Database Fundamentals

Lecture 2
Entity Relationship Diagram (ERD)

Prepared By: Eng. Doaa Soleiman

Lecture 2
Entity Relationship Diagram (ERD)

1- Database Design Phases

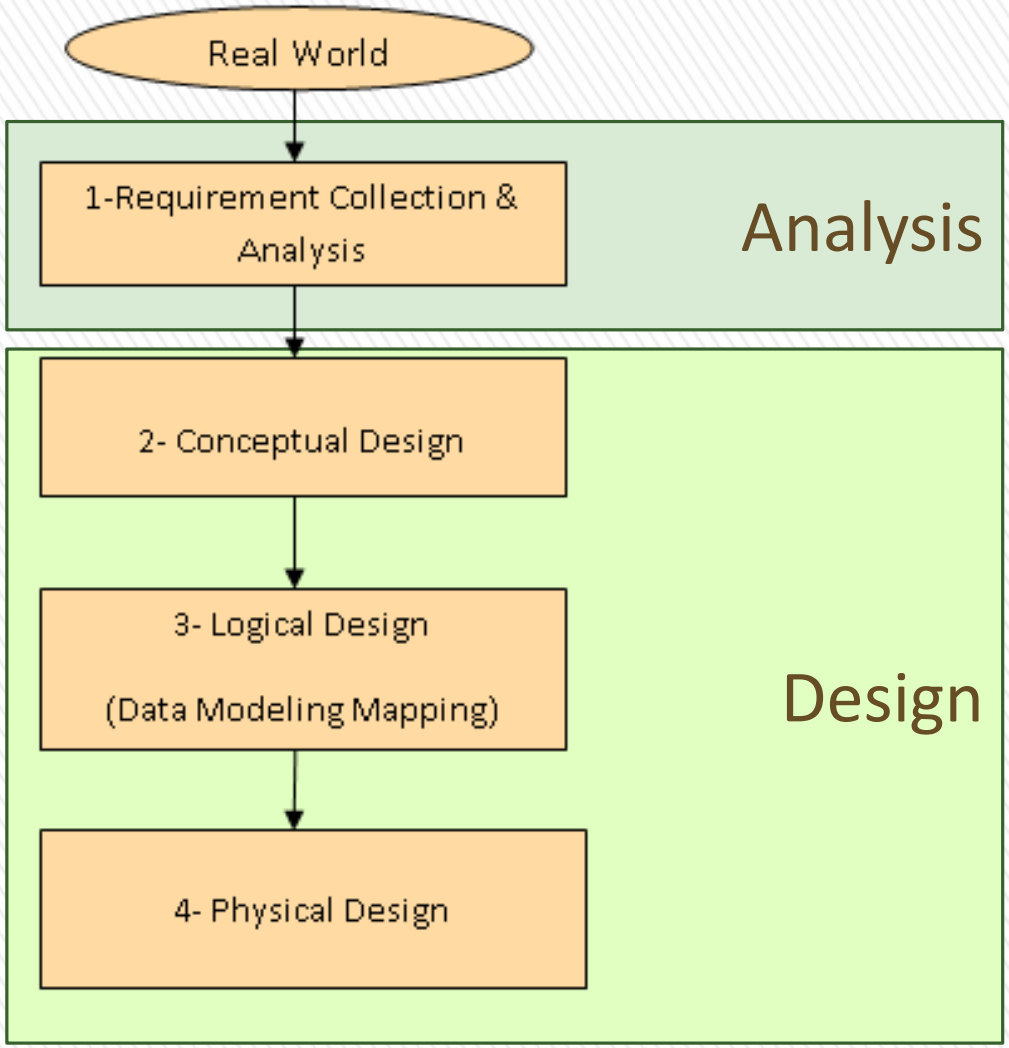
2 – ER – Model

3- Relationships

4- ERD notation

5- Example

Database Design Phases




1- Conceptual Design

Conceptual Design

- All requirements have been collected and analyzed from analysis phase.
- It called "**High level conceptual model**" which means, High Level or Conceptual data models provide concepts that are close to the way many users perceive data, entities, attributes and relationships. (Ex. **ERD**)

Entity Relationship Modeling

- Identifies information required by the business by displaying the relevant entities and the relationships between them.
- It is first proposed by Peter Chen at 1970.
- **ERD model** is often created graphically and specific software converts the graphical representation into SQL code that is required to create data structure

Entity

- Entity - An entity is a thing that exists and is distinguishable -- an object, something in the environment.
  - Types of entities: **Weak**- **Regular**

- 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.

Entity

- Regular entity:

It is represented as 

- Example: Employees, books, products, …. etc.

Weak Entity

- An entity that does not have a key attribute
- Its existence depends on some other entity.
- 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

Example: Relatives of employee

Attributes

- It is the entity properties.
- It is represented as 
- Each attribute has its own Data Type
- The attribute may have default value
- Null Attribute: The attribute that may accept NULL values

Key Attributes

- **Primary Key** is a unique identifier for the entity, it may be single or composite, It must be UNIQUE and NOT NULL.
- Each entity has only ONE Primary Key, it is one of the candidate keys.

  Example: SSN of EMPLOYEE.
- A key attribute may be composite.

  Example: (National_ID, Application_no)

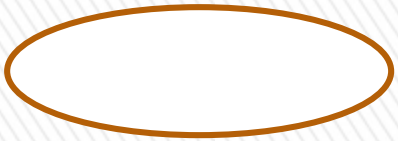
Candidate Key / Foreign Key

- Each entity has at least one **Candidate Key**, and may have more than one.
- **Foreign Key** is an attribute in one entity whose values are required to match those values of the primary key of some other entity.

Simple / Composite Attribute

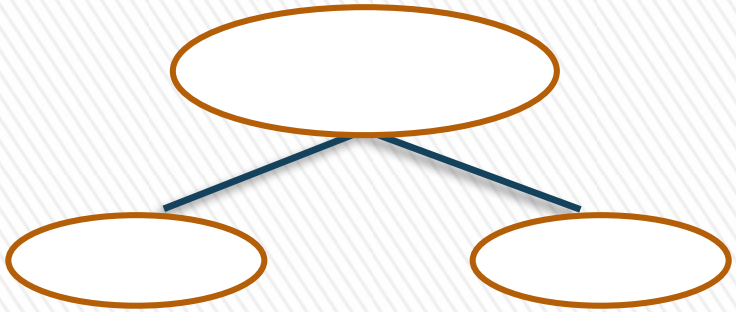
- **Simple attribute:** means it can't be

  Example: salary , grade, ….



- **Composite attribute:** means it can be divided into subparts.

  Example: Address can be divided into streetno, streetname, zone, city



Single / Multi-value Attribute

- **Single attribute:** means it has one value for particular entity

Example: name (employee have one name only)

- **Multi-value attribute:** means it has a set of values for the same entity

Example: Telephone (The same employee may have more than one telephone number)

Base/ Derived Attribute

- **Base attribute:** means that its value can't be deduced or calculated. It is represented as

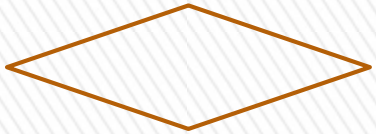
  Example: Birthdate 

- **Derived attribute:** means that its value can be deduced or calculated. It is represented as

  Example: Age 

Relationships

- Relationship is an association between two or more entities represented as



Cardinality Ratio Constraints

The cardinality of a relationship indicates the number of instances in entity class E1 that can or must be associated with instances in entity class E2.

Cardinality Ratio Constraints (Conti.)

- **One-One Relationship**

  (citizen , passport) , (Dept. , Manager)

- **One-Many Relationship**

  (student-Advisor, Customer-Order , Dept. , Employees)

- **Many- Many Relationship**

  (Student-Organization, Order-Products, Student – courses)

- **Recursive Relationships** A relationship in which the same entity participates more than once.

  (Employee – Manager)

Membership class constraint

- **Obligatory "Total Participation"**

  If every E occurrence MUST participate in R relation occurrence

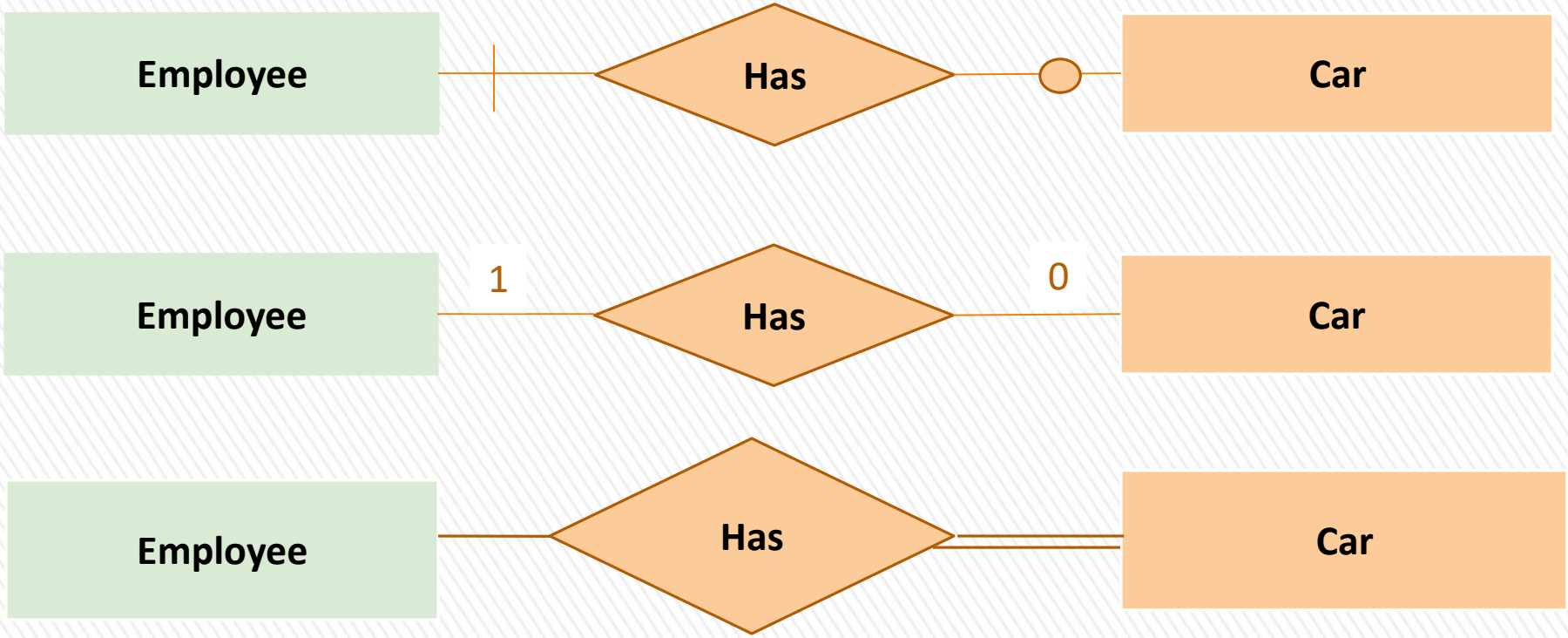
  Example: An employee MUST work for a department

- **Non Obligatory " Partial Participation"**

  If E can exists without participating in R relation occurrence

  Example: Some employees manage departments

PARTICIPATION CONSTRAINT



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

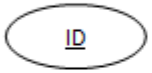
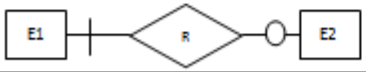
PARTICIPATION CONSTRAINT

A formal constraint: (min,max) where m, n are min and max number of times an entity participates in a relationship instance.

For example,

(0,10) means partial participation, and (1,max) means total participation.

ERD notation

	Regular Entity
	Weak entity
	Base , simple attribute
	Relationship
	Relationship with weak entity
	Derived attribute
	Key Attribute
	Composite Attribute
	One to One
	One to Many
	Many to Many
	Membership class constraint
	Relation Attribute

Example

- A company is organized into departments. Each department has a unique name, a unique number, and an 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 controlled by department

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.
- We need to store information about the number of hours per week that an employee works on each project. We also store the direct supervisor of each employee.
- We want to store the information of the dependents for each employee for insurance purposes. We keep each dependent's first name, gender, birth date and relationship to that employee.

Example

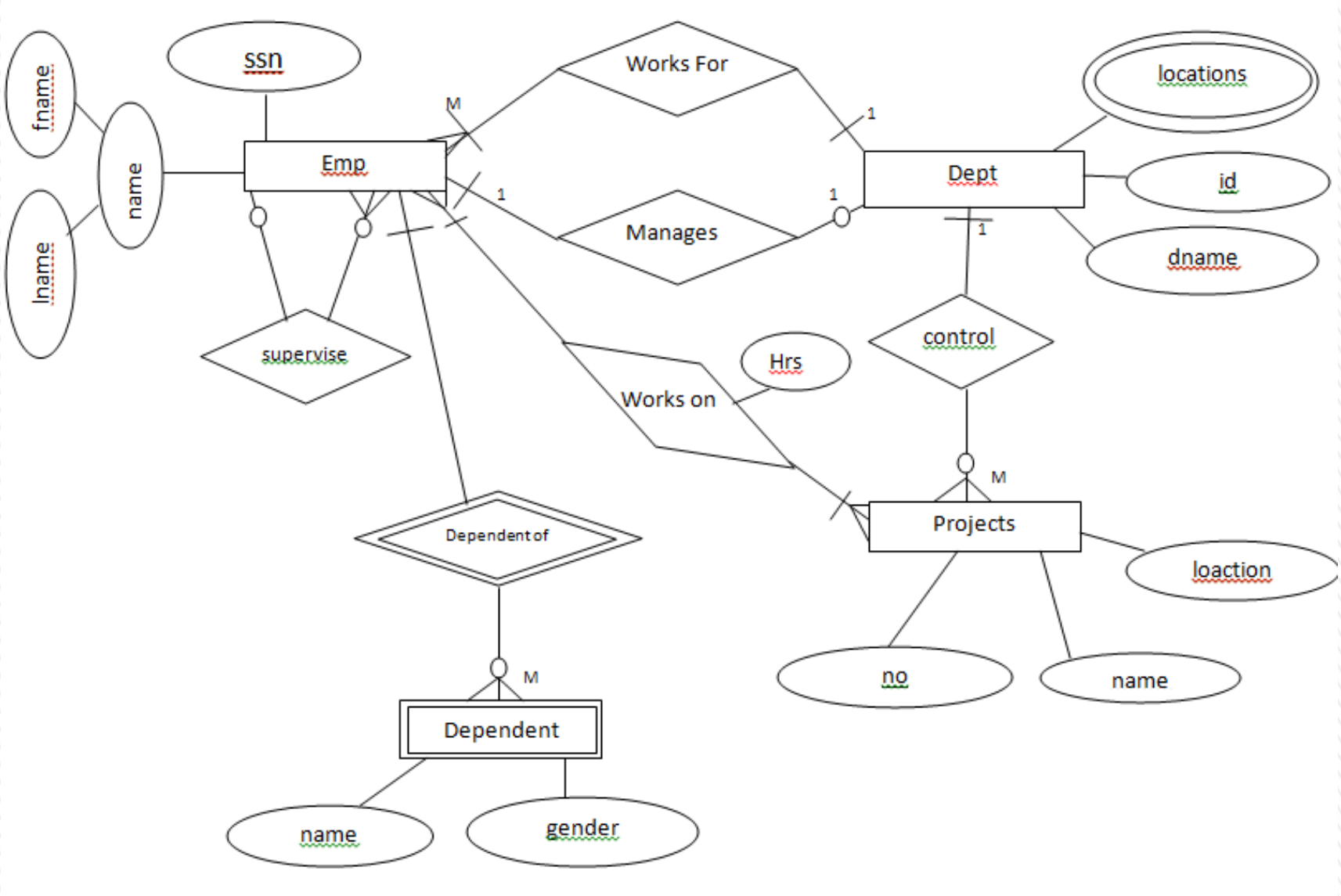
- A company is organized into **departments**. Each department has a **unique name**, a **unique number**, and a particular **employee who manages the department**. A department may have several **locations**.

- 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 controlled by department**

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**
- 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.

Solution


fname
name
lname
ssn
Emp
Works For
Manages
M
1
1
1
Dept
locations
id
dname
1
control
supervise
Hrs
Works on
M
Projects
loaction
Dependent of
M
Dependent
name
gender
no
name

ERD Case Study 1

- 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 1

- Each model must be produced at just one of the firm's factories, which are located in London, Birmingham, Bristol, Wolver Hampton 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 passes 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 keep information about the driver name , mobile number , hire date, basic salary , job grade.
- The system keep information about any changes in the allocations of the drivers to the routes.

Mapping

- 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.

Questions ?