Database Theory and Applications for Biomedical Research and Practice

BMIN 502 / EPID 635
Week 1: Introduction and simple database design

John H. Holmes, PhD

Institute for Biomedical Informatics

Objectives for today

- You will learn:
  - The difference between data, information, knowledge, and wisdom
  - About the systems approach to data and information
  - The basics of database architectures and what informs them
  - The basics of data modeling

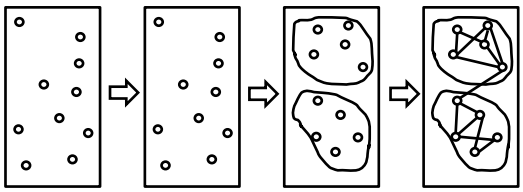
First things first…

What is (are) data, anyway???

The Information Spectrum

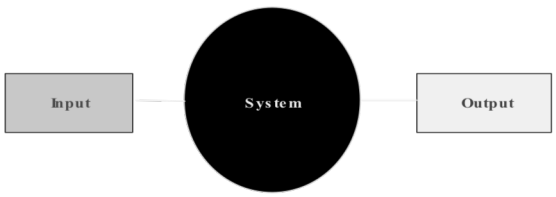
Data Information Knowledge Wisdom

Another way of looking at this



Data Information Knowledge Wisdom

A Systems Approach to Information



Information Systems

- Typically centered around a database
- Provide a means for input of data
- Provide some mechanism for data manipulation
- Provide a means for data output
- May include hardware and software
- Usually involve users in some way

Information Systems: Examples

- Databases (of all kinds)
- Patient records
- Laboratory information systems
- PACS
- Point-of-care systems
- Billing systems
- Registries

You can't have information if you don't have data!

Considerations about Clinical Data

Forms of Clinical Data

- Structured
- Unstructured
- Image
- Waveform

Types of clinical data: Structured Data

- Typically represented as discrete values in a fixed format
- Can be obtained from instrumentation
  - Laboratory, diagnostic, monitoring, etc.
- Can be selected from possible response options
  - Checkboxes, pulldowns, lists, etc.
- Examples
  - Demographics
  - Lab values
  - Medications
  - Disposition

Types of clinical data: Narrative Data

- Admission notes
- Histories and Physicals
- Progress notes
- Radiology reports
- Consult notes
- Chief complaints
- Discharge summaries
- Nurse's notes
- Patient-reported communications
- …

Types of Clinical Data: Images

- Illustrations
- Radiologic images
- Photographs

Types of Clinical Data: Waveforms

- ECG
- EEG
- Arterial pressure monitoring
- Actigraph monitoring

Functions of Clinical Data

- Basis of the historical record
- Communication between providers
- Anticipation of future problems
- Provide a legal record
- Support clinical research

All of these forms and functions (and there are others) pose challenges for database systems design, maintenance, and data analysis

Introduction to Ontologies

Some definitions (Source: ISO Standard 1087)

- Object
  - Any part of the perceivable or conceivable world
- Name
  - Designation of an object by a linguistic expression
- Concept
  - Unit of thought constituted through abstraction on the basis of properties common to a set of objects
- Term
  - Designation of a defined concept in a special language by a linguistic expression

More definitions…

- Terminology
  - Set of terms representing the system of concepts of a particular subject field
- Taxonomy
  - A classification, employing a terminology
- Nomenclature
  - System of terms that is elaborated according to pre-established naming rules
- Ontology
  - Explicit specification of a conceptualization

So, what are ontologies?

- Formally, an ontology is the statement of a logical theory
  - Definitions associate the names of entities in the *universe of discourse* (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms.
- Thus, ontologies are computable taxonomies

Let's look at an ontology used in a drug-related context…

RxNorm

National Library of Medicine

http://www.nlm.nih.gov/research/umls/rxnorm/overview.html

- Standardized nomenclature for clinical drugs
- Standard names for drugs linked connected to varying names of drugs present in many different controlled vocabularies within the Unified Medical Language System (UMLS) Metathesaurus
- Connections facilitate *interoperability* among computerized systems that process clinical drug data

RxNorm, contd.

- RxNorm is organized around normalized names, or *terms*, for clinical drugs
  - The terms contain information on ingredients, strengths, and dose forms
- The goal of RxNorm is to allow various systems using different drug nomenclature to share data efficiently

The scope of RxNorm

- Contains the names of prescription and many nonprescription formulations approved for use in the United States, including the devices that administer the medications

How do the links work?

- Links from clinical drugs (branded and generic) to:
  - Active ingredients
  - Drug components (active ingredient + strength)
  - Related brand names
  - National Drug Codes
- RxNorm links its names to many of the drug vocabularies commonly used in pharmacy management and drug interaction software and mediates between them
  - First Databank
  - Micromedex
  - MediSpan
  - Multum

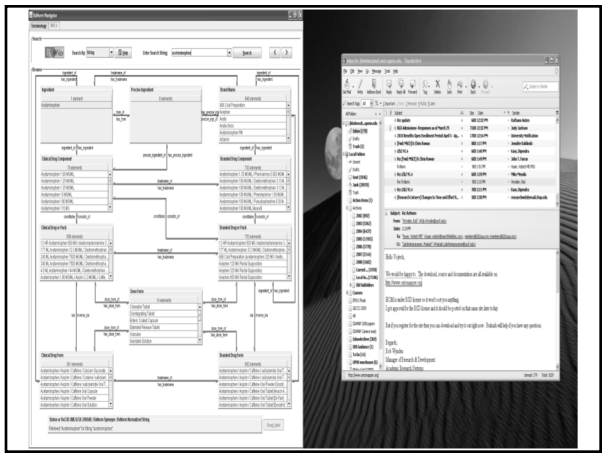
An example

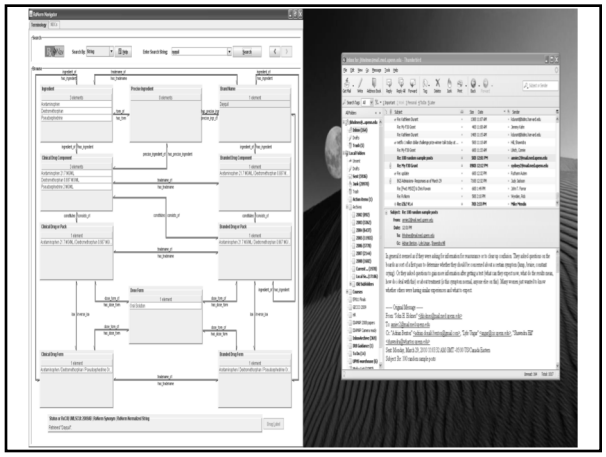
- Two preparations
  - Acetaminophen 500 MG Oral Tablet for a generic drug name
  - Acetaminophen 500 MG Oral Tablet [Tylenol] for a branded drug name
- Acetaminophen 500 MG Oral Tablet is related to Acetaminophen 500 MG Oral Tablet [Tylenol]
  - Both have relationships to Acetaminophen, Acetaminophen 500 MG, and Oral Tablet.
- Within the UMLS Metathesaurus, Acetaminophen 500 MG Oral Tablet and Acetaminophen 500 MG Oral Tablet [Tylenol] will each be linked to the different names that are used for these entities in other vocabularies

RxNav

- Java-based application that connects to the RxNorm server at NLM
- Allows users to query the RxNorm database by any of its components (ingredient, clinical drug, brand name, etc) and displays all the information related to a given component on a single page

https://mor.nlm.nih.gov/RxNav/





Introduction to Databases

What is a database?

- Logically coherent collection of data with some inherent meaning
- Databases are designed, built, and populated with data for a specific purpose, for an intended group of users
- Represent some aspect of the real world

Some database platforms

- Relational databases
  - Microsoft Access
  - FileMaker
  - Oracle
  - MySQL
- Hierarchical-relational hybrid
  - REDCap
- Graph
  - Blackfynn
  - Neo4j

The relational model

- First developed in 1969
- Data represented in:
  - Relations (tables), which are composed of:
    - Tuples (records, rows), which are composed of:
      - Columns (fields, variables)
- Facilitates
  - Rapid database design
  - Database design that reflects epistemology of the real world
  - Compact and efficient data entry and retrieval
  - Preservation of data integrity and security

A relation

ID	NAME	AGE	SEX
00001	Jones, Frank	90	Male
00002	Smith, Lottie	69	Female
00003	Mars, Manfrom	44	Male
...	...	...	...

Databases don't grow on trees…

They need to be designed, based on robust modeling methods

Introduction to Data Modeling

What is a model?

- Abstraction of a physical entity which makes it easier to grasp that entity's complexity and detail
- A model behaves like all or part of the physical entity it represents

Examples of models

- Architectural and engineering
- Process models
- Statistical models
- Data models

Types of models in building a database

- Conceptual
- Logical
- Physical

Conceptual models

- Narrative
- High-level description
- Not highly structured
- Not linked to implementation platforms

Logical models

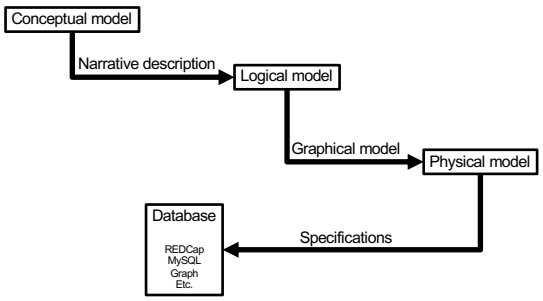
- Graphical
- High-level to moderately low-level
- Highly structured
- Not linked to implementation platforms

Physical models

- Low-level
- Highly detailed
- Wedded to specific implementation platforms

We will come back to these in a couple of weeks!

Typical database development



Logical data modeling concepts

- Entities
  - Things about which data are recorded
- Entity attributes
  - Static properties of entities (“fields” or “variables”)
- Relationships
  - Abstraction of associations between entities
- Entity instances
  - A concrete representation of an entity
- Schema
  - Text-based description of an entity

Thus…

- Entity
  - *A* patient
- Entity attributes
  - Study ID
  - Date of birth
- Relationship
  - A patient has one or more admissions
- Instance
  - *The* patient (John Smith, e.g.)
- Schema
  - We'll get to that next week!

The Entity-Relationship Model

- Logical model of
  - *Entities* represented in a data world
  - *Relationships* between those entities
- E-R models help us to understand
  - What we are collecting data about
  - What we are collecting
  - Characteristics of what we are collecting
  - How to avoid bad database designs

Entities

- Abstract "things" (people, places, events, etc.) about which data are recorded
- Examples
  - Patient (not "John Smith", but "a patient")
  - Admission (not the one on 5/10/2014, but "an admission")

Entity Attributes

- Equivalent to variables or fields
- Can be typed (text, numeric, date, etc.)
- Can be multiply instantiated within an entity instance (e.g., a patient can have >1 diagnosis)
- Can be used to identify unique records

Key Entity Attributes

- Candidate keys
  - Attributes or sets of attributes that can uniquely identify an instance of an object
- Primary key
  - The candidate key that was chosen to be the attribute or set of attributes that uniquely identifies an instance of an object
- Foreign key
  - An attribute of an object which is a primary key in another object

What makes a good key?

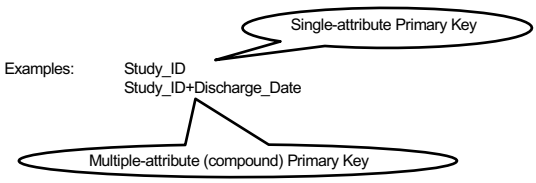
- Must be unique from instance to instance
- Can consist of more than one attribute!
  - E.g.: ID+DISCH_DX
    - To identify a unique instance of a diagnosis code for a specific subject, you need *both* the subject's ID number *and* the diagnosis code!

Some examples of bad keys

- Social Security Number
- Medical Record Number
- Subject Name
- More…

How do we identify a record in a database? Primary keys!

Primary key: The attribute or *set* of attributes that uniquely identifies an instance of an entity



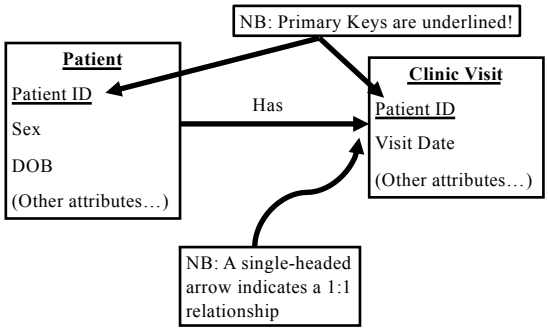
Relationships

- Abstraction of a set of associations between entities that holds systematically when applied to the real world
- Can be classified into two basic types, depending on the number of instances participating in each instance of the relationship (*cardinality*)
  - One-to-One
  - One-to-Many
  - Many-to-Many

One-to-one (1:1) Relationships

- A single instance of an entity can relate to one and only one instance of another
- *Example:* An individual patient can have only one clinic visit completed for a study

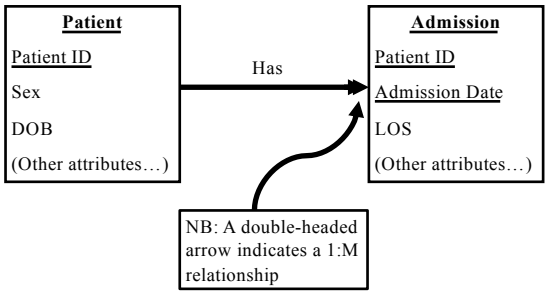
One-to-one relationship



One-to-many (1:M) Relationships

- A single instance of an entity can relate to one or more instances of another
- *Example:* An individual patient has one or more hospital admissions

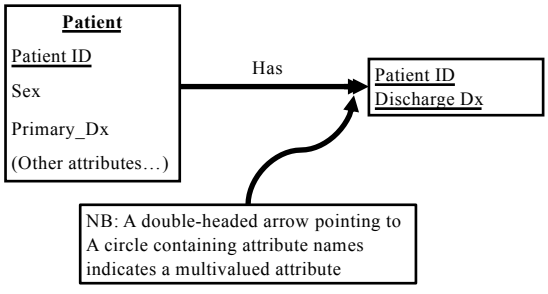
One-to-Many Relationships



Multivalued attributes

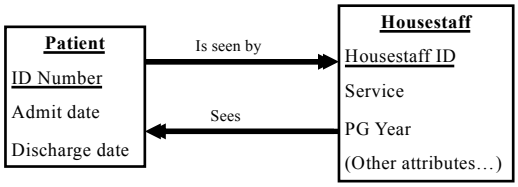
- A single *attribute* of an entity is repeated within that entity
  - Thus, a type of 1:M relationship
- *Example:* An individual patient can have one secondary discharge diagnosis, or many

Multivalued Attributes

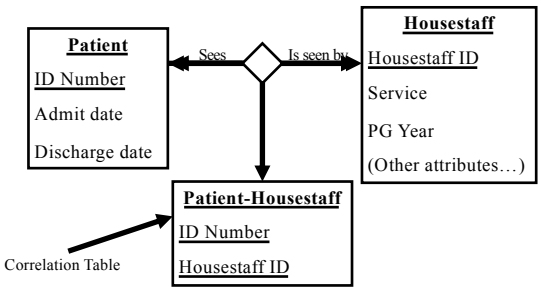


Many-to-Many Relationships

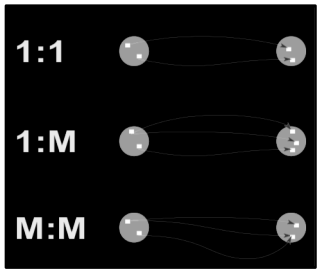
- A patient can be seen by one or more house officers
  Housestaff repeats for each patient!
- A house officer can see one or more patients
  Patient repeats for each house officer!



How to model the Many-to-Many relationship



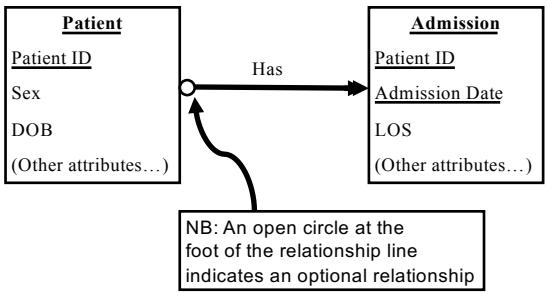
Cardinality: A Summary



Relationship Modality

- Used to indicate whether or not an entity must participate in a relationship
- Required: A patient *must* have at least one hospitalization
- Optional: A patient *may* have one or more hospitalizations

Modeling an optional 1:M



Why do we care about optional relationships?

During data entry, maintenance, and analysis, we want to know whether or not to expect an instance!

Composite attributes
A very thorny problem!

- Composite attribute: A single field or attribute that represents two or more concepts
  - Example: An attribute where the first two digits represent a study site, and the last three the subject ID: 01123
- The problem
  - In order to count the number of subjects at site 01, you would need to parse “01123”

So how do you create an E-R model???

- Make a list of all of the things about which you will collect data
  - These are your candidate entities
- For each candidate entity, list the attributes, identifying primary keys, multivalued attributes, and taking care to ensure all attributes are atomic
- Draw the relationships between the entities
  - Identify and name the 1:1 and 1:M relationships

Assignment 1:
Create an Entity-Relationship Diagram for the ABIC Registry

Submit as a Word or Powerpoint document to Canvas by 9am 1/29