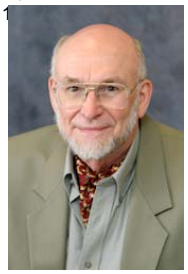
DAMA - Minnesota, 2016 June

GETITLE

1



Conceptual vs. Logical vs. Physical

Stages of Data Modeling

©Gordon C. Everest
Professor Emeritus of MIS and Database
Carlson School of Management
University of Minnesota
geverest@umn.edu http://geverest.umn.edu

Outline

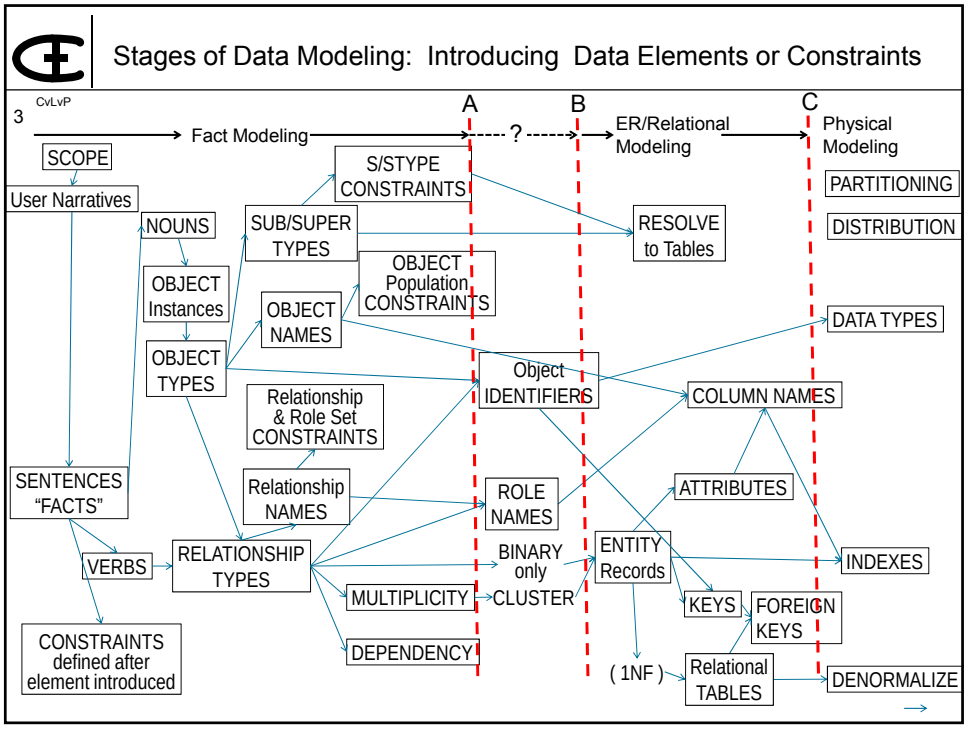
CvLvP

2

Goals of this presentation

[slide#]

N



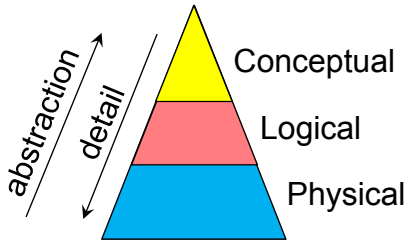
Common Understanding - Levels

4 CvLvP

See David HAY video

- **Conceptual** – high-level, enterprise-wide, abstract model
- **Physical** – How data is stored in some database system
- **Logical** – adding detail to the conceptual model, … free of physical implementation details which do not contribute to the logical understanding of the data model.
  - Often considered the ER or Relational Model.

Generally depicted as a pyramid, implying *levels* of models:

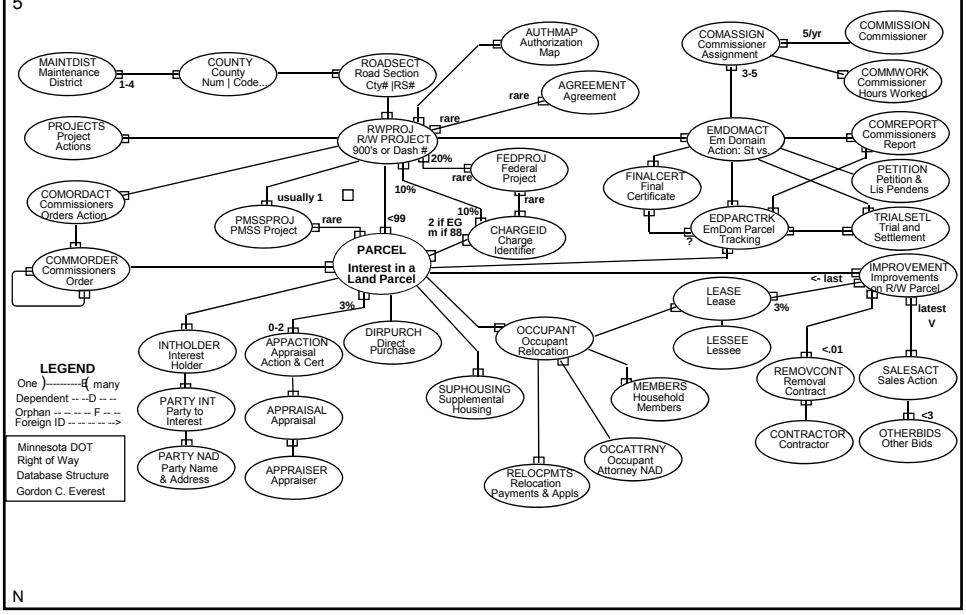


Let's look at the generic meaning of these terms, but first…

Conceptual Data Model? Logical?

DMODPRE

5



N

Conceptual Data Model?

DMODPRE

6

PARCEL

Interest in a Parcel of Land

N

Conceptual - Definition

7 CvLvP

-- Mirriam-Webster, Dictionary.com

CONCEPTUAL:

- Consisting of, relating to, concerned with… Concepts*; abstract.
- Concerned with the definitions or relations of concepts, rather than the facts.

Synonyms: theoretical, visual, imaginary.

Antonyms: real; facts.

*CONCEPT:

- an idea of what something is or how it works; something formed in the mind; a mental image.

If mental, how do we document, communicate?

If entity/object, relationship, identifier, domains – already logical?

If add attributes, foreign keys – now Relational.

Logical - Definition

8 CvLvP

LOGICAL

- Of or according to the rules of logic or formal argument; characterized by or capable of clear, sound reasoning.

Synonyms: natural, reasonable, sensible, understandable*

Logical Data Model – a model of some user domain**
complete and understandable in the detail needed to represent that domain, built according to and consistent with some formal modeling scheme, within a defined scope.

*Understandable - defined, documented, communicated.

**area of the business being modeled
- real world, user world, domain of discourse, subject area, …

Physical Data Model

9 CvLvP

- How data will be encoded and stored
- Implemented in some data system (DBMS, NoSQL…)
- Dealing with storage & processing performance, volumetrics (time & space), partitioning, distribution.

Physical vs. Logical separation

- Historically, to better understand physically stored data existing on punched cards, tape, etc. the notion of a *logical* representation was introduced to strip away storage considerations and focus on documenting just the logical aspects of the data.
- Logical derived from, a representation of… the Physical

Physical Data Model

– a stored representation of a Logical data model

Abstraction

10 CvLvP

ABSTRACTION* = "leaving something out"; Hiding

In Designing/Developing a Data Model: (can't do it all at once)

- Start with high-level preliminary sketches (top down)
  Details are still presumed to be present, yet to be added
- Work on one part or subject area at a time
- Could also start with some details (bottom up)
- Once built, (how) do you maintain the Conceptual Model? Useful?

In Presenting a Data Model: (already completed in all its detail)

- Start with a high-level view, then successively add detail
  –> VERTICAL ABSTRACTION
- One part at a time –> HORIZONTAL ABSTRACTION

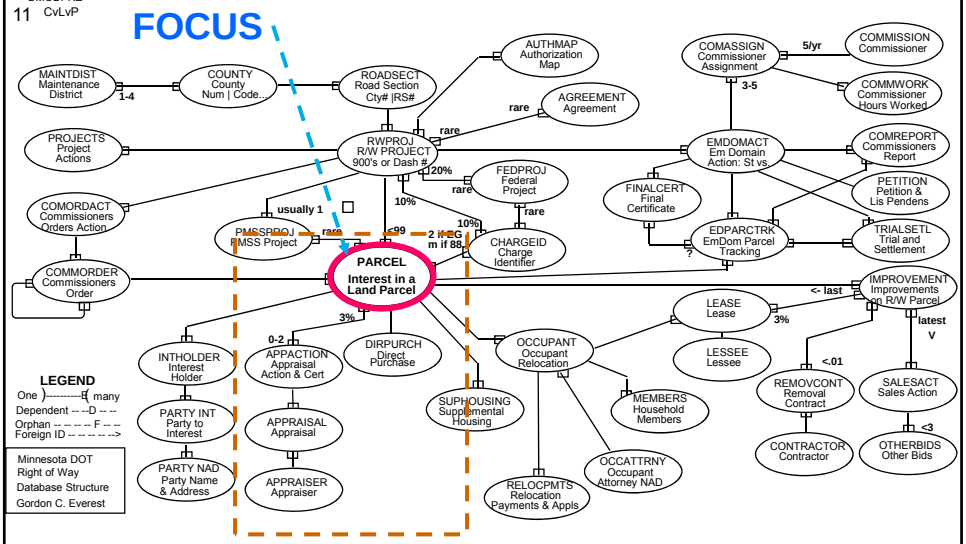
*Webster Dictionary:

abstract. (*n*) summary; shortened version. abstraction. (*n*) the act of taking away
(*v*) to take out, remove something.
(*adj*) as in abstract object (vs. concrete) - a different meaning, not useful here.

N

Sample Data Model

DMODPRE
11 CvLvP



What could improve this presentation?

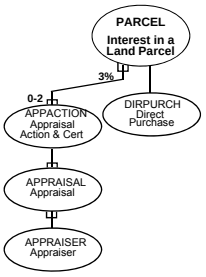
N

2.a HORIZONTAL ABSTRACTION

DMODPRE
12 CvLvP

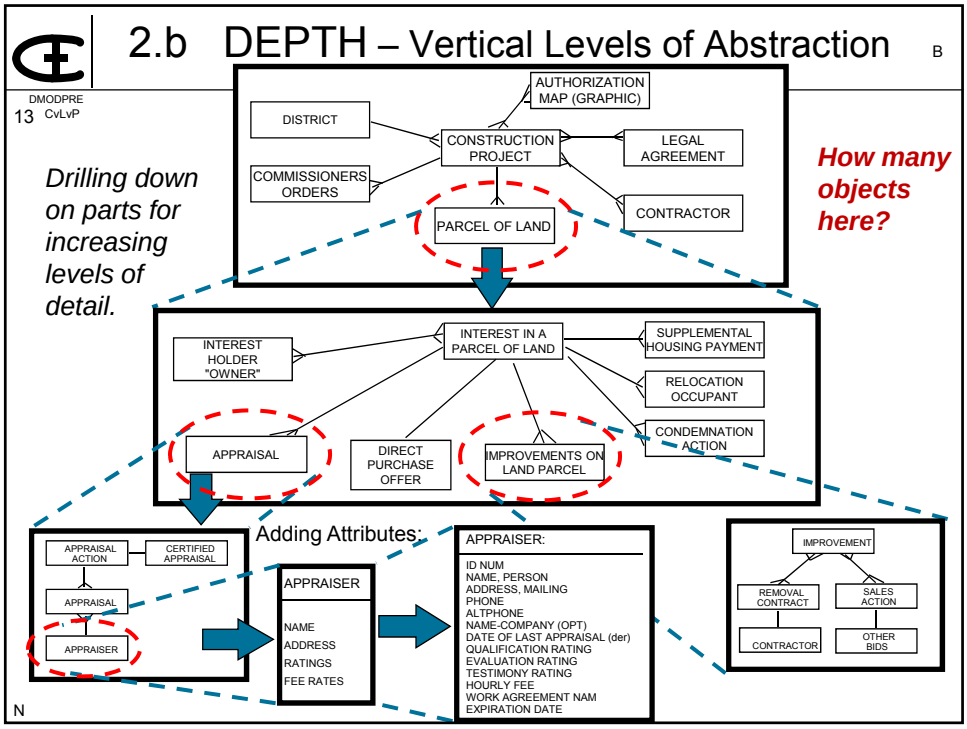
- Partitioning

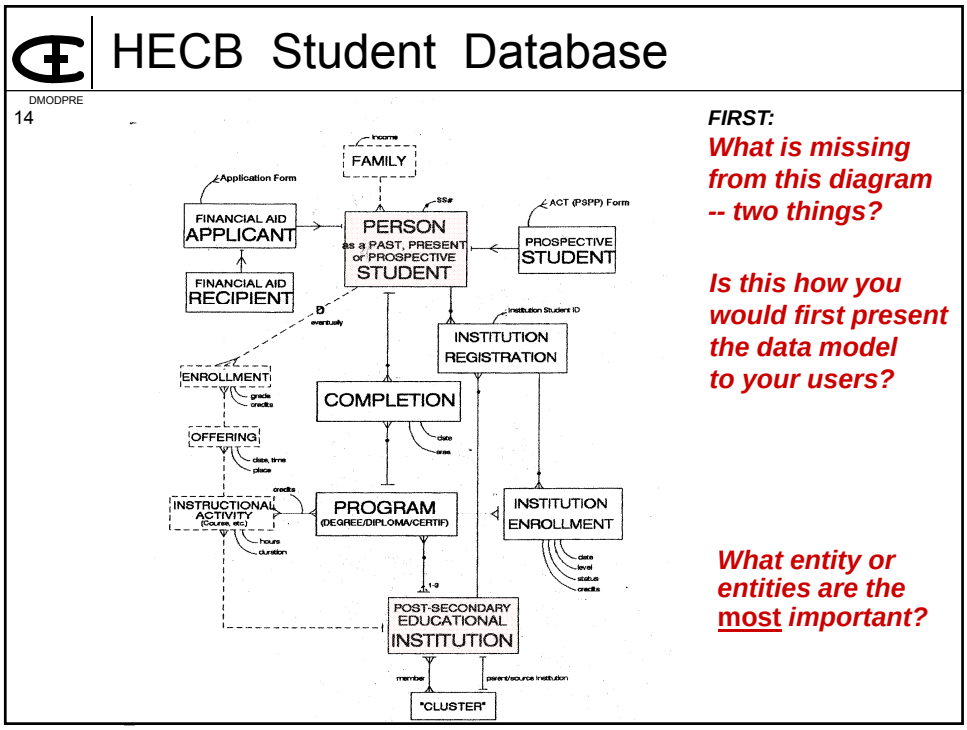
- Fencing off a part of the Diagram:
  - Often helpful to have some overlap of the partitions



Stages of Data Modeling

Presentation to DAMA, Minnesota, 2016/06

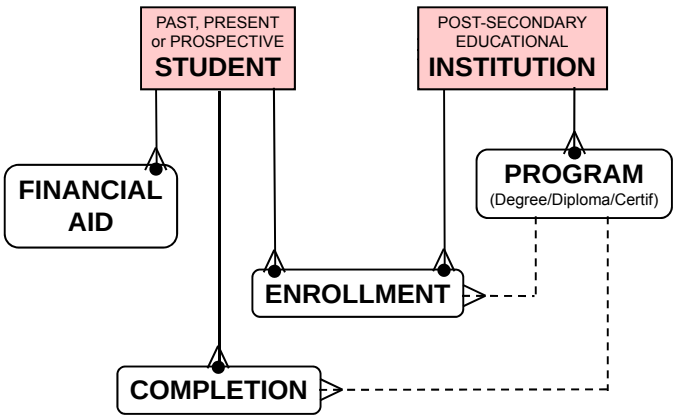




HECB Student Database

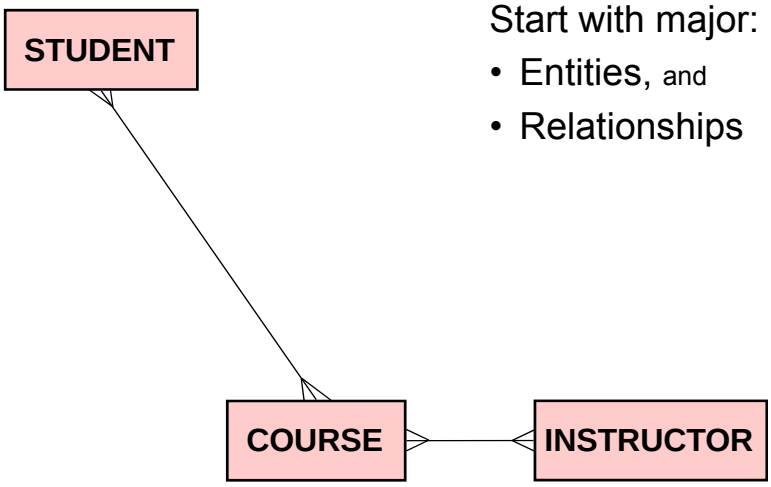
DMODPRE
15

Unfolding detail from the most important:



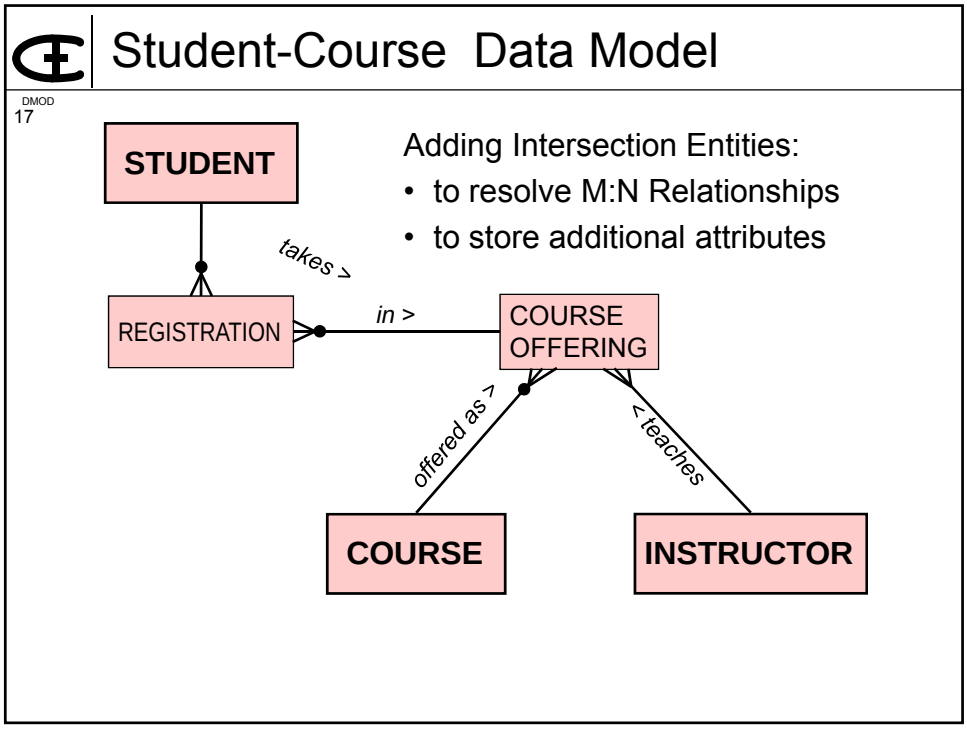
Student-Course High-Level Data Model

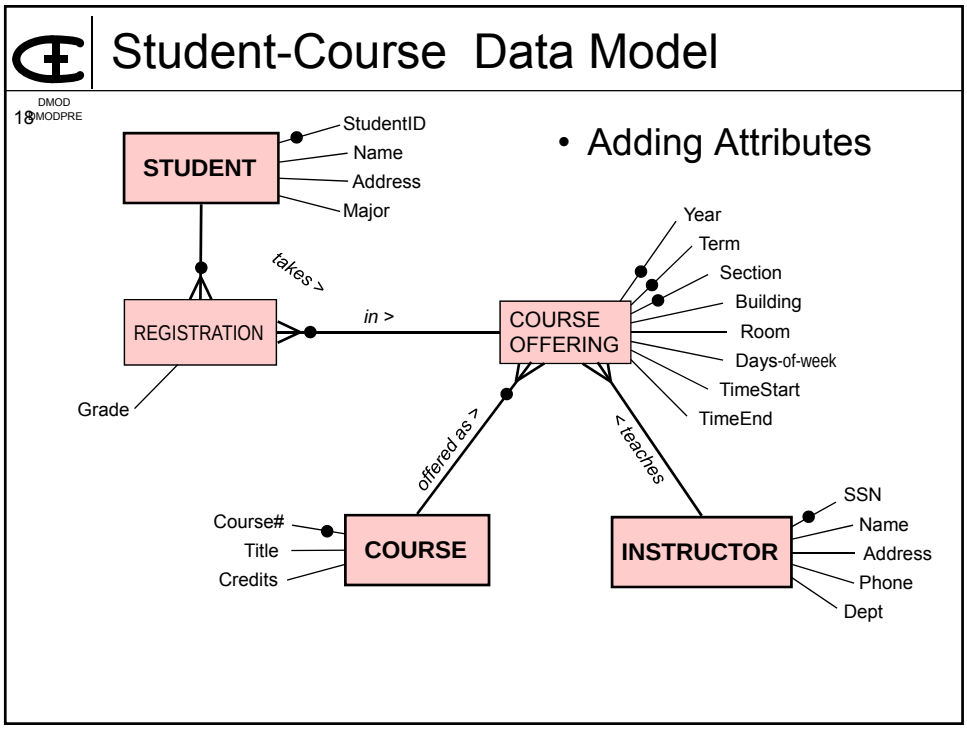
DMOD
16

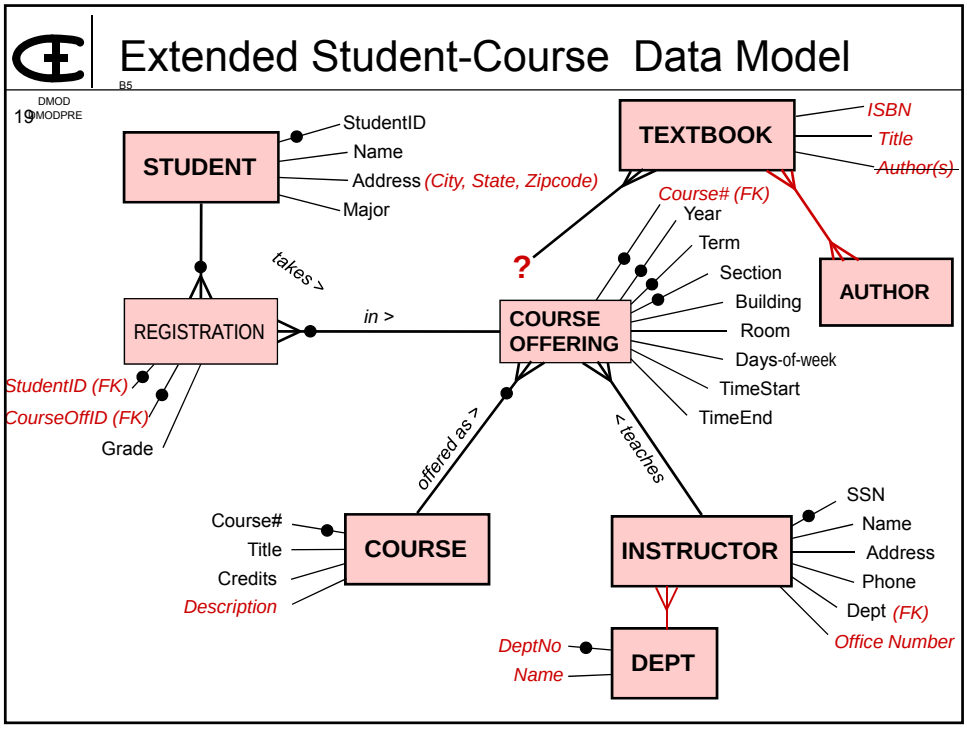


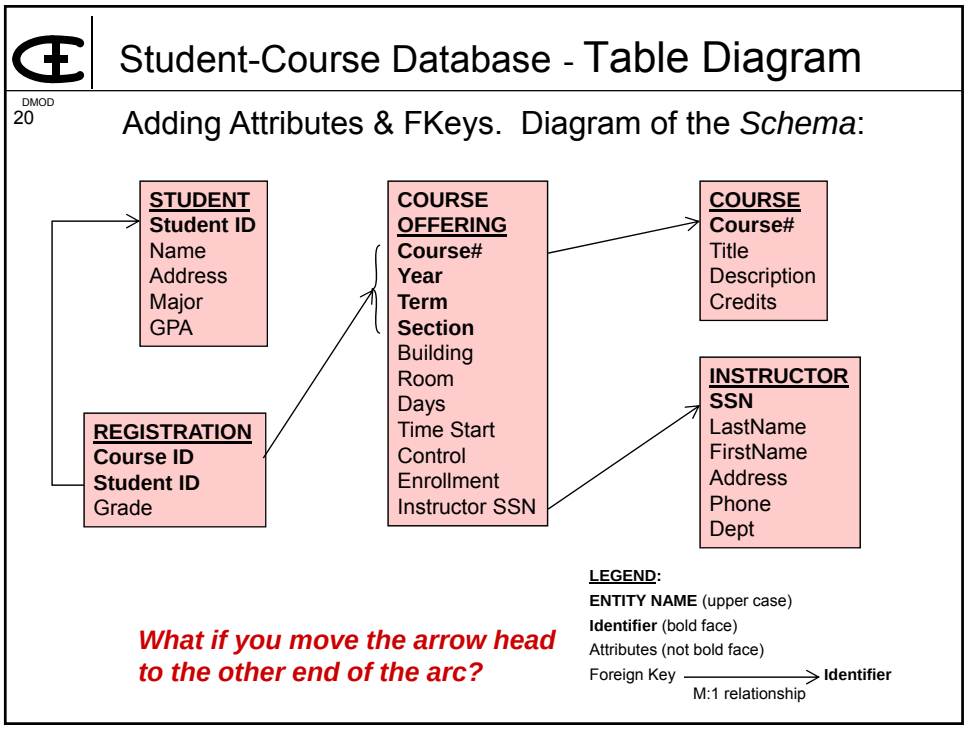
Start with major:

- Entities, and
- Relationships



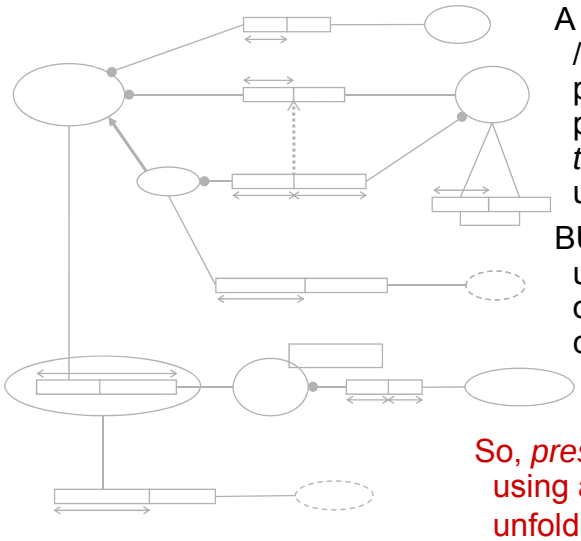






ORM Data Model - Presentation

DMODPRE
21



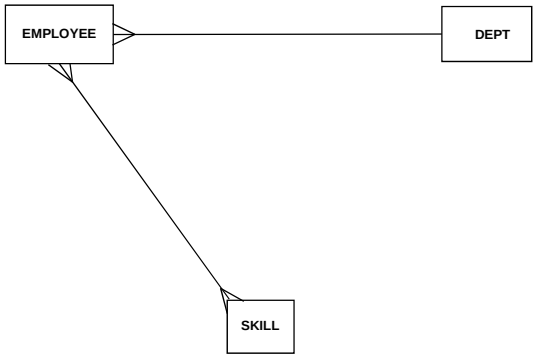
A major criticism of NIAM / ORM, both by protagonists and proponents, is that it is *too detailed*, a bottom-up design,

BUT… ER Diagrams usually *hide* the details of attributes and most constraints.

So, *present* the ORM model using a series of top-down unfolding … *abstractions*.

Abstractions of ORM Data Model

DMODPRE
22



1. Hide "Terminal" (M:1) Objects (=> Attributes)
2. Hide Reference Modes
3. Hide Constraints
4. Hide Less Important Objects & Predicates
   - Subtypes
   - Objectified Predicates
   - Reflexive Relationships
5. Hide all Predicates

   Leaving BASE Entities!
6. Add back Multiplicity char. on relationships

=> A High-level Abstract "Conceptual" Data Model...

Is this the same data model we started with?

an ER Diagram ?!!!

Levels (or Stages?) of Data Models

DMOD
23 CvLvP

- Reality - the real world User Domain, infinitely complex
- Mental Model - in our minds
  - must be formally documented so we can communicate it to others
- Conceptual Model - "natural", unconstrained, initial.
  - independent of physical storage and implementation
- Logical Model - according to a modeling scheme
  - e.g., the E-R or Relational Model (most popular today)
- Physical Model - defining storage characteristics
  - Encoding, storage structure and access methods (indexes, etc.)
- Implementation Model - for a given DataStore Manager
  - memory organization (blocking, buffering, partitioning, distribution, etc.)

Objective of Data Modeling

DMOD
24

(*WHAT* we are trying to do)

TO ACCURATELY AND COMPLETELY MODEL

SOME PORTION OF THE REAL WORLD
UNIVERSE OF DISCOURSE (UoD)
(the USER DOMAIN)

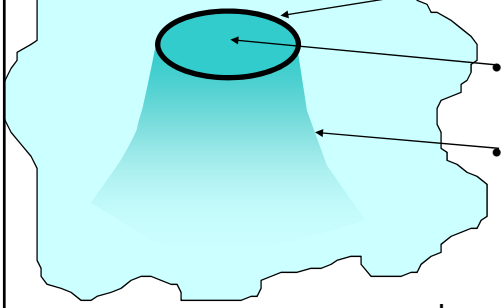
OF INTEREST TO SOME ORGANIZATION
OR COMMUNITY OF USERS.

Modeling: is Choosing...

DMOD 25

REALITY is Infinite, Complex, Multidimensional, Detailed.

- so we must *CHOOSE*:



- SCOPE / Boundary
  - where to look
- FOCUS
  - what to look for
- DEPTH / Resolution
  - how much detail to look for

... based upon our *PURPOSE*

A Model is an Abstraction

DMOD 26 CvLvP

TWO PERSPECTIVES of Data:

Abstract "Conceptual" View of the Real World

Mental Model

"Logical" "DATA" MODEL

Physical (Storage) Model

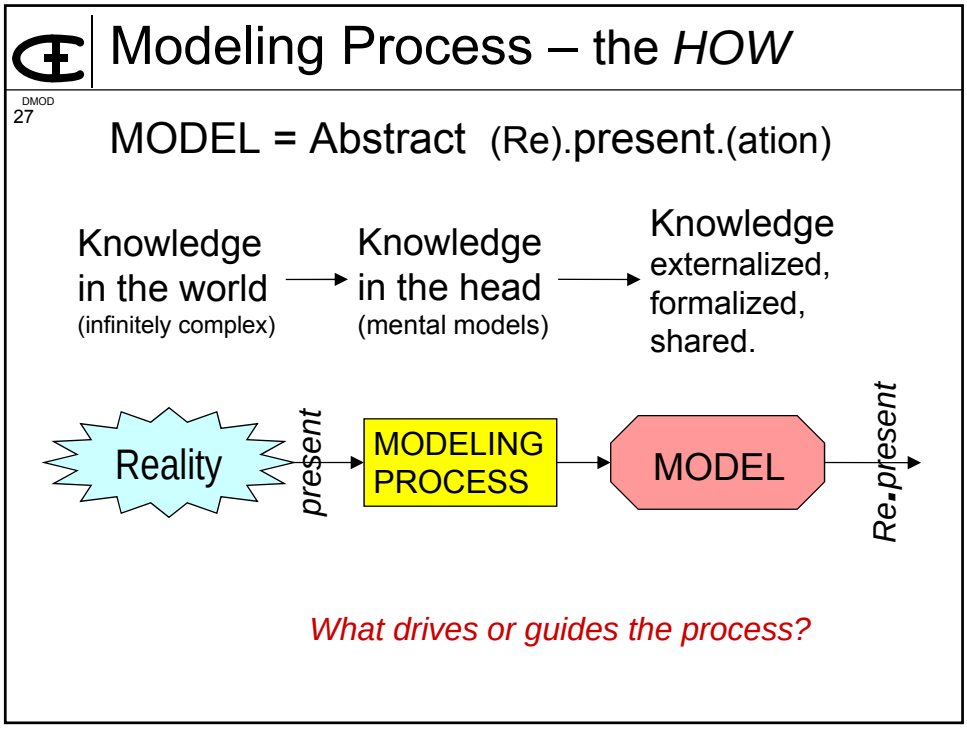
Concrete Symbols Stored on some Medium

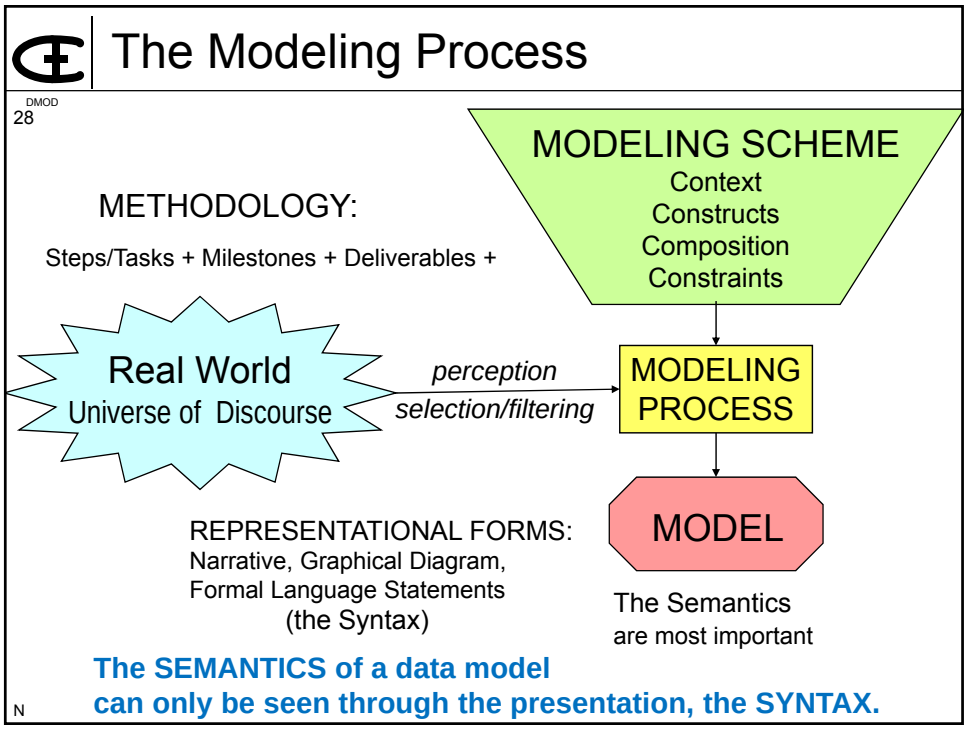
REALIZATION

Both realities are infinitely complex.

NEED some constructs to look for and use in modeling.

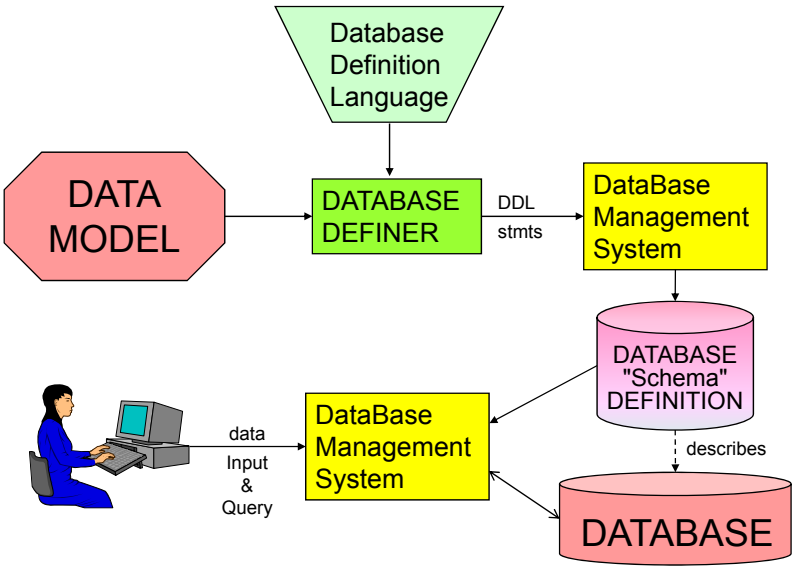
Sometimes we have the data, and try to find what it means.





Data Model to Database Realization

DMOD
29



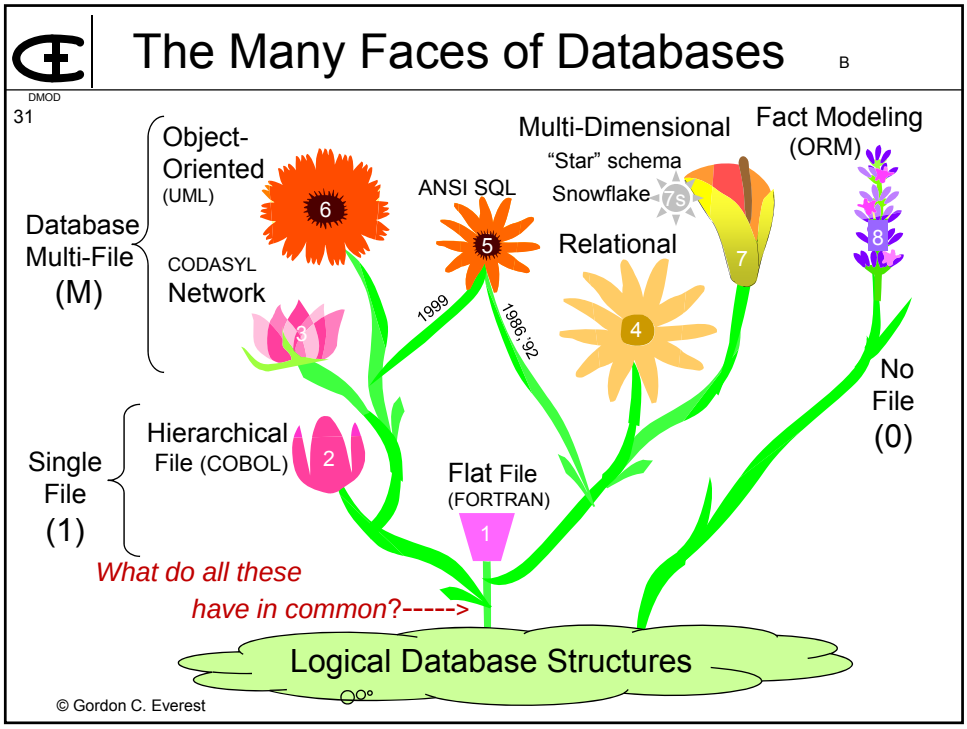
Data Modeling Schemes

CvLvP
30

- *ALL* data modeling activity and data management tools are driven or guided by some Data Modeling Scheme
- Think of it as a Meta Model (or Meta-Meta-Data)
- Tells you what to look for, what constructs to use, how to put them together (compose) with what constraints, and how to represent that all syntactically.
- Logical Rules however formal or informal
- May be developed independently of any implementation Not based on any particular implementation
- Many variants within families

Since all data modeling is driven by some modeling scheme, i.e., by some logical rules for building a model,

All data models are logical models!



Data Modeling Schemes – Types & Examples

32 CvLvP

- Developed along generation lines

SCHEME	Examples
Flat file	Fortran (1956?), spread sheet (VisiCalc, Multiplan>Excel, Quattro)
Hierarchy	COBOL*(1960), System 2000, HQL*
Network O-O ext.	CODASYL*(1971), IDMS, ANSI-NDL*, IMS (DL/1), Adabas OO-COBOL*, ANSI-SQL:1999*, UML*
E-R	Chen*(1976), IE*(Finkelstein), Barker, IDEF1X*(ERwin), ER Studio,
Relational (SQL)	Codd*(1970), SEQUEL*(1976), Oracle (1979), DB2, ANSI-SQL*(1986), Sybase, SQL Server, Dbase II
(Inverted)	Fully indexed - not really a logical scheme, Model 204, CASE 360(IBM)
Dimensional	As a "Cube": EXPRESS(6) (MDS>IRI>Oracle), Multiplan(3), MicroStrategy As a Relational Model = Star Schema*(R. Kimball), Red Brick
Fact-Based	NIAM*(1976), ORM*(1989), NORMA, FCO-IM
NoSQL	A family of tools to overcome the limitations of SQL tools. Each tool has its own modeling scheme – key-value pair, columnar, document (XML, Hierarchical), graph (nodes & edges).

*initially not an implementation but a concept paper or language specification

Modeling Schemes in NoSQL Tools

CvLvP
33

- NoSQL refers to a family of tools designed to handle Big Data, "Unstructured" Data, Fast(er than SQL tools)
- Based on particular data *storage* schemes
- Vendors have augmented their Relational/SQL tools with some of these storage schemes, and other models
- One driver is OO programming languages which handle objects of varying structure and complexity. Mapping OO to a relational structure is inadequate.

Physical Scheme	**Examples**
Key-Value pair Value - any complex structure	Dynamo, Redis, Riak, LevelDB - an index
Graph (O_1 O_2 R – triplets)	Neo4j, OrientDB, Infinite Graph, Mark Logic
"Document" (XML, JSON)	MongoDB, CouchBase, Mark Logic
(Wide) Column stores Inverse ("dual") of Tables	Cassandra, HBase

Criteria for a Data Modeling Scheme

DMOD
34

- Simple, understandable – for human communication
- Comprehensive – can model every phenomenon in the user domain, e.g., overlapping populations **==>** generalization
- Direct – visually intuitive, unambiguous
  e.g., "Fork" for manyness ——<
  – without spurious, artificial, intermediate constructs
  e.g, intersection entity (for M:N), foreign key (redundant with an arc)
- Minimal – at most one way to model a given phenomenon
- Consistent – uses same syntax for similar phenomenon
  e.g., for dependency within a record, between records, and S/Stypes
- Universal – independent of language

Outward Facing vs. Inward Facing

35 CvLvP

Outward Facing – to the business user domain

Inward Facing – to existing, stored data

- Historically we had data on punched cards or paper tape, and needed a representation which transcended its physical storage, hence, logical data models. Still inward facing.
- Next we found logical models too complex and needed to simplify, particularly at the beginning stages of development, hence conceptual data models, even before logical models.
- Then we realized that these models were really representations of things in the business user domain, hence outward facing.
- The modern approach to data modeling is to begin by modeling user domains independent of any physical storage or implementation considerations.
- More recently, we collect massive amounts of data (BIG data), it exists. Now the challenge is to process it efficiently (hence NoSQL tools), and apply analytics to make sense of it.

NOTE: Someone designed the stored data, so where are the definitions?

Data Model – Outward Facing

36 CvLvP

Initially a data model is outward facing, to the business

- Whether modeling big data, fast data, thick data, NoSQL data, Relational data (SQL) …
  (these are all representations for physical implementation)
  you still need to know, understand and document the business.
- The "first stage" data model is a fully detailed model of the business independent of any physical implementation

  BUT… capturing rich, detailed semantics which describe the user domain in the model.

"Data" Modeling is NOT about:

37 CvLvP

- Scope – what do we mean by "enterprise-wide"?
- Simplified, high-level, abstract, "conceptual"
  - Whether in data model development or
  - A matter of *presentation*, choosing to hide detail.
- Syntax – chosen notation to represent a Data Model
  - Data Modeling is about Semantics – meaning
  - The same semantic can have several notations
    - e.g., multiplicity in a relationship: --<(fork), 'M', *, -->>
- Storage, Physical Implementation, Performance
  - Only exogenous information to represent the user domain
  - No unnecessary, artificial, spurious constructs introduced on the path to implementation, e.g., FKey, 1NF, entity records!
  - User-facing, NOT database/datastore-facing

Though these are all important aspects of Data Modeling.

What to Call our model?

38 CvLvP

- Conceptual is a scoping and presentation issue
- Physical is not part of the Data Model
- Logical is what we are left with.
  But all models are logical*!*
- Data Model - but not always a model of data,
  particularly when the database has not yet been built

None of these adjectives are helpful when referring to data models so…

How do we distinguish types of data models?

What do we call the initial complete data model?

A Business Data Model

39 CvLvP

For our initial but complete, detailed data model

- Capturing *all* exogenous* information about the user domain which is of interest
- Capturing *only* exogenous information about the user domain
- Our mental models need to be externalized, and formally documented to be communicated.
- Hence, we need a modeling scheme with a rich Syntax to represent the Semantics of the user domain in the model
- Devoid of anything relating to physical storage, technology, encoding, implementation, etc.

Let's call it a "Business Data Model" (G. Witt)
Halpin calls it the Conceptual Data Model.

* relating to, developed or derived from external factors; originating from outside

Introducing Design Elements

40 CvLvP

As we move through the continuum of:

Conceptual ==> Logical ==> Physical

- How to rationalize the many differences and alternatives in logical data models?
- Logical data models differ based on which modeling elements are included in the model
  i.e. the modeling scheme

SO

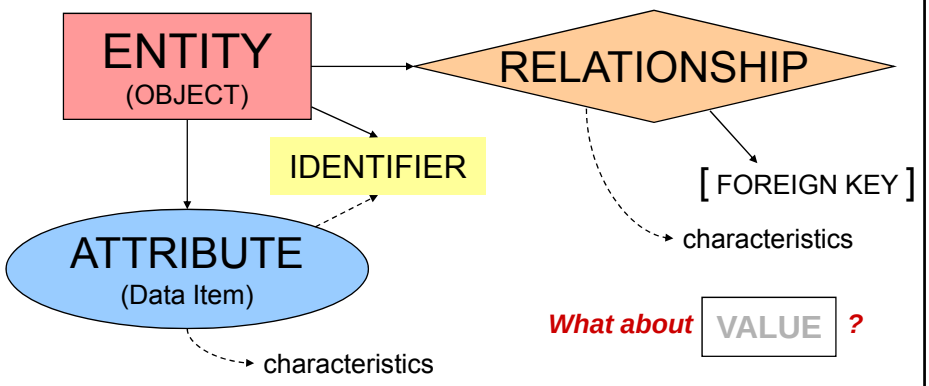
- Let's lay out the various design elements in a precedence graph reflecting order of introduction

Data Modeling Constructs

DMOD
41

What to look for: Relative emphasis differentiates Data Modeling Schemes

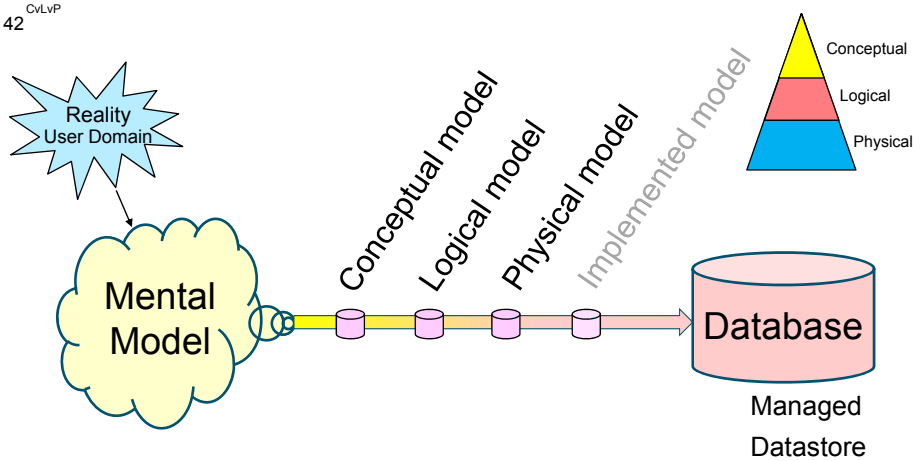
- **ER modeling** focuses on Entities and Relationships,
  de-emphasizing, even hiding Attributes.
- **Relational** (restricted ER, 1NF) focuses on Entities and Attributes,
  relegating Relationships to Foreign Keys.
- **Object Role Modeling (ORM)** folds Attribute and Entity into Object



N

Traditional "Levels" of Data Models

CvLvP
42



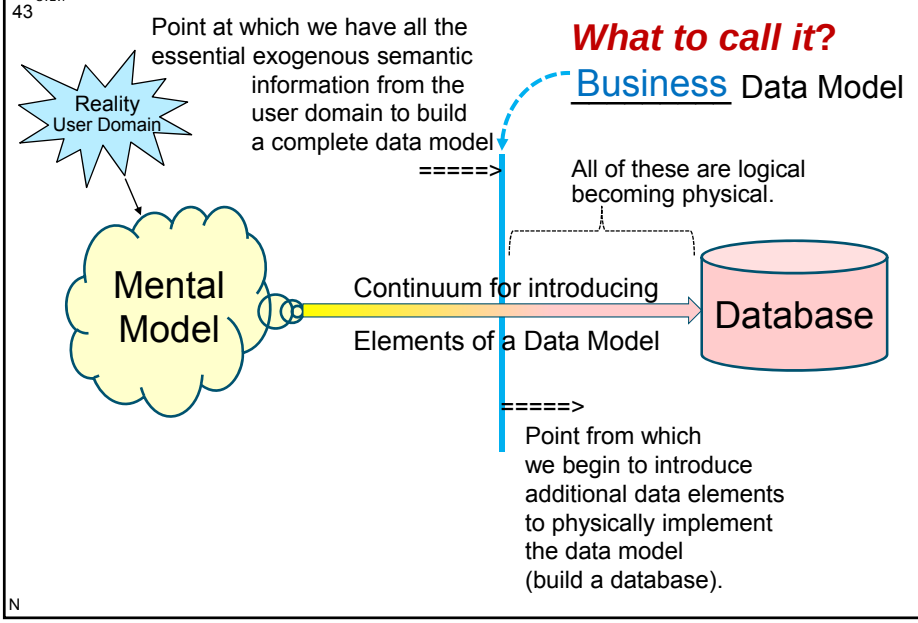
Are intersection/associative entities or foreign keys part of the logical model or the physical model?

Let's forget levels, and
focus on the ordering of design elements

N

Introducing Elements of a Data Model

CvLvP 43



N

Data Modeling Constructs

CvLvP 44

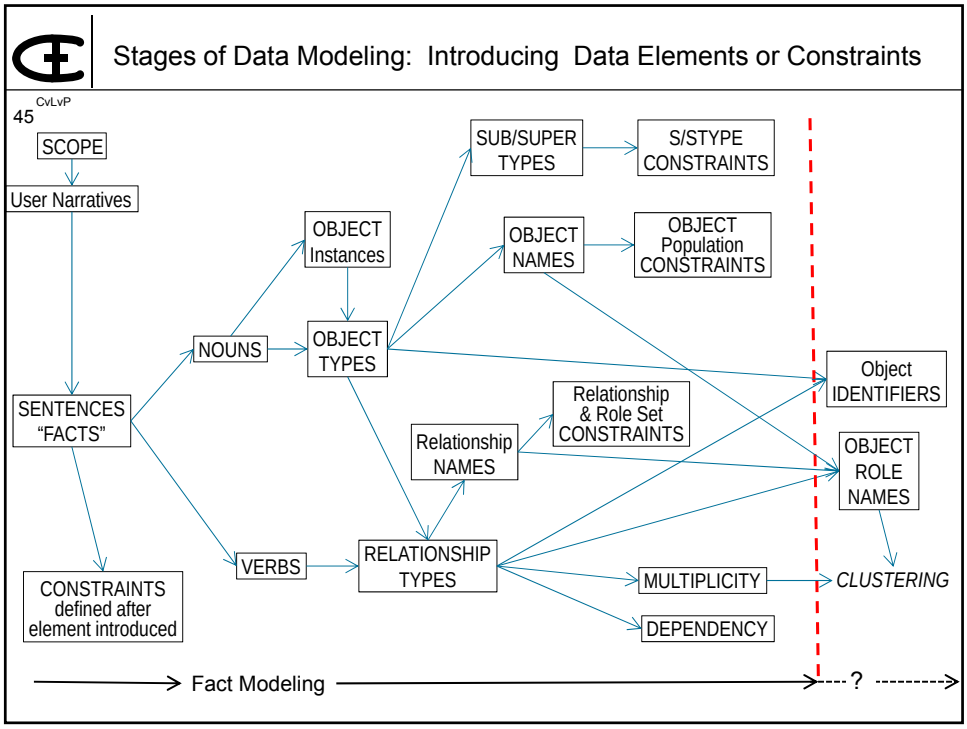
- "Things" – objects, entities, attributes
  - Names of things
  - Populations (or types) of things, Domains (of values)
  - Subtypes/Supertypes to model overlapping populations
- Relationships
  - Names of relationships, Names of Object Roles
  - Characteristics/constraints (dependency, multiplicity)
  - Ternary+++ relationships

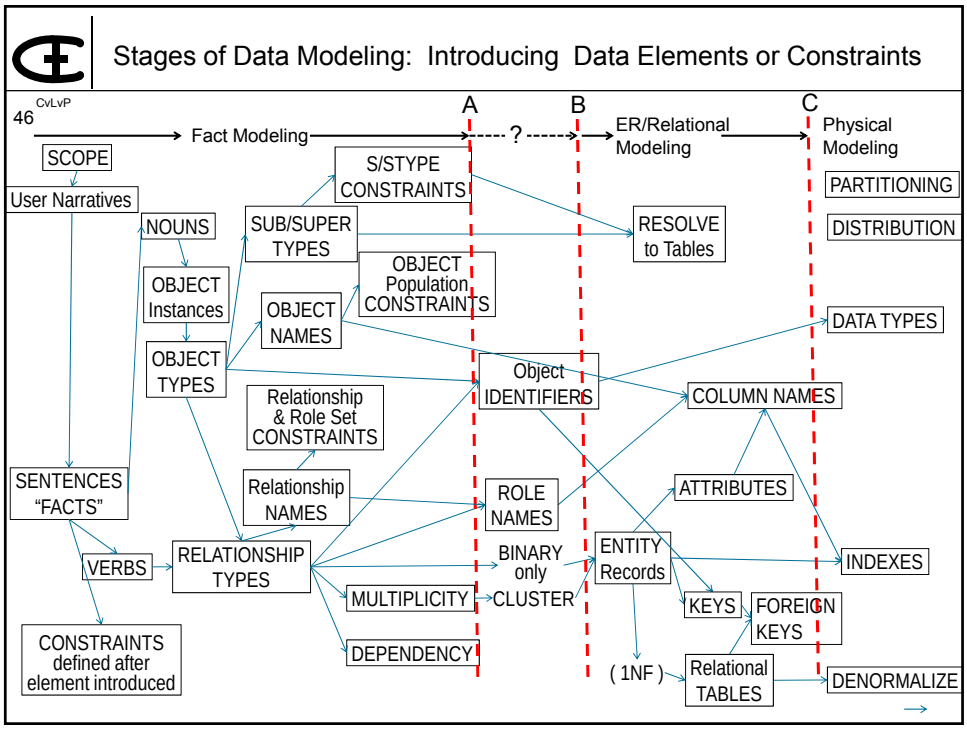
Business Data Model.
moving to implementation

- Entity records
  - Clustered attributes (based on relationships)
- Identifiers
  - Encoded representation of instances of things (IDs)
  - Keys, Foreign keys

Stages of Data Modeling

Presentation to DAMA, Minnesota, 2016/06





Observations on the Diagram

47 CvLvP

- Showing the precedence ordering of the introduction of data modeling elements
- At some point we have all the exogenous semantic information needed to complete the model. Up to that point our model is outward or user-facing
- Anything introduced later is physical realization or implementation, i.e., inward facing.

Stages:

- Fact Modeling
- >> controversial elements in between
- ER / Relational Modeling
- Physical Modeling
- Implementation Modeling

Start with User Narratives

48 CvLvP

Begin with Statements from User Domain Experts (SMEs) within a defined, agreed upon scope; they are the primary source of knowledge about the world being modeled. Then find the Model Elements

- Analyze the Vocabulary in the User Narratives
  – develop agreed upon definitions
- Breakdown user narratives
  => into elementary fact sentences
- Extract the nouns
  => become Things or Objects
- Extract the verbs
  => become Relationships
- Extract other words/phrases
  => become Constraints

Verbalize User Descriptions

noun
verb
constraint

ORMODLG
49

GIVEN A DESCRIPTION FROM THE USER(S):

Famous Foods, a small, specialty food wholesaler, fills orders for restaurants. Customers have names, addresses, etc. An order can include several products. Products have unique SKU numbers, descriptions, manufacturer, etc. The company has one big warehouse with many rooms on several floors. Each product is stored in only one bin location in the warehouse, but it can change frequently. Multiple products may be stored in the same bin. Bin numbers are only unique within a room, hence the same number can be used in different rooms. Since the bin locations can be hard to find in a room (could be on a shelf, on the floor, in a cabinet or cooler, hanging from the ceiling, etc.), and the rooms can be hard to find in the warehouse (with many hallways, doors, tunnels, split levels, mezzanines, etc.), explicit location directions must be recorded for each room and for each bin in the room. Location information is a textual narrative and is used by the pickers who run around gathering the items to fill an order. Each product has its own standard price but it may be modified by applying a discount (a fraction) on any individual order. The discount can be different for each of the products on an order, and for the same product on different orders. The quantity of each product on an order is recorded (it is not the quantity on hand or in inventory). Terms indicates the number of days during which a standard discount can be taken on the payment. The terms can vary from one customer to the next, and from one order to the next for the same customer.

Establishing the Vocabulary

CvLvP
50

Before we can develop a data model we must first carefully define our terms so we can talk about it

- From a business perspective
- By the user domain or subject matter experts
  - Listen to what they say, talking about the domain
- Initially will be fuzzy, with areas of disagreement
  - Requiring some discussion and negotiation to come to a common understanding; and documenting that
  - The most difficult and important aspect of data modeling

Call it a business glossary or [data] dictionary?

However, a glossary is usually only for nouns (objects).

We also need to define the relationships - the mortar that holds the bricks (nouns) together… and constraints.

N

Objects

CvLvP
51

- Encompass Entities,
  and Attributes… *independent of entities described*
- Derived from the nouns in the user narratives
- A single instance (of what population?)
  … or a population of individual instances
  - e.g., given the noun 'George': until it is associated with a particular, defined population, it is just a string of characters
- Each Object Population – uniquely named
- Define the population, criteria for inclusion of members, how we know we have one, what's not included, etc.
  e.g., does Employee include retired, suspended, laid off, contract, visitor, temp
- Not concerned (yet) with how the members are represented
  – identifiers - surrogate lexical encoding. e.g. Days of the Week,
  7 members, one represented by – Tuesday, Tues, Tue, Tu, Mardi, Martes, …

N

Objects – 2

CvLvP
52

- Grouping individual instances into populations
  does not occur naturally in the real world.
  The designer *chooses* to include members based on some common characteristics for some purpose(s)
- By convention we name object type populations with a singular noun, makes it easier to build sentences
- NOTE: in general, individual object populations could be overlapping, i.e., an individual could be a member of multiple populations
  e.g., an Employee could also be a Customer or a Shareholder
  This is handled using Subtype/Supertype constructs

❖ USER NARRATIVES in the Domain of Discourse
=> NOUNS => OBJECT instances
=> OBJECT TYPES => NAMES

Subtypes/Supertypes

53 CvLvP

See Dataversity Webinar

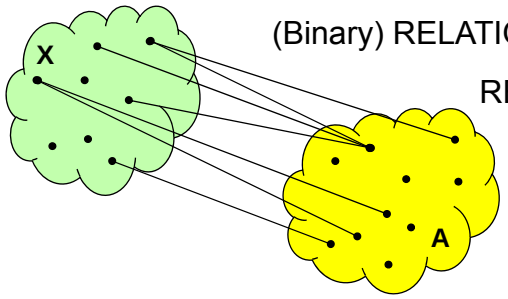
- Data modeling schemes assume Object Populations are strictly disjoint
  i.e., an individual member of an Object Population cannot be a member of any other Object Population
- We know that is not always true
  e.g., Person can be Employee, Customer, and Shareholder
  If these are modeled as separate populations, redundancy results which can lead to inconsistent data.
  Maintaining consistent data becomes a user responsibility
- S/Stype construct is used to formally represent overlapping populations. It only depends upon the nature of defined Object populations.
- Supertype is a *generalization* of its Subtypes.
  Several *constraints* can be defined on S/Stypes.

❖ OBJECT TYPES => SUBTYPE/SUPERTYPES
=> S/Stype CONSTRAINTS

Relationships

54 CvLvP

- "Connection" between or among members of one or more object populations.
  Arity = number of Roles played by Objects participating in the Relationship, e.g., Unary, Binary, Ternary, etc.



(Binary) RELATIONSHIP TYPE:

X	A

RELATIONSHIP Instances:
All valid X-A pairs
(in the R/W)

What are the characteristics of the relationship 'X-A' ?

Relationship Names and Object Roles

55 CvLvP

- Naming all Relationships and Roles not necessary, can use the Object Names as a default: Relationship "X-Y"
  - But user narratives *will* include verb phrases to reference relationships making it easier to form sentences when talking about the user domain.
- Except if there are *multiple* relationships on X-Y
  e.g., "Employee *works in* Dept" and "Employee *heads* Dept" in which case the Employee plays the role of Boss in the "*heads*" relationship.
  NOTE: role order matters, e.g.,binary has two readings
- Except if the same Object type plays *multiple* roles
  e.g., "Person *is parent of* Person" then must name the relationship or distinguish the roles as Parent and Child
- Object Role names are nouns within context of a relationship

❖ USER NARRATIVES in the Domain of Discourse
=> OBJECT TYPES =>
=> VERBS => RELATIONSHIP TYPES
=> NAMES => ROLE NAMES

Constraints on a Relationship

56 CvLvP

- The defaults are the least constrained
  - Multiple
    - every object instance may participate more than once
      e.g., many-to-many (M:N) for a binary relationship
  - Optional
    - every object instance need not participate in the relationship
- The Constraints would be: (the opposites)
  - Exclusive – *at most one*
  - Dependent (Mandatory, Required …) – *at least one*

Many different notations, sometimes confusing.
Combination called 'Cardinality' [min:max], a notational convenience

❖ RELATIONSHIP TYPES
=> EXCLUSIVITY Constraint
=> DEPENDENCY Constraint

What is an Attribute?

ORMVER
57 CvLvP

An ATTRIBUTE is … of what?

an OBJECT...

playing a ROLE

in a RELATIONSHIP

with some (other) OBJECT.

What comes first?

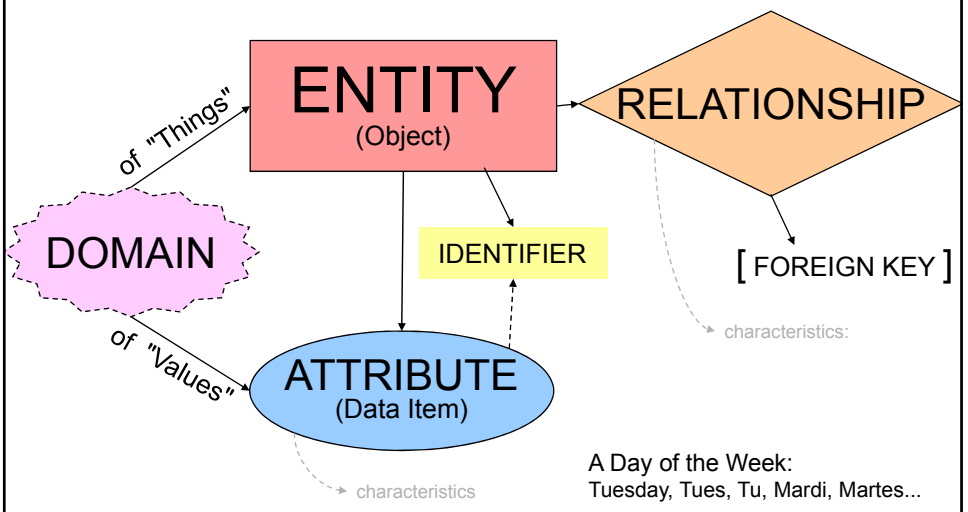
❖ RELATIONSHIPS => MULTIPLICITY => CLUSTERING =>
=> ENTITY records => ATTRIBUTES

N

Data Modeling Constructs

DMOD
58

What to look for:



What's the difference?

N