Module 1: Entity Relationship Models

- Welcome on board!
- This module will introduce you to Database Management Systems, especially Relational Databases.
- Building Entity Relationship Models is the first step to put Ideas into Models.

Learning Objectives

- Finishing this module, you will be able to:
  - Describe the need for DBMS(Database Management Systems)
  - Explain the pros and cons for different DBMS
  - Understand and explain Entities and Attributes
  - Understand and explain Relationships and Cardinalities
  - Draw ERD(Entity Relationship Diagrams)
  - Draw ERAD(Entity Relationship Attribute Diagrams)

Introduction

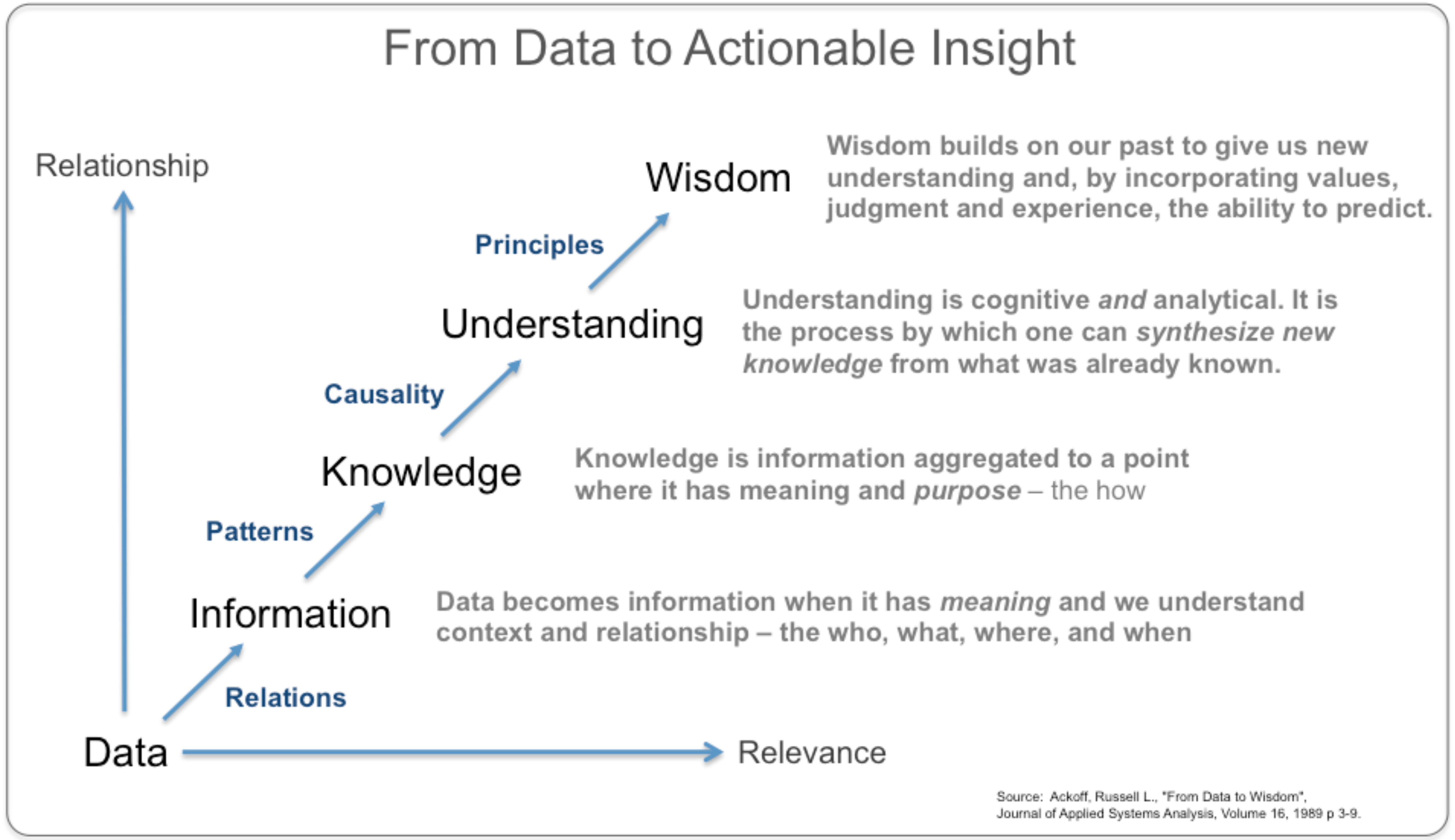
- Why we need Database Management Systems?

Example

- Data: $350
- Information: A share of GameStop as of Jan 29
- Knowledge: Comparing to the price $17.25 of a share as of Jan 4, the price rises around 2000% in a month
- Intelligence: Buy or Sell?

What Is Data?

- Data
- Information
- Knowledge
- Intelligence



From Data to Actionable Insight

Relationship

Relevance

Data

Relations

Information

Data becomes information when it has *meaning* and we understand context and relationship – the who, what, where, and when

Patterns

Knowledge

Knowledge is information aggregated to a point where it has meaning and *purpose* – the how

Causality

Understanding

Understanding is cognitive *and* analytical. It is the process by which one can *synthesize new knowledge* from what was already known.

Principles

Wisdom

Wisdom builds on our past to give us new understanding and, by incorporating values, judgment and experience, the ability to predict.

Source: Ackoff, Russell L., "From Data to Wisdom", Journal of Applied Systems Analysis, Volume 16, 1989 p 3-9.

File System Solution

- File System
  - A collection of individual files accessed by applications programs
- Limitations
  - Separated and Isolated Data
  - Data Duplication
  - Application Program Dependencies
  - Incompatible Files
  - Lack of Data Sharing

Database Solution

- Database
  - A structure that contains data
- Organize Data through
  - Entities
  - Attributes
  - Relationships

Database Management Systems

- A program, or a collection of programs through which users interact with a database
- Popular DBMS:
  - Microsoft Access
  - MySQL
  - PostgreSQL
  - MongoDB(NoSQL)

Advantage of DBMS

- Getting more information from the same amount of data
- Sharing data
- Balancing conflicting requirements
- Controlling redundancy
- Facilitating consistency
- Referential integrity
- Expanding security
- Increasing productivity
- Providing data independence

Disadvantage of DBMS

- Increased complexity
- Greater impact of failure
- More difficult recovery
- Larger file size

When is a DBMS Not Appropriate?

- In short, when the advantages cannot outweigh the disadvantages:
  - Database is small with a simple structure
  - Applications are simple, special purpose and relatively static.
  - Applications have real-time requirements
  - Examples: Traffic signal control
  - ECU patient monitoring
  - Concurrent, multi-user access to data is not required.

Contents of a DBMS

- A Database contains:
  - User Data
  - Metadata
  - Indexes
  - Application metadata

User Data

- Data users work with directly by entering, updating and viewing.
- For our purposes, data will be generally stored in tables with some relationships between tables.
- Each table has one or more columns. A set of columns forms a database record.

Metadata

- Data about data.
- Data that describe how user data are stored in terms of table name, column name, data type, length, primary keys, etc.
- Metadata are typically stored in System tables or System Catalog and are typically only directly accessible by the DBMS or by the system administrator.

Indexes

- In keeping with our desire to provide users with several different views of data, indexes provide an alternate means of accessing user data. Sorting and Searching:
- Indexes allow the database to access a record without having to search through the entire table.
- Updating data requires an extra step: The index must also be updated.

Applications Metadata

- Many DBMS have storage facilities for forms, reports, queries and other application components.
- Applications Metadata is accessed via the database development programs.
- Example: Look at the Documentor tool in MS Access. It can also show metadata for Queries, Forms, Reports, etc.

Let's Get Started

- To design a Relational Database, we start with an Entity Relationship Model (ER Model):
  - Describes what are the entities the database is going to record
  - Describes what are the attributes (and identifiers) of the entities
  - Describes how the relationships among these entities
  - To represent the Entity Relational Model in an explicit way, we use the Entity Relationship Diagram.

Entity Relationship Model

- An Entity Relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge.
- Concepts:
  - Entity
  - Attribute
  - Identifier
  - Relationship

Entity

- The identifiable abstract object of interest (in Programming, we call Class)
- Examples:
  - Students
  - Employees
  - Companies
  - Products
  - Transactions

Attributes

- Characteristics of an entity of interest.
- Examples:
  - Students: FirstName, StudentID, Major
  - Employees: ID, SSN, ContactInfo, Department, Supervisor
  - Companies: LegalName, Location, Category, Rank
  - Products: SKU#, Category, InStockPrice, InStockQuantity
  - Transactions: CustomerID, StoreID, ProductID, ProductDescription, Price, Quantity, Tax, Total, Time

Instance

- A record / member of an entity of interest (with actual value of attributes)
- Examples:
  - A Student: FirstName = "Joe", StudentID = "DB001", Major = "Data Science"
  - A Employee: ID = "E0099", SSN = "123-45-6789", ContactInfo = "N/A", Department = "Education", Supervisor = "Smith"

Identifiers

- A special attribute used to identify a specific instance of an entity
  - May be natural (your DBMS doesn't need to create it): SSN
  - May be artificial (your DBMS needs to create it): EmployeeID
- Examples:
  - Students: FirstName, StudentID, Major
  - Employees: ID, SSN, ContactInfo, Department, Supervisor
  - Companies: LegalName, Location, Category, Rank
  - Products: SKU#, Category, InStockPrice, InStockQuantity

Relationships

- An association between two (or more) entities
  - More specifically, how the members of two (or more) entities are connected. Normally, we name a relationship with a verb.
- Examples:
  - Employees and Companies:
    - How many companies a employee can work?
    - How many employees a company can have?
  - Companies and products:
    - How many companies a product can belong to?
    - How many products a company can produce?

Relationships - Degree

- A relationship can include one or more entities
- The degree of a relationship is the number of Entities that participate in the relationship
- Relationships of degree 1 are called Unary relationships (also called Recursive).
- Relationships of degree 2 are called Binary relationships. Most relationships in databases are Binary.

Relationships - Cardinality

- Cardinality refers to the number of instances of the entity involved in the relationship.
  - Also called max cardinality / multiplicity of a relationship
- There are three types:
  - 1:N (also called One to Many)
  - N:1 (also called Many to One)
  - N:M (also called Many to Many)

Relationships - Participation

- Participation of instances in a relationship may be mandatory or optional.
  - Also called optionality, minimal cardinality of a relationship
- There are two types:
  - Mandatory
  - Optional

Practice

- Your client is an online-education company. You need to help your client to record the data needed.
  - The company has more than 100 instructors, more than 400 courses, more than 50 online programs, and around 1 million students.
  - Instructors have info: Name, EmpID, SSN, DoB, Email, Salary
  - Courses have info: Title, Course#, Time, Location, Description.
  - Programs have info: Title, Chair, Office#, Contact, Description.
  - Students have info: Name, StuID, DoB, Email

Practice

- You should first find Entities, what are they?
  - Instructors, Courses, Programs, Students.
- Then, what are the attributes of these Entities?
  - Instructors have info: Name, *EmpID, SSN, DoB, Email, Salary
  - Courses have info: Title, *Course#, Time, Location, Description.
  - Programs have info: *Title, Chair, Office#, Contact, Description.
  - Students have info: Name, *StuID, DoB, Email
- Then, what are the identifiers of these Entities?

Practice

- At last, what are the relationships among these Entities?
  - An instructor may teach multiple courses; and a course might be taught by multiple instructors.
  - Each course must belong to one and only one online-program, and each program must have one or more courses.
  - A student may take multiple courses; and a course must have one or more students.
  - A student must belong to exactly one program; and a program may have one or more students.
  - A student may be friend of other students
  - A student must have exactly one instructor as advisor; and one instructor may advise one or more students

Practice

- Let's do more practice in Lab1.

Entity Relationship Diagram

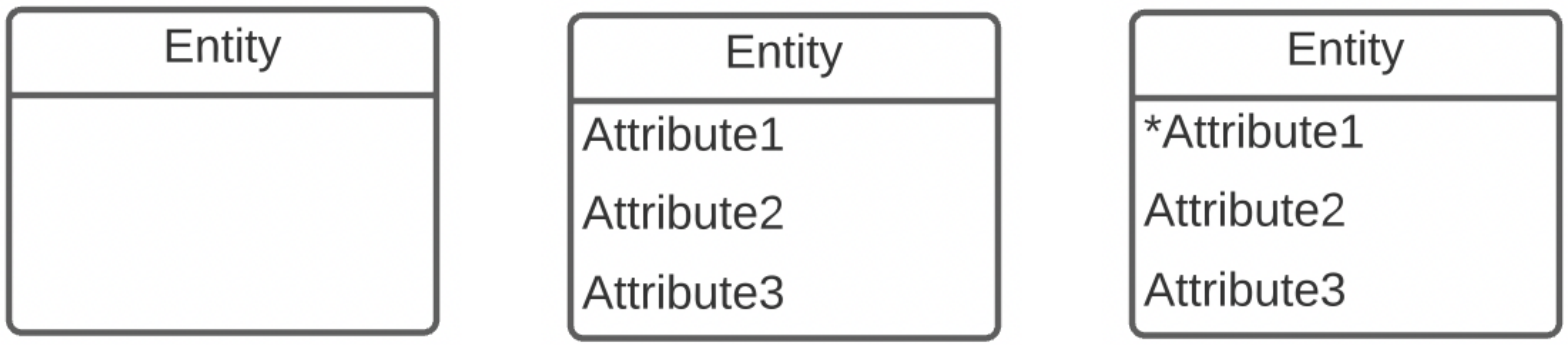
- We represent Entity Relationship Models using ERD(Entity Relationship Diagram)
    - Represent Entities (and attributes / identifiers)
    - Represent Relationships (and cardinality / participation)
    - Provide a high level picture of what the database is organized

Notations for ERD

- There are several notations used for ERD:
  - UML (Unified Modeling Language)
  - Chen
  - Crow's Foot
- We are going to use Crow's Foot for this course (and other courses in the Specialization)
- We are going to use Lucid.app to draw ERDs. You can create a free account or login with your Google account.

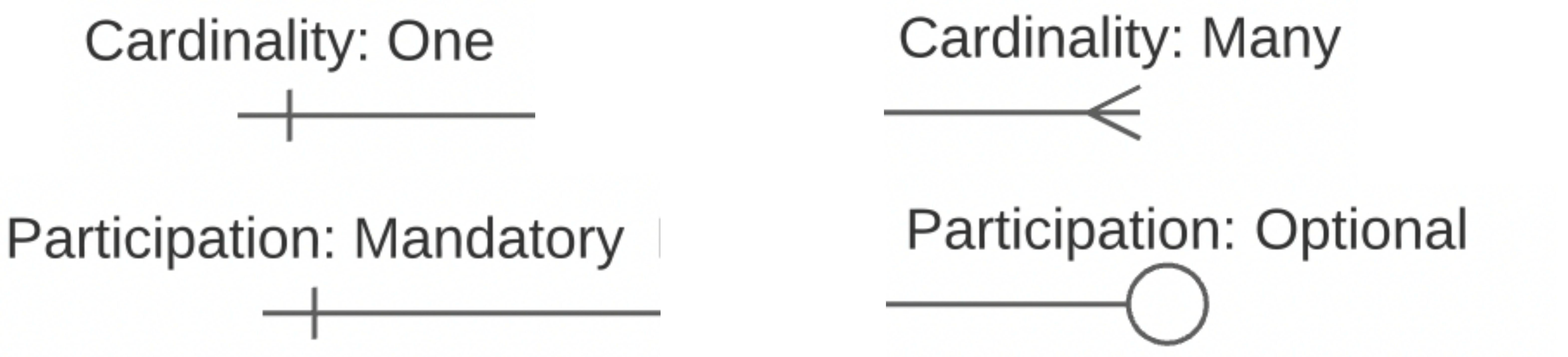
ERD with Crow's Foot

- Entities
  - Attributes
  - Identifiers



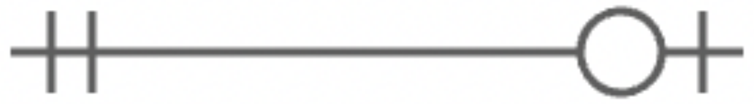
ERD with Crow's Foot

- Relationships
  - Name
  - Cardinality (One, Many)
  - Participation (Mandatory, Optional)

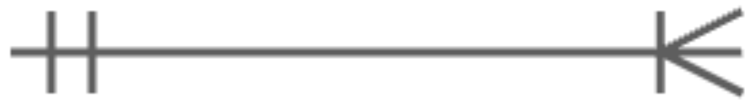


ERD with Crow's Foot

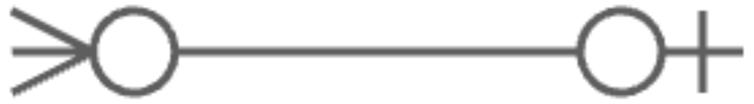
- Example
  - One Mandatory (1 and only 1) to One Optional (0 or 1)



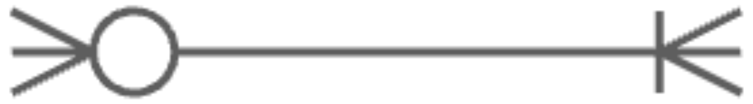
  - One Mandatory (1 and 1) to Many Mandatory (1 or N)



  - Many Optional (0 or N) to One Optional (0 or 1)

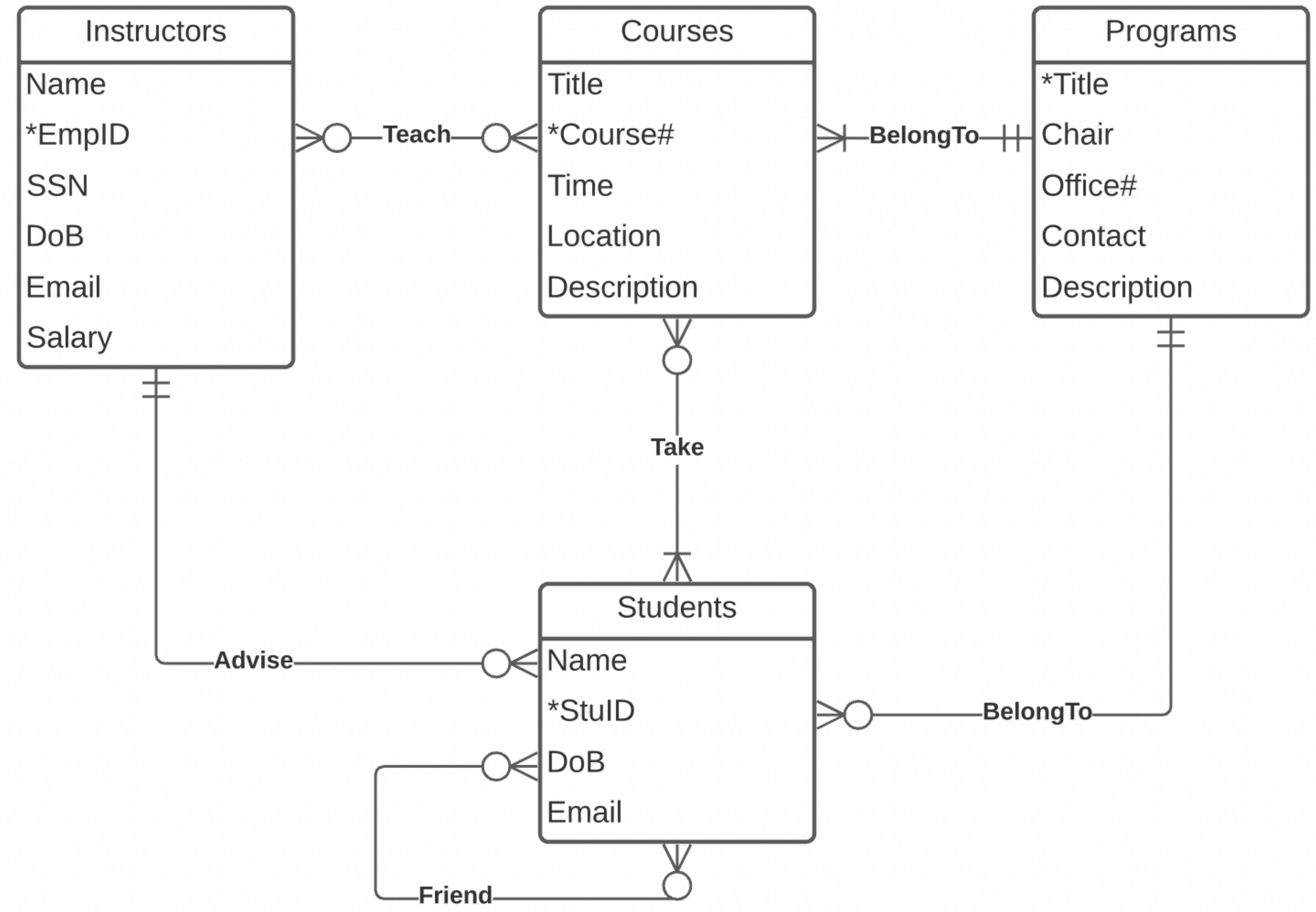


  - Many Optional (0 or N) to Many Mandatory (1 or N)



Practice

- Let's represent the simple case using Crow's Foot notation.



Practice

- Let's do more practice in Lab2.

Assignment

- Let's do the assignment to assess your understanding of Database, Entity Relationship Models, and Entity Relationship Diagrams.

Congratulations

- Now you finished Module 1!
- You should be comfortable to lay out the Entity Relationship Model, and use the Entity Relationship Diagram to represent it.
- Next step is to convert it to a Relational Model.
- See you soon.