Updated 11/2/2022

Topic 2: Database design
L11: ER modeling

Wolfgang Gatterbauer

CS3200 Database design (fa22)

https://northeastern-datalab.github.io/cs3200/fa22s3/

10/17/2022

Class warm-up

- Quick exam1 discussion (more on WED):
  - Points vs Grades
  - Was it fair and types of problems similar to problems seen in class and on HWs? Open-book vs closed book (open is more time constraint)
  - exam2 will have paper and computer components
  - Discuss example solutions next class? If yes, poll in class next time.

- Please use our various options for feedback

- Starting Database Design today

The "Surfer Analogy" for time management

Source: http://stwww.surfermag.com/files/2013/10/Yak_Charlie-970x646.jpg

Entity-Relationship Diagram (ERD) for IMDB

?

Situation:

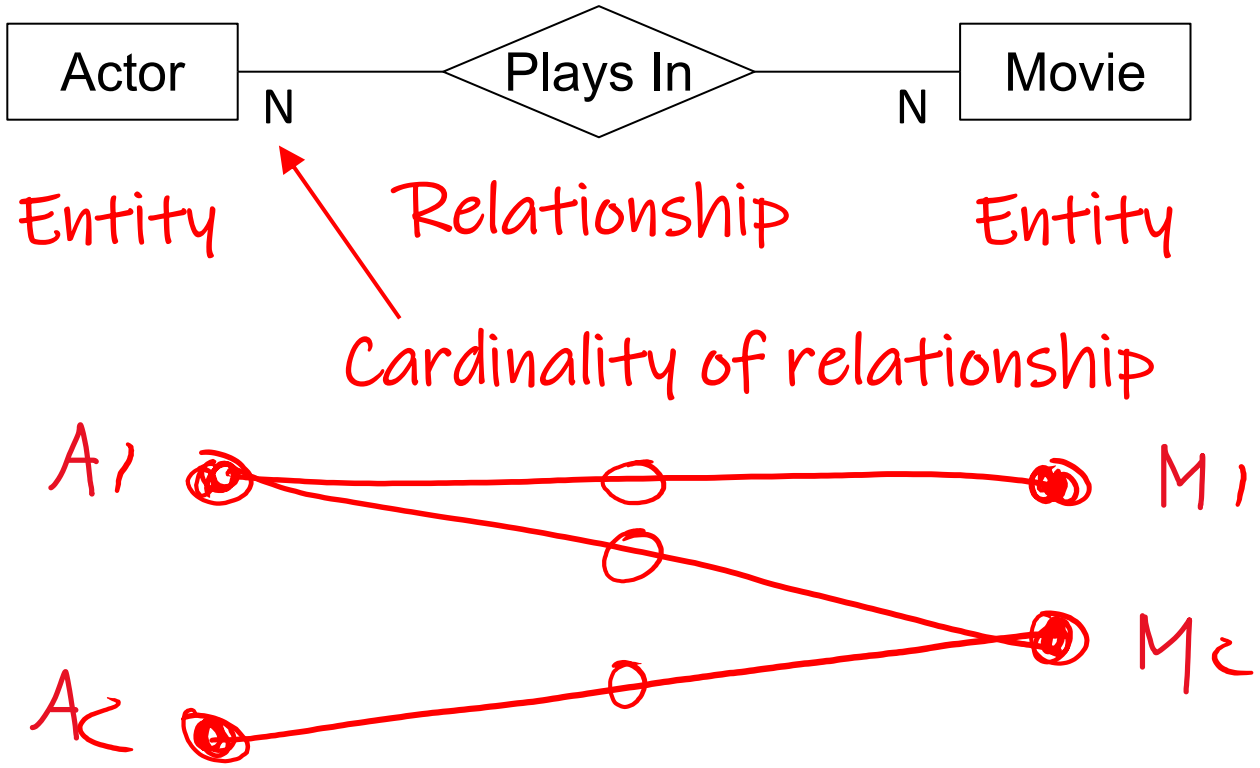
- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.

Entity-Relationship Diagram (ERD) for IMDB

Situation:

- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.

Entity-Relationship Diagram (ERD) for IMDB



Situation:

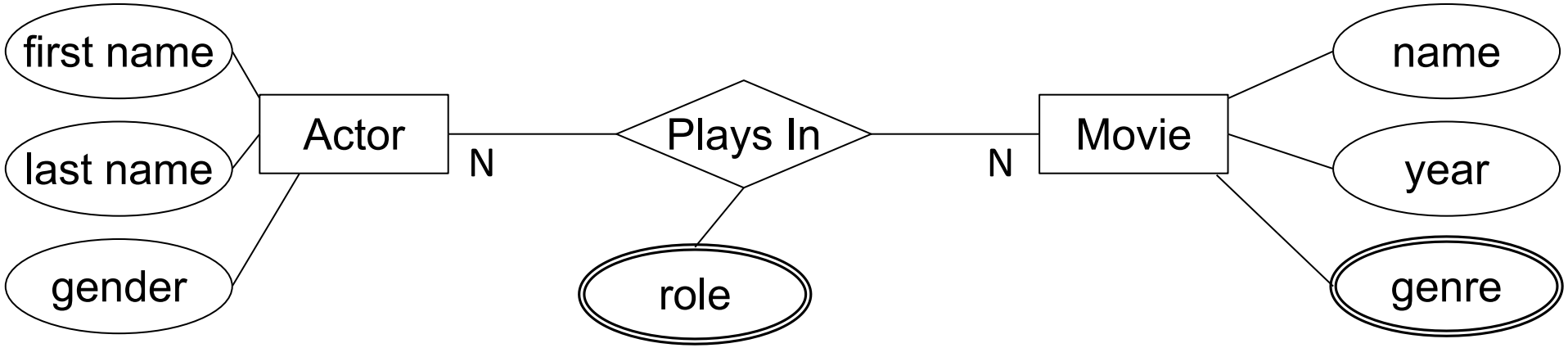
- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.

Entity-Relationship Diagram (ERD) for IMDB

Situation:

- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.
- Actors have names and gender. Movies have name, year and can have multiple genres

Entity-Relationship Diagram (ERD) for IMDB



attributes

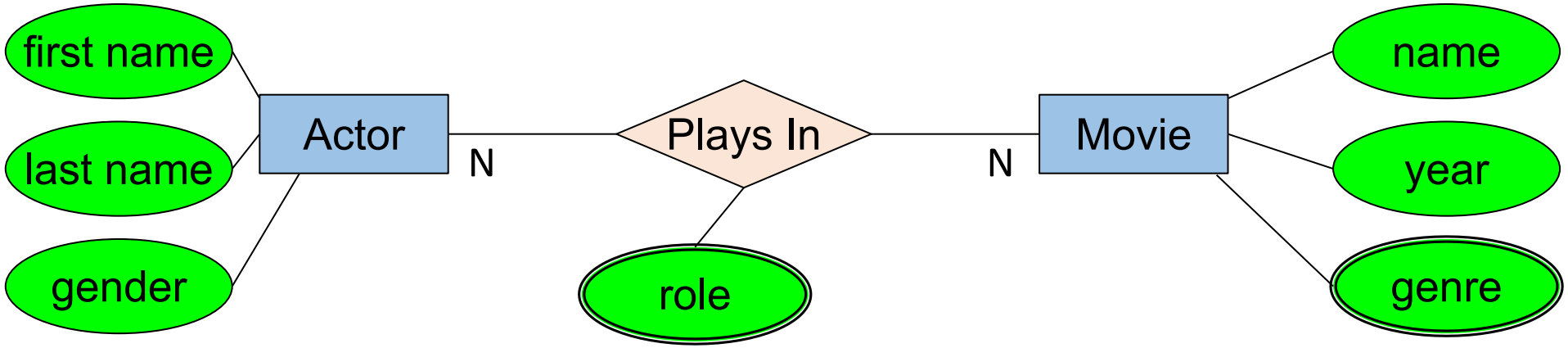
Multivalued attribute:
a movie can be assigned
0, 1 or more genres

Situation:

- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.
- Actors have names and gender. Movies have name, year and can have multiple genres

Entity-Relationship Diagram (ERD) for IMDB

color is optional



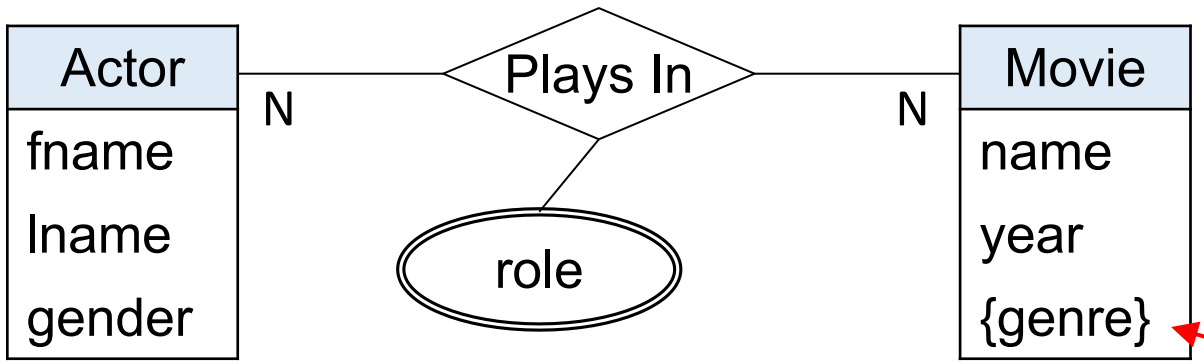
attributes

Multivalued attribute:
a movie can be assigned
0, 1 or more genres

Situation:

- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.
- Actors have names and gender. Movies have name, year and can have multiple genres

Entity-Relationship Diagram (ERD) for IMDB



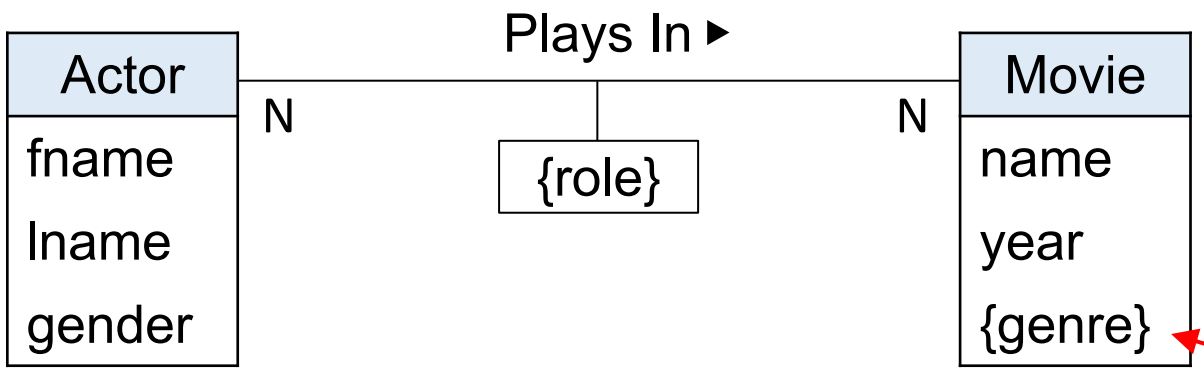
This is a mix between two notations. Don't do that!

Multivalued attribute: a movie can be assigned 0, 1 or more genres

Situation:

- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.
- Actors have names and gender. Movies have name, year and can have multiple genres

Entity-Relationship Diagram (ERD) for IMDB



alternative UML notation

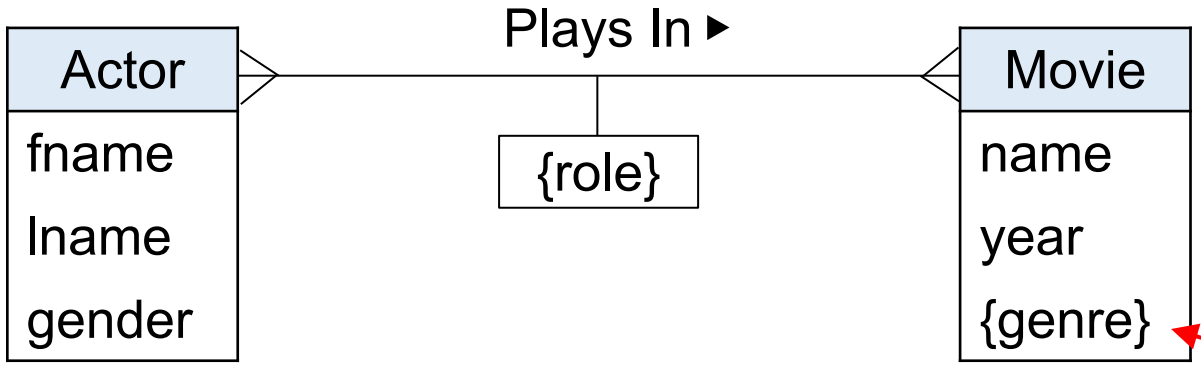
Multivalued attribute: a movie can be assigned 0, 1 or more genres

Situation:

- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.
- Actors have names and gender. Movies have name, year and can have multiple genres

Entity-Relationship Diagram (ERD) for IMDB

Crow feet notation



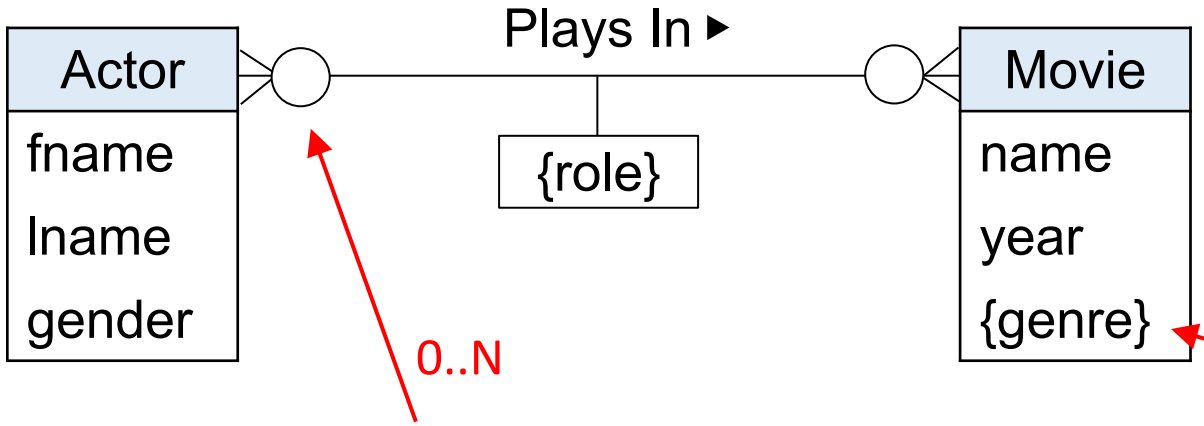
Multivalued attribute:
a movie can be assigned
0, 1 or more genres

Situation:

- There are actors and movies.
- Actors can play in multiple movies, movies can have multiple actors.
- Actors have names and gender. Movies have name, year and can have multiple genres

Entity-Relationship Diagram (ERD) for IMDB

This is still an ER diagram



How to represent this situation in a relational DBMS ?

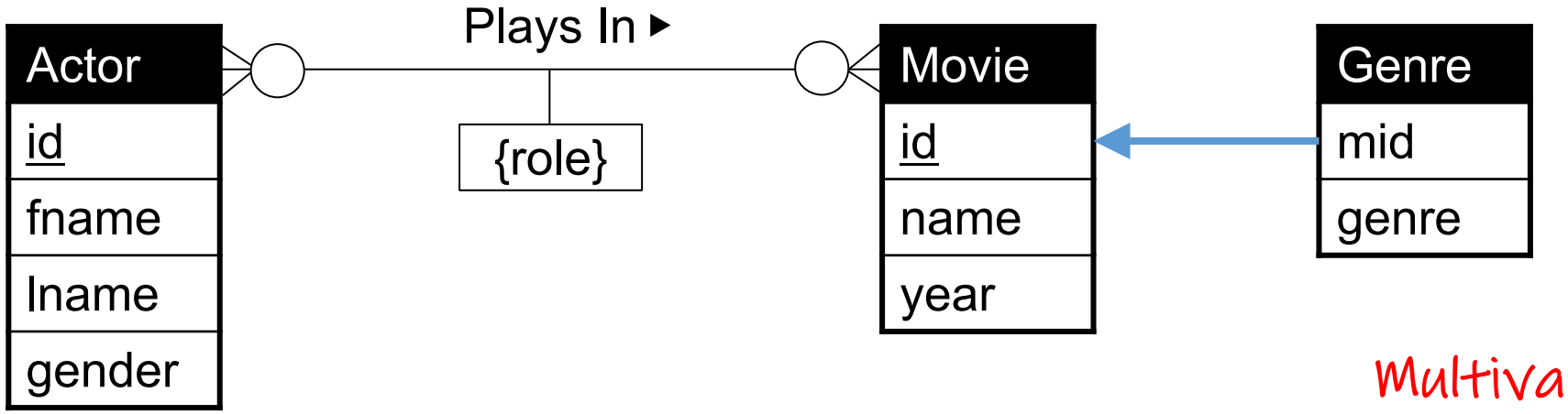
A movie can have 0 to many actors:
- min: 0 (participation constraint: "optional", not "mandatory")
- max: N = many (cardinalities)

Multivalued attribute: a movie can be assigned 0, 1 or more genres

Situation:
- There are actors and movies.
- Actors can play in 0, 1 or more movies, movies can have 0, 1, or more actors.
- Actors have names and gender. Movies have name, year and can have 0, 1, or more genres

ERD → Relational schema

This is a mix for illustration. Don't do that!



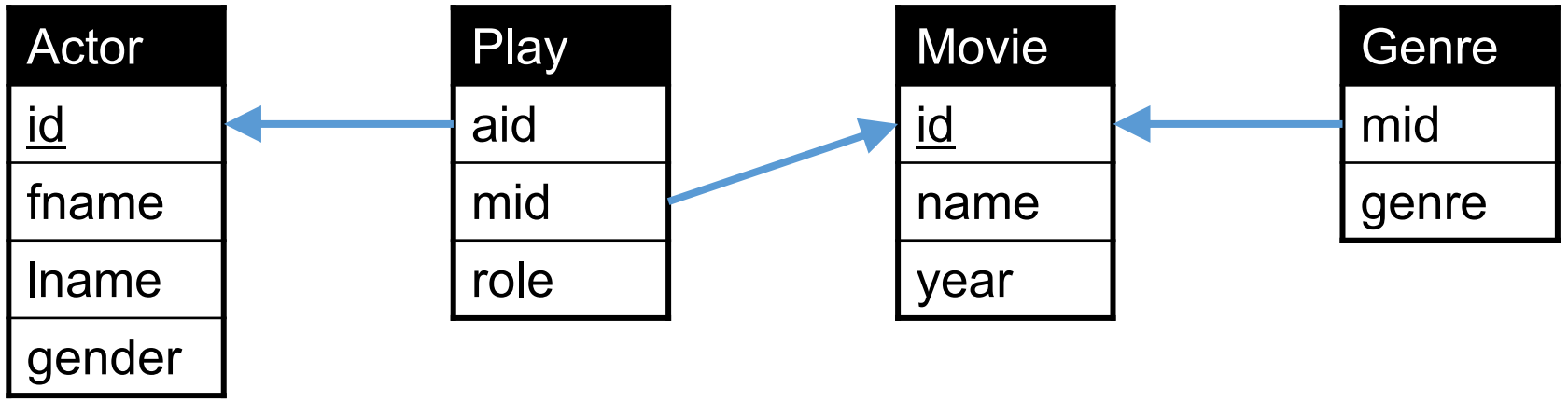
Multivalued attribute:
a movie can be assigned
0, 1 or more genres

Situation:

- There are actors and movies.
- Actors can playsin 0, 1 or more movies, movies can have 0, 1, or more actors.
- Actors have names and gender. Movies have name, year and can have 0, 1, or more genres

Relational schema for IMDB

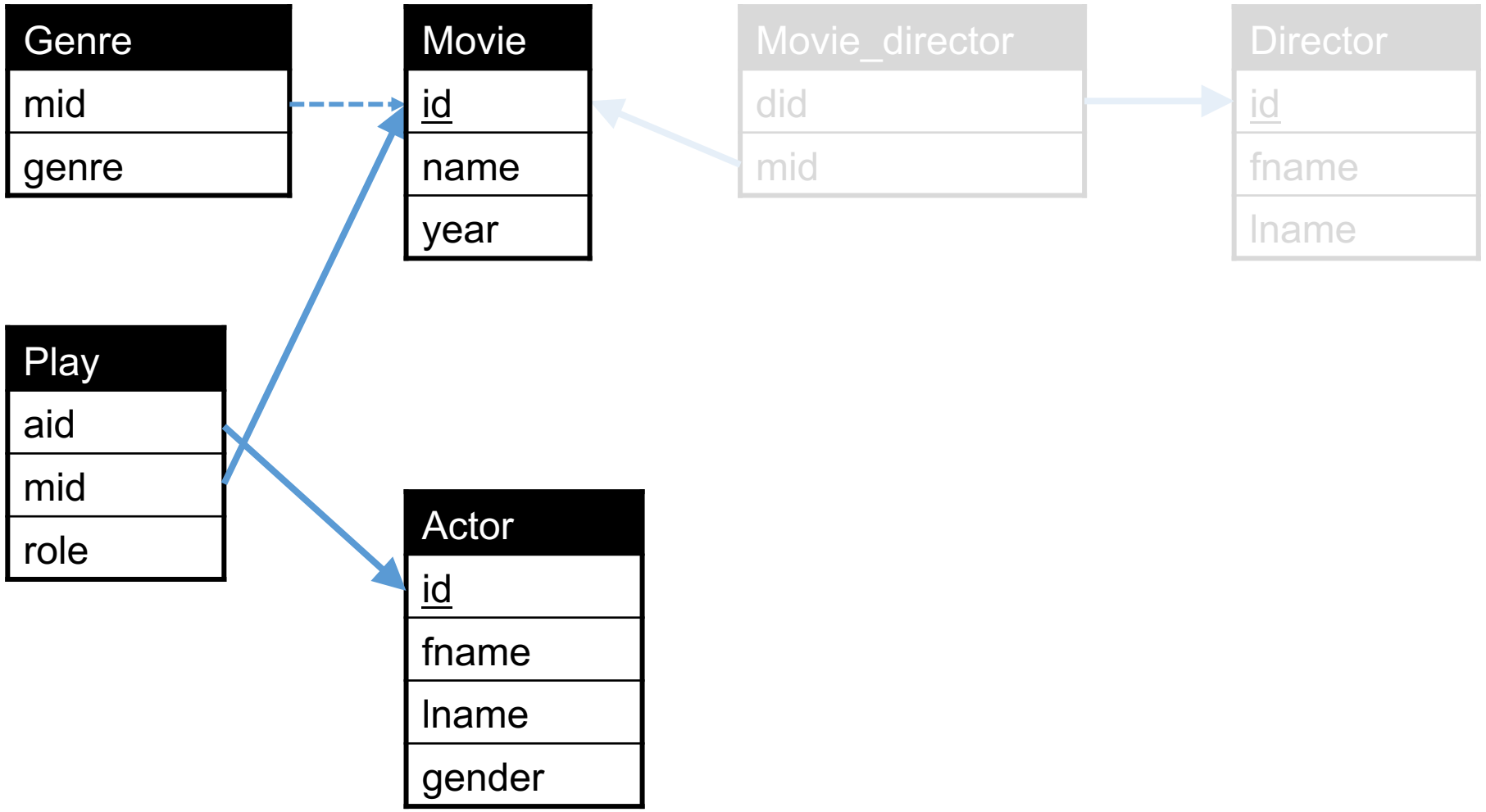
This is relational schema!



Situation:

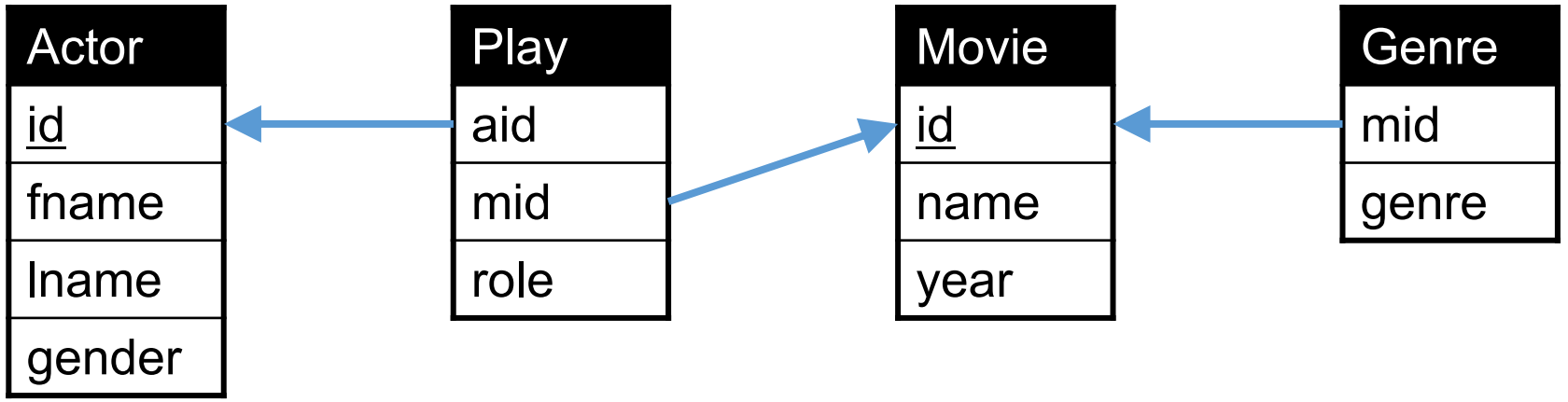
- There are actors and movies.
- Actors can play in 0, 1 or more movies, movies can have 0, 1, or more actors.
- Actors have names and gender. Movies have name, year and can have 0, 1, or more genres

Big IMDB schema (Postgres)



Relational schema for IMDB

This is relational schema!

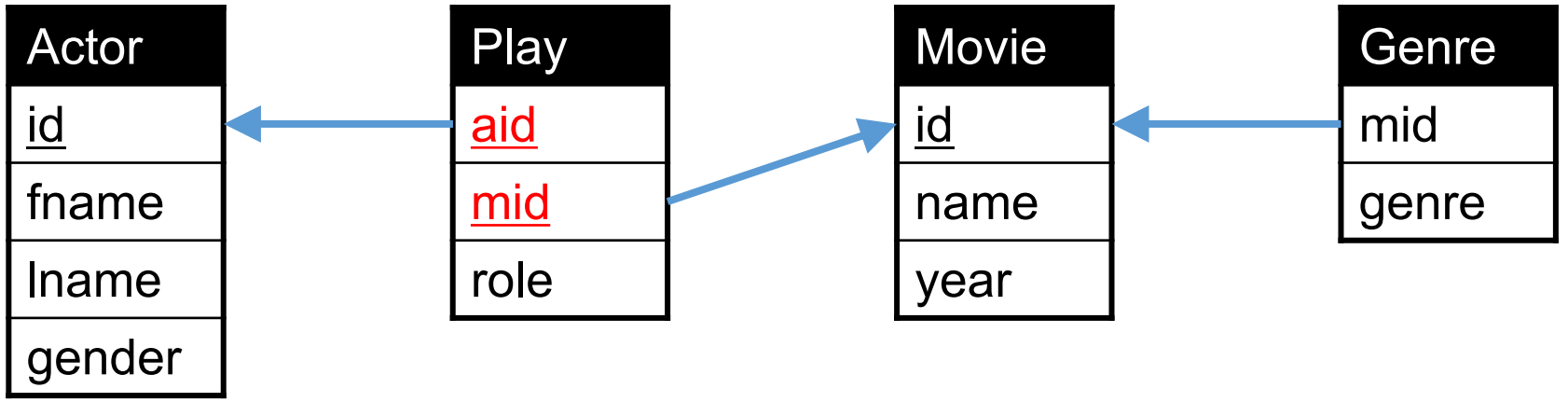


What if every actor play maximal one role in a movie? ?

Situation:

- There are actors and movies.
- Actors can play in 0, 1 or more movies, movies can have 0, 1, or more actors.
- Actors have names and gender. Movies have name, year and can have 0, 1, or more genres

Relational schema for IMDB



What if every actor play maximal one role in a movie? Define appropriate keys!

Situation:

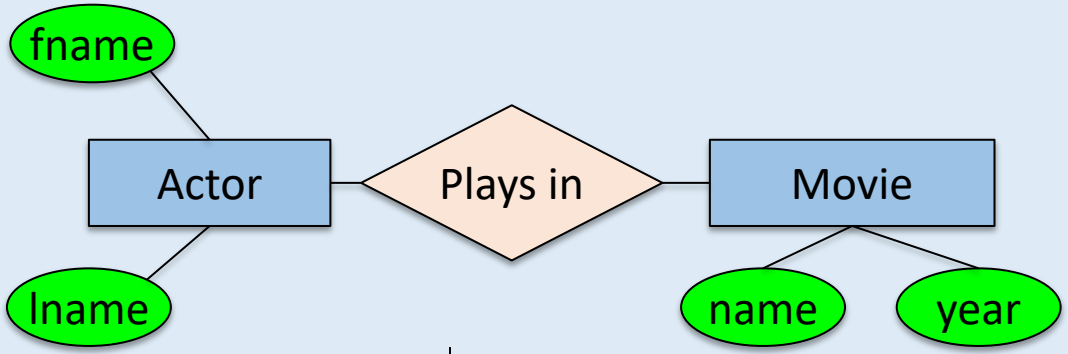
- There are actors and movies.
- Actors can play one role in 0, 1 or more movies, movies can have 0, 1, or more actors.
- Actors have names and gender. Movies have name, year and can have 0, 1, or more genres

ER modeling

Data modeling and Database Design Process

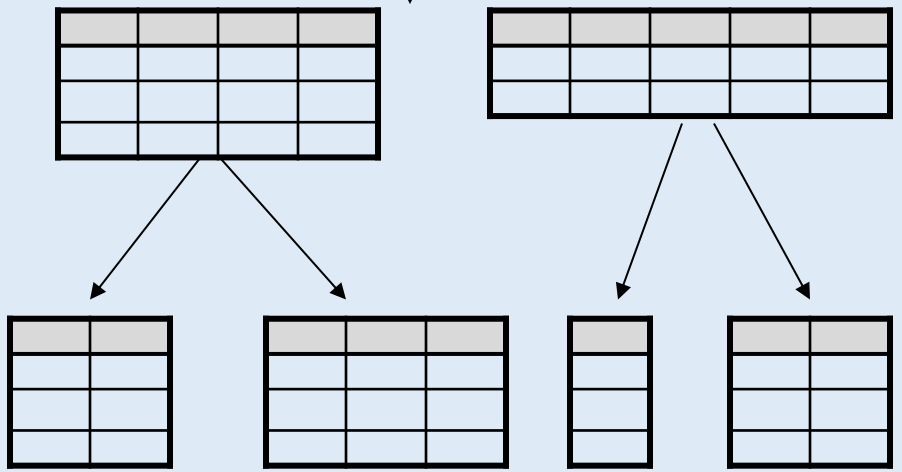
1. ER Diagram

Conceptual Model:
("technology independent")
describe main data items



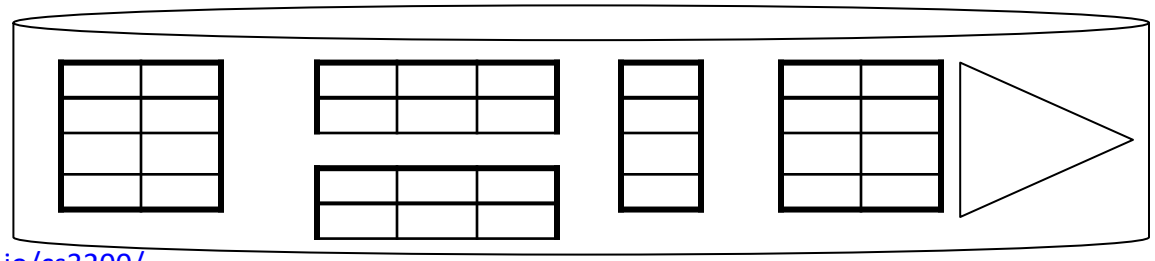
2. Relational Database Design

Logical Model
("for relational databases"):
Tables, Constraints
Functional Dependencies
Normalization:
Eliminates anomalies



3. Database Implementation

Physical Model
Physical storage details
Result: Physical Schema



Database Design

- Database design: Why do we need it?
  - Agree on structure of the database before deciding on a particular implementation

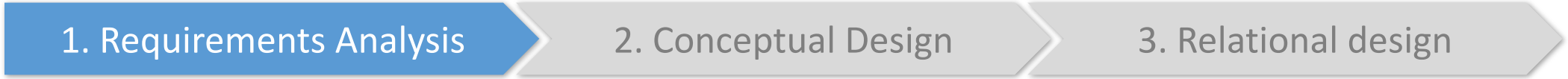
- Consider issues such as:
  - What entities to model
  - How entities are related
  - What constraints exist in the domain
  - How to achieve good designs

Relational model has only one concept: the relation (table)

ER have two, closer to model real-world situations: entities, relationships b/w/ entities

- Several formalisms for ERDs exist. We will discuss several, in particular
  - Stanford arrow notation (also our textbook **SDK**)
  - Crow's foot notation

Database Design Process



1. Requirements analysis

- What is going to be stored?
- How is it going to be used?
- What are we going to do with the data?
- Who should access the data?

Technical and non-technical people are involved

Database Design Process

ERD

Actual tables

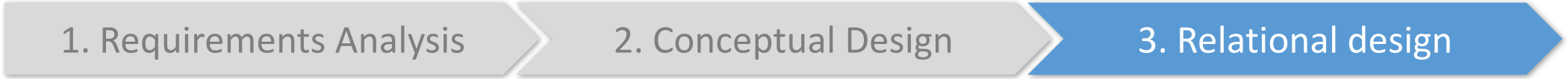
2. Conceptual Design

- A high-level description of the database
- Sufficiently precise that technical people can understand it
- But, not so precise that non-technical people can't participate

perfect fit for ER modeling

One of the main benefits of using ER diagrams instead of relational schemas directly is easier communication: The relationships are usually more visible in an ERD and the database structure becomes easier to understand and discuss

Database Design Process



ERD

Actual tables

3. Relational model

- Logical Database Design

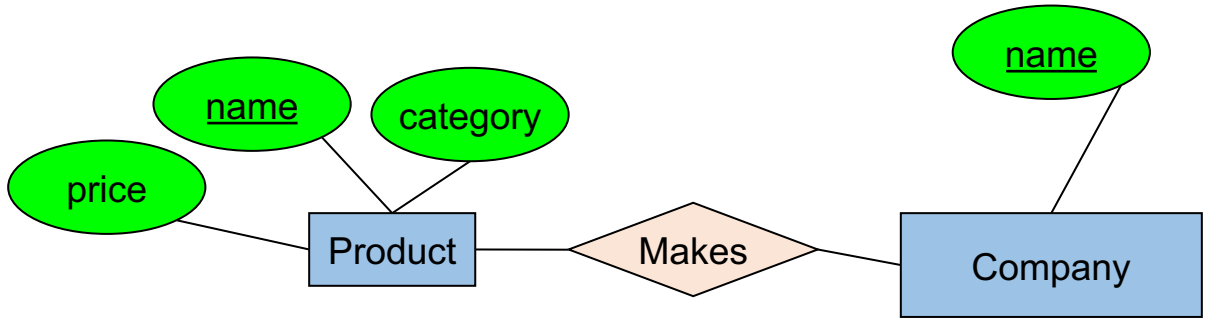
And More:

- Physical Database Design
- Security Design

Database Design Process

1. Requirements Analysis → 2. Conceptual Design → 3. Relational design

ER Diagrams are used



This process is iterated **many** times

E/R is a *visual syntax* for DB design which is ***precise enough*** for technical points, but ***abstracted enough*** for non-technical people

Interlude: Impact of the ER model

- The E/R model is one of the most cited articles in Computer Science
  - “The Entity-Relationship model – toward a unified view of data” Peter Chen, 1976
  - Compare to "business model canvas", Alexander Osterwalder 2008
    https://en.wikipedia.org/wiki/Business_Model_Canvas

- Used by companies big and small
  - You’ll know it soon enough

- "Chen notation": different from "UML"

Graphicacy

"Graphicacy is concerned with the capacities people require in order to interpret and generate information in the form of graphics."

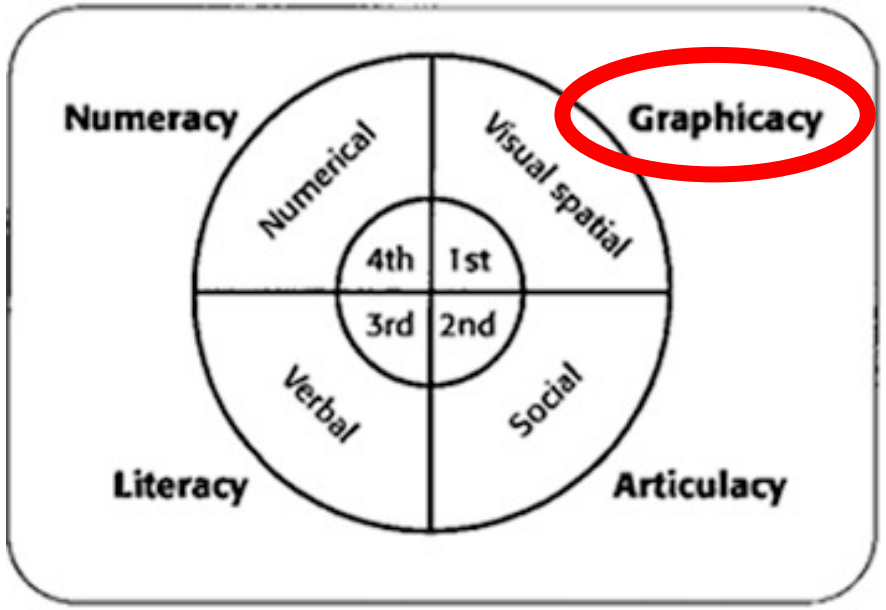
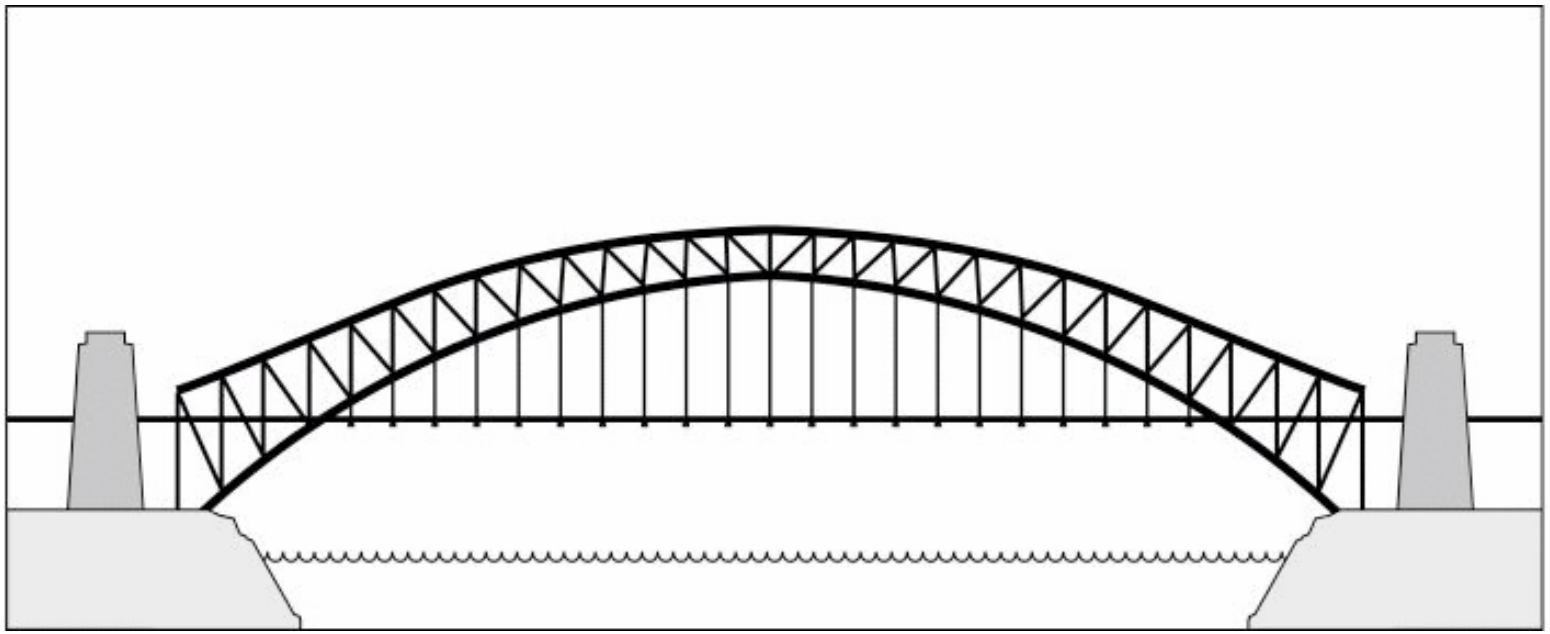
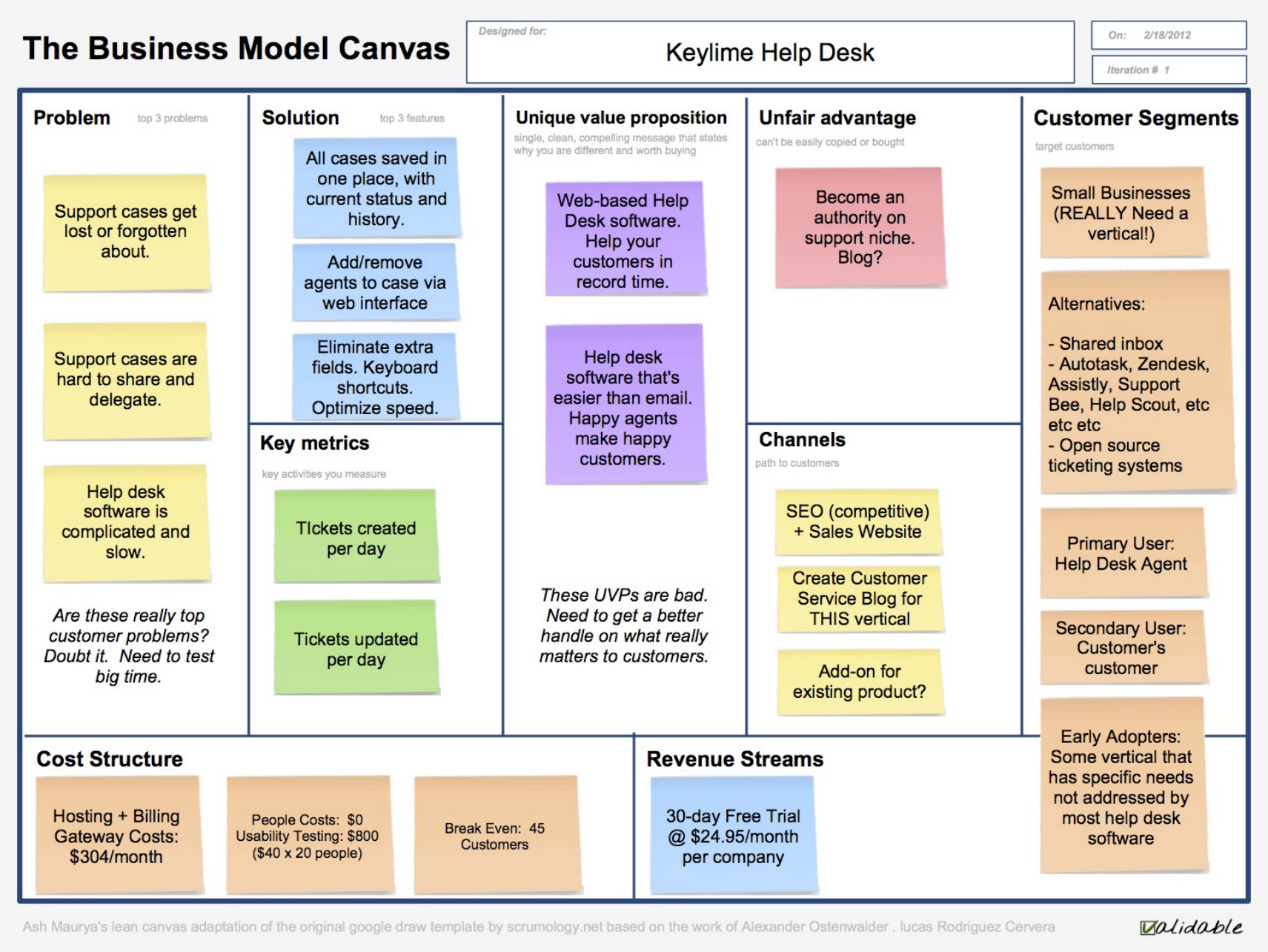


Figure 2. Balchin's "four types of ability."

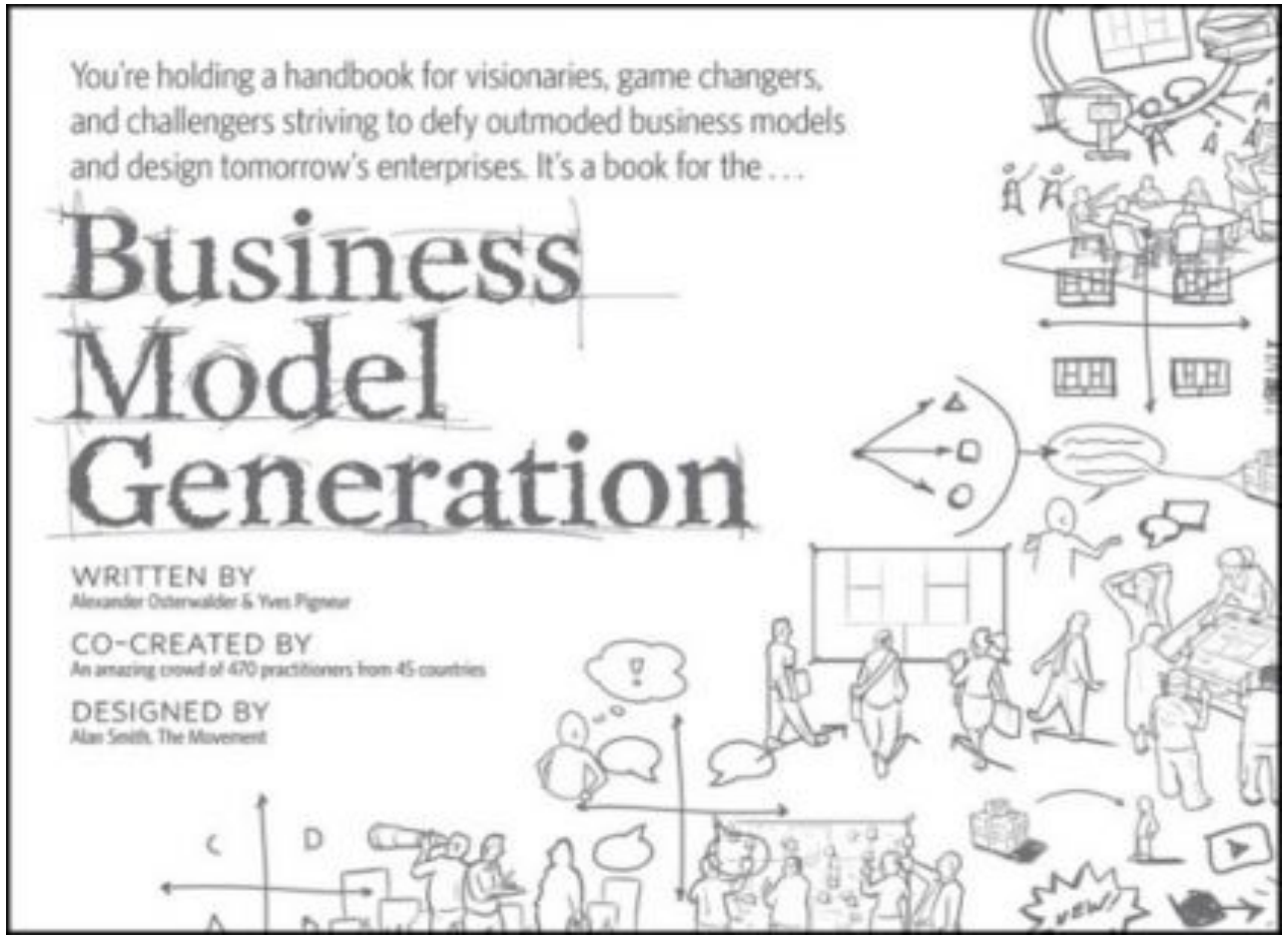


Source: W.G.V Balchin, Graphicacy 1976, http://en.wikipedia.org/wiki/Graphicacy

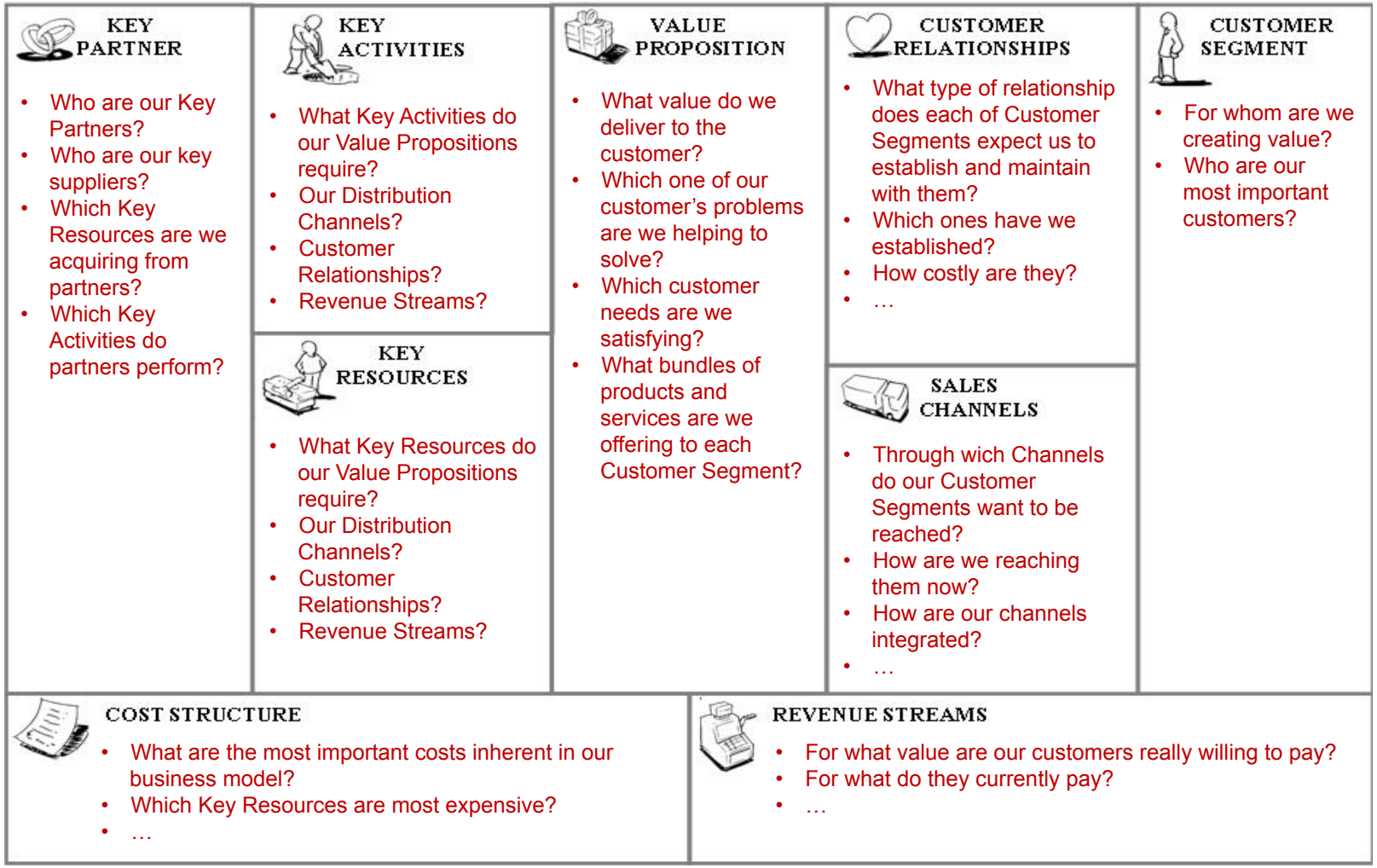
Example Framework 2: Business Model Canvas



Osterwalder & Pigneur (2010): Business Model Canvas (cont.)



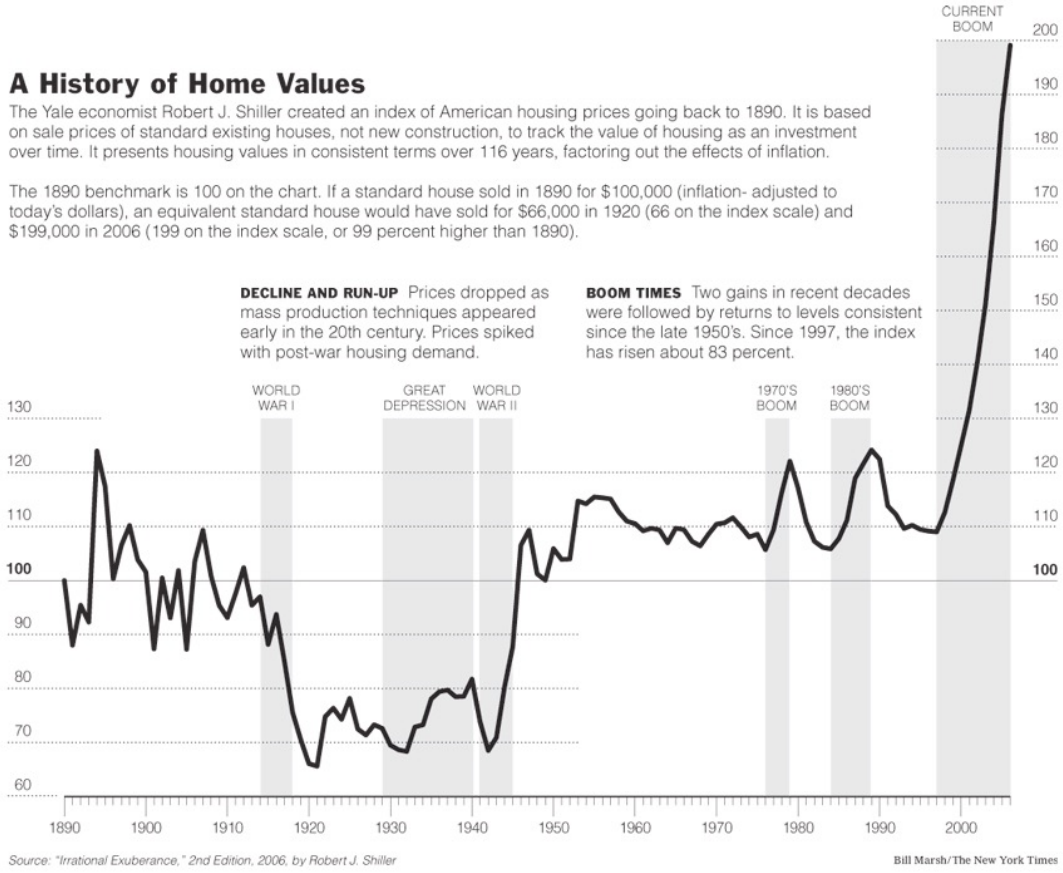
The Business Model Canvas

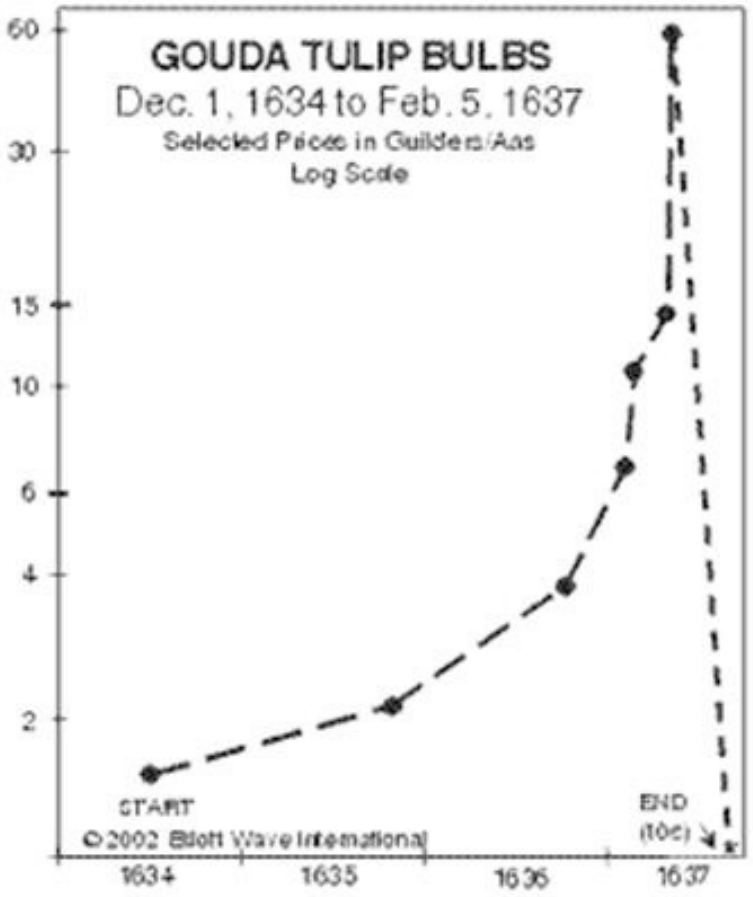


Example framework that puts things into context: Porter's 5 Forces

Source: Michael Porter, "The five competitive forces that shape strategy," HBR, Jan 2008.

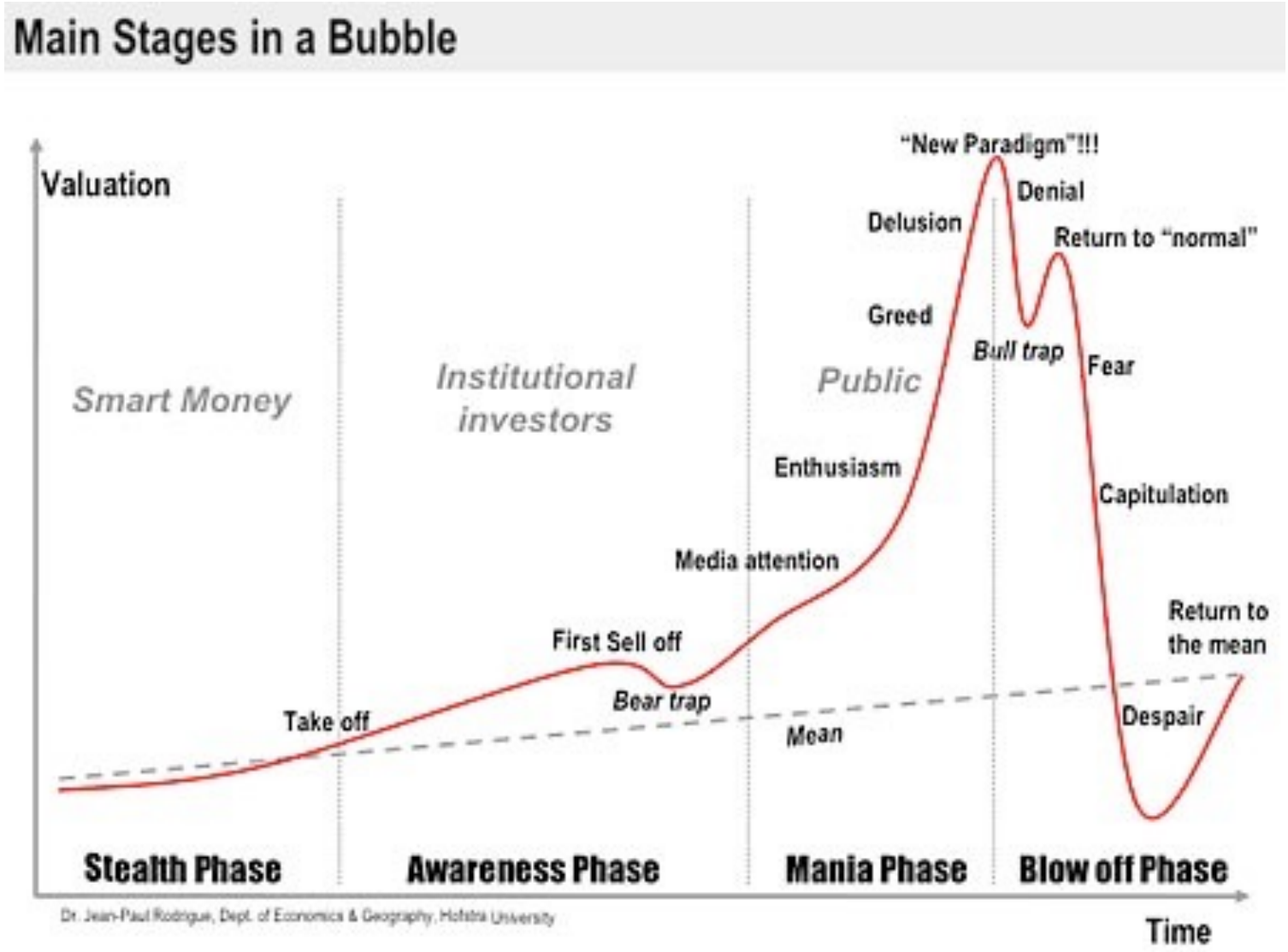
Hypes were and will always be present





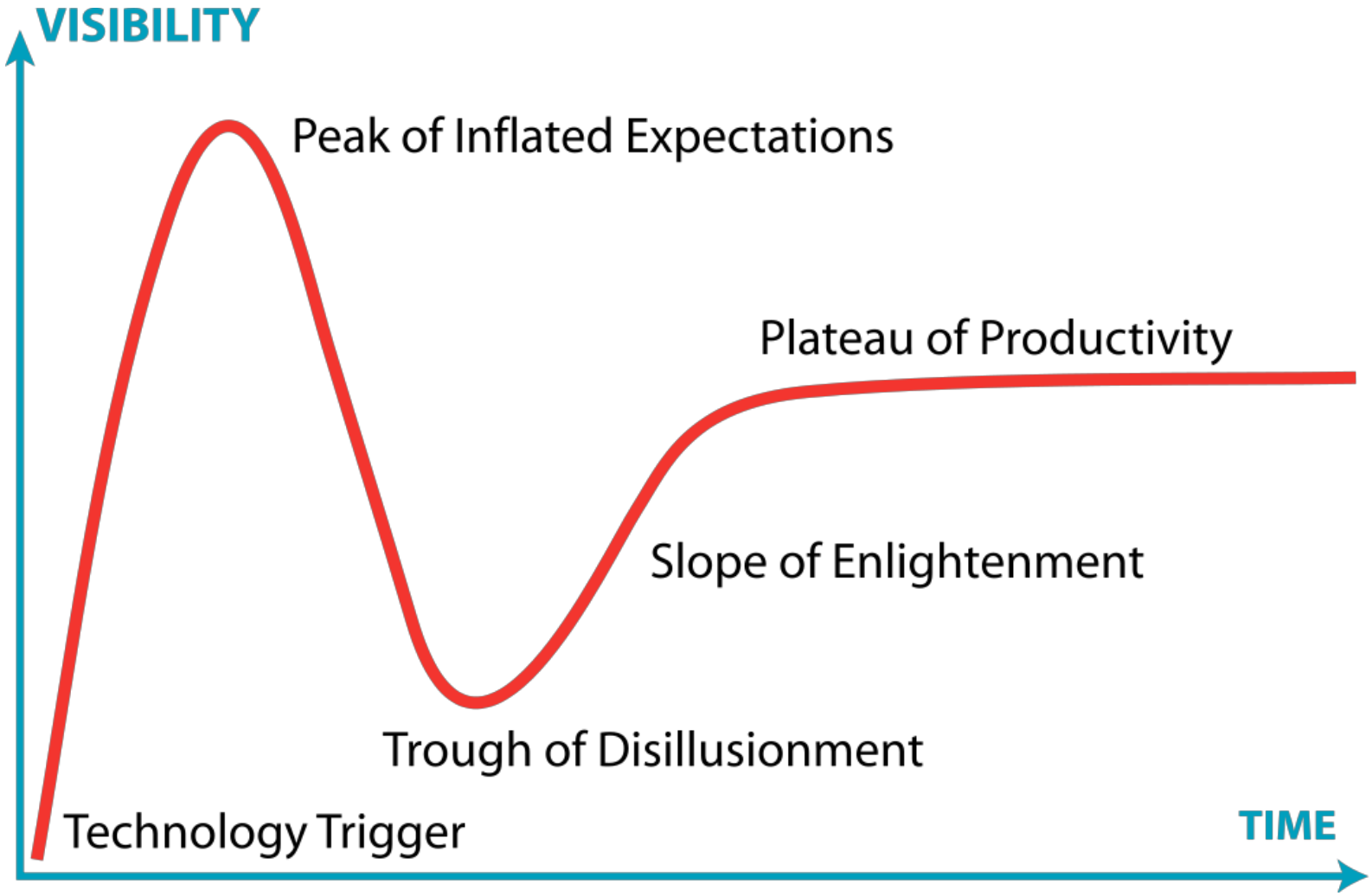
Source: http://www.accumulatingmoney.com/real-estate-mania-a-history-of-home-values/, http://www.minyanville.com/assets/Image/tulipmania.jpg
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Hypes were and will always be present



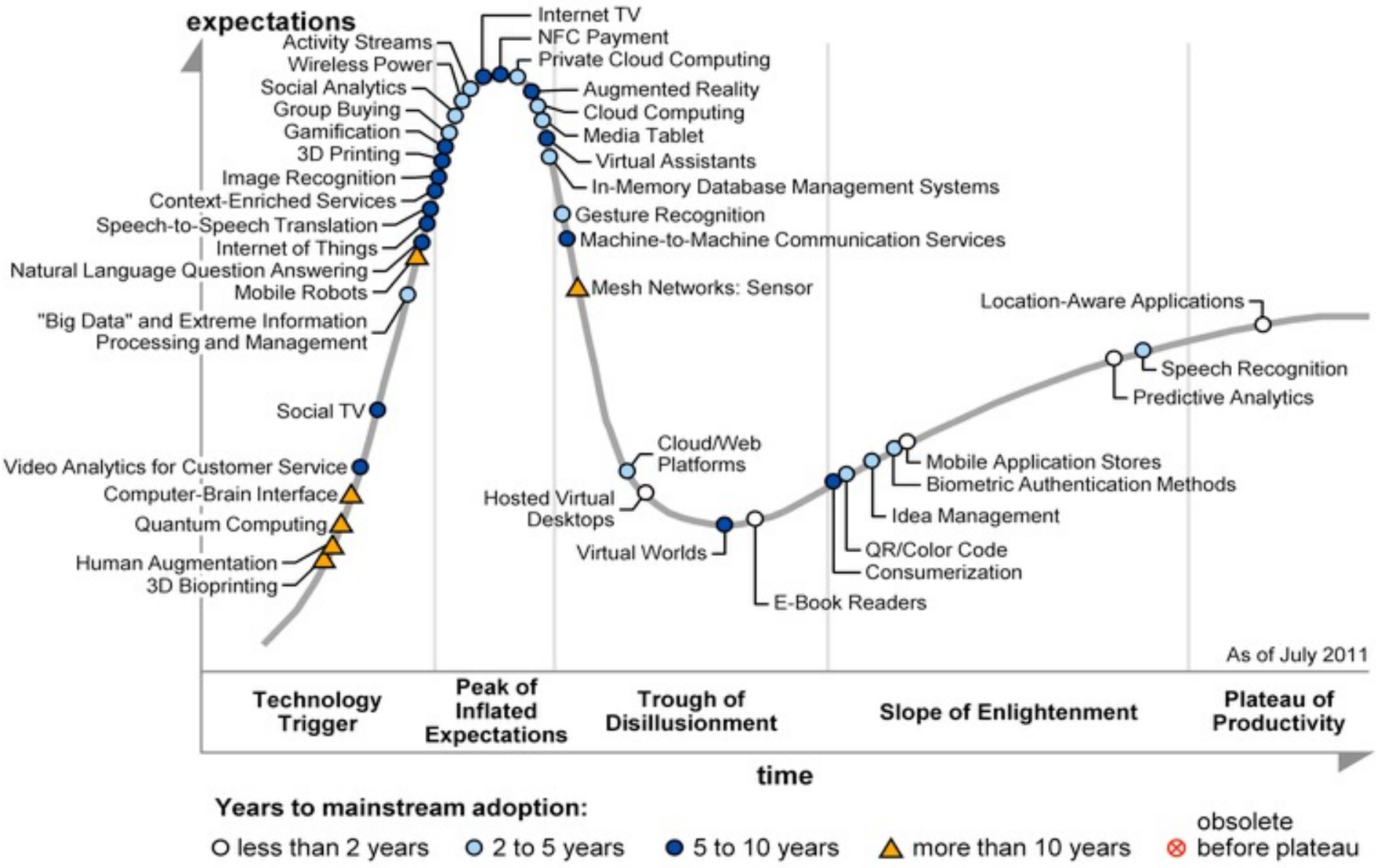
Source: http://glyndk.blogspot.com/2010/04/when-tulip-bubbles-burst.html

If you use this abstraction to analyze technology, you get Gartner's Hype Cycle



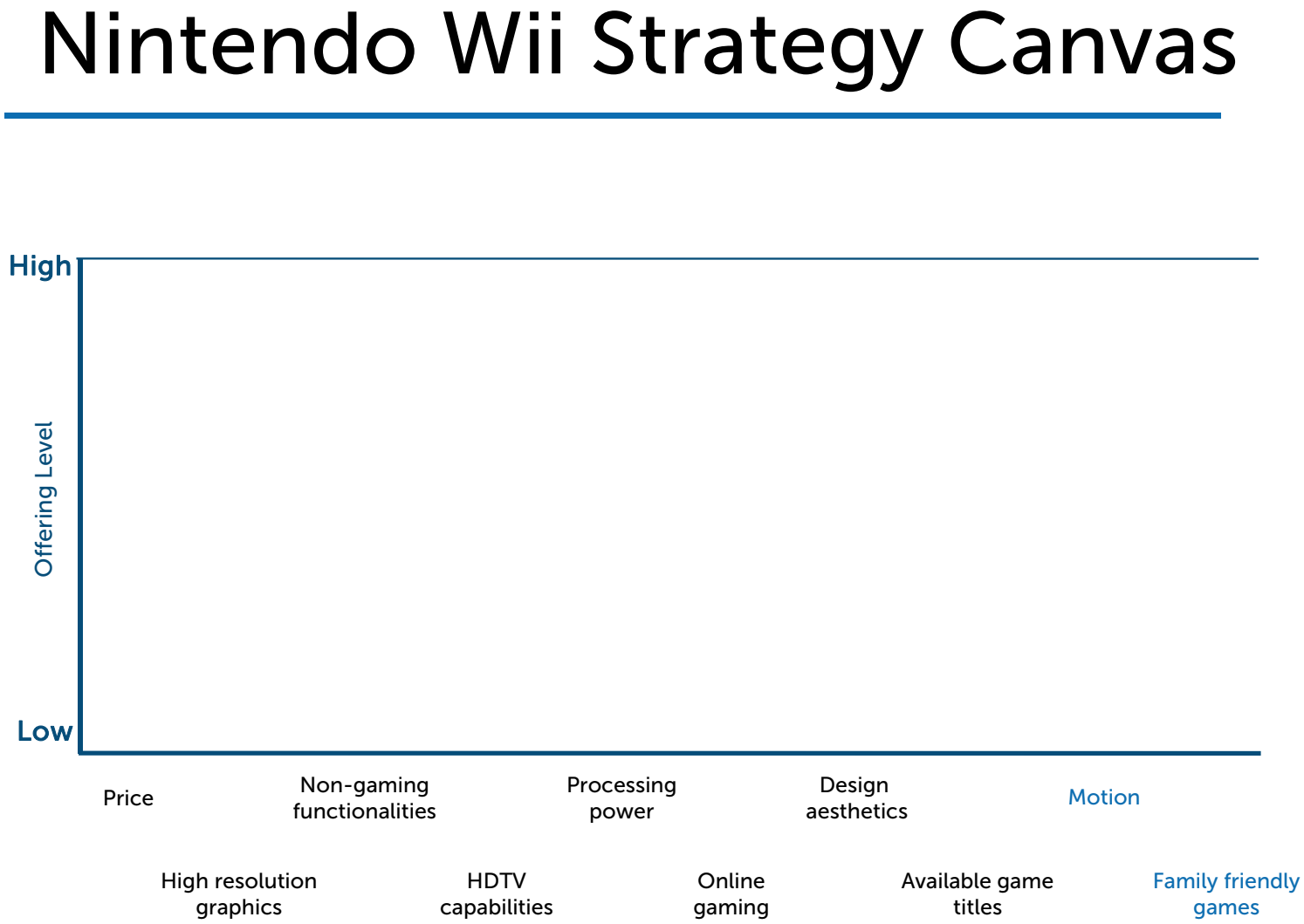
Source: http://upload.wikimedia.org/wikipedia/commons/9/94/Gartner_Hype_Cycle.svg

Hype Cycle for Emerging Technologies 2011



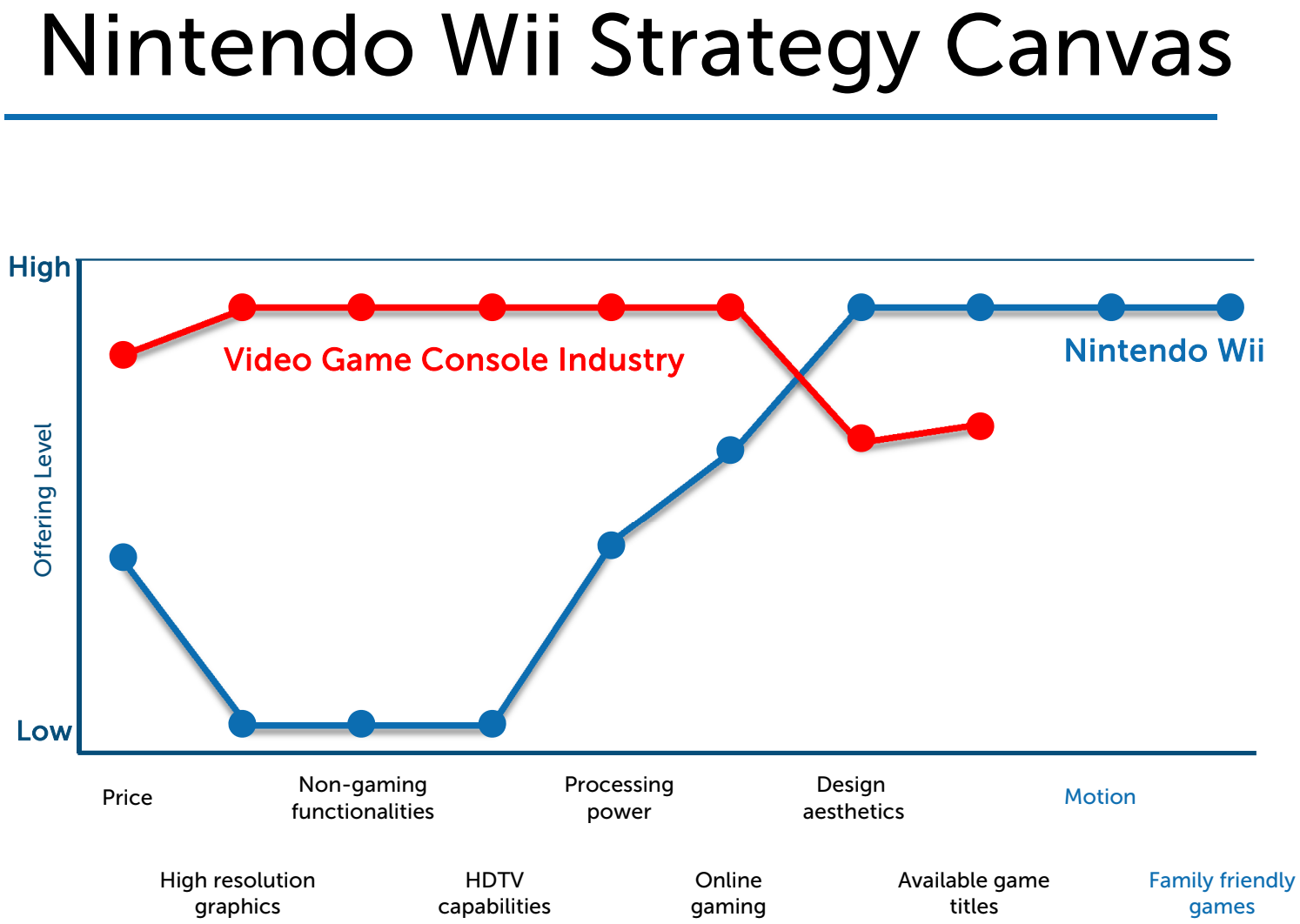
Source: Hype Cycle for Emerging Technologies 2011, Gartner (July 2011), http://www.gartner.com/it/page.jsp?id=1763814
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Strategy Canvas: Example Nintendo Wii (1/3)



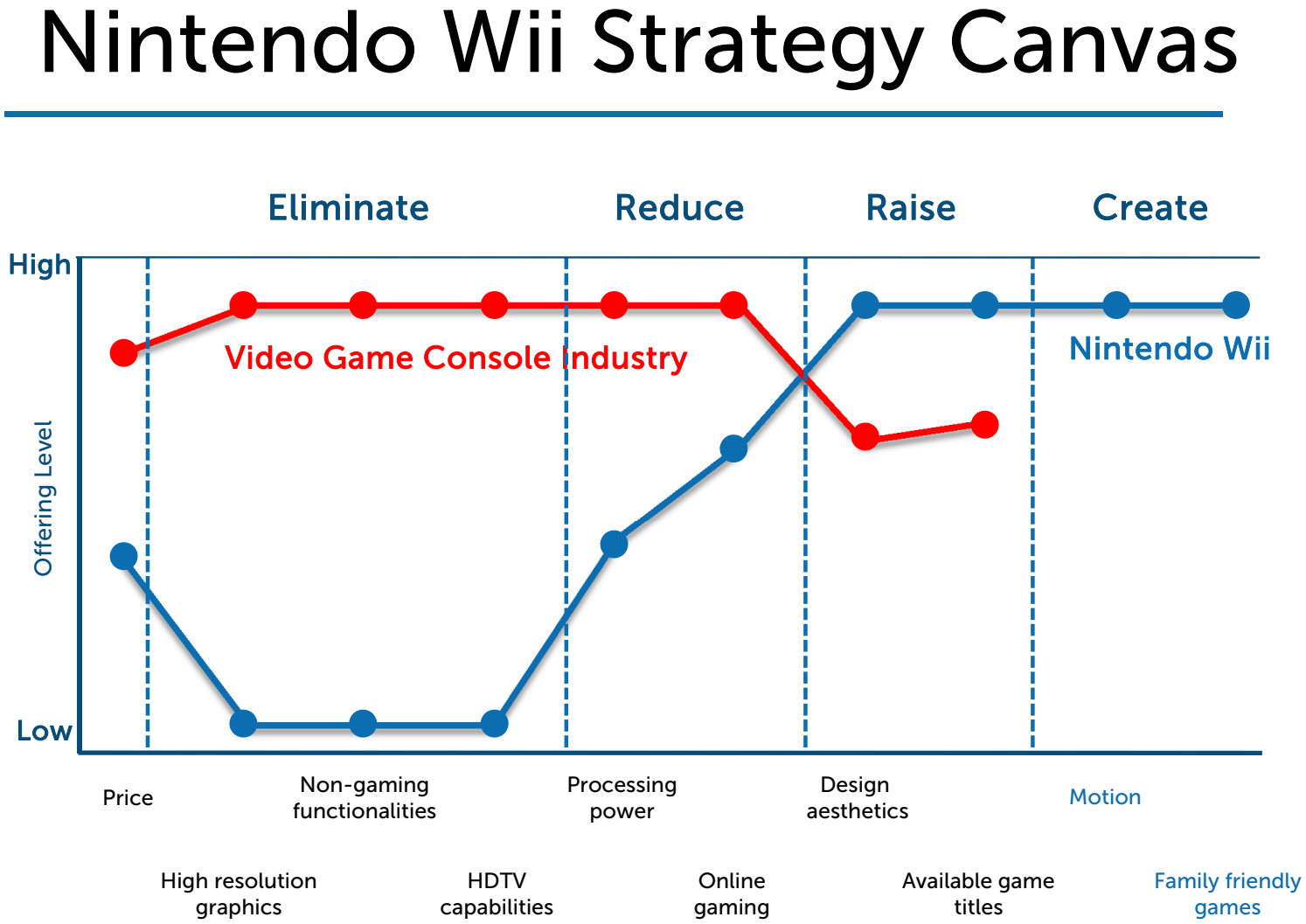
Source: INSEAD, Blue Ocean Strategy Institute, 2013.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Strategy Canvas: Example Nintendo Wii (2/3)



Source: INSEAD, Blue Ocean Strategy Institute, 2013.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

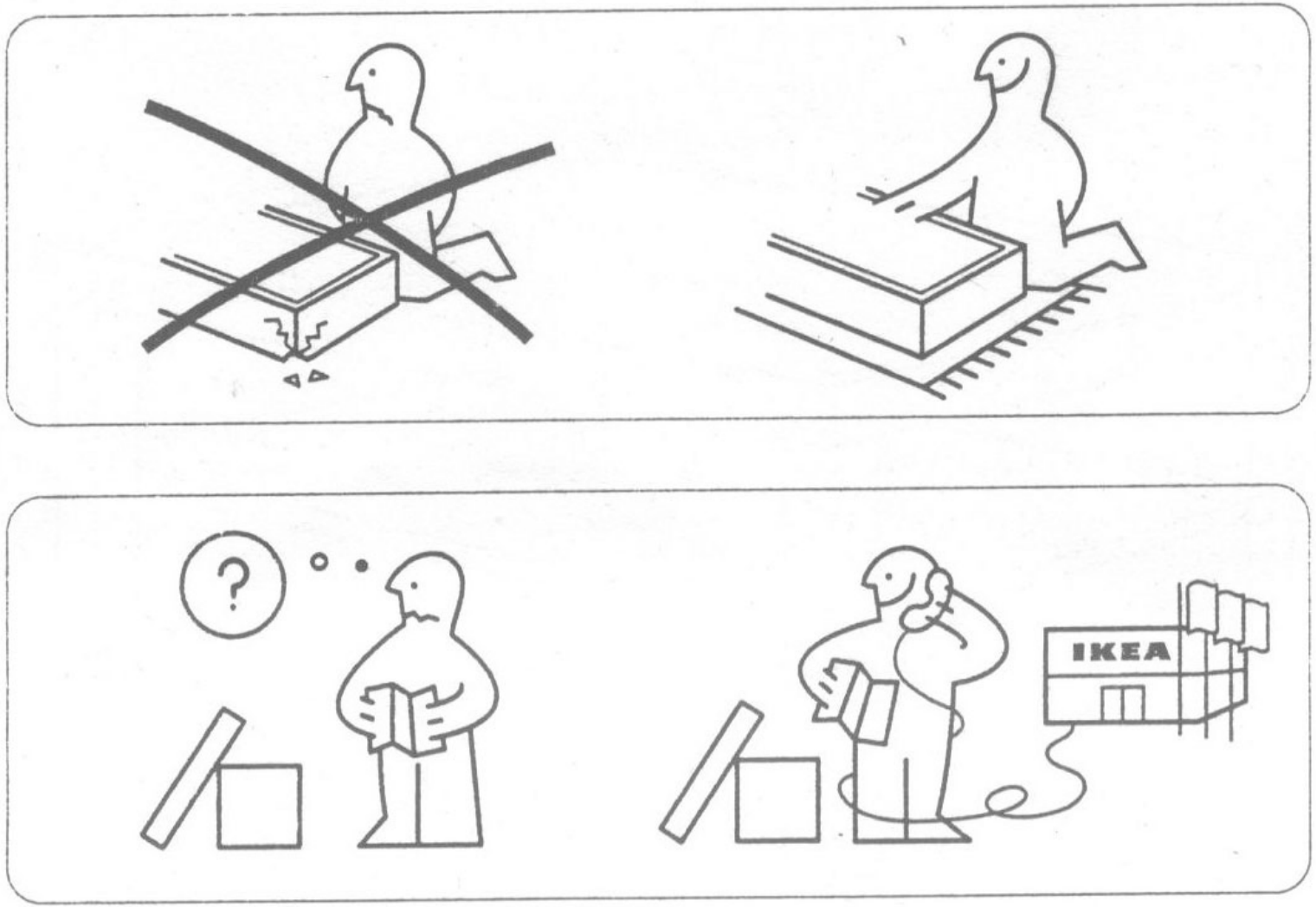
Strategy Canvas: Example Nintendo Wii (3/3)



Source: INSEAD, Blue Ocean Strategy Institute, 2013.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

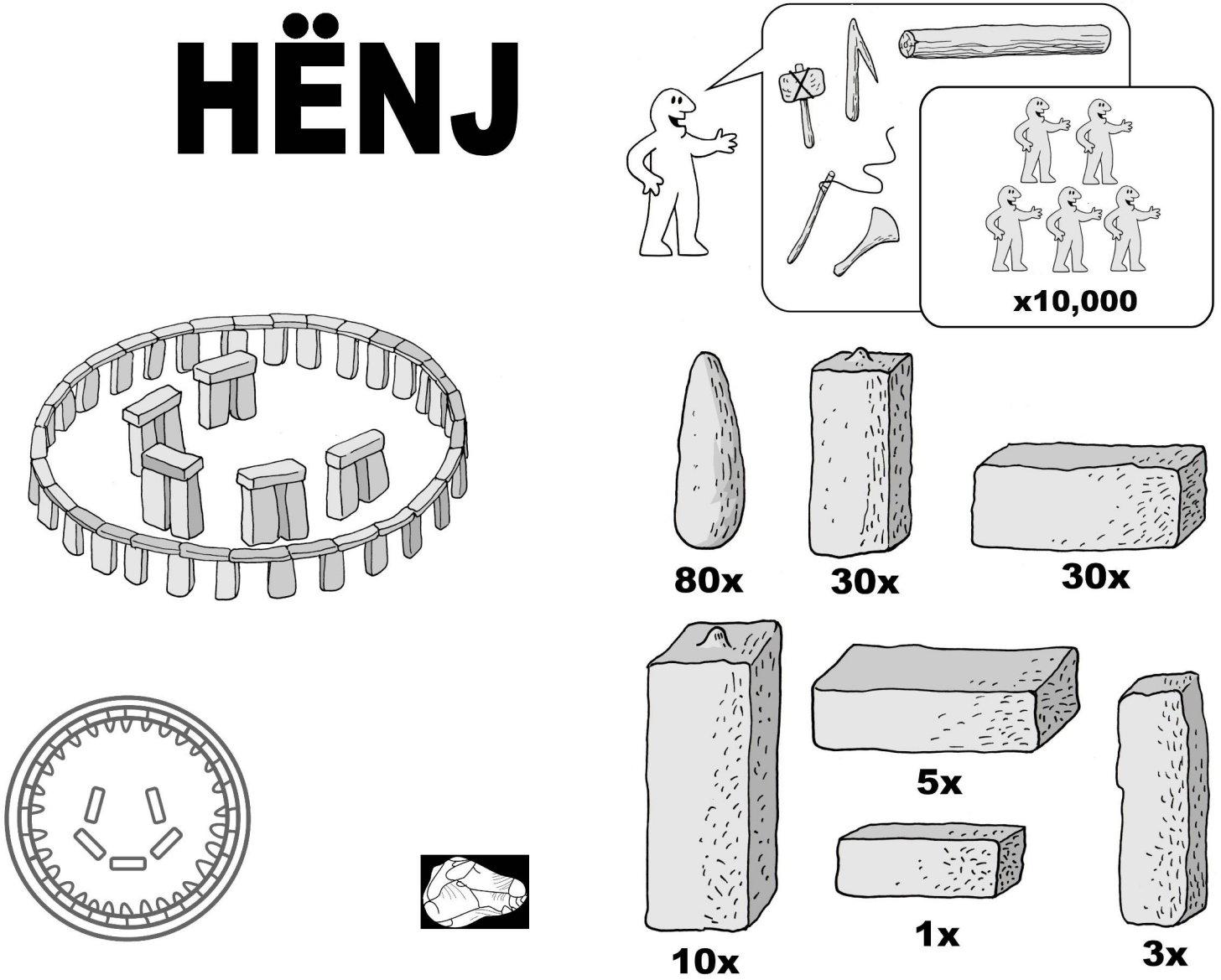
"Redefine the Market"

Pictograms for "complex instructions"



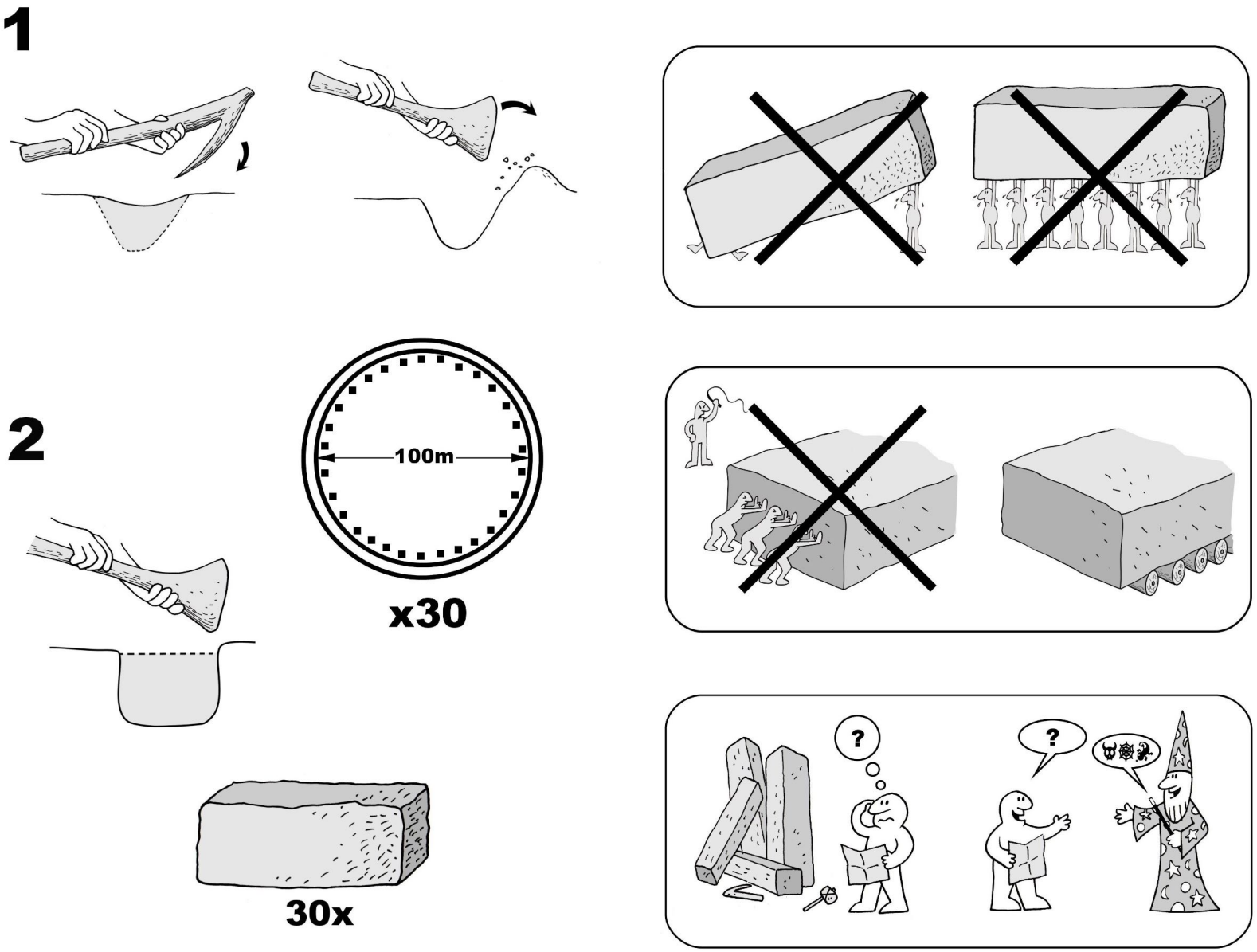
Source: Ikea

IKEA kits have many components



Source: Sergei Maslov, "Why bacteria run Linux while eukaryotes run Windows?" citing Justin Pollard, http://www.designboom.com
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

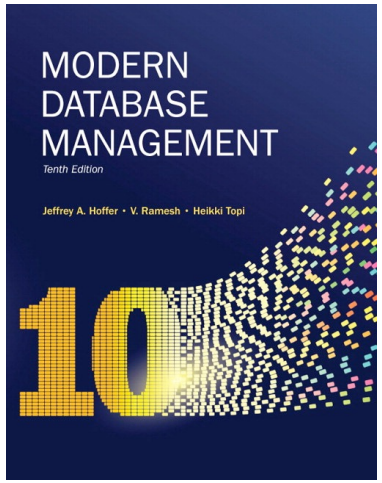
They need to be assembled to work



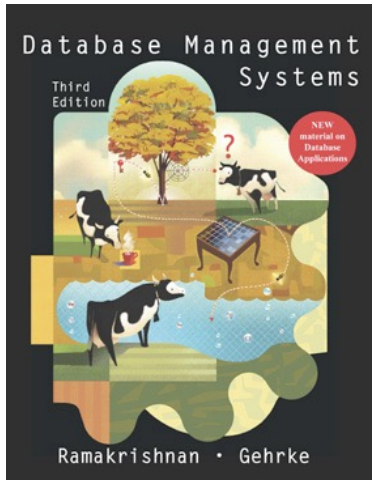
Source: Sergei Maslov, "Why bacteria run Linux while eukaryotes run Windows?" citing Justin Pollard, http://www.designboom.com
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Some comments on Notations

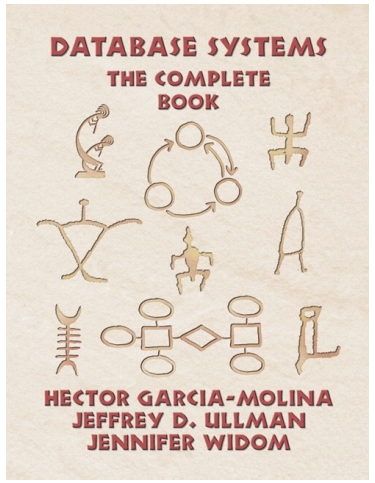
Different sources, different notations



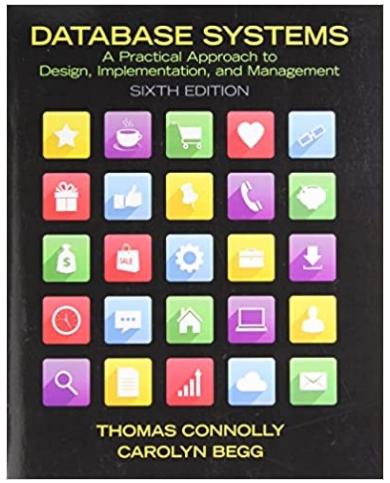
[Hoffer+'10]



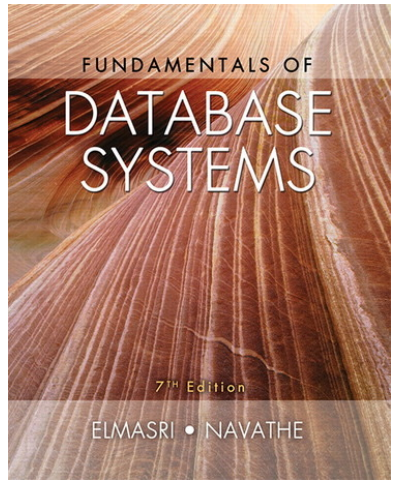
[Cow book'03]



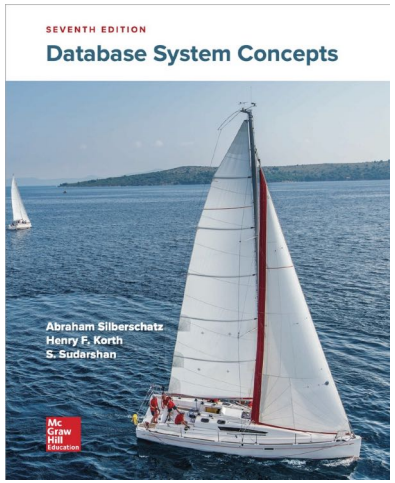
[Stanford book'08]



[Connolly+'15]



[Elmasri+'15]



[Silberschatz+'20]

Crow foot

SDK arrows

[Hoffer+'10]: Hoffer, Ramesh, Topi. Modern Database Management, 10th ed, 2010. https://www.pearson.com/us/higher-education/product/Hoffer-Modern-Database-Management-10th-Edition/9780136088394.html

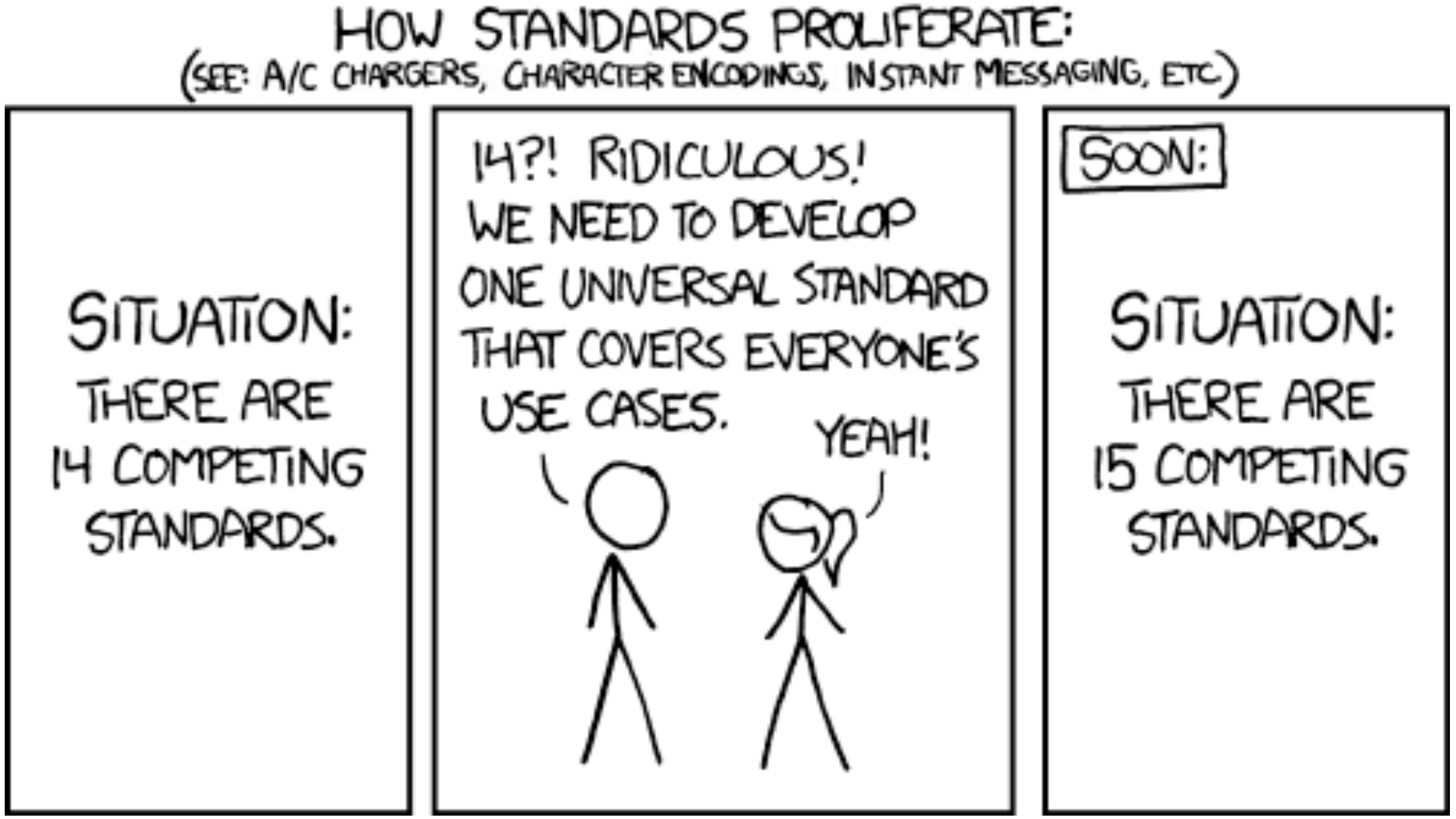
[Cow book'03]: Ramakrishnan, Gehrke, Database Management Systems, 3rd ed, 2003. http://pages.cs.wisc.edu/~dbbook/

[Stanford book'08]: Garcia-Molina, Ullman, Widom. Database Systems: The Complete Book, 2nd ed, 2008. http://infolab.stanford.edu/~ullman/dscb.html

[Connolly+'15]: Connolly, Begg. Database systems: A practical approach to design, implementation, and management, 6th ed, 2015. https://www.pearson.com/us/higher-education/program/Connolly-Database-Systems-A-Practical-Approach-to-Design-Implementation-and-Management-6th-Edition/PGM116956.html

[Elmasri+'15]: Elmasri, Navathe. Fundamentals of Database Systems, 7th ed, 2015. https://www.pearson.com/us/higher-education/program/Elmasri-Fundamentals-of-Database-Systems-7th-Edition/PGM189052.html

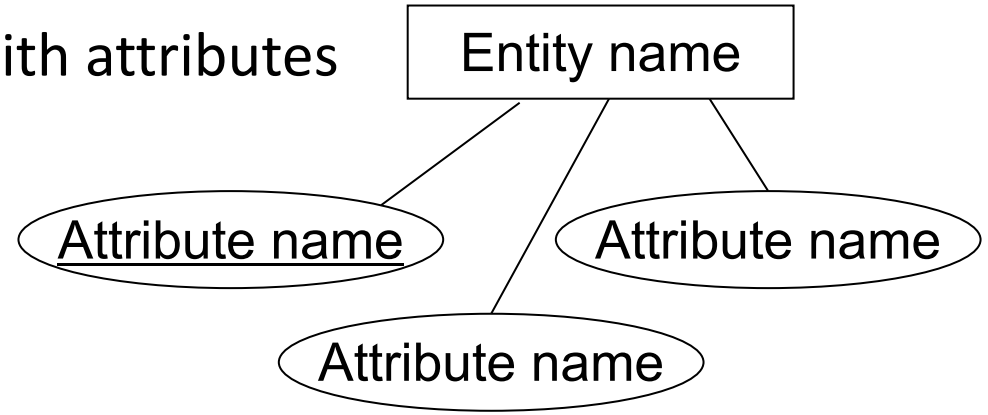
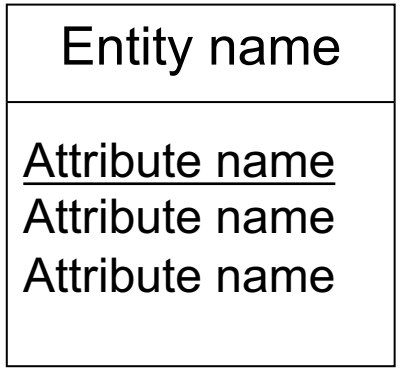
[Silberschatz+'20]: Silberschatz, Korth, Sudarshan. Database system concepts, 7th ed, 2020. https://www.db-book.com/db7



Source and further pointers: https://xkcd.com/927/, https://www.explainxkcd.com/wiki/index.php/927:_Standards
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

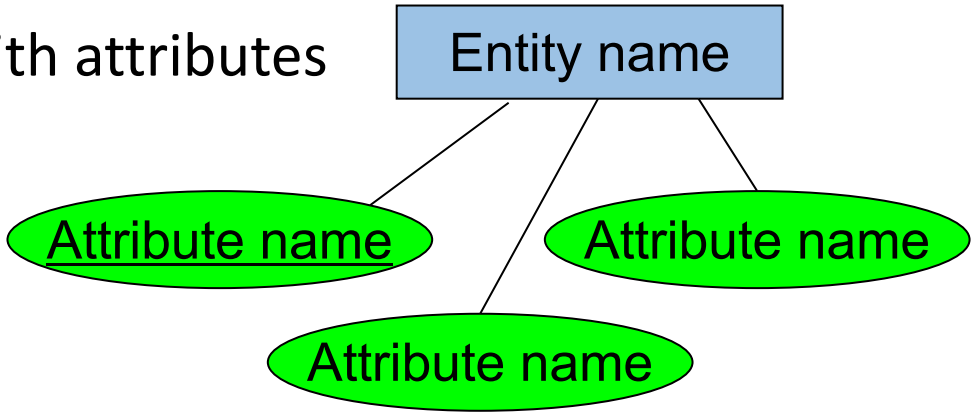
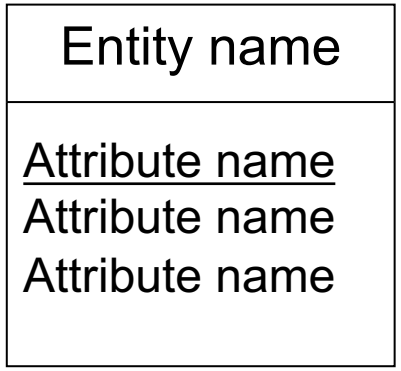
Comparison of ERD frameworks

A variant of "UML"

	Chen's	Crow's Foot
Strong entity		
Weak entity		
Entity with attributes		

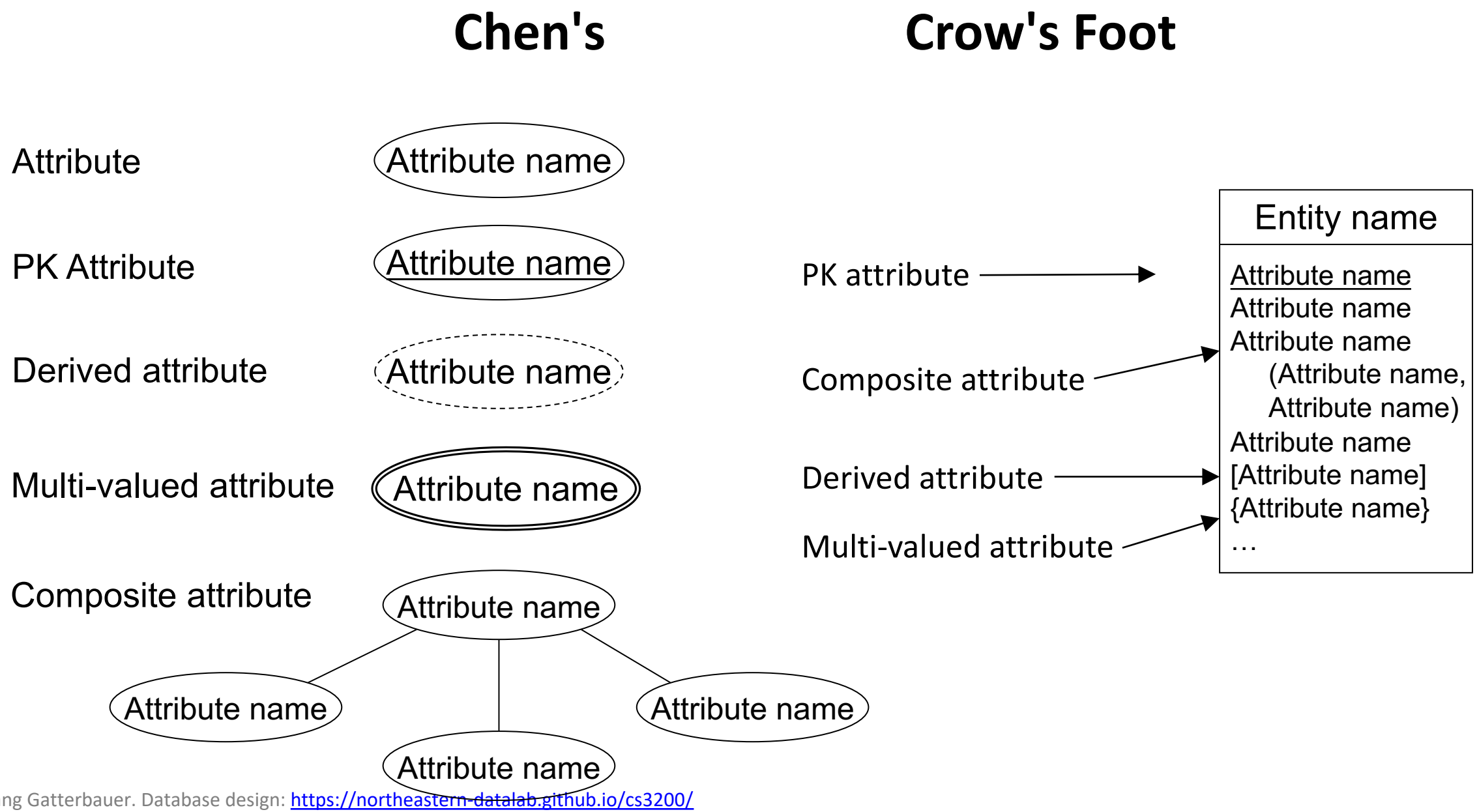
Comparison of ERD frameworks

A variant of "UML"

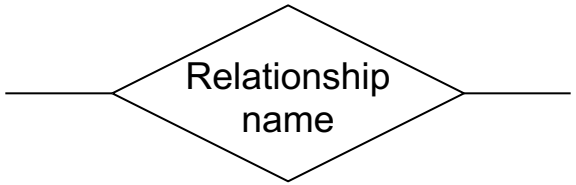
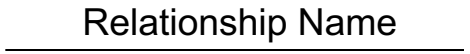
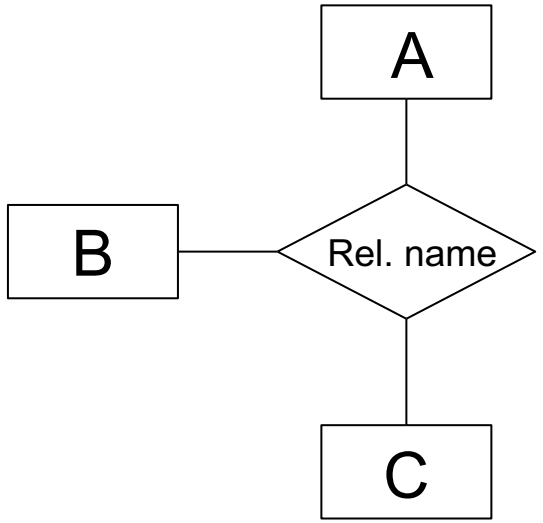
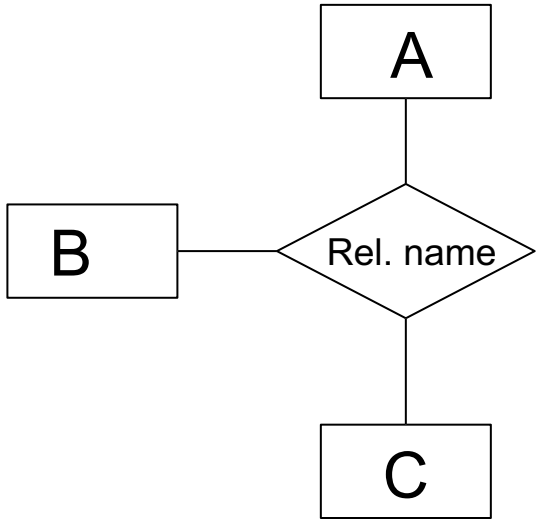
	Chen's	Crow's Foot
Strong entity		
Weak entity		
Entity with attributes		

Color is not part of the standard…

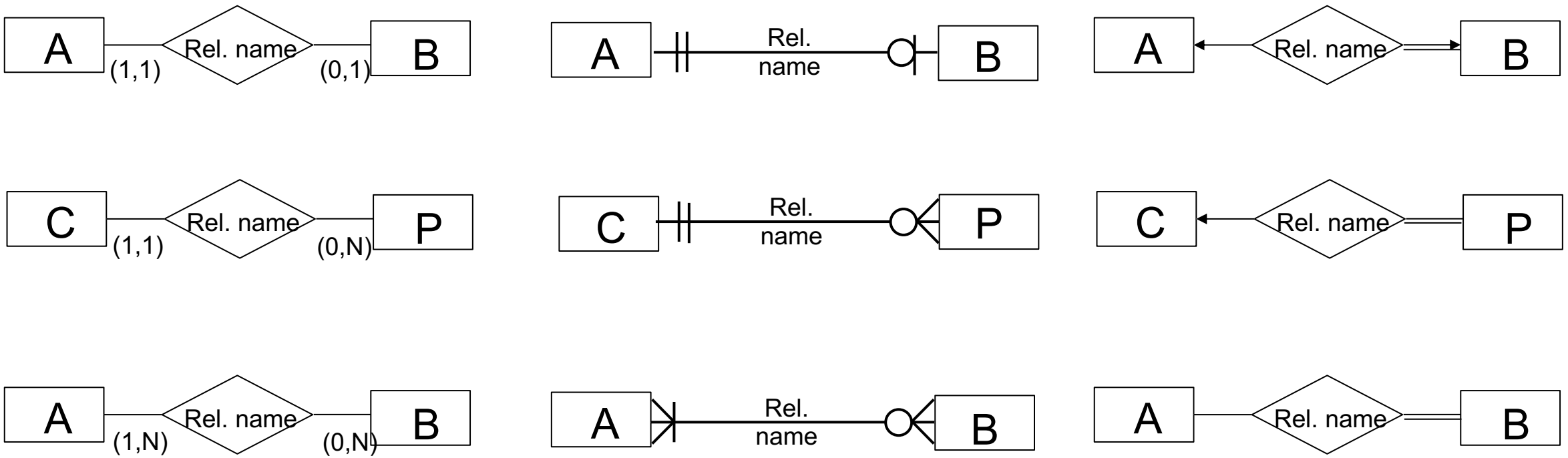
Attributes


Chen's
Crow's Foot
Attribute
Attribute name
PK Attribute
Attribute name
Derived attribute
Attribute name
Multi-valued attribute
Attribute name
Composite attribute
Attribute name
Attribute name
Attribute name
Attribute name
PK attribute
Composite attribute
Derived attribute
Multi-valued attribute
Entity name
Attribute name
Attribute name
Attribute name
(Attribute name,
Attribute name)
Attribute name
[Attribute name]
{Attribute name}
…

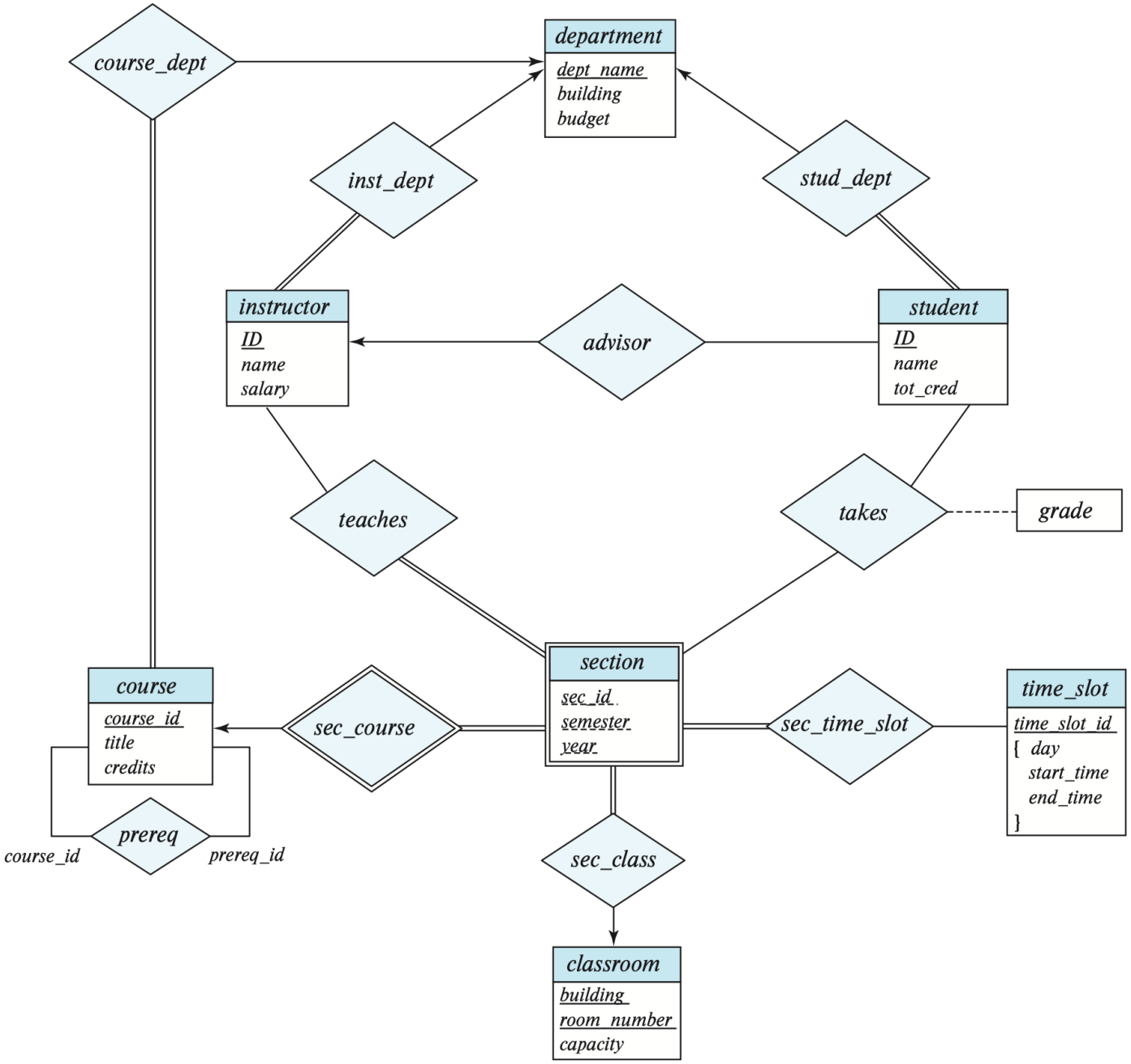
Relationships

	Chen's	Crow's Foot
Binary Relationship		
Relationship of Higher Degree		

Types of Binary Relationships

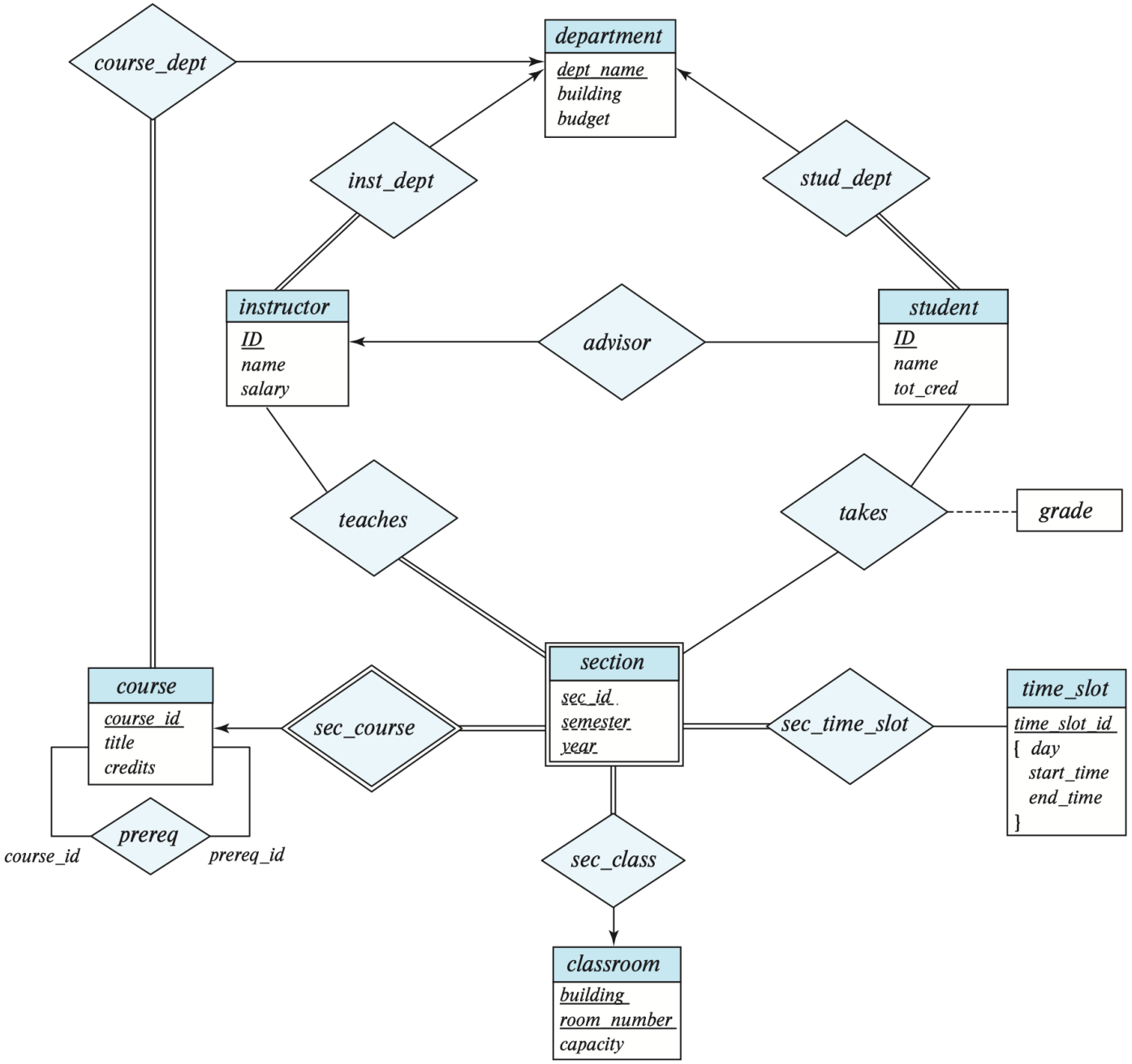


Let's read an example ERD



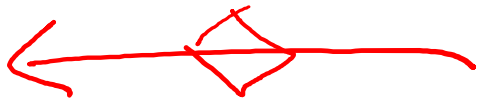
Source: "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 7th ed, 2020. Fig 6.15
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example ERD for a University Enterprise



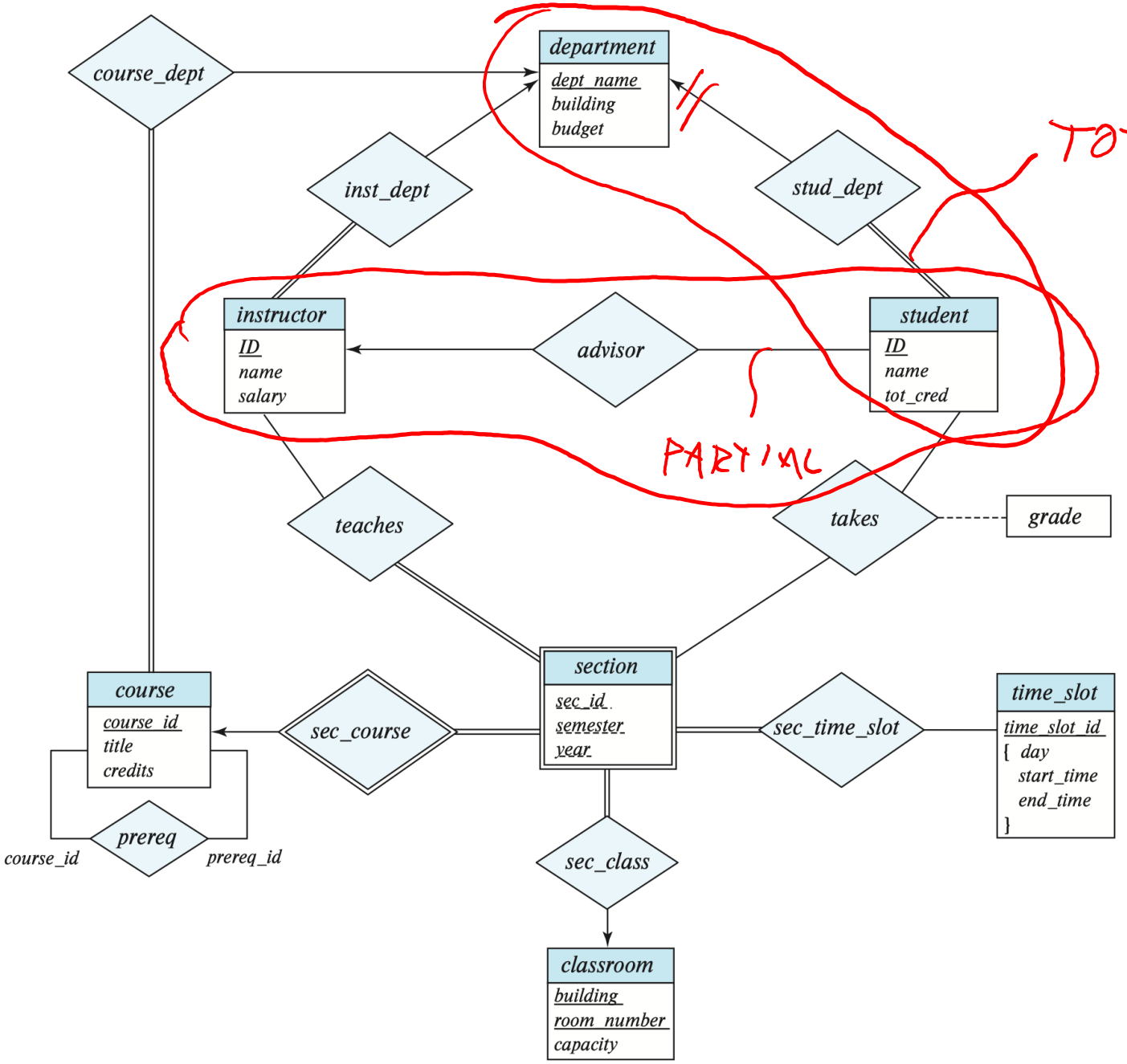
Source: "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 7th ed, 2020. Fig 6.15
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Let's read an example ERD



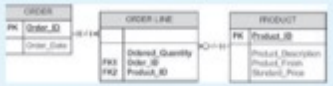
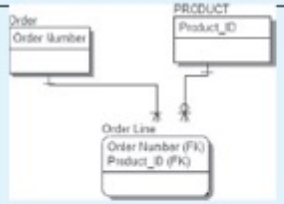
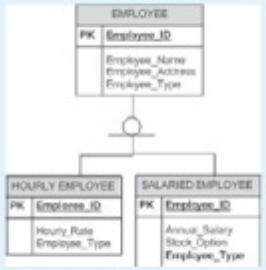
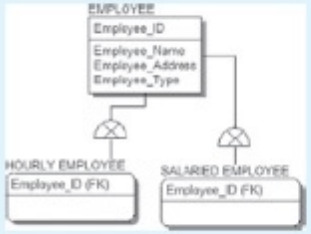
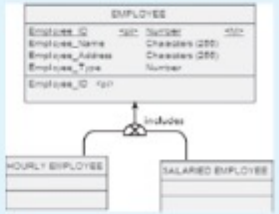
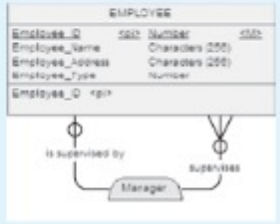
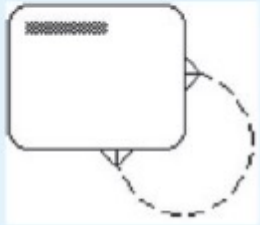
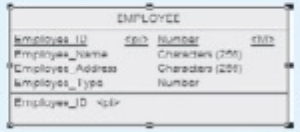
optional (partial)

mandatory (total)



Source: "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 7th ed, 2020. Fig 6.15
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Modeling Notation ("Consistency beats brilliance")

	Hoffer-Ramesh-Topi Notation	Visio PRO 2003	CA ERWin Data Modeler r7.3	Sybase PowerDesigner 15	Oracle Designer 10g
Basic Entity	Strong / Weak	EMPLOYEE	EMPLOYEE	EMPLOYEE	PRODUCT LINE
Associative Entity	Associative				(No special symbol. Uses regular Entity symbol.)
Subtypes	EMPLOYEE; HOURLY EMPLOYEE; SALARIED EMPLOYEE				SUPERTYPE; SUBTYPE A; SUBTYPE B
Recursive Relationship	Manages; EMPLOYEE	supervises / is supervised by; EMPLOYEE: Employee_ID, Employee_Name, Employee_Address, Employee_Type	EMPLOYEE: Employee_ID, Employee_Name, Employee_Address, Employee_Type		
Attributes	ENTITY NAME; Identifier; Partial Identifier; Optional; [Derived]; {Multivalued}; Composite(,)	EMPLOYEE: PK Employee_ID; Employee_Name, Employee_Address, Employee_Type	EMPLOYEE: Employee_ID; Employee_Address, Employee_Name		PRODUCT LINE; # PRODUCT_LINE_ID; * PRODUCT_LINE_NAME

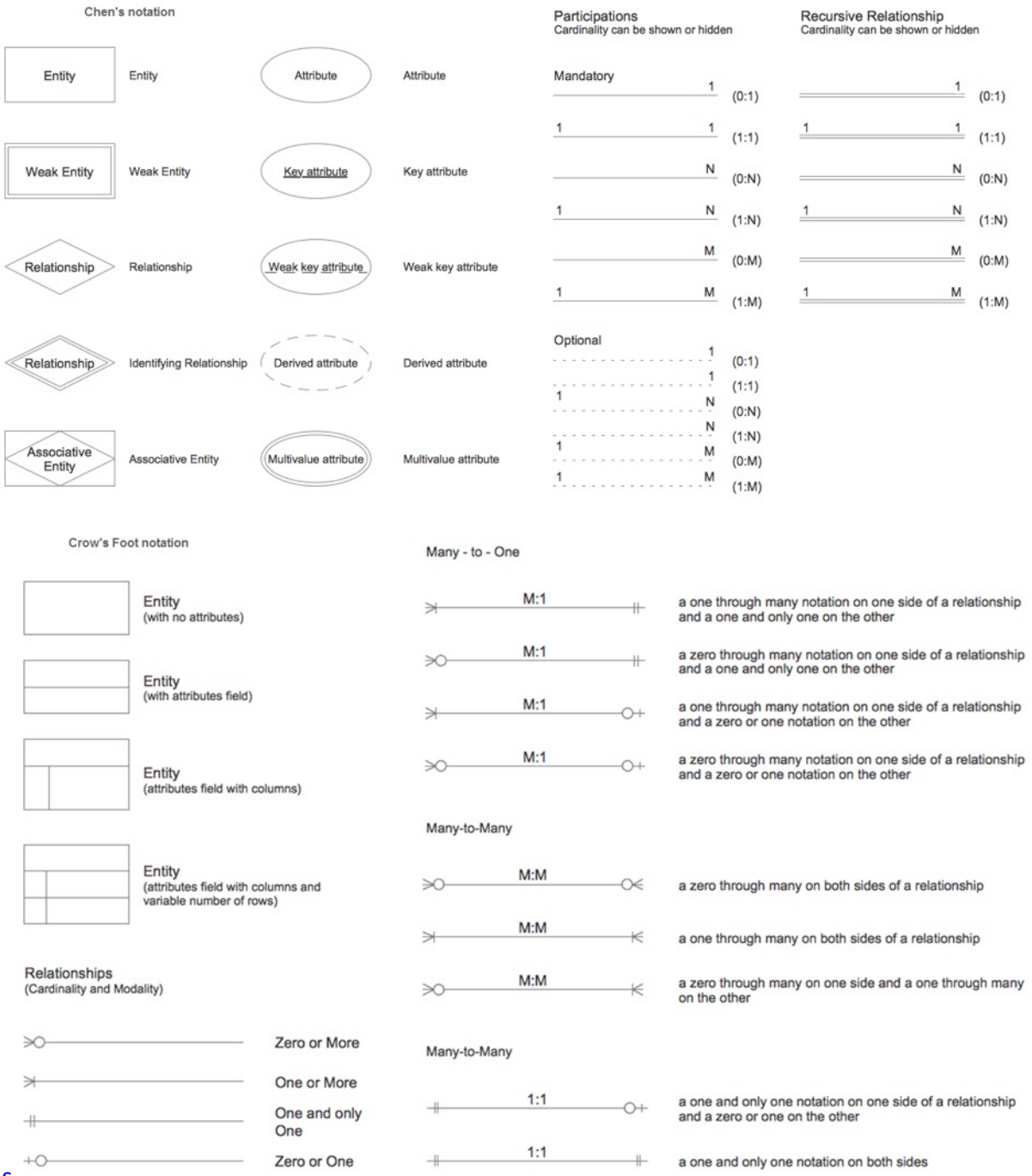
Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010., Table A-1
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Modeling Cardinality/Optionality Notations

	Hoffer-Ramesh-Topi Notation	*Visio PRO 2003*	*CA ERWin Data Modeler r7.3*	*Sybase PowerDesigner 15*	*Oracle Designer 10g*
1:1		(Not available without cardinality)	(Not available without cardinality)	0,1 0,1	
1:M		(Not available without cardinality)	(Not available without cardinality)	0,1 0,n	
M:N		(Not allowed)		0,n 0,n	
Mandatory 1:1			1		
Mandatory 1:M			P		
Optional 1:M				0,1 0,n	

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010., Table A-1
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

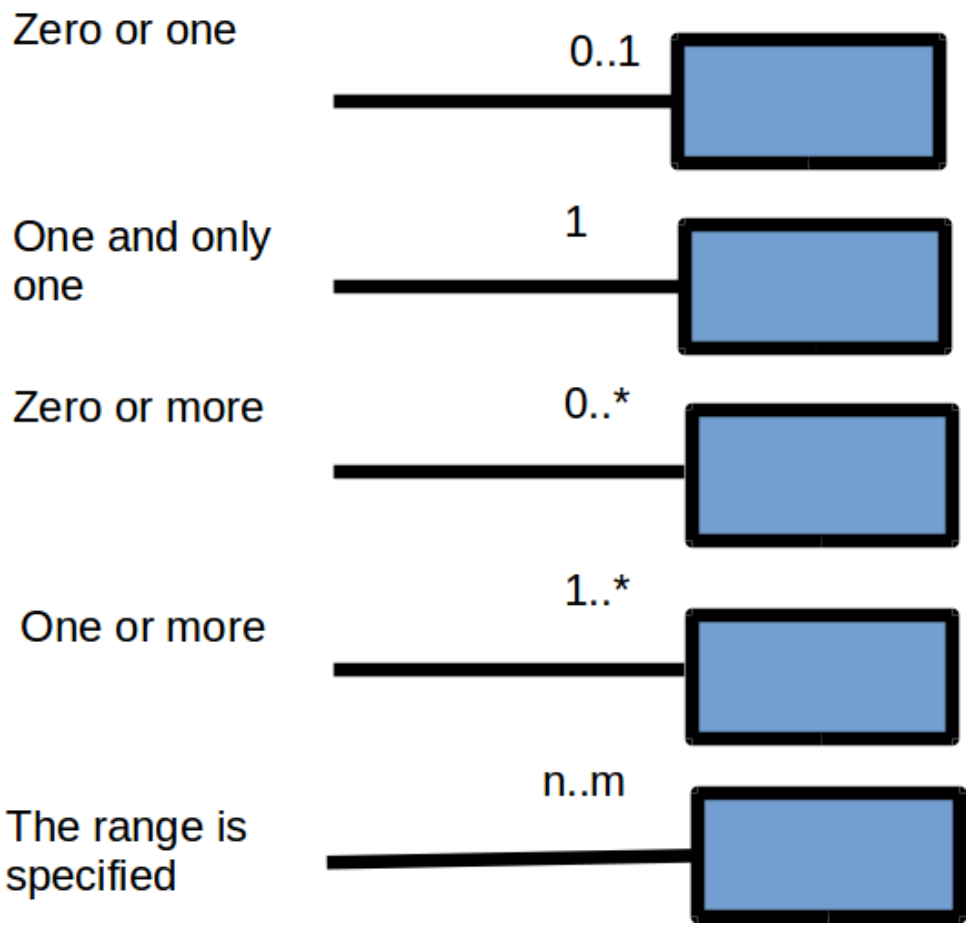
Various Notations



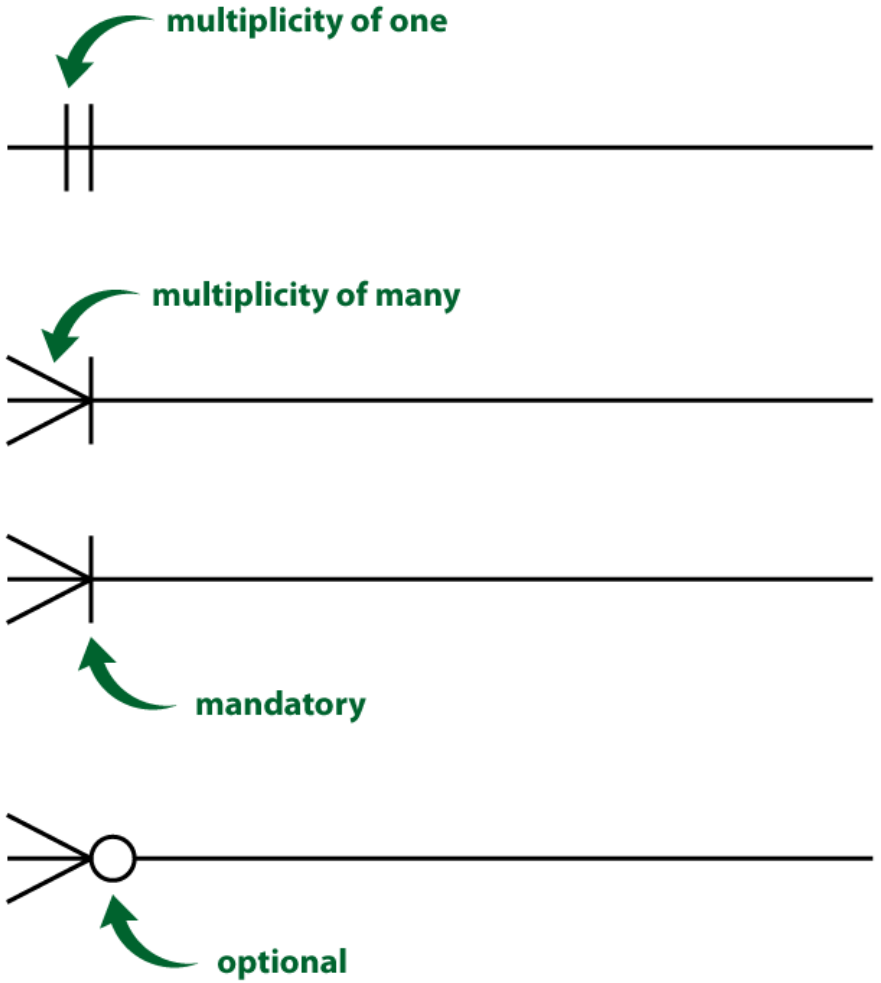
Source: http://www.conceptdraw.com/How-To-Guide/erd-entity-relationship-diagram-symbols
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Relationships with specified cardinalities

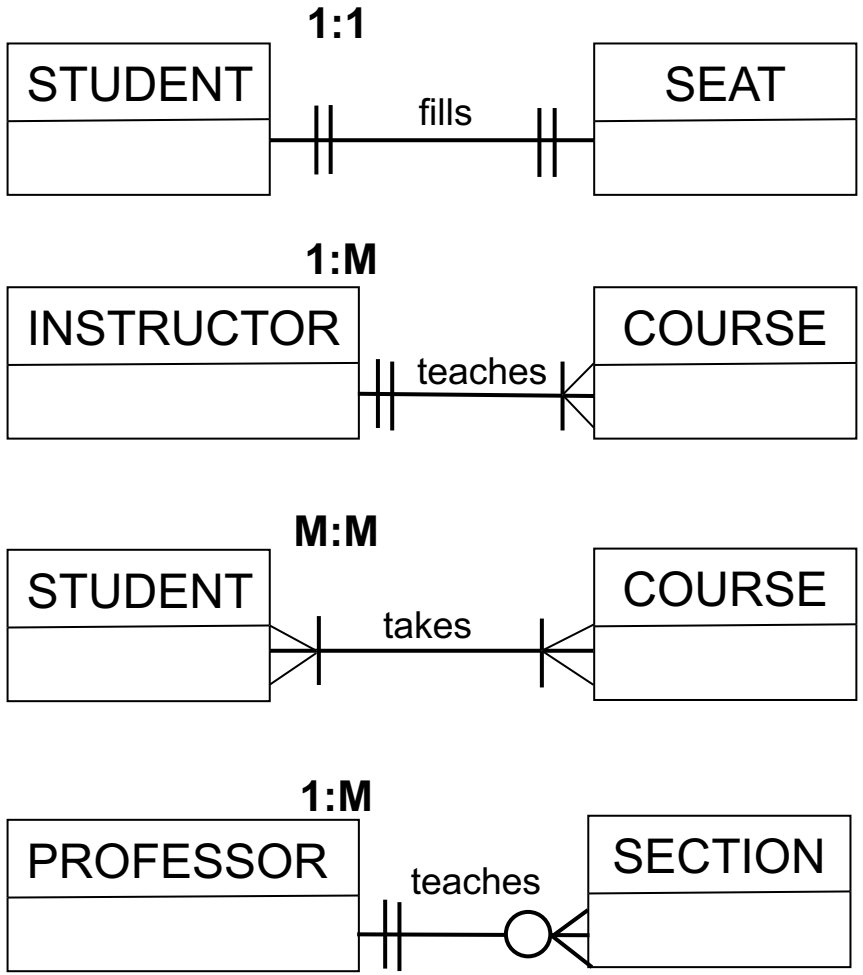
UML notation

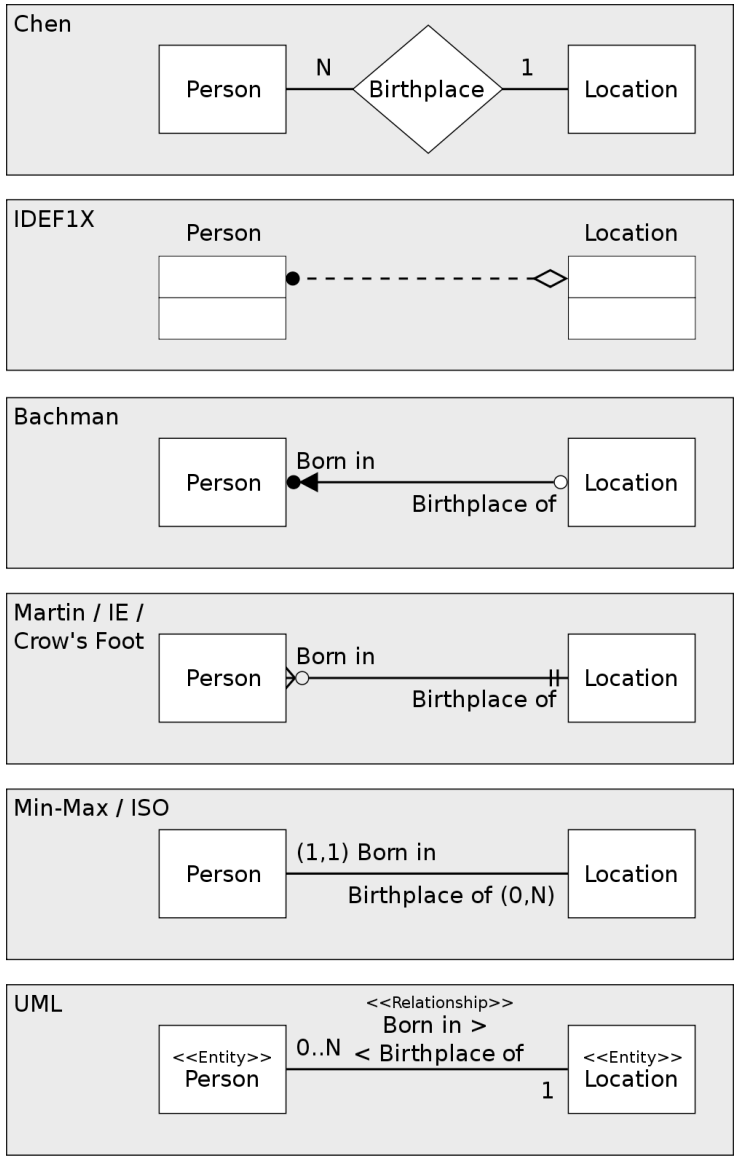


Crow's Foot



Crow's foot notation and alternatives



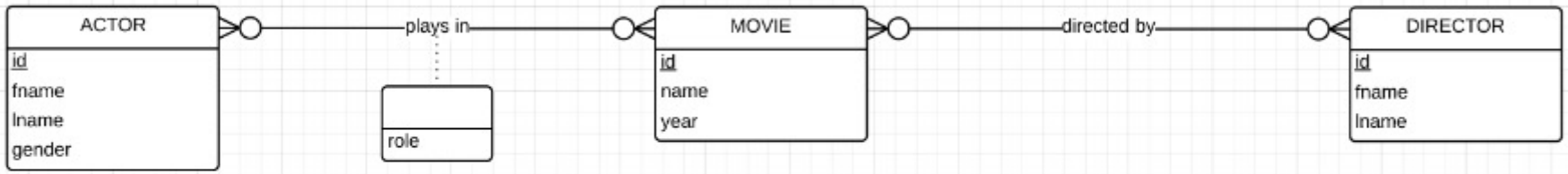


Source: http://en.wikipedia.org/wiki/Entity-relationship_model

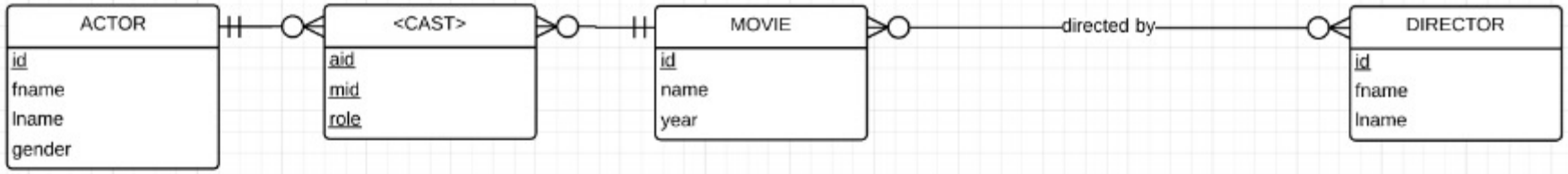
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

IMDB movie database in Lucidchart

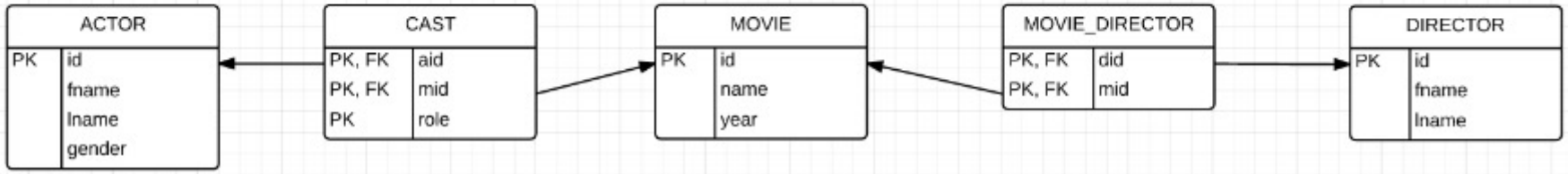
ER diagram: don't forget identifiers, but no FKs



ER diagram: CAST as associative entity can be justified



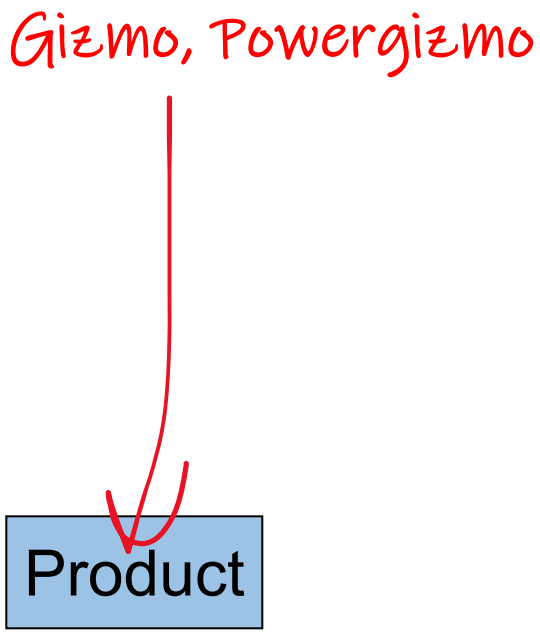
Relational schema: don't forget PKs and FKs



Entities

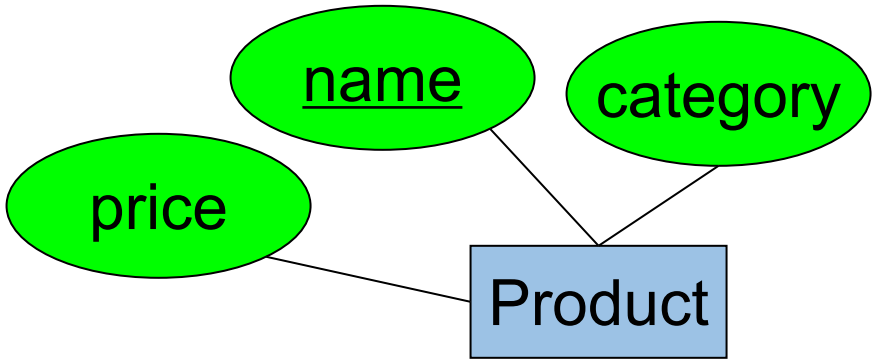
Entities and Entity Sets

- Entities & entity sets are the primitive unit of the E/R model
  - **Entities**: the individual objects, which are members of entity sets
    - Ex: A specific person or product
  - **Entity sets**: the classes or types of objects in our model
    - Ex: Person, Product
    - These are what is shown in E/R diagrams - as rectangles
    - Entity sets represent the sets of all possible entities



Entities and Entity Sets

- An entity set has attributes
  - Represented by ovals attached to an entity set

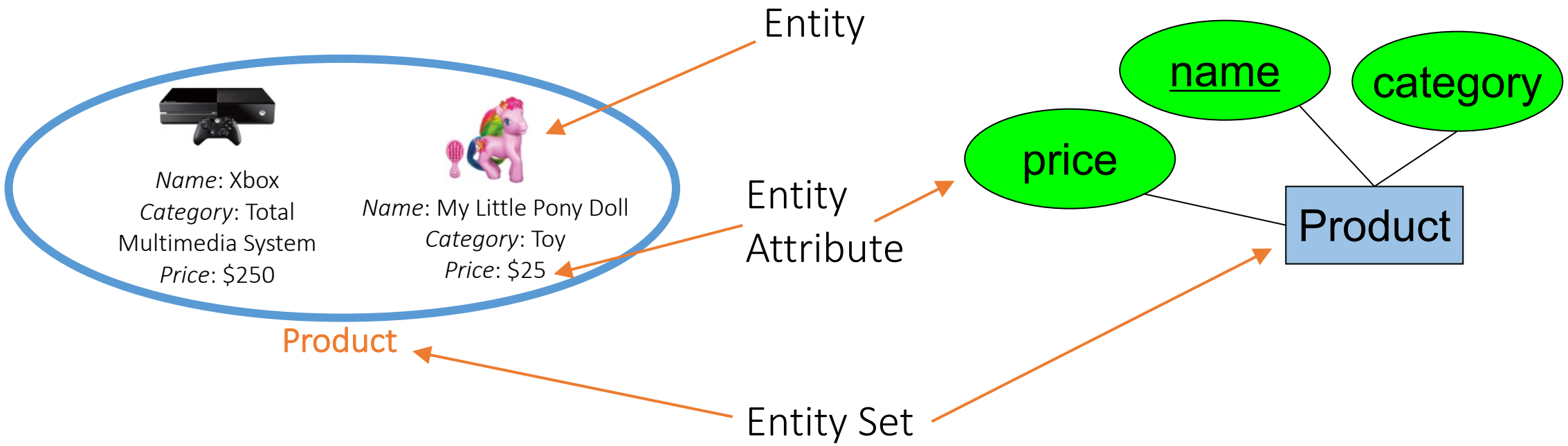


Shapes are important.
Colors are not.

Entities vs. Entity Sets

"Entities" (instances of entity sets) are not explicitly represented in ER diagrams

- Example:

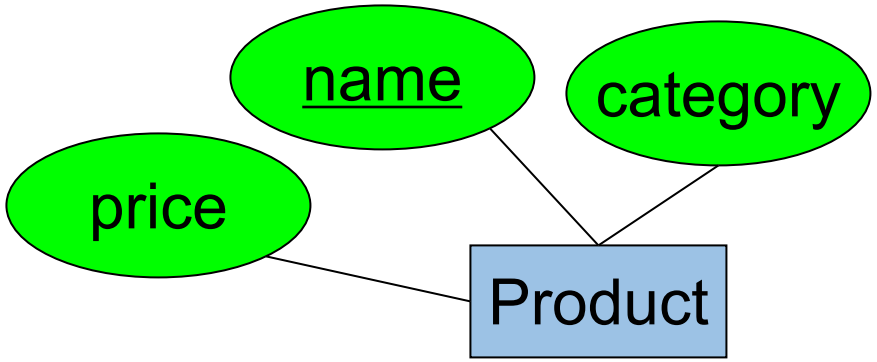


Keys

- A key is a minimal set of attributes that uniquely identifies an entity.

elments of PK (Primary Key)
are underlined

Here, {name, category} is not a key
(it is not minimal).



The ER model forces us to designate a single **primary** key, though there may be multiple candidate keys

Identifiers (Keys)

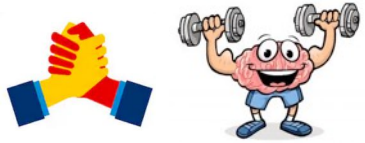
- **Identifier (Key)**: An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type
  - Can be simple or composite
  - Will not be null
  - Will not change in value
    - e.g., family name, or telephone number, or street address are not suited if those can change over time (say through marriage...)
  - Substitute new, simple keys for long, composite keys ("surrogate key")
- **Candidate Key**: an attribute (or set of) that could be a key...satisfies the requirements for being a key
- **Primary Key**: a chosen key

Naming Entities

Poor Examples	Good Examples
FormerStudentFromIowa	Student
Customers	Customer
ClientsWhoCameToBigEvent	Employee
ObscureRecmdForFrtherAction	Invoice
Order	Purchase Order
	Flight

- Guidelines for naming entity types:
  - Use singular nouns
  - Names should be specific to the organization
  - Be concise and consistent
  - Abbreviations are ok, as long as they are standardized
  - Event entity types should be named for the result of the event (e.g., "Purchased", "Registered")

Exercise (Part I): Entities / Attributes



- Assume we want to model "a situation" at Millennium College
- Identify the entities that appear on the report card
- Identify the attributes of each previously identified entity

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

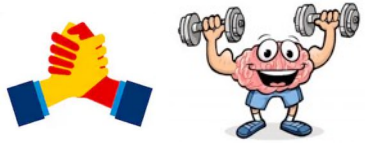
NAME:	Emily Williams	ID: 268300458
CAMPUS ADDRESS:	208 Brooks Hall	
MAJOR:	Information Systems	

COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B

?

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Exercise (Part I): Entities / Attributes



- Assume we want to model "a situation" at Millennium College
- Identify the entities that appear on the report card
- Identify the attributes of each previously identified entity

STUDENT

COURSE

INSTRUCTOR

MILLENNIUM COLLEGE
GRADE REPORT
FALL SEMESTER 200X

NAME: Emily Williams ID: 268300458
CAMPUS ADDRESS: 208 Brooks Hall
MAJOR: Information Systems

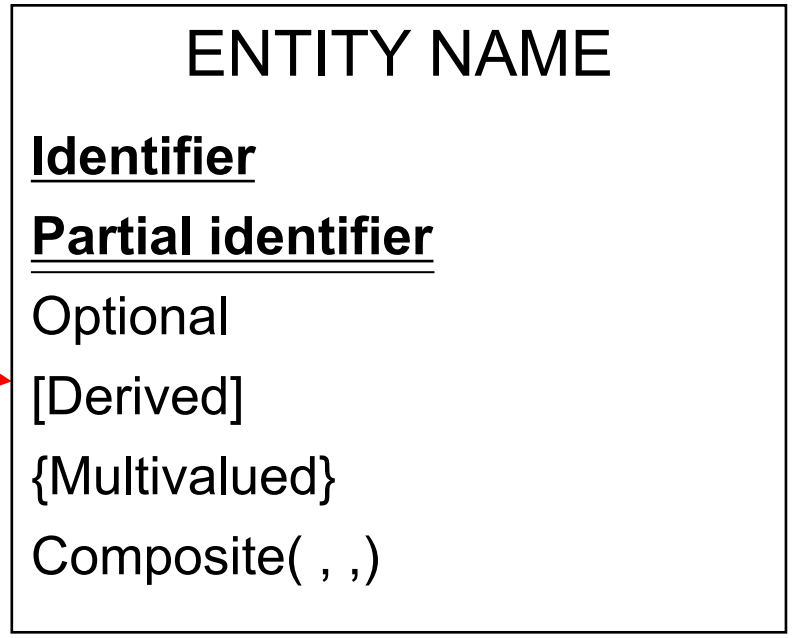
COURSE ID	TITLE	INSTRUCTOR NAME	INSTRUCTOR LOCATION	GRADE
IS 350	Database Mgt.	Codd	B104	A
IS 465	System Analysis	Parsons	B317	B

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Attributes

- A property or characteristic of an entity type

- Classifications of attributes:
  - Identifier Attributes
  - Required versus Optional
  - Stored versus Derived
  - Single-Valued versus Multivalued Attribute
  - Simple versus Composite



Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

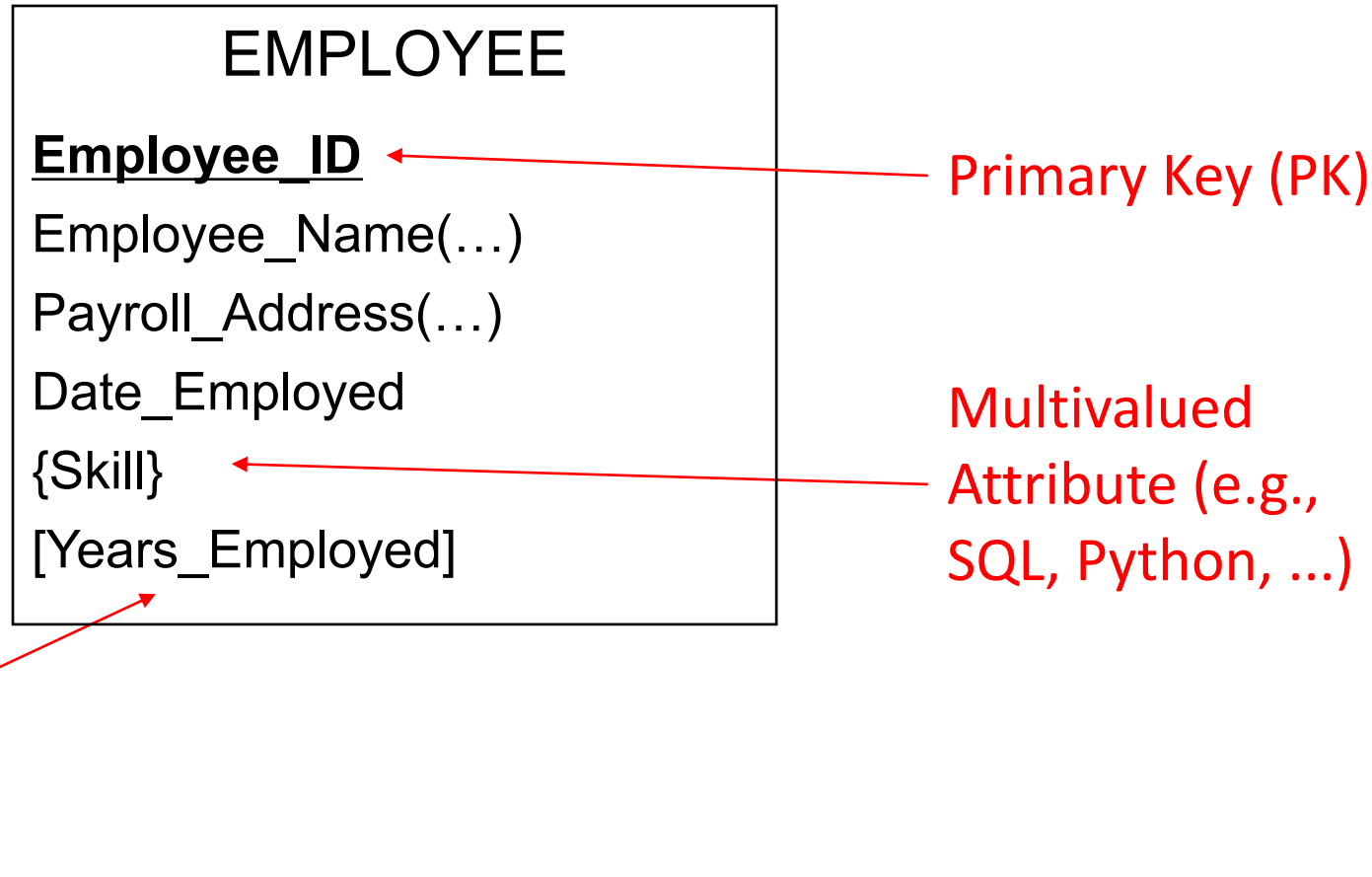
Example: Describe the Attributes

EMPLOYEE
Employee_ID
Employee_Name(…)
Payroll_Address(…)
Date_Employed
{Skill}
[Years_Employed]

?

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: Describe the Attributes



Naming Attributes

Poor Examples	Good Examples
TheDayThatThisPersonEnrolled	Date
	Birth_Date
NumEnrollInSpecificClass	NumberEnrolled
Student_Names	StudentName
ClientLastName	CourseID
	Employee_ID

- Guidelines for naming attributes:
  - Be concise
  - Use singular nouns or noun phrases
  - Names should be unique (at least within an entity type)
  - Follow a standard format (e.g., either Camelcase or "_")
  - Similar attributes should use the same qualifiers and classes: consistency! (e.g., CustomerID, ProductID)

Example: modeling flights

- Assume you want to model "flights"
- Attributes: FlightNumber, Date, NumberOfPassengers
- What would be the key / identifier?

?

Example: modeling flights

- Assume you want to model "flights"
- Attributes: FlightNumber, Date, NumberOfPassengers
- What would be the key / identifier?

FLIGHT
Flight_ID **(Flight_Number, Date)**
Number_of_Passengers
…

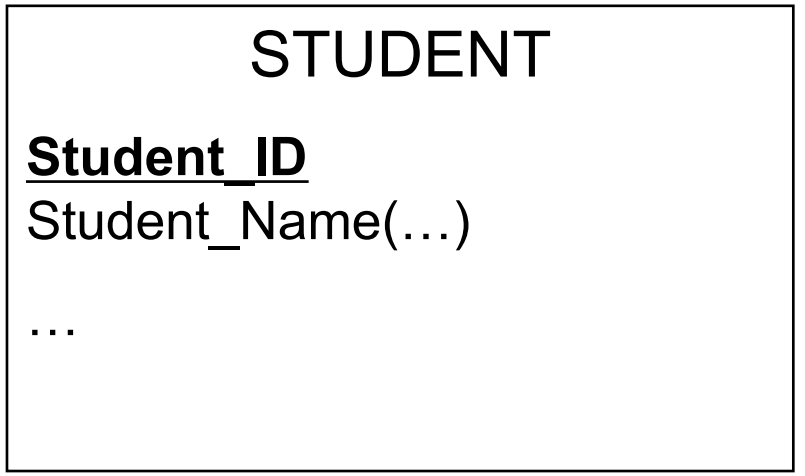
US Airways Flight 1549

The downed US Airways Flight 1549 floating on the Hudson River

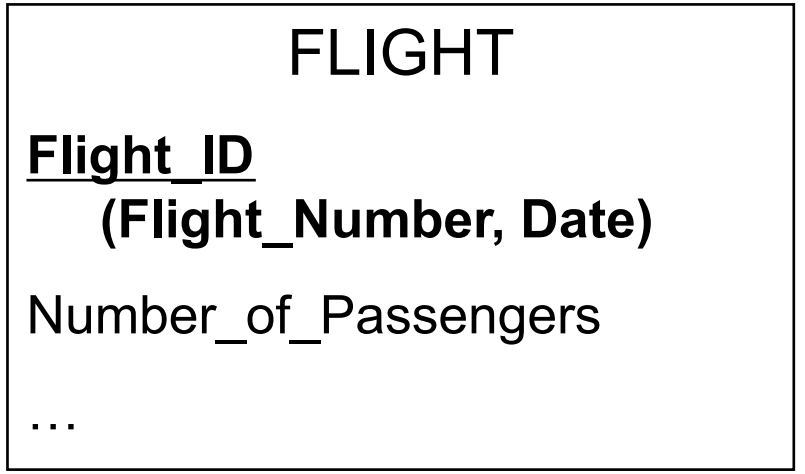
Accident summary	
Date	January 15, 2009

Identifier Examples: Simple and Composite

- Simple identifiers:
  - Single attribute uniquely identifies each entity instance
  - Identifier attribute underlined



- Composite identifiers:
  - Multiple attributes required to uniquely identifies each entity instance
  - Identifier attribute underlined and composite attributes listed below in (parentheses)



Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/