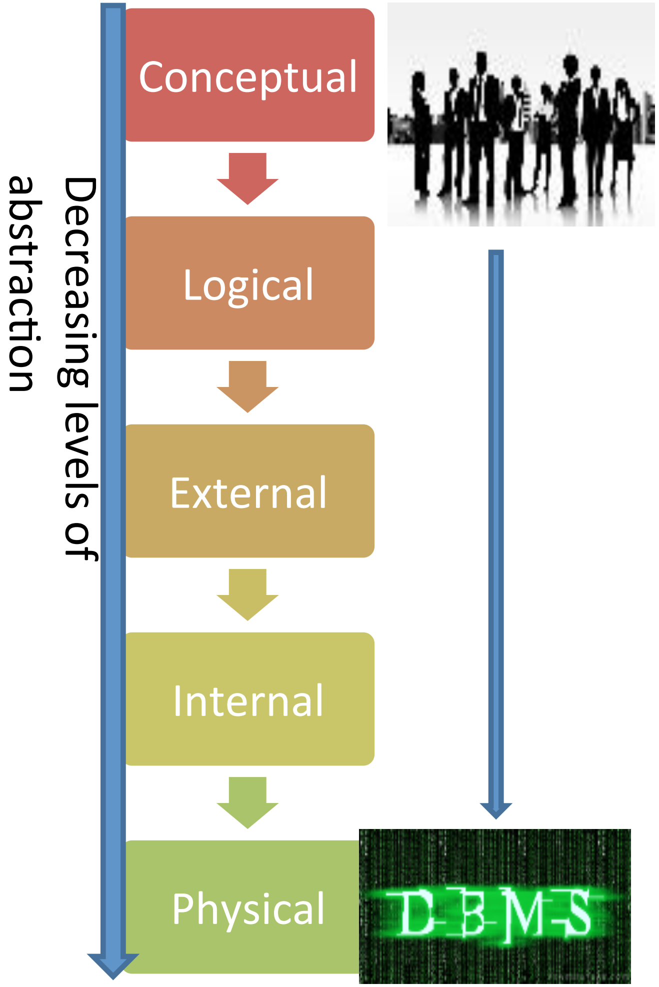
Data Modeling

Levels of Data Model Abstraction

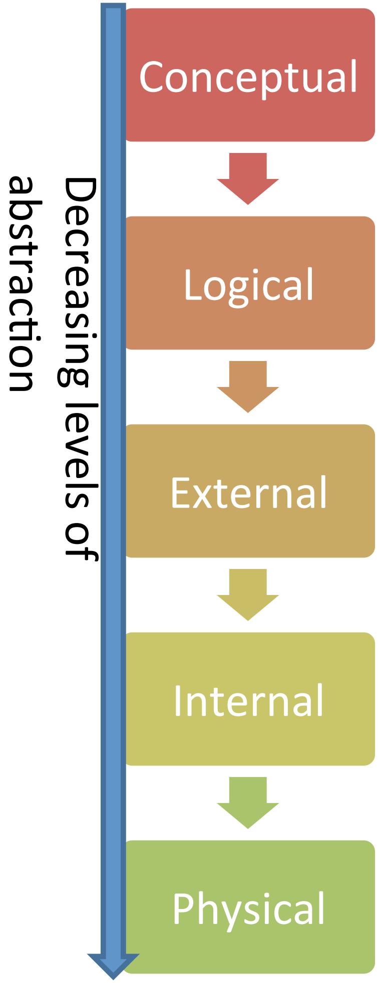


How the data are viewed and processed by people (users)

Modeling tools and processes map real-world data requirements to machine-readable database architecture

How the data are viewed and processed by machines (the DBMS)

Levels of Data Model Abstraction



- Conceptual
  - Discussion of the business rules and identification of the entities and relationships in the model. Writing the narrative of the database.
- Logical
  - Identification of attributes and keys and their types. Normalization of the data occurs here.
- External
  - The interface that applications will use to interface with the data.
- Internal
  - The internal mechanics of reading and manipulating the data.
- Physical
  - The definition of how the data will be stored on disk.

Question

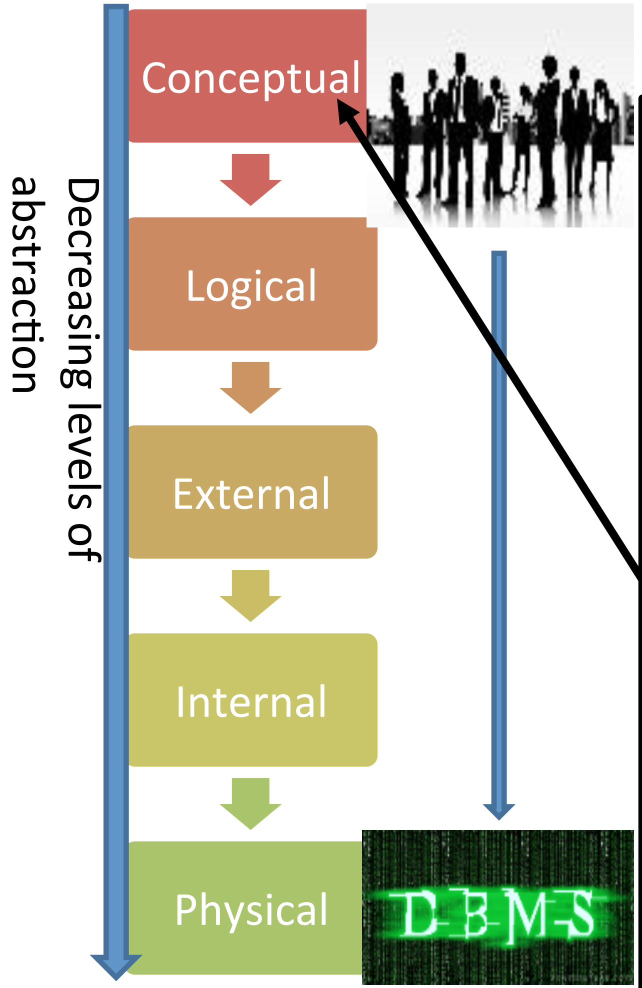
In your opinion, what is the importance of defining the levels of data abstraction?

The Conceptual Model

Learning Objectives

- **Define** *conceptual modeling* and **understand** how it fits into the larger context of database management systems.
- **Identify** the information required for building the narrative around the data.
- **Describe** characteristics of good business rules.

What Is the Conceptual Model?



Conceptual Model

- Describes the business data and the rules for governing their use and storage
- Identifies the things that are important for an organization to store
- Provides a framework for communicating data needs to and from business users
- Uses entity-relationship modeling to codify business rules in a way that can be read by technical and nontechnical people

Where to Begin?

- Conceptual modeling begins with a conversation.
- Collect any pertinent artifacts that help support the narrative defined in the conversation.
- Acquaint yourself with the problem domain.
- Never underestimate the power of listening.

Identifying the Business Rules

- Business rules govern how data are stored and accessed.
- Identify a glossary of business terms with definitions.
- Identify the “things” the business needs to track.
- How do those “things” interact with one another, and how do they constrain one another?

Characteristics of “Good” Business Rules

- Declarative
- Precise
- Atomic
- Consistent
- Expressible
- Distinct
- Business oriented

Modern Database Management (12th ed)., p. 61

Codify the Rules

- Document business rules early and often.
- Be technology-agnostic.
- Review your documentation with your stakeholders.
- Plan for flexibility.

Overview of the ER Model

The Relational Database Model

- Rooted in first-order predicate logic
  - We won’t get too deep into this during this class, but know that this is where some of the terminology comes from.
- Defines databases as sets of named relations
  - Relations can also be called *tables* (as in SQL Server, et al.).
  - Relations can also be called *entities.*
- Relations are connected through…um…relationships
  - We use data points called “foreign keys” to link records to other tables.

Data Modeling

- Identify the things for which you need to store data.
  - **Entities**
  - E.g., customers, orders, products
- Identify the characteristics of each of those things you need to store.
  - **Attributes**
  - E.g., name, address, price, order date
- Identify the business rules that connect those things together.
  - **Relationships**
  - E.g., customer places orders; order contains products

Data Modeling

In addition to identifying the things and their relationships, we need to identify the **cardinality** of the relationship.

- A customer places 1 or more orders (1..N).
- An order is placed by 1 and only 1 customer (1..1).
- A product can be on 0 or more orders (0..N).
- An order can contain 1 or more products (1..N).

Entities and Relationships

What are the entities? Relationships? Attributes?

Eve, an event organizer, has asked you to provide a database system for tracking event fundraising. Eve would like to keep track of teams who sign up for events. Eve would like to track teams by their type.

Events occur all the time, so it will be important to know when the event took place and what the fundraising goals were for each team and for the event as a whole.

Teams can participate in more than one event!

Entities and Relationships

What are the **entities**? **Relationships**? **Attributes**?

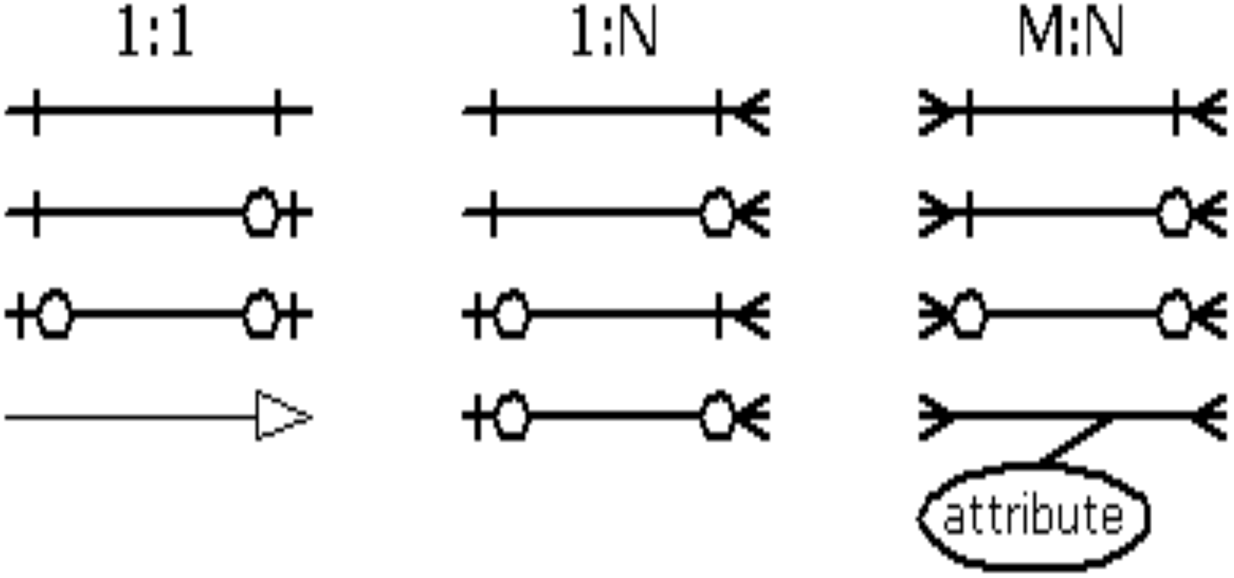
Eve, an event organizer, has asked you to provide a database system for tracking **event** fundraising. Eve would like to keep track of **teams** who sign up for **events**. Eve would like to **track teams** by their **type**.

Events occur all the time, so it will be important to know **when the event took place** and what **the fundraising goals** were for each **team** and for the **event** as a whole.

Teams can participate in more than one event!

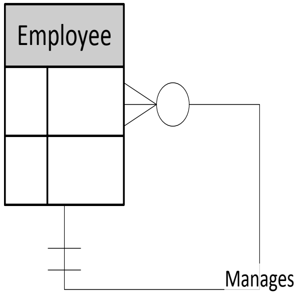
Entities and Relationships

Cardinality is expressed in diagrams with a 1 (mandatory), and 0 (optional), and a crow's foot (many).

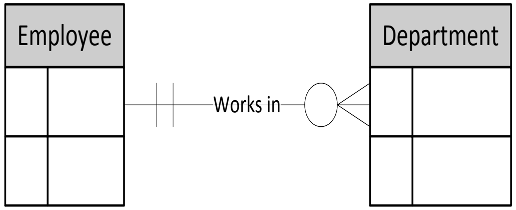


Relationships

- Unary relationships show relationships between one instance of an entity and a different instance of the same entity.
- Binary relationships show relationships between two different entity types.



Unary Relationship



Binary Relationship

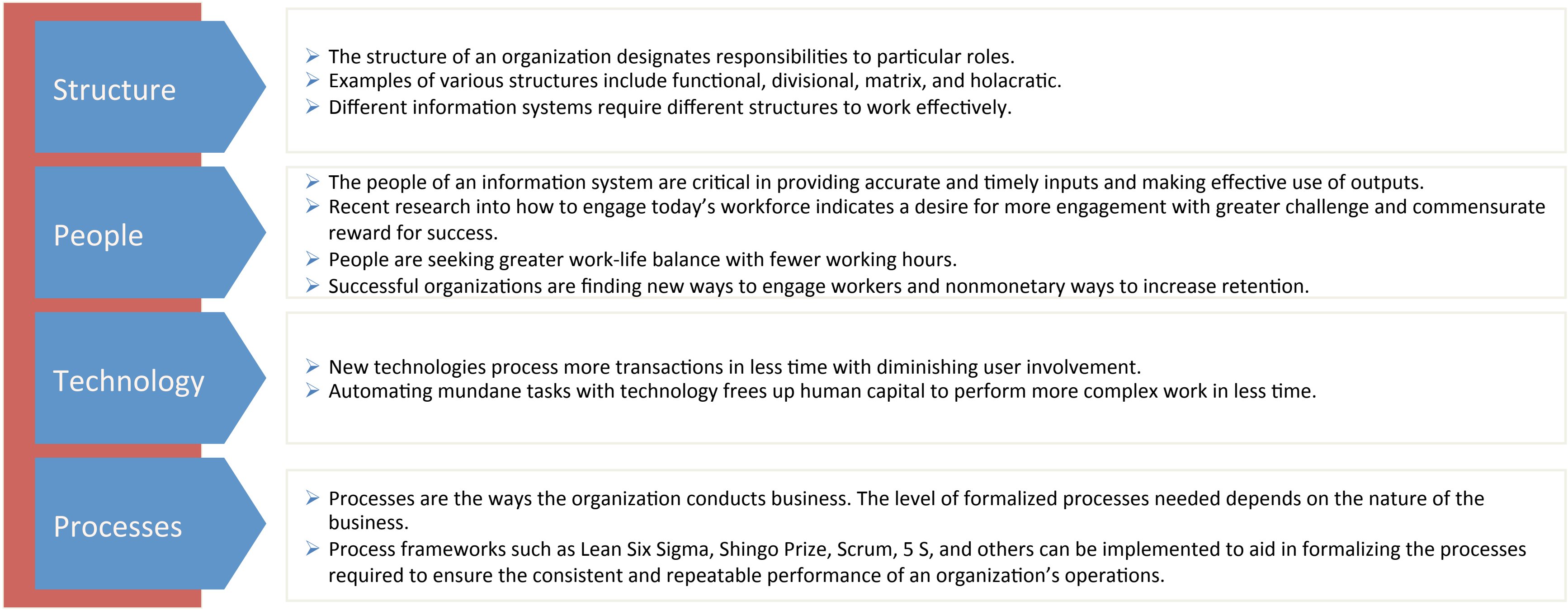
Modeling Organizational Rules

Organizational Rules

- A data model represents rules within an organization.
- Rules provide direction for organizational stakeholders to react to events that affect their organizations.
- The data model is a reflection of the rules and policies governing the creation and maintenance of data.
- Represent rules unambiguously and concisely.

Information Systems Components

Information systems in organizations can be divided into four parts: structure, people, technology, and process. These four parts of information systems in organizations are all equally important.



Structure

- The structure of an organization designates responsibilities to particular roles.
- Examples of various structures include functional, divisional, matrix, and holacratic.
- Different information systems require different structures to work effectively.

People

- The people of an information system are critical in providing accurate and timely inputs and making effective use of outputs.
- Recent research into how to engage today's workforce indicates a desire for more engagement with greater challenge and commensurate reward for success.
- People are seeking greater work-life balance with fewer working hours.
- Successful organizations are finding new ways to engage workers and nonmonetary ways to increase retention.

Technology

- New technologies process more transactions in less time with diminishing user involvement.
- Automating mundane tasks with technology frees up human capital to perform more complex work in less time.

Processes

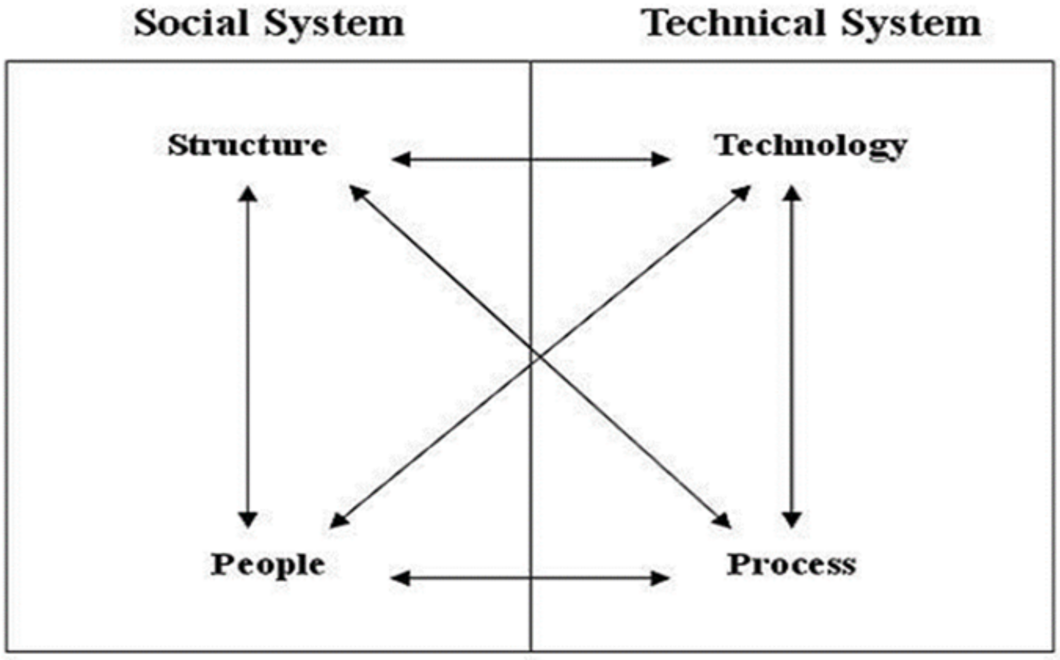
- Processes are the ways the organization conducts business. The level of formalized processes needed depends on the nature of the business.
- Process frameworks such as Lean Six Sigma, Shingo Prize, Scrum, 5 S, and others can be implemented to aid in formalizing the processes required to ensure the consistent and repeatable performance of an organization's operations.

Social and Technical Systems

The four parts of information systems work together to form the entirety of the system. Each part must be well-designed and managed. These four parts belong to one of two groups: the social system and the technical system.

Social System
The people and their roles and responsibilities

Technical System
The technology and processes of the system



(Piccoli, 2012, p. 29)

Organizational Rules

Review of characteristics of good business rules

- Declarative
- Precise
- Atomic
- Consistent
- Expressible
- Distinct
- Business oriented

Sample Rules

- “Customer orders are only accepted if the customer’s account is in good standing.”
- “Students can only register for a course if they have successfully completed all prerequisites or if a waiver has been received by the professor of record.”
- “Every production job must be associated with a customer order.”
- “Library materials kept past the due date incur a late fee.”

Gathering Rules

- Ways to identify rules:
  - Interview stakeholders.
  - Review SOP and policy documents.
  - Review source documents (POs, invoices, vouchers, etc.).
- Tips:
  - Be good listener.
  - Approach the documentation process as if you are training to do the job of the stakeholder.
  - Document, document, document.

Modeling Entities and Attributes

Entities

- The “things” about which we want to maintain data
  - Person
  - Place
  - Object
  - Event
  - Concept
- An entity will have many instances, each identifiable by some descriptive data.

Types vs. Instances

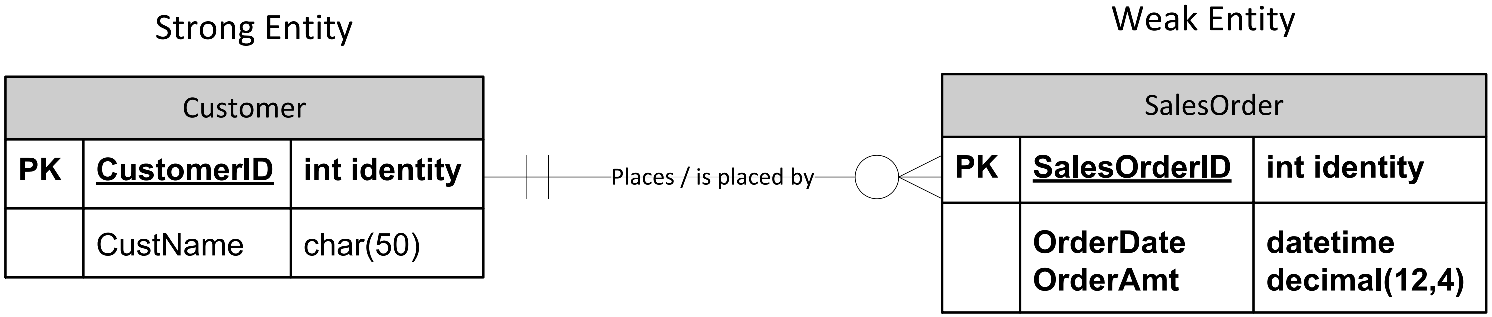
- **Entity type** is a collection of instances of people, places, objects, events, concepts.
- Instances are single occurrences of a single entity type.

Entity	Instance 1	Instance 2	Instance 3
Musician	Saul Hudson, Guitar	Les Claypool, Bass	Pepper Keenan, Guitar
Building	Sears Tower, Chicago, USA	Taj Mahal, Agra, India	Eiffel Tower, Paris, France
Operating system	Ubuntu, Linux Kernel, Canonical	Android OS, Linux Kernel, Google	Fedora, Linux Kernel, Red Hat

Strong vs. Weak Entities

- Strong entity type exists independently of other entity types.

- Weak entity type depends on other entities to exist.



Strong Entity

Customer		
PK	**<u>CustomerID</u>**	**int identity**
	CustName	char(50)

Weak Entity

SalesOrder		
PK	**<u>SalesOrderID</u>**	**int identity**
	OrderDate **OrderAmt**	**datetime** **decimal(12,4)**

Naming Entities

- Singular noun
- Specific to the organization
- Concise
- Name event entities after the result of the event
- Name should be consistently used across all documentation (avoid using synonyms)

Defining Entities

- What is it?
- How is it identified?
- Which instances are included and which are left out?
- When is it created or deleted?
- Does it ever change into a different entity? When and how?
- What historical information is important to track for this entity?
- Be consistent! Answer all of these questions using the same basic language.

Attributes

- Data points that describe entities
- Attributes of a sample customer entity
  - ID
  - Name
  - Address
  - Phone
  - Fax
  - Web Site
  - Contact Person

Required vs. Optional

- Do we need to know this attribute for every instance? If so, that's a required attribute.
- If we don't necessarily HAVE to track it, it is optional.

Customer Entity Attribute	Required or Optional?
ID	Required
Name	Required
Address	Required
Phone	Required
Fax	Optional
Web Site	Optional
Contact Person	Required

Simple vs. Composite

- Composite attributes can be broken down into smaller meaningful parts.
  - Name: Forename, Surname, Company Name, Title, Salutation
  - Address: Street, City, State, Postal Code, Country

- Simple attributes cannot be meaningfully broken down.
  - ID
  - Phone (if country code, area code, and exchange aren't meaningful to the organization)

Scalar vs. Multivalued

- Scalar attributes have a single value for each entity instance (ID, OrderDate, OrderAmount, etc).

- Multivalue attributes can have one or more values per entity instance (Books Borrowed, Customer Ship To Address).

** To get a jump on the process, consider making multivalue attributes their own entities.*

Stored vs. Derived

- Stored attributes are values that must be entered directly.
  - Order item quantity
  - Order date
  - Book title
- Derived attributes are calculated from other stored and derived attributes.
  - Order subtotals
  - Count of materials rented
  - Aged accounts receivable balance

Identifiers

- Attributes or combinations of attributes that identify an instance of an entity
- Can be automatically generated (best practice)
- Must be unique to the instance of an entity
  - Name is not a good choice, but Name and a combination of other attributes might work.

Naming Attributes

- Singular noun or noun phrase
- Must be unique to the entity
- Pick a format and be consistent

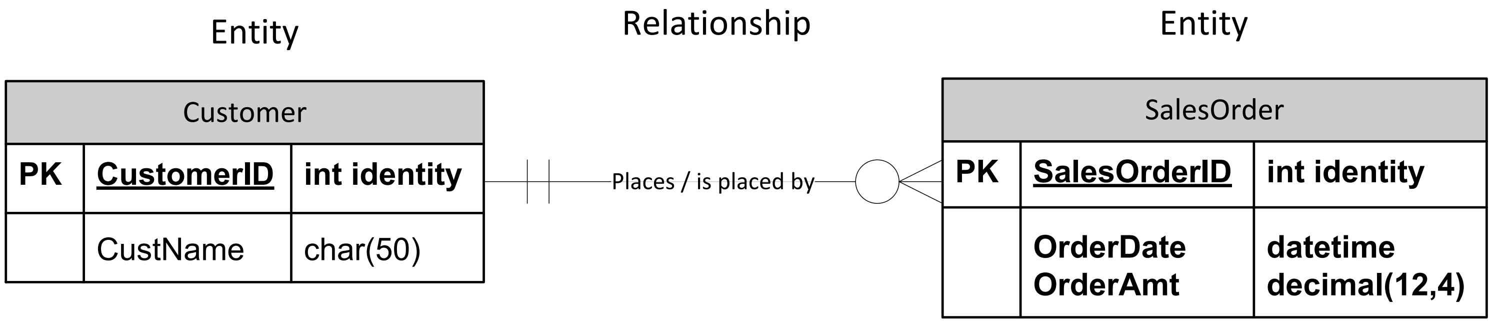
Defining Attributes

- What is the attribute and why is it important?
- What is included and what is not?
- Are there any synonyms for this attribute?
- What is the source of the values?
- Required or optional? Derived or stored?
- Can it change and how?
- What are the maximum and minimum (logical) values for this attribute?
- What are the relationships to other attributes?
- Be consistent!

Modeling Relationships

Relationships

- Represent interactions among entity instances
- Relationships and their characteristics represent business rules



Entity

Relationship

Entity

Customer

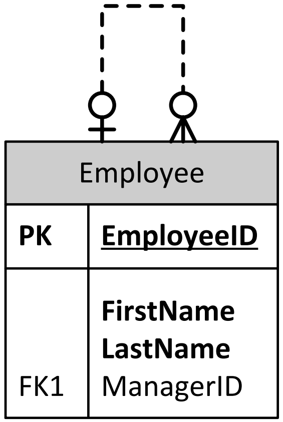
PK	**<u>CustomerID</u>**	**int identity**
	CustName	char(50)

Places / is placed by

SalesOrder

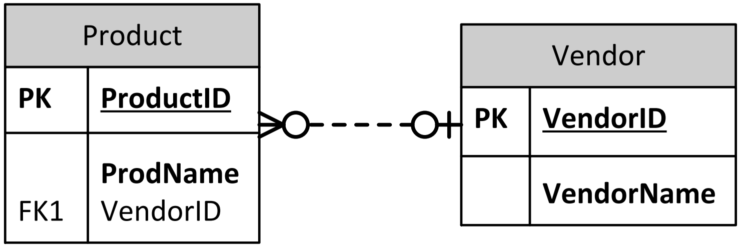
PK	**<u>SalesOrderID</u>**	**int identity**
	OrderDate **OrderAmt**	**datetime** **decimal(12,4)**

Degree of a Relationship



Employee	
PK	**EmployeeID**
  FK1	**FirstName** **LastName** ManagerID

Unary



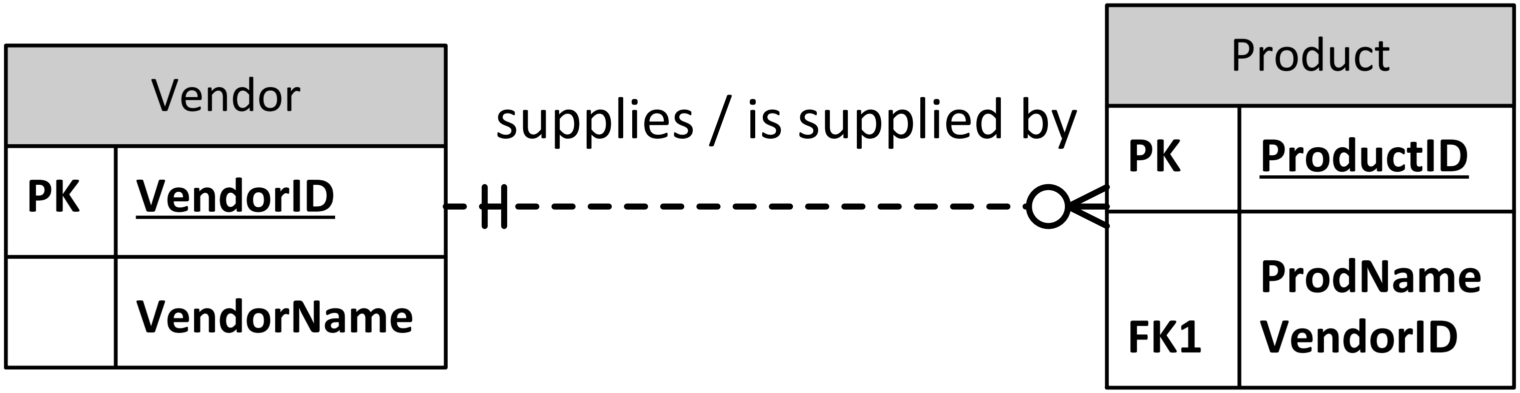
Product	
PK	**ProductID**
 FK1	**ProdName** VendorID

Vendor	
PK	**VendorID**
	VendorName

Binary

Cardinality of a Relationship

Determines the number of possible instances of each participating entity are allowed in a relationship instance. Often shown in “crow’s foot” notation.



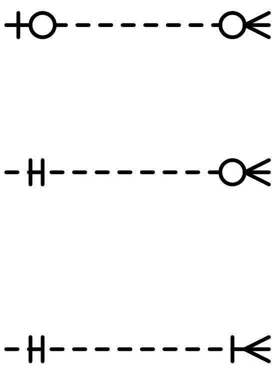
Vendor	
PK	VendorID
	VendorName

supplies / is supplied by

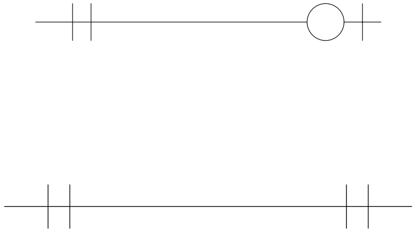
Product	
PK	ProductID
 FK1	ProdName VendorID

Cardinality of a Relationship

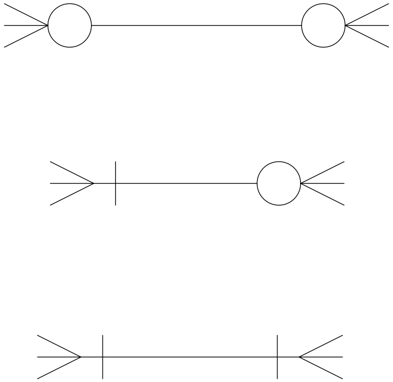
1 to Many



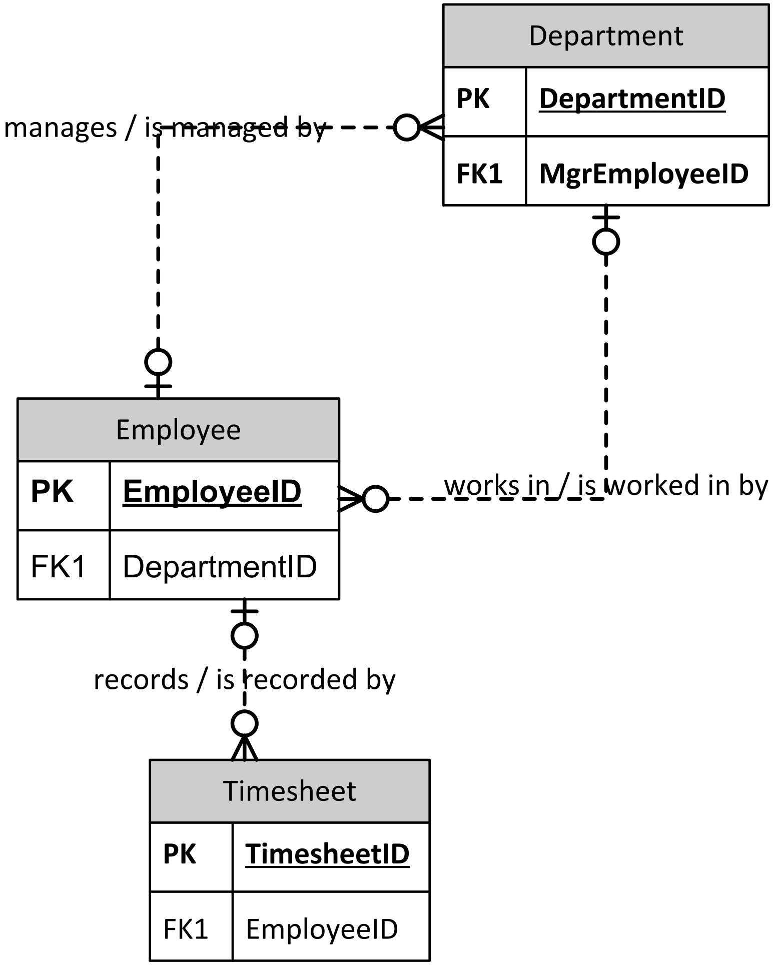
1 to 1



Many to Many



ERD



Department	
PK	**DepartmentID**
FK1	**MgrEmployeeID**

Employee	
PK	**EmployeeID**
FK1	DepartmentID

Timesheet	
PK	**TimesheetID**
FK1	EmployeeID

SYRACUSE UNIVERSITY

SUOS CULTORES SCIENTIA CORONAT