


# ERel Modeling to ORMModeling


DAMA Day, Madison, WI, 2011 November



GETITLE

## DAMA-Wisconsin DAMA Day


Madison, 2011 November



# Data Modeling:


from ER/Relational [EAR]  
Modeling  
to

# ORMModeling



©Gordon C. Everest  
Professor Emeritus of MIS and Database  
Carlson School of Management  
University of Minnesota

[geverest@umn.edu](mailto:geverest@umn.edu)
[www.tc.umn.edu/~geverest](http://www.tc.umn.edu/~geverest)



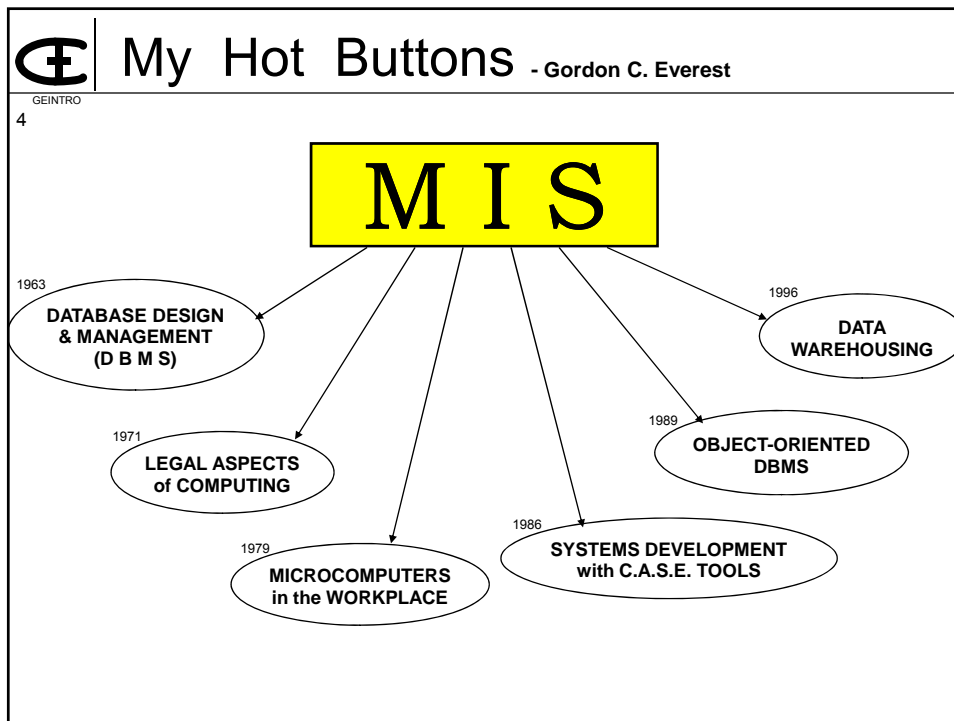
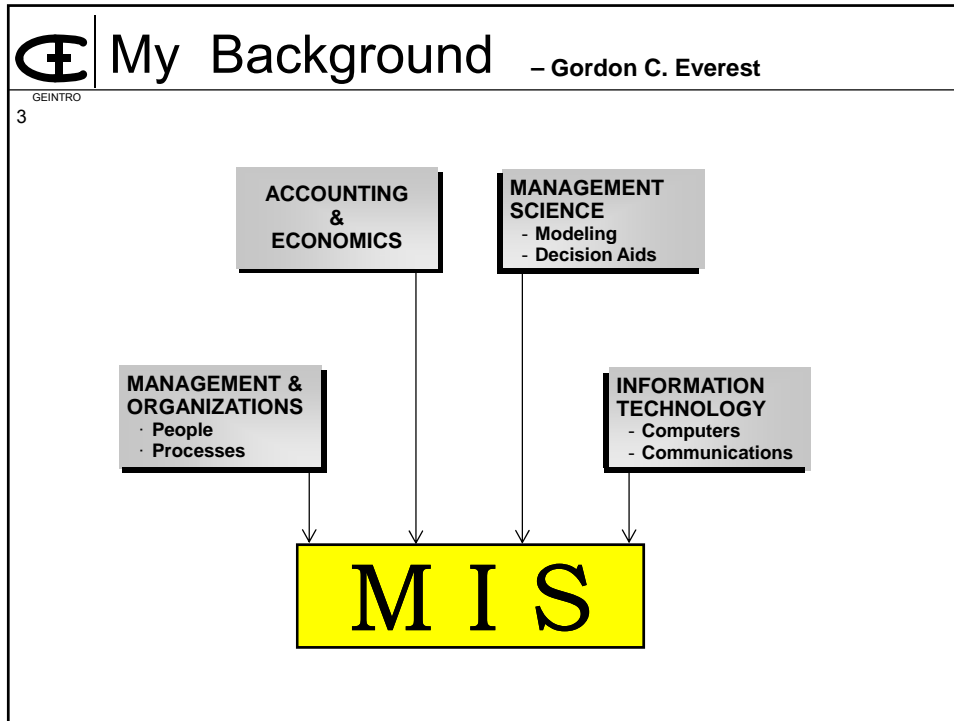
ORMVER

## Outline – ERel Modeling to ORMModeling

		<u>Slide#</u>
2	• Introduction: GE, my course, attendee profile	03
	• Data modeling - objective, purpose, process, constructs	10
	• Decoding a data model diagram (H1)	24
	• Semantics presumed by [ X   A   B ]	
	• Attributes and Relationships	31,35
	• Encoding changed semantics (H2): producing a data model diagram	39
	• “Table Think” – process & problems, attributes vs. relationships	49
	– Difficulties representing relationships, M:N, Ternaries++	
	– Functional dependency and normalization	
	• A “new” way of thinking – Object Role Modeling (ORM)	65
	– Transitioning from ERel to ORM, Process, Constructs; Defining Object populations	
	– Elementary Fact Sentences – objects and predicates	89
	• Constraints – Uniqueness (Exclusivity), Mandatory (Dependency), Reference Mode, Ternary relationships and objectified predicates, Value set, Frequency, Role population, Ring (on a reflexive relationship)	99 107 112
	• Deriving (normalized) tables	119
	• ORM data model presentation – abstractions	120
	• ORM Reprise – Why? Why not? Benefits	122
	• ORM Modeling tools – Visio, NORMA	127
	• Where ORM fits in the Data Modeling world	131
	• Departing words	135

# ERel Modeling to ORModeling


DAMA Day, Madison, WI, 2011 November



# ERel Modeling to ORModeling


DAMA Day, Madison, WI, 2011 November



	<h2>Gordon Everest – Quartet Singing</h2>
5	
	

	<h2>The Advanced Database Design Course</h2>
6	<p>DESCRIPTION:</p> <ul style="list-style-type: none"><li>• Teaches a new way of thinking - ORM<ul style="list-style-type: none"><li>✓ Avoids “table think” and all of its consequent problems<ul style="list-style-type: none"><li>The need for normalization; diagrams difficult to understand</li></ul></li><li>✓ Learning ORM is the treatment for “Tableitis”</li></ul></li><li>• Mix of students &amp; working professionals</li><li>• Professionals often have more trouble unlearning what they know, have learned, and have practiced</li><li>• Prior knowledge and experience<ul style="list-style-type: none"><li>– Used a DBMS to setup and query/manipulate tables</li></ul></li><li>• Now offered online, piggy back on a face-to-face class</li><li>• This workshop arises out of my observations of problems stemming from faulty thinking – “Table Think”</li></ul>

# ERel Modeling to ORModeling



DAMA Day, Madison, WI, 2011 November


	<h2>Student Feedback</h2>
<small>431INTRO</small> 7	<p>..... Students are the best advocates for the course – talk to them.</p> <ul style="list-style-type: none"><li>• What former students say about the course:<ul style="list-style-type: none"><li>– “Enjoyed your class tremendously, one of the best I had at the U!”</li><li>– “Great professor” “Brilliant instructor” “Loved the class”</li><li>– “Clear presentation” “challenging assignments” “good feedback”</li><li>– “Instructor provided passion about the subject material in such a way it made my approach to the material take on a level of seriousness I hadn’t expected”</li></ul></li><li>• Regarding their career:<ul style="list-style-type: none"><li>– A solid foundation for success as an IT/IS professional</li><li>– “I quickly became the in-house expert, among seasoned modelers”</li><li>– “You can’t believe how bad it is out there in practice”</li><li>– “I can’t tell you how often I come across poorly designed databases”</li></ul></li><li>• Very high ratings from the students<ul style="list-style-type: none"><li>– Aggregate of 5.8/6 on SRT (up to 6.9/7 on the former SET)</li><li>– Many come back to thank me for the course</li><li>– Encourage their colleagues to take the course</li></ul></li></ul>

	<h2>Attendee Profile - Job</h2>	
<small>ORMVER</small> 8	<h3>Practicing Data Modeler / Data Architect</h3>	
	<ol style="list-style-type: none"><li>1. Main part of my job</li><li>2. Doing it frequently but not my main job</li><li>3. Incidental to my current job</li><li>4. Only an observer of others doing it</li></ol>	

# ERel Modeling to ORModeling

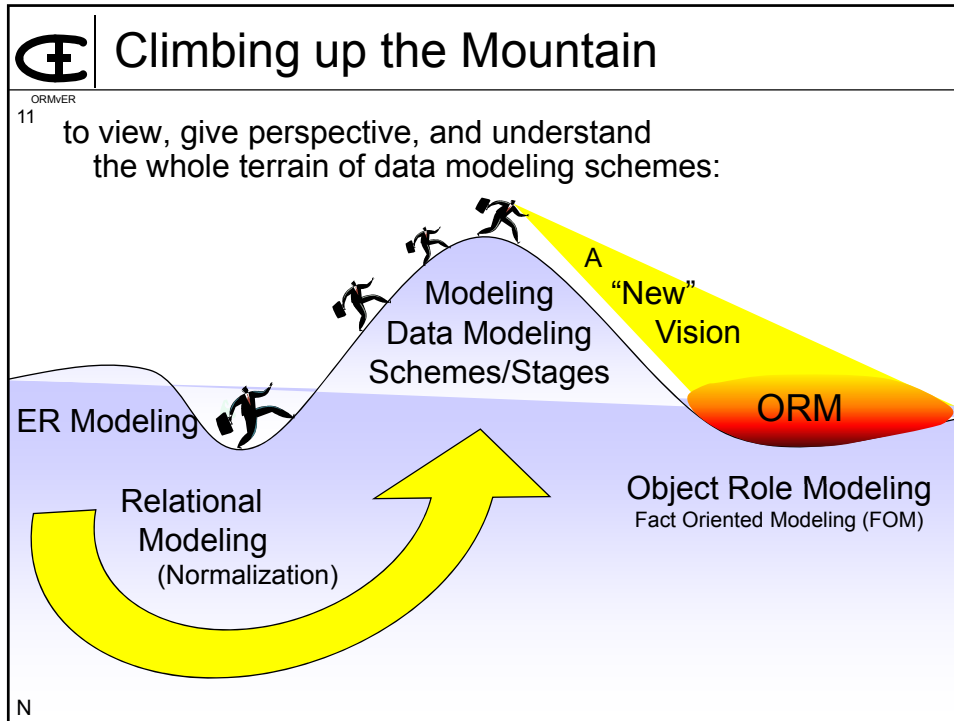
DAMA Day, Madison, WI, 2011 November

	<h2>Attendee Profile</h2>	
<small>ORM/ER</small> 9	<h3>Level of Knowledge / Experience</h3> <p><i>How good a data modeler do you consider yourself?</i></p> <p>Expert = can handle very complex design situations.</p> <ol style="list-style-type: none"><li>1. Expert</li><li>2. Very Good</li><li>3. Good</li><li>4. Fair</li><li>5. Poor</li></ol>	

	<h2>Problems and Solutions</h2>	
<small>ORM/ER</small> 10	<p><b>OBJECTIVES FOR THIS PRESENTATION:</b></p> <ul style="list-style-type: none"><li>• Show several <b>PROBLEMS</b> with ER modeling schemes, (actually, <i>any</i> “record-based” modeling scheme).</li><li>• Identify the <b>ROOT CAUSE</b> of the problems <i>To stop there would be irresponsible, so...</i></li><li>• Show you a better way – a <b>SOLUTION</b> using Object Role Modeling (ORM)<ul style="list-style-type: none"><li>– Avoid "Table Think"; Defer building Tables.</li><li>– Encourage you to adopt ORM thinking</li><li>– Explore using an ORM tool (VisioEA/Team edn., NORMA)</li><li>– Does NOT mean abandoning current practice</li></ul></li></ul>	

# ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November



**ORM/ER** | Data Modeling B



431INTRO  
12



- STARTS from some expression of the users world to be modeled... *in data*
  - Applications depend on a well-designed database
- TRADITIONAL APPROACH – think ER/Relational tables
- PROBLEM: some data items not in the right place
- SOLUTION: to find errors, apply the rules of normalization
- NORMALIZATION - the Achilles heel of data modeling
  - Even professional data modelers get it wrong
- REMEDY for violations => record decomposition

***So wouldn't it be nice to have:***



- Modeling scheme which avoids the need for normalization
  - => ORM (Object Role Modeling, or Fact Oriented Modeling)
- Modeling tool for ORM => NORMA (also Visio in VS)
  - To automatically generate tables... in fifth normal form!


# ERel Modeling to ORModeling

	<h2>Record-Based Data Modeling</h2>	
	<p><small>DMOD</small> 13</p> <ul style="list-style-type: none"><li>• Commonly called Entity Relationship (ER) Modeling</li><li>• Attributes clustered into Entity Records (or Tables)</li><li>• Focus on <u>E</u>ntities and <u>R</u>elationships (hence ER) suppressing attributes in ER Diagrams (hence no explicit representation of identifiers) leaving open the nature of the intra-record structure.</li><li>• Most general case <i>allows</i>:<ul style="list-style-type: none"><li>– “Nested” Multivalued attributes or repeating groups Hence not in first normal form (1NF) (should still satisfy other normal forms – 2NF, 3NF, ...)</li><li>– Direct representation of M:N relationships between entities</li><li>– Attributed relationships (i.e., with attributes)</li><li>– Ternary (and higher order) relationships</li><li>– Subtypes and Supertypes</li></ul></li><li>• Restricting <i>all</i> of the above gives the <u>Relational Model</u><ul style="list-style-type: none"><li>– Atomic (single-valued) attributes; binary relationships (FKey)</li></ul></li></ul> <p><i>=&gt; Relational Table Diagrams are often called ERDiagrams</i></p> <p><small>N</small></p>	

	<h2>Object Role Modeling (ORM)</h2>	
	<p><small>431INTRO</small> 14</p> <ul style="list-style-type: none"><li>• The ultimate end of record decomposition<ul style="list-style-type: none"><li>– Where there are “tables” with at most one “attribute,” i.e., for each relationship</li></ul></li><li>• The starting point for “conceptual” data modeling<ul style="list-style-type: none"><li>– Not constrained to put everything in tables</li><li>– No need to decide what is an entity and what an attribute</li></ul></li><li>• Does not replace ER / Relational modeling<ul style="list-style-type: none"><li>– Just a stage before... implementation in a Relational DBMS</li></ul></li><li>• Uses one construct: Elementary Fact Sentence<ul style="list-style-type: none"><li>– Made up of nouns (objects) and verbs (relationships)</li><li>– Enables verbalization of a data model diagram, which is much easier for business users to comprehend</li></ul></li><li>• Able to capture and diagrammatically represent much richer semantics, integrity constraints, business rules</li><li>• Easier to correctly model very complex situations</li></ul>	

# ERel Modeling to ORModeling

	<h2>Data Modeling</h2>	
<small>DMOD 15</small>	<p>What Data Modeling Scheme is dominant today? <i>What's Wrong with ER Modeling?</i> (at the root)</p> <p>BEFORE WE CAN ANSWER THAT:</p> <p><i>What</i> do we do in Modeling? the <u>objective</u>, the deliverables.</p> <p><i>Why</i> do we do Data Modeling? the <u>purpose</u>.</p> <p><i>How</i> do we do Data Modeling? the <u>process</u>; guidelines.</p> <p><i>Who</i> does Data Modeling?</p> <p><i>When</i> do we do Data Modeling?</p>	
<small>N</small>		

	<h2>Logical Database Design</h2> <h3>Objective, Principles, Benefits</h3>	<small>B</small>
<small>DMOD 16</small>	<p>OBJECTIVE of LOGICAL DATABASE DESIGN: The <i>WHAT</i></p> <p style="background-color: yellow; text-align: center;"><i>TO ACCURATELY AND COMPLETELY MODEL SELECTED PORTIONS OF THE REAL WORLD OF INTEREST TO A COMMUNITY OF USERS.</i></p> <ul style="list-style-type: none"><li>• USERS (COLLECTIVELY) WILL ALWAYS KNOW MORE ABOUT A DATA STRUCTURE THAN THE SYSTEM KNOWS, OR THAN COULD BE DEFINED TO THE SYSTEM.</li><li>• WHAT IS NOT FORMALLY DEFINED TO THE SYSTEM, THE SYSTEM CANNOT MANAGE . . . THE USERS MUST!</li><li>• THEREFORE, NEED TO <u>CAPTURE RICH SEMANTICS</u> WITH COMPREHENSIVE DATA MODELING and DEFINITION, INCLUDING INTEGRITY CONSTRAINTS AND OPERATIONS.</li></ul> <p><i>Let the 'system' do it!      Implications for a Tool!</i></p>	

# ERel Modeling to ORModeling

**☒ Purpose of Modeling – the *Why***

DMOD 17

To Facilitate *Human* Communication, Understanding, Validation

- Capture semantics – all relevant, important details
- Document – record and remember
- Understand – learn, raise questions, record answers, refine
- Communicate – shared with all interested parties
  - Users, stakeholders, management, developers
- Validate – a complete and accurate representation
  - Internal validation – consistent with the modeling rules
  - External validation – *Who can do this?*

SECONDARY:

- Blueprint to Build (a Database)

N

**☒ Modeling Process – the *HOW***

DMOD 18

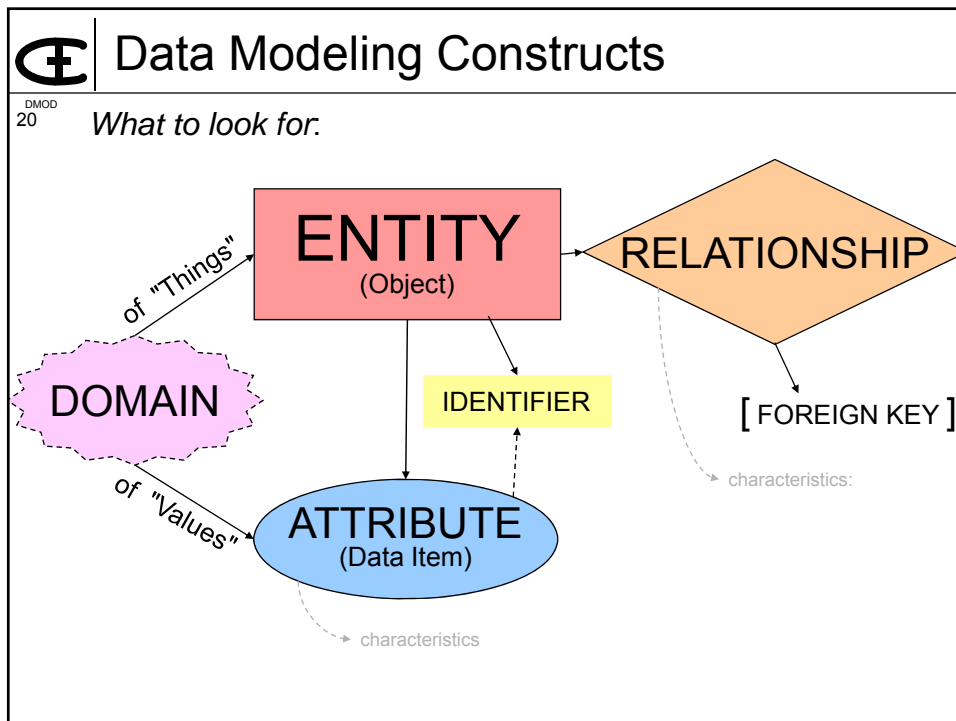
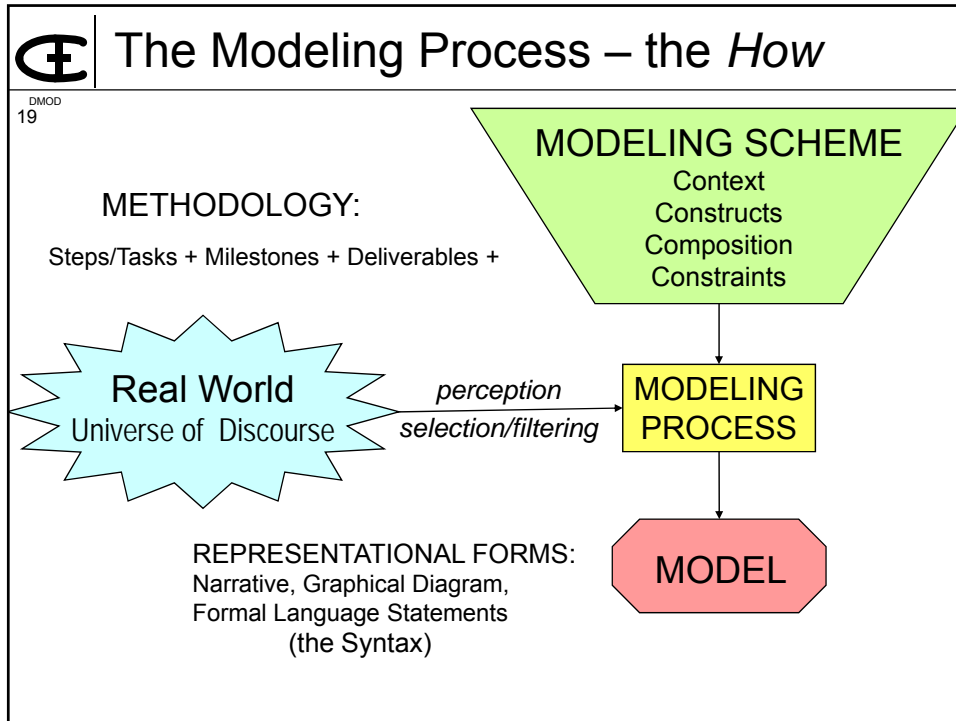
MODEL = Abstract (Re).present.(ation)

Knowledge in the world (infinitely complex) → Knowledge in the head (mental models) → Knowledge externalized, formalized, shared.

Reality → *present* → MODELING PROCESS → MODEL → *Re-present*

*What drives or guides the process?*

# ERel Modeling to ORModeling



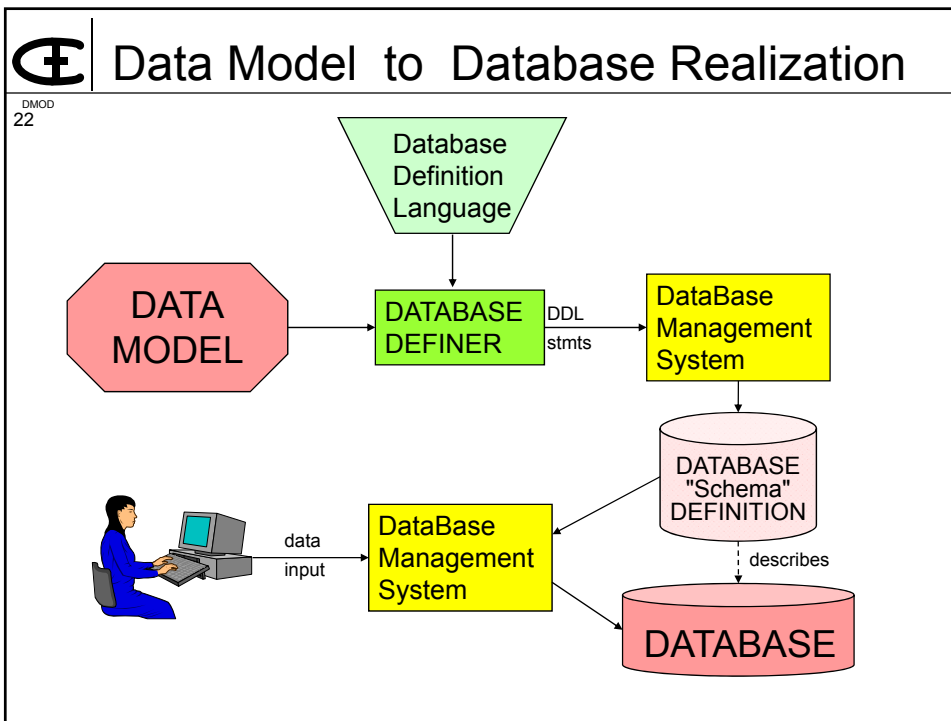
# ERel Modeling to ORModeling

## ☒ Data Modeling Constructs

DMOD 21


*Relative emphasis differentiates Data Modeling Schemes:*

- ER modeling focuses on Entities and Relationships, de-emphasizing, even hiding Attributes.
- Relational (restricted ER) focuses on Entities and Attributes (single valued), relegating Relationships to Foreign Keys.
- Object Role Modeling (ORM) focuses on Objects/Domains and (Roles in) Relationships  
NOTE: there are NO Attributes *per se*



# ERel Modeling to ORModeling


DAMA Day, Madison, WI, 2011 November

 Your First Cut Data ModelsT

ORMWER  
23

- What you present to your (business) users
  - after talking to the user domain (subject matter) experts
  - before making any refinements for implementation.
    1. Attributes (whether or not explicitly shown)
    2. Entity tables (ER records or Relational)
    3. Identifiers
    4. Foreign keys
    5. Normalized
    6. Relationship arcs
    7. Many-to-Many (M:N) relationships
    8. Ternary<sup>+</sup> relationships
    9. Subtypes/Supertypes
    10. Integrity constraints/Business rules

*What is missing from this list?*  
*What does the user really need to see?*

 Record-based DesignG

ORMWER  
24

*WHAT SEMANTICS ARE PRESUMED  
BY THE FOLLOWING RECORD STRUCTURE?*

<u>X</u>	A	B
----------	---	---

*ASSUME everything known  
about X, A, and B is shown.*

H1


1. What does it say about **X** ?
2. What does it say about **A** ?
3. What does it say about the relationship **X-A** ?
4. What does it say about the relationship **A-B** ?


Think in terms of:  
Existence, Attributes, Roles, Relationships,  
and relationship characteristics of Multiplicity, and Dependency.

State what is, NOT what is not, or what could be, or by construction, or processing.

N

# ERel Modeling to ORModeling


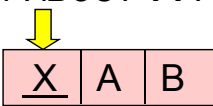
	<h2>Record-based Design</h2>	B		
	<p>ORM/ER 25</p> <p><i>WHAT SEMANTICS ARE PRESUMED BY THE FOLLOWING RECORD STRUCTURE?</i></p> <table border="1" data-bbox="626 485 837 541"><tr><td><u>X</u></td><td>A</td><td>B</td></tr></table> <p><i>Do we know if it is Normalized? - to Third Normal Form (3NF)? How do we know?</i></p> <p><i>Can a DBMS or data modeling tool help?</i></p> <p><i>Why can't we know?</i></p> <p>We must ASSUME it is Normalized, Otherwise ambiguity in what [ <u>X</u>   A   B ] means.</p>	<u>X</u>	A	B
<u>X</u>	A	B		

	<h2>Why Abstract Diagrams?</h2>									
	<p>ORM/ER 26</p> <table border="1" data-bbox="698 1253 909 1310"><tr><td><u>X</u></td><td>A</td><td>B</td></tr></table> <p>Rather than:</p> <p>EMPLOYEE</p> <table border="1" data-bbox="482 1442 1105 1493"><tr><td><u>Emp#</u></td><td>Name</td><td>...</td><td>SkillCode</td><td>Dept ID</td><td>...</td></tr></table> <ul style="list-style-type: none"><li>• Examples can be dangerous, but...</li><li>• Examples can help confirm one's own understanding.</li><li>• If you can understand these concepts in the abstract you will be better able to do data modeling for real.</li></ul>	<u>X</u>	A	B	<u>Emp#</u>	Name	...	SkillCode	Dept ID	...
<u>X</u>	A	B								
<u>Emp#</u>	Name	...	SkillCode	Dept ID	...					

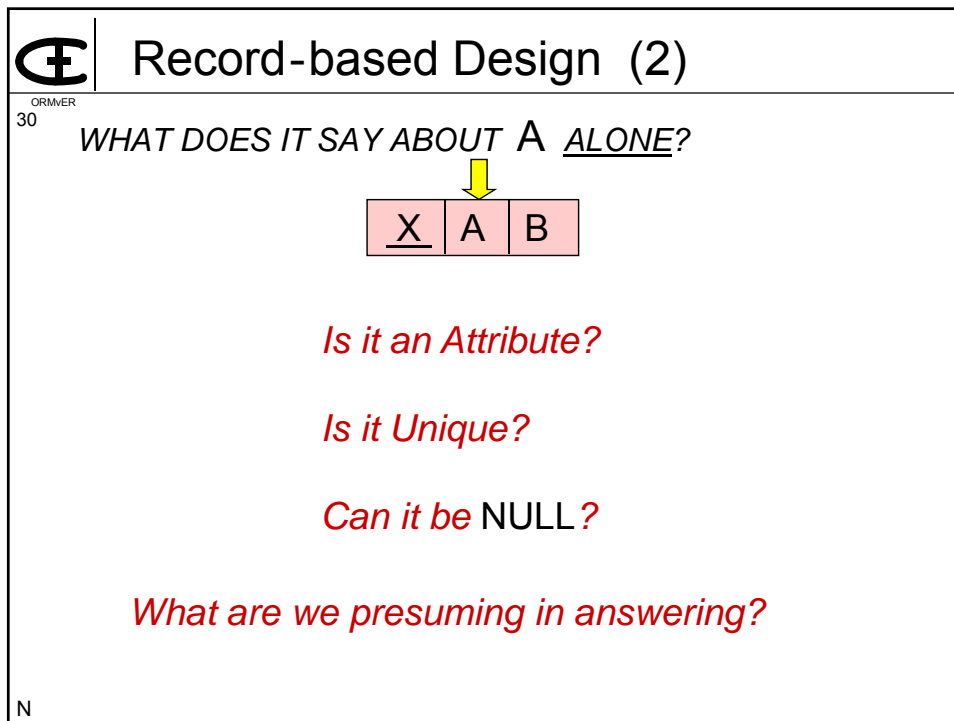
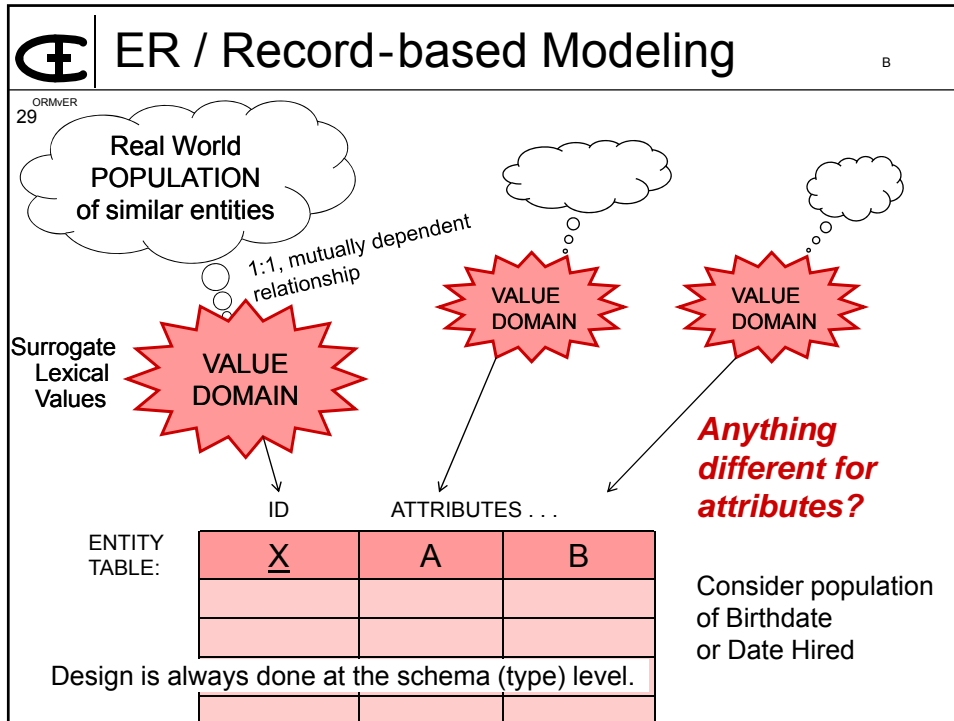
# ERel Modeling to ORM Modeling

DAMA Day, Madison, WI, 2011 November

	Countdown Timer
<small>GESTART</small> 27	Click to START Minutes remaining:  <b>10</b>

	Record-based Design (1)
<small>ORM/ER</small> 28	WHAT DOES IT SAY ABOUT X ?  
N	

# ERel Modeling to ORMModeling



# ERel Modeling to ORM Modeling

**Ⓔ** What is an Attribute? Ω

ORM/ER  
31

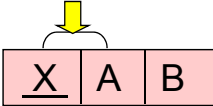
**An ATTRIBUTE  
is an OBJECT  
playing a ROLE  
in a RELATIONSHIP  
with some (other) OBJECT.**

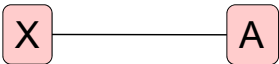
*What comes first?*

**Ⓔ** Record-based Design (3)

ORM/ER  
32

WHAT DOES IT SAY ABOUT THE RELATIONSHIP X-A ?




What if I just drew: 


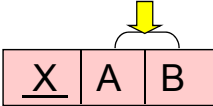
*What could you say?*

*How the table view frames our thinking !*

N

# ERel Modeling to ORM Modeling

	<h2>Record-based Design</h2>	B			
<small>ORM/ER</small> 33	<h3>REPRESENTING THE RELATIONSHIP X-A ?</h3>				
<table border="1" style="margin: auto;"><tr><td style="text-align: center;"><u>X</u></td><td style="text-align: center;">A</td><td style="text-align: center;">B</td></tr></table>			<u>X</u>	A	B
<u>X</u>	A	B			
<p><i>What if A ...</i></p> <ul style="list-style-type: none"><li>• <i>can be an orphan?</i> <span style="float: right;">See H2.5</span></li><li>• <i>has additional attributes?</i> <span style="float: right;">See H2.6</span></li><li>• <i>can be multi-valued?</i> <span style="float: right;">See H2.7</span></li></ul>					
<p>To handle these situations this diagram is wrong or insufficient (as we will see in H2).</p>					

	<h2>Record-based Design (4)</h2>	B			
<small>ORM/ER</small> 34	<h3>WHAT DOES IT SAY ABOUT THE RELATIONSHIP A-B ?</h3>				
<table border="1" style="margin: auto;"><tr><td style="text-align: center;"><u>X</u></td><td style="text-align: center;">A</td><td style="text-align: center;">B</td></tr></table> 			<u>X</u>	A	B
<u>X</u>	A	B			
<ul style="list-style-type: none"><li>• <i>Is there a relationship between A and B in <u>X</u> ?</i></li><li>• <i>Could there be a relationship between A and B ?</i></li><li>• <i>If so, what would it look like?</i></li><li>• <i>Would the table of <u>X</u> still be normalized?</i> <span style="float: right;"><small>H</small></span></li><li>• <i>Would there be any Foreign keys?</i></li></ul>					
N					



# ERel Modeling to ORModeling

E

## Relationship Characteristics

B1 Σ

DMOD 37 Each characteristic defined in *both directions*, on each entity:

- **NAME**
  - Singular VERB phrase
- **MULTIPLICITY<sup>D</sup> / EXCLUSIVITY**
  - RELATE TO MULTIPLE or/ AT MOST ONE
  - 1:1    1:MANY    MANY:MANY (M:N)
- **DEPENDENCY**, MANDATORY, REQUIRED, EXHAUSTIVE / OPTIONAL, ORPHAN<sup>D</sup>
  - MUST HAVE - - - AT LEAST ONE, ELSE "ORPHANS" ALLOWED
- **Over time**
  - DELAYED Dependency
  - FIXED to an Entity instance
- **CRITERIA**
  - BASIS IN MATCHING VALUES - "FOREIGN IDENTIFIERS"
  - ONE / MULTIPLE DATA ITEMS (composite)
  - Identifier or not => Multiplicity
- **DEGREE**
  - BINARY, TWO - - - > TERNARY, OR MORE ENTITIES.

ALL REFLECTED IN A GRAPHIC DIAGRAM !

E

## H1 - Typical Average Results

B

ORMVER 38

40% miss: There exists a population of things called A.

90% miss: There exists a relationship between X and A.

100% miss: X is a descriptor of A (even when asked!)

35% miss/wrong: dependency/optionality characteristic.

60% miss/wrong: multiplicity/exclusivity characteristic.

50% say or imply there is a relationship between A and B.

What might the results look like if presented with:

1. Object Domains
2. Relationships
3. Constraints

NO Tables, Identifiers or For. Keys

# ERel Modeling to ORModeling

E

## H2 – Data Model Diagrams

ORMWER  
39

# H2

Now that we know what [X | A | B ] means,

Encode Semantics to produce a Relational Table Diagram.

Variations on X A B commonly encountered.

E

## Communication

DMDPRE  
40  
ORMWER

... sending and receiving.

# ERel Modeling to ORModeling

E

## Diagram Notation

B

---

NORM  
41

**NAME**

(Name)

- Icon to represent a domain or population of instances of one type of thing or value set.
- The NAME reflects the type of thing/object.

**ENTITY NAME**

<u>Identifier</u>	Attr1	Attr2	...
-------------------	-------	-------	-----

Values uniquely identify members of the entity population.

X	Y	A	B
---	---	---	---

= Composite Key

- One entity table (horizontal) ...
- Column values represent members of the attribute population
- Adorn attribute names to indicate special characteristics e.g. Required •, Unique, [Multivalued]

**Relationships between entity tables:**

X	A	...
---	---	-----

—•

<u>Y</u>	...	X
----------	-----	---

MANY, else one

X	...
---	-----

←

Y	...	X
---	-----	---

Foreign Key ...as a pointer

... or vertical:

LINE ITEM	
OrderNo (PK,FK1)	
<u>ItemID#</u> (PK,FK2)	
ItemDesc [o]	
Quantity •	

DEPENDENT (Mandatory, Required), else Optional/Orphan.

Two-valued logic requires a default.

E

## H2 – All Attributes Optional

T

---

ORM/ER  
42

1. There exists an X (one or more instances) with NO values for A and B , ..., i.e., for *all* attributes.

HINT: Think “What is the correct design” then pick it.

1. 

<u>X</u>
----------
2. 

<u>X</u>		
----------	--	--
3. 

<u>X</u>	A	B
----------	---	---
4. 

<u>X</u>	—	A	B
----------	---	---	---
5. 

<u>X</u>	A <sup>o</sup>	B <sup>o</sup>
----------	----------------	----------------

N

<sup>o</sup> optional

© Gordon C. Everest, All rights reserved.

# ERel Modeling to ORModeling

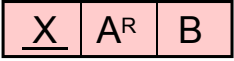
E

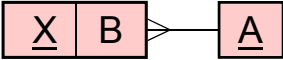
## H2 – An Orphan Attribute

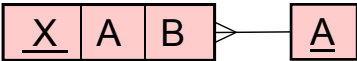
T

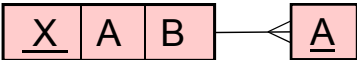
ORM/ER  
43

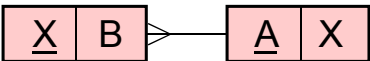
5. There exists an A (one or more) which is NOT associated with any X, that is, some A s are orphans. There is nothing else of interest about A.


1. 

2. 

3. 

4. 

5. 



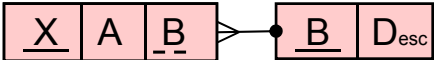
E

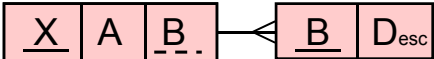
## H2 – Attributes of an Attribute

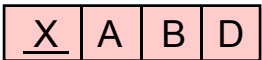
T

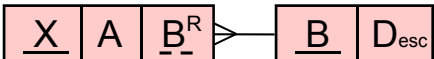
ORM/ER  
44

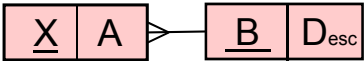
6. Suppose we now have an additional data attribute D which is of interest to us about B. Many got this, but missed 5. *Is B still dependent on X?*

1. 

2. 

3. 

4. 

5. 

Special case: a 'Decode Table' for B

# ERel Modeling to ORModeling

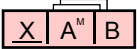
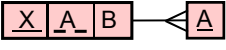
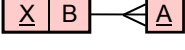
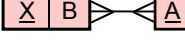



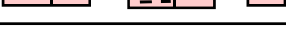
E

## H2 – Multi-valued Attribute

T

ORM/ER 45

7. X is NOT exclusive on A , that is, there may be multiple values of A for a given X . Many missed this. First, understand the semantics

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 

N

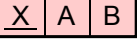
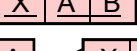
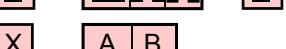
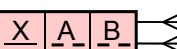
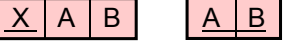



E

## H2 – Attributes with M:N Relationship

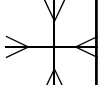
T

ORM/ER 46

8. There exists a M:N relationship between A and B , in addition to the relationship each of them has with X .

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 

N



## ERel Modeling to ORM Modeling

DAMA Day, Madison, WI, 2011 November

Page 24

E

### H2 – Attributes with 1:M Relationship

T

ORM/ER  
47

9. There is a many-to-one relationship between A and B since we observe that for every different or unique value of A, the values of B are all the same value.

1. 

<u>X</u>	A	B
----------	---	---
2. 

<u>X</u>	<u>A</u>	B
----------	----------	---

⇒

<u>A</u>
----------
3. 

<u>X</u>	<u>B</u>
----------	----------

⇒

<u>B</u>	A
----------	---
4. 

<u>X</u>	<u>A</u>
----------	----------

⇒

<u>A</u>	B
----------	---
5. 

<u>X</u>	<u>A</u>
----------	----------

⇐

<u>A</u>	B
----------	---

N

E

### H2 - Problems for the Students

ORM/ER  
48

- Much confusion with the Foreign Key
  - Inconsistent with the relationship arc
  - Must have an ID to point to
  - Can only represent at most a 1:Many relationship
- Putting an Attribute or Foreign Key in a table means the entity can have at most one of them.
- *Every Relationship* must be represented *somewhere* in the model with a pair of values (for a binary relationship)
- M:N Relationship means there must be a composite key *somewhere* in the model, and vv.
- Just because two attributes are together in a table does not mean there is a relationship.

**Ⓔ** The problem

ORMWER  
49

## “TABLE THINK”


... is the problem

*Is the Relational Data Model for people or for the “machine/system”?*

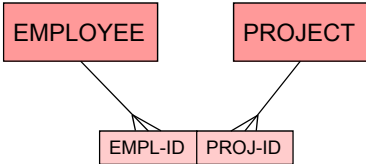
*Why do we draw Data Model Diagrams?*

See description of “Tableitis” – a serious malady in our discipline.

**Ⓔ** Representing a M:N Relationship

DMOD 50 Another Pattern:  WATSON5, Ch.5, p.115.

- If you cannot store multiple Projects (or Project IDs) in an Employee record, or multiple Employees (or Employee IDs) in a Project record (as is the case in a Relational Database), then ... you *must* introduce an “Intersection Entity” between them to represent the Many-to-Many Relationship.

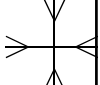


- The Intersection Entity also provides the place to store additional attributes of the relationship e.g., Hours Worked, Rate of Pay, ...

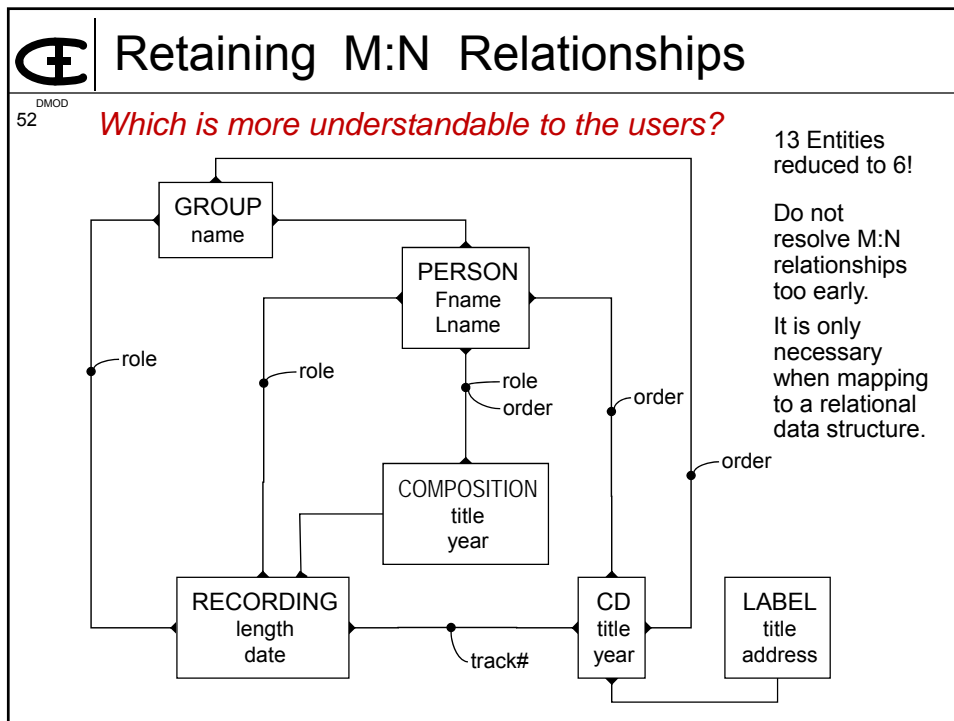
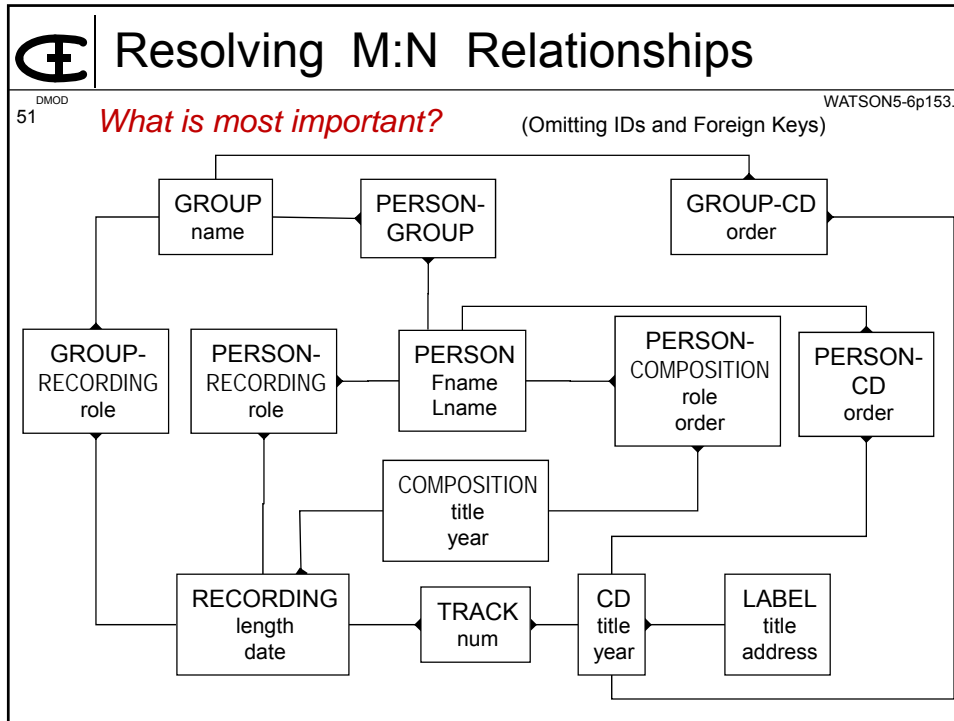
*What is the IDentifier?      Where store attributes if 1:M?*

*What is the problem with this representation?*

*What is the underlying cause of the problem?*



# ERel Modeling to ORModeling



# ERel Modeling to ORModeling

E

## Representing a Ternary Relationship

DMOD  
53

While we can develop a consistent notation for binary relationships, ternary relationships are a problem. Sample EER (Teorey) notation:

- *If one of the entities is single valued, is it really ternary? Or "attributed" binary?*
- *What lends uniqueness to each instance of the relationship?*
- *How to verbalize the relationship? Which order?*
- *How to represent Multiplicity / Exclusivity ?*
- *How to represent Dependency? Must have all 3?*

E

## Representing Relationships

ORM/ER  
54

Multiple *different* ways:

(1) Intra-record =>  
Entity with Attributes

(3) Between/Among Attributes?  
*Spurious Associations?*

(2) Inter-record =>  
between Entities,  
Entity with another Entity

*What if:*

- *A has some other attributes?*
- *some A's can be orphans ?*
- *Is A still dependent on X ?*

*Where are the Foreign Keys?*

*How are they different?*

# ERel Modeling to ORModeling

E

## Representing a Relationship

ORM/ER  
55

- The schema design level:
- The instance (data) level:
- What is the Identifier?  
Don't know until...?

X	A

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

<u>X</u>	A
----------	---

X	<u>A</u>
---	----------

<u>X</u>	<u>A</u>
----------	----------

<u>X</u>	A
----------	---

Must know the multiplicity/exclusivity characteristics of X-A *before* you can put X and A in a Table Diagram.  
That requires the designation of an identifier;  
a foreign key requires the prior designation of an identifier.

E

## "TABLE THINK"

ORM/ER  
56

The process: focus on entity tables

- Find Entities
  - Top down from a macro view to find major entities, or
  - Bottom up from a list of data items (say from a DFD)
- Design a Table for each entity
  - clustering or adding attributes with each entity
- Columns as attributes of the entity – Name, Type
- Designate/define the Identifier
- Represent Entity Relationships with Foreign Keys
- [Normalize – not required or enforced by the system.]

To jump in and think of putting data into tables means we must have already decided on:

- Entities, Attributes, Entity Identifiers, Foreign Keys (relationships), and relationship characteristics

© Gordon C. Everest, All rights reserved.

# ERel Modeling to ORModeling

E

Attributes in a Table

ORMWER  
57

- When putting Attributes in a Table, we generally don't ask:
  - What is the object population it represents?
  - Could there be members of that population that would not be related to any member of this entity population but for which we would still be interested in recording?
  - Any other entity have/share the same attribute?
  - Anything else of interest about this entity?
  - Is it really an entity in its own right?  
if so, does it need its own table?

*We make assumptions implicitly about these questions when we go directly to designing tables.*

E

Attribute Characteristics

ORMWER  
58

An attribute of what (entity or object)?

***Nothing is an Attribute by itself.***

- All about the relationship with some entity X A
  - Optional or Required, conditional existence, derived
  - Single valued (or multi-valued which must be resolved)
  - Foreign key (relationship of this entity with another entity)
- All about identifying and describing instances A
  - Lexical surrogate for members of the object population
  - Name(s), description
  - Data type, format
  - Size, precision
  - Value set (domain of values) - range/s, list, encoded

# ERel Modeling to ORModeling

E

## Attribute Migration

ORM/ER  
59

Attribute... or Entity?

Given:

EMPLOYEE				
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px 10px;"><u>Emp#</u></td> <td style="padding: 2px 10px;">Name</td> <td style="padding: 2px 10px;">...</td> <td style="padding: 2px 10px;">SkillCode</td> </tr> </table>	<u>Emp#</u>	Name	...	SkillCode
<u>Emp#</u>	Name	...	SkillCode	

*What is SkillCode?*

Now add a Description, etc. for SkillCode:

*Now what is SkillCode?*

*What is the Relationship?*

*What is SkillCode in Employee table?*

Suppose an Employee has multiple Skills:

*Now what is the Entity?*

(In SQL, every table must have a name)

*What are the Relationships?*

*Is SkillCode still an attribute of Employee?*

E

## Functional Dependency in Relationships

Σ

NORM  
60

Basis for Database/Table Normalization.

$$A \leftarrow f(X)$$

A is dependent on X, and the Relationship is exclusive on A, multiple on X.

**Clustered** into a record/table for entity of X:

<u>X</u>	A	...
----------	---	-----

**RULE:** Store A with its Determinant as the Key.

*Encompasses all rules of Normalization!*

There can only be *one* A for each X.

There can be multiple Xs for a given A.

# ERel Modeling to ORModeling

E

## Normalization – Testing your Understanding

NORM  
61 ORMER

Assuming that A is single valued with respect to X (i.e. 1NF).  
(enforced by construction)

GIVEN: Could you have a violation of: (if not, why not?)

X	A
---	---

2NF?

3NF?

4NF or 5NF?

X	A	B
---	---	---

X	A	B
---	---	---

Clustering attributes into tables  
the modeler may make a mistake.  
Normalization is the test,  
Record decomposition is the remedy.

If you don't cluster attributes into tables,  
you cannot violate the rules of normalization.  
Normalization becomes unnecessary and irrelevant.


E


## “TABLE THINK” – Problems-1

ORMER  
62

- Anchors you on the entity of the identifier, which frames your thinking in one direction.
- Inter-entity relationships represented redundantly with a foreign key; in the conceptual model users see it twice.
- Difficult to represent the entire population of an object, whether entity or attribute domain.
- Enforcing single-valued (atomic) attributes makes it impossible to directly represent M:N relationships. It requires an intersection “entity.”
- When speaking of attribute characteristics, most of the time it is characteristics of the relationship with the entity. (required, optional, single-valued)
- Putting an attribute in an entity table presumes a particular relationship with the entity without being explicitly defined, consequently we can do it wrong, requiring normalization.

# ERel Modeling to ORM Modeling

	At the Root,
<small>ORM/ER 63</small>	<h1 style="color: red;">What's wrong with ERel Modeling?</h1>
N	H

	Problems with ERel Modeling - Summary
<small>ORM/ER 64</small>	<ul style="list-style-type: none"><li>• Cannot capture the "conceptual" view directly, Must <i>mentally</i> map to the "logical" (record-based) view by clustering Attributes into Entity records/tables.<ul style="list-style-type: none"><li>– Modeler must <i>a priori</i> choose whether Entity or Attribute</li><li>– Too much clustering; attributes in the wrong place</li><li>– Ignores (or presumes normalized) intra-record structure (that is, relationships between/among Attributes)<ul style="list-style-type: none"><li>- creates (implies) spurious inter-attribute relationships</li></ul></li></ul></li><li>• Human modeler is responsible for normalization     remedy is <i>always</i> record decomposition</li><li>• Must choose unique names<ul style="list-style-type: none"><li>– for attributes in a record; for spurious new "entities"</li><li>– column names = domain + role; lose object domains</li></ul></li><li>• Modeling / Processing dilemma:<ul style="list-style-type: none"><li>– Complete representation of an entity object - more clustering</li><li>– Full normalization (1NF) – decomposition, more fragmentation</li></ul></li><li>• Indirect representation of M:N relationships with intersection "entity"</li><li>• Difficulty representing Ternary relationships</li><li>• Stability of the query language (SQL)</li></ul>


All a consequence of clustering !


All are solved in ORM !

## ERel Modeling to ORModeling

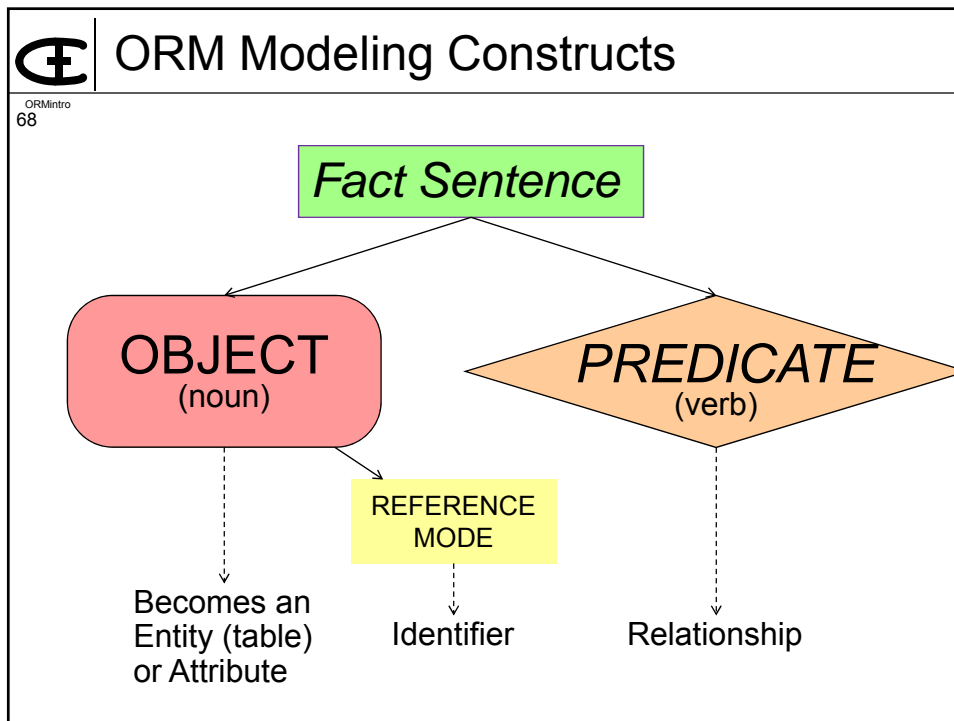
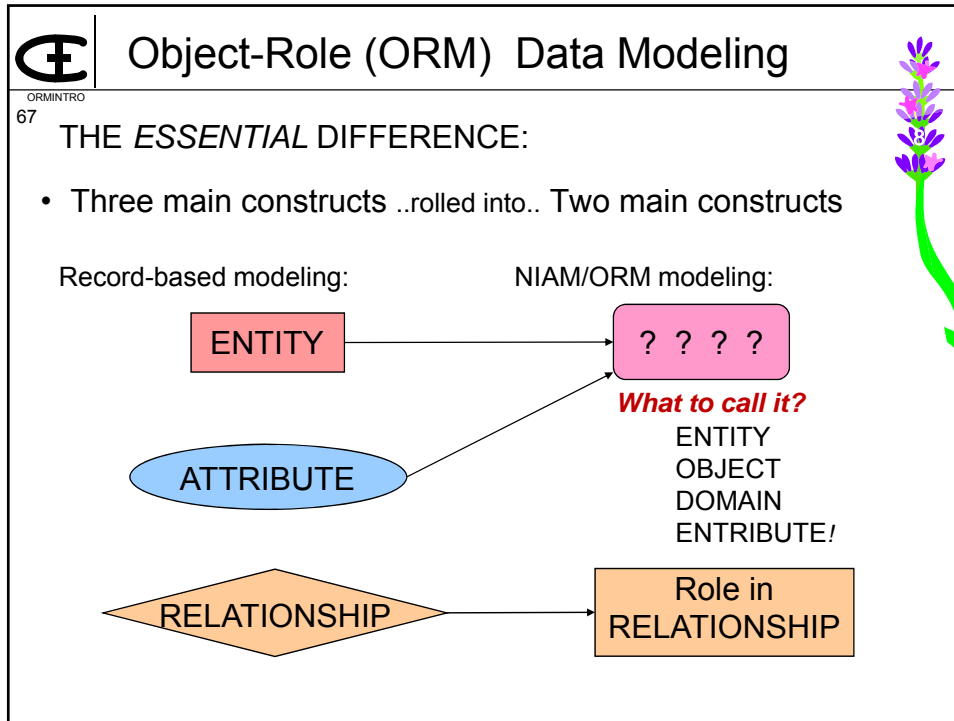
DAMA Day, Madison, WI, 2011 November

Page 33

	<h3>A "New" Way of Thinking</h3>
<small>ORM/ER 65</small>	<h1>Object Role Modeling</h1> <ul style="list-style-type: none"><li>• Formerly called Binary Modeling or NIAM</li><li>• Also called Fact Oriented Modeling (FOM)</li></ul> <p>Transitioning your thinking from ERel Modeling</p>

	<h3>Overcoming Limitations of ER Modeling?</h3>
<small>ORM/ER 66</small>	<h1>I will show you still a more excellent way</h1> <p>– PAUL, I Cor 12.31</p> <p>“IF I SPEAK ... of entity tables or records BUT HAVE NOT ... ORM I AM A NOISY GONG OR A CLANGING SYMBOL... (getting tangled up in anomalies, redundancies, missing semantics, &amp; normalization). IF I HAVE ALL KNOWLEDGE ... of ER modeling BUT HAVE NOT ... ORM I AM NOTHING.</p> <p>N</p>

# ERel Modeling to ORM Modeling



## ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November

Page 35



### Object Role Modeling

ORM/ER  
69

The PROCESS:

1. Model object populations first.
    - Name with singular nouns reflecting R/W object
    - Objects shown only once in an ORM diagram
  2. Represent *all relationships explicitly defined and only one way.*
    - Name roles objects play in relationships
- ====> thus can read out elementary fact sentences
- Defer definition of identifiers, data (storage) types, “attributes,” foreign keys, constraints.



### “TABLE THINK” – My Epiphany

ORM/ER  
70

- When we design a relational database we are actually modeling all and only the relationships among things in the users’ world
  - Relationships both within and between entity tables
  - Clustering (“throwing in”) attributes into tables without explicitly defining the intra-entity relationships.
  - Designers make mistakes – thinking of tables first
- We can only speak of attributes in the context of a relationship: \_\_\_\_\_ is an attribute of \_\_\_\_\_?
- We are not modeling *all object domains* first.

# ERel Modeling to ORM Modeling

E

Defining an Object Population

ORMWER  
71

- What is the population of A?
- How can you get the population of A?

1. 

<u>X</u>	A
----------	---

Any orphan A's?
2. 

<u>X</u>	A
----------	---

<u>Y</u>	A
----------	---

Is A still dependent on X?
3. 

<u>X</u>	A
----------	---

<u>Y</u>	A
----------	---

<u>A</u>	P	Q
----------	---	---

Any FKKeys?
4. 

<u>X</u>	A	B	A <sub>2</sub>
----------	---	---	----------------

<u>Y</u>	A
----------	---

<u>A</u>	P	Q
----------	---	---

How many times do you define a population of A's? Redundant?

E

A Better Data Model Diagram

ORMWER  
72

```

graph TD
    X((X)) --- B((B))
    X((X)) --- A((A))
    A((A)) --- Y((Y))
    A((A)) --- P((P))
    A((A)) --- Q((Q))
    Y((Y)) --- A((A))
    X((X)) <--> A((A))
            
```

1. Objects (domains) shown only once each  
Not distinguish Entities and Attributes
2. All Relationships shown, and the same way.
3. Add multiplicity constraints.
4. Where are the tables?

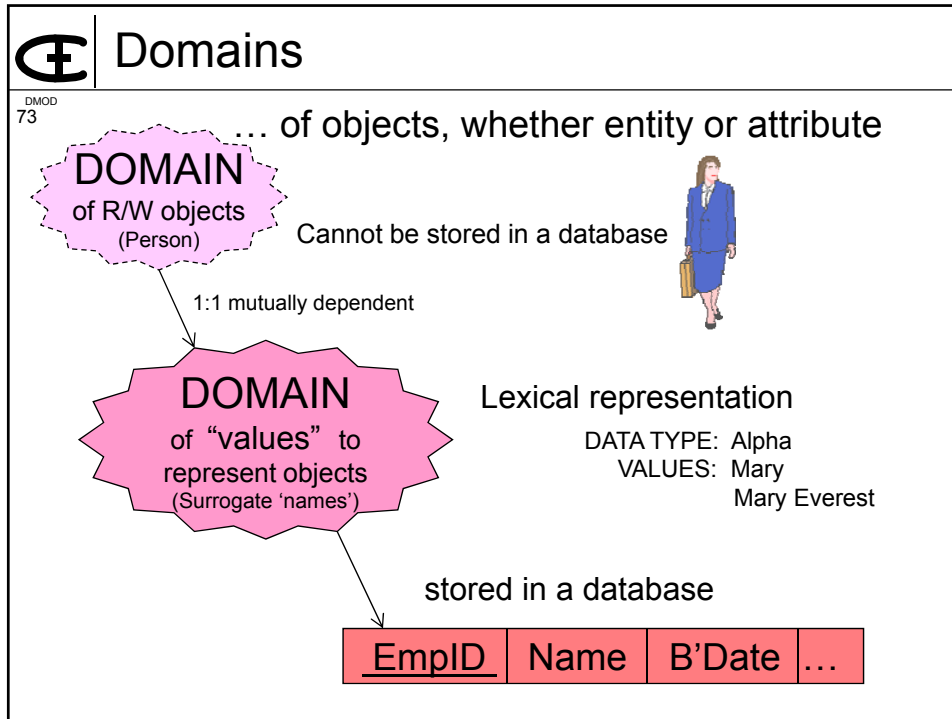
Which is easier to design? To understand?

<u>X</u>	A	B	A <sub>2</sub>
----------	---	---	----------------

<u>Y</u>	A
----------	---

<u>A</u>	P	Q
----------	---	---

# ERel Modeling to ORModeling



E

## ENTITY or OBJECT

Σ

DMOD 74

The main, pivotal construct in data structures and design.

- An information system deals with material objects, abstract concepts, events, etc. in the "real" world ... called *entities*.
- Arbitrarily *selected* as being of interest within the domain of interest to a community of users.
- In design, we form a *population* of similar entities based upon some common characteristics, and define an *entity type* to represent that population.
- Usually *multiple* entity instances of interest - form a population.
- All entity type populations are *mutually exclusive... to the system*  
(non-overlapping)
- Some may be *independent* "base" entities, others dependent.
- An entity is anything about which we store facts.
- Something distinct, identifiable, distinguishable, separable.
- The entity type *definition* is the template or "schema" by which all members of the population are described.

# ERel Modeling to ORModeling

E

## Data Modeling

ORM/ER  
75

### THINKING ABOUT ATTRIBUTES:

Record-Based (ER):

Object-Role (ORM):

**OBJECTS (ENTITIES) have "ATTRIBUTES" (DESCRIPTORS)  
by playing ROLES in RELATIONSHIPS with other entities.**

E

## Record-based Design

B

ORM/ER  
76

### WHAT DOES THIS "RECORD" REPRESENT?

ENTITY:

<u>X</u>	A	B	C
----------	---	---	---

<u>X</u>	A
----------	---

Design minimal "records" with *at most one* non-key domain.  
Remedy for Normal form violations is Decomposition.  
This is the ultimate end of Record Decomposition.

<u>X</u>	B
----------	---

*Now what do these "records" represent?*  
Perhaps Codd was right in naming it a \_\_\_\_\_!


<u>X</u>	C
----------	---

Avoids spurious associations, e.g., A-B ...  
Could there be any violations of normal forms?


What about representing the entity **X**? or any domain?

What if **A** is related to (or attribute of) other "entities"?

# ERel Modeling to ORM Modeling



## Transform Record-based (ER) Design



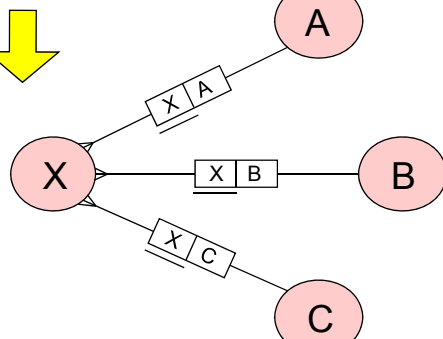
ORM/ER  
77

TO REALLY REPRESENT THE ENTITY DOMAINS


*What do you assume?*


<u>X</u>	A	B	C
----------	---	---	---

**O**bject  
**R**ole  
**M**odel:



*OBJECTS (ENTITIES) have "ATTRIBUTES" (DESCRIPTORS) by playing ROLES in relationships with other entities.*



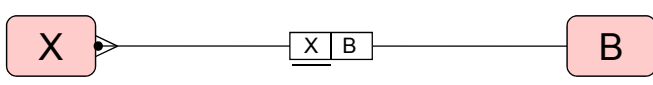


## ORM Basic Diagram

ORM/ER  
78

An elementary fact sentence – the building block

Object	Predicate	Object
(Employee)	role of X   role of B (works in   employs)	(Department)







*Describe each of the Object Populations.*

*Describe the Relationships.*

*Describe the Constraints.*

# ERel Modeling to ORModeling

	<h2 style="margin: 0;">Record-Based Modeling</h2>	Ex.1	
<small>ORM/ER</small> 79	<p>GIVEN TWO FACTS (conceptually):</p> <ul style="list-style-type: none"> <li>• one about the CITY a PERSON <i>lives in</i></li> <li>• another about the CITY a PERSON <i>works in</i></li> </ul> <p>ASSUME:</p> <ul style="list-style-type: none"> <li>• every person has to live and work in a city</li> <li>• each person can live and work in only one city (at a time)</li> <li>• not interested in anything more about persons or cities</li> </ul> <p>EXAMPLE (two elementary fact instances):</p> <ul style="list-style-type: none"> <li>• Gordon Everest lives in Roseville and * works in Minneapolis</li> </ul> <p> <b>DIAGRAM A CONCEPTUAL DATA MODEL</b>                  – to represent this information (a database to contain these facts)</p>		

	<h2 style="margin: 0;">Record-Based Data Model</h2>	Ex.2								
<small>ORM/ER</small> 80	<p>for PERSON <i>lives in / works in</i> a CITY</p> <ul style="list-style-type: none"> <li>• What is the entity and what is the attribute?</li> <li>• Would it make any sense to say (to a novice layperson - a user):                         <ul style="list-style-type: none"> <li>– CITY was an "attribute" of PERSON not an entity (no table)?</li> </ul> </li> <li>• Doing more than is necessary at the conceptual level</li> </ul> <table style="border-collapse: collapse;"> <tr> <td style="padding: 5px;">PERSON</td> <td style="padding: 5px;"> <ul style="list-style-type: none"> <li>• cannot have CITY and CITY as attributes of PERSON</li> <li>• column/attribute name reflects " entity + role "</li> <li>• CITY as an entity/object is lost (not its own table)</li> <li>• what if there is a CITY where no one lives or works</li> <li>• some add concept of a DOMAIN in SQL (but in a Relational database it is not in our table diagram!)</li> </ul> </td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">PersonID [key]</td> <td></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">LiveCity</td> <td></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">WorkCity</td> <td></td> </tr> </table>		PERSON	<ul style="list-style-type: none"> <li>• cannot have CITY and CITY as attributes of PERSON</li> <li>• column/attribute name reflects " entity + role "</li> <li>• CITY as an entity/object is lost (not its own table)</li> <li>• what if there is a CITY where no one lives or works</li> <li>• some add concept of a DOMAIN in SQL (but in a Relational database it is not in our table diagram!)</li> </ul>	PersonID [key]		LiveCity		WorkCity	
PERSON	<ul style="list-style-type: none"> <li>• cannot have CITY and CITY as attributes of PERSON</li> <li>• column/attribute name reflects " entity + role "</li> <li>• CITY as an entity/object is lost (not its own table)</li> <li>• what if there is a CITY where no one lives or works</li> <li>• some add concept of a DOMAIN in SQL (but in a Relational database it is not in our table diagram!)</li> </ul>									
PersonID [key]										
LiveCity										
WorkCity										

# ERel Modeling to ORModeling

E

## Object-Role Model

Ex.3

ORMWER  
 81

for: PERSON *lives in* / *works in* CITY

**FORML** language statements (verbalization):

- PERSON *lives in* CITY
- Every PERSON *lives in* some CITY
- Each PERSON *lives in* at most one CITY

... Similarly for *works in*

FACT

- Required / Mandatory
- Unique (Identifier)

G

Now add: PERSON *makes sales calls in* multiple CITIES

E

## Steps in ORM Modeling

ORModig  
 82

- Start from what you know; what the user domain experts tell you



1. Break down to elementary fact sentences
2. Find the nouns => become the objects
3. Find the verbs => relationships/predicates
4. Find the constraints/business rules  
in words like: every, must, may, many, only, if, when, plurals...
5. Put them all in a diagram
6. Validate the diagram (internal and external)
7. Generate the relational tables


Verbalization of a diagram for people (going backwards)

## ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November

Page 42

	<h3>ORM - Digging a little deeper</h3>
<small>ORMODLG</small> 83	<p>THE PROCESS: Familiarize &gt; Verbalize &gt; Symbolize</p> <ul style="list-style-type: none"><li>• Gather evidence from User domain<ul style="list-style-type: none"><li>– look at forms, reports, files</li><li>– what users say; what users do</li></ul></li><li>• Transform into Elementary Facts</li><li>• Discover Objects (nouns) and Relationships (verbs)</li><li>• Document findings in:<ul style="list-style-type: none"><li>– Elementary fact sentences</li><li>– ORM Data Model Diagram</li></ul></li></ul> 

	<h3>“New” Modeling Approach</h3>
<small>ORMVER</small> 84	<ol style="list-style-type: none"><li>1. Start with users’ description of the domain to be modeled; break the narrative down into elementary fact sentences; identify the nouns (objects), verbs (relationships), and constraints</li><li>2. Think <u>object populations</u> (types) for entities and “attributes”<ul style="list-style-type: none"><li>- give each a name, description, and criteria for inclusion</li></ul></li><li>3. Think <u>relationships</u> among members of those populations</li><li>4. Define the <u>characteristics</u> of those relationships (both ways)<ul style="list-style-type: none"><li>- dependency/optionality, multiplicity/exclusivity</li></ul></li><li>5. Define additional <u>constraints</u> on objects and relationships<ul style="list-style-type: none"><li>• Draw/add to a diagram at each stage.</li><li>• Then you are ready to automatically derive the tables</li></ul></li></ol> <p>NOTE: No tables, no identifiers, no data item types, no foreign keys, no distinguishing attributes vs. entities... yet!</p> <p>FOCUS on object populations and relationships.</p> <p style="text-align: center;"><b>TABLE THINK</b> puts the cart before the horse and frames our view of the world.</p>

# ERel Modeling to ORM Modeling

E

noun  
verb  
constraint

## Verbalize User Descriptions

ORMDLG

85 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.

E

noun  
verb  
constraint

## “New” Modeling Approach - Example

ORMVER

86

- First Object populations (nouns)
  - Each has its own “table” with one column – the identifier

X

A

X

A
- Then Relationships (verbs) and Roles w.r.t. each other
  - Not presuming the nature of the relationship

X

R

A

X A

*What is the identifier?*
- Then add Constraints
  - Dependency/optionality, and multiplicity/exclusivity

X

R

A

X A

Now we can build a table:

**NOTE: This is the relationship!**

# ERel Modeling to ORModeling

## Transforming ERel to ORM

ORM/ER  
87

**LEGEND:**  
Primary Key  
 Foreign Key  
 1 — M

Customer
Order
Line Item
Item

C N A P
O D T C
I Q P<sub>r</sub>
I D<sub>e</sub> P<sub>r</sub> L

ADD Foreign Keys (redundant). **NOTE: Duplicate fields - C P<sub>r</sub> I O .**  
 Break out all Entities and Attributes into separate Objects:  
 ADD Relationships (presuming above to be fully normalized):

**NOTE: All Object Domains are shown only once!**  
**NOTE: All Functional Dependencies are now explicitly shown!**

## Data Modeling in ORM

ORM/ER  
88

Try it yourself, starting with =>

Modify / extend the diagram with these semantics:

For each X:

1. A is REQUIRED (●)
2. A is UNIQUE for all X
3. A is MULTIVALUED(←)
4. A is INDEPENDENT (!)  
i.e., can be ORPHAN

Also:

5. B is FUNCTIONALLY DEPENDENT on A
6. B and C are RELATED
7. D is an ATTRIBUTE of A

**NOTE:**

- No "Attributes"
- No "TABLE THINK"
- No Foreign Keys
- No Normalization
- Focus on Object Domains
- Think *all* Relationships


N

# ERel Modeling to ORModeling


<b>Ⓔ</b>	<h2 style="margin: 0;">Elementary Fact Sentences</h2>	$\Sigma$																																	
ORMODLG 89	ENGLISH GRAMMAR - Structure of a Sentence:	HALPIN08-\$3.3.																																	
<table style="margin: auto; border: none;"> <tr> <td style="padding: 0 10px;">SUBJECT</td> <td style="padding: 0 10px;">+ PREDICATE</td> <td style="padding: 0 10px;">[ + OBJECT ]</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">↓</td> <td style="text-align: center;">↓</td> </tr> <tr> <td style="text-align: center;">NOUN</td> <td style="text-align: center;">VERB</td> <td style="text-align: center;">NOUN</td> </tr> </table> <p style="margin-top: 10px;"><i>Elementary Fact -- cannot be decomposed into pieces which collectively provide the same information as the original fact.</i></p> <table style="margin-top: 10px; margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 20px;">George</td> <td style="padding-right: 20px;"><i>runs.</i></td> <td style="padding-right: 20px;"></td> <td style="text-align: right;">=&gt; UNARY</td> </tr> <tr> <td>George</td> <td><i>runs</i></td> <td>to the store.</td> <td style="text-align: right;">=&gt; BINARY</td> </tr> <tr> <td>George</td> <td><i>likes</i></td> <td>to run.</td> <td></td> </tr> <tr> <td>NOT: George</td> <td><i>likes</i></td> <td>to run <u>and</u> jump.</td> <td></td> </tr> <tr> <td>George</td> <td><i>does <u>not</u> like to</i></td> <td>run. ... CLOSED WORLD ASSUMPTION</td> <td></td> </tr> <tr> <td>George and Mary</td> <td><i>like</i></td> <td>to run. ... together!</td> <td></td> </tr> </table> <p style="margin-top: 10px;"><u>If</u> George <i>runs</i>, <u>then</u> Mary <i>runs</i>.</p> <p><u>All</u> people who <i>run</i> are happy! ...HANDLED WITH ROLE SET CONSTRAINTS</p>			SUBJECT	+ PREDICATE	[ + OBJECT ]	↓	↓	↓	NOUN	VERB	NOUN	George	<i>runs.</i>		=> UNARY	George	<i>runs</i>	to the store.	=> BINARY	George	<i>likes</i>	to run.		NOT: George	<i>likes</i>	to run <u>and</u> jump.		George	<i>does <u>not</u> like to</i>	run. ... CLOSED WORLD ASSUMPTION		George and Mary	<i>like</i>	to run. ... together!	
SUBJECT	+ PREDICATE	[ + OBJECT ]																																	
↓	↓	↓																																	
NOUN	VERB	NOUN																																	
George	<i>runs.</i>		=> UNARY																																
George	<i>runs</i>	to the store.	=> BINARY																																
George	<i>likes</i>	to run.																																	
NOT: George	<i>likes</i>	to run <u>and</u> jump.																																	
George	<i>does <u>not</u> like to</i>	run. ... CLOSED WORLD ASSUMPTION																																	
George and Mary	<i>like</i>	to run. ... together!																																	

<b>Ⓔ</b>	<h2 style="margin: 0;">Elementary Binary Fact</h2>	
ORMODLG 90	(Instance vs. Type)	
<p><u>Fact Instance:</u></p> <p>PERSON with name 'Jones' <i>works in</i> DEPARTMENT with code '2110'</p> <p><b>BINARY:</b> having 2 object 'holes'</p> <p style="text-align: center;">“ _____ <i>works in</i> _____ ”</p> <p><u>Elementary Binary Fact Type:</u></p> <p style="text-align: center;">“PERSON <i>works in</i> DEPARTMENT”</p> <p>We work at the schema (meta) level in data modeling.</p>		

# ERel Modeling to ORModeling



## Which are Elementary Facts?




ORMODLG  
91
HALPIN08-§3.3.

*Elementary Fact -- cannot be decomposed into 'sub'facts which collectively provide the same information as the original fact.*

- “Ann smiles”
- “Ann smiles frequently”
- “Ann smiles and laughs”
- “Ann smiles and Bob smiles”
- “Ann smiles or Bob smiles”
- “Ann does not smile” ... CLOSED WORLD ASSUMPTION
- "Ann smiles at Bob"
- "Ann & Bob smile at each other"
- “Ann smiles at Bob in the morning”
- “If Ann smiles, then Bob smiles”
- “All people who smile are happy” ...HANDLED WITH ROLE SET CONSTRAINTS
- “Ann goes to School in Kansas” ... 4NF (2 binaries?), 5NF (all combinations?)

NOT USE - Logical Connectors (NOT, AND, OR, IF) or  
- Logical Quantifiers (ALL, SOME).

N5
4



## Objects

ORMODLG  
92

- Things in the “Real” World
  - Reference/Base/independent Entities
  - Dependent (“weak”) Entities
  - Events
  - “Terminal” things - we have no further interest in
  - Compound entities = an “objectified” relationship/predicate
- Naming
  - A surrogate to represent and reference those things
  - Singular noun
  - INSTANCE (VALUE) e.g., “George”
  - ROLE e.g., “Employee”
  - GENERIC e.g., “Person”
  - REFERENCE MODE e.g., “name”
- Type
  - a population of similar things (“instances”)
    - Are we interested if there is only one?
  - Disjoint from all other object populations (mutually exclusive)

# ERel Modeling to ORM Modeling

E

Fact Sentence - Verbalize

ORMODLG  
93
HALPIN08-§3.3.

- A Fact = a Predicate + Object(s) => Sentence
- THINK: Objects... playing Roles... in a Relationship
- Naming: object instances versus object types
  - e.g. "Ann" is an instance of "Person"
- Arity - the number of object "holes" in the Predicate
  - UNARY:               - "Ann *smiles*"
    - only 2 states: true/false, present/absent, yes/no
    - making the closed world assumption
  - BINARY:             - "Ann *likes to* run"
    - most common
    - has an inverse - "Running *is liked by* Ann" (passive voice)
    - Inverse name is never the same (else symmetric, handled differently)
  - TERNARY:           - "Ann *married* Bob *in* 1967"
    - with types:       - "PERSON *married* PERSON *in* YEAR"
    - verbalizing can be difficult with more than 2 (sequence problem)

E

Symbolize: ORM Constructs

ORMODLG  
94
HALPIN08-§3.4.

- OBJECT (ENTITY, CONCEPT) - noun ... in a "soft" rectangle
- PREDICATE (RELATIONSHIP) - verb ... in a box = *role name(s)*.
  - one or more roles: unary, binary, ternary, +++

Unary Predicate:

# ERel Modeling to ORM Modeling

E

Symbolize: ORM Constructs

ORMODLG  
95
HALPINo8- §3.4.

Often wish to generalize a Unary Predicate to a Binary Predicate:

What about 'Person Frowns'? ... or laughs? or ...

E

Symbolize: ORM Constructs

ORMODLG  
96

Binary Predicate:

Elementary Binary Fact Sentence:

Verbalization:

*“PERSON works in DEPARTMENT”*  
*“DEPARTMENT employs PERSON”*

# ERel Modeling to ORM Modeling

**Ⓔ** Exercise – ORM Diagrams
G

ORMODLG  
97
HALPIN08-Ex.3.4 #8, p.94.

Find the illegal ORM Diagrams; explain why:

A

B

C

D

E

F

G

H

I

J

L

RULE:

H

**Ⓔ** Nested 'Objectified' Predicate/Fact

ORMODLG  
98
HALPIN08-§3.4, p.88.

Adding 'attributes' to a relationship:

What kind of relationship?

What must exist to represent the relationship?

Where do we put the Grade?

What is the determinant of Grade?





# ERel Modeling to ORModeling

**GE**
**Uniqueness Constraints - Exercises**
G

ORMCON 103

- The keys for certain fact types are as shown. On this basis, which of these fact types are definitely splittable? – Halpin<sub>08</sub> Ex.§4.5, #1.

(a)

(b)

(c)

(-)

(e)

(f) Okay?

(g)

(h) Why not?

RULE:

**GE**
**Primary Reference Identifier**

ORMCON 104

- an External Uniqueness Constraint (U) is like a Composite Key. If object reference scheme requires a composite key, replace (U) with (P) for Primary Key:

E.g., Room numbers are only unique within a Building, so a composite of Building Name and Room Number is required to uniquely identify a physical room.

In ORM these are equivalent. ... tables?

NOTICE the placement of the forks.

# ERel Modeling to ORModeling

E

Mandatory Role Constraint

ORMCON 105
Halpin08-§5.2.

- the Optional / Dependent characteristic of a relationship
- also called:
  - total role constraint (other notations:  $\forall$  (T) )
  - exhaustibility

– every EMPLOYEE *must* work in some DEPARTMENT.

- Disjunctive mandatory: every EMPLOYEE *must* receive a SALARY *or* earn a COMMISSION (or both = inclusive)

- External Disjunctive Mandatory:
  - Verbalize it:*

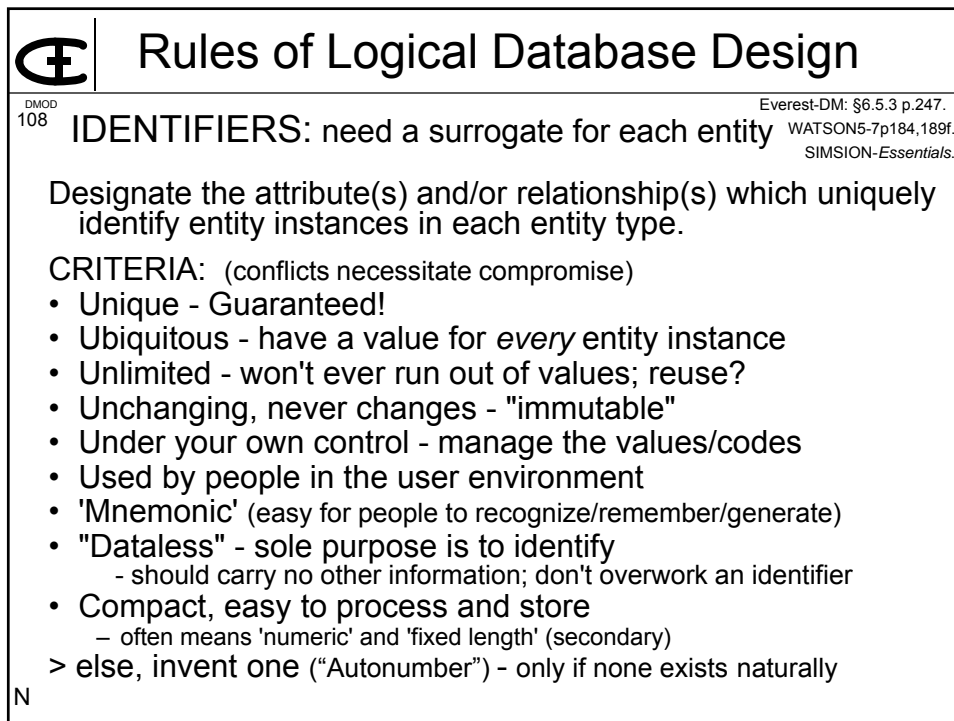
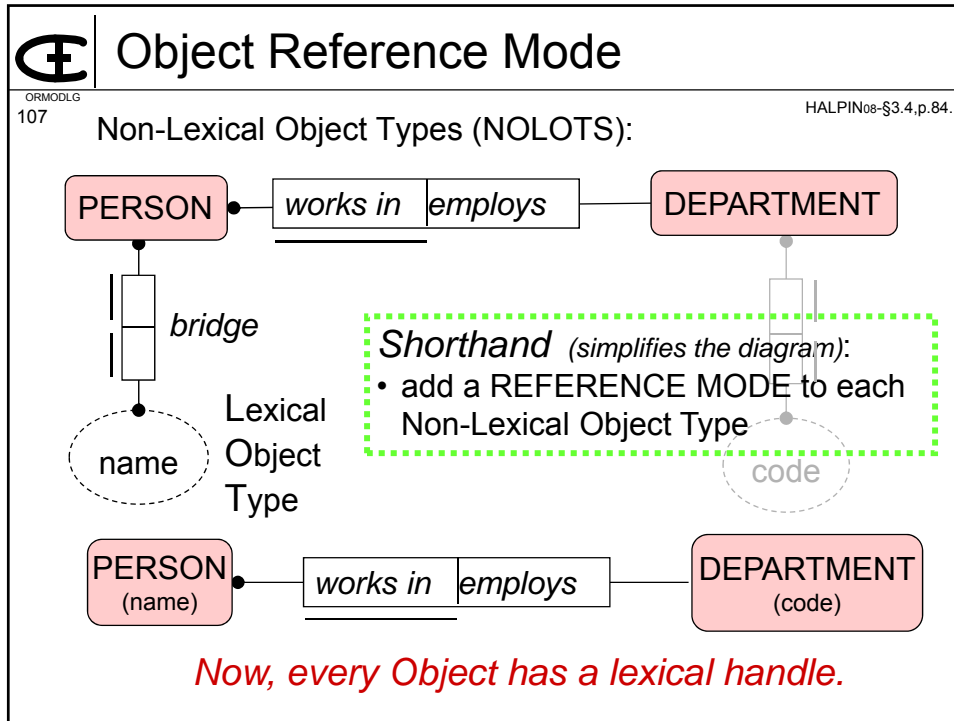
E

Mandatory Role Constraint - Rules

ORMCON 106
Σ


- Every Object (type) has a population.
- Every Role has a population... *of what?*
- The Pop(Object) = UNION of Pop of *all* its roles, except...
- World recorded in the database cannot be stronger than what can be tolerated in the real world
- All roles in a predicate are mandatory (all or nothing)
- Mandatory constraint is *implied* if an object plays only one role in the database (not shown unless...)
- Default Disjunctive Mandatory is always implied across *all* roles played by an object in the database, i.e., each object instance (value) must play a role with at least one other object in the database ...
- If not, i.e., there may be orphans of that object type, then must declare the object to be INDEPENDENT. (shown in *NORMA* with a ! after the name)
- NEVER show the implied disjunctive mandatory constraint - *why?*

# ERel Modeling to ORModeling



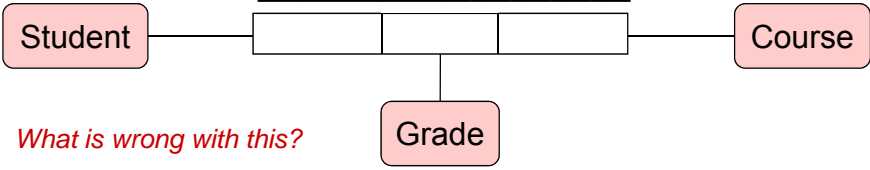
# ERel Modeling to ORM Modeling

DAMA Day, Madison, WI, 2011 November

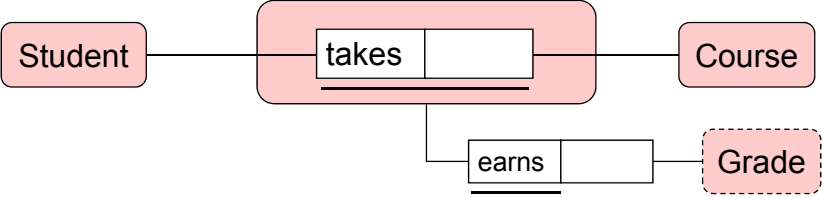


## Ternary++ Relationship Nested 'Objectified' Predicate/Fact


ORMCONSTR  
109



What is wrong with this?



- Equivalent, ... IF grade is mandatory. *How?*
- Ternary is best for clarity, ... if only one "attribute" on the relationship.
- Nested is best if... several roles on the nested predicate (avoids redundancy) and/or more might be added later.

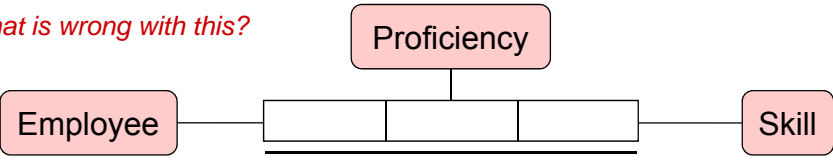


## Ternary Facts... are suspect (4NF)

ORMCONSTR  
110
HALPIN08-\$4.3.

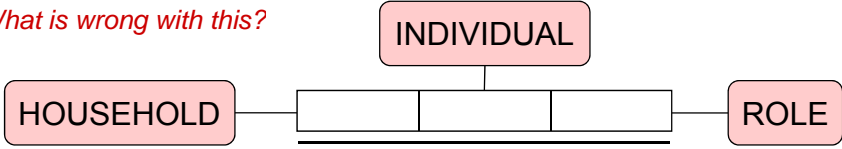
- Must examine carefully (if you really think you have one)

What is wrong with this?



For any two, can there be multiple values of the third?

What is wrong with this?



# ERel Modeling to ORModeling

## Ternary++ Relationships

ORMCONSTR 111
HALPIN08-§3.4,p.88.

*What about this?*

CONTRIBUTION:

DATE	DONOR	AMOUNT	FUND
------	-------	--------	------

*In "Table Think"  
what is the first question you ask?*

In ORM, define in sequence:

1. Objects / Domains
2. Relationships between/among them
3. Reference Mode (Identifier) for each Object
4. Constraints.

*ORM gives an explicit means to analyze and represent higher order relationships.*

## Value Set Constraint

ORMCONSTR 112
Halpin08-§6.3.

On the Population of an Object, defined by:

- Enumeration - { M , F }
- Range (if an ordering) – { 1 ... 10 }
- Pattern of Characters –
  - e.g. { a15} = 15 alpha characters
  - e.g. { d6.d2 } = up to 6 digits followed by 2 digits after a decimal point
- Reference Entity (Table) or Master List (of codes) to contain all possible values of the Entity, *even if not used elsewhere in the database.*
  - Declare the Entity as 'Independent' to allow for orphans, e.g., ...

Object

**SKILL !**  
(code)

# ERel Modeling to ORModeling

E

Frequency Constraints

ORMCONSTR 113
Halpin08-§7.2.

- **Role Frequency**
  - Limits the number of times an object can play a role; or the number of times a role (or role combination) can appear in a fact table.
  - Place on the role(s) of the Predicate:
    - an 'n' or a range
    - optionally, with comparator operators ( <, >, ...)
    - or a Range - (min ... max)
- **Object Cardinality** (indirect in *NORMA* thru Value Set)
  - Limits the size of an Object Population

Not in Record-based modeling schemes!

E

Role Population Constraints

ORMCONSTR 114
Halpin08-§6.4.

- Roles must be on the *same* entity type population
- Only meaningful if *both* roles are optional
- Applies at the type level, on whole populations
- Based on *existence* in populations, not value-based

May also apply to multiple roles in the predicates. (a composite)

<p><b>EQUALITY</b></p> <p>R1 IFF R2</p> <p>R1 ↔ R2</p> <p style="text-align: center;">⊖</p> <p>pop(R1)=pop(R2)</p>	<p><b>SUBSET</b></p> <p>If R2, then R1.</p> <p>R2 → R1</p> <p style="text-align: center;">⊆</p> <p>pop(R2)≤pop(R1)</p>	<p><b>EXCLUSION</b></p> <p>If R2, then -R1 and vv.</p> <p style="text-align: center;">⊗</p> <p>pop(R1)^pop(R2)=NULL</p>
--	--	---

Not in Record-based modeling schemes!

# ERel Modeling to ORModeling

E

## Homogeneous Predicate Reflexive Relationship, Ring Fact Type

ORMCONSTR 115
Halpin08-\$7.3.

- PERSON ---- *parent of* | *child of* ---- PERSON (M:M)
- PERSON ---- *mother of* | *child of* ---- PERSON (1:M)
- PERSON ---- *husband of* | *wife of* ---- PERSON (roles)

PERSON

parent of

child of

### Ring Constraints

Ann	Ann	NO ->	<u>I</u> Rreflexive
Ann	Barb	TRUE	
Barb	Ann	NO ->	<u>A</u> Symmetric => IR
Barb	Curt	TRUE	
Ann	Curt	NO ->	<u>I</u> nTransitive => IR
Curt	Ann	NO ->	<u>A</u> Cyclic => AS => IR

(=> implies)

Not in Record-based modeling schemes

E

## Ring Constraints

ORMCON 116

Applies at Instance Level.

IF

→

THEN

→

IRreflexive:  $\neg(xRx)$

SYMmetric: IF  $xRy$ , THEN  $yRx$  (forced)

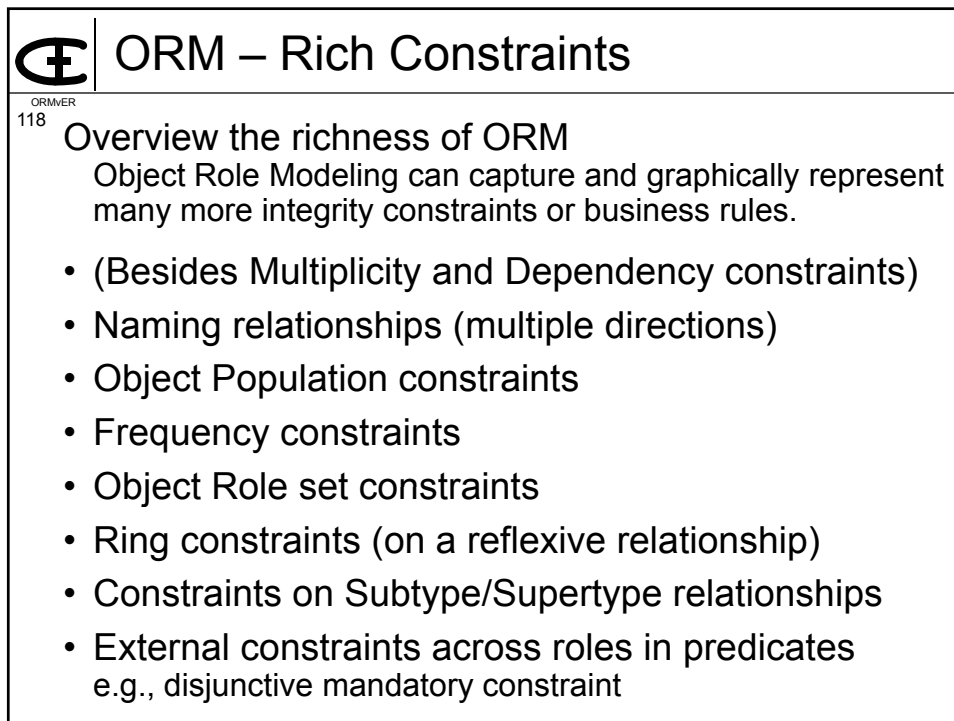
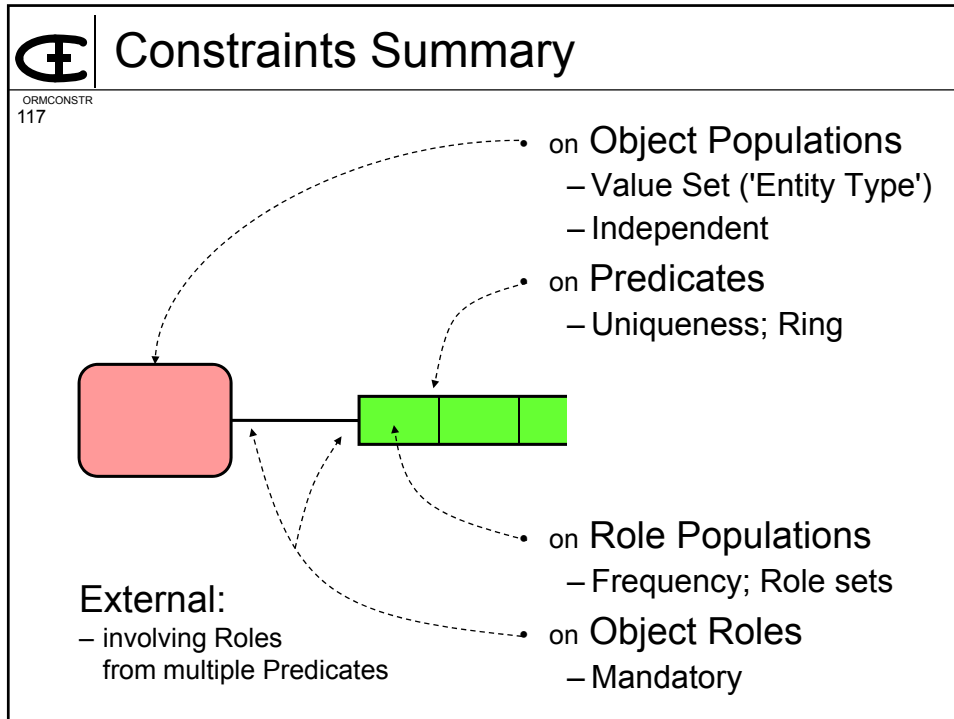
ANtiSymmetric: IF  $xRy$ , THEN  $\neg(yRx)$ , but  $x=y \neq \Rightarrow IR$

ASymmetric: IF  $xRy$ , THEN  $\neg(yRx)$ .  $\implies IR$

InTransitive: IF  $xRy$  AND  $yRz$ , THEN  $\neg(xRz)$ .  $\implies IR$

ACyclic: IF  $xRy$  AND  $yRz$ , THEN  $\neg(zRx)$ .  $\implies AS (IR)$

# ERel Modeling to ORM Modeling



# ERel Modeling to ORM Modeling

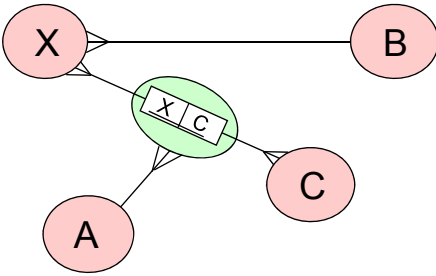
E

Deriving the Tables

ORMWER  
119

Two simple rules: GIVEN a complete ORM diagram:

1. Each object which has a M:1 relationship with any other object, gets its own table.  
 Conversely, each object which *only* plays a functional dependency on other objects does NOT get its own table (unless declared independent)
2. Every relationship (predicate) which is based on a composite key, gets its own table.



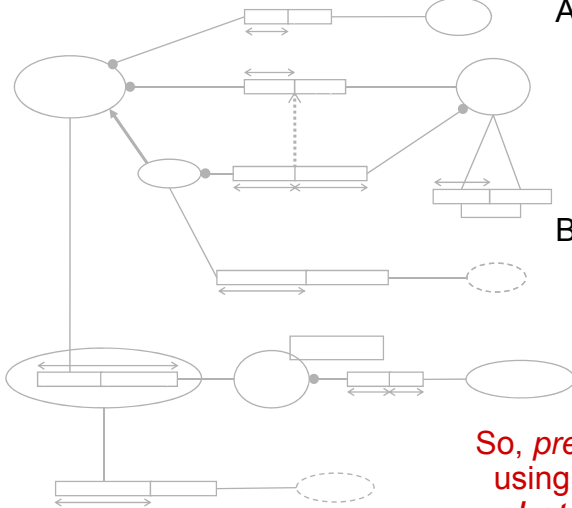
<u>X</u>	B	
<u>X</u>	C	A

Guaranteed to be fully normalized!

E

ORM Data Model - Presentation

DMODPRE  
120




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 *abstractions*.

© Gordon C. Everest, All rights reserved.

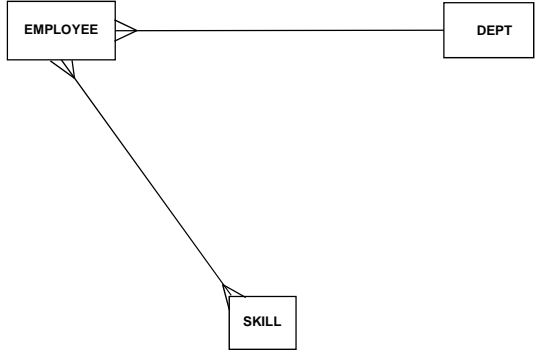
# ERel Modeling to ORM Modeling

DAMA Day, Madison, WI, 2011 November



DMODPRE  
124

## Abstractions of ORM Data Model




```
graph TD
    EMPLOYEE --- DEPT
    EMPLOYEE --- SKILL
```

1. Hide "Terminal" (M:1) Objects ( $\Rightarrow$  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

*Is this the same data model we started with?*

*$\Rightarrow$  A High-level Abstract Conceptual Data Model...  
an ER Diagram ?!!!*



ORMVER  
122

## This "New" Approach is not so New

- Binary Modeling - 1976 Sjr Nijssen
- Nijssen Information Analysis Method (NIAM)
- Others: M. Senko, Wm. Kent, SUMM Meta Model, Express, ...
- Object Role Modeling-ORM, 1989, Terry Halpin
- Fact Oriented Modeling-FOM (Europe)

## ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November

Page 62



### Why NIAM/OR Modeling?

ORMINTRO  
123

- roots in both LOGIC & LINGUISTICS
- based on one modeling construct: the fact sentence
- o more expressive, understandable - diagrams & verbalization
- more, richer semantics in the diagram (than E-R, EER, IDEF1X)
- capture and represent all functional dependencies  
=>avoids normalization problems with record-based modeling
- diagrams can be populated with actual data samples
- more stable under changes to the application domain
- abstraction levels equivalent to E-R modeling
- better meets criteria for good data modeling
- organizations that switched wouldn't go back to E-R
- direction of Standards (SUMM, UDM, ...)
- now supported with a viable PC-based CASE tool



### Why Not NIAM?

ORMINTRO  
124

*If NIAM/OR modeling is superior why is it not more popular and widely used; why the lack of support?*

- Few academics involved
- CDC kept it proprietary - in Europe and USA
- Lack of strong vendor support
- Fighting the dominance and "blind" acceptance of ER/Table Diagramming and Relational DBMS
- Difficult to grasp without good training
- Too detailed; bottom up design - NOT!
- Lack of support for mental chunking and Levels of Abstraction - NOT!
  - ENTITIES, ATTRIBUTES, VALUES (LOTS), RELATIONSHIPS are all represented the same way. Diagrams should give visual prominence in proportion to semantic importance.

## ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November

Page 63



### What's Done in ORM?

ORMINTRO  
125

that is different than in ER/Relational modeling:

- Uses two constructs, OBJECT and ROLE/Relationship, rather than three as in ER: Entity, Attribute, Relationship
  - Elementary FACT SENTENCE is the basic construct  
OBJECT = Subject, Object (noun)  
RELATIONSHIP = Predicate (verb)  
hence can directly *verbalize* a data model,  
including all the constraints in the model diagram
  - Represents *all* DOMAINS *directly* (Entity, Attribute Value)
  - Explicitly represents *all* Relationships  
including all Functional Dependencies.
- => *Defers clustering attributes to form entity tables.*



### Resulting Benefits of ORM

ORMINTRO  
126

Benefits over *all* record-based modeling schemes:

(ER, EER, IDEF1X, IE, Barker, Relational, SOM, OO, UML, ...)

- No need to determine *a priori* if something is an entity or an attribute (i.e., will have its own Table)
- Avoids the traps of "Table Think"
- Direct, unambiguous representation of higher-order relationships (ternary, ++)
- Can represent more, richer semantics and constraints in a data model diagram
- Can automatically generate the relational data model
- Generates a fully normalized relational data model

**NO disadvantages for "conceptual" data modeling**  
(vs. ER, record-based, high-level "logical" data modeling)

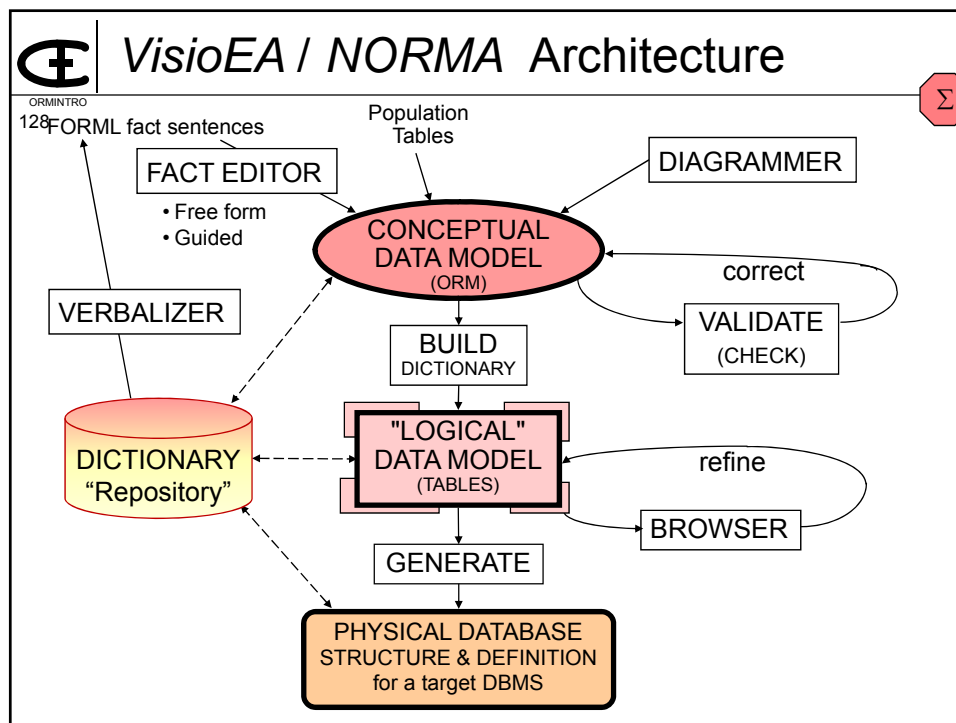
# ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November

## ☒ NORMA for ORM2


ORMINTRO  
127




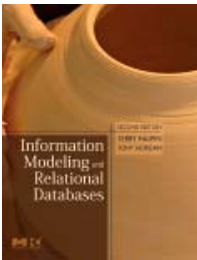
- Open source (SOURCEFORGE) - free plug-in to VS.net
- Simplified drawing rules  
Easier graphical user interface (and editor)
- Updated graphical notation (ORM2)
  - More compact display
  - Language Neutral (internationalization)
  - Agreed on by a majority of 18 experts
- Some new features
- Extended views
  - Selectively display/suppress detail
  - [Layering; several abstraction mechanisms]... to come



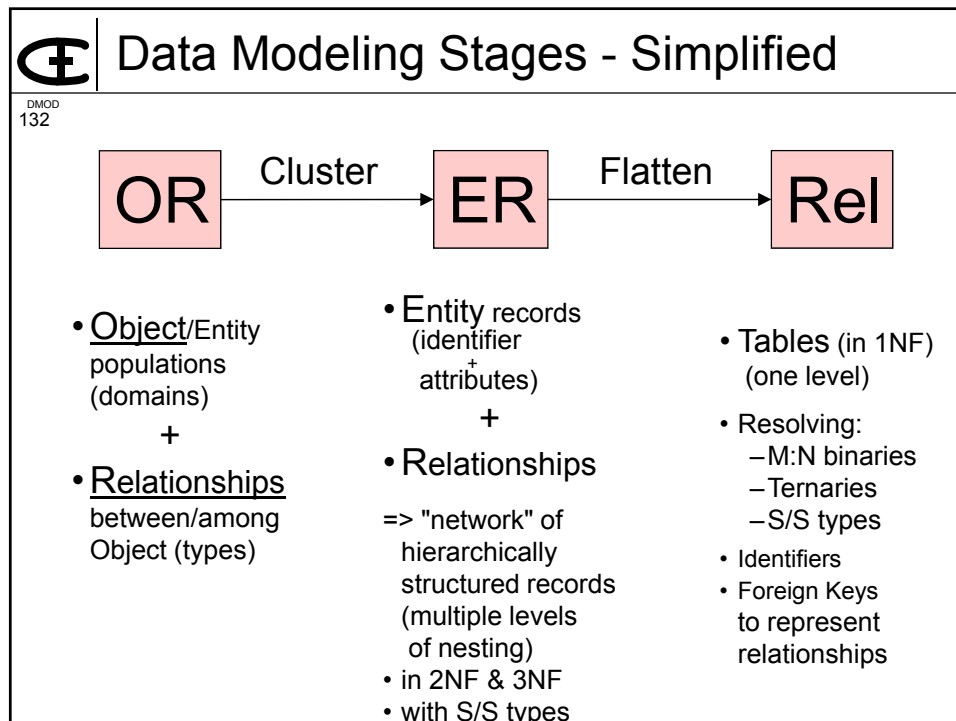
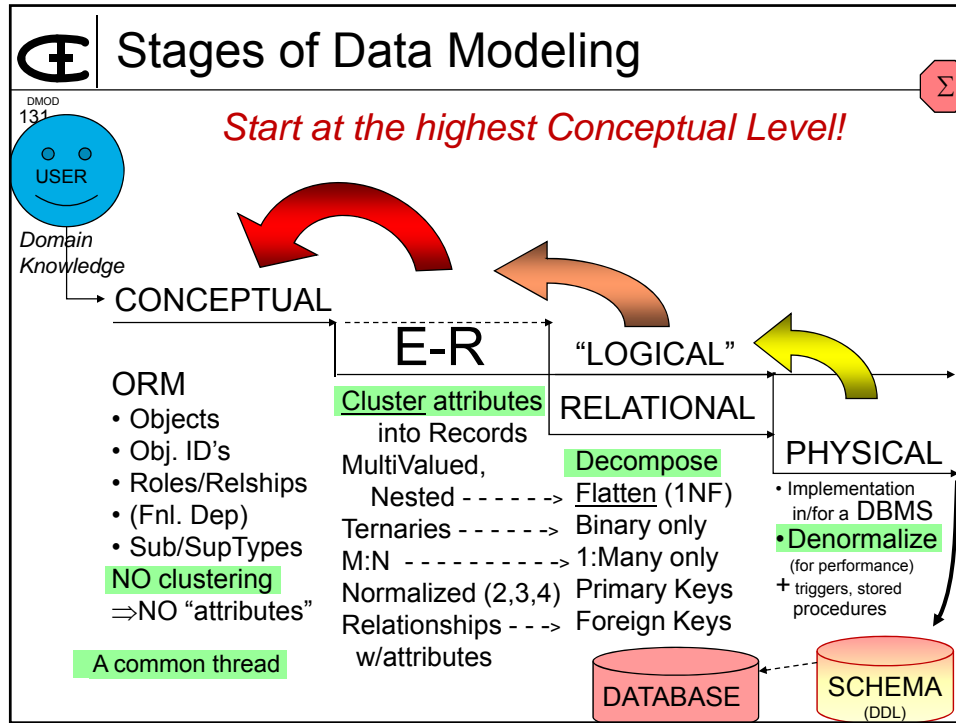
# ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November

 ORMINTRO 129	<h2>Methodology - Steps in OR Modeling</h2> <p>(Conceptual Schema Design Procedure – CSDP)</p> <ol style="list-style-type: none"><li>1. <b>Familiarize</b> with real world Universe of discourse</li><li>2. <b>Verbalize</b> sentences of elementary facts</li><li>3. <b>Symbolize</b> build the <u>conceptual ORM model</u> diagram</li><li>4. <b>Constrain</b> the roles in predicates</li><li>5. <b>Validate</b> the conceptual data model</li><li>6. <b>Map</b> into neutral, record-based, <u>logical (Relational) tables</u></li><li>7. <b>Refine</b> the table definitions</li><li>8. <b>Generate</b> <u>physical database</u> definition (DDL) for target DBMS</li></ol>
--	--

 ORMINTRO 130	<h2>Resources on ORM</h2> <p><b>BOOK:</b></p> <ul style="list-style-type: none"><li>• Terry Halpin (UQ &gt; VISIO &gt; Microsoft &gt; Neumont U &gt; LogicBlox), and Tony Morgan <i>Information Modeling and Relational Databases</i>, 2 ed, Morgan Kaufmann Publishers (Elsevier), 2008, 976p.</li></ul> <p><b>Database DESIGN TOOLS:</b></p> <ul style="list-style-type: none"><li>• <i>VisioEA</i>, in Visual Studio.net, Team editions (MicroSoft)</li><li>• <i>NORMA</i> - ORM2 tool, Open source software Free from: <a href="http://www.sourceforge.net/projects/orm">www.sourceforge.net/projects/orm</a> Runs as a plugin to VS. – Commercial version – <a href="http://www.LogicBlox.com">www.LogicBlox.com</a> (in development)</li></ul> <p><b>WEB SITES:</b></p> <ul style="list-style-type: none"><li>• <a href="http://www.ORMfoundation.org">www.ORMfoundation.org</a> – the main site; forums</li><li>• <a href="http://www.orm.net">www.orm.net</a> – Halpin's web site</li><li>• <i>Journal of Conceptual Modeling</i>, <a href="http://www.inconcept.com/jcm">www.inconcept.com/jcm</a></li></ul> <p><b>SPRING CLASS:</b></p> <ul style="list-style-type: none"><li>• IDSc 4431 (for UofMN Undergrads, Grads, working Professionals)</li><li>• Fully online</li><li>• More information at: <a href="http://www.tc.umn.edu/~geverest/DModCourse.html">www.tc.umn.edu/~geverest/DModCourse.html</a></li></ul>	  
--	---	--

# ERel Modeling to ORModeling



# ERel Modeling to ORModeling

E

## Taxonomy of Data Structures

Σ

DMOD 133
Based on what the *system* knows!
Everest-DM-4p.121.

**(1) SINGLE FILE ( E-A )**

- FLAT FILE "TABLE" \*
- HIERARCHICAL - nested repeating groups  
e.g. COBOL

**(M) MULTIFILE ( E-R --> E[A]R )**

- 'CODASYL' NETWORK - hierarchical records
- RELATIONAL ( E-A-[R] ) - flat records
- GENERAL ("M:N", TERNARY+) RELATIONSHIPS \*

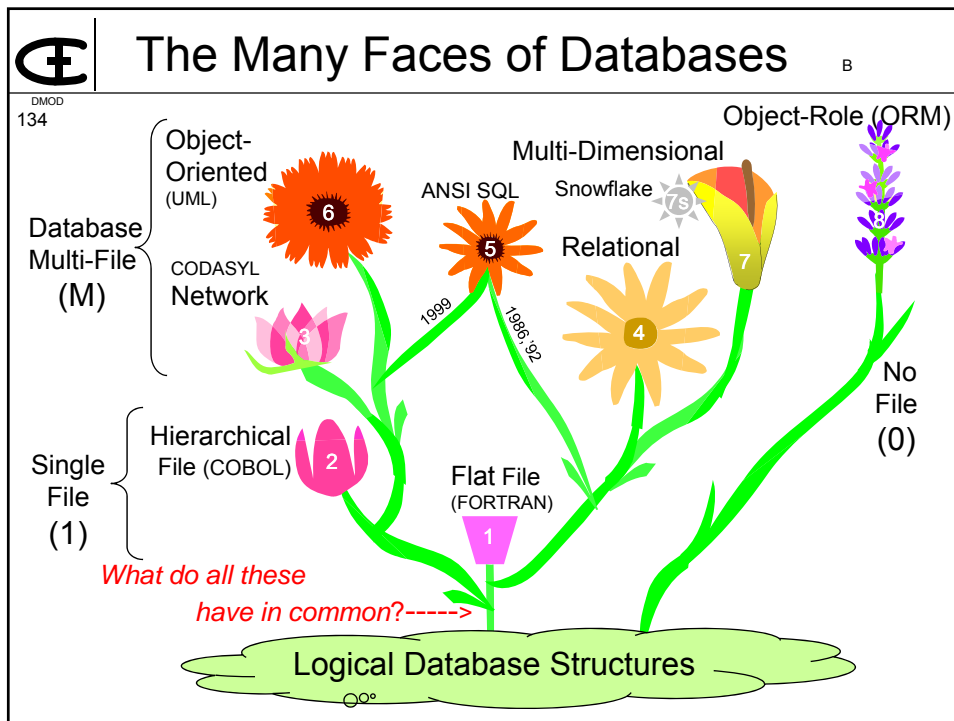
**(0) "NO" FILE \* ( O-R )** - no clustering of data items into records

- BINARY RELATIONSHIPS -e.g. NIAM/"Binary" Modeling
- GENERAL RELATIONSHIPS -e.g. ORM, Fact Oriented Modeling (Object-Role Modeling - Halpin)

RECORD-  
BASED

(Clustered  
Data Items)

\* 3 missing from the "traditional" taxonomy.



## ERel Modeling to ORModeling

DAMA Day, Madison, WI, 2011 November

Page 68



### Departing Words

ORM/ER  
135

- Give ORM/FOM a chance in your own thinking
- Learn about and explore ORM
- Go into all the world and tell them about ORM..
  - Data modeling problems stem from “TABLE THINK”
  - The Good News: There is a better way to do data modeling
    - Show them the way
  - It avoids the need for normalization
  - There are supporting software tools



### From ERel Modeling to ORModeling

136

*Questions?*

©Gordon C. Everest  
Professor Emeritus  
Carlson School of Management  
University of Minnesota

[geverest@umn.edu](mailto:geverest@umn.edu)

[www.tc.umn.edu/~geverest](http://www.tc.umn.edu/~geverest)