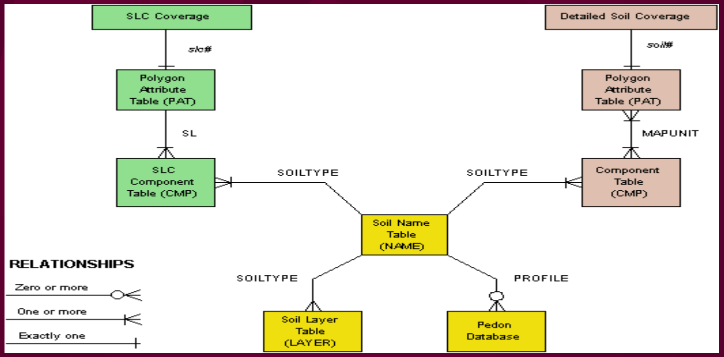
Entity-Relationship Model

Database Modeling



Database Modeling

- A small business may have upwards of 20 tables in its data model, a medium size business may have upwards of 40 tables, and a large business may have over 100 tables
- There are many ways to arrange data attributes into tables – some good some not so good
- We need an efficient and easily communicated way to design databases and get the designs reviewed and approved

Database Modeling (con't)

- Normalization (discussed in a later lesson) is a quantitative modeling process to design the relational tables needed but is slow and cumbersome for large databases
- Thus graphical methods are widely used
- Normalization is still an important consideration, but is typically used after the design diagram is translated into relational tables

E-R Model

- Developed by Peter Chen (LSU) in 1976
- Extended by Chen and many others over the years - models, terminology, notations vary
- Is a general entity level model, and can be translated into specific database types such as relational databases

Entities

- An entity is something that exists and can be identified; it is significant to the scope of the problem being analyzed
- An entity may be physical, conceptual, or abstract

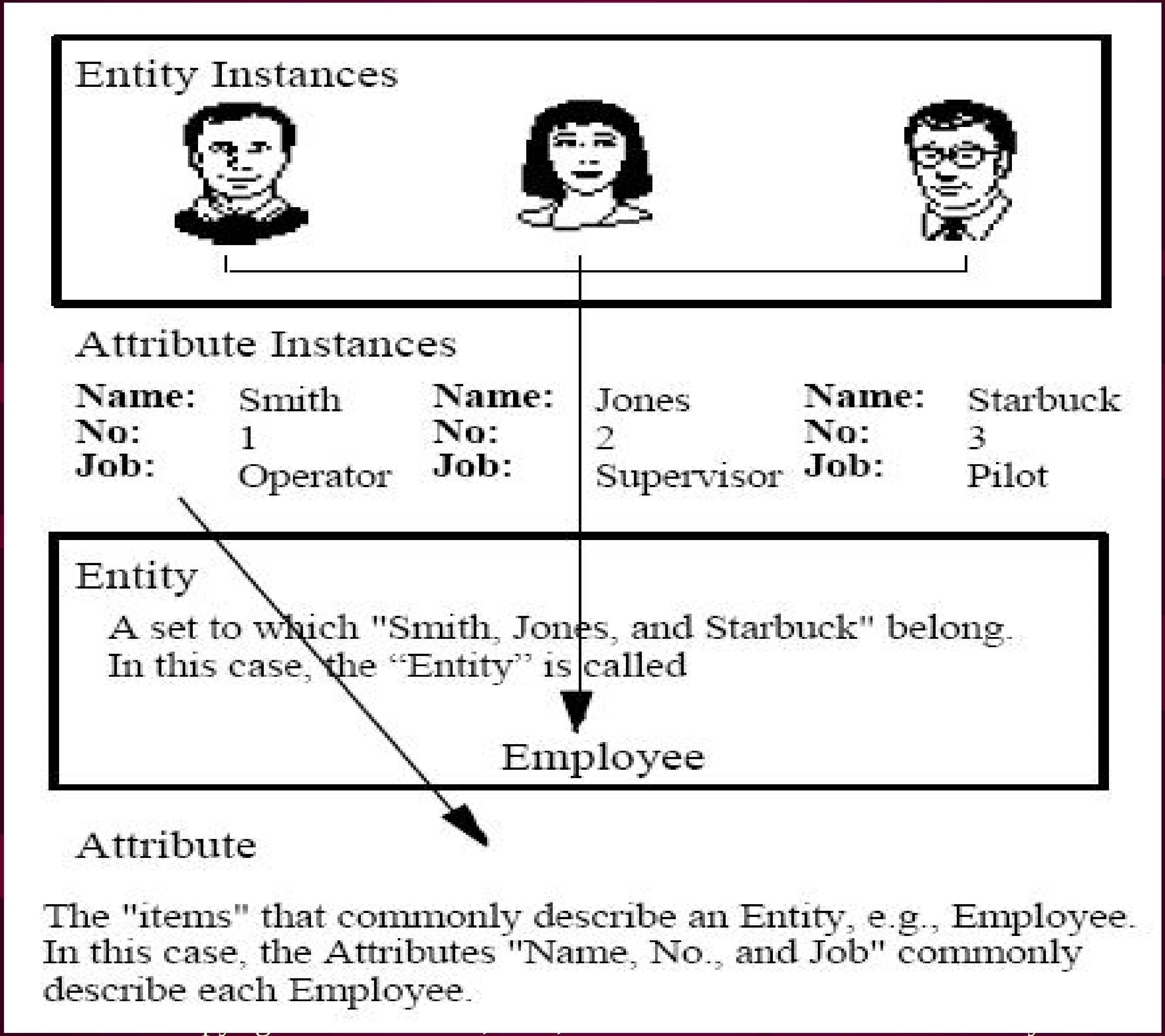
Entities (con't)

- Entities (Objects) are grouped into Entity Classes (" Types of Things")
- Represented textually by capital letters
- Terms entity class and entity used somewhat interchangeably in common database jargon

Attributes

- Entities have attributes, or sometimes called properties
- Employee attributes:
  - EmployeeName
  - DateOfHire
  - Department
- “Camel” notation is common
- All instances (objects) of an Entity Class have the same attributes – but different values of the attributes

Employee Entity Class



Entity Instances

Attribute Instances

Name:	Smith	**Name:**	Jones	**Name:**	Starbuck
No:	1	**No:**	2	**No:**	3
Job:	Operator	**Job:**	Supervisor	**Job:**	Pilot

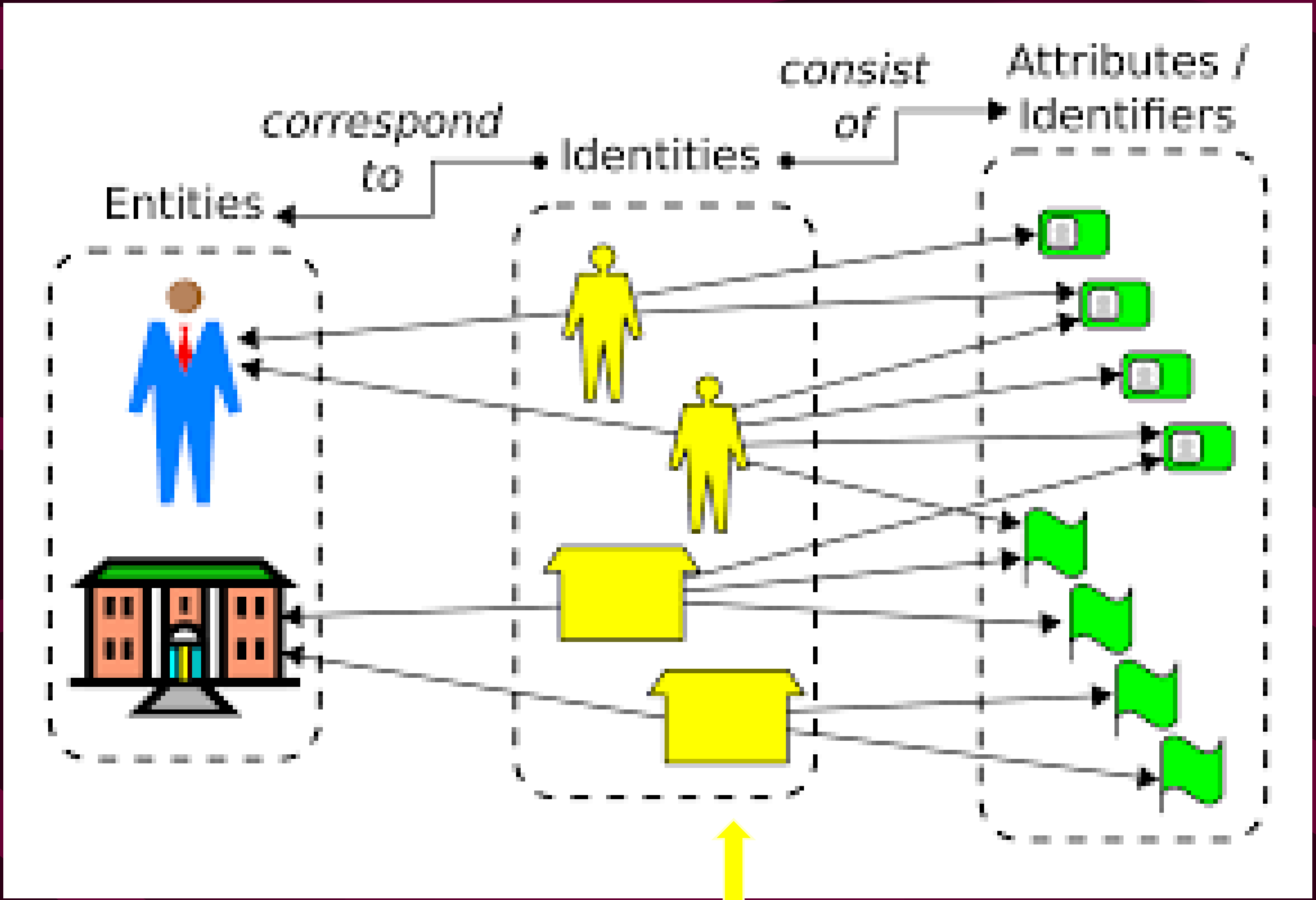
Entity

A set to which "Smith, Jones, and Starbuck" belong.
In this case, the "Entity" is called

Employee

Attribute

The "items" that commonly describe an Entity, e.g., Employee.
In this case, the Attributes "Name, No., and Job" commonly describe each Employee.


consist
of
Attributes /
Identifiers
correspond
to
Identities
Entities

Instances

Attribute Types

- Simple (ie weight)
- Composite (aggregate)
  - address
    - Street
    - City
    - State
    - Zip
- Multi valued (repeating groups)
  - person’s autos
- Not all E-R methods (or tools) allow other than simple types

Domain (allowable values of attributes) Types

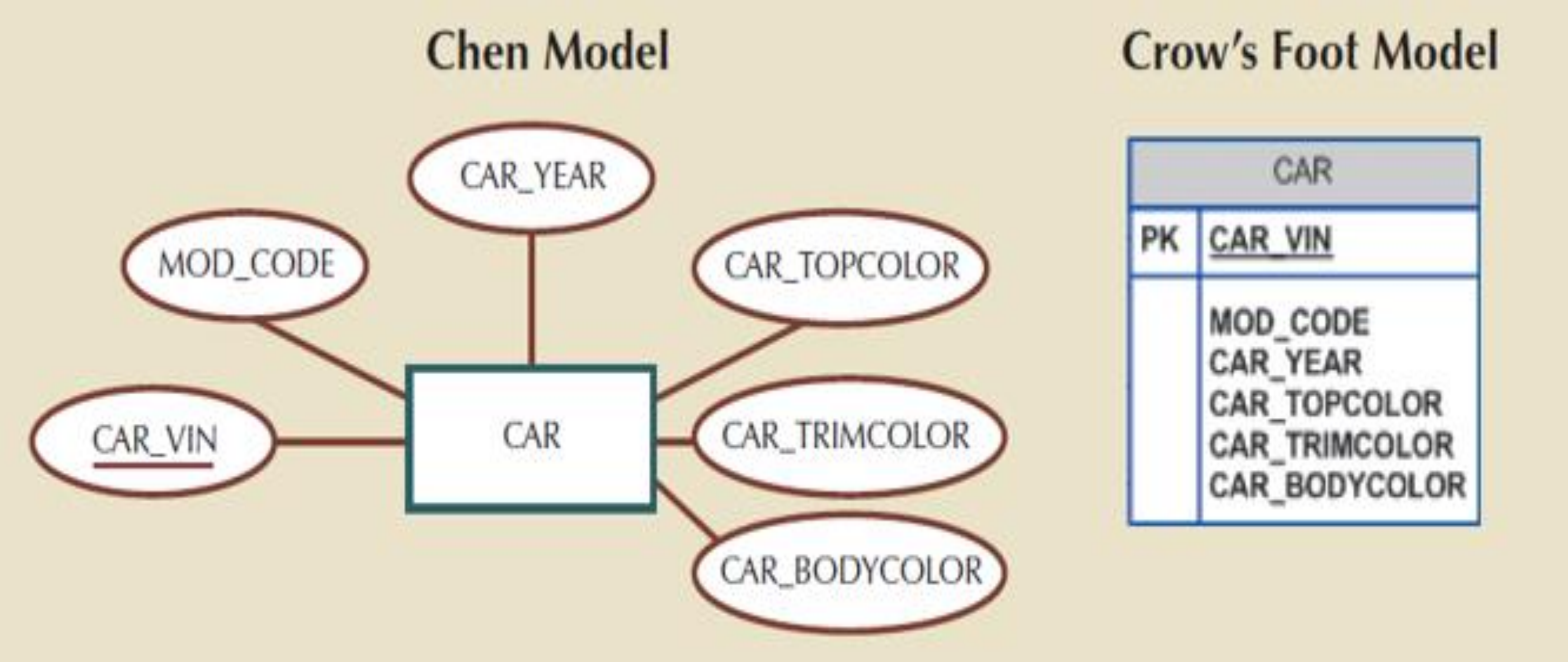
- Continuous
  - weight [141.45]
  - address
- Discrete
  - year
- Enumerated (few discreet values)
  - eye color (Brown, Blue, Gray, Green)
  - State (AL, AR, ...)
  - Sex (M,F)
- In GUI's, types correspond to different types of controls used on forms (single line edits [text box], check or radio box, drop down, slider, spinner, ...

What type of GUI controls with which domain types ?

Identifiers

- Entities (or Entity Class instances) have names or numbers that identify them
- Examples:
  - Auto: vehicle ID number
  - Person: social security number
  - Company: federal ID number
  - Customer: customer ID number
- Identifiers are one (or more) of the attributes of the Entity
- Identifiers may be unique or non-unique

Chen's Original Notation for Attributes



	CAR
PK	CAR_VIN
	MOD_CODE CAR_YEAR CAR_TOPCOLOR CAR_TRIMCOLOR CAR_BODYCOLOR

Unique Identifier underlined.

Attribute Naming Conventions

- Attribute names:
  - Should be unique within the entity
  - Should use the entity abbreviation as a prefix if used in other entities
  - Should be descriptive of the characteristic
  - Identifiers should use suffixes such as _ID, _NUM, or _CODE (such as Employee_ID)
  - Should not be a reserved word
  - Should not contain spaces or special characters such as @, !, or &

Customer Entity Class

[all entities in this class have these attributes]

- CustNumber (typically a unique identifier)
- CustName
- Address
- City
- State
- Zip
- Phone [composite & multi-valued]
  - Description
  - AreaCode
  - LocalNumber

Instance 1 (Entity 1)

- 12345
- A1 Ford
- 123 Union
- Memphis
- TN
- 38112
- Phone

Voice	Fax	Pager
901	901	901
365-1232	365-4321	323-6543

Instance 2

- 45678
- Bill’s Garage
- 321 Poplar
- Memphis
- TN
- 38118
- Phone
  - Voice1 Voice2
  - 901 902
  - 345-9876 345-9877

Relationships

- Relationship Classes are associations among Entity Classes
- Relationship instances are associations among entity instances
- Normally active or passive verbs that clearly indicate the nature of the relationship
- A relationship class may involve many entity classes, the number of which is the "degree" of the relationship
- Many models only have relationships of degree 2, called "binary" relationships

Example Degree 2 Relationships

- Two entity classes
- PERSON --- DOG
  - A person owns dog(s)

- SALESPERSON --- ORDER
  - A salesperson has placed certain orders

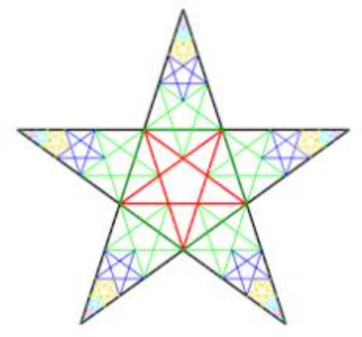
Example Degree 3 Relationships

- Three entity types
- AUTO DEALER --- MANUFACTURER --- TYPE
  - A dealer stocks certain types of vehicles (cars, trucks, vans) from certain manufacturers

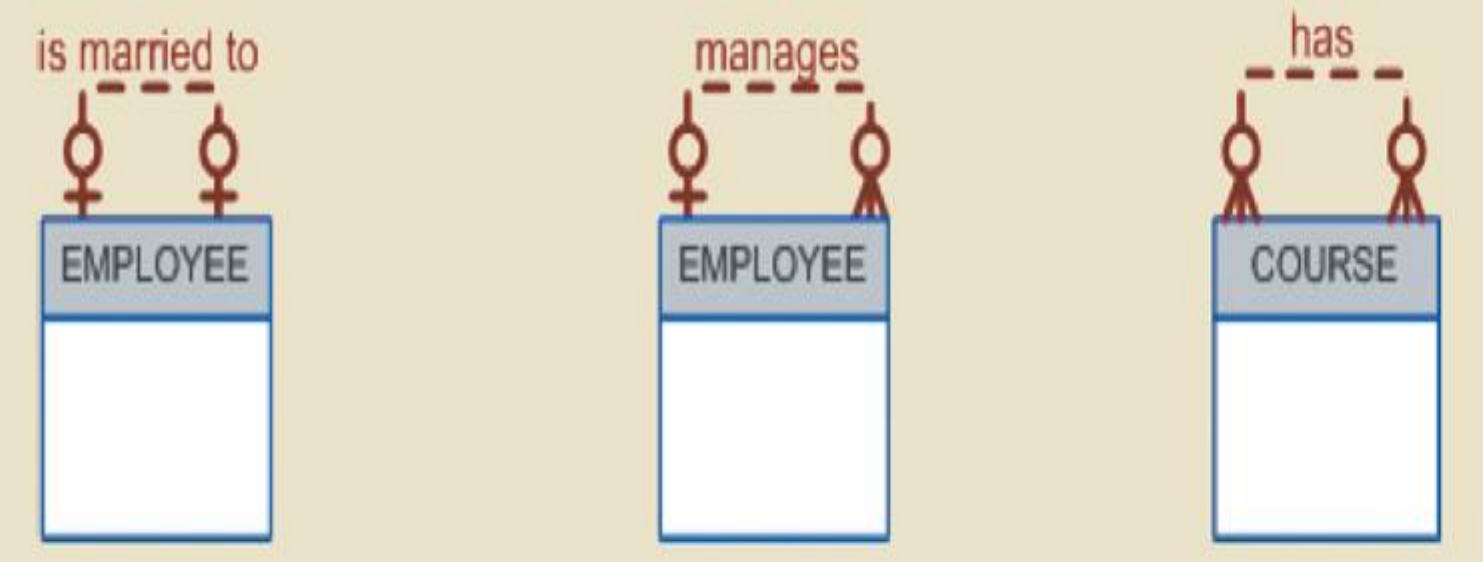
EMPLOYEE	SHIFT	JOB
Bob	1st Shift	Fork Lift
Tom	1st Shift	Load Truck
Carol	2nd Shift	Packing
Bob	2ndShift	Checking

Degree 1 - Recursive (Unary) Relationship

- Association among <u>same entity classes</u>:
  - EMPLOYEE ----- MANAGER
  - G/L ACCOUNT ---- MASTER ACCOUNT
  - PART --- ASSEMBLY

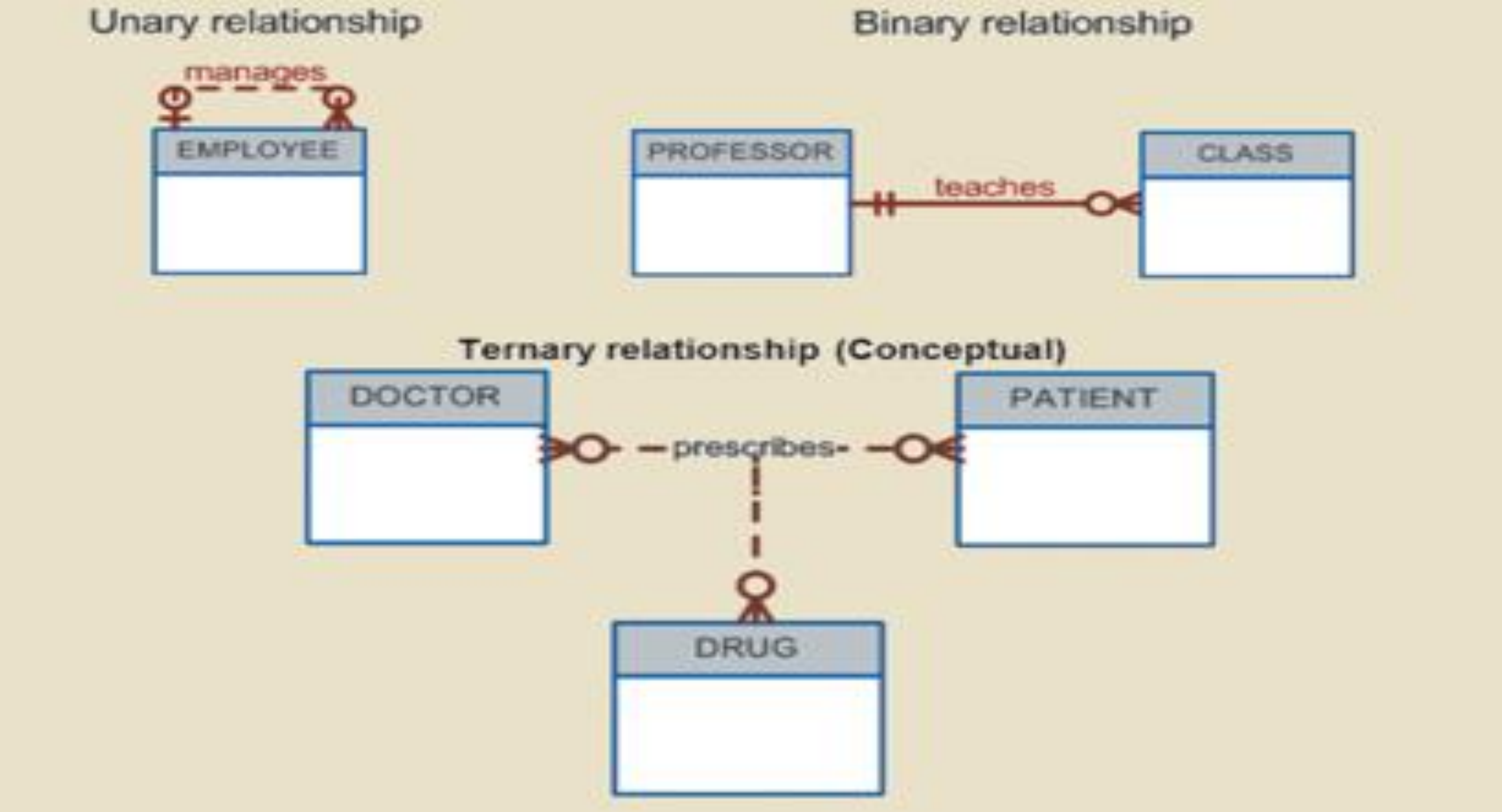


Common Recursive Relationships



Prerequisites

Relationship Degree



Cardinality [cardinality constraints]

- A relationships is a two directional mapping
- Each direction can have constraints on the number of associated entities
- Maximum Cardinality - the maximum number of associated entities
  - Typically one or many
- Minimum Cardinality - the minimum number of associated entities
  - Typically optional (0) or mandatory (1)

SALESPERSON - ORDER Relationship

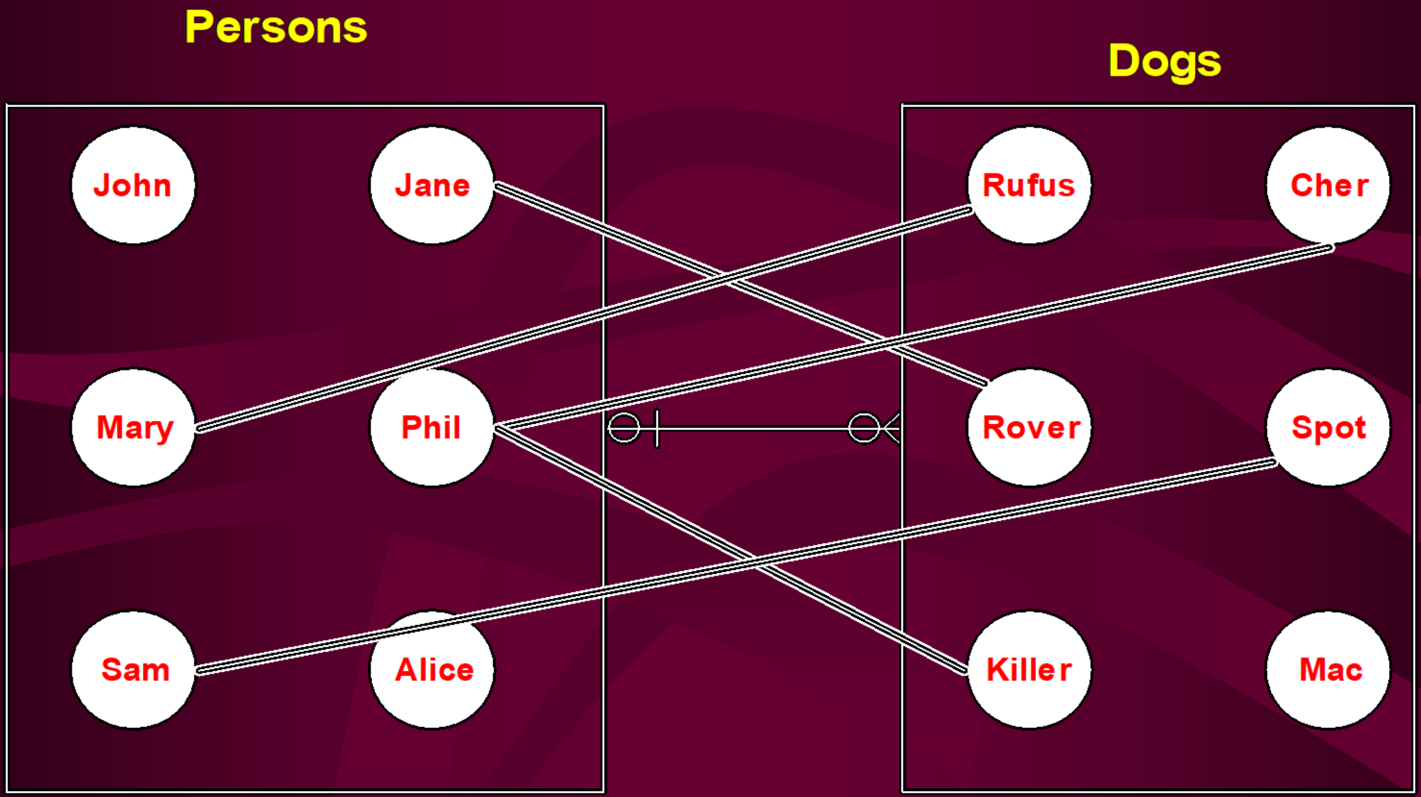
- 1 - Each salesperson can have none or many orders
  - maximum cardinality: N (many)
  - minimum cardinality: 0 (optional)
- 2 - Each order belongs to one and only one salesperson
  - maximum cardinality:1
  - minimum cardinality: 1 (required)

PERSON - DOG Relationship

- Each person can own none or many dogs
  - maximum cardinality: N
  - minimum cardinality: 0 (optional)
- Each dog is owned by none or one person
  - maximum cardinality:1
  - minimum cardinality: 0 (optional)

Person - Dog Relationship

[overall relationship depends upon “business rules”]



What are the entity classes ?

What are the entities (instances) ?

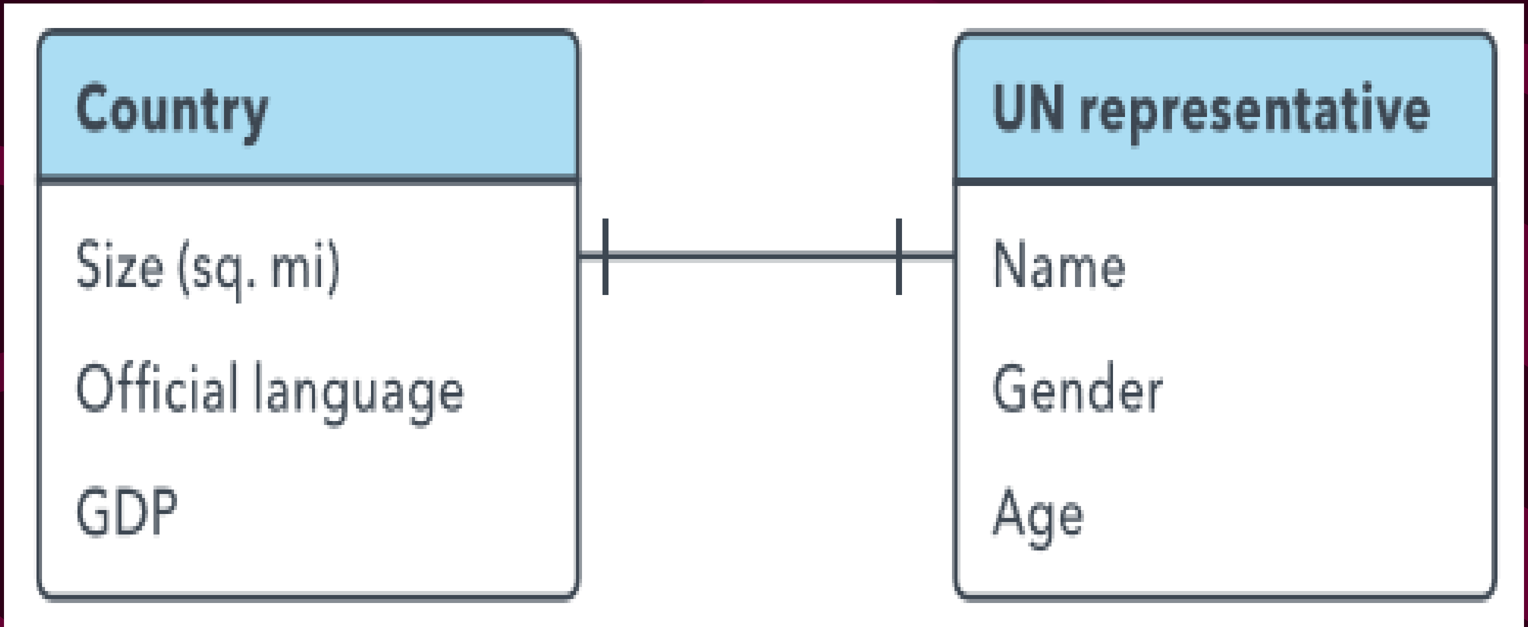
What is the relationship (or relationship class) ?

What are the relationship instances ?

Types of Binary Relationships

- Classified according to maximum cardinality
- Often called “has a” relationships
- One to One (PERSON to COMPUTER)
- One to Many (SALESPERSON to ORDER)
- Many to Many (ORDER to PRODUCT)

One to One Relationship



Country

Size (sq. mi)

Official language

GDP

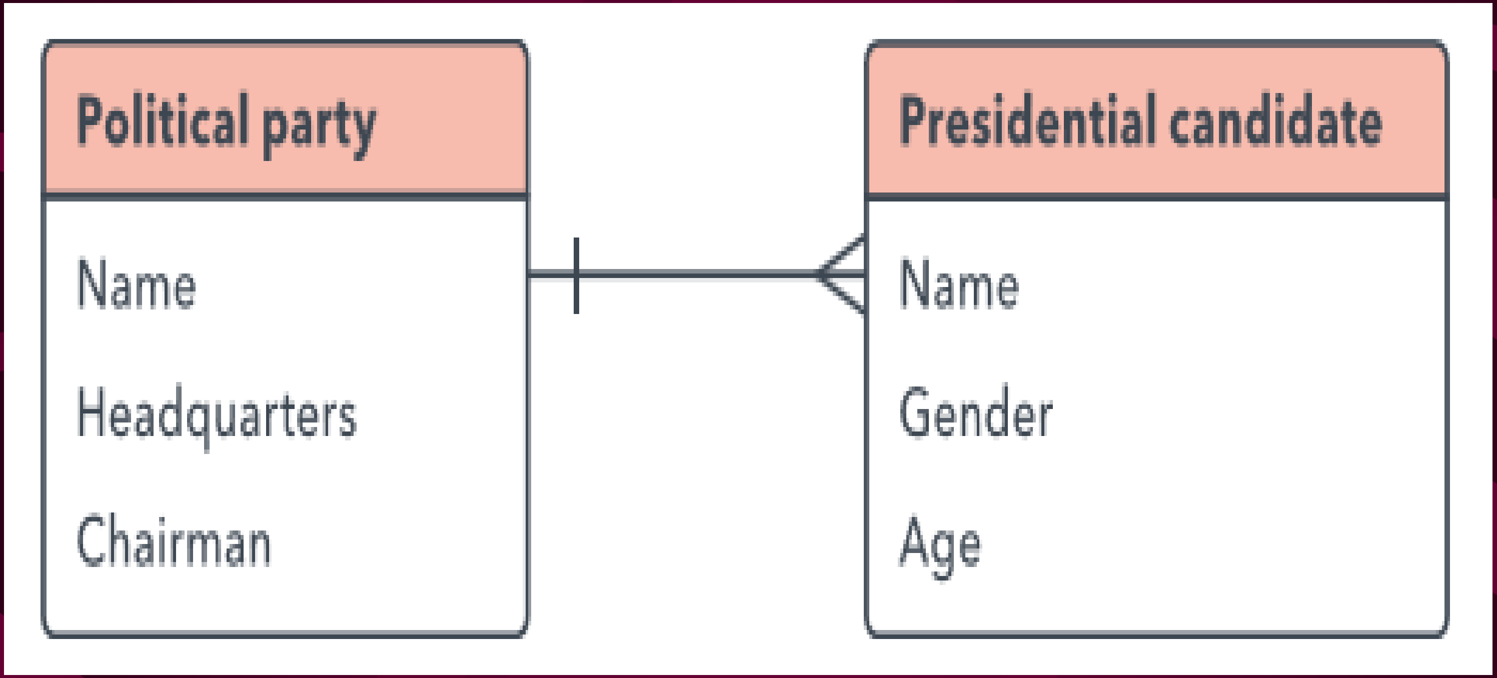
UN representative

Name

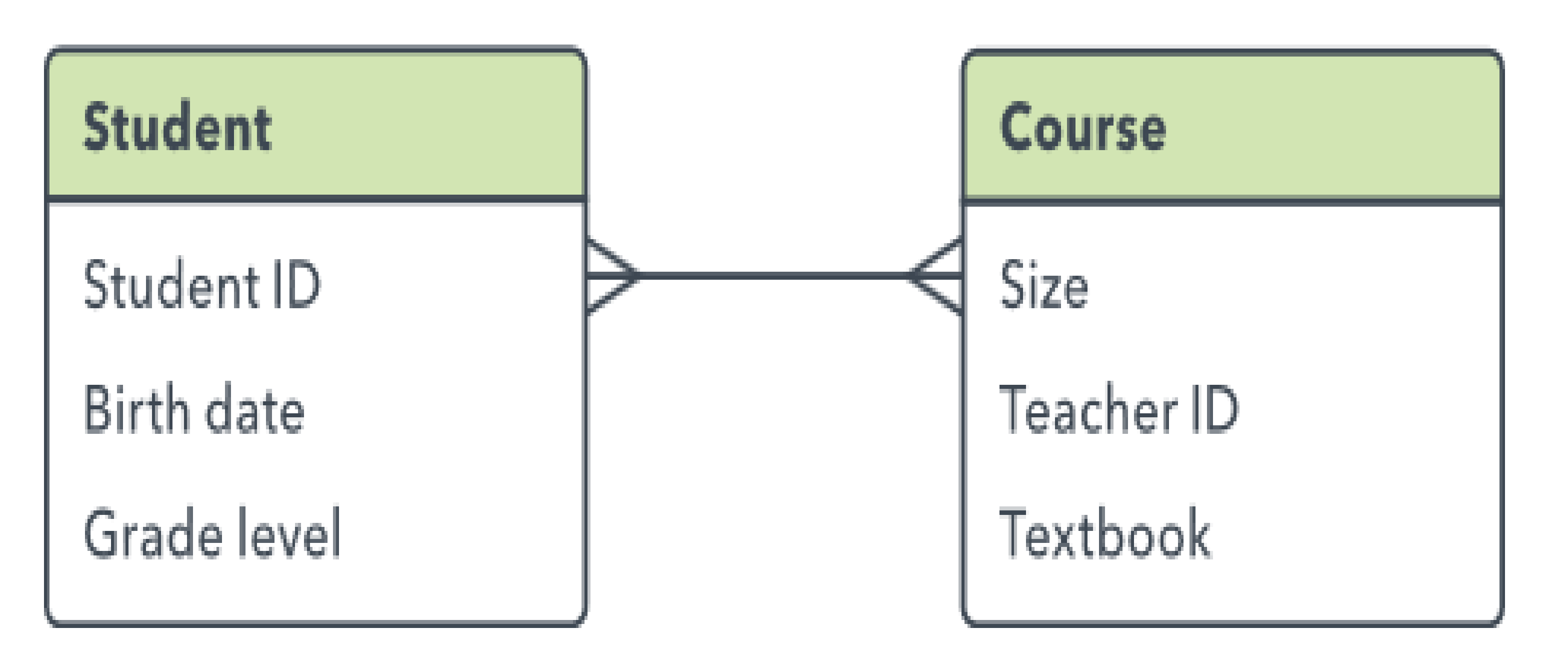
Gender

Age

One to Many Relationship



Many to Many Relationship



Student

Student ID

Birth date

Grade level

Course

Size

Teacher ID

Textbook

Why Classify Degree and Types of Relationships?

Don’t look ahead !

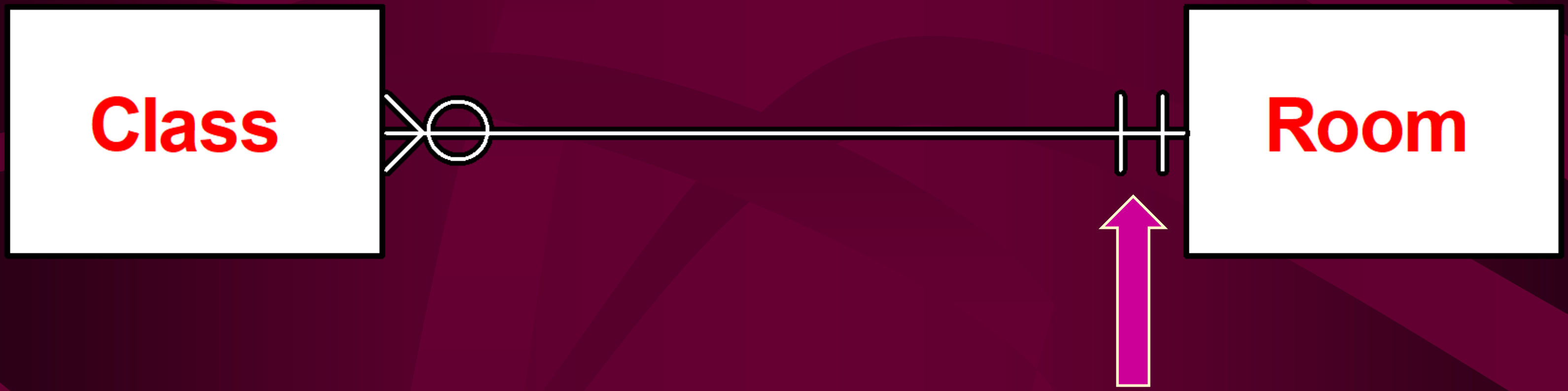
Why Classify Degree and Types of Relationships?

- Different degrees and types of relationships have different rules to translate the E-R diagram into relational tables!

E-R Design Conventions & Notation

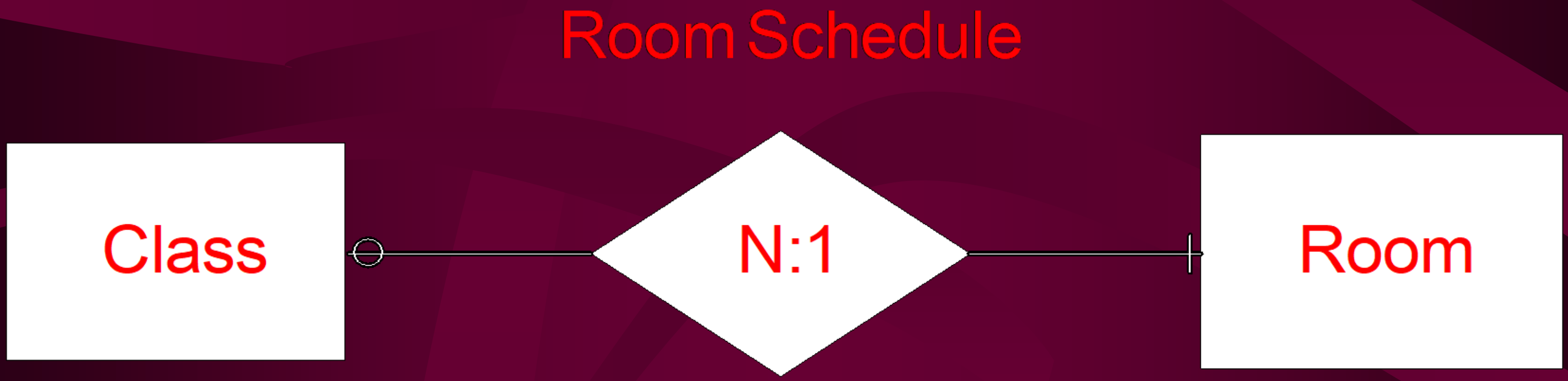
- **Symbolic** - shows relationships in symbols with cardinality integers
- **Crow's Foot (or IE, Information Engineering)** - shows relationships and cardinality with lines – most common in database work
- **IDEF** -Integrated Definition (US Std)
  - Mostly incorporated into Extended ER Method
- **UML** - Unified Modeling Language [International Standard]

Crow's Feet



Often a min and max of one is shown as one bar.

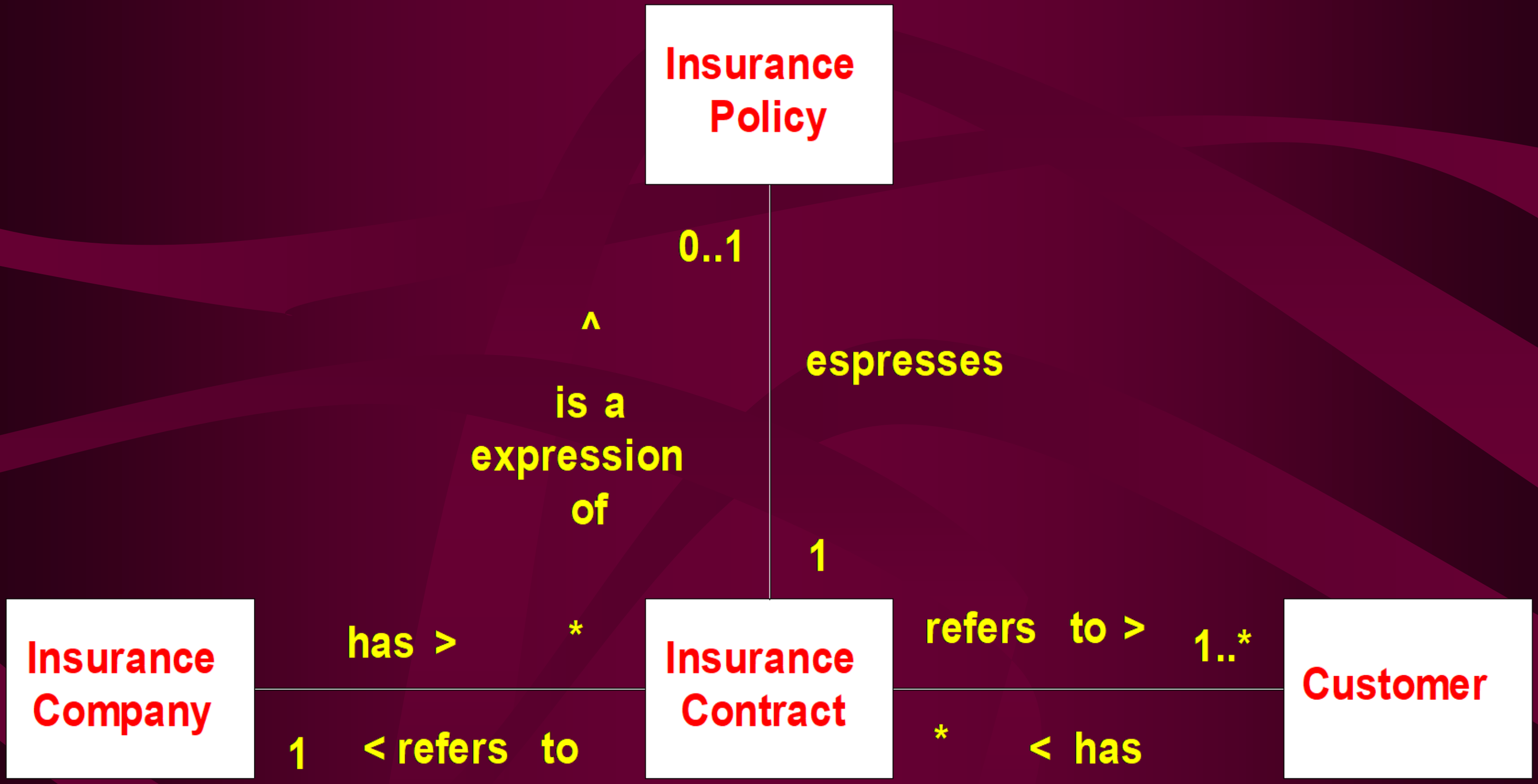
Symbolic Representation [max cardinalities in diamond]



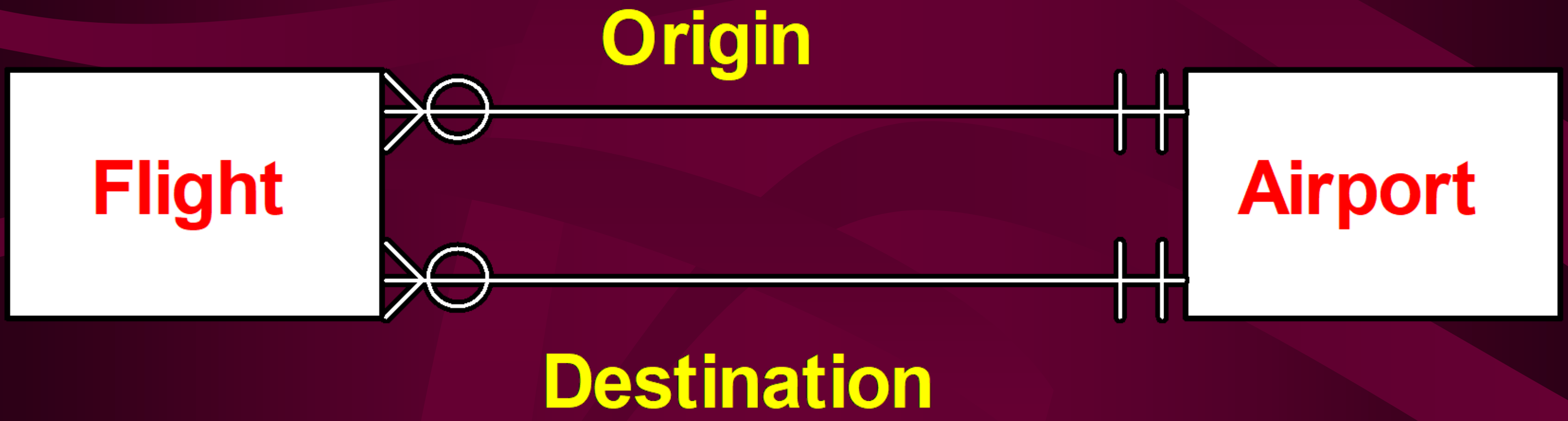
UML

- Cardinalities:
  - – Format: min..max (or enumeration list)
  - – min (typically 0 or 1)
  - – max (typically 1 or *)
  - – 0..* can be shortened to *
  - – 1..1 can be shortened to 1

UML


Insurance Policy
0..1
^
is a expression of
espresses
1
Insurance Company
has >
*
1
< refers to
Insurance Contract
refers to >
1..*
*
< has
Customer

Multiple Relationships Between Entities

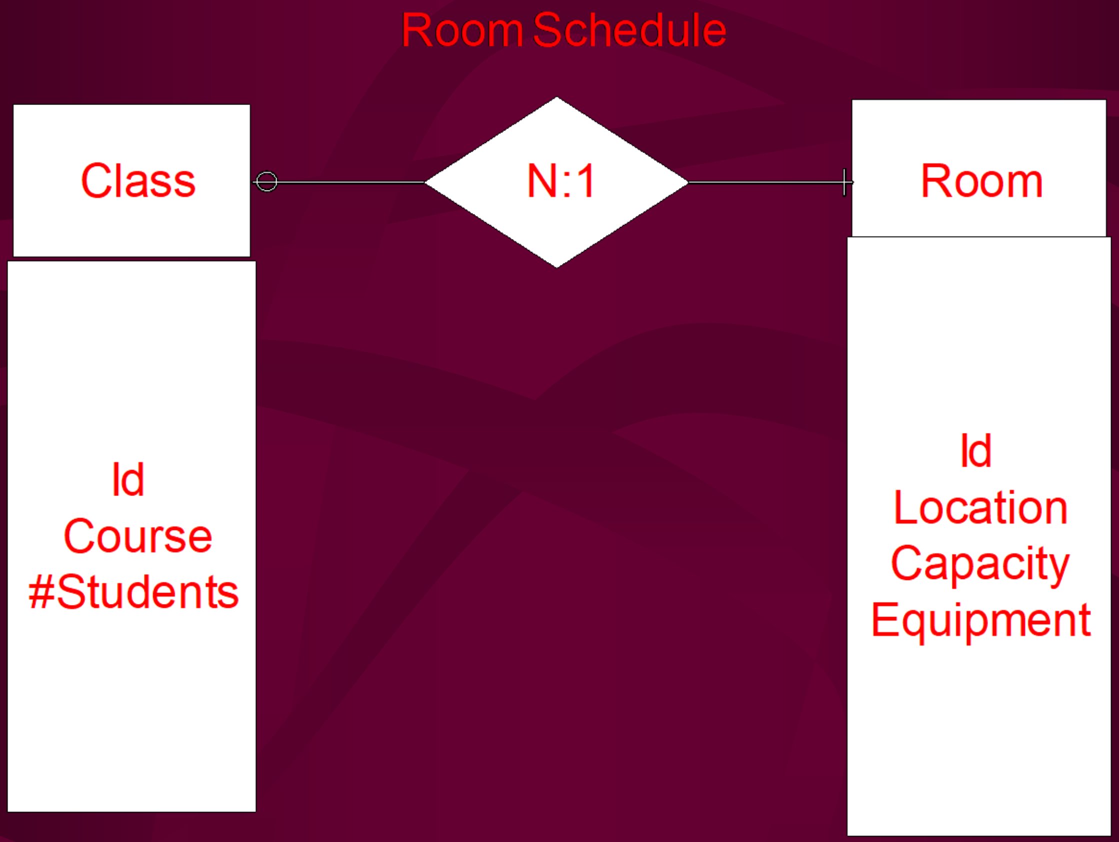


Copyright Dan Brandon, PhD,PMP   Christian Brothers University

Attributes in E-R Diagrams

- Represented as symbols connected to entity
- or
- Represented as text next to Entity symbol
- or
- Represented as text within Entity symbol
- or
- Optional pop-up windows

E-R Diagrams with Attributes



Room Schedule

Class

N:1

Room

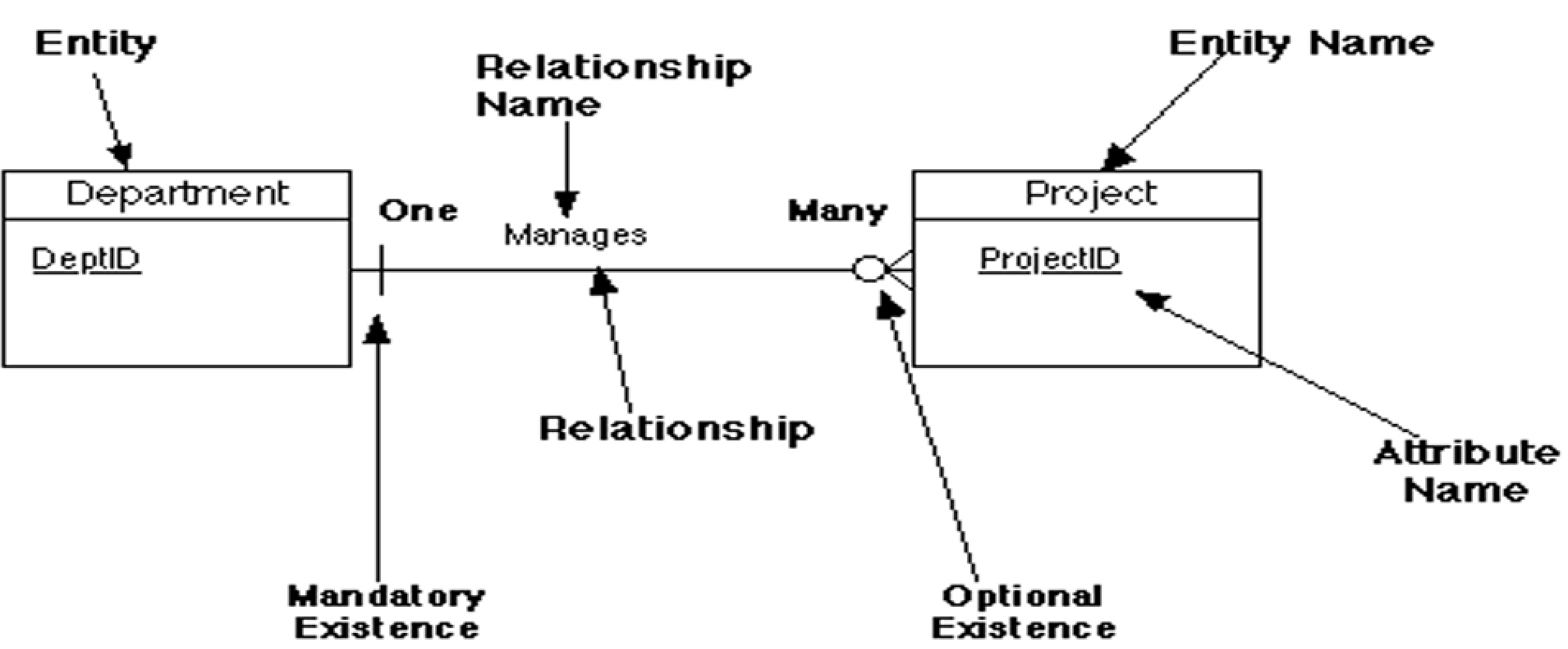
Id
Course
#Students

Id
Location
Capacity
Equipment

Entity - Level of Detail

- On an E-R diagram, there is typically a choice of 3 levels of detail:
  - – Entity Name only
  - – Name and unique identifier (underlined)
  - – Name, unique identifier, and other attributes

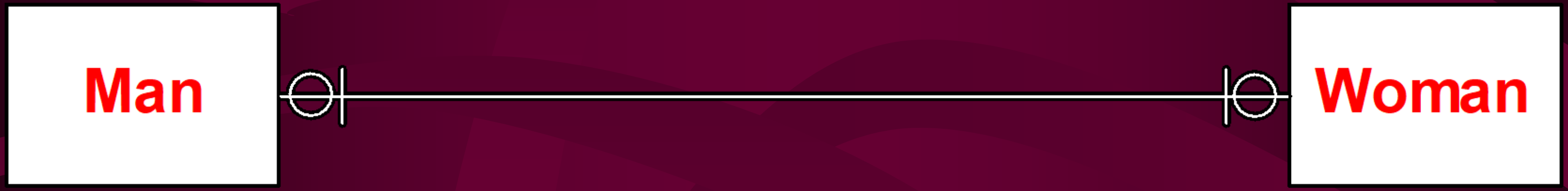
Simple E-R Drawing with Notation



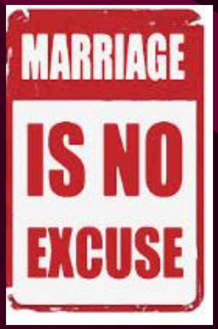
Relationship Attributes

- Relationship Classes can also have attributes
- The “Current Marriage” relationship between MAN and WOMAN may have an attribute of “rating”
- That attribute concerns each marriage instance, not each man or woman

Relationship: Current Marriage



Attribute :rating



- What is the attribute on the many-to-many relationship between ORDER and PRODUCT ?

Don’t look ahead !

- The relationship between ORDER and PRODUCT has attributes:
  - quantity ordered

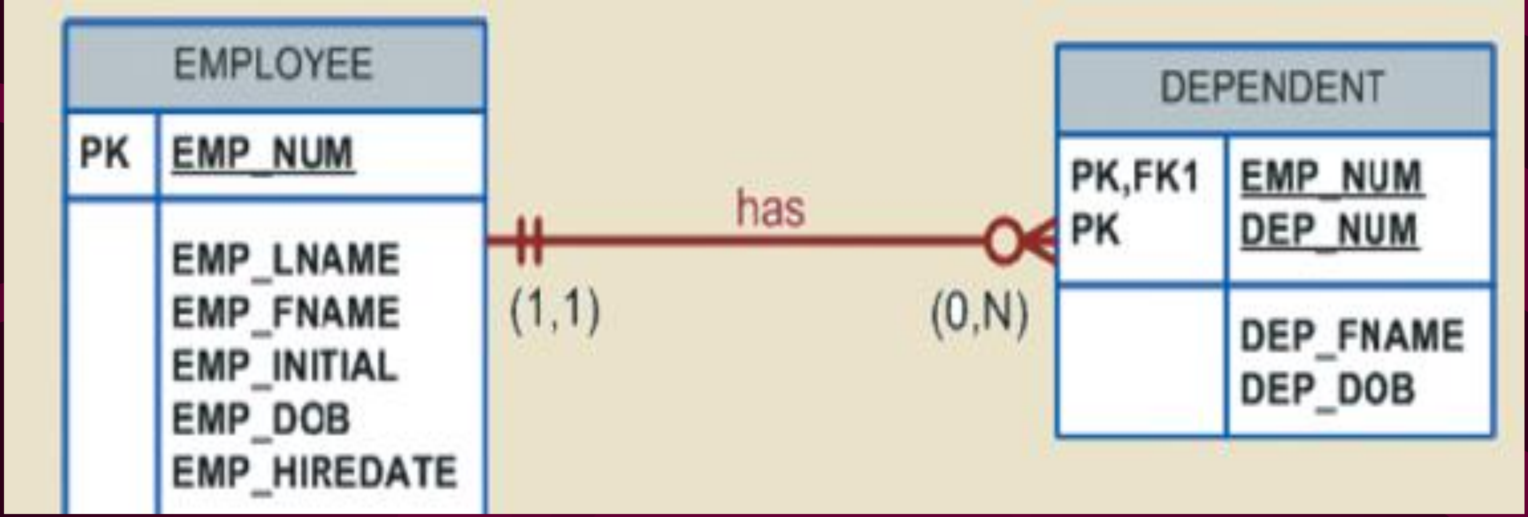
Weak Entities

- Those entities whose presence depends upon another entity
- Example: In a pharmacy database, the PRESCRIPTION entity depends on the PATIENT entity (if a patient is deleted, all their prescriptions need to be deleted)
- Round corners of entity symbol for weak entity
- May also round corners of relationship diamond [or use a different type of line] to show which entity weak entity is dependent upon (if more than one relationship exist with weak entity) - Weak entity will typically have a “one and only one” cardinality with entity that it is dependent upon

ID - Dependent Weak Entities

- ID-dependent entities - weak entities who are also dependent for their unique identifier
- Example:
  - Product & Product/Version

ID Dependent Weak entity



EMPLOYEE	
PK	EMP_NUM
	EMP_LNAME EMP_FNAME EMP_INITIAL EMP_DOB EMP_HIREDATE

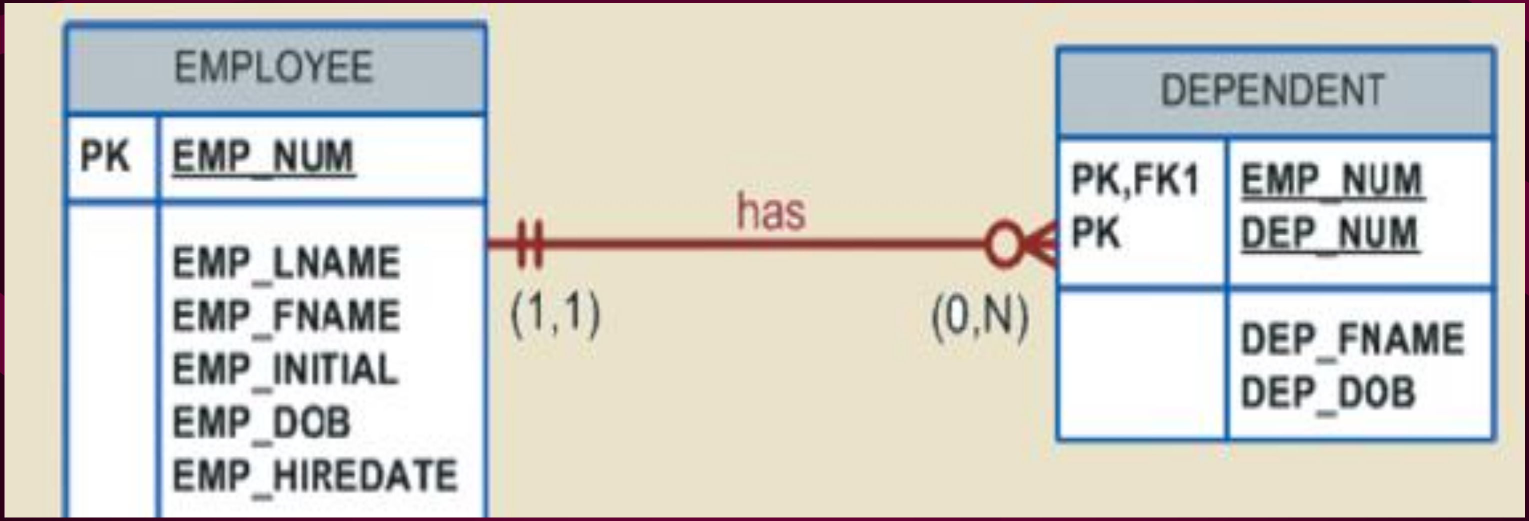
DEPENDENT	
PK,FK1 PK	EMP_NUM DEP_NUM
	DEP_FNAME DEP_DOB

Any problems with this model ?

Don’t look ahead !

Possible problems…

- Best not to assume that the LNAME of dependent is the same as employee
- What if two employee’s have the same dependents ?
- If the employee leaves the company, we may still have to keep dependents in the database for continued insurance obligations

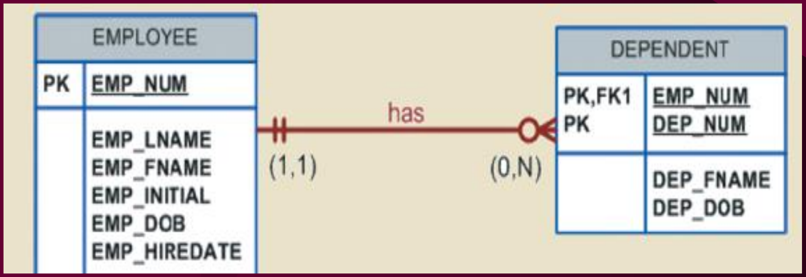


EMPLOYEE	
PK	EMP_NUM
	EMP_LNAME EMP_FNAME EMP_INITIAL EMP_DOB EMP_HIREDATE

DEPENDENT	
PK,FK1 PK	EMP_NUM DEP_NUM
	DEP_FNAME DEP_DOB

Relationship Strength

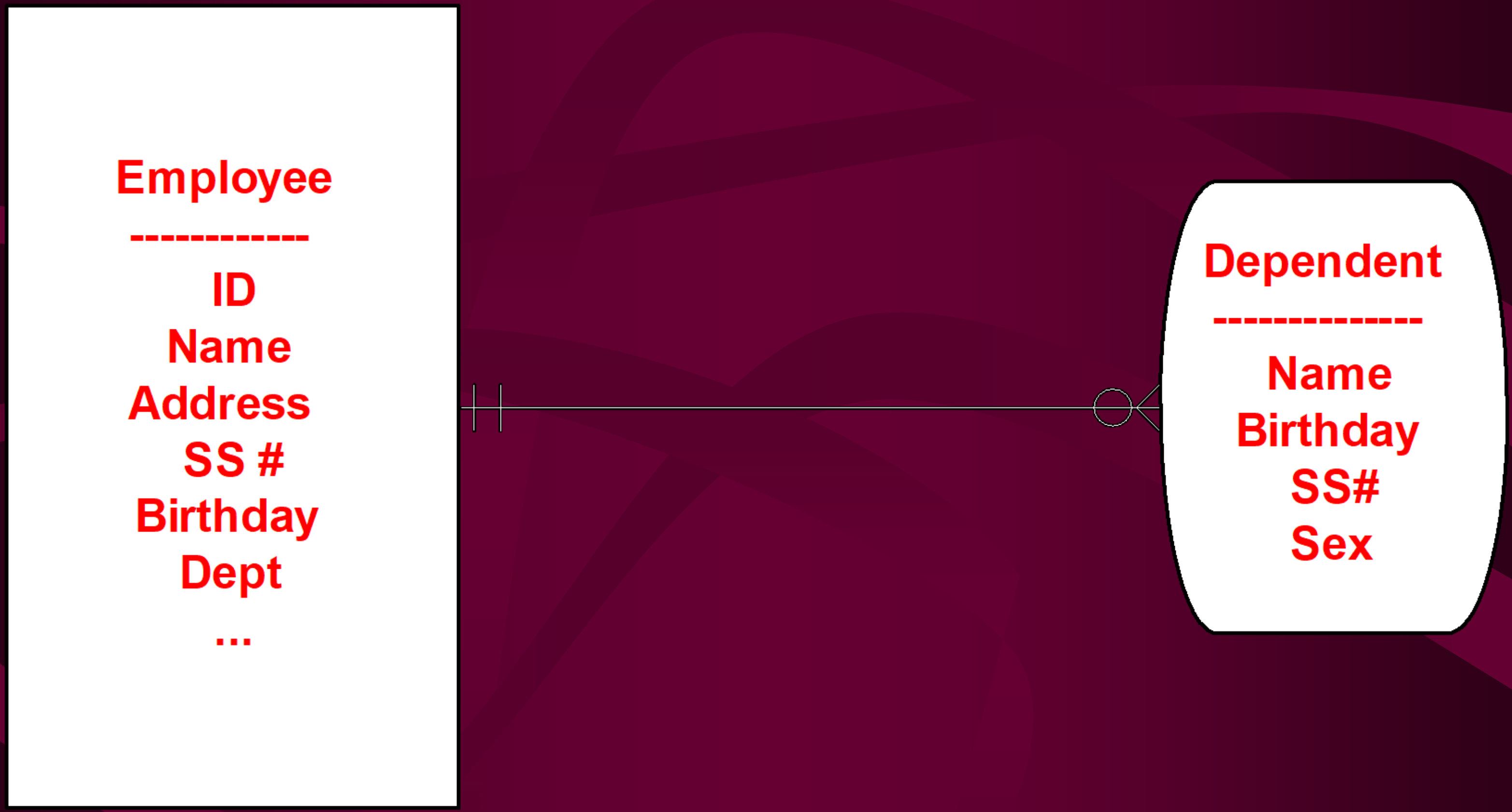
- Weak (non-identifying) relationship
  - Primary key of the related entity does not contain a primary key component of the parent entity
- Strong (identifying) relationships
  - ID dependent weak entity
  - Primary key of the related entity contains a primary key component of the parent entity



Multivalue Attributes

- Multivalue attributes can be represented as separate entities in an ER model (if the CASE product does not support multivalue attributes directly)
- This separate entity may be weak and possibly ID dependent
- As an example, an employee may have multiple dependents
  - Instead of “dependent” being an attribute of employee, it would be a weak entity with a relationship to the employee entity class (if each dependent has one and only one employee)

Is the Dependent entity weak ?
Is the Dependent entity ID dependent ?



Employee

ID
Name
Address
SS #
Birthday
Dept
...

Dependent

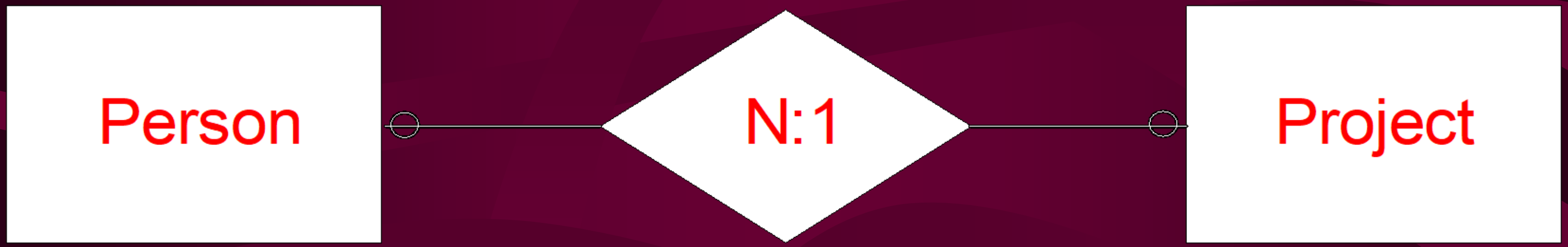
Name
Birthday
SS#
Sex

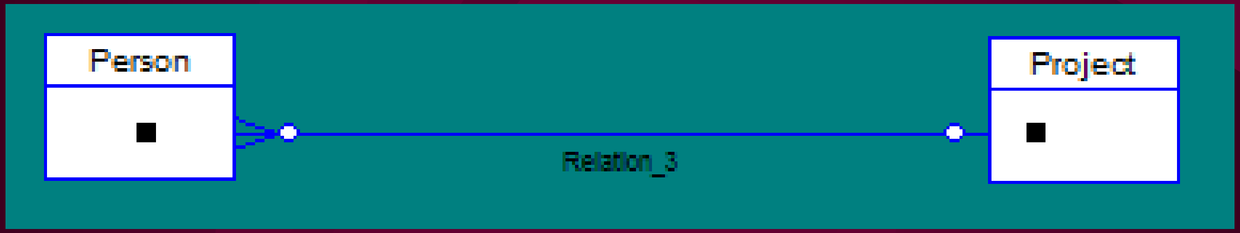
Exercise: Project - Person Relationship

- Draw the relationship between persons and projects
- Each person may be assigned to at most one project

Don’t look ahead !

One Project per Person



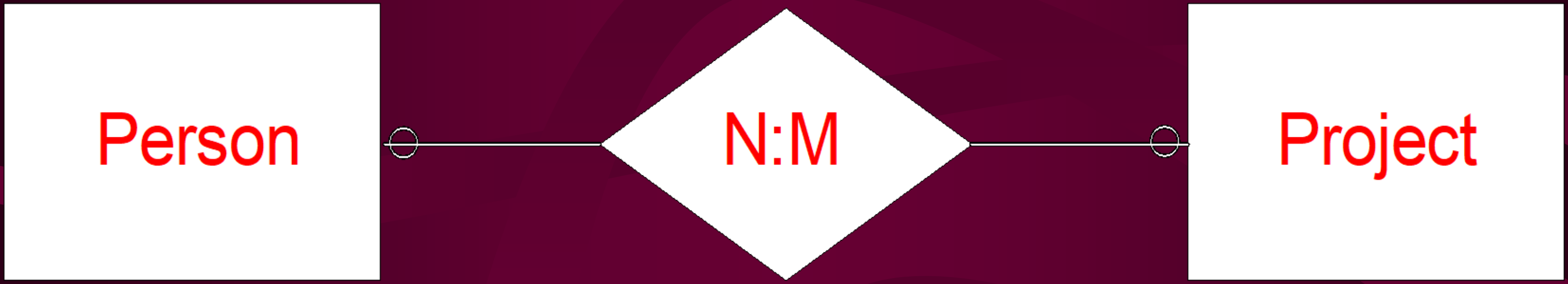


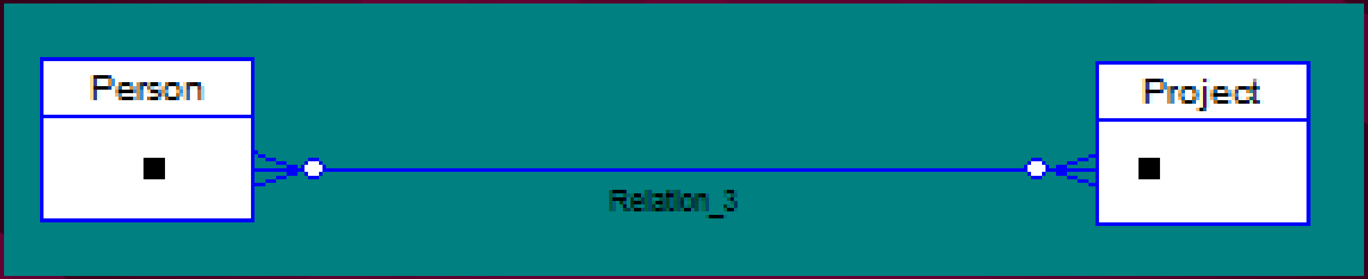
Some ER tools use the solo zero to mean min 0 max 1 cardinality.

- Change the work situation so that a person can now work on more than one project

Don’t look ahead !

Many Projects per Person



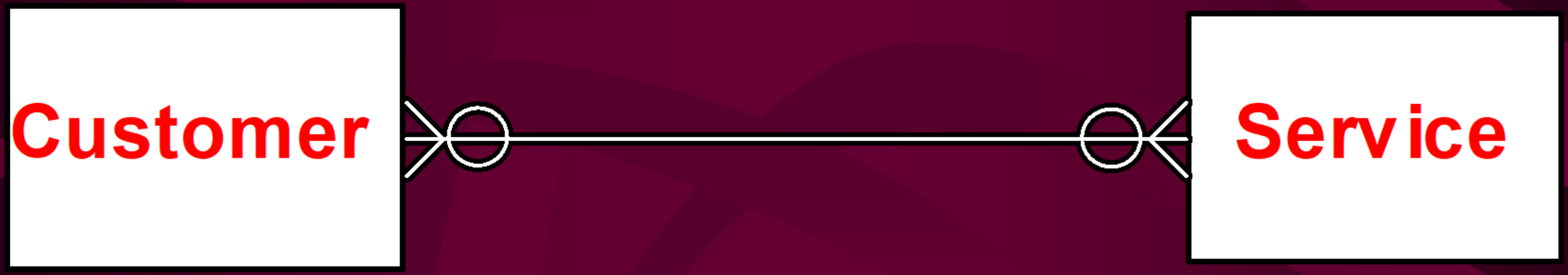


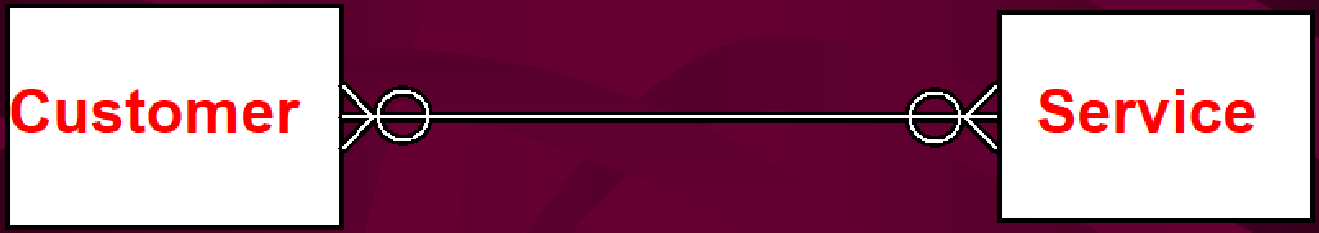
Class Exercise

- Draw the E-R diagram for a “service database” involving customers and services, where we offer multiple services to multiple customers

Don’t look ahead !

What type of binary relationship is this ?





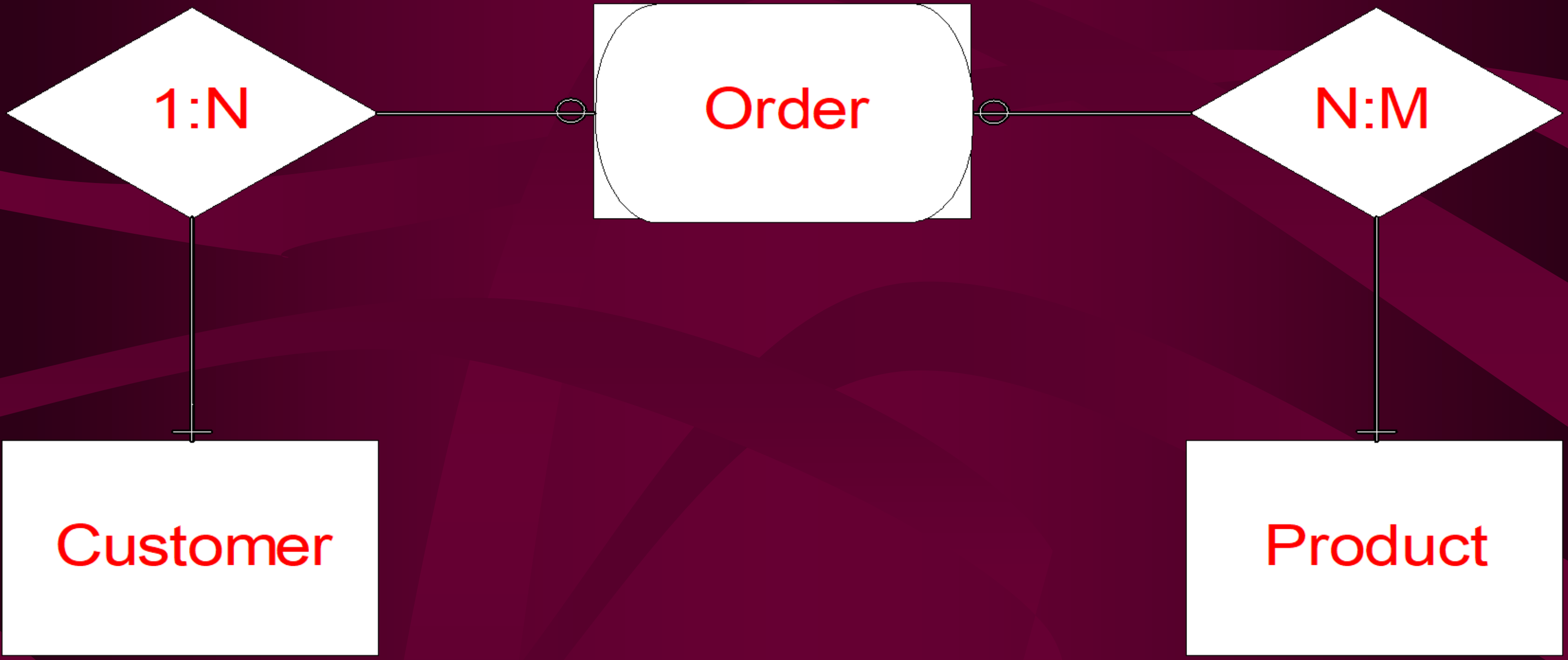
- If we charge each customer the same for a certain service, for which entity is the charge rate an attribute ?
- What if we charge different rates for the same service to different customers ?
  - Attribute in customer entity ?
  - Attribute in service entity ?

Class Exercise

- Draw the E-R diagram for customers, orders, and products:
  - Customer - customer makes orders for products
  - Order - orders by customers are for one or more products.
  - Products - finished products are ordered by customers (products are listed with their unit weight and unit cost)

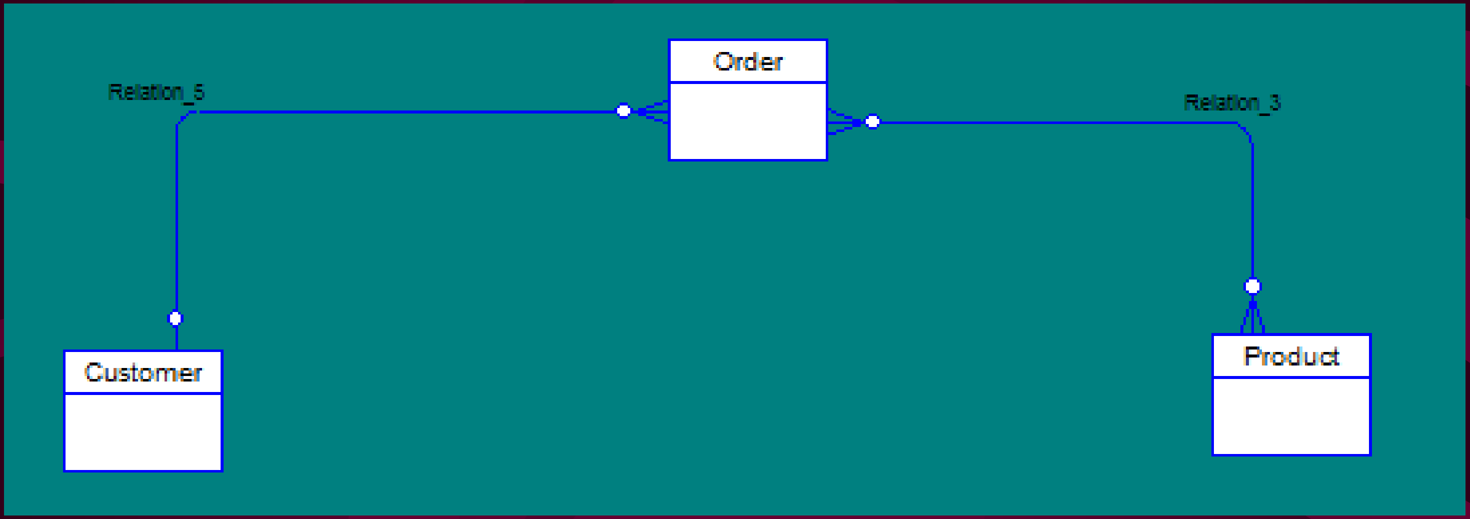
Don’t look ahead !

Customer - Order - Product



What about deleting customers ?
What about deleting products ?

Customer - Order - Product



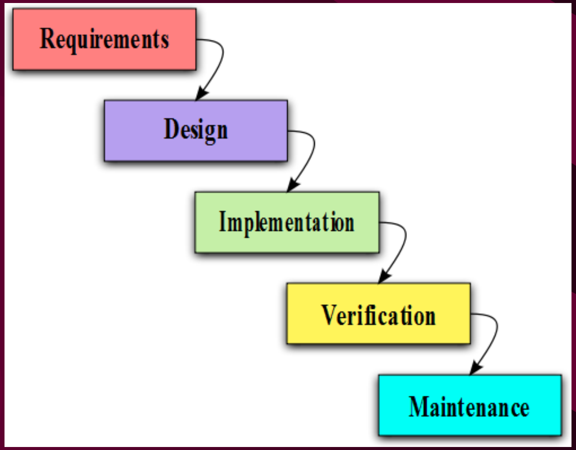
Where does the “quantity ordered” attribute go ?

Customer - Order – Product (con't)

- In classical MIS systems there is not a direct relationship between Customers and Products
- However in new "customer driven" systems such as Internet applications, such a relationship may be set up to keep track of which products (or product types) are "preferred" by customers; see www.amazon.com

Requirements Analysis

- Requirements analysis (MIS 351)
  - User interviews
  - Study of "artifacts"
  - Use case analysis
  - Workflow analysis
  - Information needs
    - Reporting
    - Queries
    - Processing



IDEF1X

- IDEF (Integrated Definition) became a U.S. national standard in 1993
  - IDEF1X (Definition 1, Extended)
  - Robert Brown 1979 [Lockheed]
- Only used for relational databases, and includes the concept of a domain
- Often required in US government contract work
- Extended versions of E-R diagrams now employ many IDEF concepts

Design Completeness

- It is very important to understand the requirements and get the relationships correct initially
- All necessary relationships should be included
- There should not be unnecessary or redundant relationships
- It is easy to alter relationships (ie one to many changed to many to many) at design time
- <u>After is systems is completed, and especially after data is in use (data loaded), it is very costly to modify !</u>

E-R Model Contents

- Entities
- Domains/Attributes
- Relationships
- What’s not in most/many E-R Models:
  - Other Business Rules (constraints, triggers, etc.)
  - Performance (indexes, physical layouts)

Business Rules - not part of standard E-R Diagram

- Some may be part of the data model and implemented in the schema
  - Constraints (including referential integrity)
  - Triggers
  - Stored procedures
- Some may be part of the application tool metadata (and thus reside with the database)
  - Formats
  - Defaults
  - Validations (simple attribute constraints)
- Others are implemented in the application programming language

Class Exercise

- Consider the following entities:
  - Auto Insurance Company
  - Insurance Agents
  - Policyholders
  - Policies
  - Vehicles

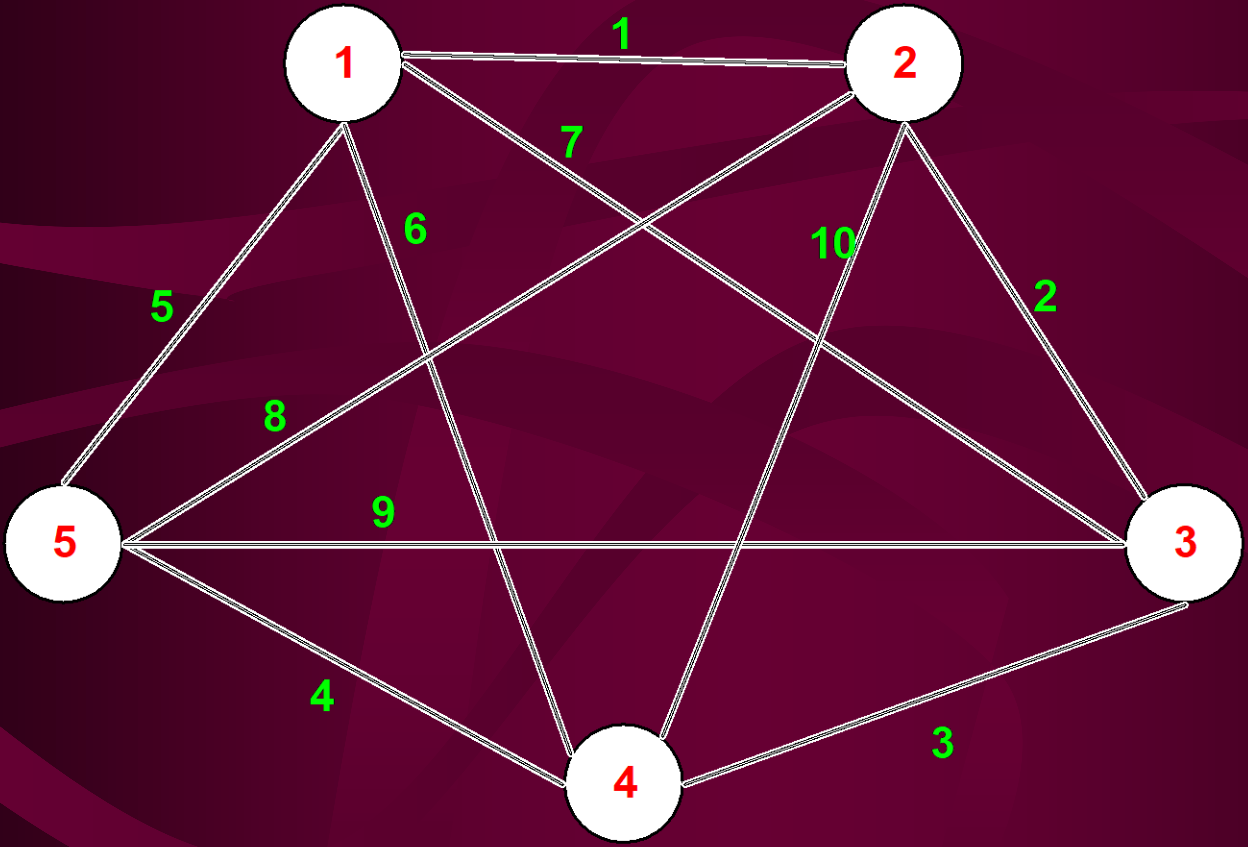
- What are the relationships, draw an E-R diagram
- Consider what different relationships might exists between agents and companies, between policyholders and agents, etc.

Don’t look ahead !

Class Exercise (con't)

- There are five entity classes
  - Auto Insurance Company
  - Insurance Agents
  - Policyholders
  - Policies
  - Vehicles
- How many possible relationships are there ?

N(N-1)/2   10 possible



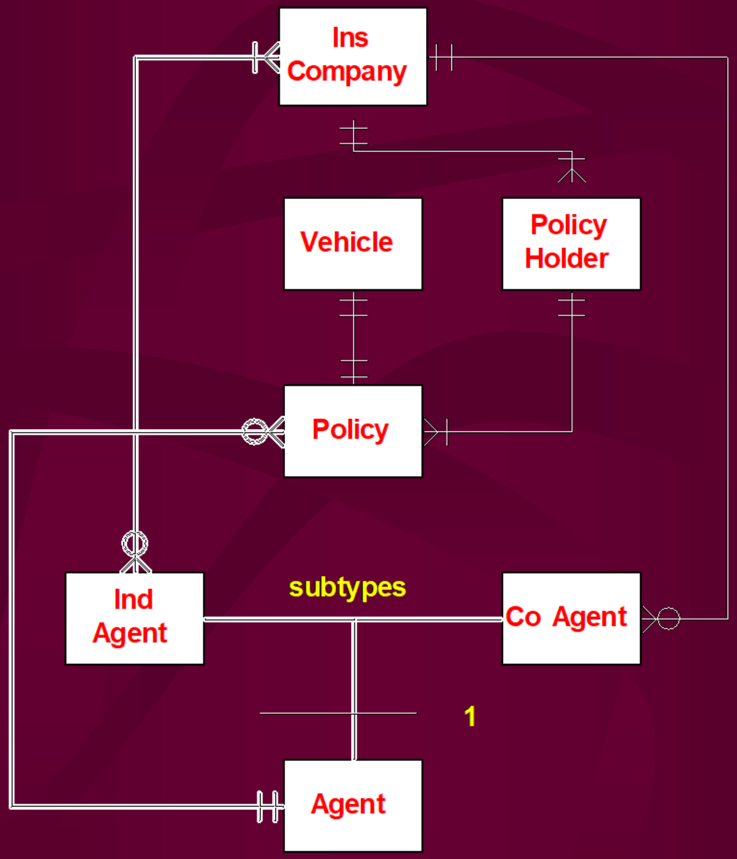
Selecting Relationships

- All relationships necessary for business requirements
- No
  - Unnecessary relationships
  - Redundant relationships

Class Exercise (con't)

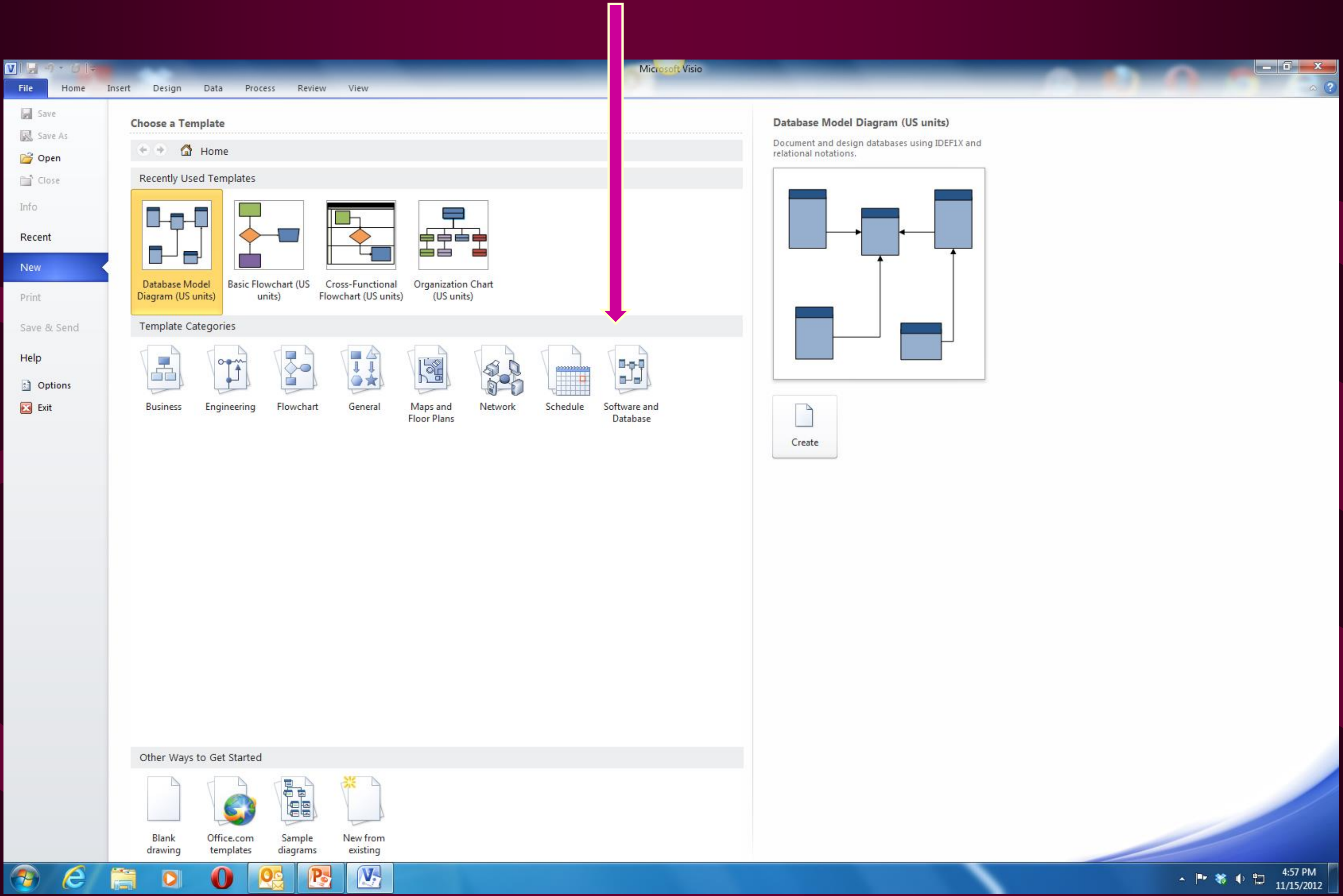
- Normally you will have a word problem or set of requirements which generally specifies domains and business rules
- This exercise was purposely vague:
  - Company agents vs Independent agents
  - Agents exclusive rights to policyholder (or not)

One of many models:



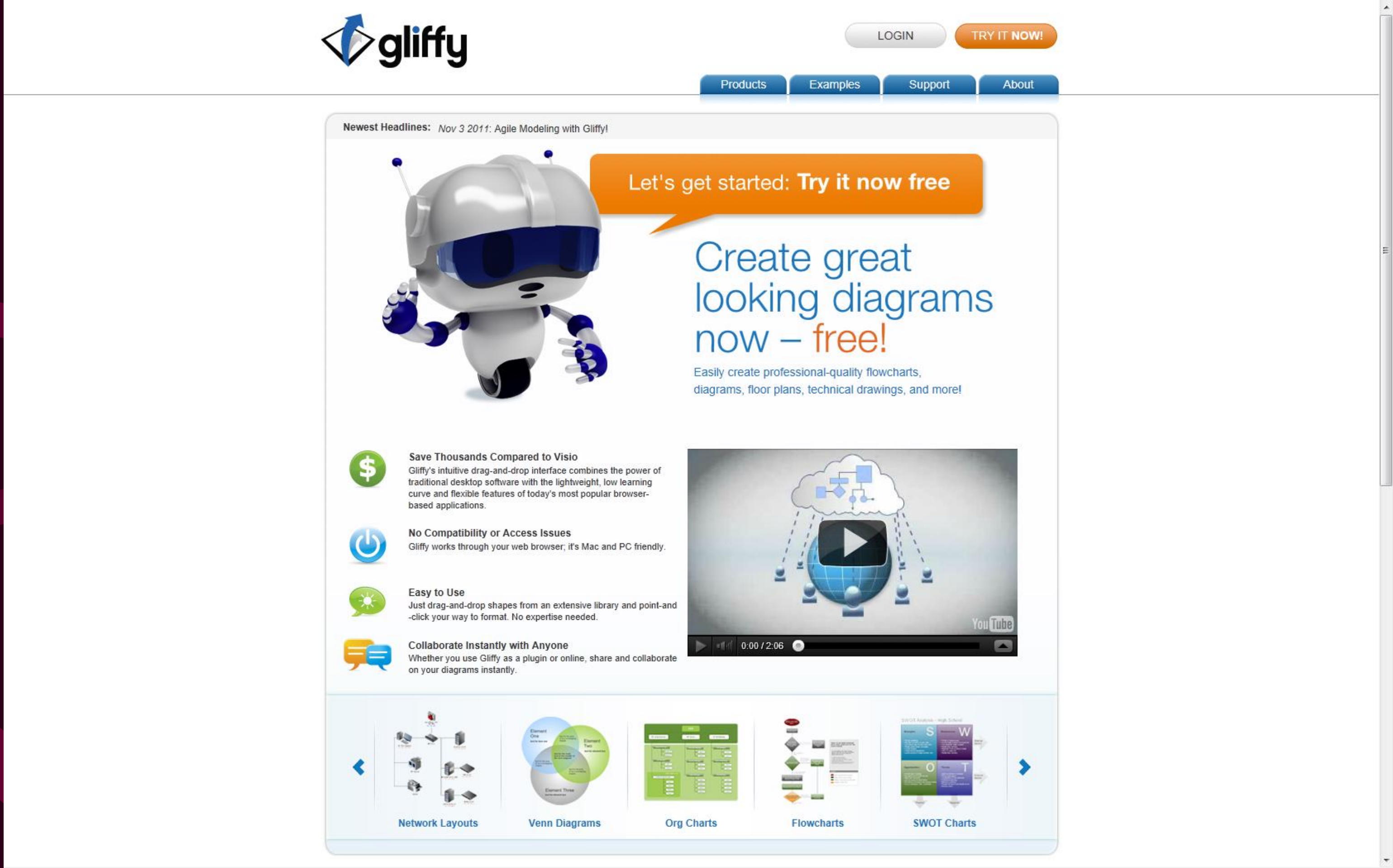
Sub types discussed
in a later lesson.

Drawing Products
[i.e. Visio Professional]



Document and design databases using IDEF1X and relational notations.

Gliffy
[www.gliffy.com]



Newest Headlines: *Nov 3 2011*: Agile Modeling with Gliffy!

Let's get started: **Try it now free**

Create great looking diagrams now – free!

Easily create professional-quality flowcharts, diagrams, floor plans, technical drawings, and more!

Save Thousands Compared to Visio
Gliffy's intuitive drag-and-drop interface combines the power of traditional desktop software with the lightweight, low learning curve and flexible features of today's most popular browser-based applications.

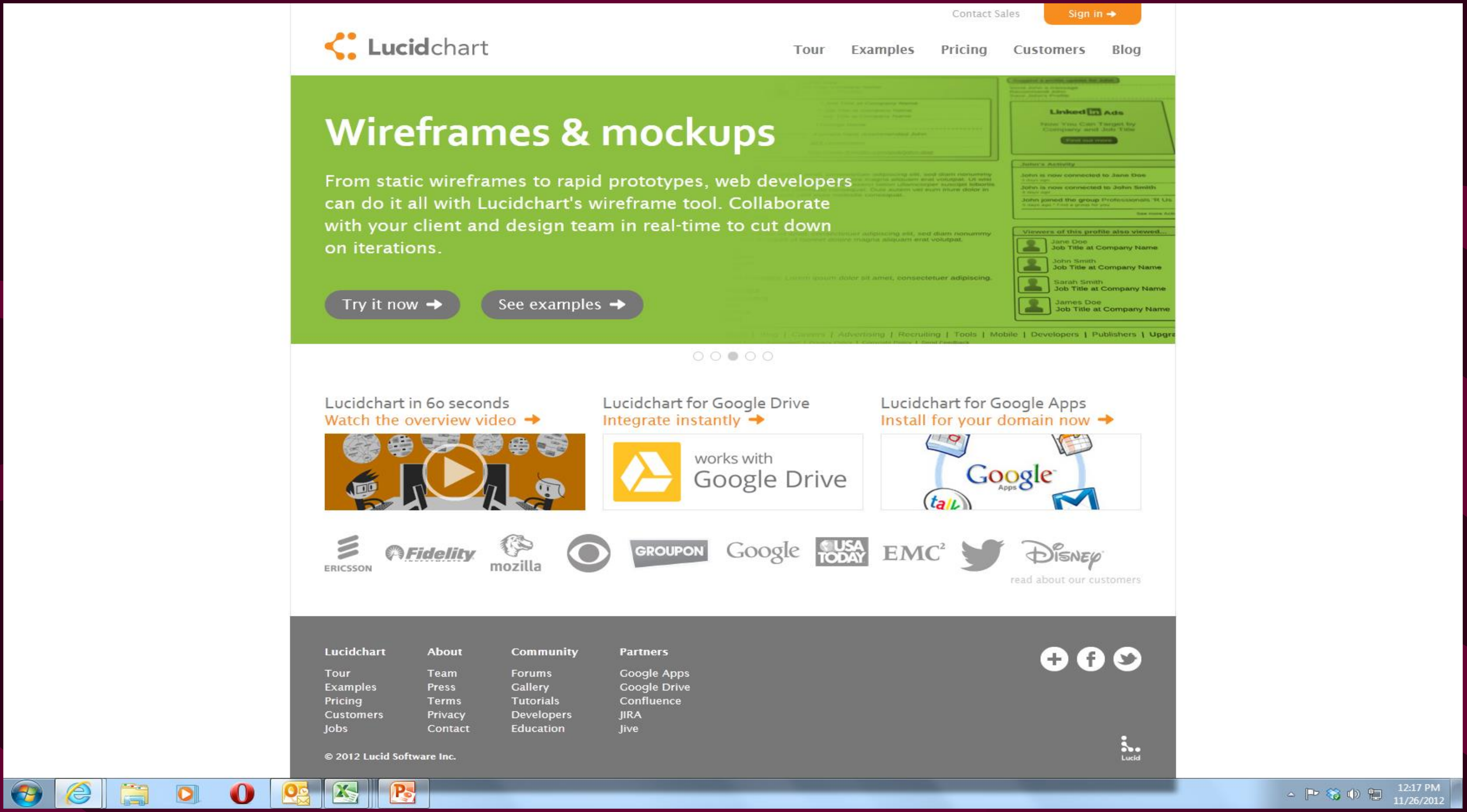
No Compatibility or Access Issues
Gliffy works through your web browser; it's Mac and PC friendly.

Easy to Use
Just drag-and-drop shapes from an extensive library and point-and-click your way to format. No expertise needed.

Collaborate Instantly with Anyone
Whether you use Gliffy as a plugin or online, share and collaborate on your diagrams instantly.

LucidChart

[https://www.lucidchart.com/]



Wireframes & mockups

From static wireframes to rapid prototypes, web developers can do it all with Lucidchart's wireframe tool. Collaborate with your client and design team in real-time to cut down on iterations.

Lucidchart in 60 seconds
Watch the overview video

Lucidchart for Google Drive
Integrate instantly

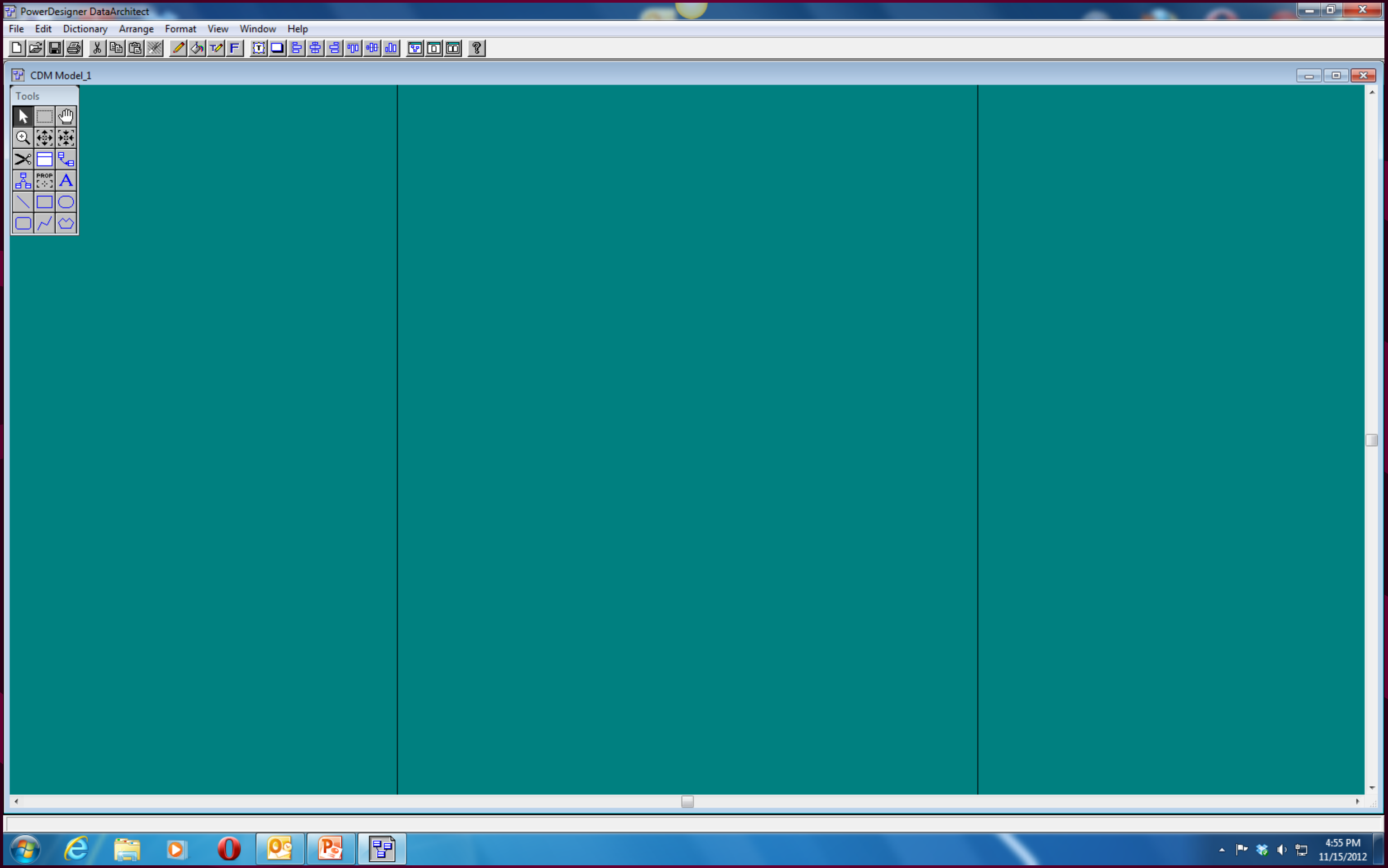
Lucidchart for Google Apps
Install for your domain now

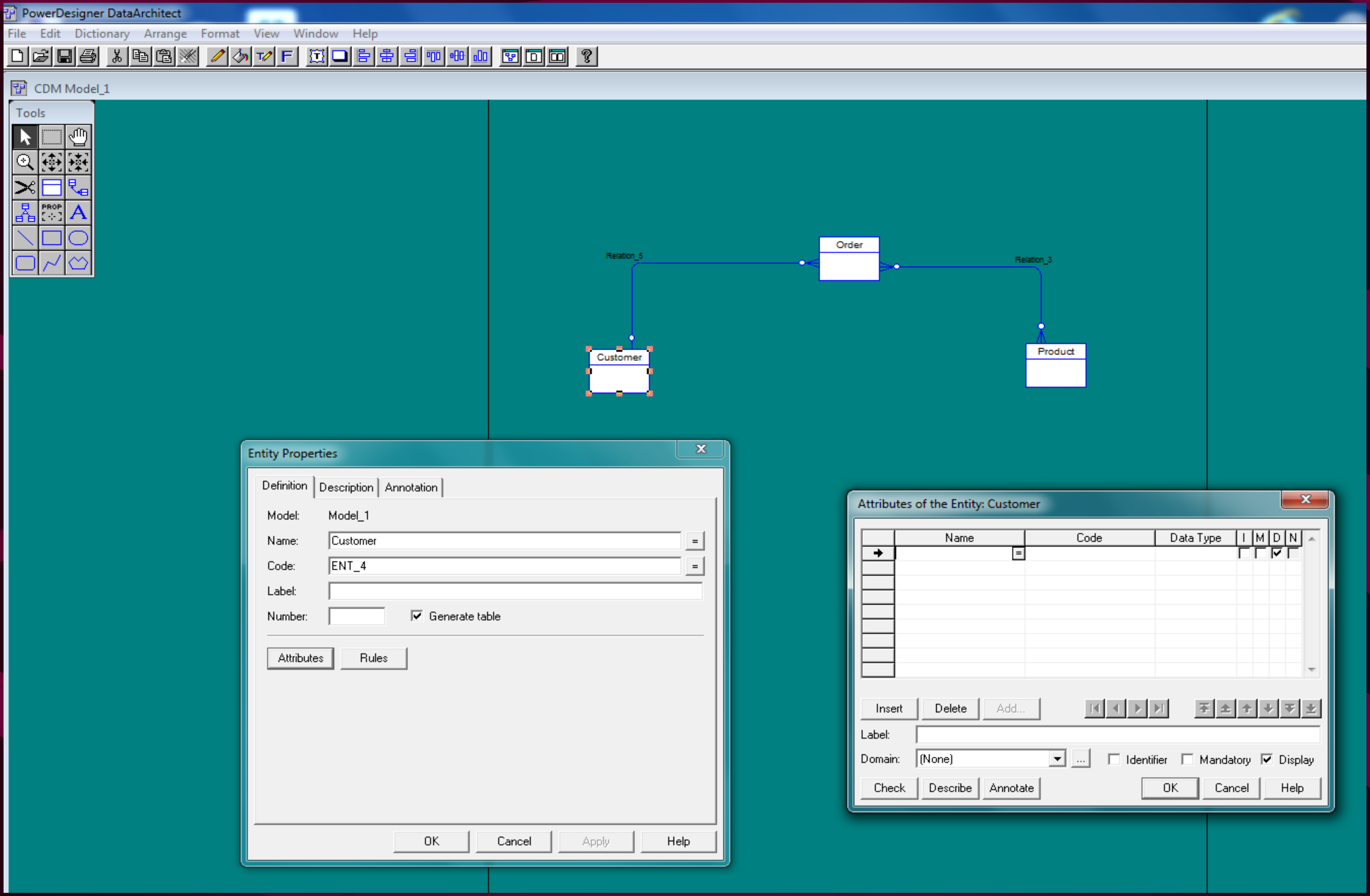
© 2012 Lucid Software Inc.

Integrates with the CBU Google apps account, the free version limits you to 60 objects/document.

CASE Products

[i.e. Data Architect, can automatically create database and/or SQL for table creation]


PowerDesigner DataArchitect
File Edit Dictionary Arrange Format View Window Help
CDM Model_1
Tools
4:55 PM
11/15/2012



Entity Properties

Definition | Description | Annotation

Model: Model_1

Name: Customer

Code: ENT_4

Label:

Number:

☑ Generate table

Attributes of the Entity: Customer

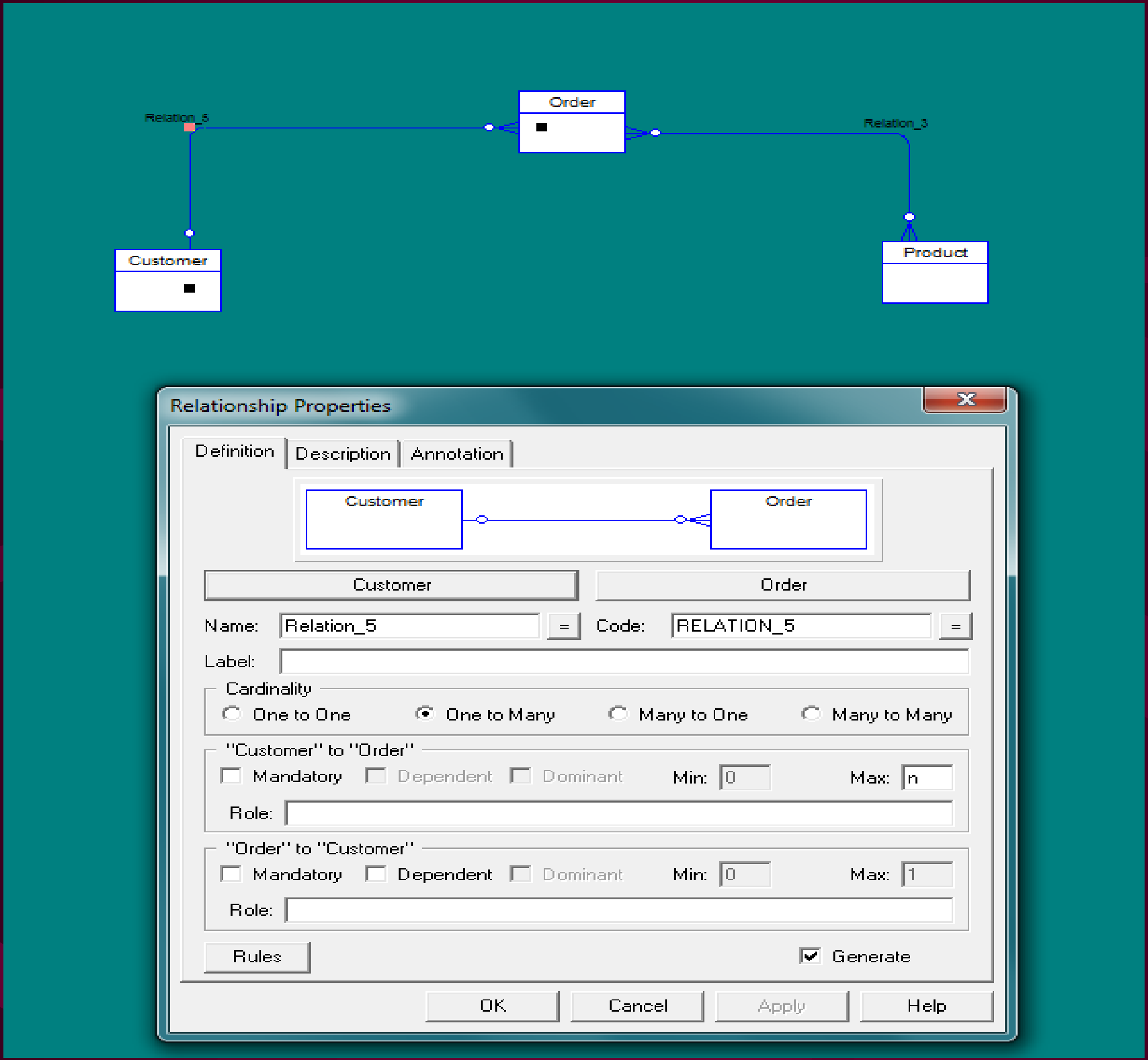
Name	Code	Data Type	I	M	D	N
			□	□	☑	□

Label:

Domain: (None)

□ Identifier □ Mandatory ☑ Display

Relationship Properties



Customer | Order

Name: Relation_5   Code: RELATION_5

Label:

Cardinality

○ One to One   ● One to Many   ○ Many to One   ○ Many to Many

"Customer" to "Order"

□ Mandatory   □ Dependent   □ Dominant   Min: 0   Max: n

Role:

"Order" to "Customer"

□ Mandatory   □ Dependent   □ Dominant   Min: 0   Max: 1

Role:

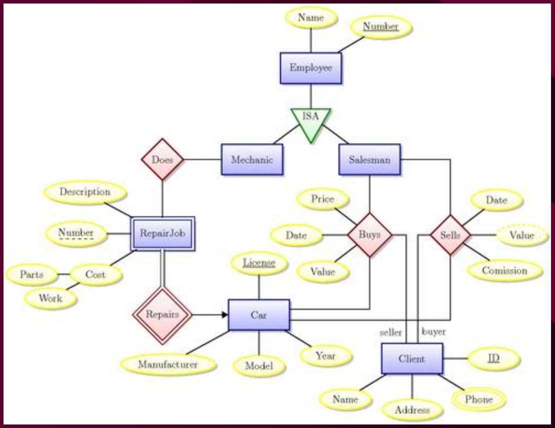
☑ Generate

References

- Chen, P., “The Entity-Relationship Model - Towards a Unified View of the Data Model”, ACM Transactions on Database Systems, 1976
- Bruce, T., Designing Quality Databases with IDEF1X Information Models, Dorset House
- Data Modeling Essentials by Graeme Simsion and Graham Witt
- Entity-Relationship Modeling: Foundations of Database Technology by B. Thalheim
- Data Modeling and Database Design by Richard W. Scamell and Narayan S. Umanath
- Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges by David W. Embley and Bernhard Thalheim

Homework

- Textbook Chapter 4
- Textbook Questions 1 thru 9
- Textbook Problem 1


Name
Number
Employee
ISA
Does
Mechanic
Salesman
Description
Number
RepairJob
Price
Date
Buys
Sells
Date
Value
Value
Comission
License
Parts
Cost
Work
Repairs
Car
seller
buyer
Manufacturer
Model
Year
Client
ID
Name
Address
Phone