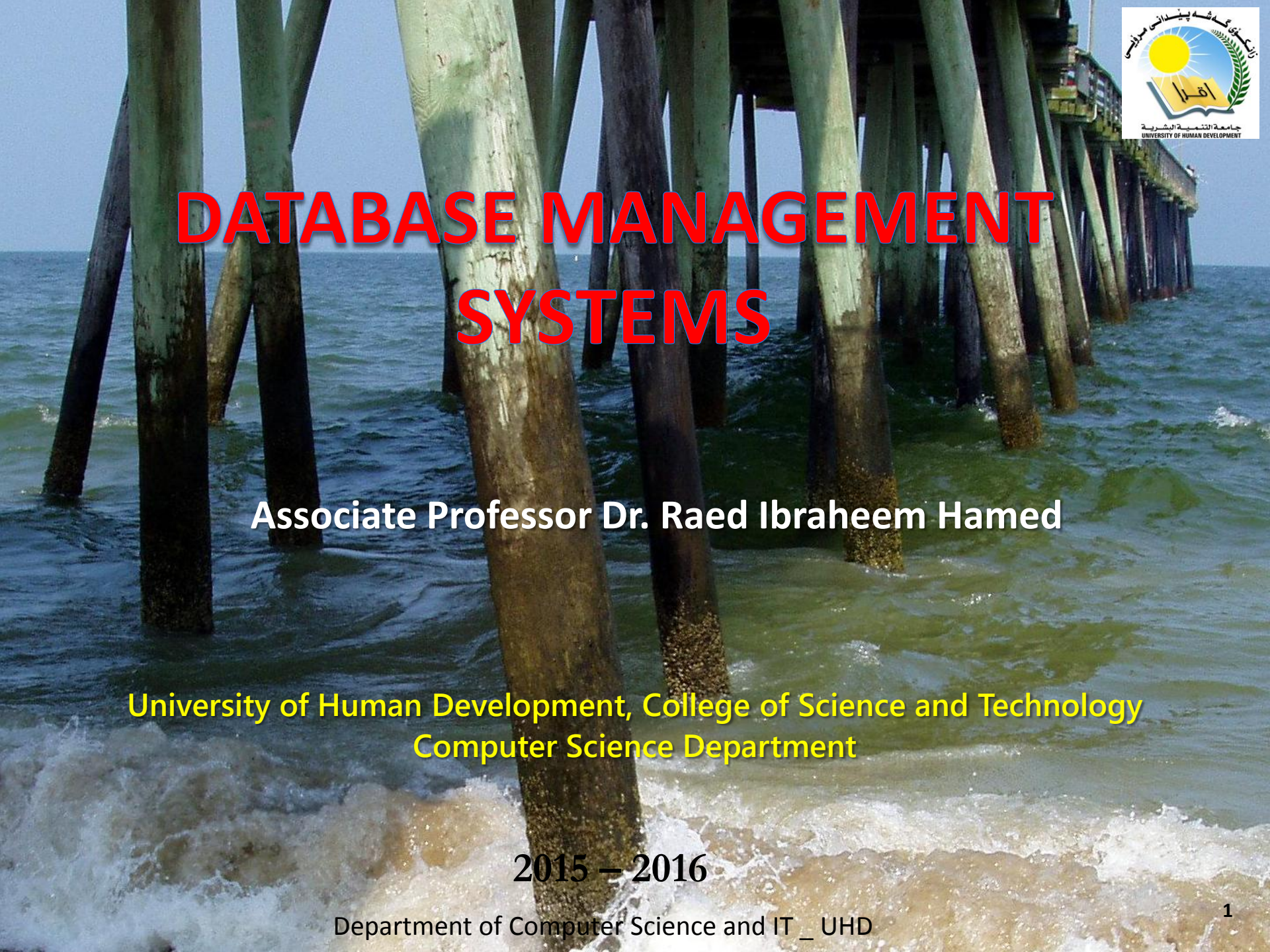


DATABASE MANAGEMENT SYSTEMS

Associate Professor Dr. Raed Ibraheem Hamed

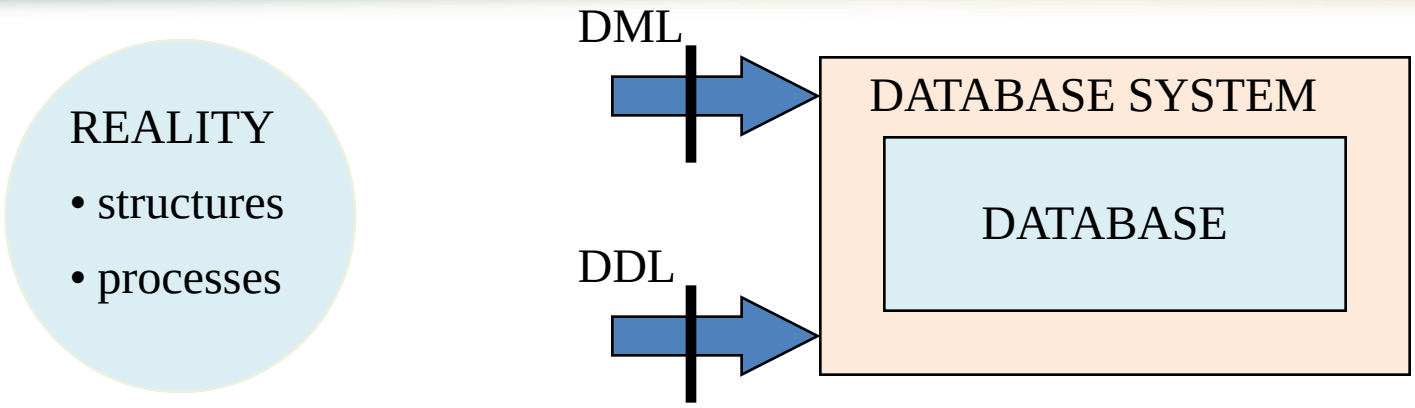
University of Human Development, College of Science and Technology
Computer Science Department

2015 – 2016

Points to Cover

- Models of Reality
- Why Use Models?
- A Map Is a Model of Reality
- A Message to Map Makers
- Use a DBMS when this is important
- Data Modeling
- Process Modeling
- Database Design
- Abstraction

Database as a Model of Reality

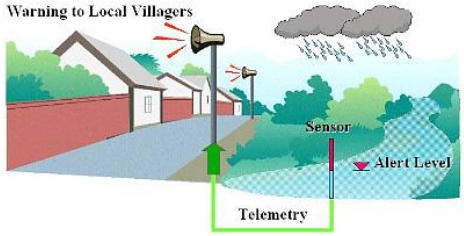


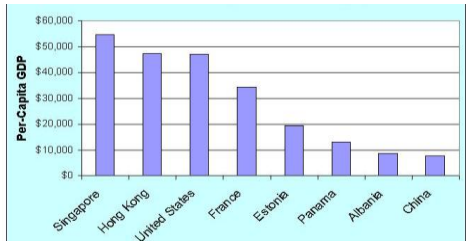
A database is a model of reality. It represents a set of views, held by a community for users, on how they retrieve and process information. This model should be accurate and at the same time simple.

- DDL: Data Definition Language
- DML: Data Manipulation Language

Why Use Models?

- Models can be useful when we want to examine or manage part of the real world
- The costs of using a model are often considerably lower than the costs of using or experimenting with the real world itself
- Examples:
  - **airplane simulator**
  - **nuclear power plant simulator**
  - **flood warning system**
  - **model of US economy**
  - **map**





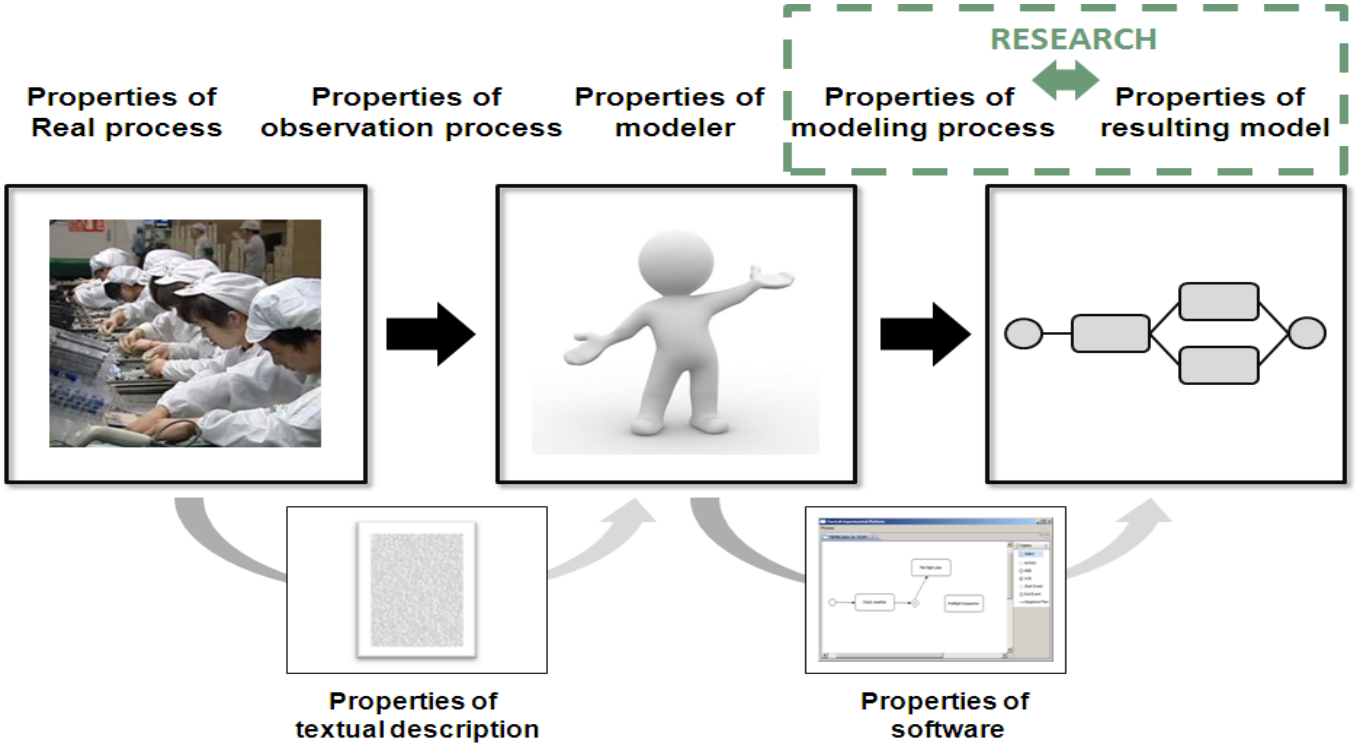
A Map Is a Model of Reality

A Message to Map Makers

- A model is a means of communication
- Users of a model must have a certain amount of knowledge in common
- A model on emphasized selected aspects
- A model is described in some language
- A model can be with error
- A message to map makers keep all the details.

Process Modeling

Process modeling is a technique for organizing and documenting the structure and flow of data through a system's processes.



Database Design

The purpose of database design is to create a database which

- → is a model of structures of reality
- → supports queries and updates modeling processes of reality
- → runs efficiently

Data Abstraction

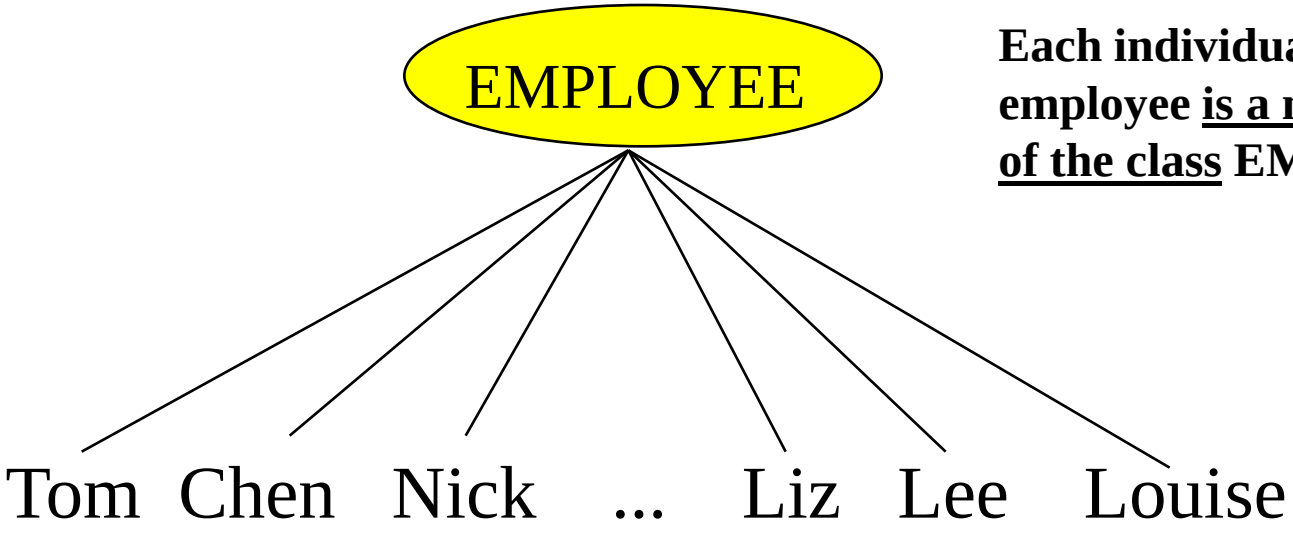
Data abstraction is the reduction of a particular body of **data** to a simplified representation of the whole. **Abstraction**, in general, is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics.

We will discuss three kinds of abstraction:

- Classification
- Aggregation
- Generalization

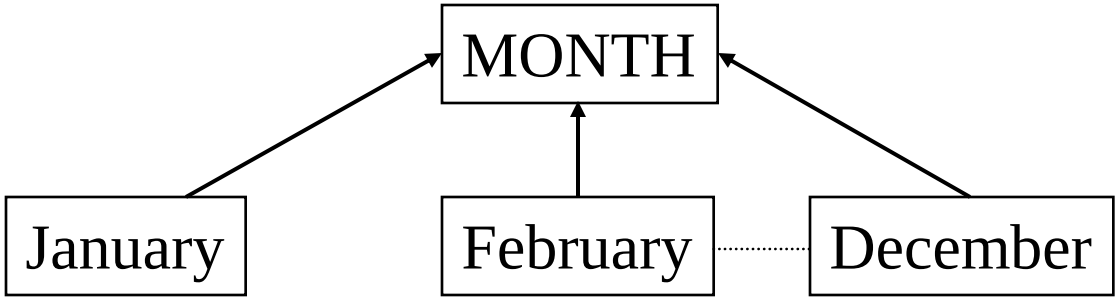
Classification Abstraction

In a classification we form a concept in a way which allows us to decide whether or not a given phenomena is a member of the extension of the concept.



Each individual employee is a member of the class EMPLOYEE

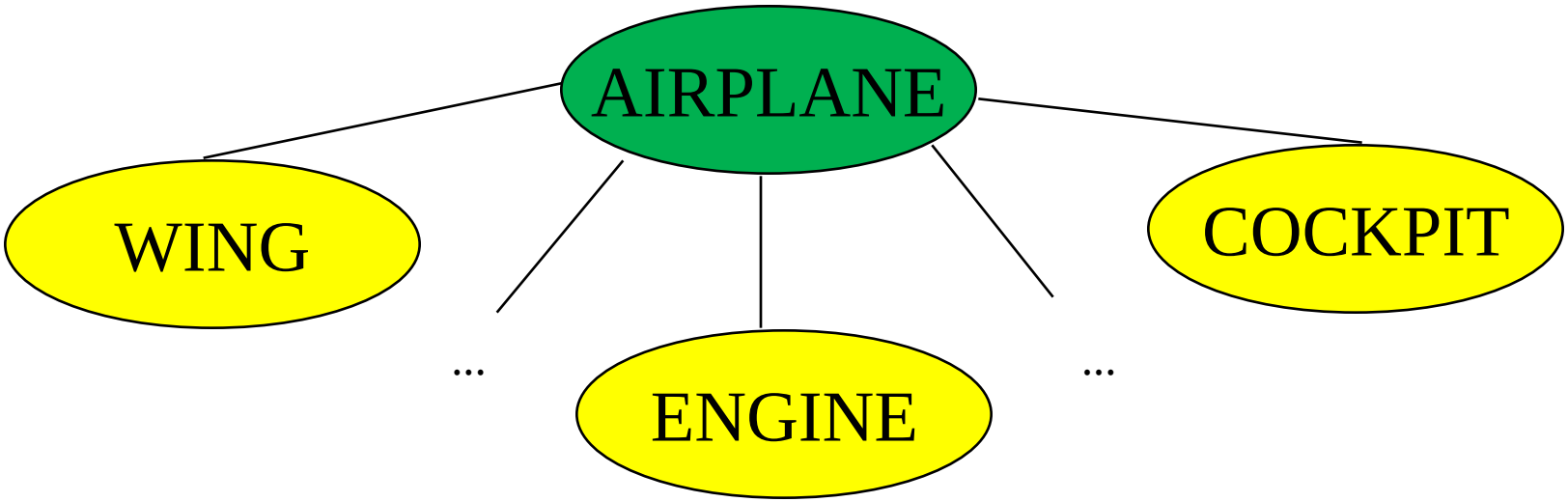
Classification Abstraction (contd.)



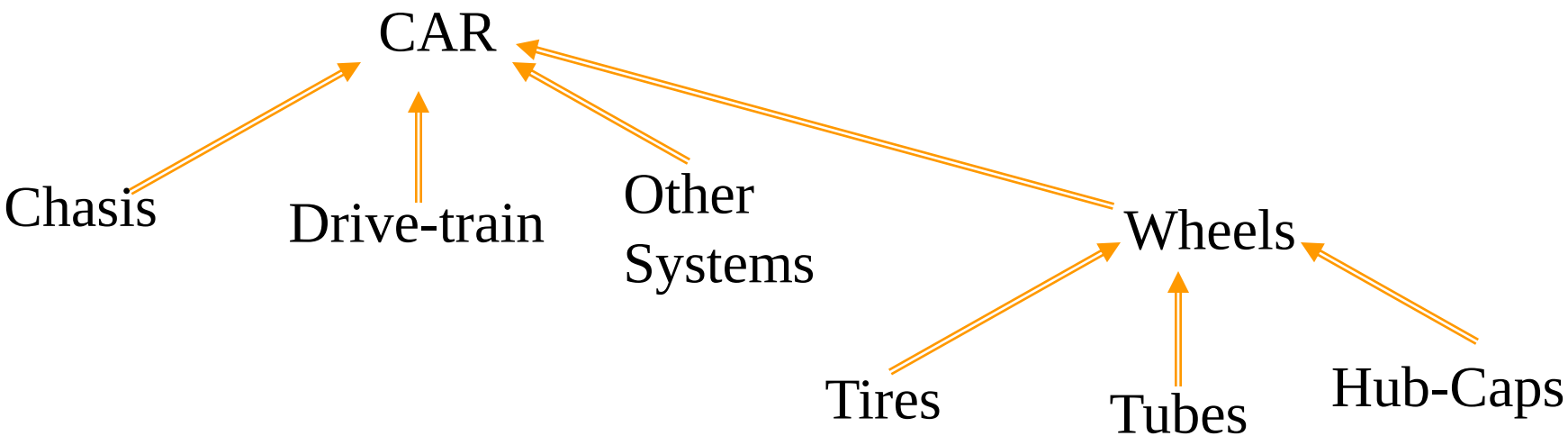
January, February etc. are members of the class “MONTH”

Aggregation Abstraction

In an aggregation we form a concept from existing concepts. The phenomena that are members of the new concept's extension are composed of phenomena from the extensions of the existing concepts



Aggregation Abstraction



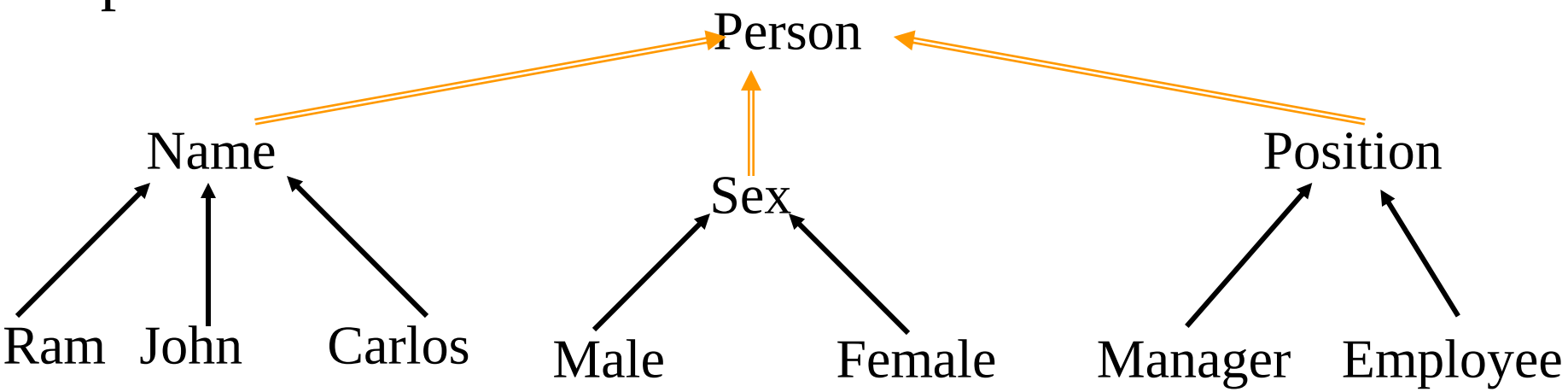
Root class: CAR
Component Classes: Chassis, Drive-Train, Other Systems, Wheels
Root class: Wheels
Component Classes: Tires, Tubes, Hub-Caps

Classification and Aggregation

Classification and Aggregation are used to build schemas

Example: class Person

Representation:

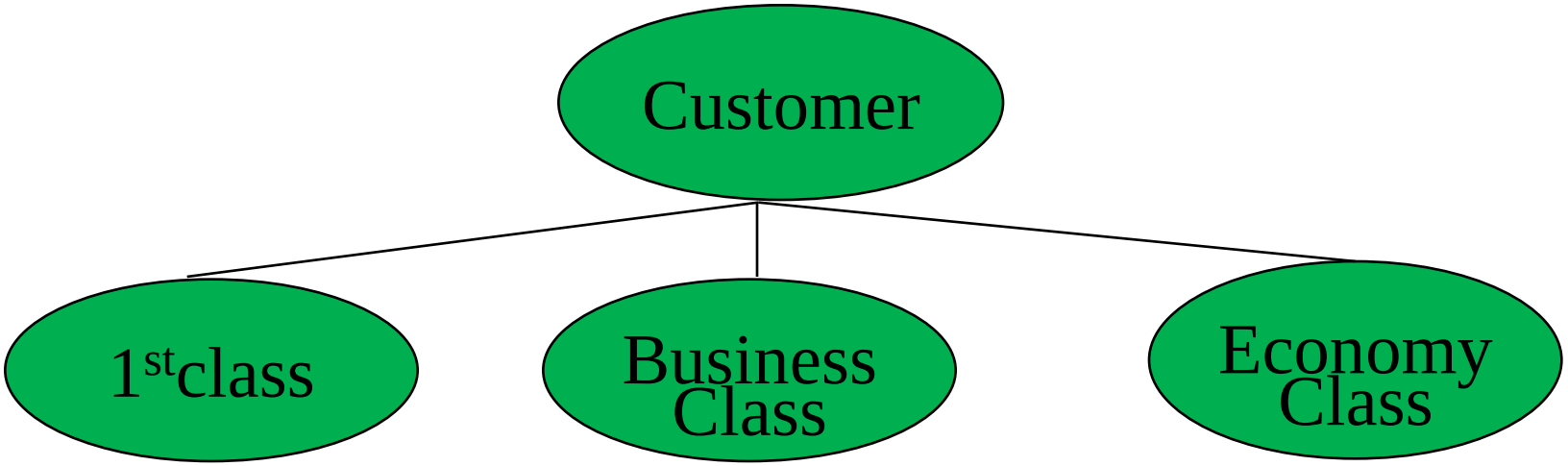


Name, Sex, and Position aggregate into Person. They are classes themselves.

Ram, John, Carlos are classified into Name or Name is a classification of Ram, John, Carlos

Generalization

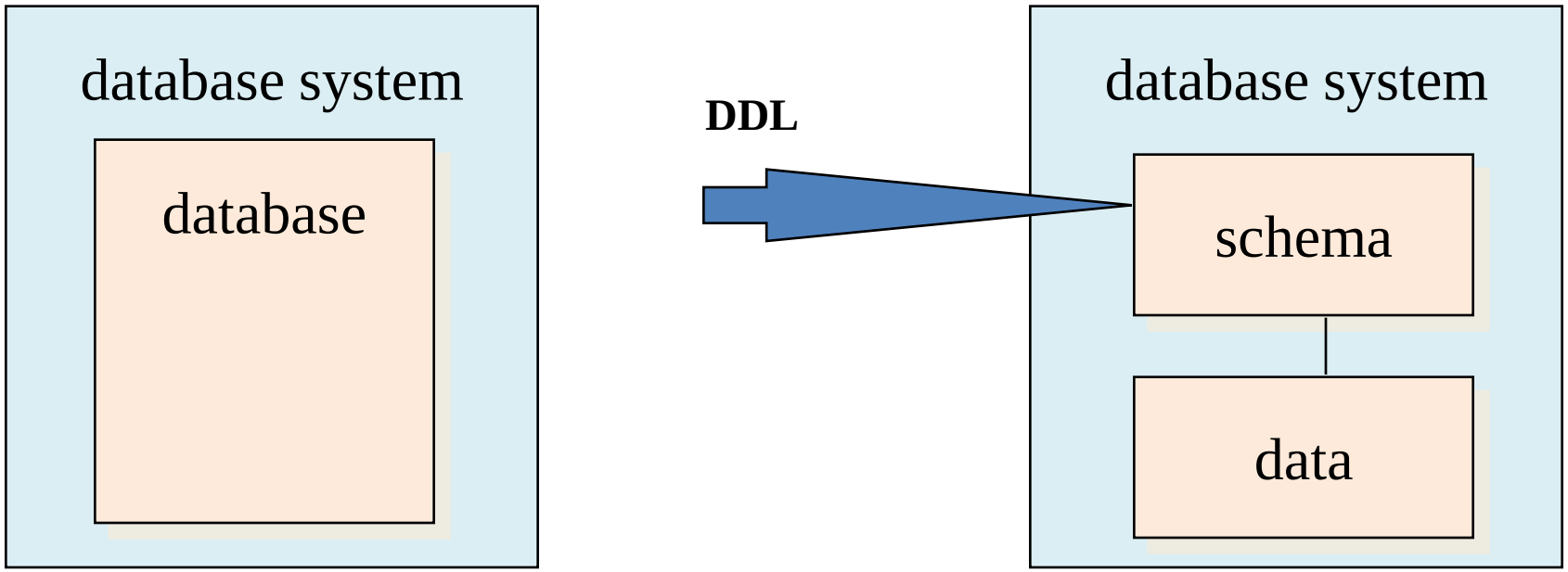
In a generalization we form a new concept by confirming common aspects of existing concepts, leaving out special aspects



Types of Abstractions

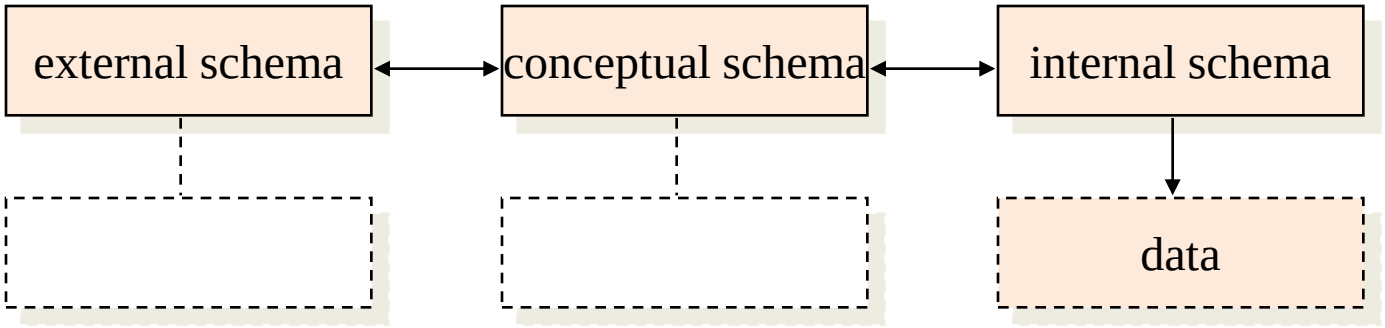
Classification	A is a member of class B
Aggregation	B,C,D are aggregated into A A is made of/composed of B,C,D
Generalization	B,C,D can be generalized into A,

3-Level DB Architecture

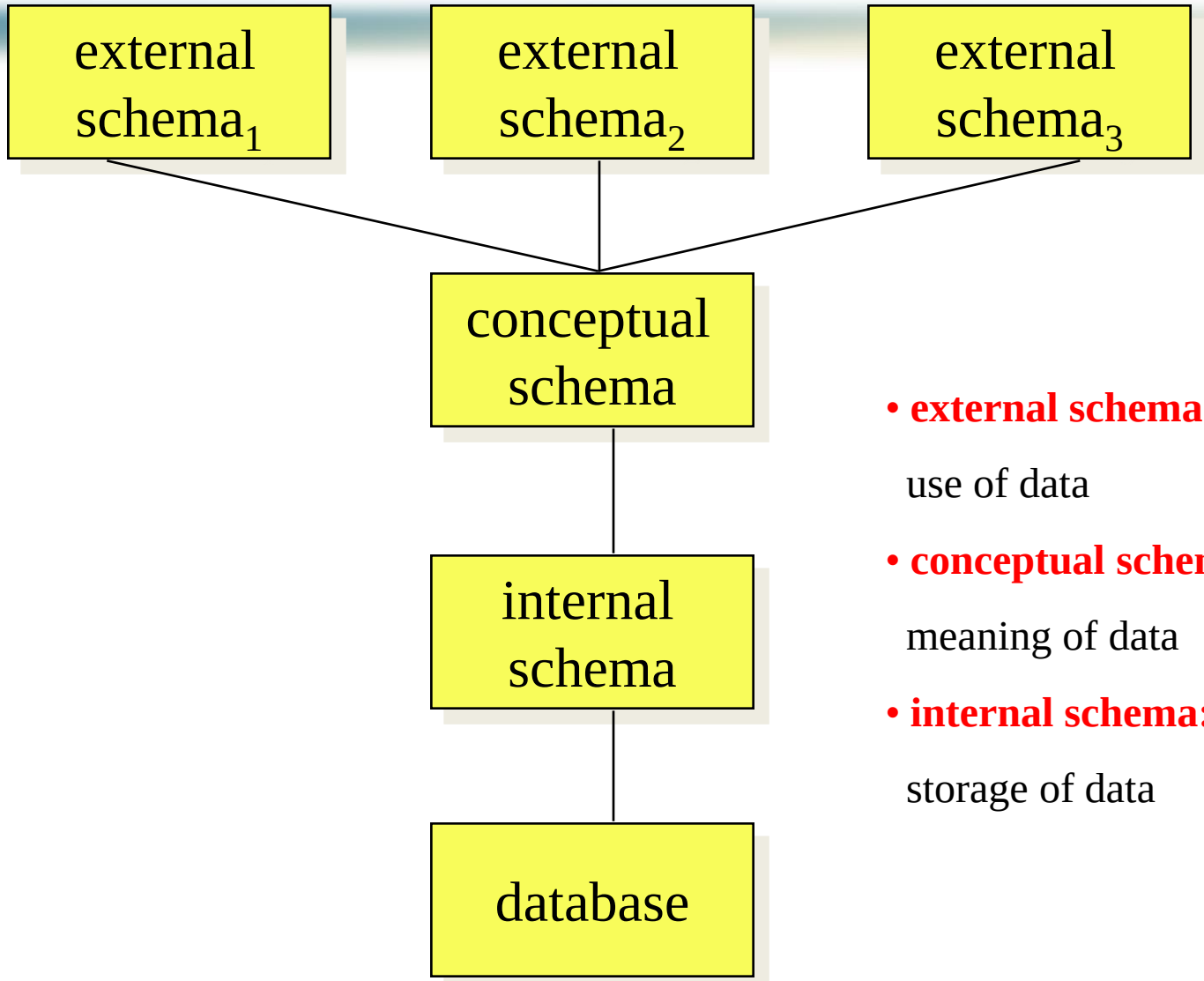


- a database is divided into **schema** and **data**
- the **schema** describes the **types**
- the **data** describes the **extension (data)**

3-Level DB Architecture



3-Level DB Architecture



- **external schema:**

  use of data

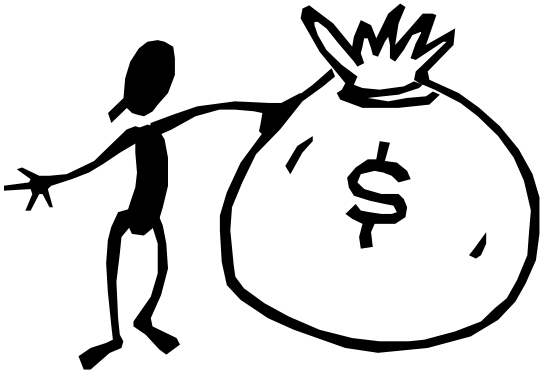
- **conceptual schema:**

  meaning of data

- **internal schema:**

  storage of data

Thank you



???