

Database fundamentals

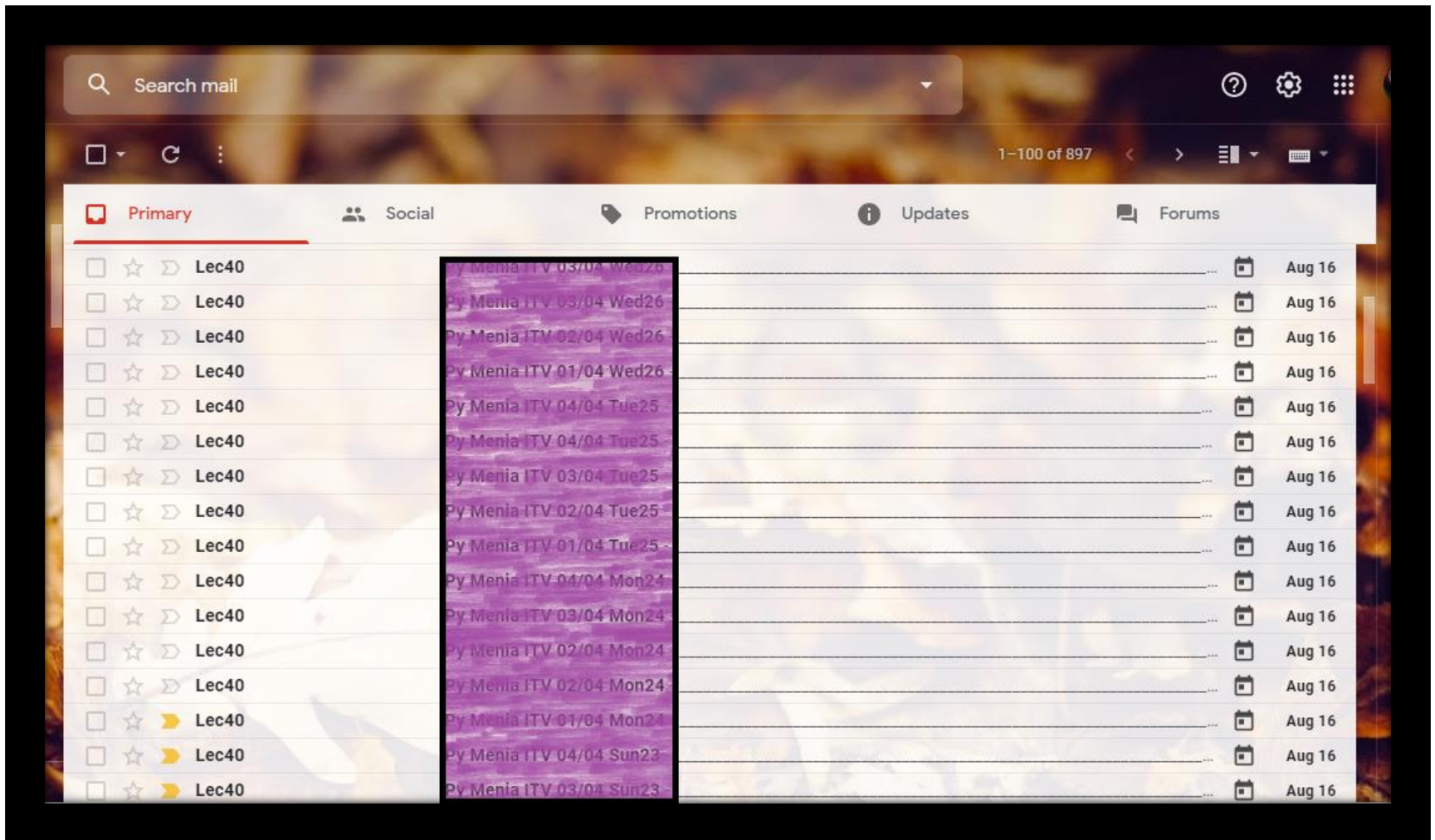
Prepared by

Noha Shehab
Open-source Dept.
ITI

Agenda

- File based system
- Define Database, Database System
- Identify the Database Properties
- Define DBMS
- Functions of DBMS
- Advantages and Disadvantages of Database Systems
- ERD

Example



File Based System

- It is a collection of programs that perform services for the end user.
- Each Program defines and manages its own data



Limitations Of File based System Approach

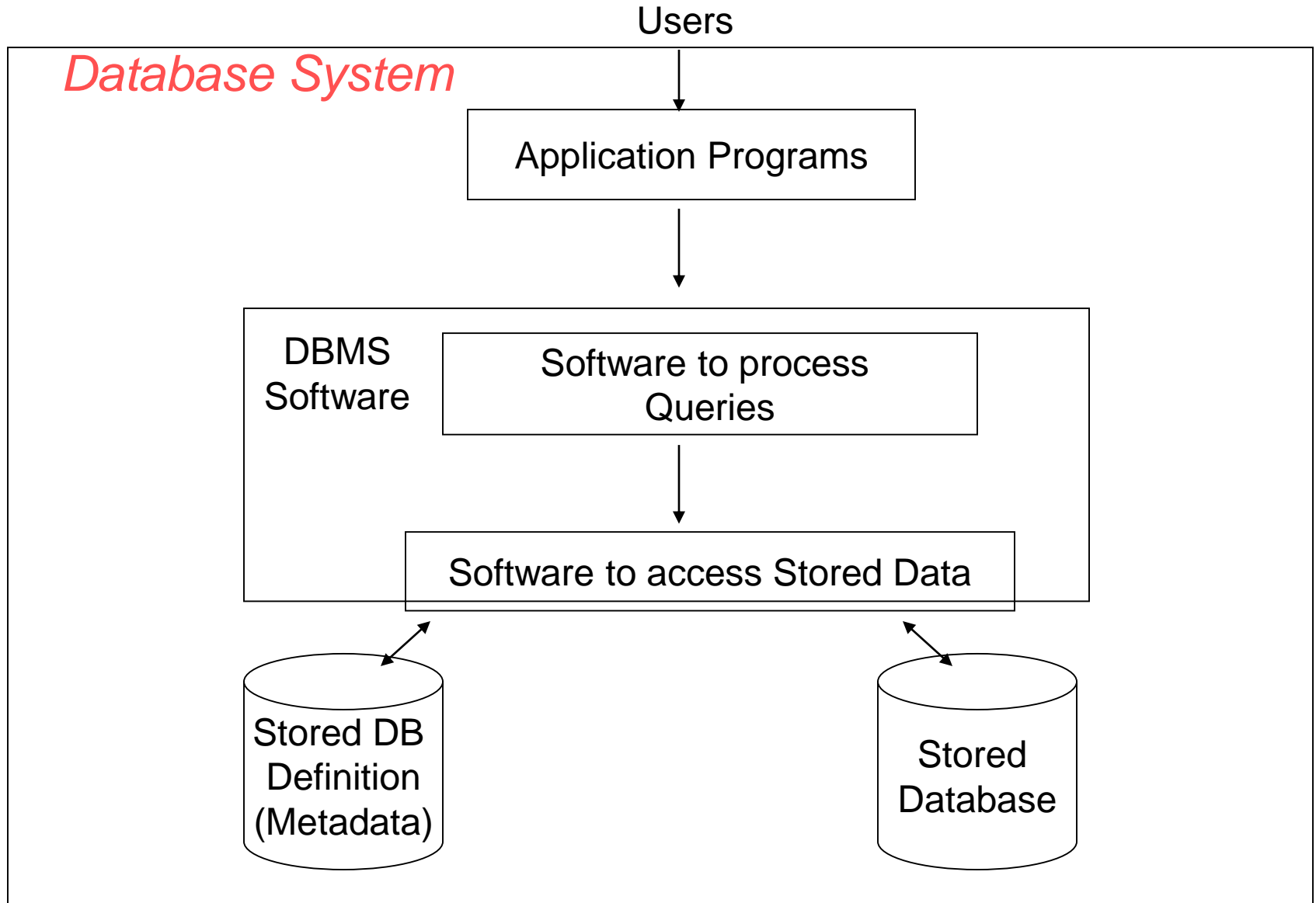
- Separation & Isolation Of data
- Duplication Of data
- Program Data Dependence
- Incompatible File Formats



Basic Definitions

- **Database:** A collection of related data.
- **Database Management System (DBMS):** A software package/system to facilitate the creation and maintenance of a computerized database.
- **Database System:** The DBMS software together with the data itself. Sometimes, the applications are also included. (Software + Database)

Database Management System (DBMS)



DBMS Advantages

- **Controlling Redundancy.**
- **Restricting Unauthorized Access.**
- **Sharing data.**
- **Enforcing Integrity Constraints.**
- **Inconsistency can be avoided.**
- **Providing Backup and Recovery.**

DBMS Disadvantages

- **DBMS is expensive**
- **May be incompatible with any other available DBMS**



Database Users

Database Administrator
(DBA)

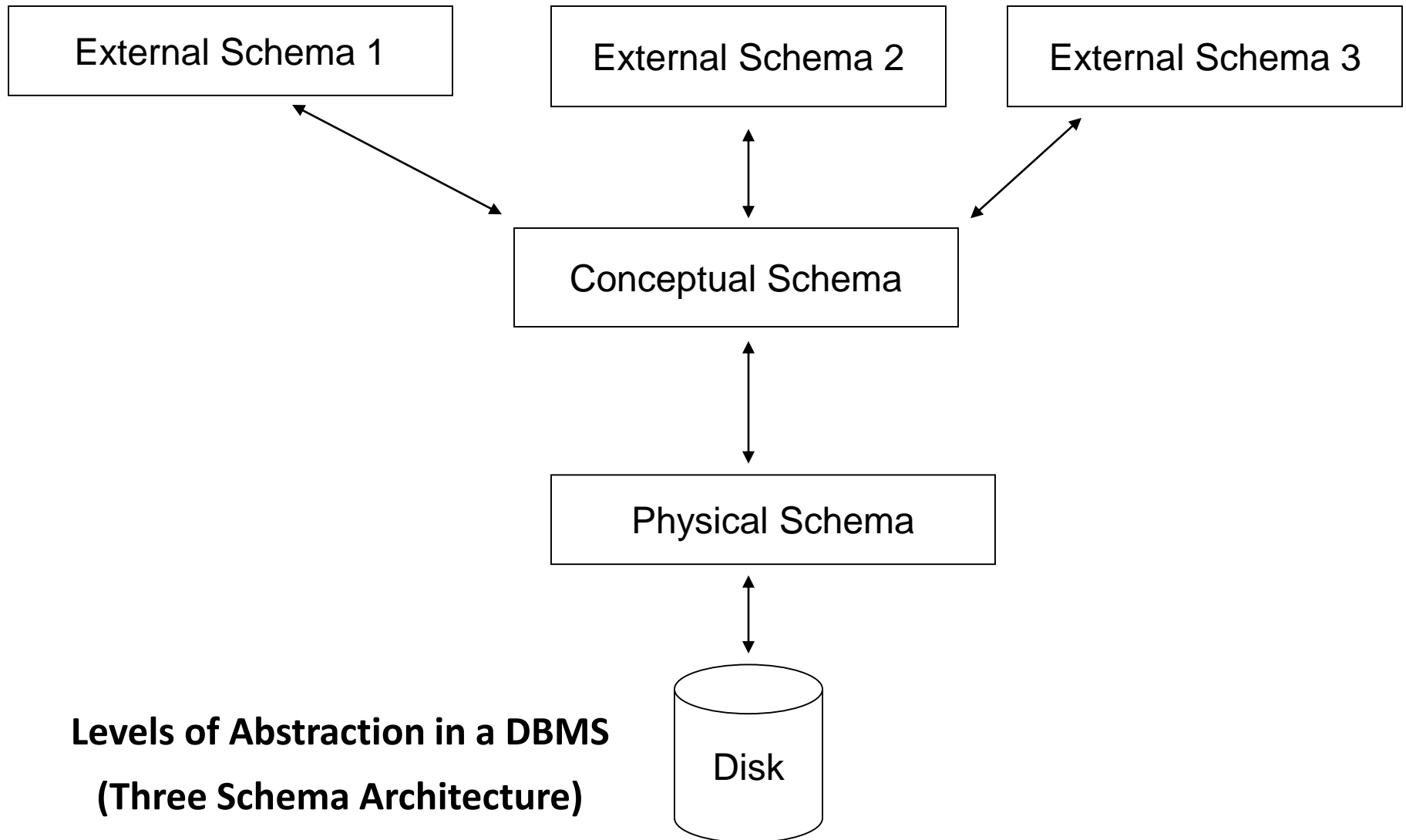
System Analysts

Database Designer

Application programmers

End users

DBMS Architecture



Three Level/Schema Architecture

- **External – What the user sees:** focus on what different types of users will see when viewing the database. They are concerned with what data the user will see and how the data will be presented to the user.
- **Conceptual – The logical model:** focus on the logical nature of the data representation. They are concerned with what is represented rather than how it is represented.(define database structures such as tables and constraints)
- **Internal – The physical model:** place the emphasis on how the data are represented in the database or on how the data structures are implemented.

Mappings

- **Definition:** It is the processes of transforming requests and results between levels.
- These mappings may be time-consuming. However, a certain amount of mapping between the conceptual and internal levels is necessary.

Data Independence

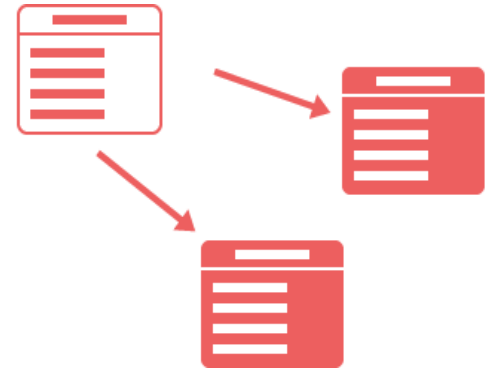
- **The capacity to change the schema at one level without having to change the schema at the next higher level.**

Data Models

- **High Level or Conceptual data models provide concepts that are close to the way many users perceive data, entities, attributes and relationships. (Ex. ERD)**
- **Physical data models describes how data is stored in the computer and the access path needed to access and search for data.**

Entity Relationship Modeling

- **Entity-Relationship Diagram (ERD):** identifies information required by the business by displaying the relevant entities and the relationships between them.



Entity Relationship Modeling

- In building a data model a number of questions must be addressed:
 - What entities need to be described in the model?
 - What characteristics or attributes of those entities need to be recorded?
 - Can an attribute or a set of attributes be identified that will uniquely identify one specific occurrence of an entity?
 - What associations or relationships exist between entities?

Entities

- **Entity - An entity is a thing in the real world with an independent existence. Physical existence (for example, a particular person, car) or conceptual existence (for instance, a job, or a university course). Types of entities: Weak- Regular**

Entities

- Has its own identity that distinguishes it from other entities.
- Examples:
 - Person: PROFESSOR, STUDENT
 - Place: STORE, UNIVERSITY
 - Object: MACHINE, BUILDING
 - Event: SALE, REGISTRATION
 - Concept: ACCOUNT, COURSE

Entities

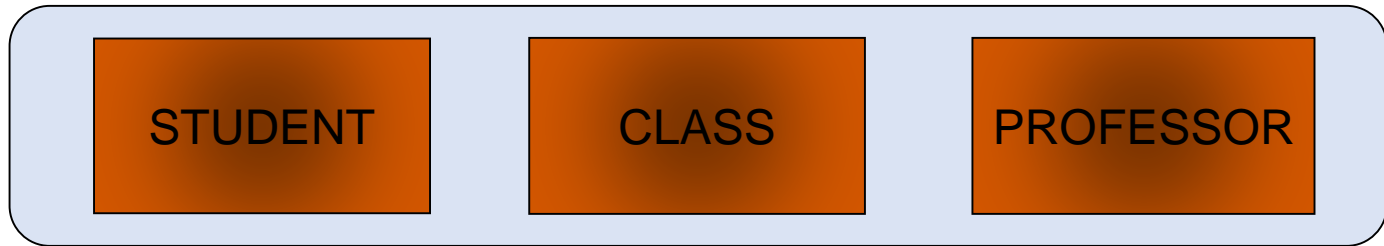
- **Entity Instance** - An instance is a particular occurrence of an entity. For example, each person is an instance of an entity, each car is an instance of an entity, etc.

Entities

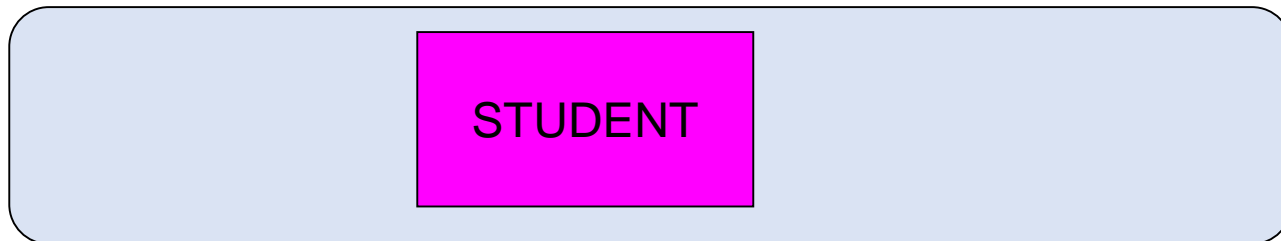
- Entity type name should be:
 - A singular noun and in capital letters.
 - Descriptive and specific to the organization.
 - Concise.
 - Named for the result of the event, not the activity or process of the event.

Entities

- **Entity Type** is a collection of entities that share common properties or characteristics.



- **Entity Instance** is a single occurrence of an entity type.



Entities should always be placed in a rectangle!

Weak Entity Types

An entity that does not have a key attribute

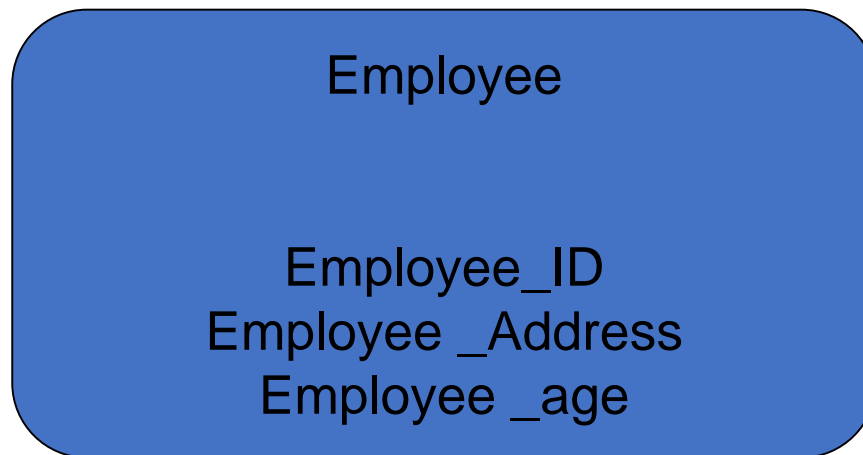
A weak entity must participate in an identifying relationship type with an owner or identifying entity type

Entities are identified by the combination of:

- A partial key of the weak entity type
- The particular entity they are related to in the identifying entity type

Attributes

- **Attribute** - The particular properties that describe the entity. An EMPLOYEE entity may be described by the employee's name, age, address and salary attributes.



Types of Attributes

Key: an attribute whose values are distinct (unique) for each entity and can be used to uniquely identify the record

Multi-valued: has a set of values for the same entity instance

Composite: can be divided into smaller subparts

Derived: can be calculated from another attribute or entity

Single/Simple: Attributes that are not divisible and have a single value for a particular entity instance

Key Attribute

- **Single Key:** For example, SSN of EMPLOYEE.
- **Composite Key:** the combination of the attribute values that together form a key and must be distinct for each entity. For example, ID and Application_no.
- **Candidate Key:** when an entity type has more than one key, those are candidate keys.

Relationships

Relationships - A relationship is a connection between entity classes.

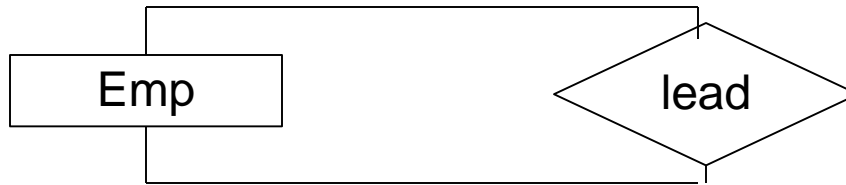
Degree of a Relationship: is the number of participating entity .

Cardinality Ratio: specifies the maximum number of relationship.

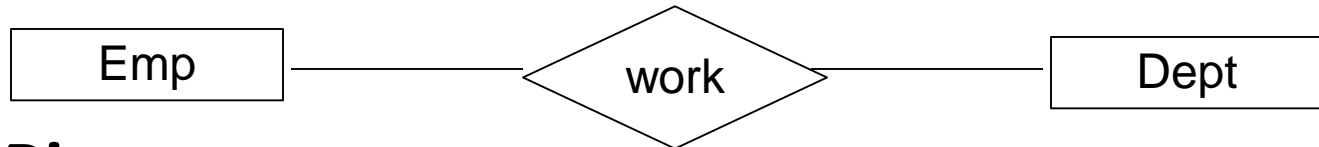
Participation: specifies the minimum number of relationship instances that each entity can participate with.

Relationships

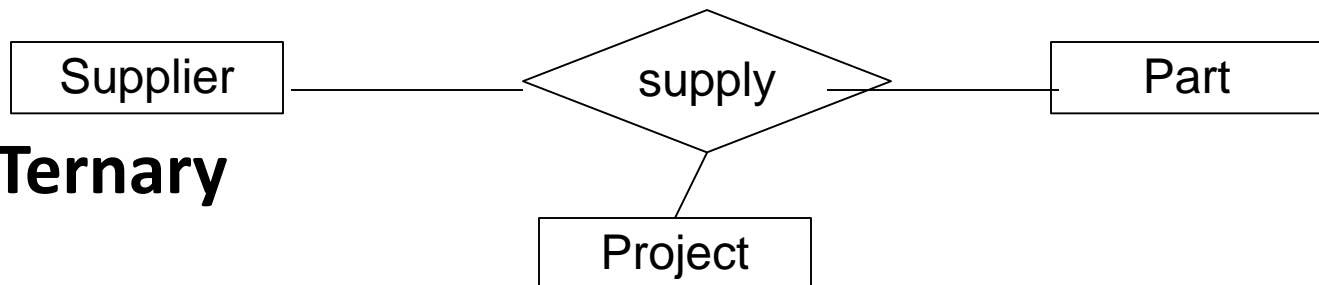
- **1.Degree of a Relationship**



- **a. Unary/ Recursive**



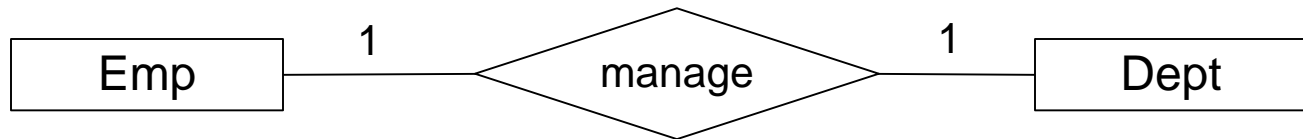
- **b. Binary**



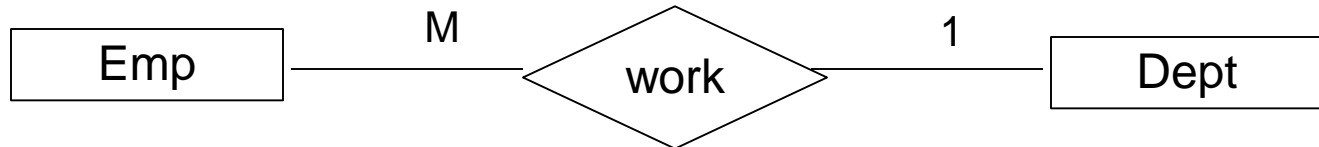
- **c. Ternary**

Relationships

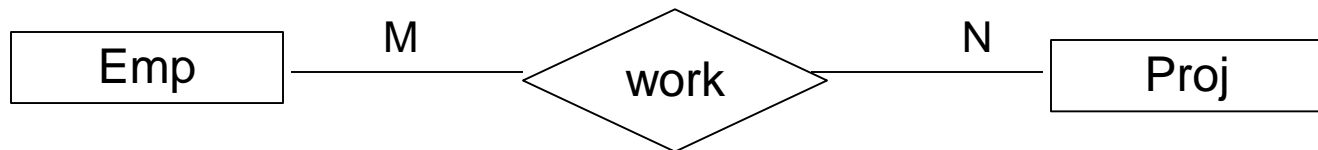
- **2. Cardinality Ratio**



- **a. One to one**



- **b. One to many**



- **c. Many to many**

Relationships

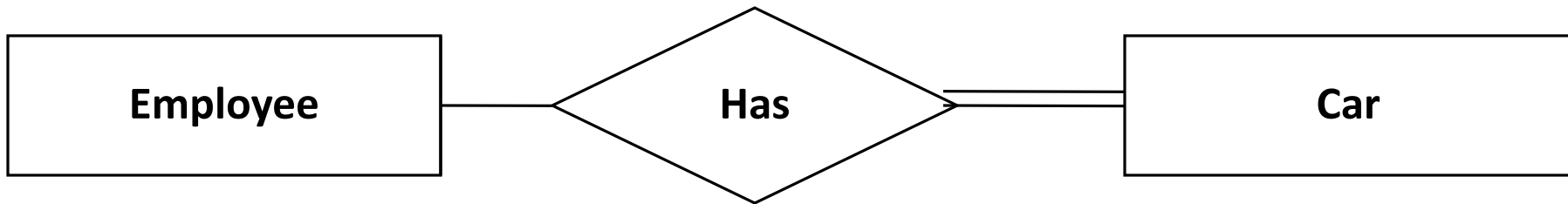
- 2. Participation
 - a. Total/ Full Dependency/ Mandatory
 - b. Partial/ Partial Dependency/ Optional



Partial

Total

Participation Example



- - An Employee may have a car.
- - A Car must be assigned to particular employee

Participation Example



- A department may hire many employees (Zero or more)
 - - An employee must be employed by a department
- (Department membership is Optional, Employee membership is Mandatory)














ERD Notations

- Rectangles represent ENTITY CLASSES
- Circles represent ATTRIBUTES
- Diamonds represent RELATIONSHIPS

Arcs - Arcs connect entities to relationships. Arcs are also used to connect attributes to entities. Some styles of entity-relationship diagrams use arrows and double arrows to indicate the one and the many in relationships. Some use forks etc.

- Underline - Key attributes of entities are underlined.

SUMMARY OF ERD NOTATION

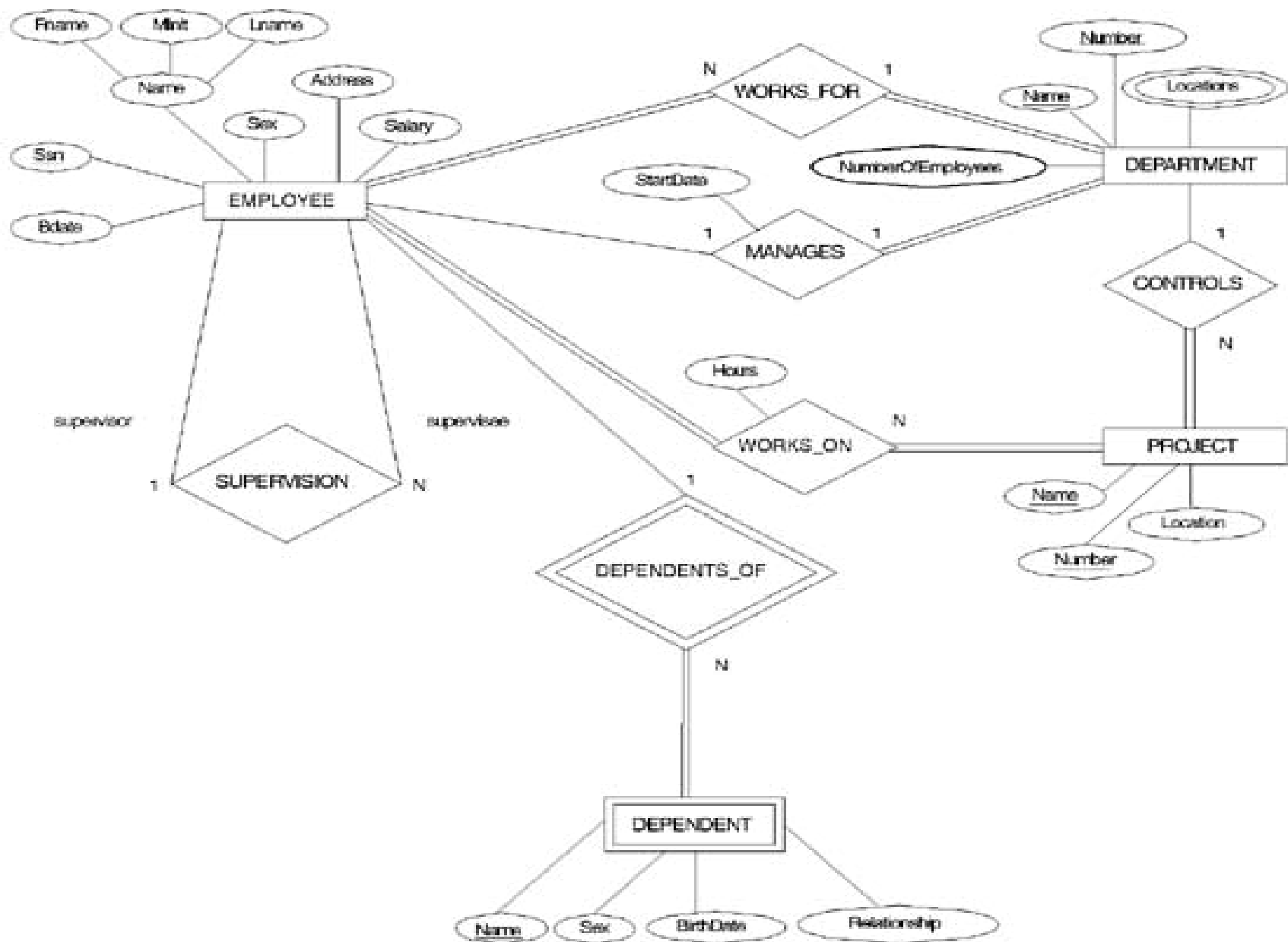
Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF E_2 IN R
	CARDINALITY RATIO 1:N FOR $E_1:E_2$ IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R
	

An Example

- A company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. A department may have several locations.
- A department may control a number of projects, each of which has a unique name, a unique number, and a single location. A project must be controlled by a department

An Example (Cont'd)

- We store employee's name, social security number, address, salary, gender and birth date. An employee must be assigned to one department and must work on one or more projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, gender, birth date and relationship to that employee.



ERD Case Study

- An organization makes many models of cars, where a model is characterized by a unique name and a suffix (such as GL or XL) and an engine size.
- Each model is made up from many parts and Each
 - part has a description , an id code, production year, and many images.
- each part may be used in the manufacturing of
 - more than one model

ERD Case Study

- Each model must be produced at just one of the firm's factories, which are located in London, Birmingham, Bristol, Wolverhampton and Manchester - one in each city. Each factory has number of machines, capacity, and computer system used (OS , DBMS, Internet).
- A factory produces many models of cars and many types of parts.

ERD Case Study 2

- A country bus company owns a number of buses. A **bus** is characterized by number, No. of Chairs, Options (AC , Automatic, PS) , and brand-name
- Each bus is allocated to a particular route, although some routes may have several buses
- Each route is described by KM, start point, end point and the duration.

ERD Case Study 2

- Each route can pass through a number of towns.
- A town may be situated along several routes. We keep track of unique name and station names in each town.
- One or more drivers are allocated to one route during a period of time. The system keeps information about the driver name, mobile number, hire date, basic salary, job grade.
- The system keeps information about any changes in the allocations of the drivers to the routes.
-

ERD Narrative: Lab

- **A database for a banking system is used to control withdrawal, deposit and loan transactions with customers.**
- **Banks which use this system have many branches; each branch has a unique name, unique address and phone.**
- **The system stores information about customers as unique customer ID, name, address, and phones.**

ERD Narrative (Cont'd)

- Each customer has one Account identified by unique Account number, amount, last transaction date (Day, Month and Year).
- The system records Transaction number, Transaction type, Transaction date, Transaction amount and time. The system records the branch name where the transaction occurred.
- A Customer can make any type of transactions (Withdrawal or Deposit) from any branch of the bank.