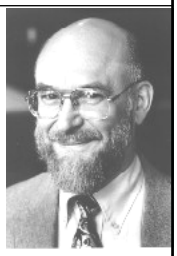
35START

DAMA-Int'l, Boston, 2007/3

Object Role Modeling



A Better Way to do Data Modeling

geverest@csom.umn.edu
©Gordon C. Everest
Professor Emeritus, Carlson School of Management
and College of Continuing Education
University of Minnesota

NIAM/ORM Teaching Experiences

ORMINTRO
2

THE COURSE:

- **Advanced Database Design & DBAdministration**
- **Taught NIAM modeling since 1975, but inadequate**

NEEDED:

- **Written Teaching Materials**
  - ***Conceptual Schema and Relational Database Design,*** **Nijssen & Halpin, Prentice-Hall (Australia), 1989; 2e 1995.**
  - ***Information Modeling and Relational Databases*****, Halpin, 2001, Morgan Kaufman.**
- **Data Modeling CASE Tool**
  - ***InfoDesigner*** **from ServerWare, 1993**
  - ***InfoModeler*** **from Asymetrix, 1994; then InfoModelers, Inc.**
  - ***VisioModeler*** **from Visio, (Enterprise 5.0) acquired 1998, released 1999 acquired by Microsoft, 1999/11.**
  - ***VisioEA*****, part of Visual Studio.net EA edition, released 2002/2**

NIAM / ORM Data Modeling

ORMINTRO
3

What's in the Name?

Nijssen --- **N**atural language
I nformation
Analysis
Method
or
Object
Role, **R**elationship
Modeling

NOTE: Has no relationship with Object-Oriented ___.

also: Binary-Relationship or just Binary

Outline

ORMINTRO
4

Roadmap for this presentation:

1. Data Modeling
   - ER (most general) => Relational Modeling
   - Problems with Record-based modeling
2. Transitioning from ER/Relational to ORM
3. Modeling steps in ORM
   - Verbalize - elementary fact sentences
   - Symbolize - ORM data model diagram
   - Add Constraints
   - Abstracting a data model diagram... for presentation
4. Place of ORM in a Taxonomy of Data Modeling Schemes
5. Summary and Reprise

Logical Database Design
Objective, Principles, Benefits

DMOD 5

OBJECTIVE of LOGICAL DATABASE DESIGN:

TO ACCURATELY AND COMPLETELY MODEL SELECTED PORTIONS OF THE REAL WORLD OF INTEREST TO A COMMUNITY OF USERS.

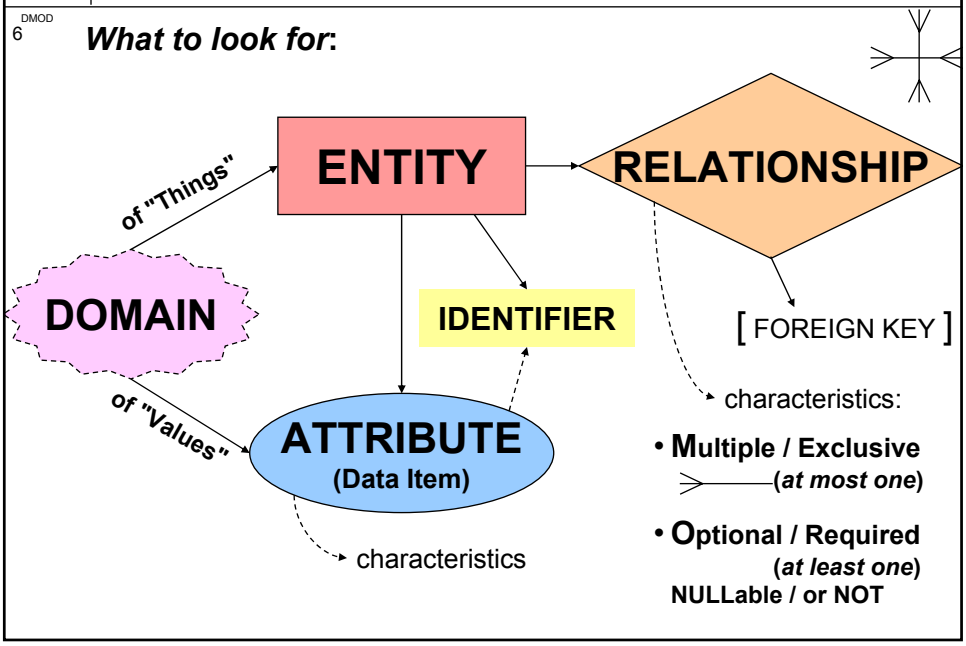
- **USERS (COLLECTIVELY) WILL ALWAYS KNOW *MORE* ABOUT A DATA STRUCTURE THAN THE SYSTEM KNOWS, OR THAN COULD BE DEFINED TO THE SYSTEM.**
- **WHAT IS NOT FORMALLY DEFINED TO THE SYSTEM, THE SYSTEM CANNOT MANAGE . . . THE USERS MUST!**
- **THEREFORE, NEED TO CAPTURE RICH SEMANTICS WITH COMPREHENSIVE DATA MODELING and DEFINITION, INCLUDING INTEGRITY CONSTRAINTS AND OPERATIONS.**

Let the 'system' do it! *Implications for a Tool!*

Data Modeling Constructs

DMOD 6

***What to look for*:**



Data Modeling Constructs

DMOD
7

***Relative emphasis differentiates Data Modeling Schemes*:**

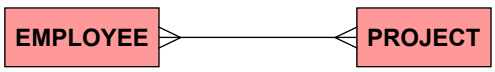
- **ER modeling focuses on**
  Entities and Relationships,
  de-emphasizing, even hiding Attributes.
- **Relational (restricted ER) focuses on**
  Entities and Attributes (single valued),
  relegating Relationships to Foreign Keys.
- **Object Role Modeling (ORM) focuses on**
  Objects/Domains and (Roles in) Relationships

Representing a M:N Relationship

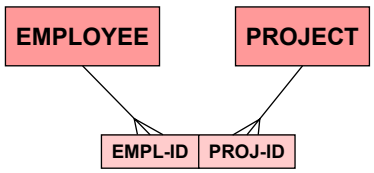
DMOD
8

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?***

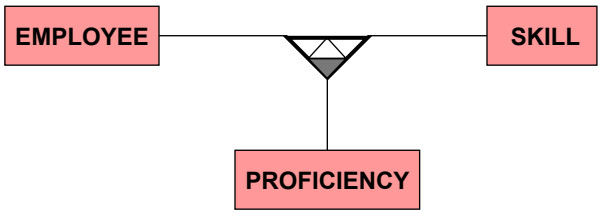
What is the problem with this representation?

N

Representing a Ternary Relationship

DMOD
9

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?**

Record-Based Modeling Ex.1 G

ORMvER
10

GIVEN TWO FACTS (conceptually):

- **one about the CITY a PERSON *lives in***
- **another about the CITY a PERSON *works in***

ASSUME:

- **every person has to live and work in a city**
- **each person can live and work in only one city (at a time)**
- **not interested in anything more about persons or cities**

EXAMPLE (a Fact Instance):

- **Gordon Everest lives in Roseville and * works in Minneapolis**

DIAGRAM A CONCEPTUAL DATA MODEL

– **to represent this information (a database to contain these facts)**

Record-Based Data Model Ex.2

ORMvER
11

for PERSON *lives in / works in* a CITY

- **What is the entity and what is the attribute?**
- **Would it make any sense to say (to a novice layperson - a user):**
  - **CITY was an "attribute" of PERSON?**
- **Doing more than is necessary at the conceptual level**

PERSON

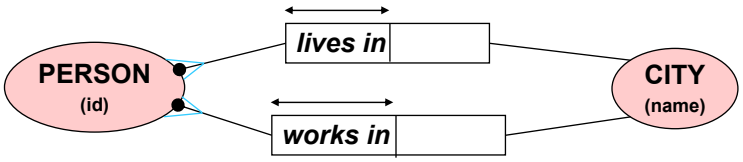
PersonID [key]
LiveCity
WorkCity

- cannot have CITY and CITY as attributes of PERSON
- column/attribute name reflects " entity + role "
- CITY as an entity/object is lost (not its own table)
- what if there is a CITY where no one lives or works
- some add concept of a DOMAIN

Object-Role Model Ex.3

ORMvER
12

for PERSON *lives in / works in* CITY



FORML language statements:

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

FACT

● Mandatory

↔ Unique

Record-Based Modeling Ex.4 G

ORMVER
13

for an additional fact:

- **A PERSON *makes sales calls* in multiple CITIES**

DIAGRAM the extended conceptual data model

- **can you add an attribute "*SalesCallCities*" to PERSON?**

FLAT Record-Based Modeling is even worse:

- **create a new table SALESCALLS with a compound key**
  - **Is this a real entity in the conceptual view?**

EXTEND THE OBJECT-ROLE DATA MODEL

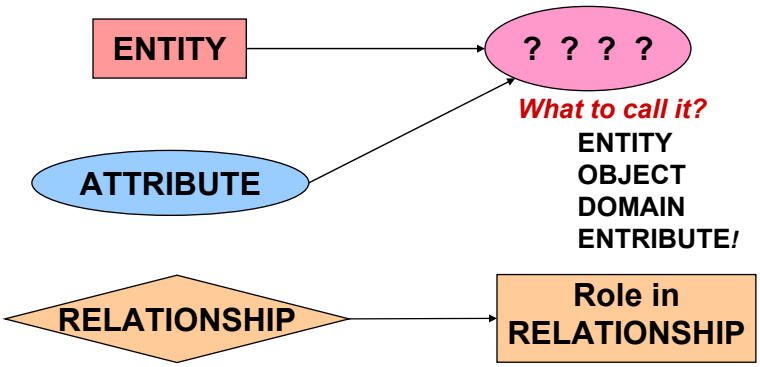
Object-Role (ORM) Data Modeling

ORMINTRO
14

THE *ESSENTIAL* DIFFERENCE:

- **Three main constructs ..rolled into.. Two main constructs**

Record-based modeling: **NIAM/ORM modeling:**



Record-based Design

B

ORMVER
15

WHAT DOES THIS "RECORD" REPRESENT?

ENTITY:

X	A	B	C

X	A

X	B

X	C

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.

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

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 other "entities"?

ER / Record-based Modeling

DMOD
16

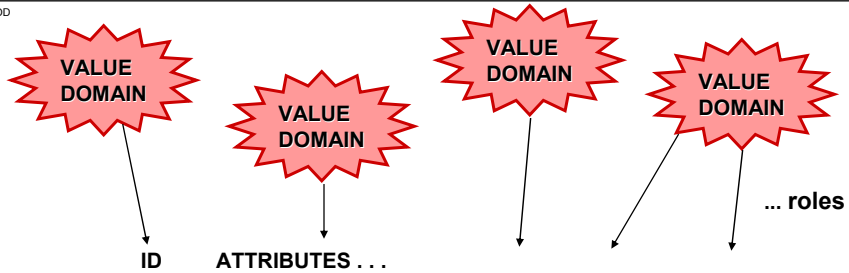


TABLE:

X	A	B	C	D

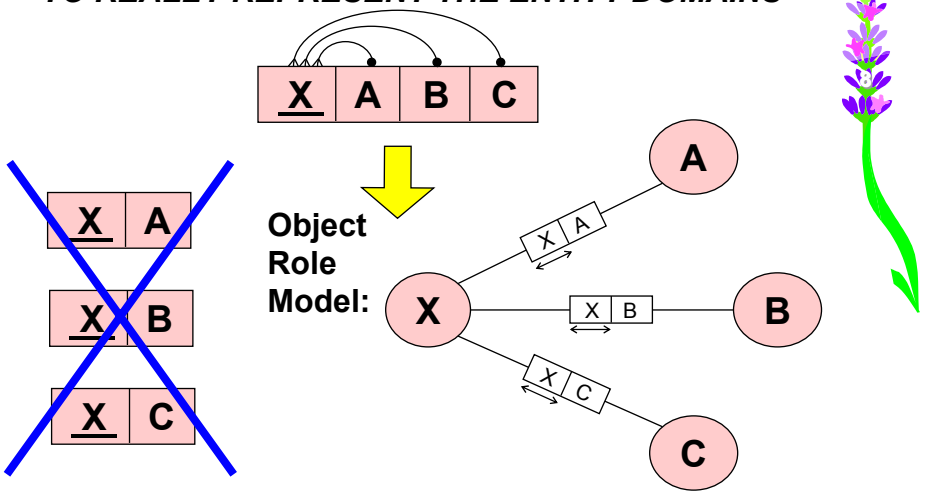
CLUSTERING of ATTRIBUTES into RECORDS/RELATIONS

- **NOT a necessary or desirable first step**
- **gets us into trouble: if too much, must decompose to normalize**

Transform Record-based (ER) Design

ORMvER
17

TO REALLY REPRESENT THE ENTITY DOMAINS



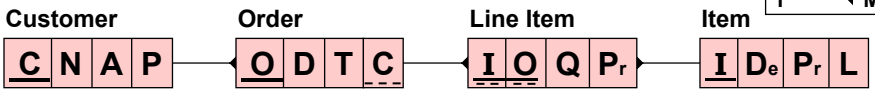
OBJECTS (ENTITIES) ***have*** **"ATTRIBUTES"** ***(DESCRIPTORS) by playing*** **ROLES** ***in relationships with other entities.***

N

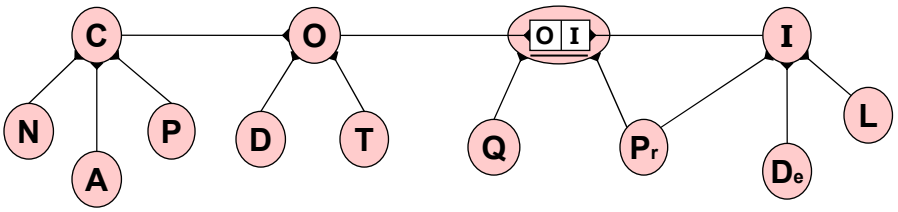
Transforming ERel to ORM

ORMvER
18

LEGEND:
Primary Key
Foreign Key
1 — M



ADD Foreign Keys (redundant). **NOTE: Duplicate fields - C Pr 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

ORMVER
19

Try it yourself, starting with =>

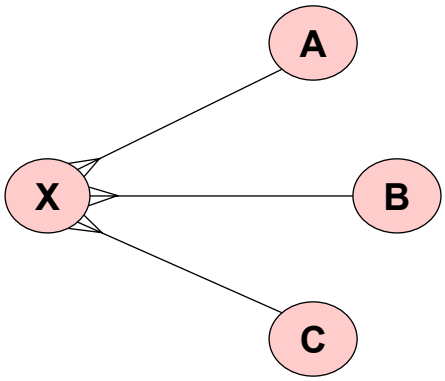
Modify / extend the diagram with these semantics:

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

Data Modeling Terminology

ORMINTRO
20

	O-R ("conceptual")	E-R ("logical")	COBOL/DBTG "NETWORK" ("physical" implementation)	RELATIONAL ("physical" implementation)
	OBJECT	ENTITY (TYPE)	RECORD TYPE	RELATION, or TABLE
		ATTRIBUTE	DATA ITEM (ELEMENT)	COLUMN, or FIELD
	FACT SENTENCE	INSTANCE	RECORD	ROW, or TUPLE
	Reference Mode	IDENTIFIER	IDENTIFIER	KEY
	PREDICATE	RELATIONSHIP	CODASYL "SET"	FOREIGN KEY
	CONSTRAINT			CONSTRAINT

ORM - Digging a little deeper

ORMODLG
21

THE PROCESS: Familiarize > Verbalize > Symbolize

- **Gather evidence from User domain**
  - **look at forms, reports, files**
  - **what users say; what users do**
- **Transform into Elementary Facts**
- **Discover Objects (nouns) and Relationships (verbs)**
- **Document findings in:**
  - **Elementary fact sentences**
  - **ORM Data Model Diagram**

Elementary Fact Sentences

ORMODLG
22

HALPIN01-p.61.

ENGLISH GRAMMAR - Structure of a Sentence:

SUBJECT + PREDICATE [+ OBJECT]

NOUN VERB NOUN

Elementary Fact -- cannot be decomposed into pieces which collectively provide the same information as the original fact.

	George	***runs.***		=> UNARY
	George	***runs***	**to the store.**	=> BINARY
	George	***likes***	**to run.**	
NOT:	**George**	***likes***	**to run <u>and</u> jump.**	
	George	***does <u>not</u> like to***	**run.** … CLOSED WORLD ASSUMPTION	
	George and Mary	***like***	**to run. … together!**	

<u>If</u> George *runs*, <u>then</u> Mary *runs*.

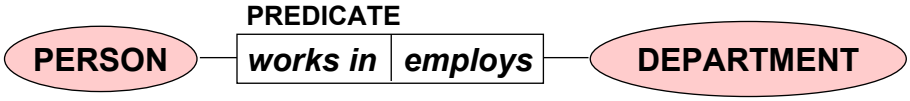
<u>All</u> people who *run* are happy! …HANDLED WITH ROLE SET CONSTRAINTS

Symbolize: ORM Constructs

ORMODLG
23

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

Elementary Binary Fact Sentence (schema):



Verbalization: (both ways)

"PERSON works in DEPARTMENT"
"DEPARTMENT employs PERSON"

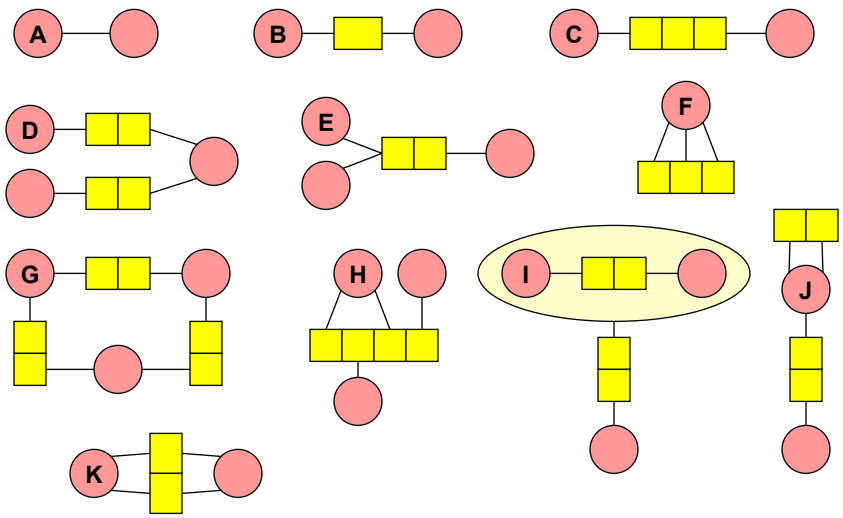
Exercise – ORM Diagrams

G

ORMODLG
24

HALPIN01-Ex.3.4-7, p.92.

Find the illegal ORM Diagrams; explain why:

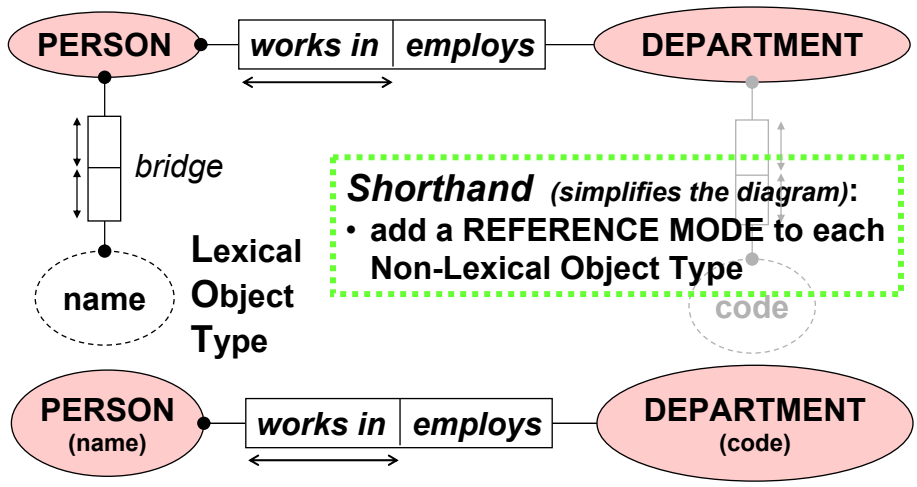


Object Reference Mode

ORMODLG
25

HALPIN01-p.79.

Non-Lexical Object Types (NOLOTS):



Now, every Object has a lexical handle.

Adding ORM Constraints

ORMODLG
26

- **Verbalization:** "PERSON *works in* DEPARTMENT"
  "DEPARTMENT *employs* PERSON"

- **DEPENDENCY (REQUIRED or MANDATORY):**
  "PERSON *must work[s] in some* DEPARTMENT"

EXCLUSIVITY (UNIQUENESS):
"PERSON *works in at most one* DEPARTMENT"

Uniqueness Constraint

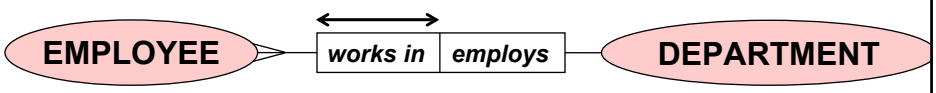
ORMCON
27

- to represent the exclusivity / multiplicity characteristic of a relationship (ER focuses on the Entities)
- ORM focuses on the Elementary Fact Sentence, so... construct a 'fact table' of representative instances:

EMPLOYEE *works in* DEPARTMENT

Peterson	2000
Lynn	2000
Carr	2100
Callagan	2100
Guttman	2110
:	:

- Put a line across the role(s) that make the predicate unique:



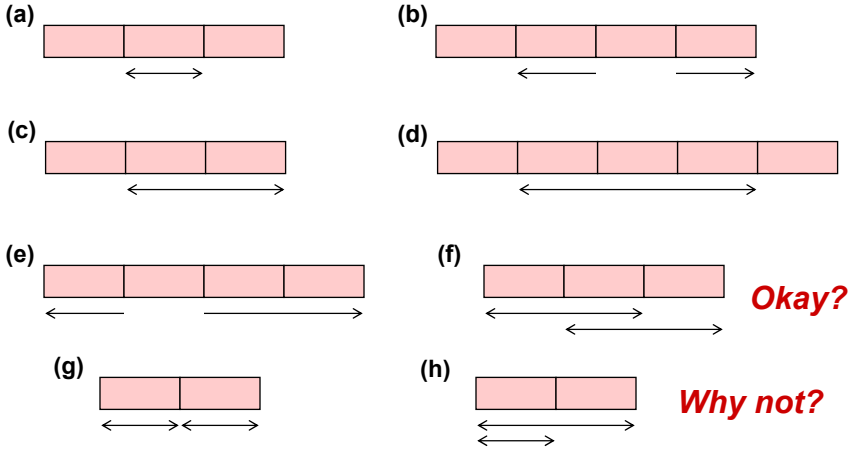
An EMPLOYEE can participate in the relationship with DEPARTMENT *at most once.*

Uniqueness Constraints - Exercises

G

ORMCON
28

- The keys for certain fact types are as shown. On this basis, which of these fact types are definitely splitable? – Halpin$_{01}$ §4.5, p.151



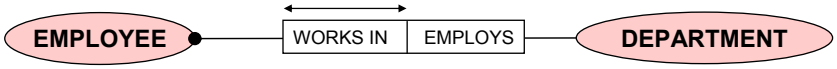
RULE:

Mandatory Role Constraint

ORMCON
29

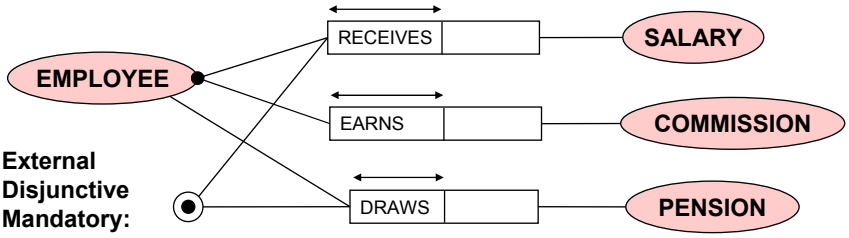
Halpin$_{01}$-§5.2.

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



– *every* EMPLOYEE *must* work for some DEPARTMENT.

- Disjunctive mandatory:

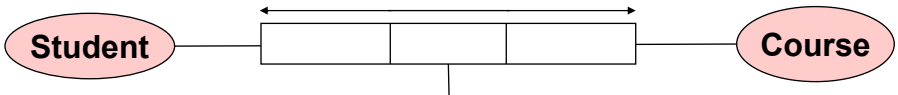


- External Disjunctive Mandatory:

Ternary++ Relationship
Nested 'Objectified' Predicate/Fact

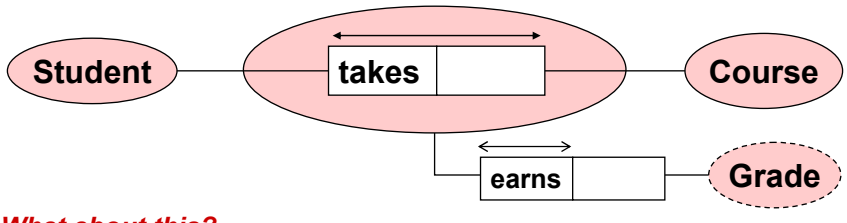
ORMODLG
30

HALPIN$_{01}$-p.85.



What is wrong with this?

Must have a Grade



What about this?

CONTRIBUTION:

DATE	AMOUNT	FUND

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

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

Value Set Constraint

ORMCON 31

Halpin$_{01}$-§6.3,p.220.

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) to contain all possible values of the Entity, *even if not used elsewhere in the database*.
  - Declare the Entity as '<u>Independent</u>' to allow for orphans, e.g., ...

Object

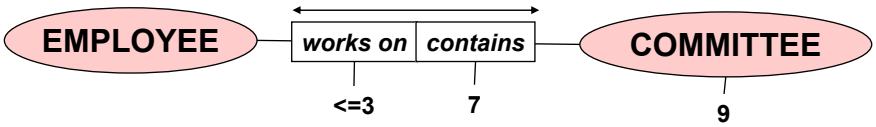
SKILL !

Frequency Constraints

ORMCON 32

Halpin$_{01}$-§7.2,p.278.

- **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 *VisioEA* thru Value Set)
  - Limits the size of an Object Population



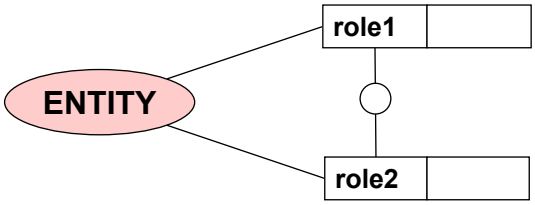
Not in Record-based modeling schemes!

Role Population Constraints

ORMCON 33

Halpin$_{01}$-§6.4,p.229.

- Applies at the type level, on whole populations
- Only consider if *both* roles are optional
- Only apply when roles are on the *same* entity type population.



May also apply to the tuple populations (both roles in predicate) or to ternary predicates.

EQUALITY (=)

R1 IFF R2

R1 ⟷ R2

pop(R1)=pop(R2)

SUBSET (⊆)

If R2, then R1.

R2 ⟶ R1

pop(R2)<=pop(R1)

EXCLUSION (X)

If R2, then -R1 and vv.

pop(R1)^pop(R2)=NULL

Not in Record-based modeling schemes!

Ring Constraints

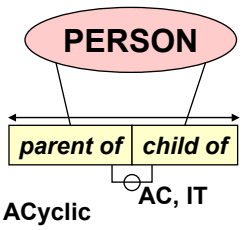
ORMCON 34

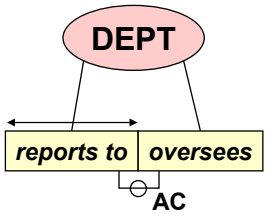
Halpin$_{01}$-§7.3,p.283.

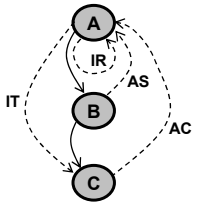
on a **Reflexive Relationship**
(Ring Fact Type, Homogeneous Predicate)

EXAMPLES:

- PERSON ----- *parent of* | *child of* ----- PERSON (M:M)
- PERSON ----- *mother of* | *child of* ----- PERSON (1:M)
- PERSON ----- *husband of* | *wife of* ----- PERSON (roles, NOW or EVER)
- DEPARTMENT ---- *reports to* | *oversees* ---- DEPARTMENT



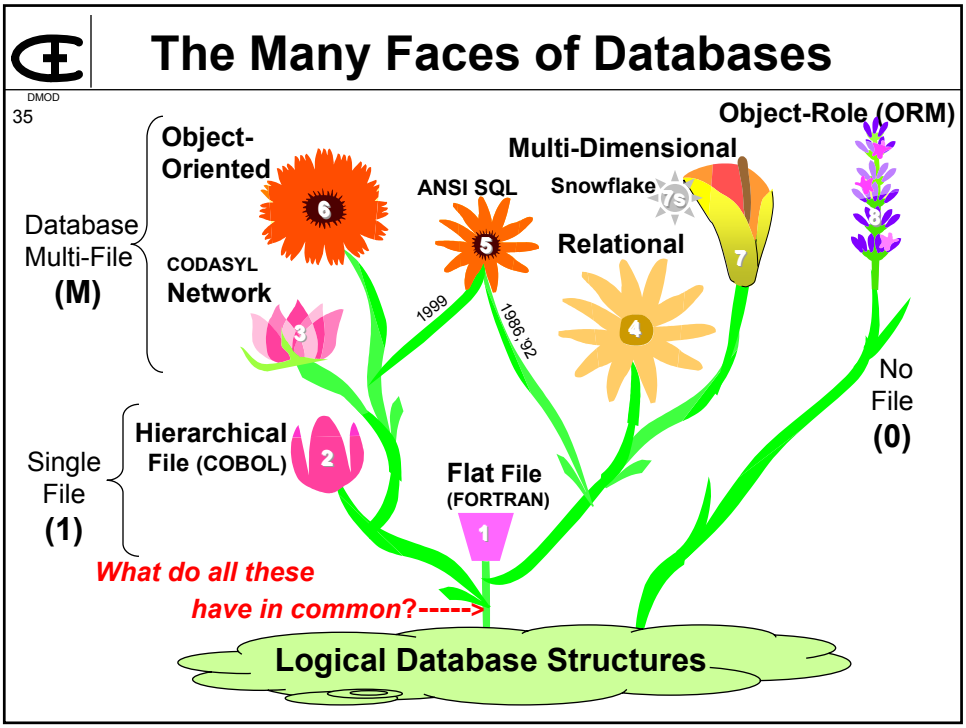


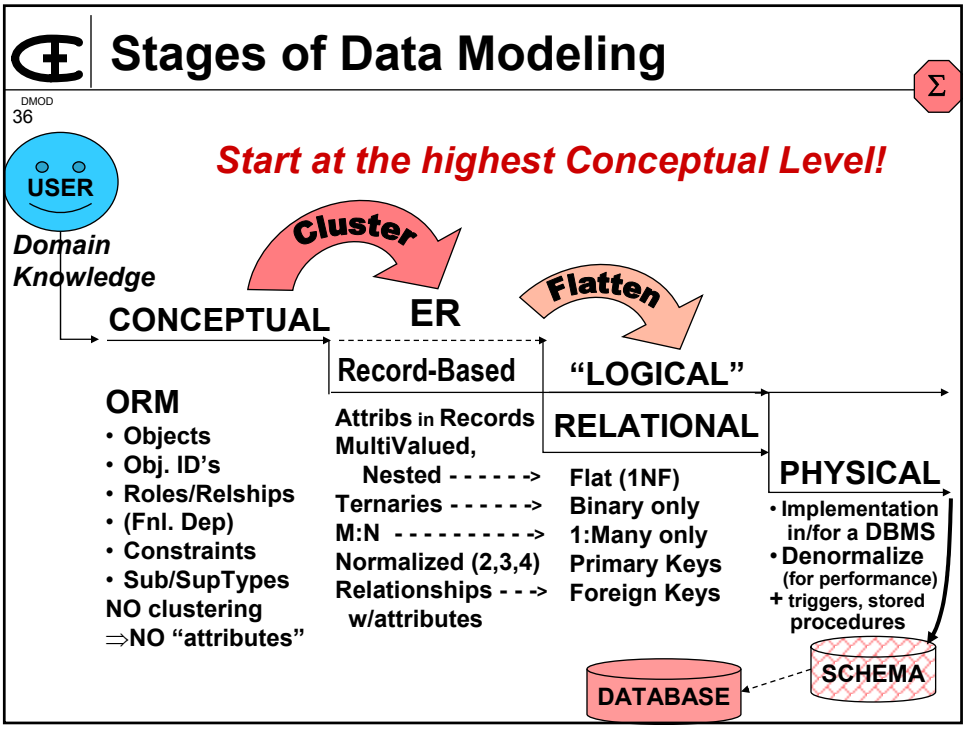


ACyclic
InTransitive
WHICH ALSO IMPLIES
ASymetric
IRreflexive

All possible relationship combinations are allowed unless Ring Constraints are declared.

Not in Record-based modeling schemes!





Problems with ER Modeling - Summary

ORMVER
37

All a consequence of clustering !

- **Cannot capture the "conceptual" view directly, Must *mentally* map to the "logical" (record-based) view by clustering Attributes into Entity records/tables.**
  - **Modeler must *a priori* choose whether Entity or Attribute**
  - **Too much clustering; attributes in the wrong place**
  - **Ignores (or presumes normalized) intra-record structure (that is, relationships between/among Attributes) - creates (implies) spurious inter-attribute relationships**
- **Human modeler is responsible for normalization remedy is *always* record decomposition**
- **Must choose unique names**
  - **for attributes in a record; for spurious new "entities"**
  - **column names = domain + role; lose object domains**
- **Modeling / Processing dilemma:**
  - **Complete representation of an entity object - more clustering**
  - **Full normalization (1NF) – decomposition, more fragmentation**
- **Indirect representation of M:N relationships**
- **Difficulty representing Ternary relationships**
- **Stability of the query language (SQL)**

All are solved in ORM !

Why NIAM/OR Modeling?

ORMINTRO
38

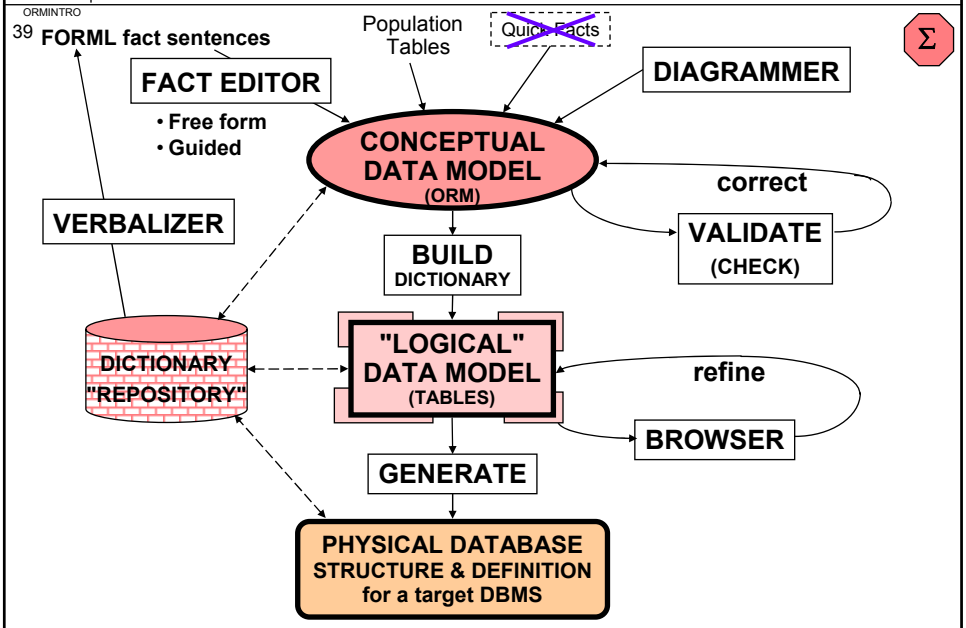
- **roots in both LOGIC & LINGUISTICS**
- **based on one modeling construct: the fact sentence**

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, Express, OMG-Busn Rules...)
- **now supported with a viable PC-based CASE tool**

VisioModeler / *VisioEA* Architecture

ORMINTRO
39



What's Done in ORM?

ORMINTRO
40

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, Value)**
- **Explicitly represents *all* Relationships**
  including all Functional Dependencies.

Resulting Benefits of ORM

ORMINTRO
41

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
(ER, record-based, high-level "logical" data modeling)

Why Not NIAM?

ORMINTRO
42

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

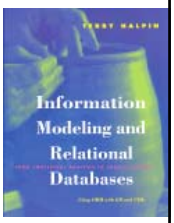
- **Few academics involved**
- **CDC kept it proprietary - in Europe and USA**
- **Lack of strong vendor support**
- **Too detailed; bottom up design - NOT!**
- **Lack of support for mental chunking and Levels of Abstraction - NOT!**
  - **ENTITIES, ATTRIBUTES, VALUES (LOTS), RELATIONSHIPS are all represented the same way. Diagrams should give visual prominence in proportion to semantic importance.**

Resources on ORM

ORMINTRO
43

BOOK:

- **Terry Halpin** (UQ -> VISIO -> Microsoft -> Northface -> Neumont Univ.), ***Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design***, Morgan Kaufmann Publishers, San Francisco, 2001, 763 pgs



Database DESIGN TOOL(s):

- ***VisioEA***, in Visual Studio.net Enterprise Architect ed.
- 'NORMA' - ORM2 tool is under development
  - improved notation

WEB SITES:

- www.orm.net = Halpin's web site
- Journal of Conceptual Modeling, www.inconcept.com/jcm

COURSES:

- IDSc 4431 (for U of MN CSOM Undergrads)
- INet 4131 (U of MN College of Continuing Education)
  - Online version is under development