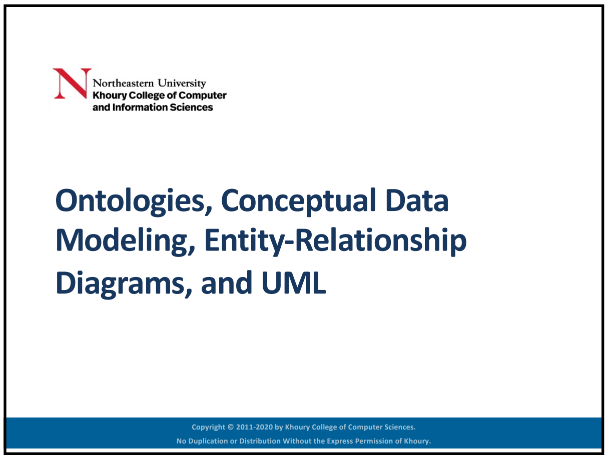
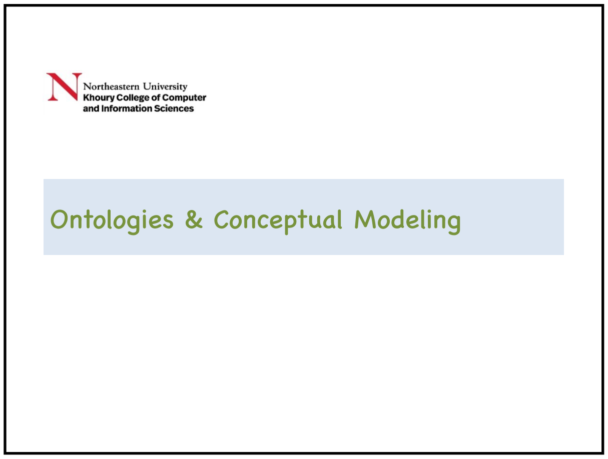


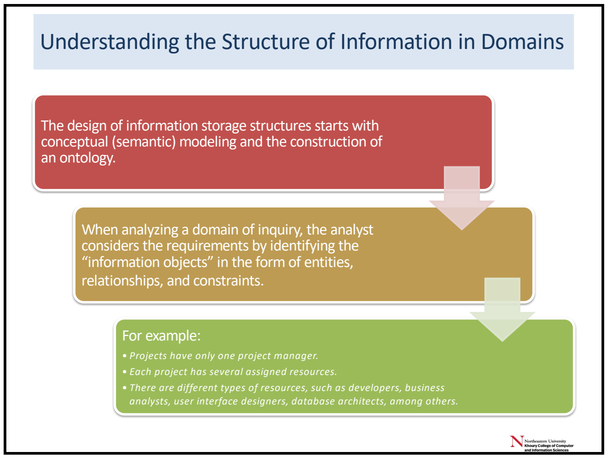
Northeastern University
Khoury College of Computer and Information Sciences

Ontologies, Conceptual Data Modeling, Entity-Relationship Diagrams, and UML



Northeastern University
Khoury College of Computer and Information Sciences

Ontologies & Conceptual Modeling



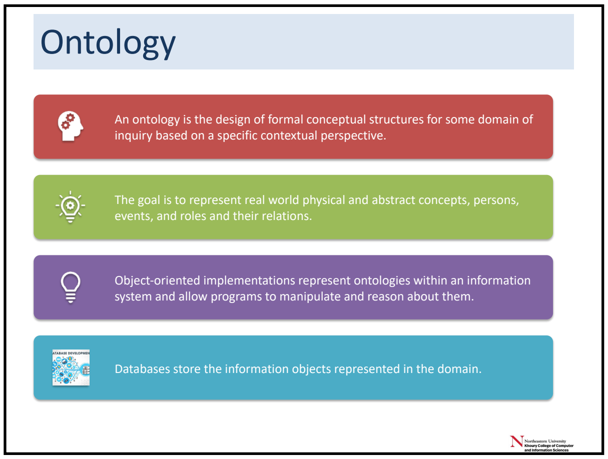
Understanding the Structure of Information in Domains

The design of information storage structures starts with conceptual (semantic) modeling and the construction of an ontology.

When analyzing a domain of inquiry, the analyst considers the requirements by identifying the "information objects" in the form of entities, relationships, and constraints.

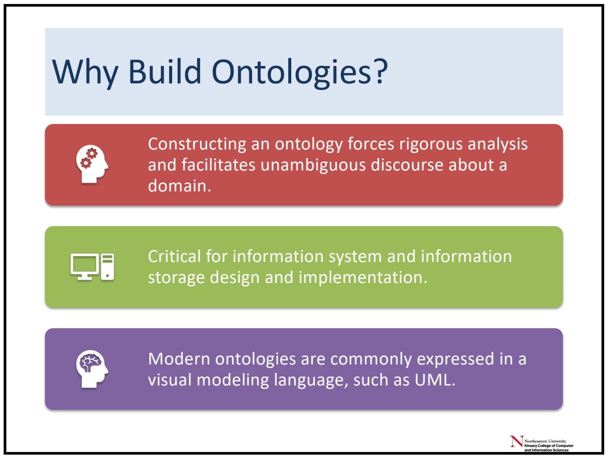
For example:

- *Projects have only one project manager.*
- *Each project has several assigned resources.*
- *There are different types of resources, such as developers, business analysts, user interface designers, database architects, among others.*



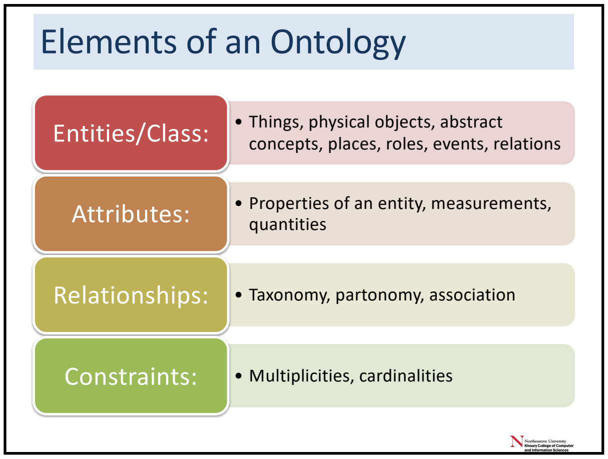
Ontology

- An ontology is the design of formal conceptual structures for some domain of inquiry based on a specific contextual perspective.
- The goal is to represent real world physical and abstract concepts, persons, events, and roles and their relations.
- Object-oriented implementations represent ontologies within an information system and allow programs to manipulate and reason about them.
- Databases store the information objects represented in the domain.



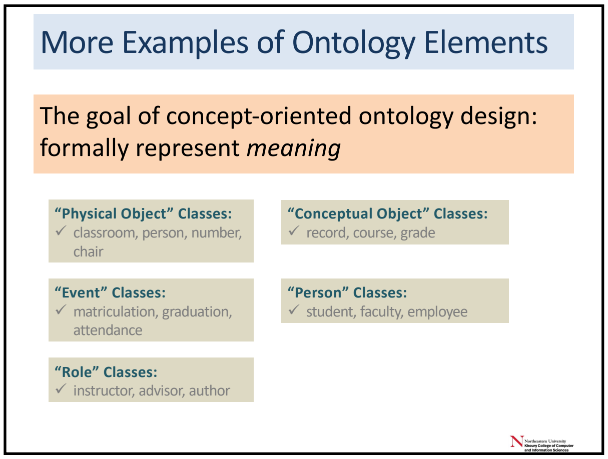
Why Build Ontologies?

- Constructing an ontology forces rigorous analysis and facilitates unambiguous discourse about a domain.
- Critical for information system and information storage design and implementation.
- Modern ontologies are commonly expressed in a visual modeling language, such as UML.



Elements of an Ontology

Entities/Class:	• Things, physical objects, abstract concepts, places, roles, events, relations
Attributes:	• Properties of an entity, measurements, quantities
Relationships:	• Taxonomy, partonomy, association
Constraints:	• Multiplicities, cardinalities



More Examples of Ontology Elements

The goal of concept-oriented ontology design: formally represent *meaning*

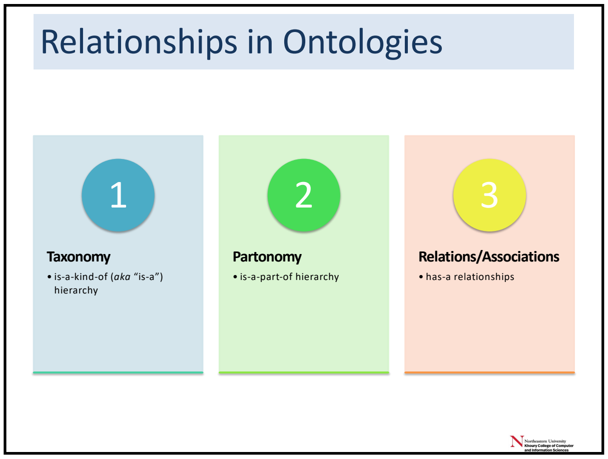
"Physical Object" Classes:
✓ classroom, person, number, chair

"Conceptual Object" Classes:
✓ record, course, grade

"Event" Classes:
✓ matriculation, graduation, attendance

"Person" Classes:
✓ student, faculty, employee

"Role" Classes:
✓ instructor, advisor, author



Relationships in Ontologies

1

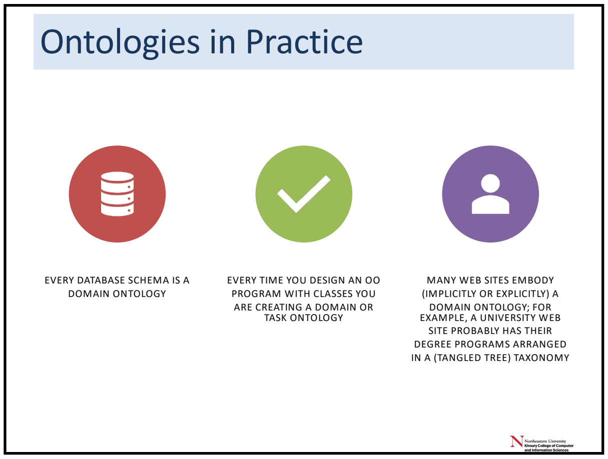
Taxonomy
- is-a-kind-of (*aka* "is-a") hierarchy

2

Partonomy
- is-a-part-of hierarchy

3

Relations/Associations
- has-a relationships



Ontologies in Practice

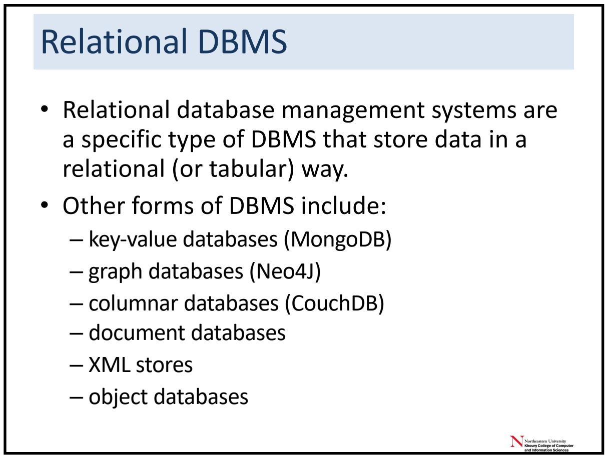
EVERY DATABASE SCHEMA IS A DOMAIN ONTOLOGY

EVERY TIME YOU DESIGN AN OO PROGRAM WITH CLASSES YOU ARE CREATING A DOMAIN OR TASK ONTOLOGY

MANY WEB SITES EMBODY (IMPLICITLY OR EXPLICITLY) A DOMAIN ONTOLOGY; FOR EXAMPLE, A UNIVERSITY WEB SITE PROBABLY HAS THEIR DEGREE PROGRAMS ARRANGED IN A (TANGLED TREE) TAXONOMY

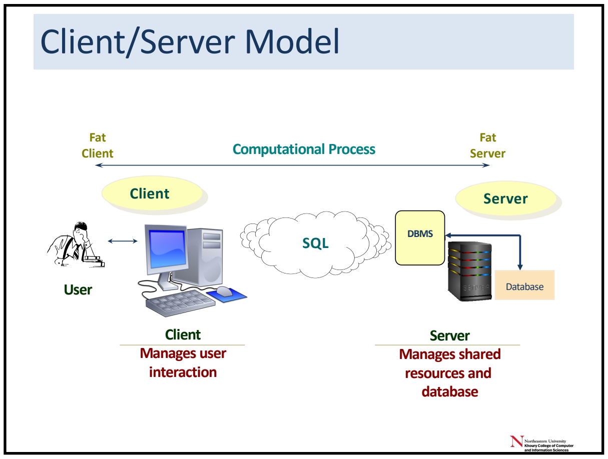
Northeastern University
Khoury College of Computer and Information Sciences

Database Architectures



Relational DBMS

- Relational database management systems are a specific type of DBMS that store data in a relational (or tabular) way.
- Other forms of DBMS include:
  - key-value databases (MongoDB)
  - graph databases (Neo4J)
  - columnar databases (CouchDB)
  - document databases
  - XML stores
  - object databases



Client/Server Model

Client
Manages user interaction

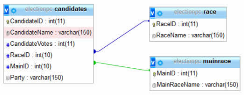
Server
Manages shared resources and database

Servers vs Cloud File Storage

- An analogy:
  - Take-out menus and using them to find a dish to order…

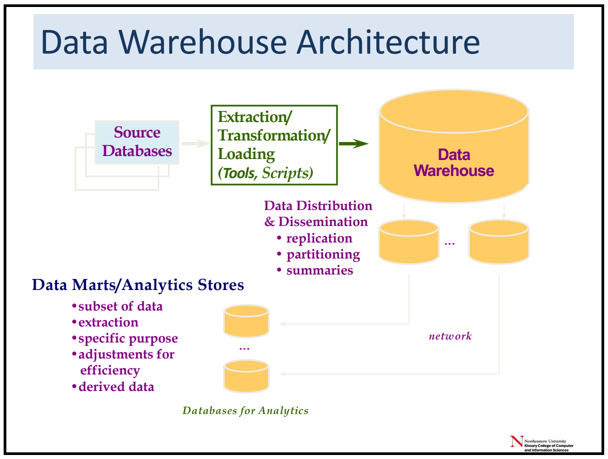
Relational Database Model

- The relational data model is the most commonly used database structures today:
  - physical data aspects are hidden
  - data records are logically stored in tables
  - each table contains related entries in rows
  - each row represents a data record
  - SQL is used to access the data records



Database Types

- Organizations employ databases in two primary ways:
  - On-line transaction processing (OLTP)
    - support customer service
    - interactive queries and real-time updates
    - concurrent access by many users
  - On-line analytical processing (OLAP)
    - facilitate decision support
    - analytical queries, no real-time update



Data Warehouse Architecture

Source Databases

Extraction/ Transformation/ Loading (*Tools, Scripts)*

Data Warehouse

Data Distribution & Dissemination
- replication
- partitioning
- summaries

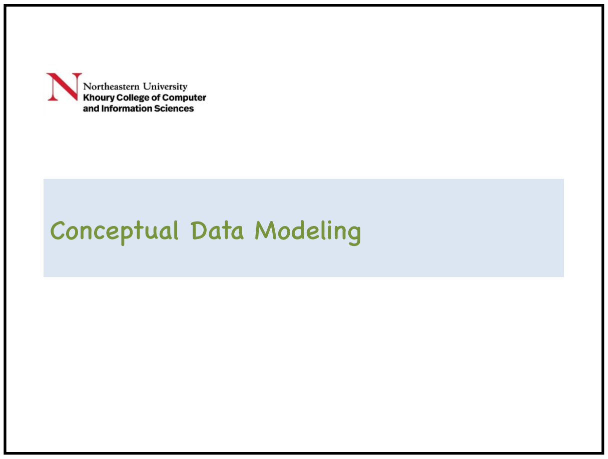
...

Data Marts/Analytics Stores
- subset of data
- extraction
- specific purpose
- adjustments for efficiency
- derived data

...

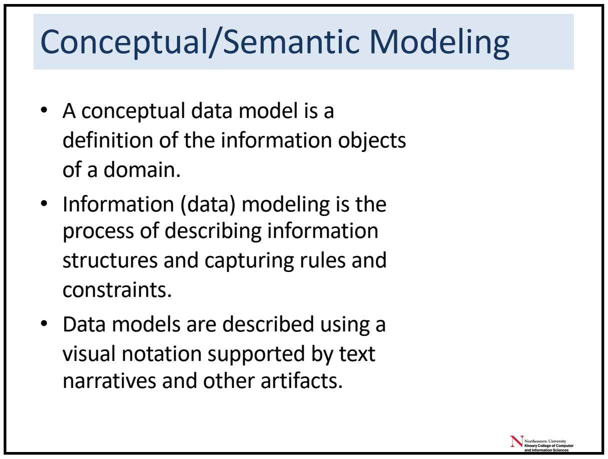
network

Databases for Analytics



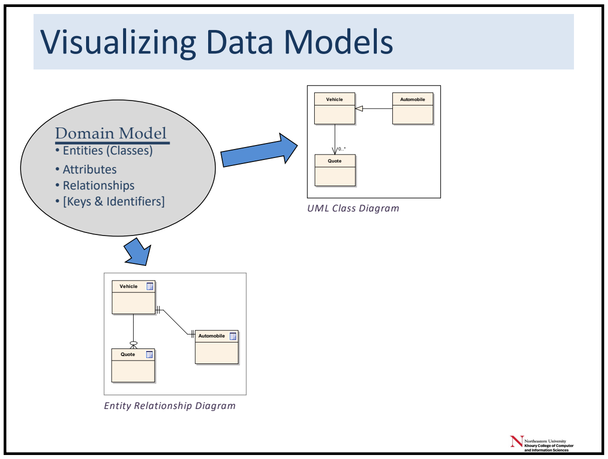
Northeastern University Khoury College of Computer and Information Sciences

Conceptual Data Modeling



Conceptual/Semantic Modeling

- A conceptual data model is a definition of the information objects of a domain.
- Information (data) modeling is the process of describing information structures and capturing rules and constraints.
- Data models are described using a visual notation supported by text narratives and other artifacts.



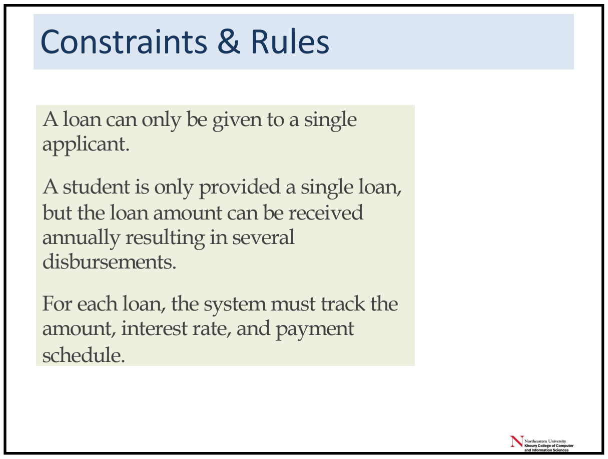
Visualizing Data Models

Domain Model

- Entities (Classes)
- Attributes
- Relationships
- [Keys & Identifiers]

UML Class Diagram

Entity Relationship Diagram

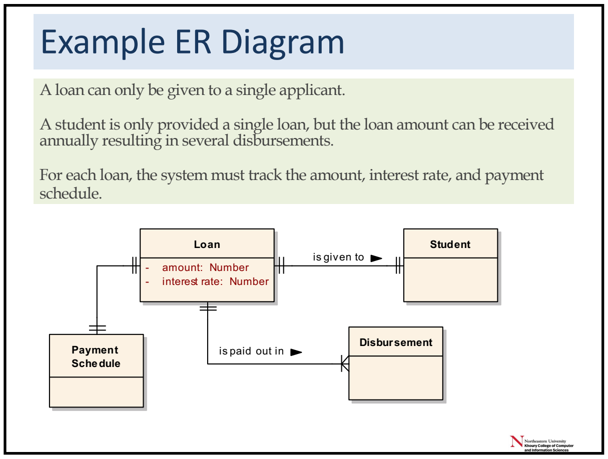


Constraints & Rules

A loan can only be given to a single applicant.

A student is only provided a single loan, but the loan amount can be received annually resulting in several disbursements.

For each loan, the system must track the amount, interest rate, and payment schedule.

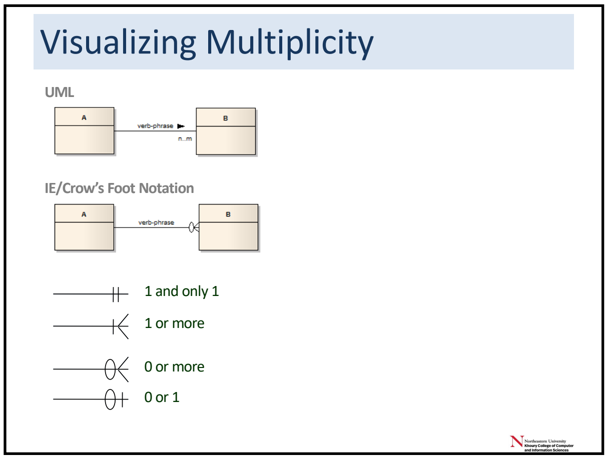


Example ER Diagram

A loan can only be given to a single applicant.

A student is only provided a single loan, but the loan amount can be received annually resulting in several disbursements.

For each loan, the system must track the amount, interest rate, and payment schedule.



Visualizing Multiplicity

UML

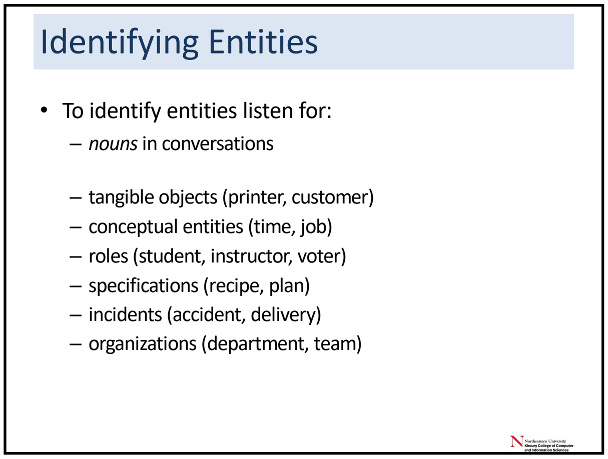
IE/Crow's Foot Notation

1 and only 1

1 or more

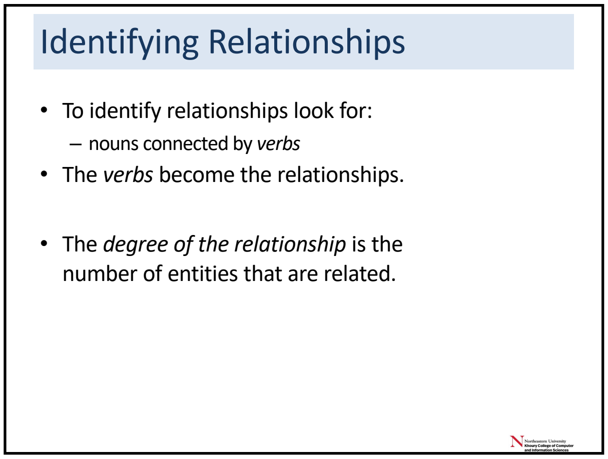
0 or more

0 or 1



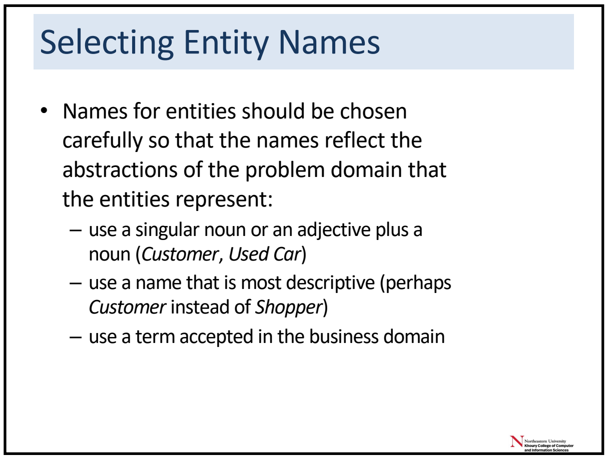
Identifying Entities

- To identify entities listen for:
  - *nouns* in conversations
  - tangible objects (printer, customer)
  - conceptual entities (time, job)
  - roles (student, instructor, voter)
  - specifications (recipe, plan)
  - incidents (accident, delivery)
  - organizations (department, team)



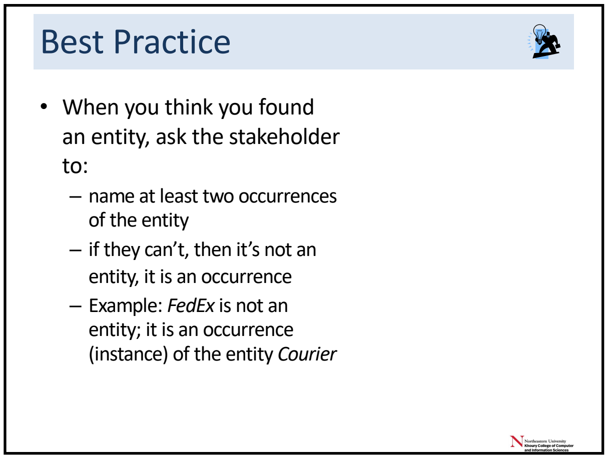
Identifying Relationships

- To identify relationships look for:
  - nouns connected by *verbs*
- The *verbs* become the relationships.
- The *degree of the relationship* is the number of entities that are related.



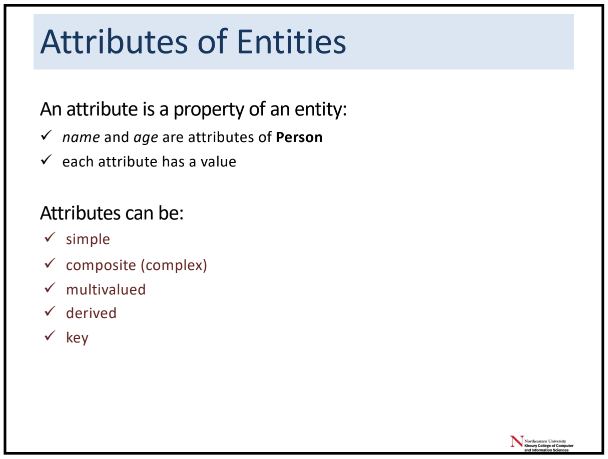
Selecting Entity Names

- Names for entities should be chosen carefully so that the names reflect the abstractions of the problem domain that the entities represent:
  - use a singular noun or an adjective plus a noun (*Customer*, *Used Car*)
  - use a name that is most descriptive (perhaps *Customer* instead of *Shopper*)
  - use a term accepted in the business domain



Best Practice

- When you think you found an entity, ask the stakeholder to:
  - name at least two occurrences of the entity
  - if they can’t, then it’s not an entity, it is an occurrence
  - Example: *FedEx* is not an entity; it is an occurrence (instance) of the entity *Courier*



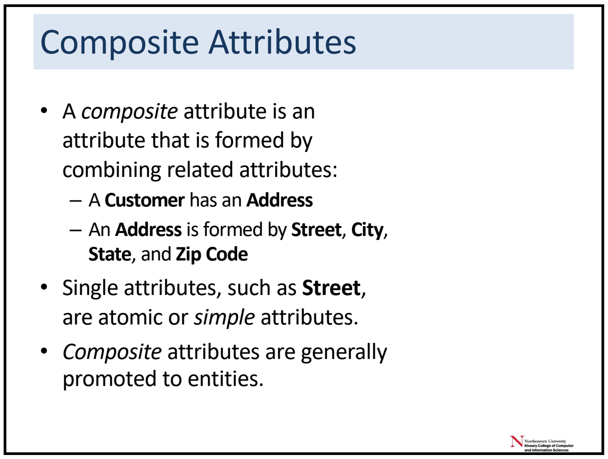
Attributes of Entities

An attribute is a property of an entity:

- ✓ *name* and *age* are attributes of **Person**
- ✓ each attribute has a value

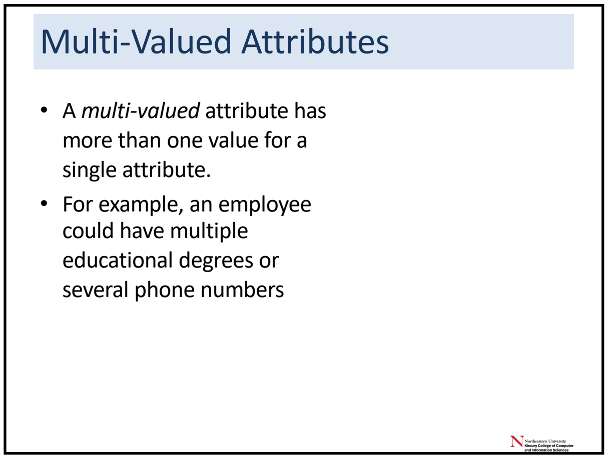
Attributes can be:

- ✓ simple
- ✓ composite (complex)
- ✓ multivalued
- ✓ derived
- ✓ key



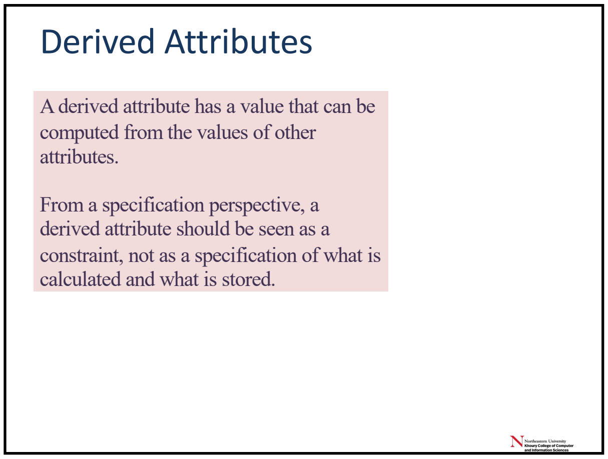
Composite Attributes

- A *composite* attribute is an attribute that is formed by combining related attributes:
  - A **Customer** has an **Address**
  - An **Address** is formed by **Street**, **City**, **State**, and **Zip Code**
- Single attributes, such as **Street**, are atomic or *simple* attributes.
- *Composite* attributes are generally promoted to entities.



Multi-Valued Attributes

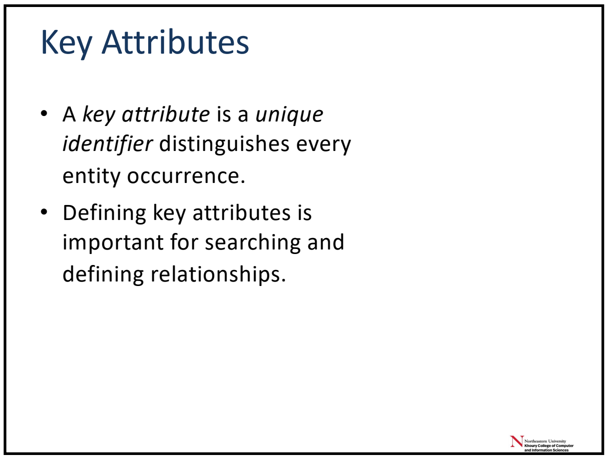
- A *multi-valued* attribute has more than one value for a single attribute.
- For example, an employee could have multiple educational degrees or several phone numbers



Derived Attributes

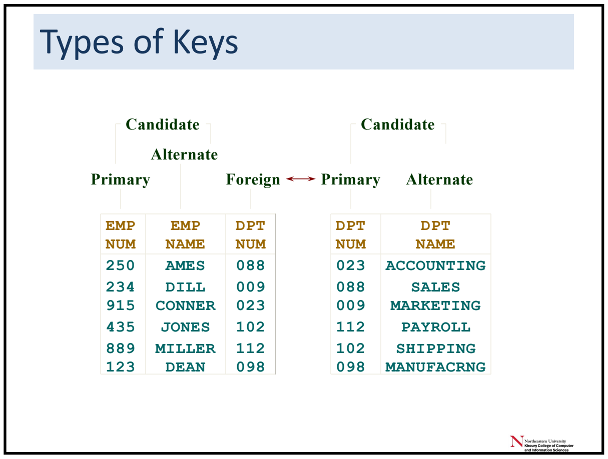
A derived attribute has a value that can be computed from the values of other attributes.

From a specification perspective, a derived attribute should be seen as a constraint, not as a specification of what is calculated and what is stored.



Key Attributes

- A *key attribute* is a *unique identifier* distinguishes every entity occurrence.
- Defining key attributes is important for searching and defining relationships.



Types of Keys

Candidate

Alternate

Primary

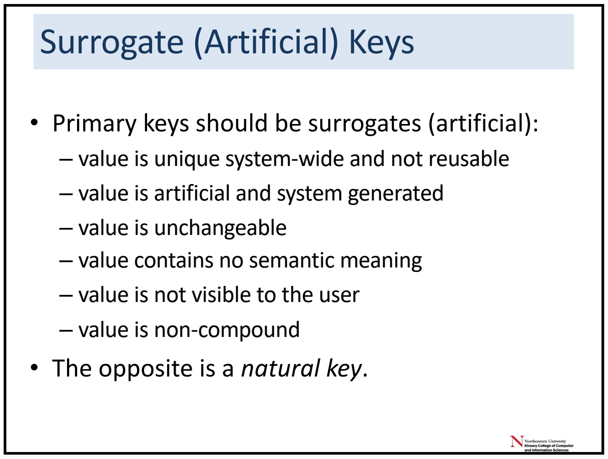
Foreign ←→ Primary

Alternate

Candidate

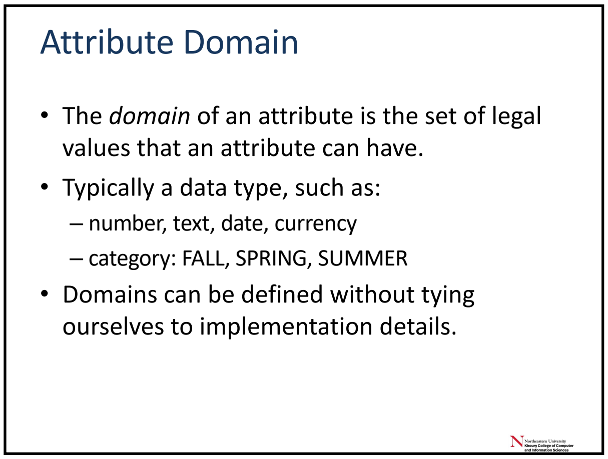
EMP NUM	EMP NAME	DPT NUM
250	AMES	088
234	DILL	009
915	CONNER	023
435	JONES	102
889	MILLER	112
123	DEAN	098

DPT NUM	DPT NAME
023	ACCOUNTING
088	SALES
009	MARKETING
112	PAYROLL
102	SHIPPING
098	MANUFACRNG



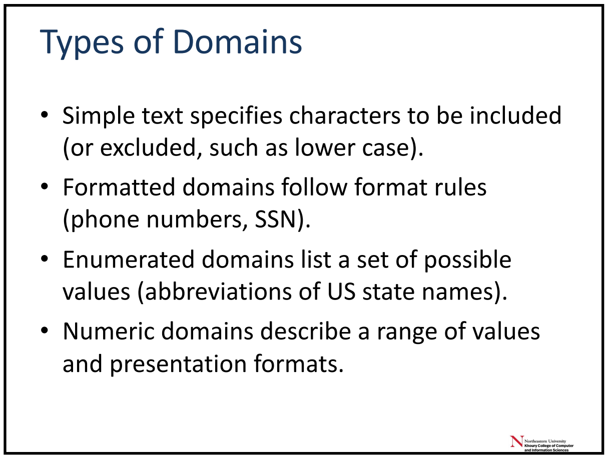
Surrogate (Artificial) Keys

- Primary keys should be surrogates (artificial):
  - value is unique system-wide and not reusable
  - value is artificial and system generated
  - value is unchangeable
  - value contains no semantic meaning
  - value is not visible to the user
  - value is non-compound
- The opposite is a *natural key*.



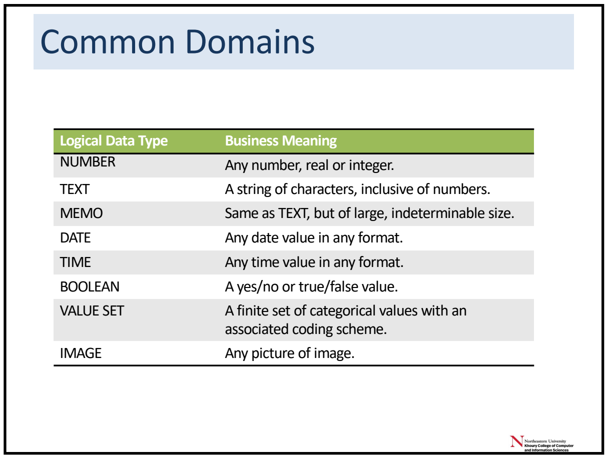
Attribute Domain

- The *domain* of an attribute is the set of legal values that an attribute can have.
- Typically a data type, such as:
  - – number, text, date, currency
  - – category: FALL, SPRING, SUMMER
- Domains can be defined without tying ourselves to implementation details.



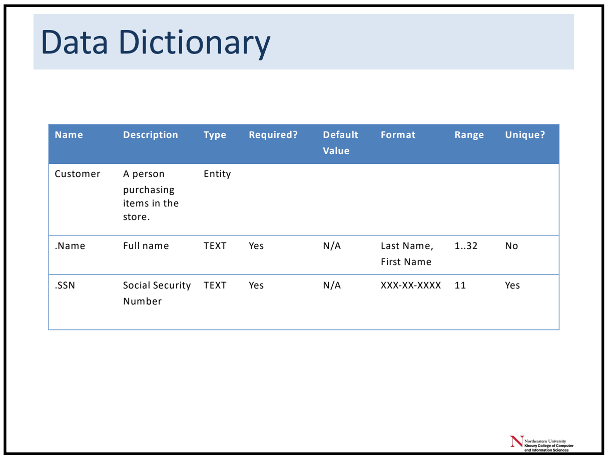
Types of Domains

- Simple text specifies characters to be included (or excluded, such as lower case).
- Formatted domains follow format rules (phone numbers, SSN).
- Enumerated domains list a set of possible values (abbreviations of US state names).
- Numeric domains describe a range of values and presentation formats.



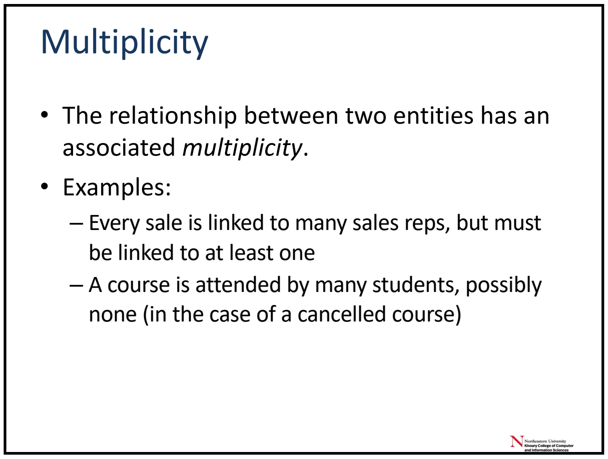
Common Domains

Logical Data Type	Business Meaning
NUMBER	Any number, real or integer.
TEXT	A string of characters, inclusive of numbers.
MEMO	Same as TEXT, but of large, indeterminable size.
DATE	Any date value in any format.
TIME	Any time value in any format.
BOOLEAN	A yes/no or true/false value.
VALUE SET	A finite set of categorical values with an associated coding scheme.
IMAGE	Any picture of image.



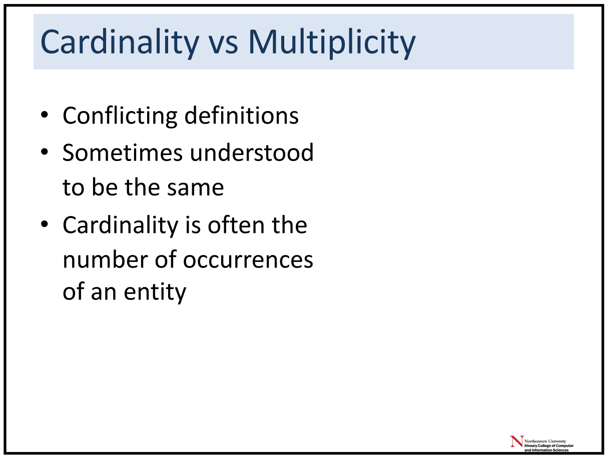
Data Dictionary

Name	Description	Type	Required?	Default Value	Format	Range	Unique?
Customer	A person purchasing items in the store.	Entity					
.Name	Full name	TEXT	Yes	N/A	Last Name, First Name	1..32	No
.SSN	Social Security Number	TEXT	Yes	N/A	XXX-XX-XXXX	11	Yes



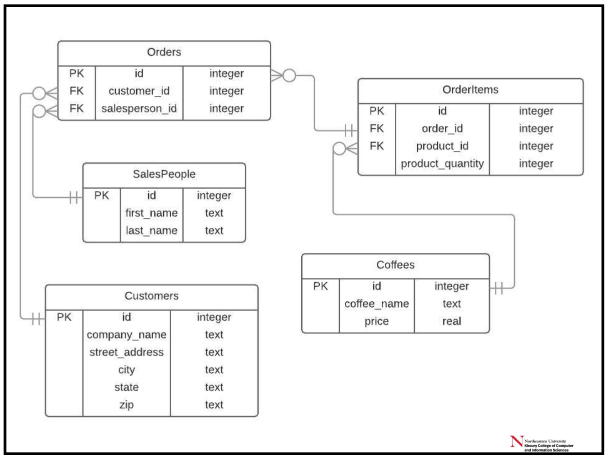
Multiplicity

- The relationship between two entities has an associated *multiplicity*.
- Examples:
  - Every sale is linked to many sales reps, but must be linked to at least one
  - A course is attended by many students, possibly none (in the case of a cancelled course)



Cardinality vs Multiplicity

- Conflicting definitions
- Sometimes understood to be the same
- Cardinality is often the number of occurrences of an entity



Orders

PK	id	integer
FK	customer_id	integer
FK	salesperson_id	integer

OrderItems

PK	id	integer
FK	order_id	integer
FK	product_id	integer
	product_quantity	integer

SalesPeople

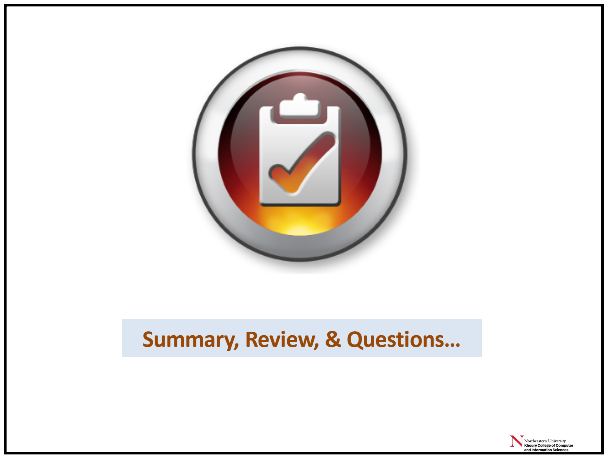
PK	id	integer
	first_name	text
	last_name	text

Coffees

PK	id	integer
	coffee_name	text
	price	real

Customers

PK	id	integer
	company_name	text
	street_address	text
	city	text
	state	text
	zip	text



Summary, Review, & Questions…