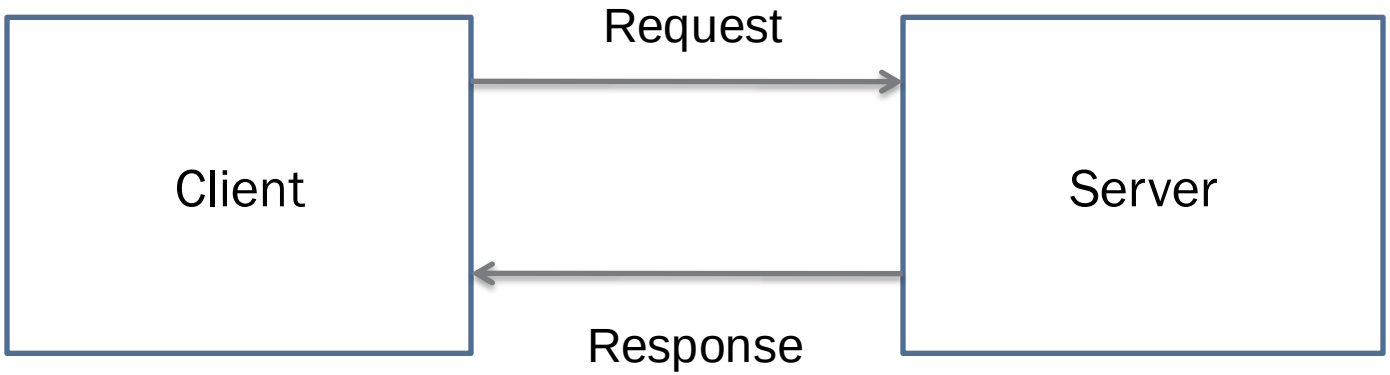
❖ Advantages of DBMS over file Processing

6) Data security – Having control over the database the DBA can ensure that access to the database is through proper channels and can be define the access right of any user to any data items or defined subset of the database.

The Security system must prevent corruption of data either accidentally or maliciously.

7) Performance and efficiency – The DBA can structure the database to in such way it's provide an overall services that is best for the enterprise

❖ Client –Server Architecture



❖ Client –Server Architecture

➢It's the Application Architecture where two or more system/Machine are connected through network.

➢ where one Machine is **Server** another is **Client** .

➢On *Client Machine user (Database User)* request for **Information** and *server Machine on which the database system runs*, which gives the **Response to client** .

1. Two –tier Architecture

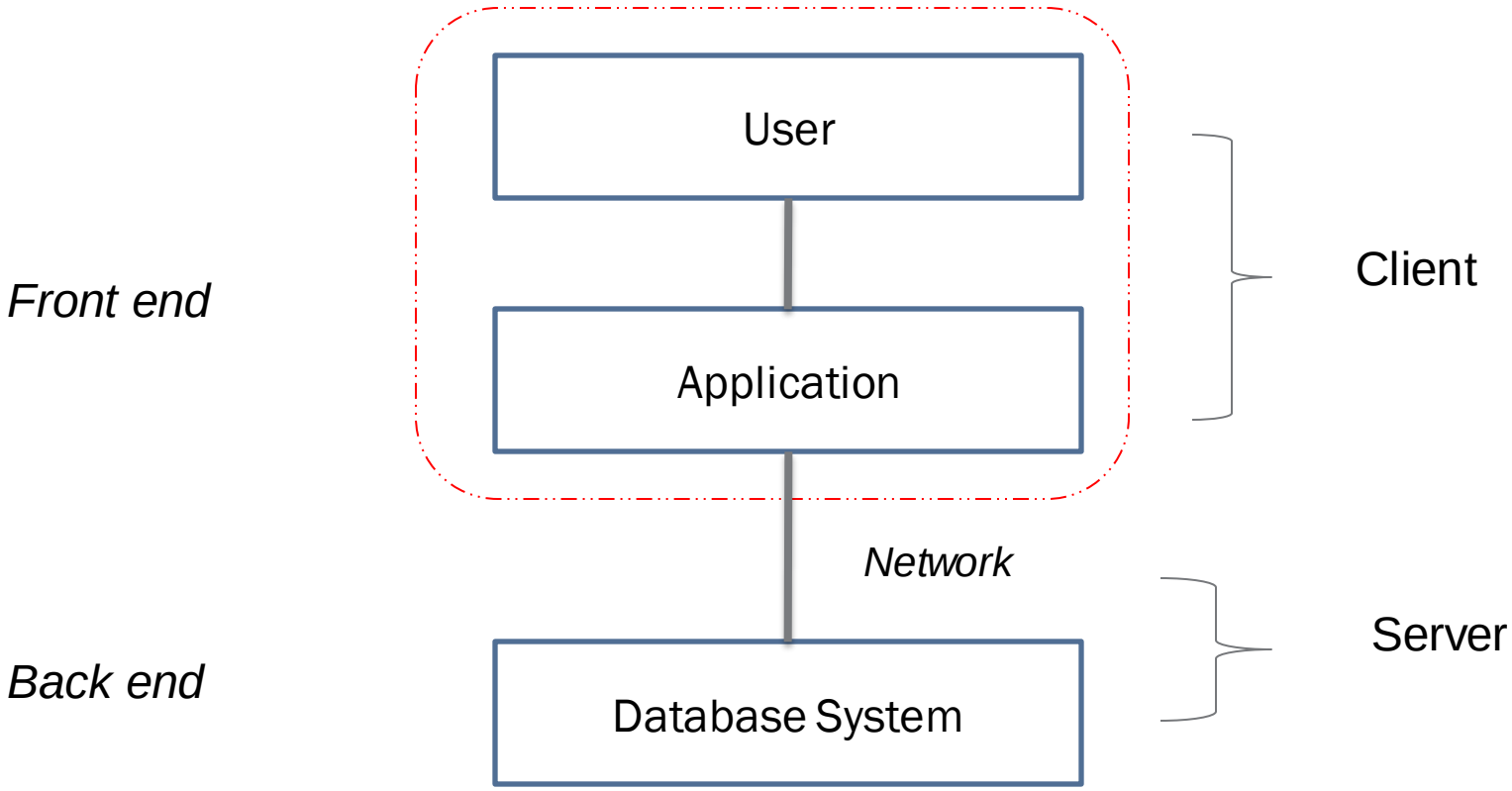


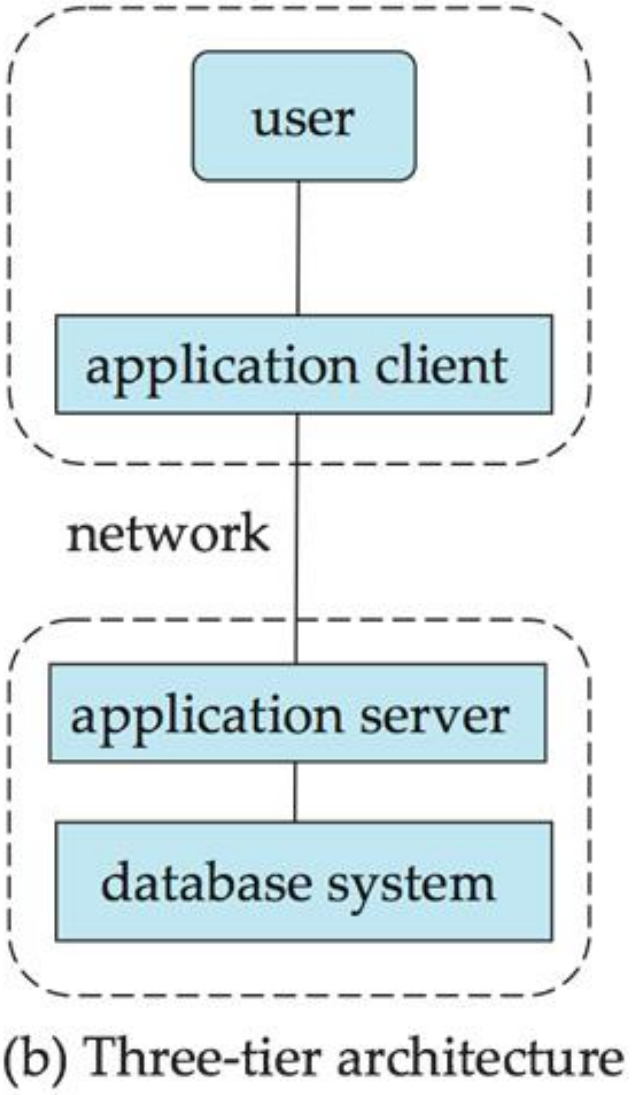
Fig. Two –tier Architecture

1. Two –tier Architecture

➢The Client machine, which invokes database system functionality at the server machine through query language statements.

➢Application Program interface standards like **ODBC** and **JDBC** are used for interaction between the client and the server.

2. Three–tier Architecture



Front end

Middleware

Back end

Client

Server

2. Three–tier Architecture

➢in a three-tier architecture, the **client machine** acts as merely a front
end and does not contain any direct database calls. Instead, the client end communicates with an application server, usually through a forms interface.

➢The application server in turn communicates with a database system to access data.

➢Three-tier applications are more appropriate for large applications, and for applications that run on the World Wide Web.

❖Data Models

- It is a set of concepts to describe the *structure* of a database, and certain *constraints* that the database should obey.
- Introduced by Dr. E. F. Codd in 1970.

- It describes simple, defined concepts of how user identifies the data.

- This data is in form of table which represents real world's person, place, things, or event.

Types of Data Model

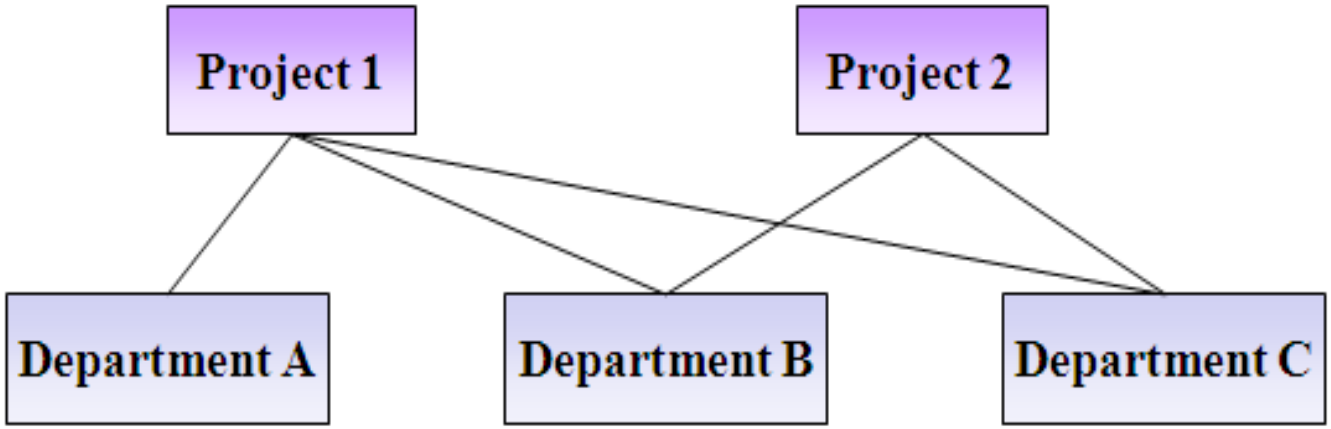
A) Network Data Model

B) Hierarchical Model

C) Relational Model

A) Network Data Model

- An expansion of the hierarchical database model with an owner-member relationship in which a member may have many owners
- It allows many to many relationships in data. Although 1:N or M:N relationship also permitted.
- It organizes data in in form of arbitrary graph



A) Network Data Model

Advantages:

- Network Model is able to model complex relationships and represents semantics of add/delete on the relationships.

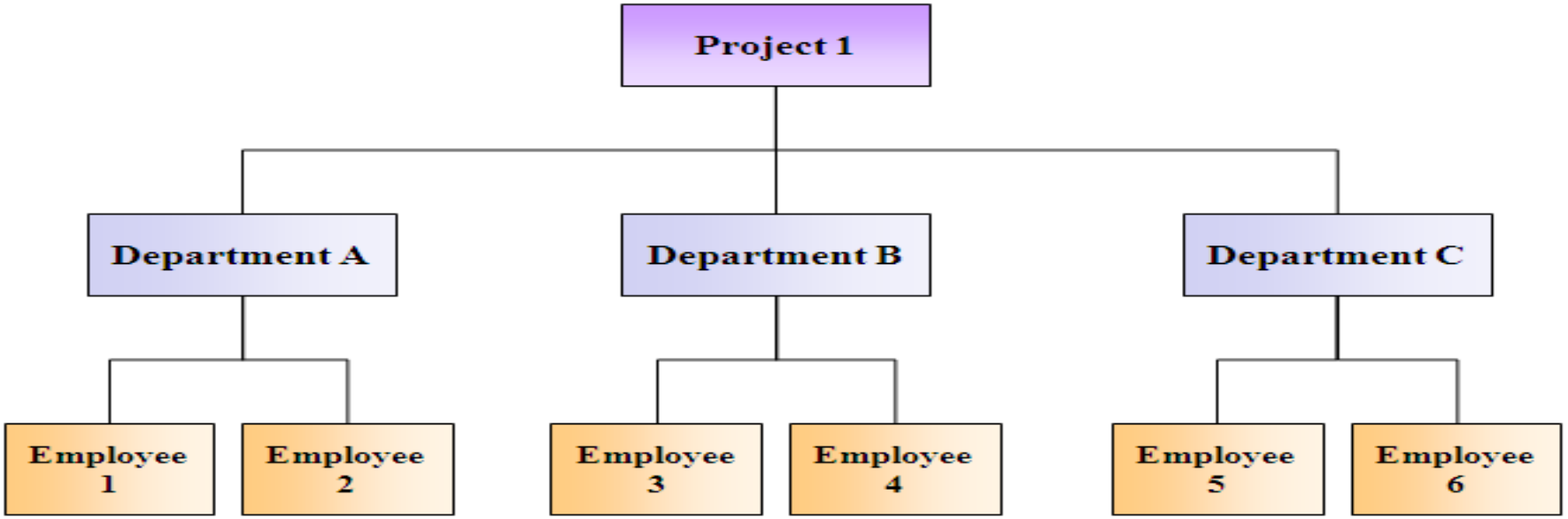
- Can handle most situations for modelling using record types and relationship types.
- Language is navigational; uses constructs like FIND, FIND member, FIND owner, FIND NEXT within set, GET etc. Programmers can do optimal navigation through the database.

Disadvantages:

- Navigational and procedural nature of processing Database contains a complex array of pointers that thread through a set of records.
- Little scope for automated "query optimization

B) Hierarchical Model

- A data model in which data are organized in a top-down, or inverted tree structure.
- It represents Parent-Child relationship structure.
- It organizes data in a tree structure.
- It allows 1:N mapping between record types.



B) Hierarchical Model

Advantages:

- Hierarchical Model is simple to construct and operate
- Corresponds to a number of natural hierarchically organized domains - e.g., assemblies in manufacturing, personnel organization in companies
- Language is simple; uses constructs like GET, GET UNIQUE, GET NEXT, GET NEXT WITHIN PARENT etc.

Disadvantages:

- Navigational and procedural nature of processing
- Database is visualized as a linear arrangement of records
- Little scope for "query optimization"

C) Relational Model

- It was introduced by Dr. E. F. Codd in 1970.
- Data is represented with the help of tables (Person, Place, Things, and Events etc.)
- All data elements are placed in two-dimensional tables, called relations that are equal to files.
- **It performs-**
  - Selecting: Data manipulation that eliminates rows according to certain criteria
  - Projecting: Data manipulation that eliminates columns in a table
  - Joining: Data manipulation that combines two or more table
  - Linking: Relating tables in a relational database together

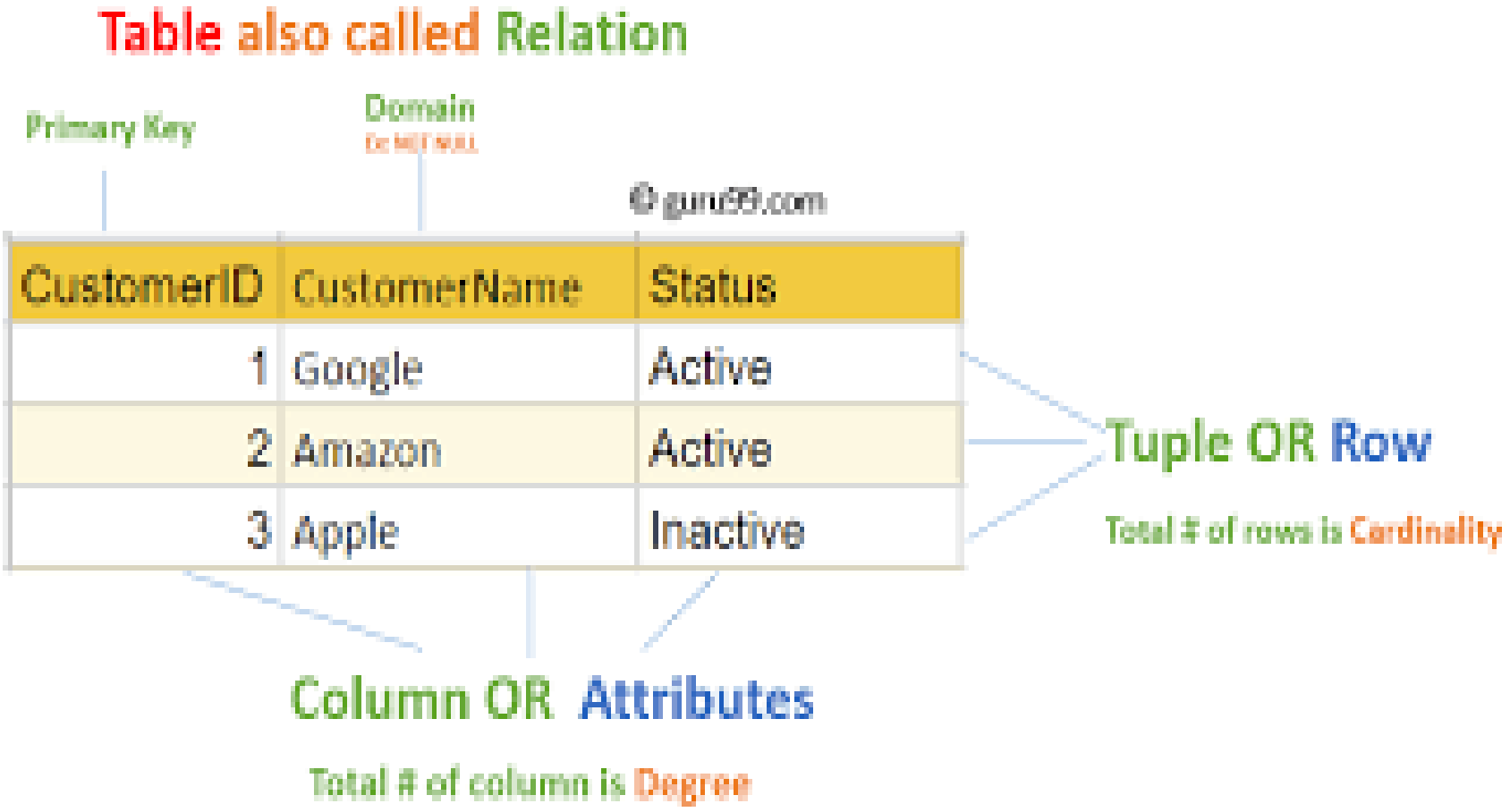


Table also called Relation
Primary Key
Domain
© guru99.com
CustomerID
CustomerName
Status
1
Google
Active
2
Amazon
Active
3
Apple
Inactive
Tuple OR Row
Total # of rows is Cardinality
Column OR Attributes
Total # of column is Degree

C) Relational Model

1. Attribute:

- It is Name of a column.
- It is also called as Arity, Degree or Order of a Table.
- Ex. For Emp table, E-Name, E-City etc. are the attributes of table.
- Number of Attributes in a relation or table is called as Degree of a relation.

2. Tuple:

- It is row in a table. Ex. Particular record in a table represents a tuple.
- Number of tuples in a relation or table is called as Cardinality of a relation.

3. Domain:

- It is set of allowable values for attribute

4.Null Value:

- It indicates that value for corresponding attribute is either not available or not applicable or absent.
- It is not equal to Zero.

C) Relational Model

Data Table 1: Project Table

Project Number	Description	Dept. Number
155	Payroll	257
498	Widgets	632
226	Sales manager	598

Data Table 2: Department Table

Dept. Number	Dept. Name	Manager SSN
257	Accounting	421-55-99993
632	Manufacturing	765-00-3192
598	Marketing	098-40-1370

Data Table 3: Manager Table

SSN	Last Name	First Name	Hire Date	Dept. Number
005-10-6321	Johns	Francine	10-7-65	257
549-77-1001	Buckley	Bill	2-17-79	650
098-40-1370	Fiske	Steven	1-5-85	598

Relational Key's Concept

1. Primary Key

- PRIMARY KEY constraint uniquely identifies each record in a database table.
- Primary keys must contain unique values.
- A primary key column cannot contain NULL values.
- To ensure that each record is unique in each table, we can set one field to be *Primary Key* field. For e. g. rno is a primary key attribute for student table.

Relational Key's Concept

2. Foreign Key

- A foreign key is a combination of column with a value based on the primary key values from another table.
- A foreign key constraint also known as referential constraint corresponds to a actual values of the primary key in the another table.
- Foreign Keys link to data in other tables

Relational Key's Concept

3. Super key

- It is a set of one or more attributes that, taken collectively, to identify uniquely an item in the entity set. For example, SSN is a superkey.

4. Candidate key

- Sometimes in relation, there are more than one attributes are having the unique identification property. Those attributes are known as candidate key. E.g. Consider Student (RNO, NAME, PER, BRANCH), if RNO and NAME are unique then both are known as Candidate keys.

Network Model	Hierarchical Model	Relational Model
Allowed the network model to support many to many relationships	One to many or one to one relationships	One to One, One to many, Many to many relationships
A record can have many parents as well as many children.	Based on parent child relationship	Based on relational data structures
CODASYL (Conference on Data Systems Languages)	Does not provide an independent standalone query interface	Relational databases are what brings many sources into a common query (such as SQL)
Retrieve algorithms are complex and symmetric	retrieve algorithms are complex and asymmetric	Retrieve algorithms are simple and symmetric
Does not suffer from any insertion anomaly.	Cannot insert the information of a child who does not have any parent.	Does not suffer from any insert anomaly.
Free from update anomalies.	Multiple occurrences of child records which lead to problems of inconsistency during the update operation	Free form update anomalies
Free from delete anomalies	Deletion of parent results in deletion of child records	Free from delete anomalies

❖ Data Modeling using E-R Model

- An E-R Diagram is a pictorial representation of the entities and the relationship between them.
- From this user can use the information structure of the application at a glance.
- Afterwards these ER Diagrams are used to design tables and databases.
- Relationships are just as important as entities—they are data that need to be stored in the DB
- Most relationships are binary, but they may be ternary or more as well
- In the E/R model, the structure of data is represented graphically, as an "entity-relationship diagram"

❖ Data Modeling using E-R Model

- E-R Model is based on a perception of a real world that consist of a set of basic objects called entities and relationships among this objects

Entity:

- Individual Object in a real world is called as Entity.
- An Entity is a thing in the real world with an independent existence.
- Eg. In a Relation Stud_Info "Stud" is an Entity.

Entity Set:

- Group of Objects in a system is called as Entity Set.
- Entity set ia a set of entities of the same type that share the same properties or Attributes.
- Ex. In a Relation Stud_Info "Stud", "Dept" is called as Entity Set.

❖ Data Modeling using E-R Model

Attributes:

- The properties of an Entity(ies) in the called as attribute.
- Ex. In a Relation Stud_Info "Stud" having "Attribute" Stud-Name

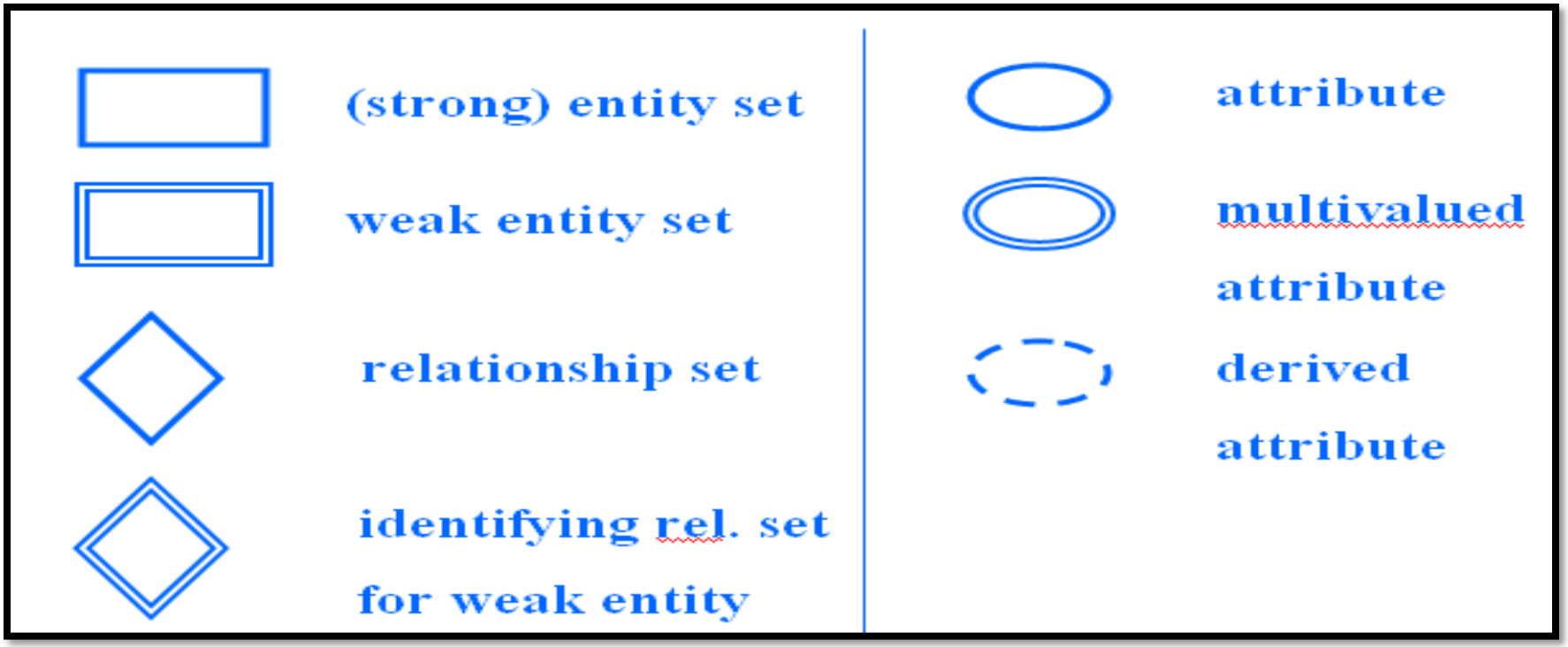
Relationship:

- The meaningful association among two or more entity Sets is called as Relationship
- Ex. For showing relation between "Stud_Info " & "Dept" Works keyword is used

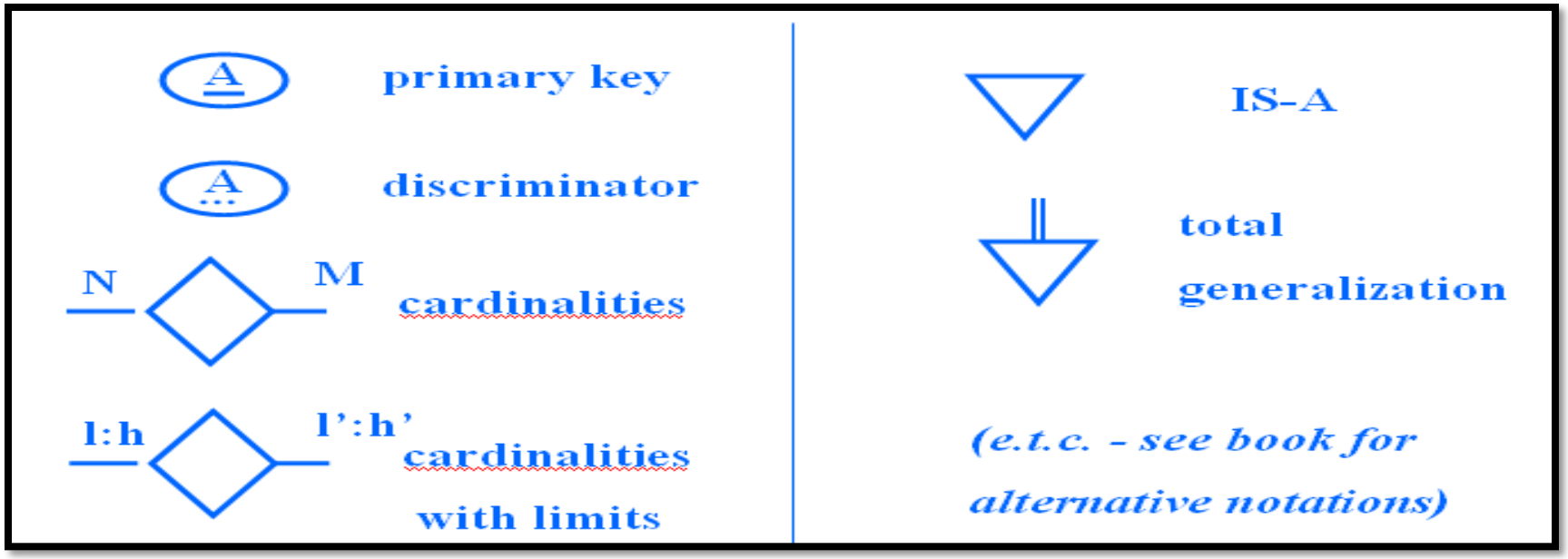
Keys:

A column or set of columns used to uniquely identify the record in a Relation

Components of E-R Model

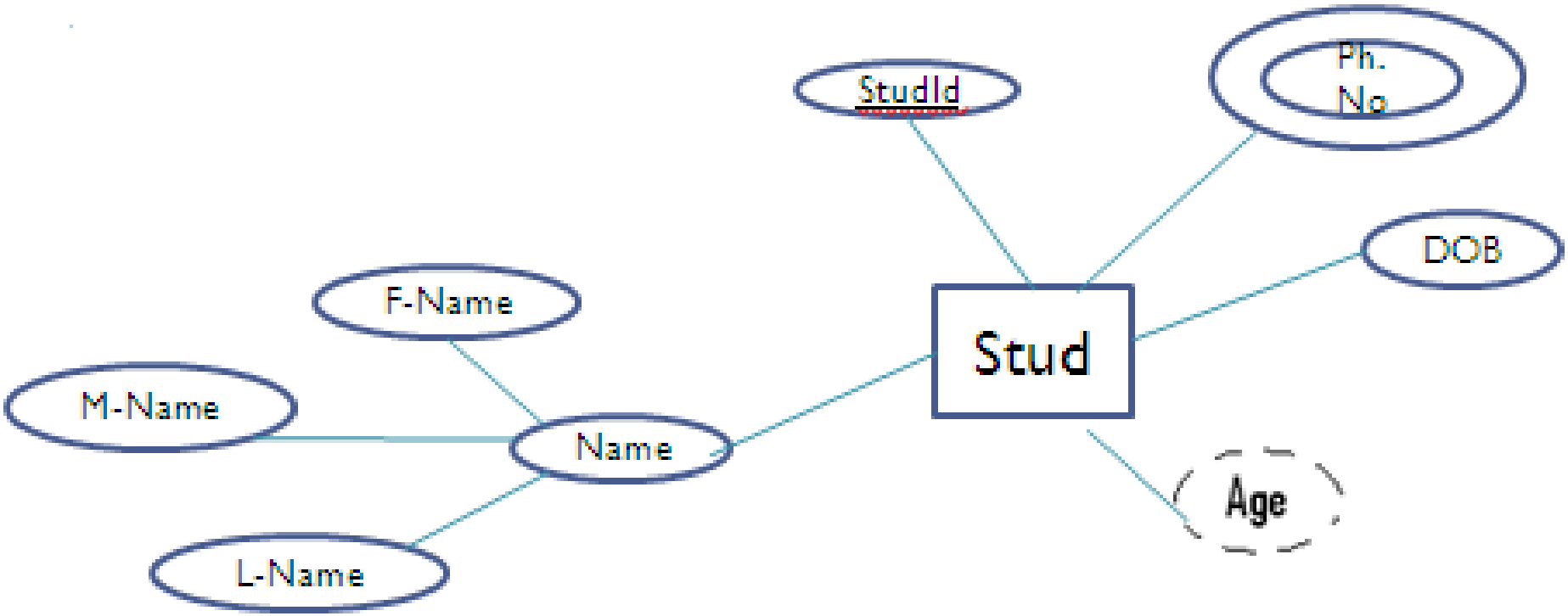


Components of E-R Model

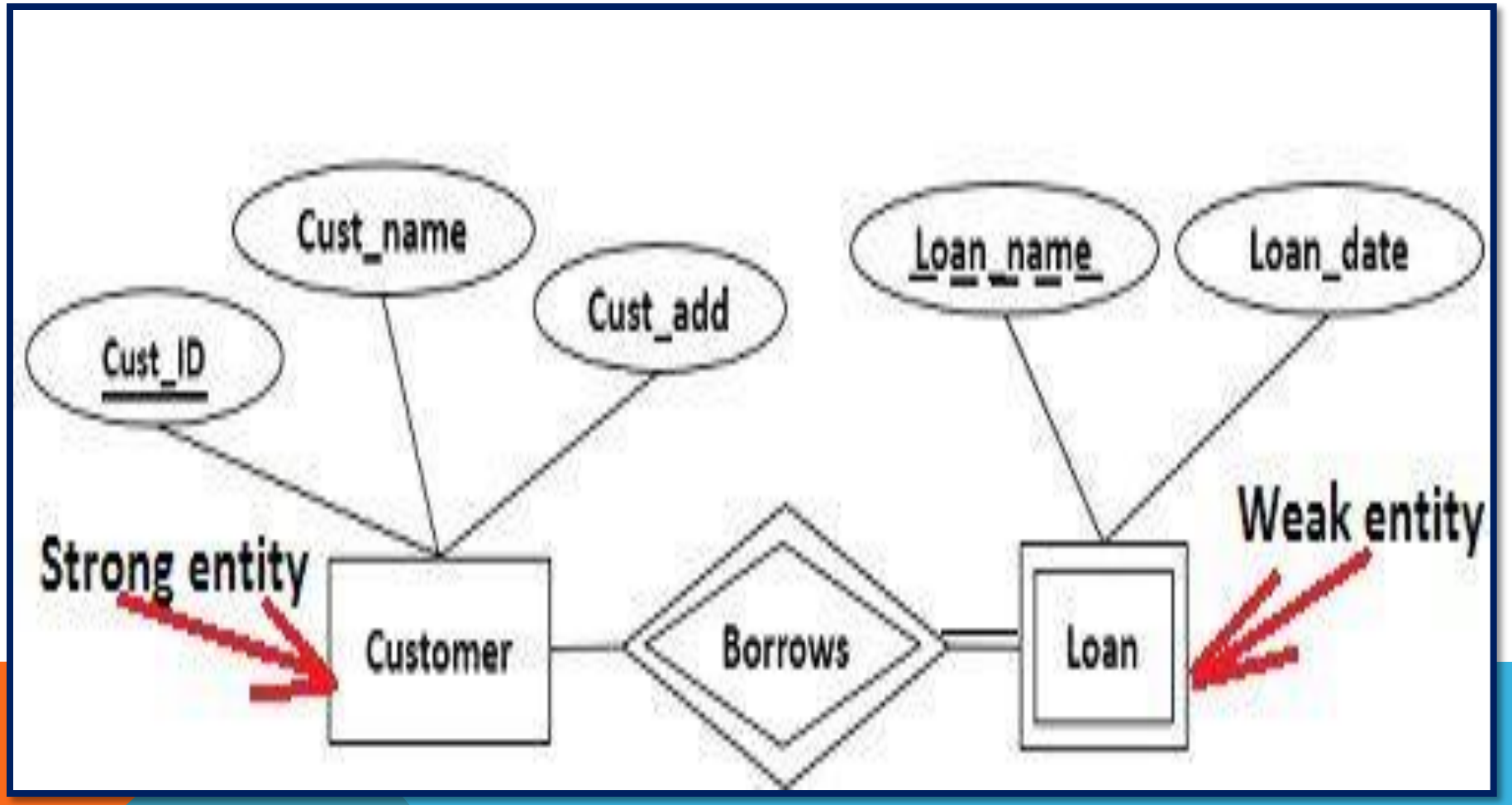


❖ Types of Attributes

1. Simple or Composite Attributes
2. Single Valued or Multivalued Attributes
3. Derived Attribute



❖ Strong And Weak Entity Set



❖ Strong And Weak Entity Set

Weak Entity Sets

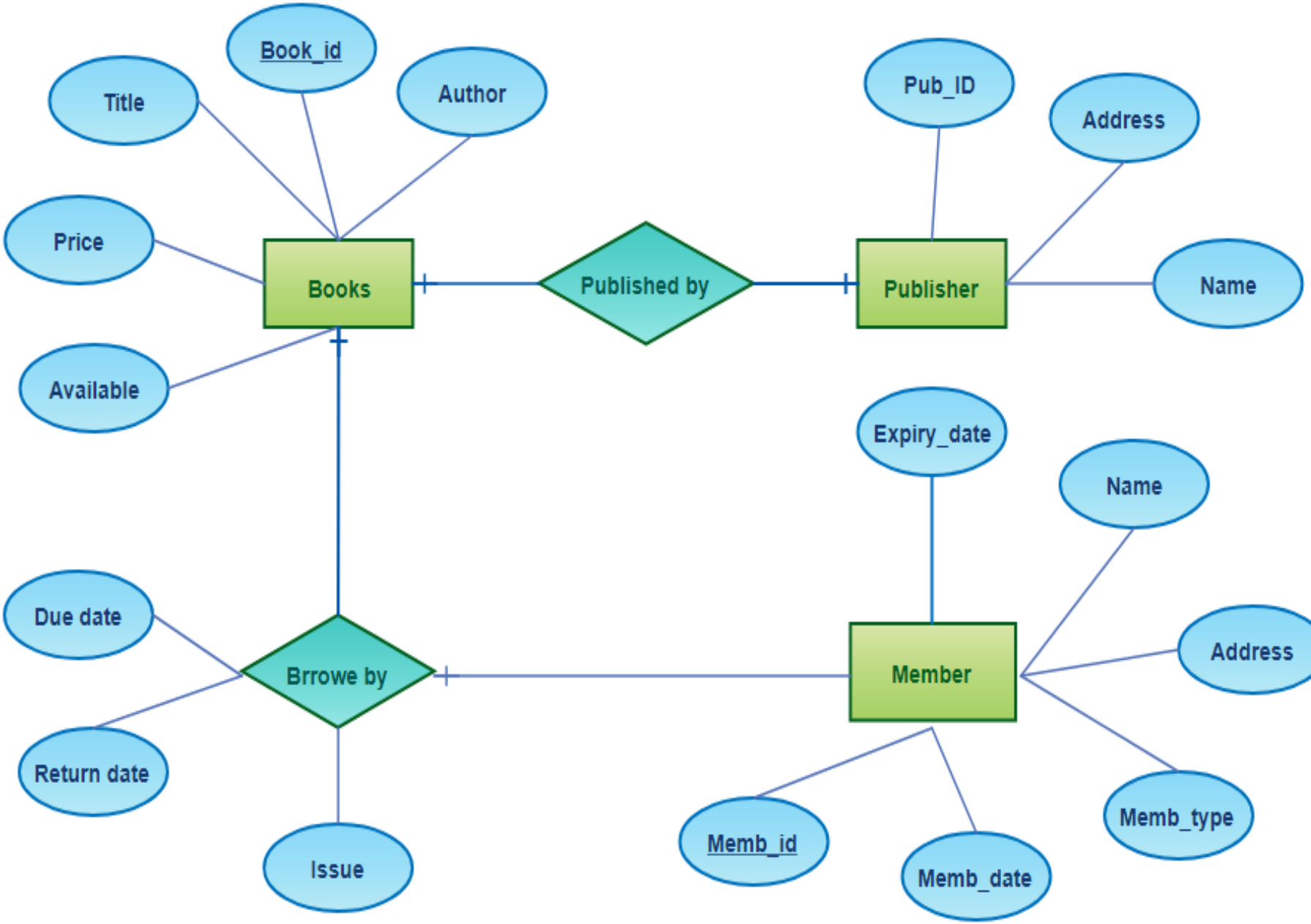
- The entity set that does not have sufficient attributes to define the primary key are called weak entity set.
- The weak entity set is denoted by double Rectangle.

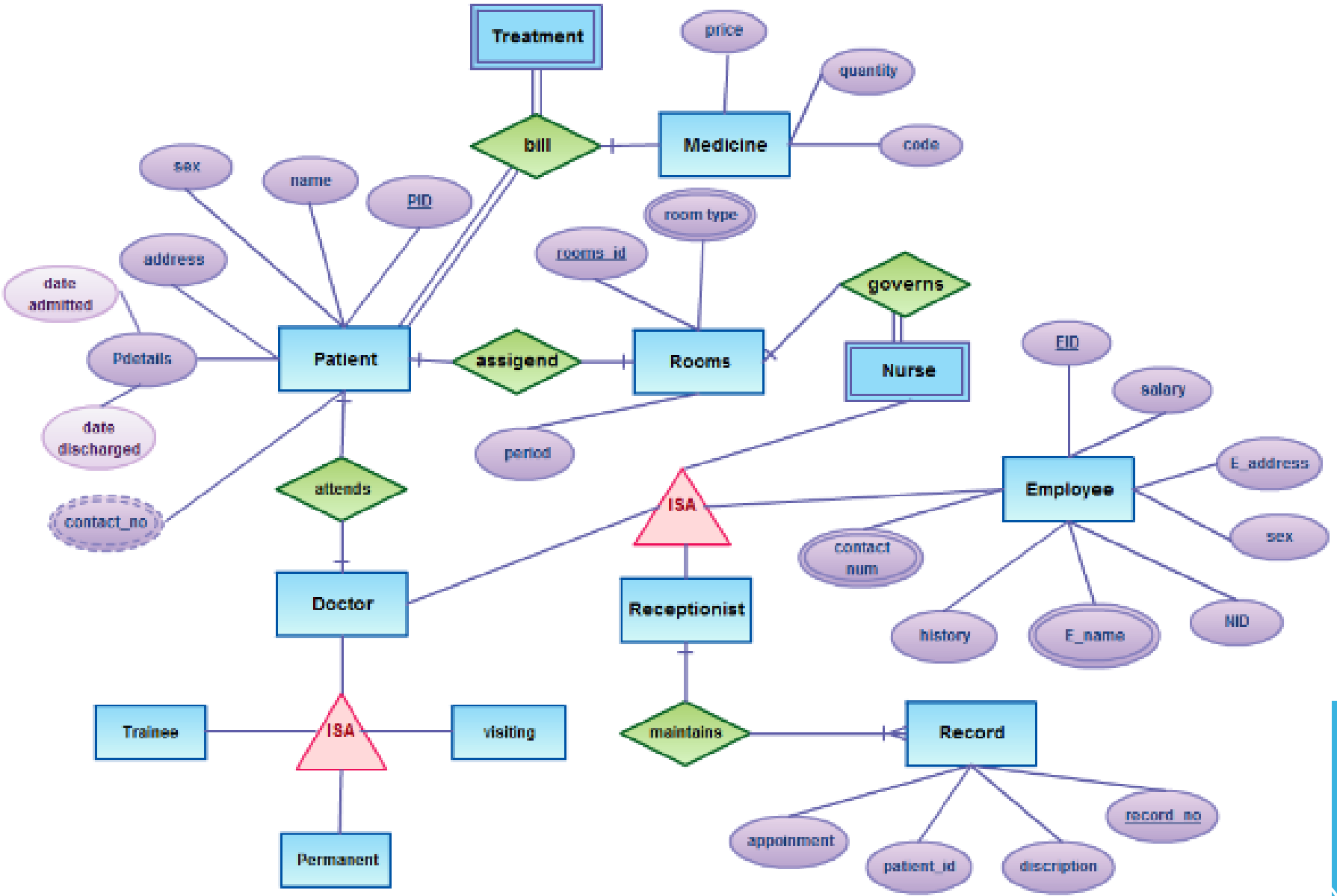
Strong Entity Sets

- An Entity set having sufficient attributes to form a Primary key is called as Strong Entity Set.
- It is also called as Identifying or owner Entity set.
- Relationship among strong Entity Set & Weak Entity Set is called as Identifying Relationship.

❖ Strong And Weak Entity Set

Strong Entity set	Weak Entity Set
The Strong entity has a primary key.	The weak entity has a partial discriminator key.
The Strong entity is independent of any other entity in a schema.	Weak entity depends on the strong entity for its existence.
Strong entity is denoted by a single rectangle.	Weak entity is denoted with the double rectangle.
The relation between two strong entities is denoted by a single diamond simply called relationship.	The relationship between a weak and a strong entity is denoted by Identifying Relationship denoted with double diamond.
Strong entity may or may not have total participation in the relationship.	Weak entity always has total participation in the identifying relationship shown by double line.


Book_id
Title
Author
Pub_ID
Address
Price
Books
Published by
Publisher
Name
Available
Expiry_date
Name
Due date
Brrowe by
Member
Address
Return date
Memb_type
Memb_id
Memb_date
Issue


Treatment
price
quantity
bill
Medicine
code
sex
name
PID
address
room type
rooms_id
date admitted
governs
Pdetails
Patient
assigend
Rooms
Nurse
EID
salary
date discharged
period
attends
ISA
Employee
E_address
contact_no
contact num
sex
Doctor
Receptionist
history
E_name
NID
Trainee
ISA
visiting
maintains
Record
appoinment
patient_id
discription
record_no
Permanent

Difference Between DBMS & RDBMS

DBMS	RDBMS
1. DBMS stands for Data Base Management System.	1. RDBMS stands for Relational Data Base Management System.
2. It is old version of software to handle the database.	2. It is latest version of software to handle the database.
3. It can only relate one table to another table.	3. It can relate one database to another database.
4. Its data security is low as compared to RDBMS.	4. Its level of data security is very high as compared to DBMS.

Difference Between DBMS & RDBMS

DBMS	RDBMS
5. It supports single user only.	5. It supports multiple users
6. Its data storage capacity is low.	6. Its data storage capacity is very high.
7. It treats Data as files internally.	7. It treats Data as tables internally.
8. It supports 3 rules of E.F. Codd .	8. It supports minimum 6 rules of E.F. Codd.
9. It needs low software and hardware requirements.	9. It needs high software and hardware requirements
10. DBMS is used for simpler business applications.	10. RDBMS is used for more complex applications.

THANK YOU!!!!!!!!!!!!