[IF-5-OT7:TD] Foundation of data engineering

MCF Riccardo Tommasini

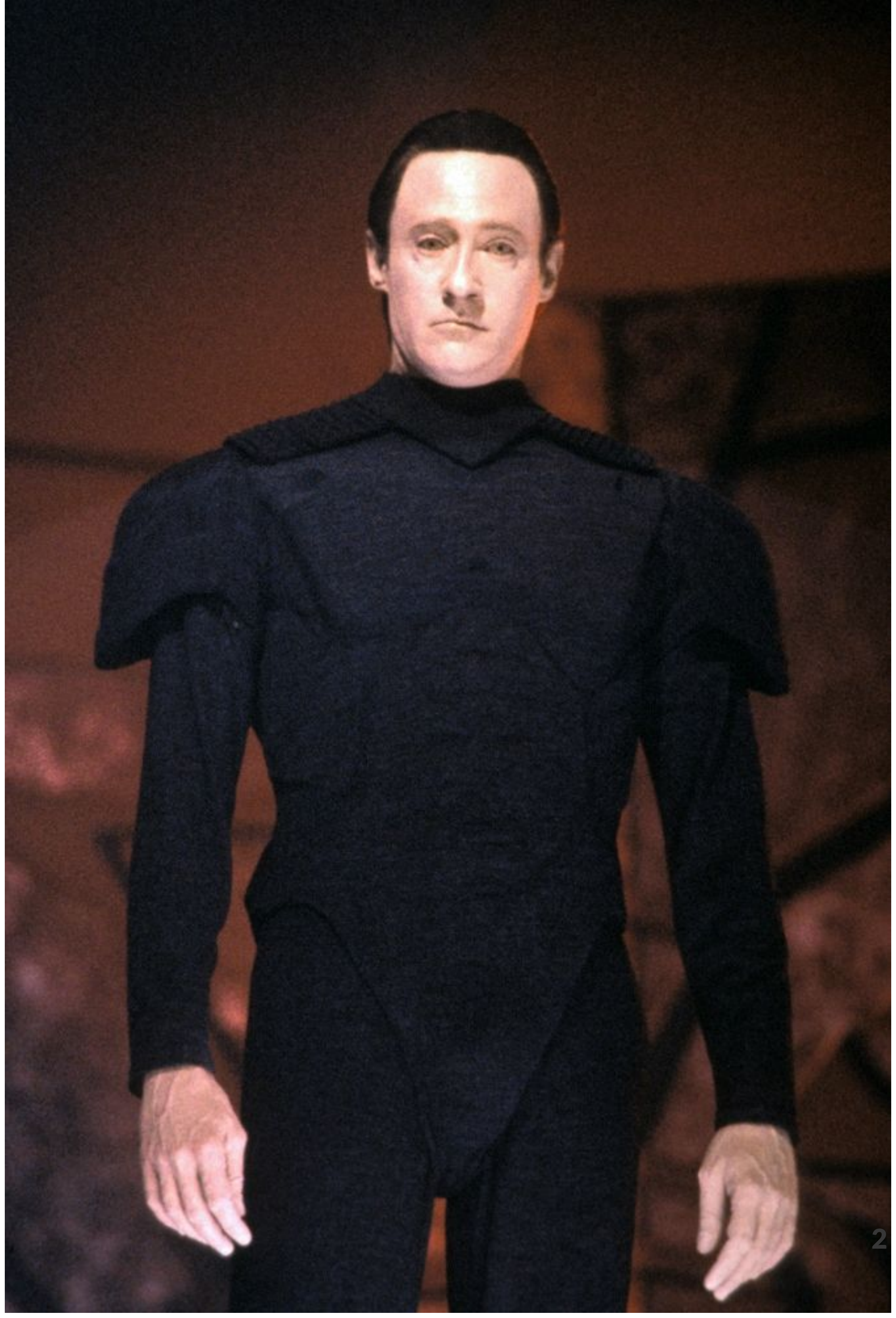
http://rictomm.me

riccardo.tommasini@insa-lyon.fr

INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
LYON

Data Modelling

It is the process of defining the structure of the data for the purpose of communicating[11] or to develop an information systems[12].



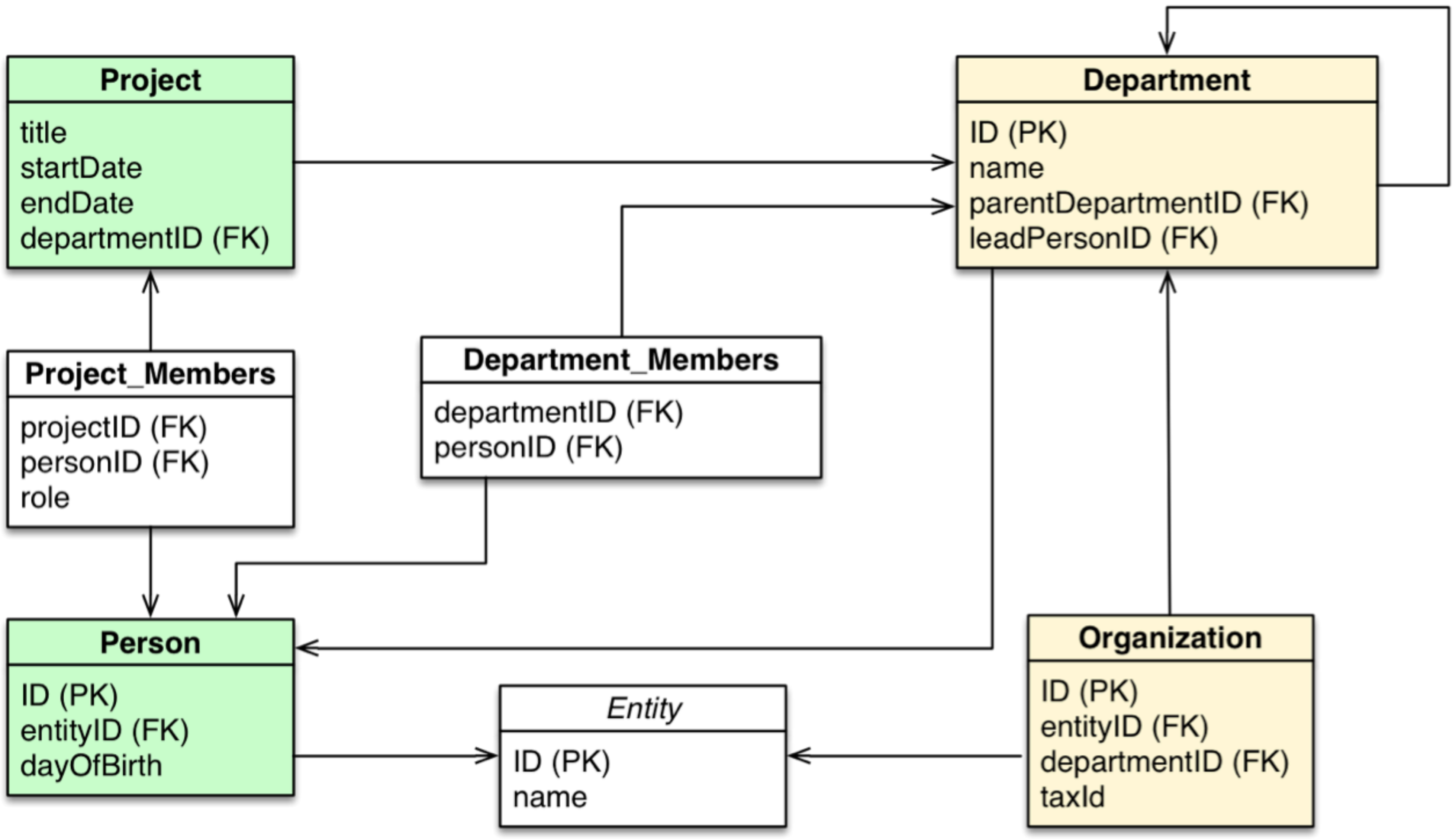
[11] between functional and technical people to show data needed for business processes

[12] between components of the information system, how data is stored and accessed.

What is a data model?

- A data model represents the structure and the integrity of the data elements of a (single) applications 2
- Data models provide a framework for data to be used within information systems by giving specific definitions and formats.
- The literature of data management is rich of data models that aim at providing increased expressiveness to the modeller and capturing a richer set of semantics.

Any Example?



Project
title
startDate
endDate
departmentID (FK)

Department
ID (PK)
name
parentDepartmentID (FK)
leadPersonID (FK)

Project_Members
projectID (FK)
personID (FK)
role

Department_Members
departmentID (FK)
personID (FK)

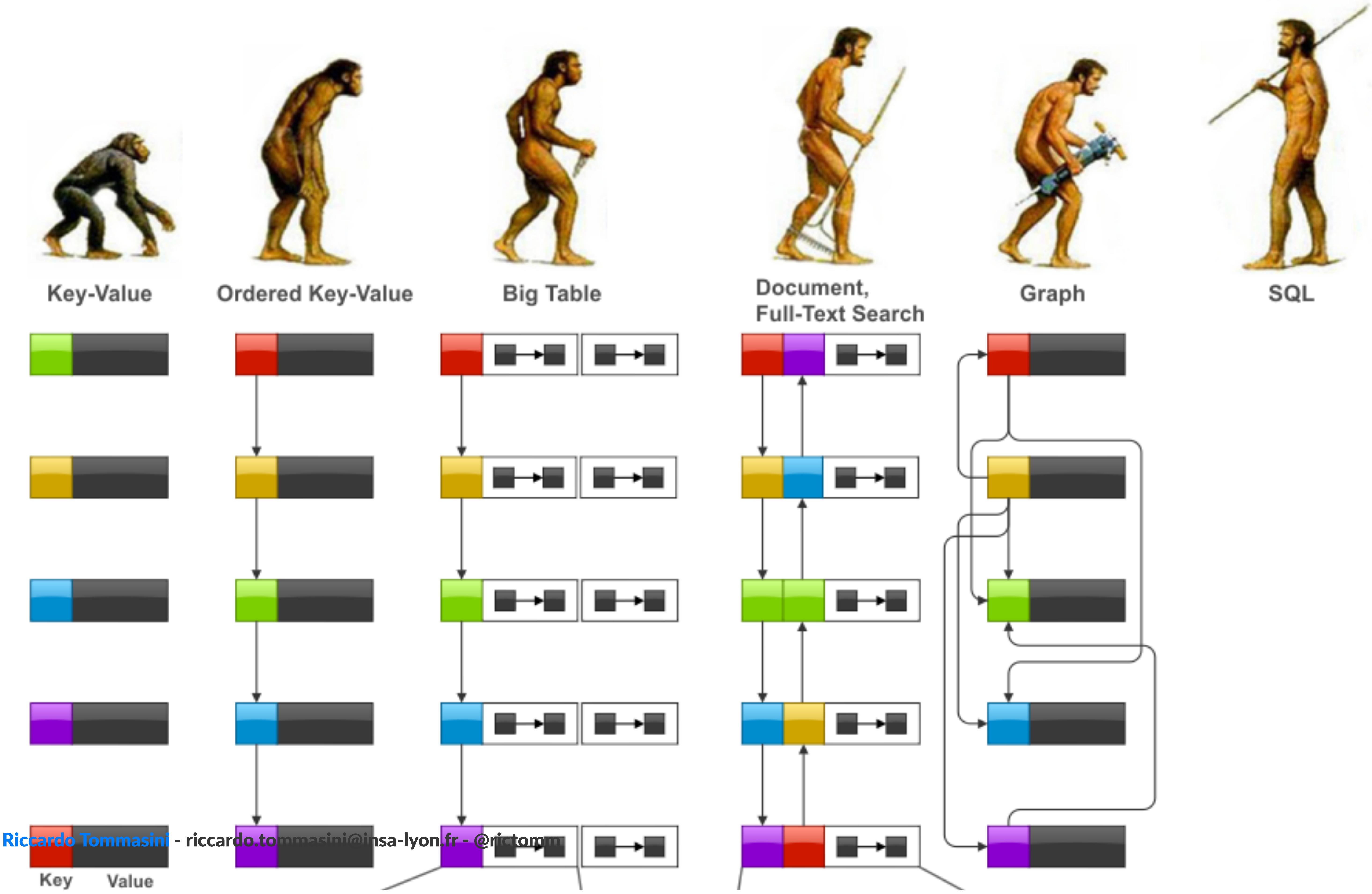
Person
ID (PK)
entityID (FK)
dayOfBirth

Entity
ID (PK)
name

Organization
ID (PK)
entityID (FK)
departmentID (FK)
taxId

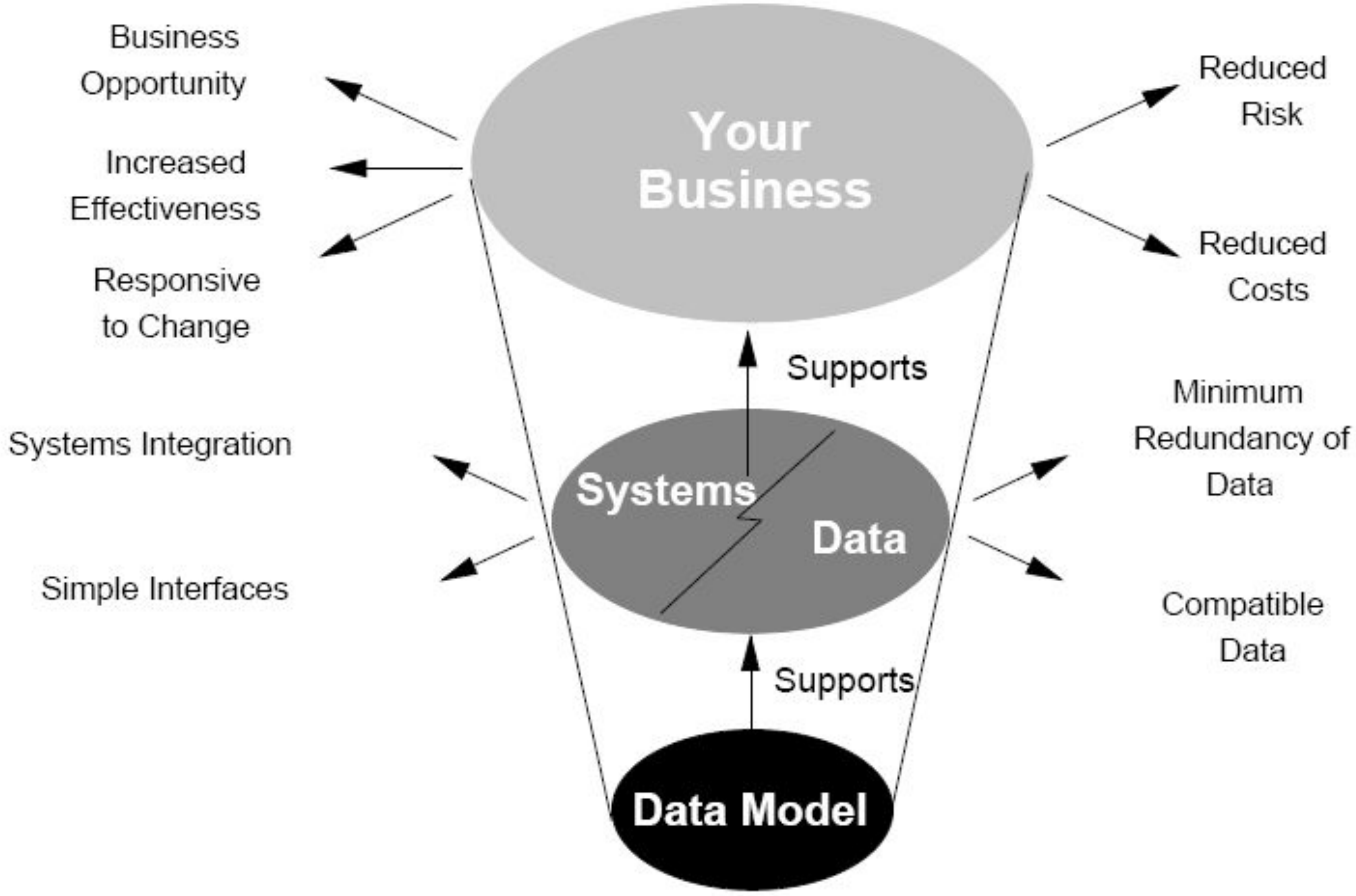
History of Data Models[5]

[5] by Ilya Katsov


Key-Value
Ordered Key-Value
Big Table
Document,
Full-Text Search
Graph
SQL
Key
Value

> Data models are perhaps the most important part of developing software. They have such a profound effect not only on how the software is written, but also on how we think about the problem that we are solving[13].
>
> — *Martin Kleppmann*

[13] Designing Data-Intensive Applications

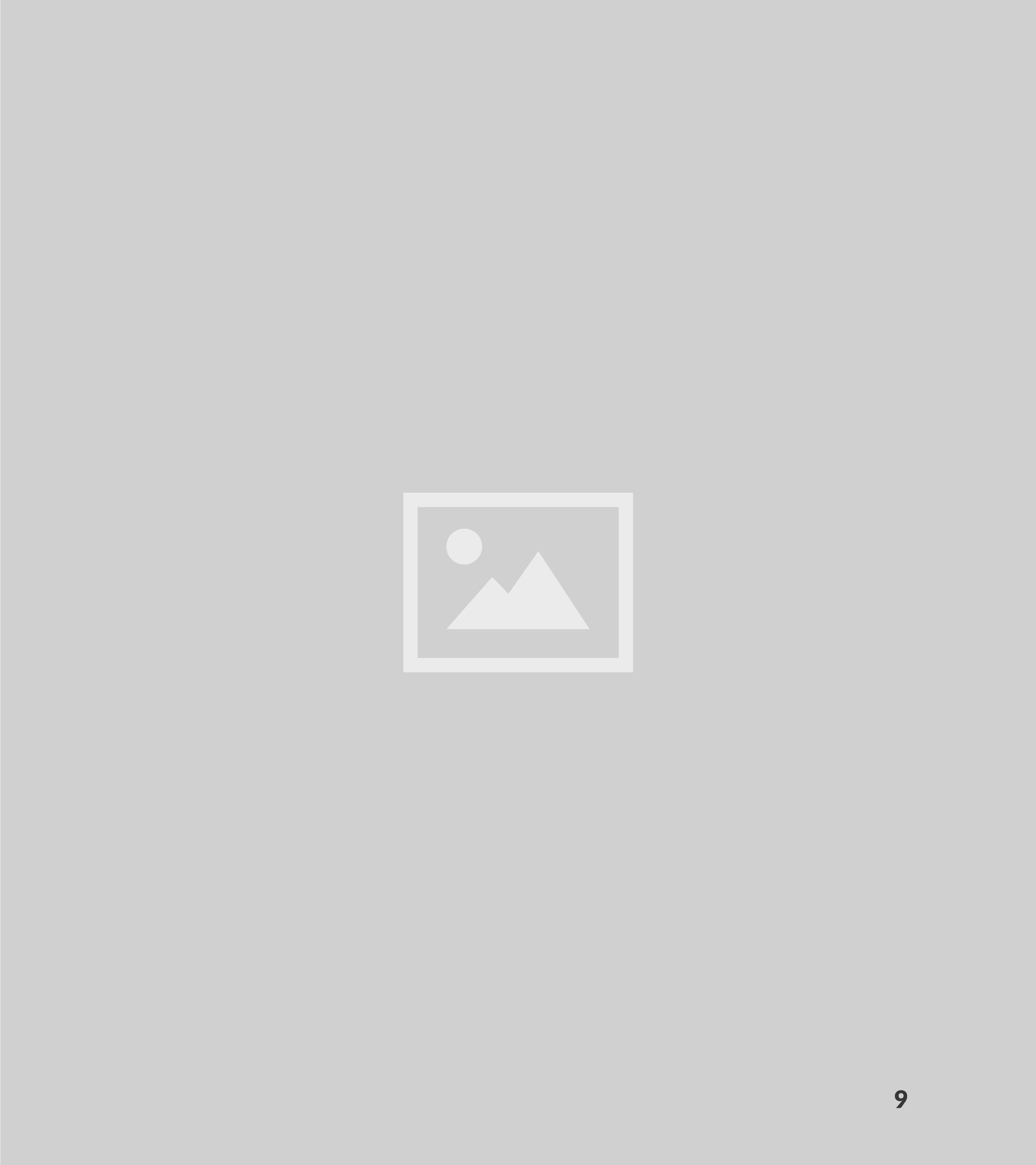

Business Opportunity
Increased Effectiveness
Responsive to Change
Your Business
Reduced Risk
Reduced Costs
Supports
Systems Integration
Simple Interfaces
Systems
Data
Minimum Redundancy of Data
Compatible Data
Supports
Data Model

Level of Data Modelling

Conceptual: The data model defines *WHAT* the system contains.

Logical: Defines *HOW* the system should be implemented regardless of the DBMS.

Physical: This Data Model describes *HOW* the information system will be implemented using a specific technology[14].



[14] physical

Conceptual model is typically created by Business stakeholders. The purpose is to organize, scope and define business concepts and rules. Definitions are most important this level.

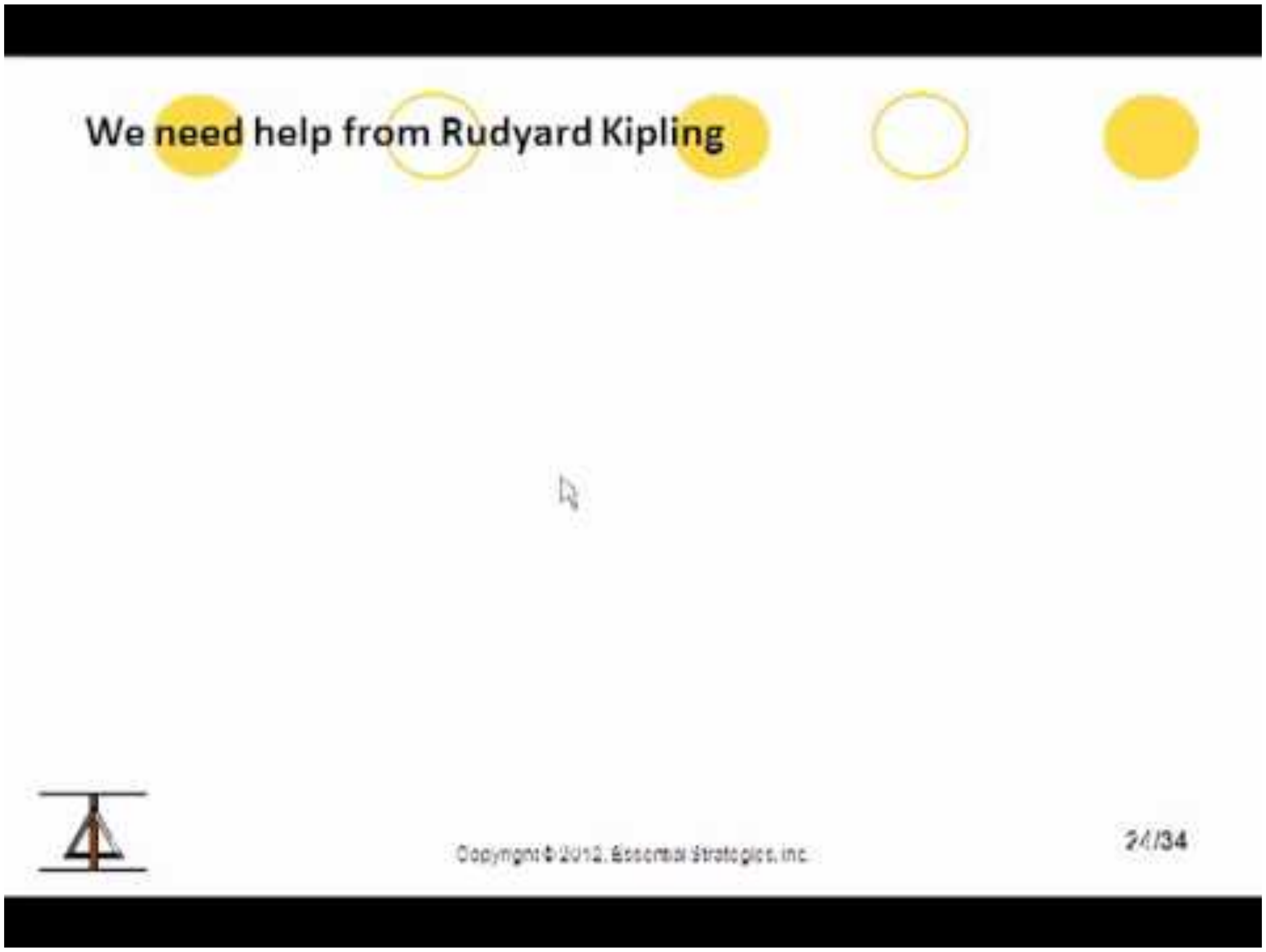
Logical model is typically created by Data Architects. The purpose is to developed technical map of rules and data structures. Business rules, relationships, attribute become visible. Conceptual definitions become metadata.

Physical model is typically created by DBA and developers. The purpose is actual implementation of the database. Trade-offs are explored by in terms of data structures and algorithms.

A Closer Look[15]

[15] slides & video by Donna Burbank

The variety of data available today encourages the design and development of dedicated data models and query languages that can improve both BI as well as the engineering process itself.



We need help from Rudyard Kipling

Copyright © 2012, Essential Strategies, Inc.

24/34

Conceptual

- Semantic Model (divergent)
  - Describes an enterprise in terms of the language it uses (the jargon).
  - It also tracks inconsistencies, i.e., semantic conflicts
- Architectural Model (convergent)
  - More fundamental, abstract categories across enterprise

Logical

Already bound to a technology, it typically refers already to implementation details

- Relational
- Hierarchical
- Key-Value
- Object-Oriented
- Graph

Since it has a physical bias, you might be tempted to confuse this with the physical model, but this is wrong.

Physical

The physical level describes how data are **Stored** on a device.

- Data formats
- Distribution
- Indexes
- Data Partitions
- Data Replications

...an you are in the Big Data World

Data Modelling Techniques

According to Len Silverston (1997) only two modelling methodologies stand out, top-down and bottom-up.

- Bottom-up models or View Integration models are often the result of a reengineering "Reengineering (software)") effort. These models are usually physical, application-specific, and incomplete from an enterprise perspective. They may not promote data sharing, especially if they are built without reference to other parts of the organization.7(https://en.wikipedia.org/wiki/Data*Modelling#cite*note-SIG97-7)
- Top-down logical data models, on the other hand, are created in an abstract way by getting information from people who know the subject area. A system may not implement all the entities in a logical model, but the model serves as a reference point or template.7(https://en.wikipedia.org/wiki/Data*Modelling#cite*note-SIG97-7)

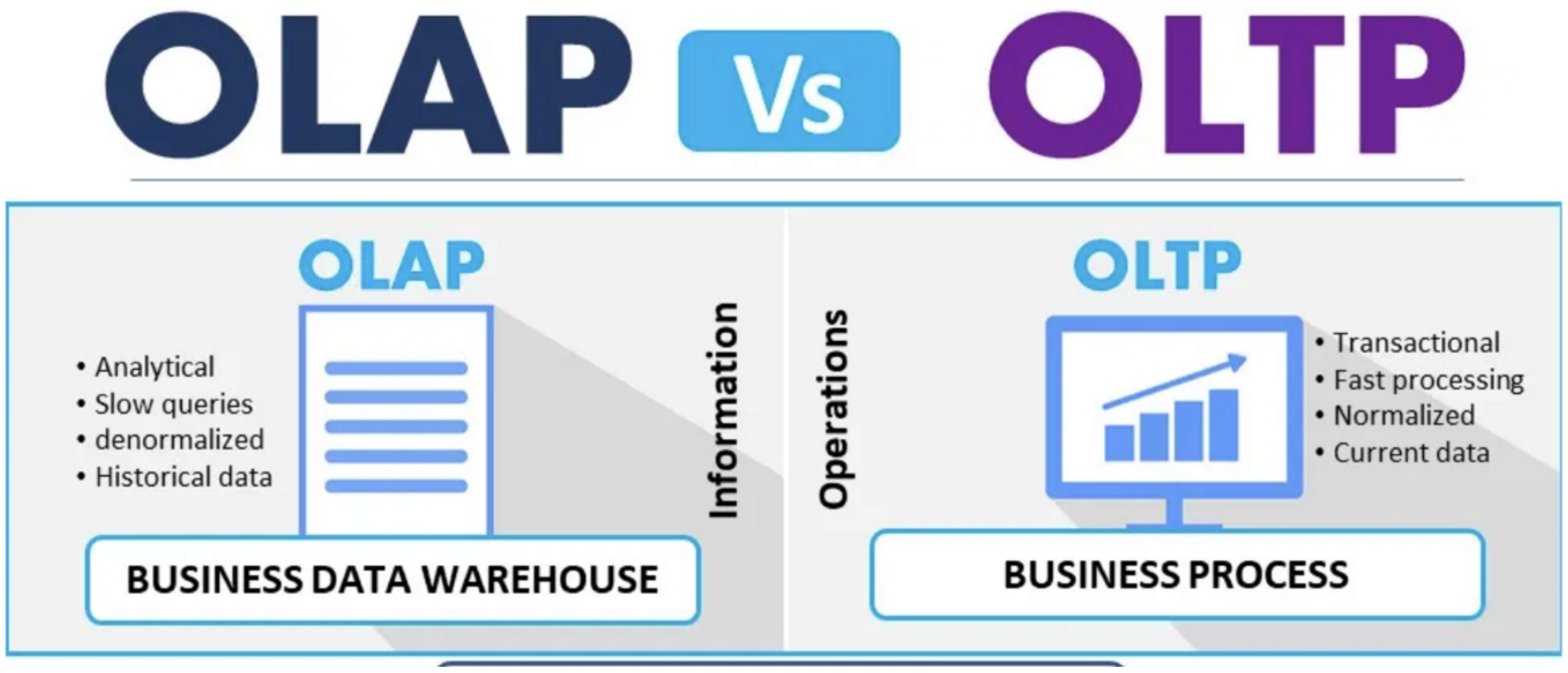
source: wikipedia

Data Modelling Techniques[18]

- **Entity-Relationship (ER) Modelling**[^19] prescribes to design model encompassing the whole company and describe enterprise business through Entities and the relationships between them
  - it complies with 3rd normal form
  - tailored for OLTP
- **Dimensional Modelling** (DM)[^110] focuses on enabling complete requirement analysis while maintaining high performance when handling large and complex (analytical) queries
  - The star model and the snowflake model are examples of DM
  - tailored for OLAP
- **Data Vault (DV) Modelling**[^111] focuses on data integration trying to take the best of ER 3NF and DM
  - emphasizes establishment of an suitable basic data layer focusing on data history, traceability, and atomicity
  - one cannot use it directly for data analysis and decision making
- **Domain Driven Design**[^112] focuses on designing software based on the underlying domain.
  - promotes the usage of an ubiquitous language help communication between software developers and domain experts.
  - replaces the conceptual level for NOSQL

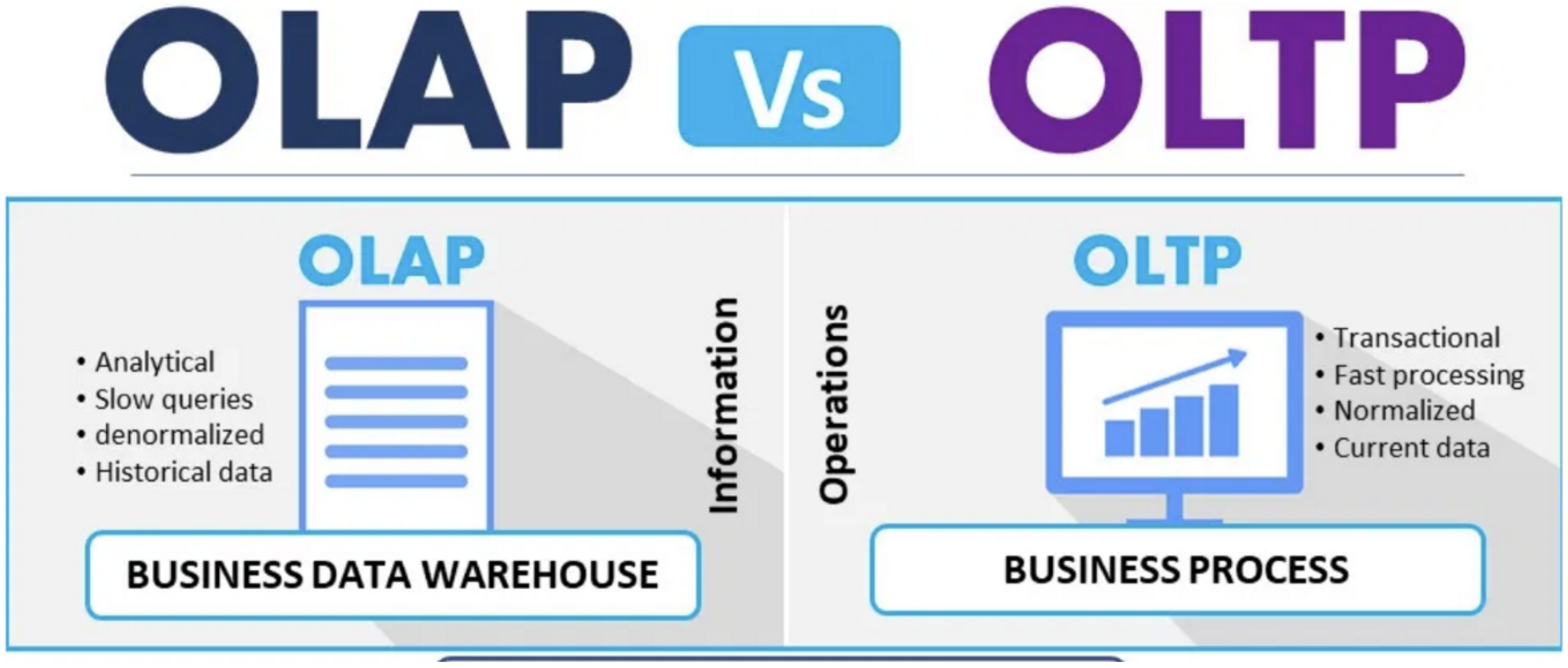
[18] source
[^19]: by Bill Inmon
[^110]: Ralph Kimball, book ‘The Data Warehouse Toolkit — The Complete Guide to Dimensional Modelling"
[^111]: https://en.wikipedia.org/wiki/Data*vault*Modelling
[^112]: Evans, Eric. Domain-driven design: tackling complexity in the heart of software. Addison-Wesley Professional, 2004.

Let's Talk about Workloads



- **OLTP** systems are usually expected to be **highly available** and to process transactions with low latency, since they are often critical to the operation of the business.
- **OLAP** queries are often written by business analysts, and feed into reports that help the management of a company make better decisions (business intelligence).

Workloads



- **OLTP** systems are usually expected to be **highly available** and to process transactions with low latency, since they are often critical to the operation of the business.
- **OLAP** queries are often written by business analysts, and feed into reports that help the management of a company make better decisions (business intelligence).

Online Transactional Processing

Because these applications are interactive, the access pattern became known as **online**

Transactional means allowing clients to make low-latency reads and writes—as opposed to batch processing jobs, which only run periodically (for example, once per day).

Refresh on ACID Properties

- ACID, which stands for Atomicity, Consistency, Isolation, and Durability[11]
- **Atomicity** refers to something that cannot be broken down into smaller parts.
    - It is not about concurrency (which comes with the I)
- **Consistency** (overused term), that here relates to the data *invariants* (integrity would be a better term IMHO)
- **Isolation** means that concurrently executing transactions are isolated from each other.
    - Typically associated with serializability, but there weaker options.
- **Durability** means (fault-tolerant) persistency of the data, once the transaction is completed.
- The terms was coined in 1983 by Theo Härder and Andreas Reuter [16]

[11] between functional and technical people to show data needed for business processes

[16] Theo Härder and Andreas Reuter: “Principles of Transaction-Oriented Database Recovery,” ACM Computing Surveys, volume 15, number 4, pages 287–317, December 1983. doi:10.1145/289.291

Online Analytical Processing

An OLAP system allows a data analyst to look at different cross-tabs on the same data by interactively selecting the attributes in the cross-tab

Statistical analysis often requires grouping on multiple attributes.

Example[^121]

Consider this is a simplified version of the sales fact table joined with the dimension tables, and many attributes removed (and some renamed)

sales (item*name*, *color*, *clothes*size, quantity)

item_name	color	clothes_size	quantity
dress	dark	small	2
dress	dark	medium	6
...	...	...	...
pants	pastel	medium	0
pants	pastel	large	1
pants	white	small	3
pants	white	medium	0
shirt	white	medium	1
...	...	...	...
shirt	white	large	10
skirt	dark	small	2
skirt	dark	medium	5
...	...	...	...

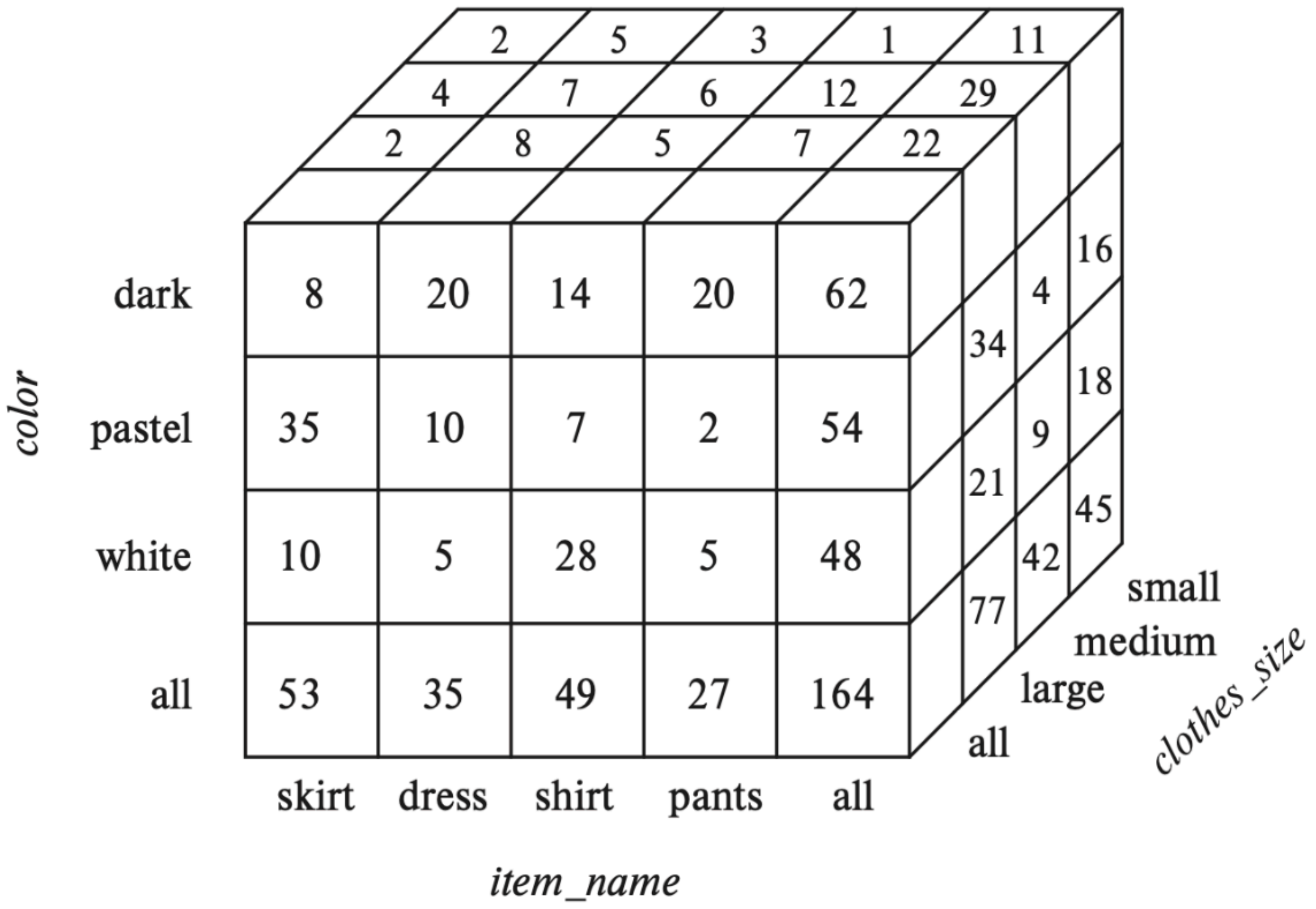
Cross-tabulation of sales by item name and color

	dark	pastel	white	total
skirt	8	35	10	53
dress	20	11	5	36
shirt	22	4	46	72
pants	23	42	25	90
total	73	92	102	267

columns header: color
rows header: item name

Data Cube[^121]

- It is the generalization of a Cross-tabulation



color	skirt	dress	shirt	pants	all
dark	8	20	14	20	62
pastel	35	10	7	2	54
white	10	5	28	5	48
all	53	35	49	27	164

Cheat Sheet of OLAP Operations[17]

- **Pivoting**: changing the dimensions used in a cross-tab
  - E.g. moving colors to column names
- **Slicing**: creating a cross-tab for fixed values only
  - E.g fixing color to white and size to small
  - Sometimes called dicing, particularly when values for multiple dimensions are fixed.
- **Rollup**: moving from finer-granularity data to a coarser granularity
  - E.g. aggregating away an attribute
  - E.g. moving from aggregates by day to aggregates by month or year
- **Drill down**: The opposite operation - that of moving from coarser granularity data to finer-granularity data

[17] Database System Concepts Seventh Edition Avi Silberschatz Henry F. Korth, S. Sudarshan McGraw-Hill ISBN 9780078022159 link

Summary OLTP vs OLAP[13]

Property	OLTP	OLAP
Main read pattern	Small number of records per query, fetched by key	Aggregate over large number of records
Main write pattern	Random-access, low-latency writes from user input	Bulk import (ETL) or event stream
Primarily used by	End user/customer, via web application	Internal analyst, for decision support
What data represents	Latest state of data (current point in time)	History of events that happened over time
Dataset size	Gigabytes to terabytes	Terabytes to petabytes

[13] Designing Data-Intensive Applications

Data Modelling for Databases

- Works in phases related to the aforementioned levels of abstractions[31]
- Uses different data models depending on the need:
  - Relational, Graph, Document...
- Tries to avoid two major pitfalls:
  - **Redundancy**: A design should not repeat information
  - **Incompleteness**: A design should not make certain aspects of the enterprise difficult or impossible to model
- Optimized for OLTP

[31] Also known as Database Design

The biggest problem with redundancy is that information may become inconsistent in case of update

Before, let's refresh

Relational Database

A relational database consists of...
- a set of relations (tables)
- a set of integrity constraints

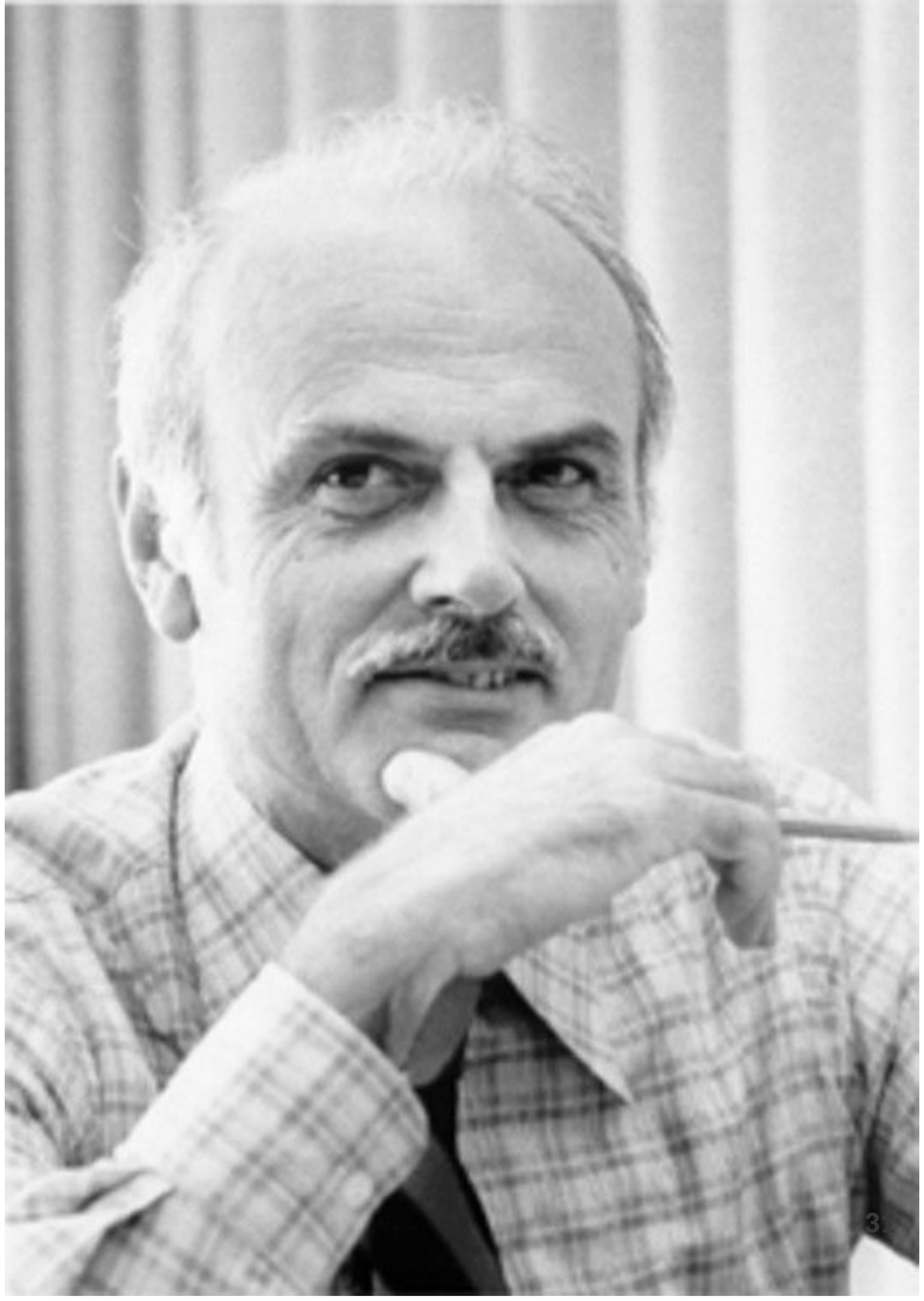
If the database satisfies all the constraints we said it is in a valid state.

An important distinction regards the **database schema**, which is the logical design of the database, and the **database instance**, which is a snapshot of the data in the database at a given instant in time.

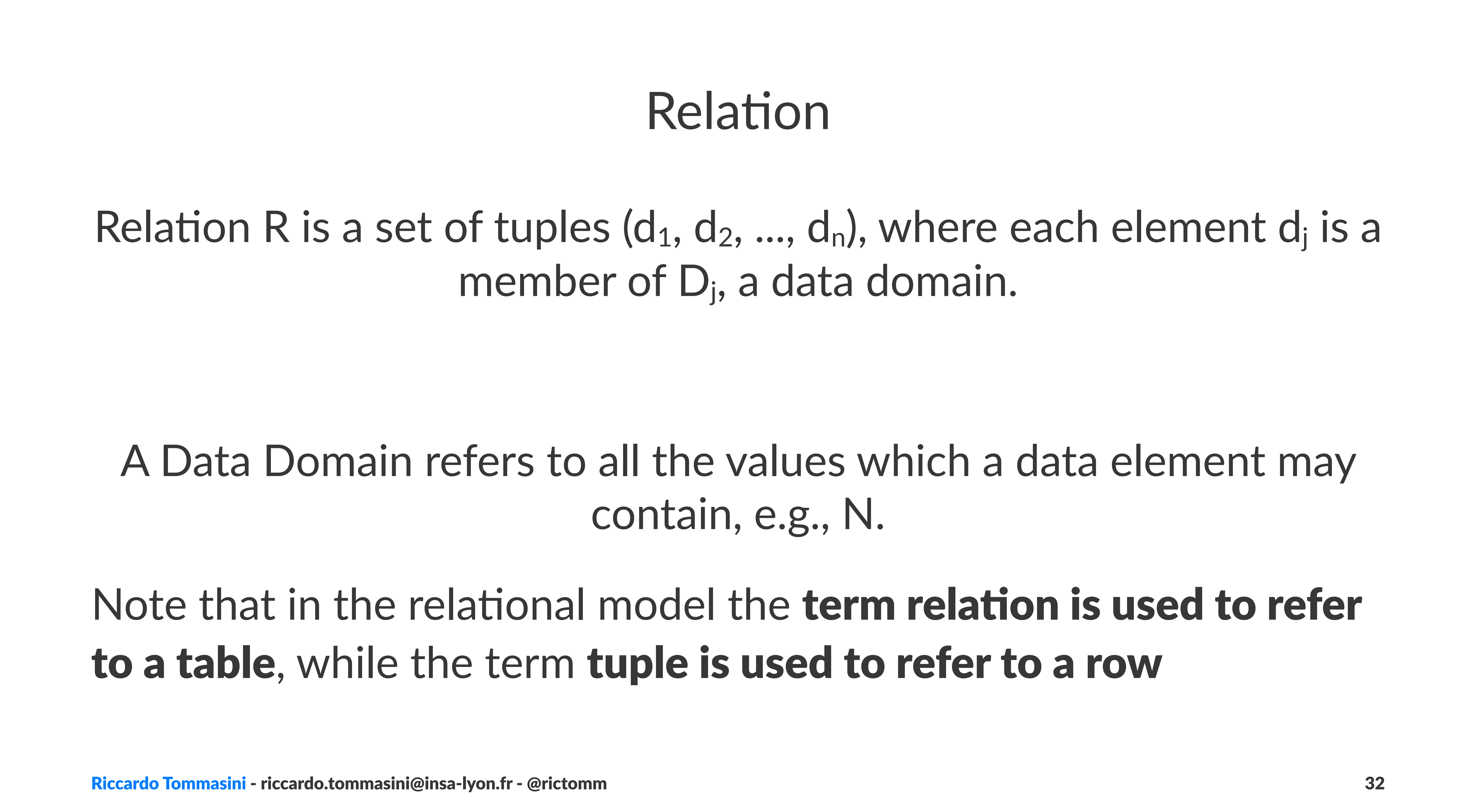
Relational Model [32]

A formal mathematical basis for databases based on set theory and first-order predicate logic

Underpins of SQL



[32] Extra Read Codd, Edgar F. "A relational model of data for large shared data
banks." Communications of the ACM 13.6 (1970): 377-38z



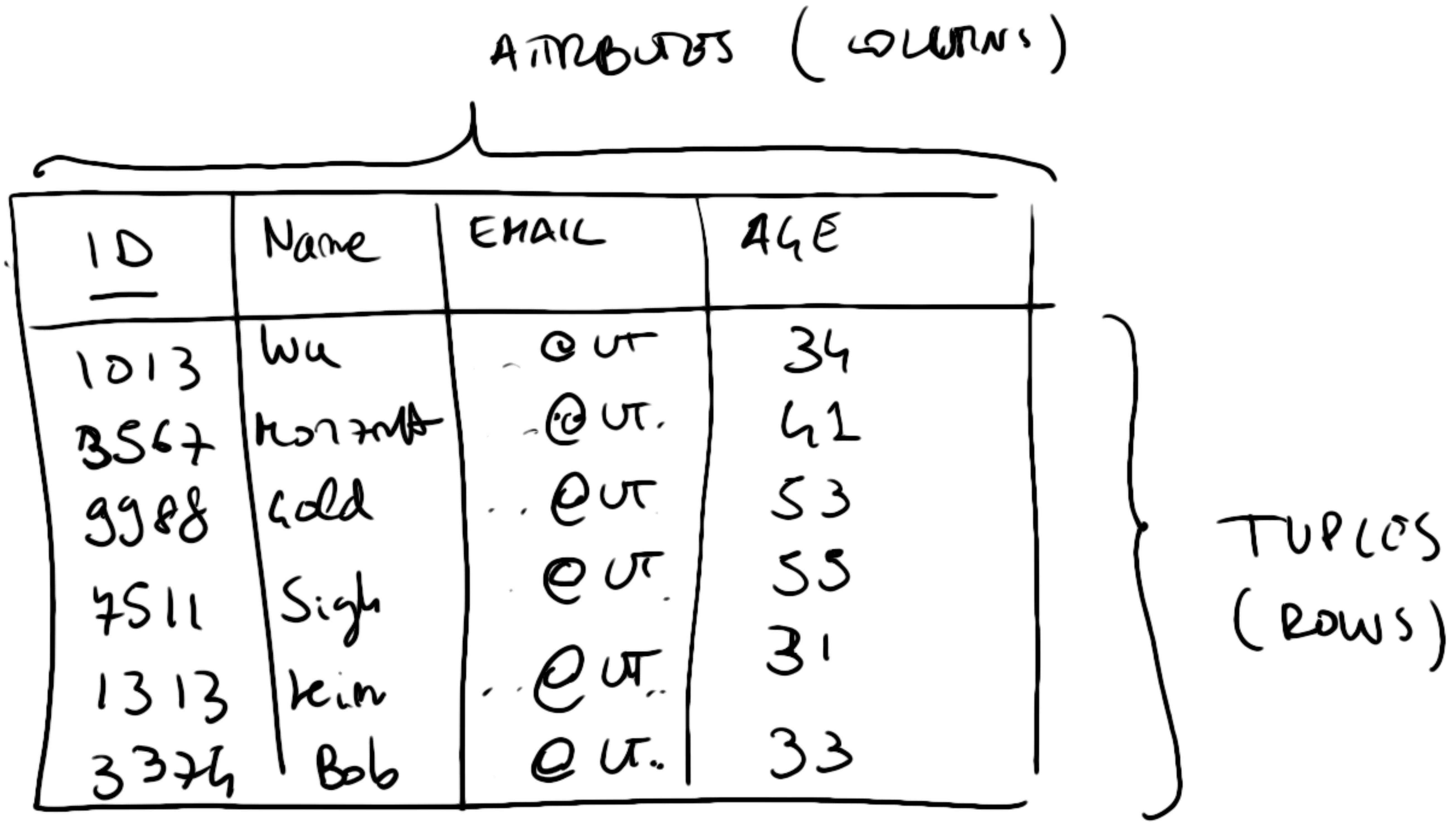
Relation

Relation R is a set of tuples (d_1, d_2, ..., d_n), where each element d_j is a member of D_j, a data domain.

A Data Domain refers to all the values which a data element may contain, e.g., N.

Note that in the relational model the **term relation is used to refer to a table**, while the term **tuple is used to refer to a row**

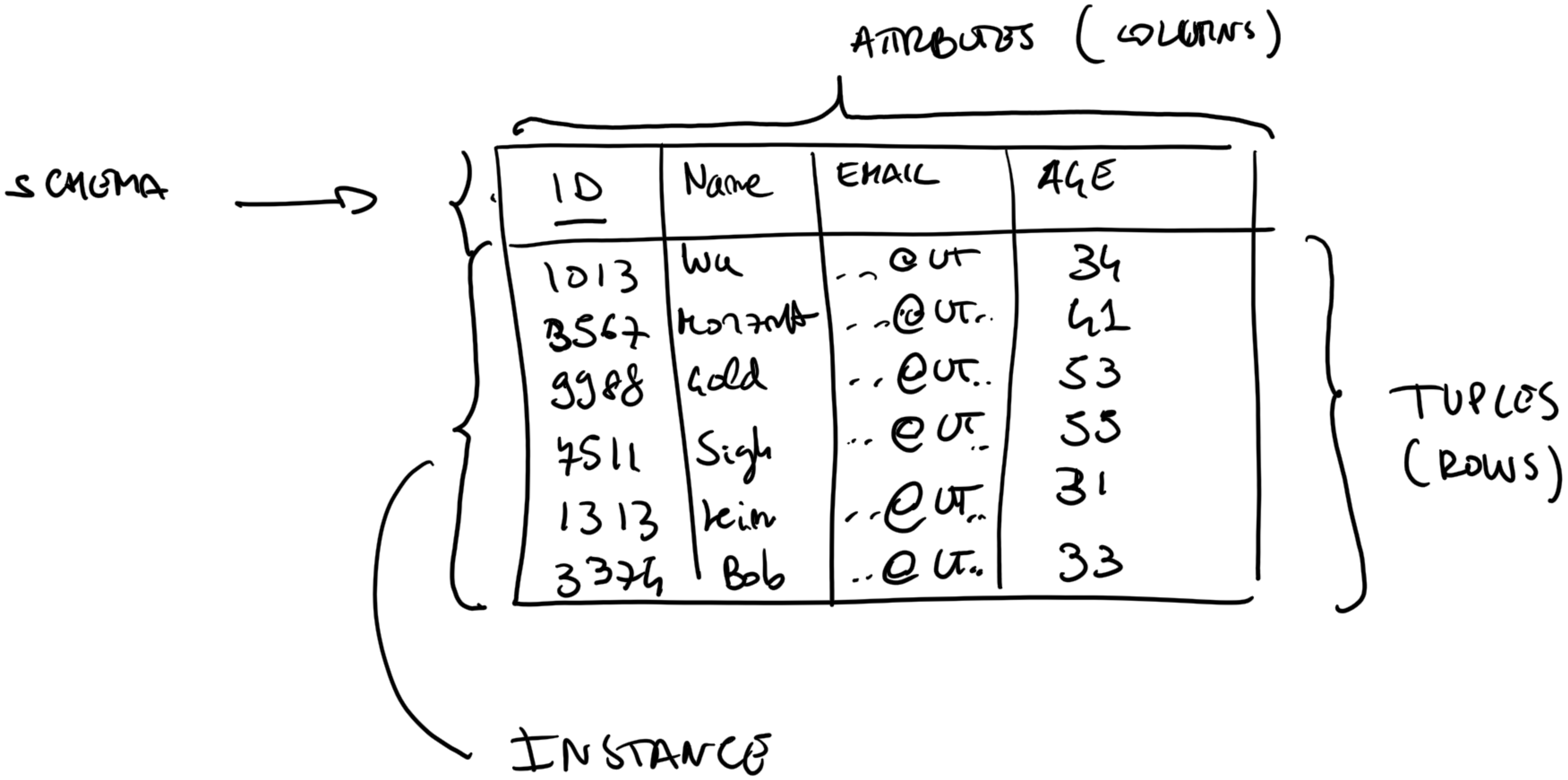
In mathematical terms, a tuple indicates a sequence of values. A relationship between n values is represented mathematically by an n-tuple of values, that is, a tuple with n values, which corresponds to a row in a table.



ID	Name	EMAIL	AGE
1013	Wu	@UT	34
3567	Kozzma	@UT.	41
9988	Gold	..@UT	53
7511	Sigh	@UT	55
1313	Kim	..@UT..	31
3374	Bob	@UT..	33

Relation Schema

- corresponds to the notion of **type** in programming languages
- consists of a list of **attributes** and their corresponding domains
- a **relation instance** corresponds to the programming-language no- tion of a value of a variable

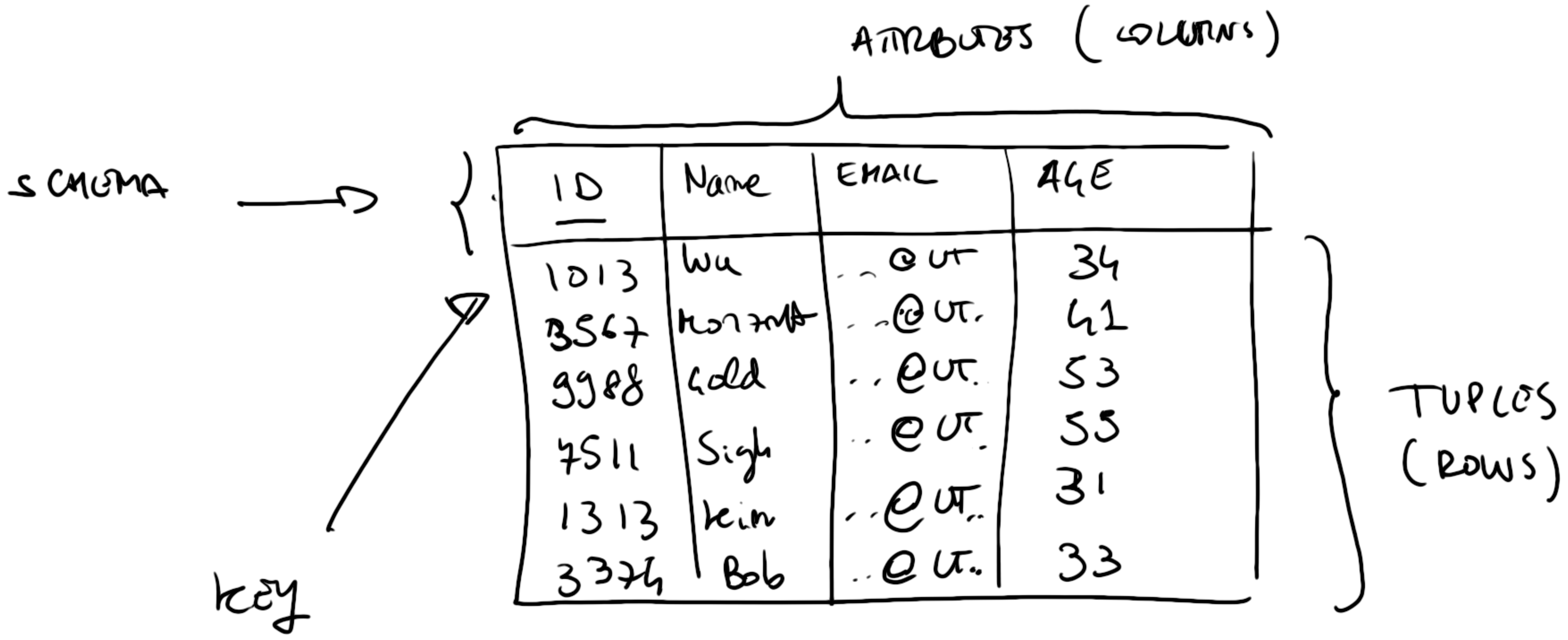


ID	Name	EMAIL	AGE
1013	Wu	..@UT	34
3567	Korzma	..@UT..	41
9988	Gold	..@UT..	53
7511	Sigh	..@UT..	55
1313	Kim	..@UT..	31
3374	Bob	..@UT..	33

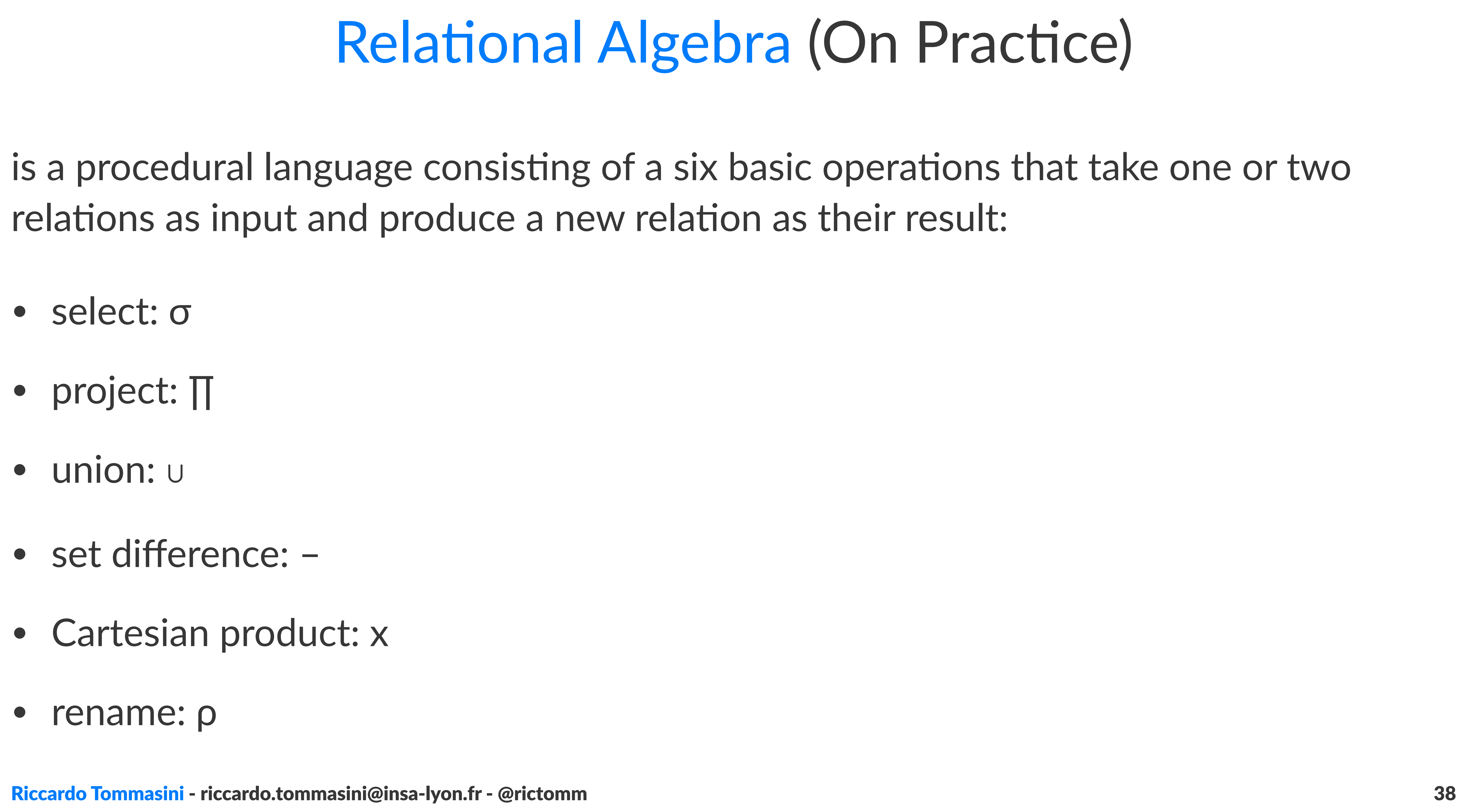
Keys

- A **superkey** is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation
- **candidate keys** are superkeys for which no proper subset is a superkey
- primary key is the chosen candidate key
- foreign key is s set of attributes from a referenced relation.

If K is a superkey, then so is any superset of K



<u>ID</u>	Name	EMAIL	AGE
1013	Wu	..@ut	34
3567	Korzma	..@ut.	41
9988	Gold	..@ut.	53
7511	Sigh	..@ut.	55
1313	Kein	..@ut..	31
3374	Bob	..@ut..	33



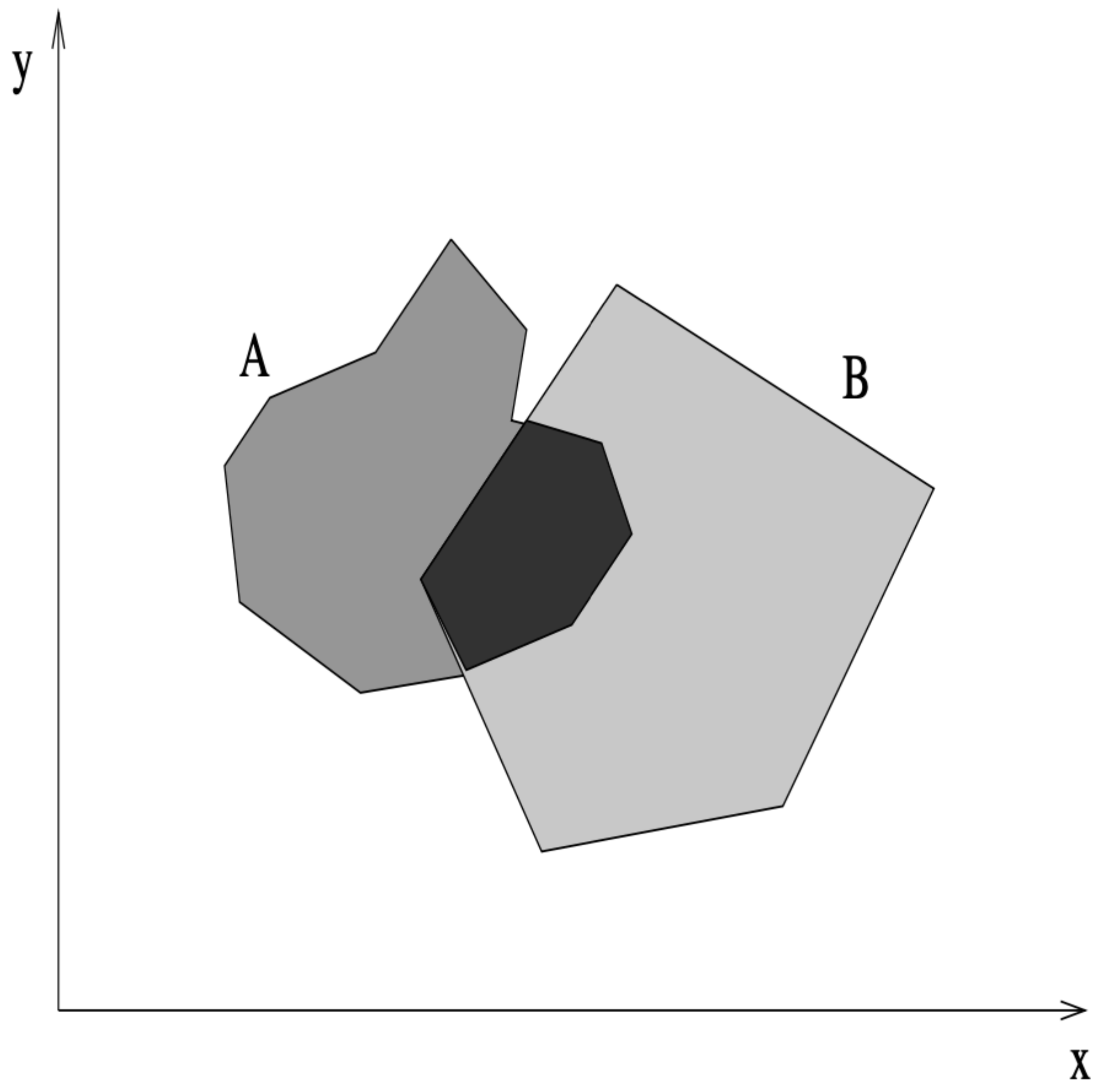
Relational Algebra (On Practice)

is a procedural language consisting of a six basic operations that take one or two relations as input and produce a new relation as their result:

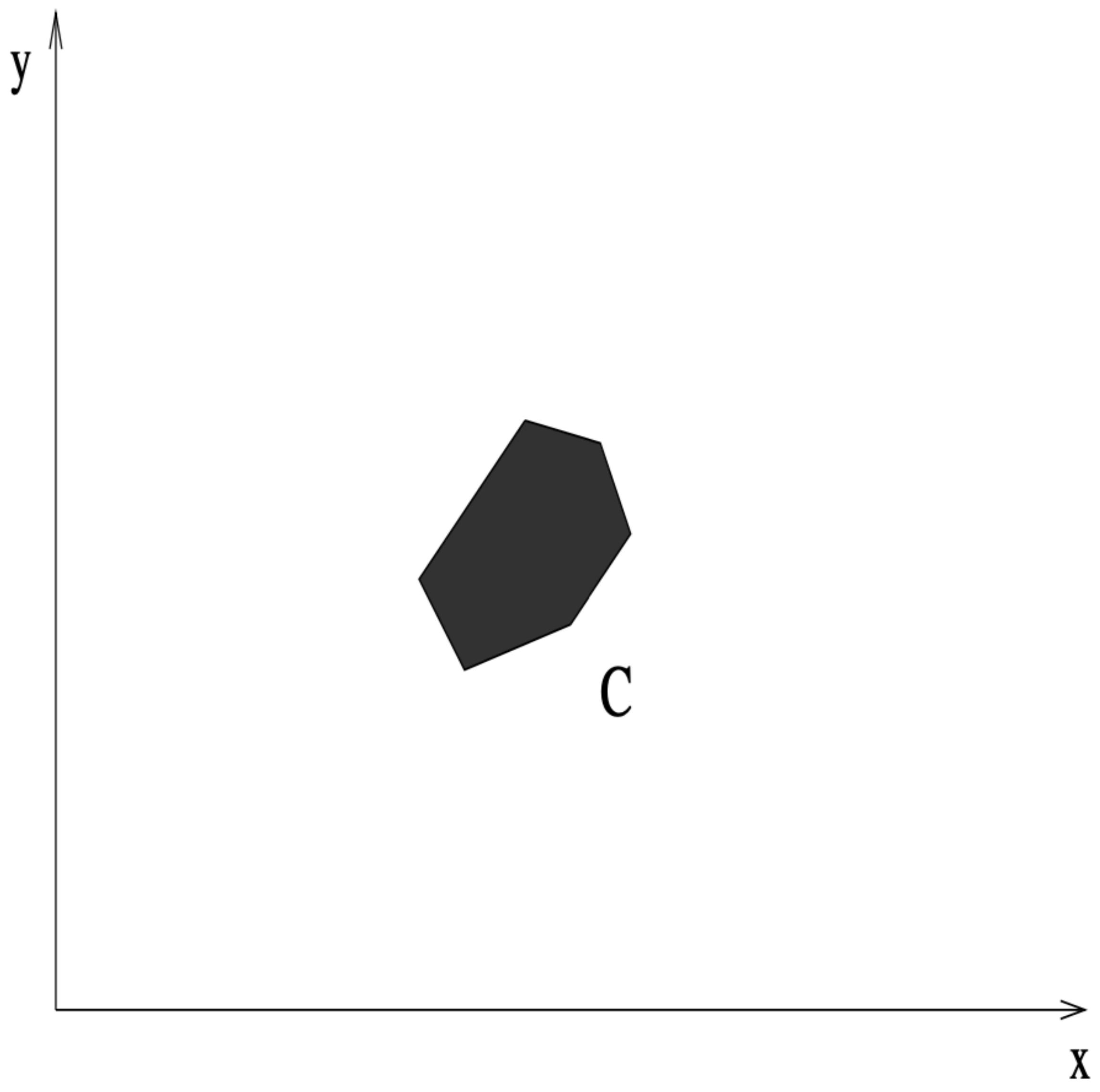
- select: σ
- project: ∏
- union: ∪
- set difference: –
- Cartesian product: x
- rename: ρ

Question: What is an algebra?

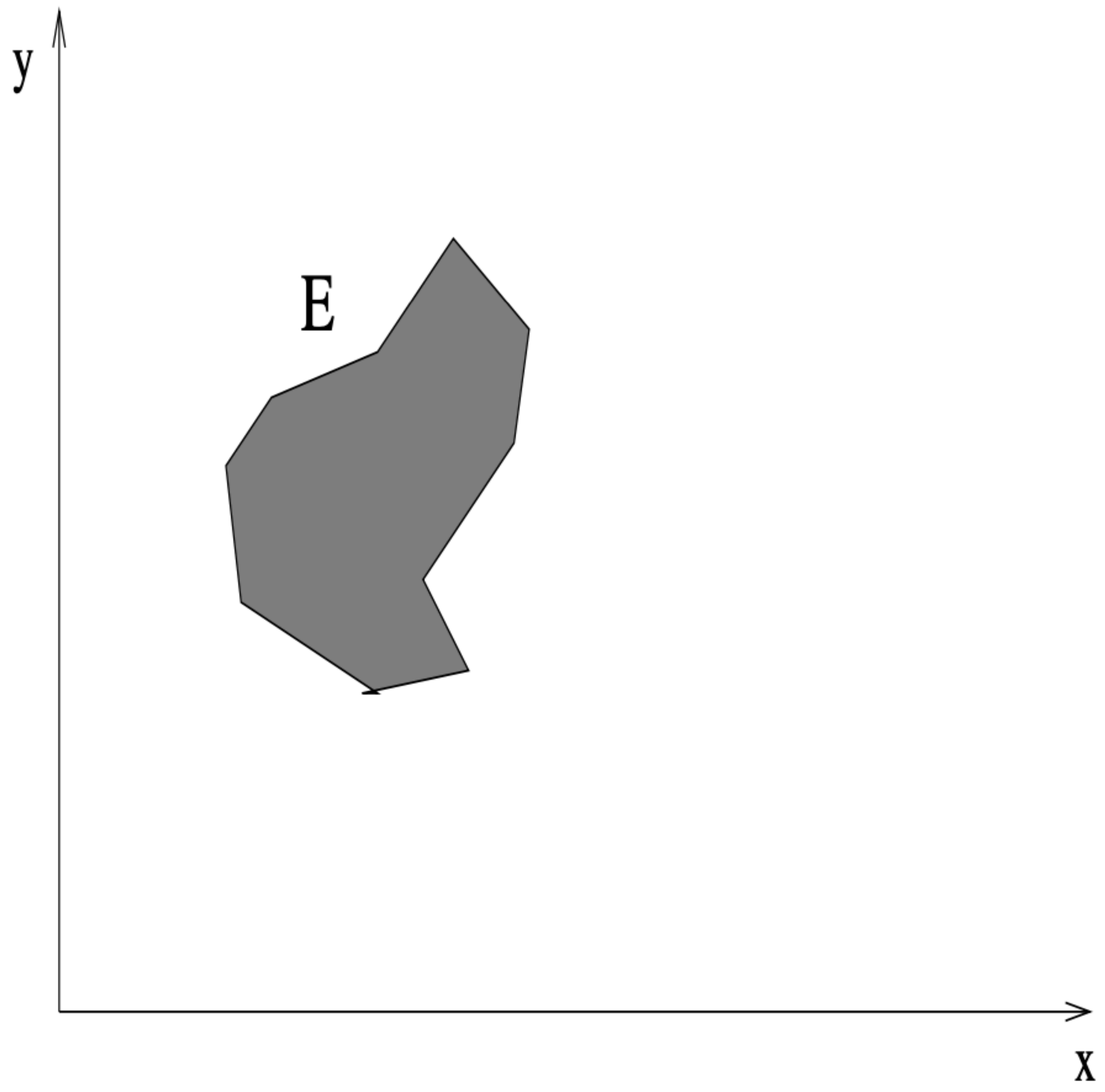
Two Sets



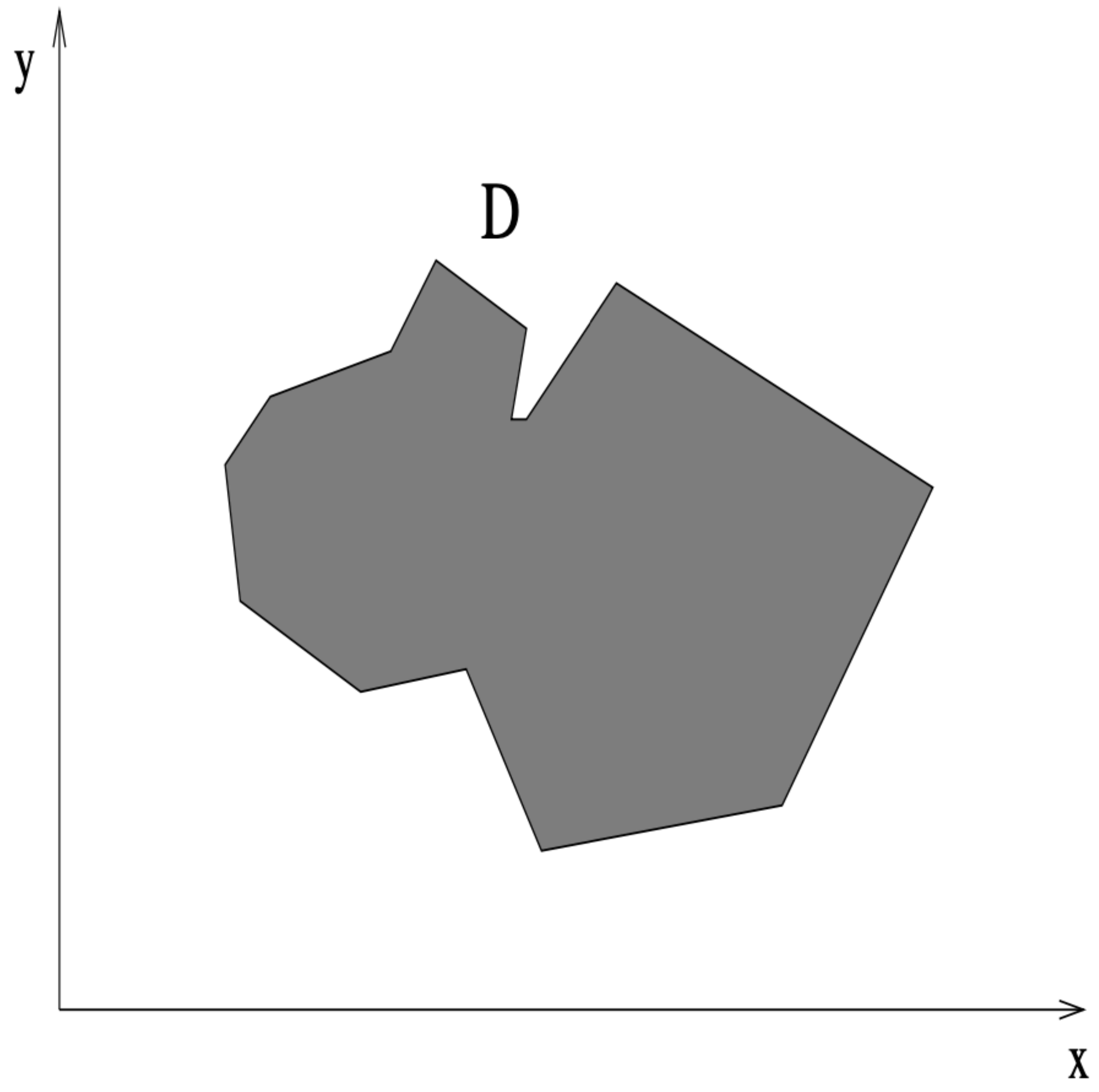
Intersection

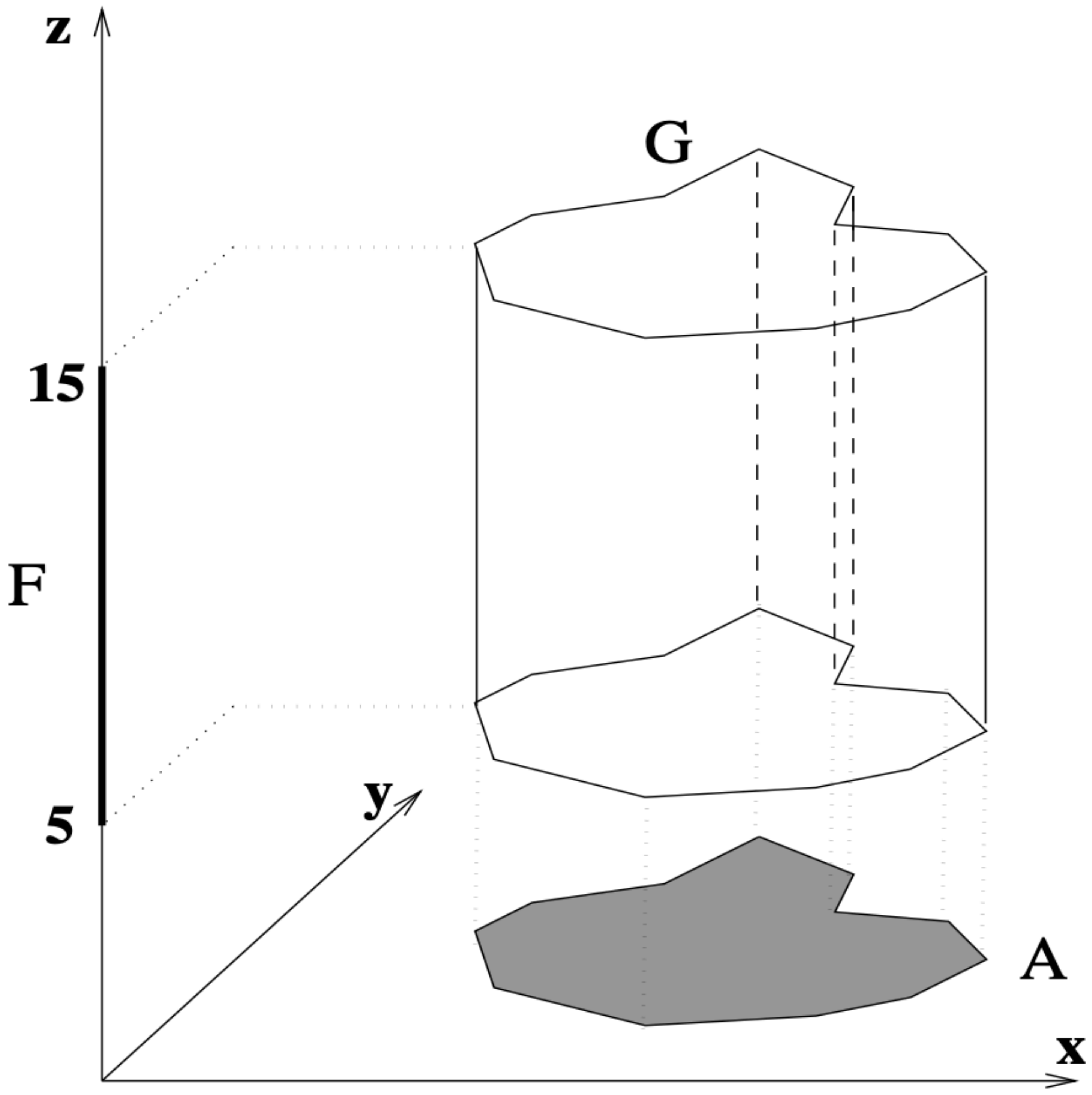


Difference

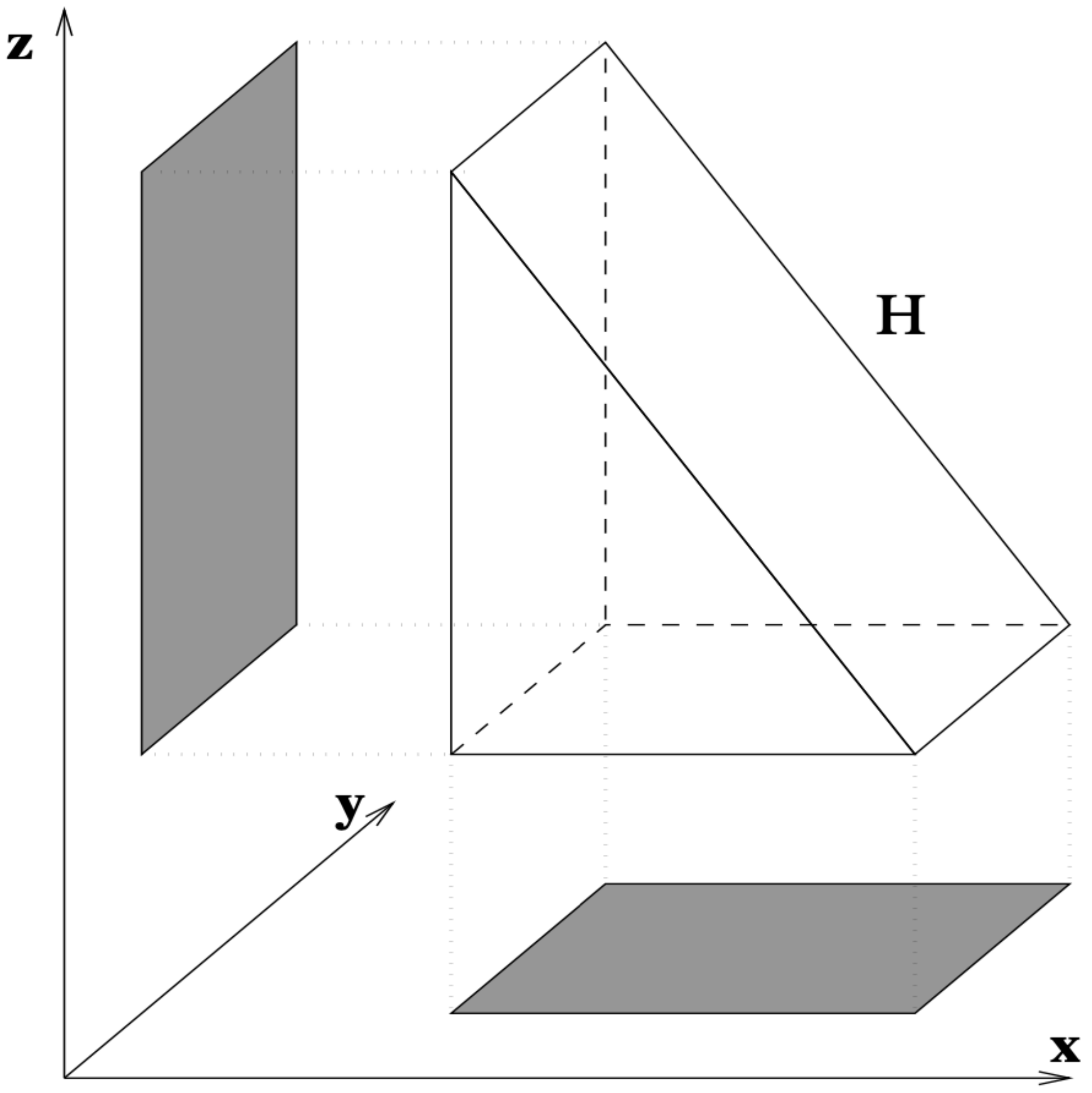


Union

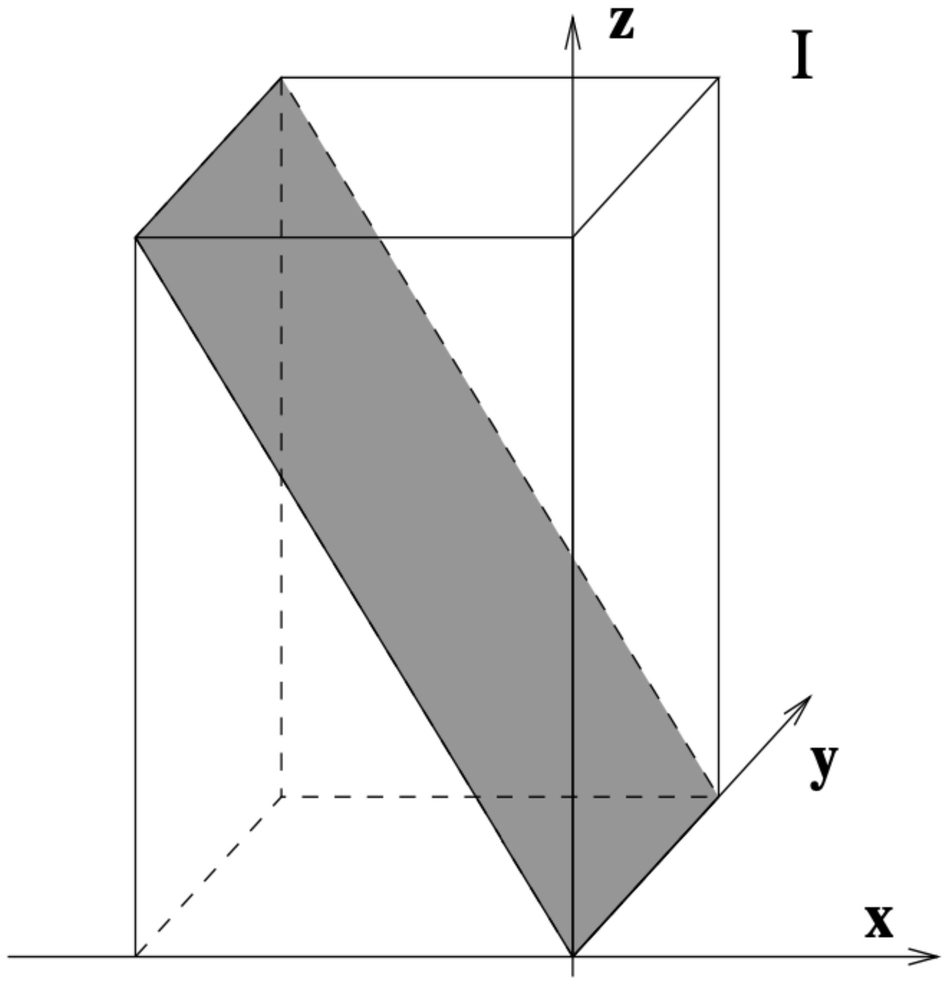



z
G
15
F
5
y
A
x

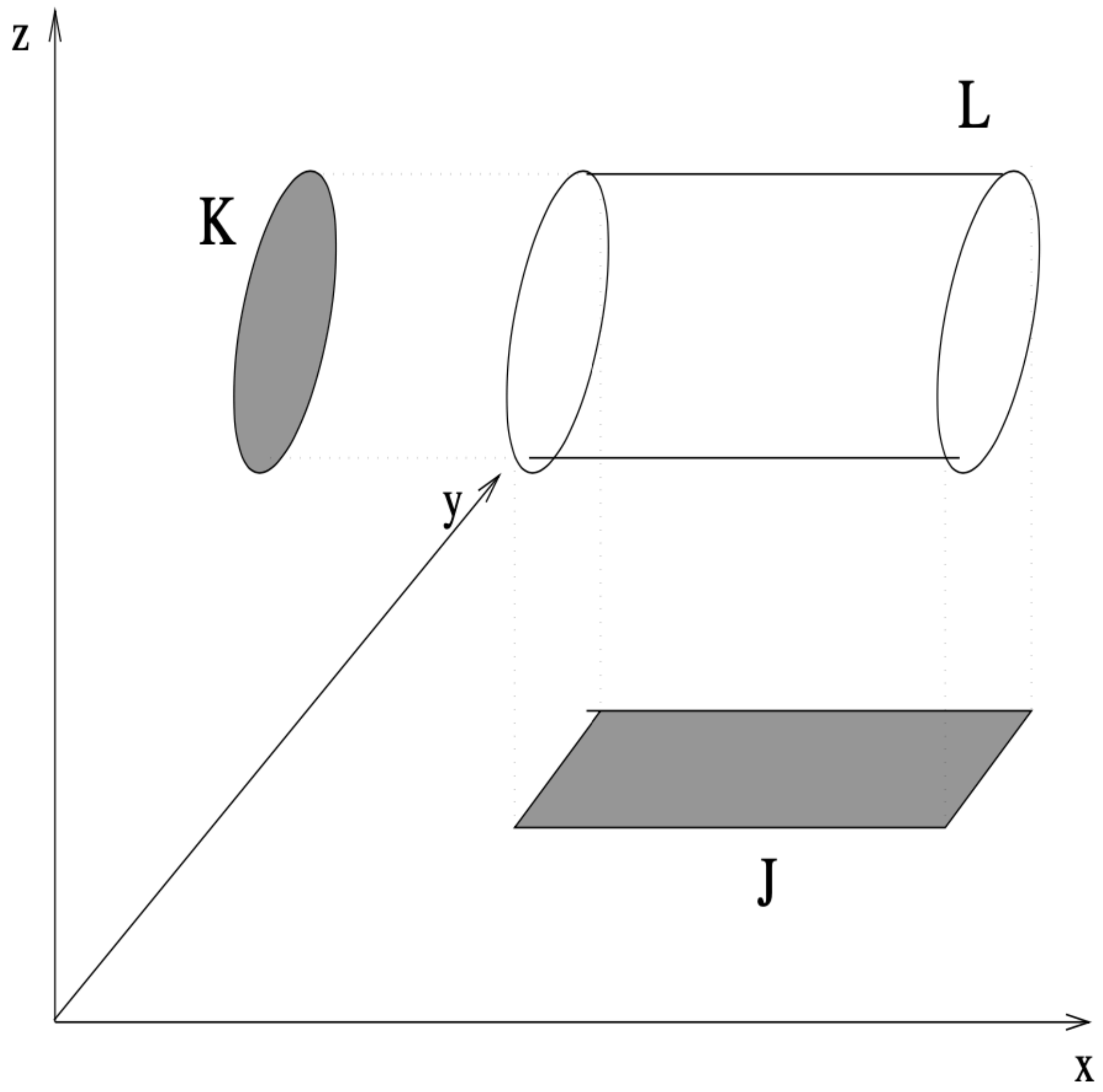
Projection



Selection

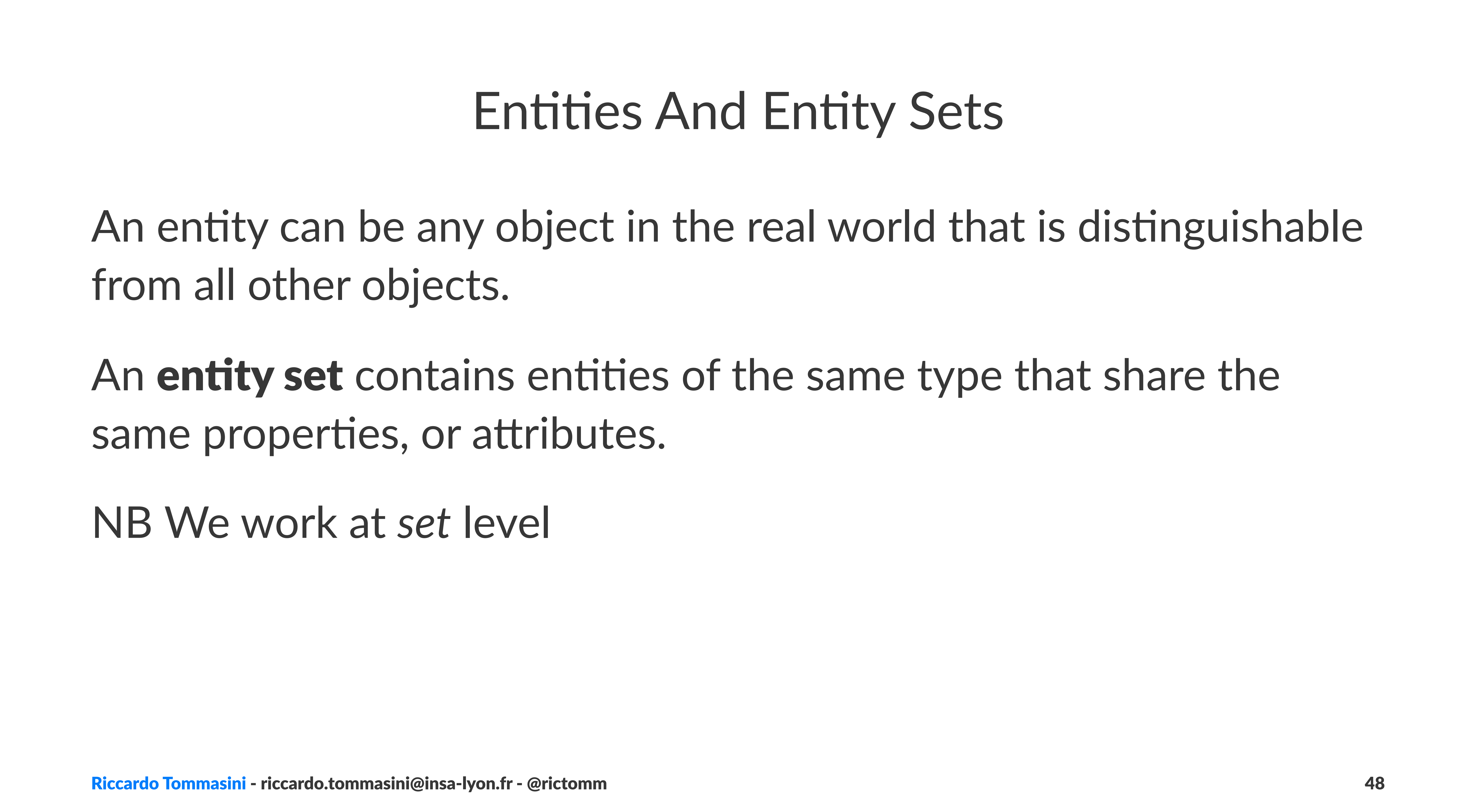


Natural JOIN



Entity-Relationship (ER) Model

- Outputs a conceptual schema.
- The ER data model employs three basic concepts:
  - entity sets
  - relationship sets and
  - attributes.
- It is also associated with diagrammatic representation try out



Entities And Entity Sets

An entity can be any object in the real world that is distinguishable from all other objects.

An **entity set** contains entities of the same type that share the same properties, or attributes.

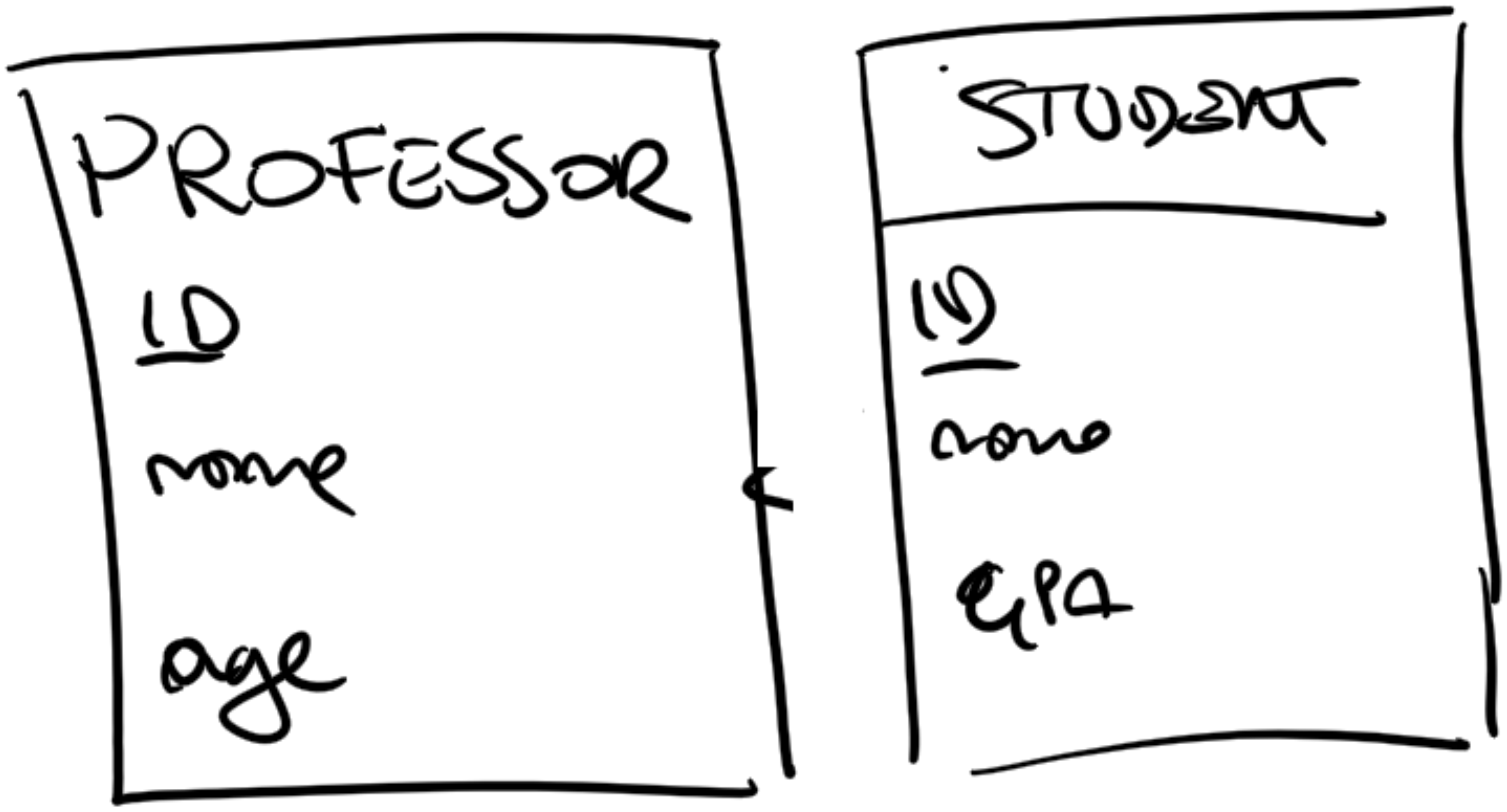
NB We work at *set* level

Ask the students

Examples of entities:

 - University
 - Department
 - Persons
 - Courses
- Examples of entity sets
 - Professors and Students
 - Data Science coruses: curriculms

Syntax



fields are what we call attribtues

Relationships and Relationship Sets

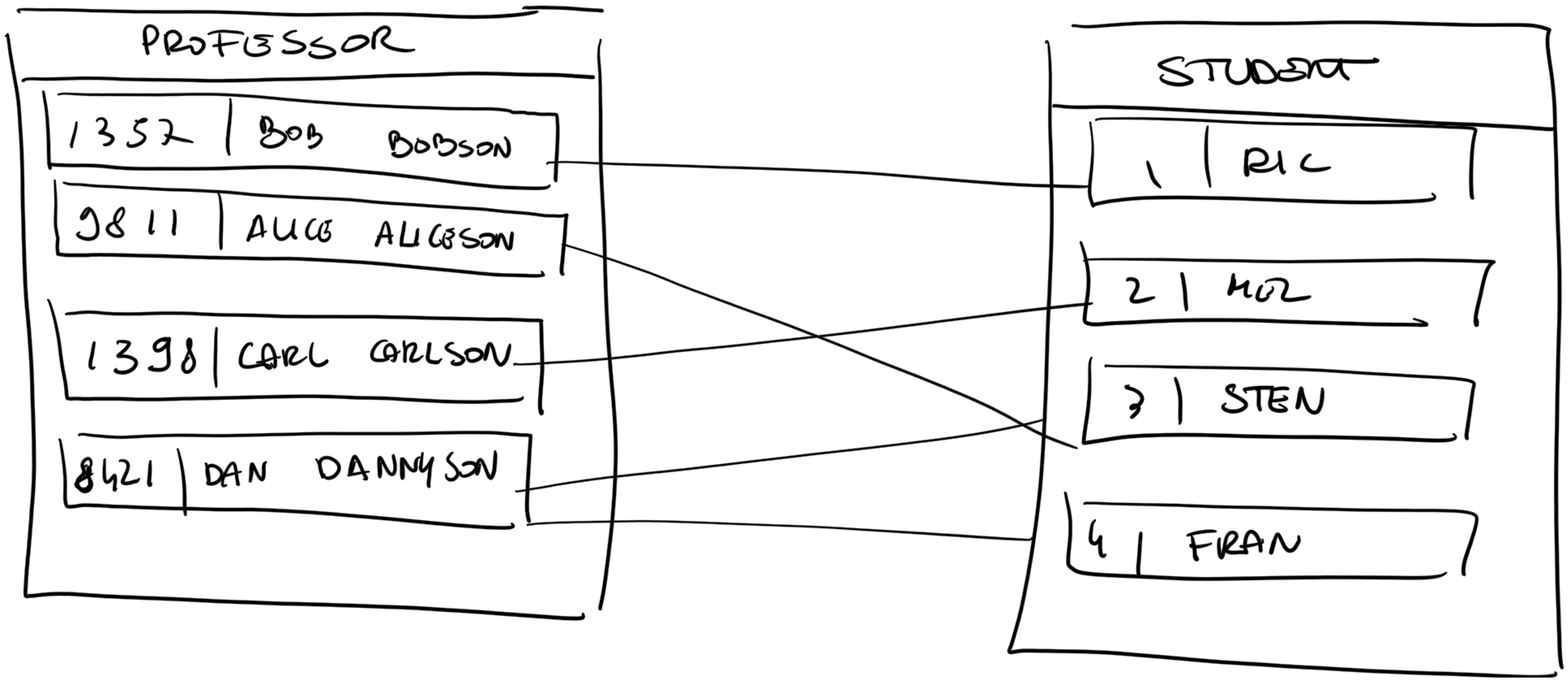
A **relationship** is an association among several entities.

A **relationship set** is a set of relationships of the same type.

Examples of entities:

- advisor
- attendee
- enrollment

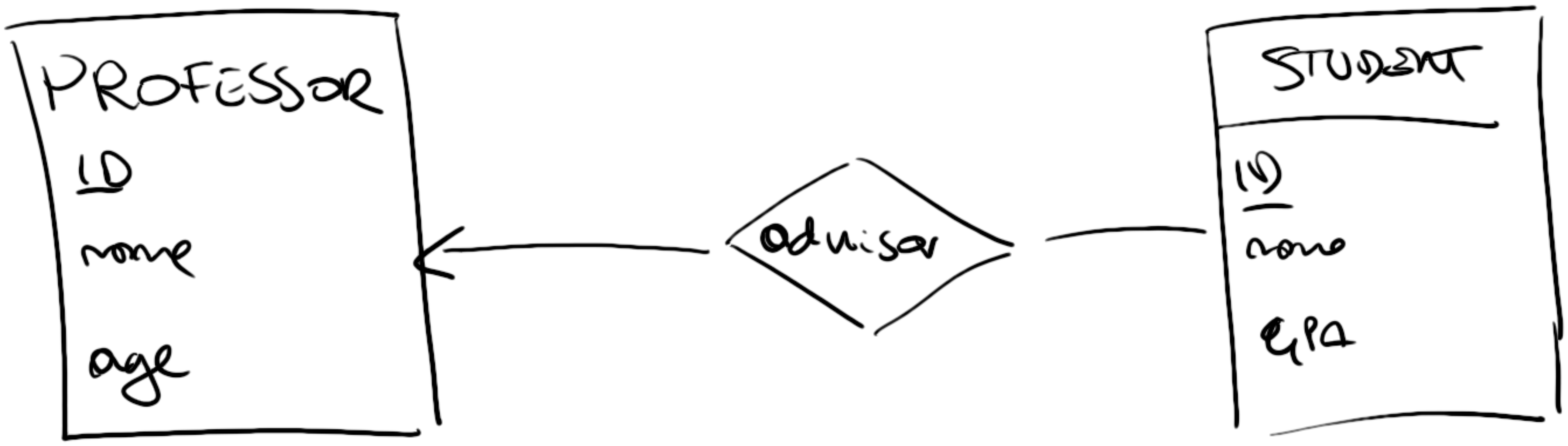
Intution



PROFESSOR	
1352	BOB BOBSON
9811	ALICE ALICESON
1398	CARL CARLSON
8421	DAN DANNYSON

STUDENT	
1	RIC
2	MOZ
3	STEN
4	FRAN

Syntax



ER works under the assumption that most relationship sets in a database system are binary. Relationships between more than two entity sets are rare.

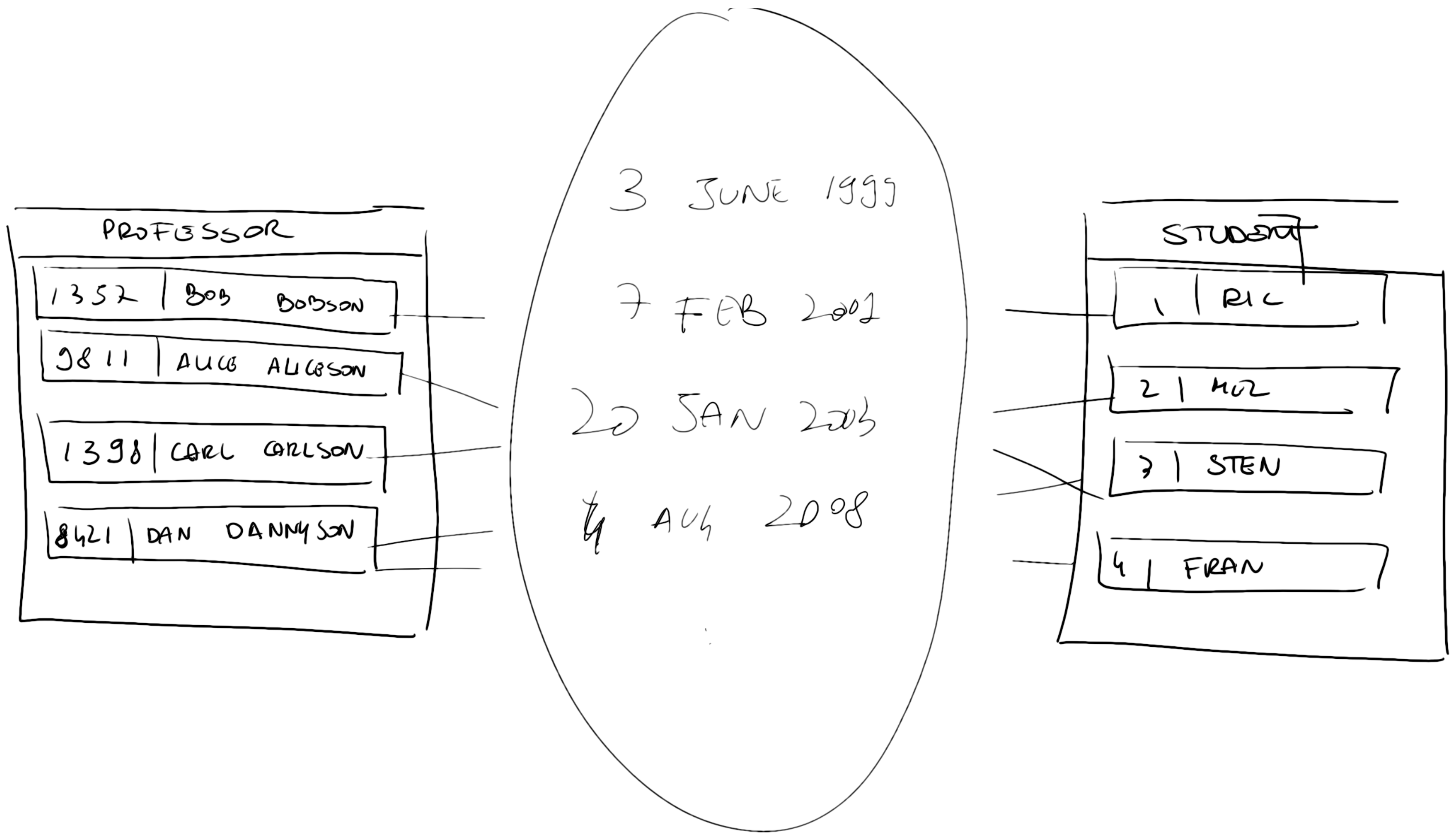
Attributes and Values

attributes. Attributes are descriptive properties possessed by each member of an entity set.

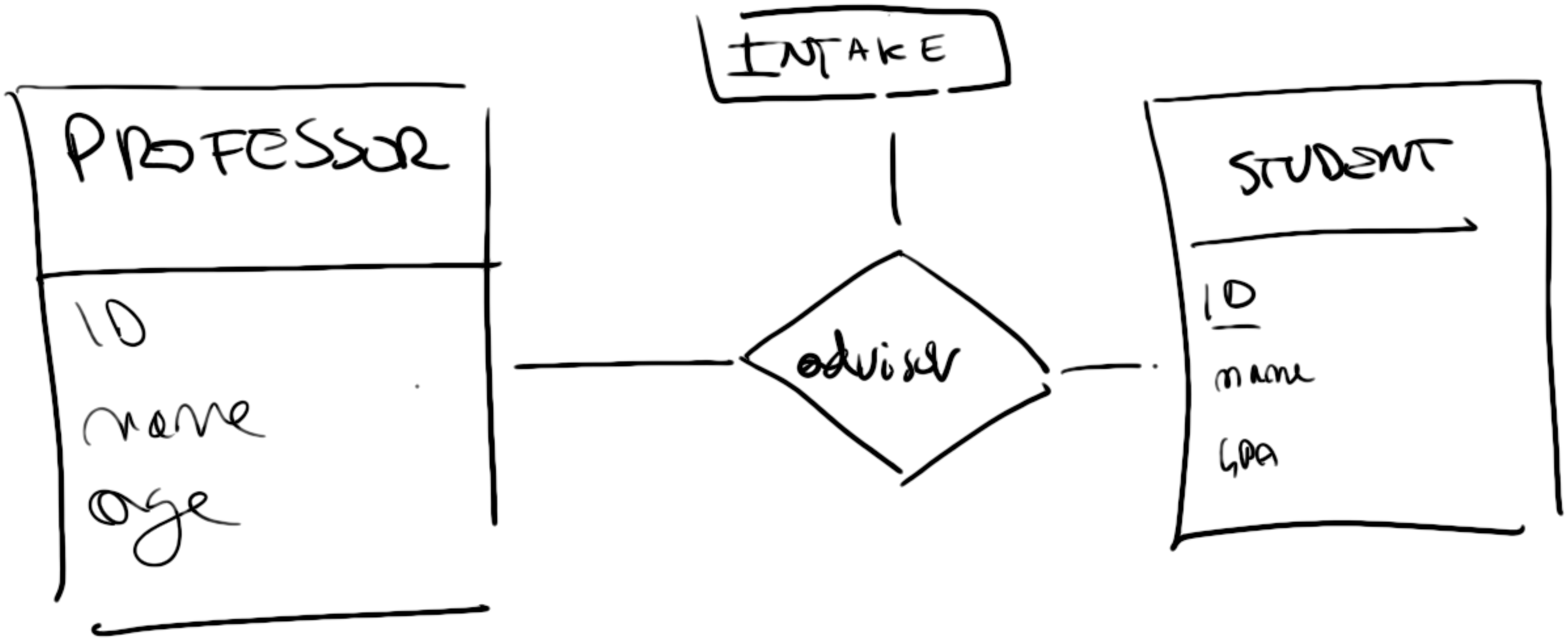
Each entity has a **value** for each of its attributes.

Also relationshis may have attributes called **descriptive attributes**.

Intution


PROFESSOR
1352 | BOB BOBSON
9811 | ALICE ALICESON
1398 | CARL CARLSON
8421 | DAN DANNYSON
3 JUNE 1999
7 FEB 2001
20 JAN 2003
4 AUG 2008
STUDENT
1 | RIC
2 | MOZ
3 | STEN
4 | FRAN

Syntax



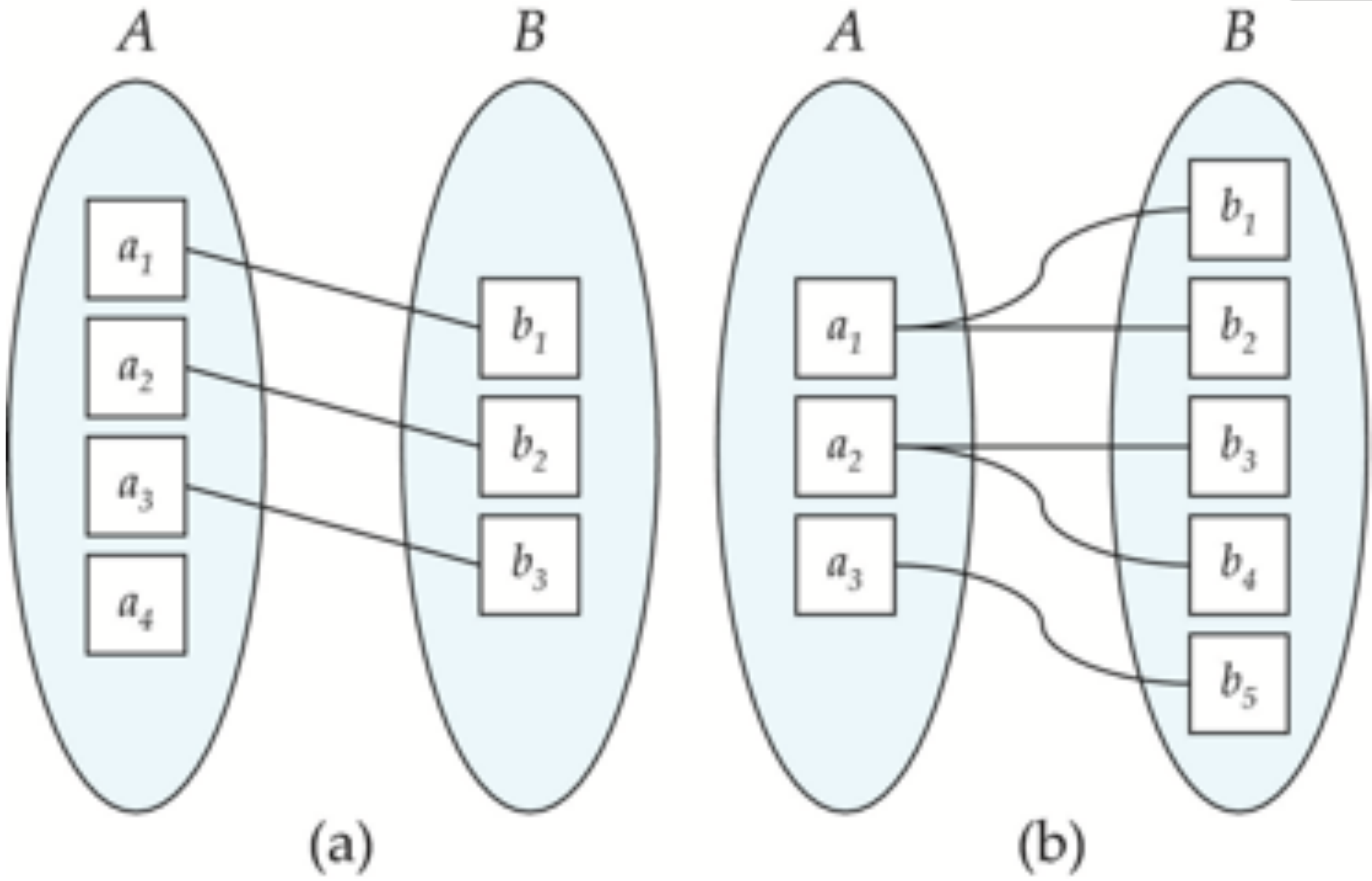
Cardinality

For a binary relationship set the mapping cardinality must be one of the following types:

- One to one
- One to many
- Many to one
- Many to many

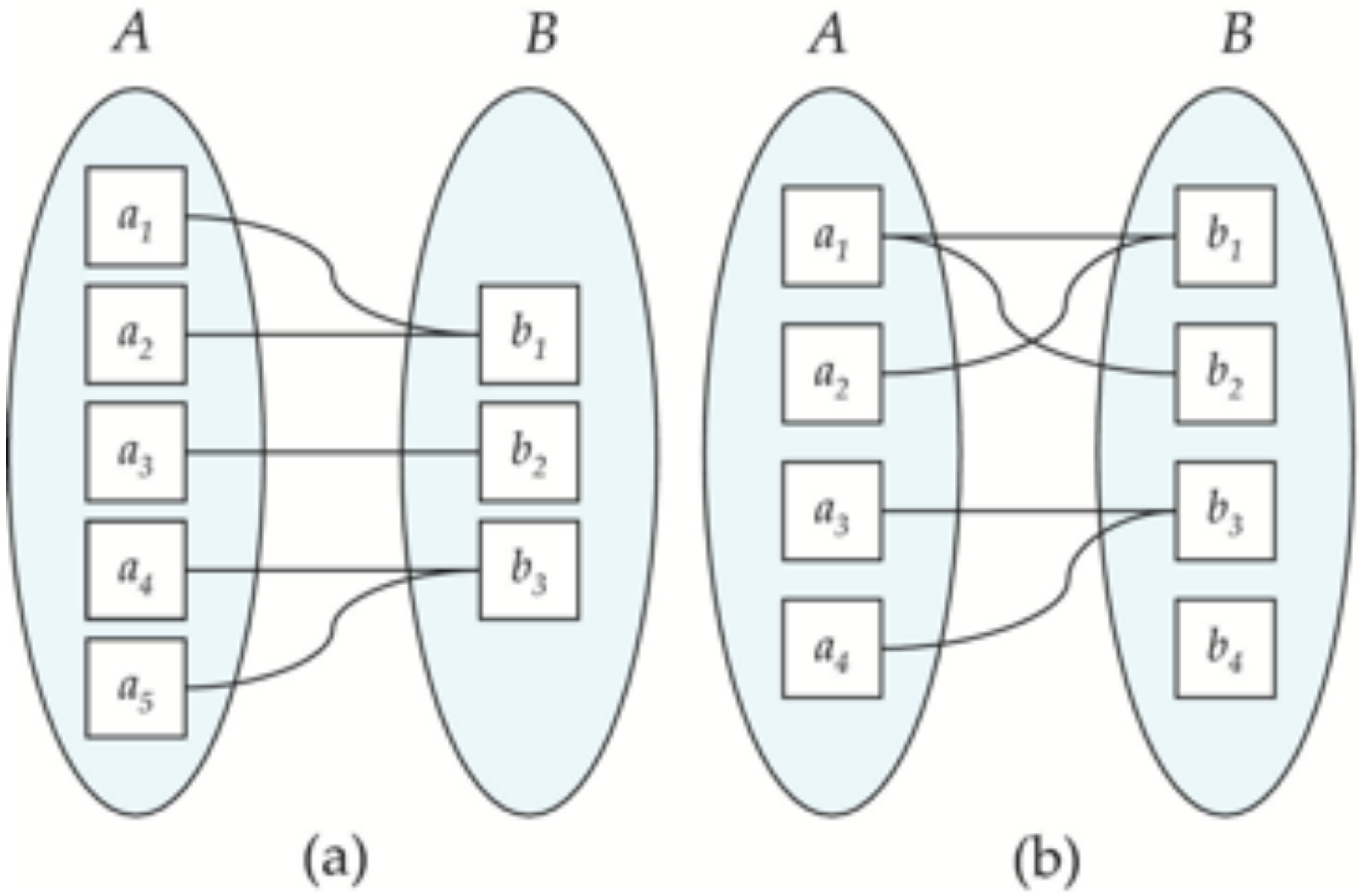
Cardinality Visualized

- (a) One to One
- (b) One to Many

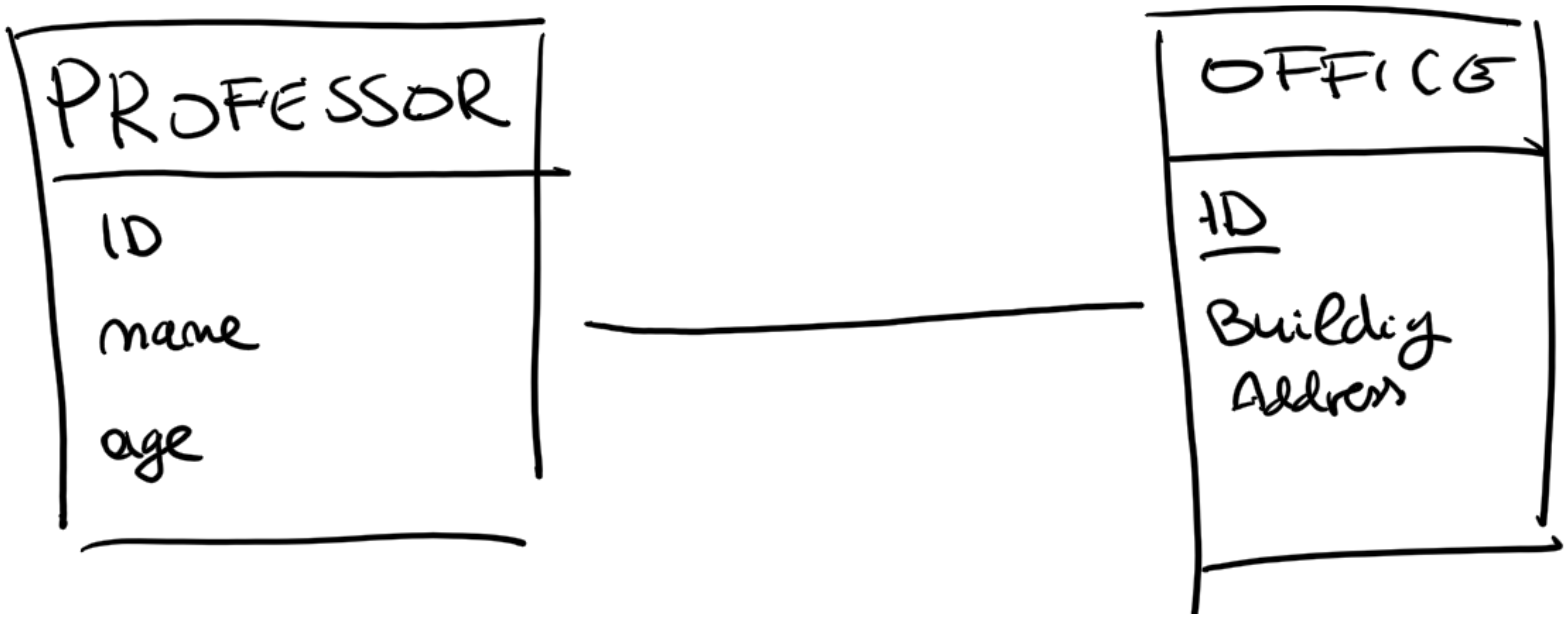


Cardinality Visualized

- (a) Many to One
- (b) Many to Many

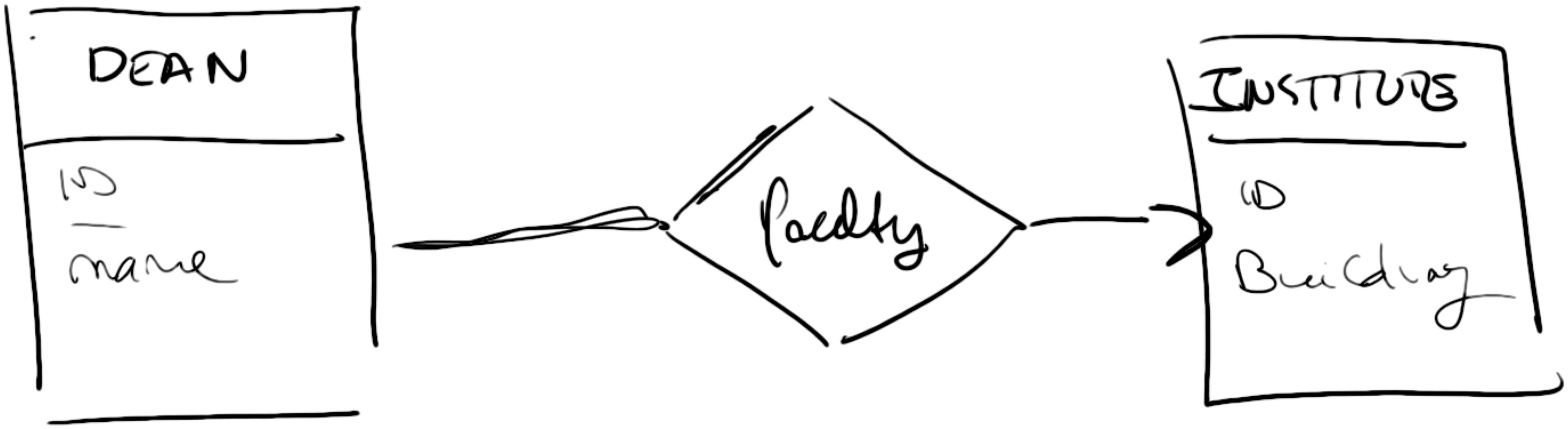


One to Many



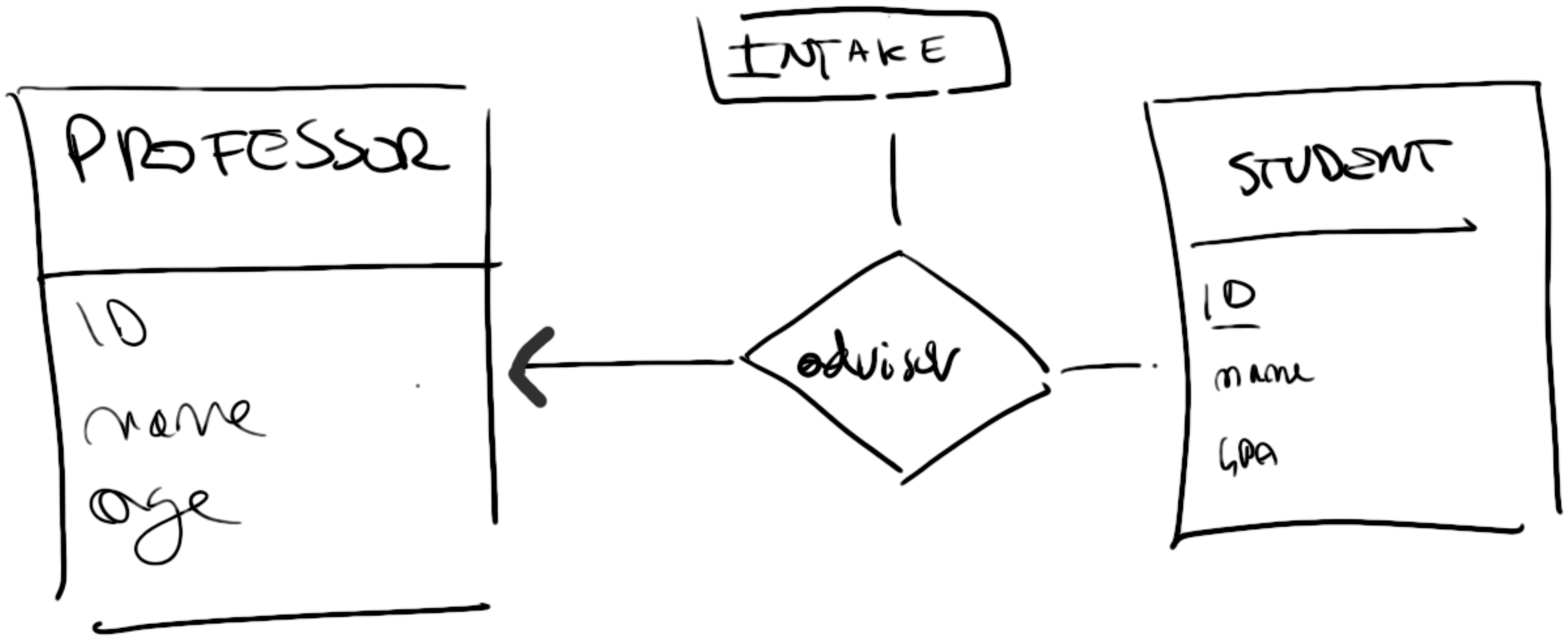
A (full) professor has one office
an office hosts one full professor

One to Many



A Dean is associated with many institutes
An Institute has only one dean

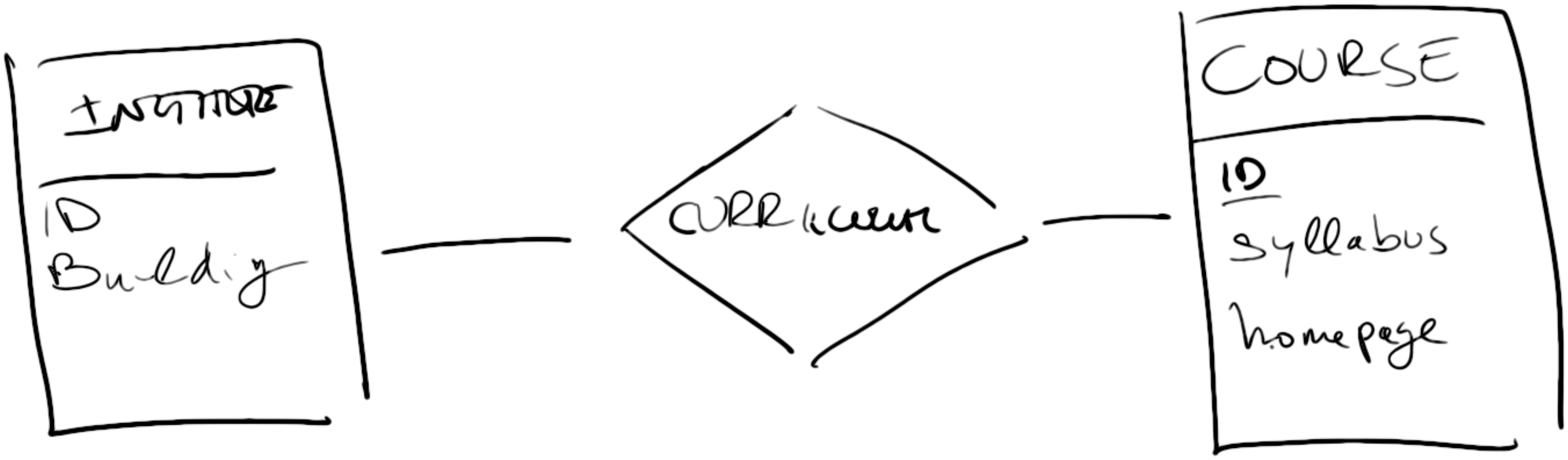
Many to One



A professor advises many students but a student has only one advisor.

Many students share the same advisor but they only have one.

Many to Many



A course is associated to many insitute in the context of a curriculum
An institute offers many courses within a curriculum

Keys

- Provide a way to specify how entities and relations are distinguished.
- *Primary key* for Entity Sets
    - By definition, individual entities are distinct (set)
    - From database perspective, the differences among them must be expressed in terms of their attributes
- *Primary Key* for Relationship Sets
    - We use the individual primary keys of the entities in the relationship set.
    - The choice depends on the mapping cardinality of the relationship set.

Choice of Primary key for Binary Relationship

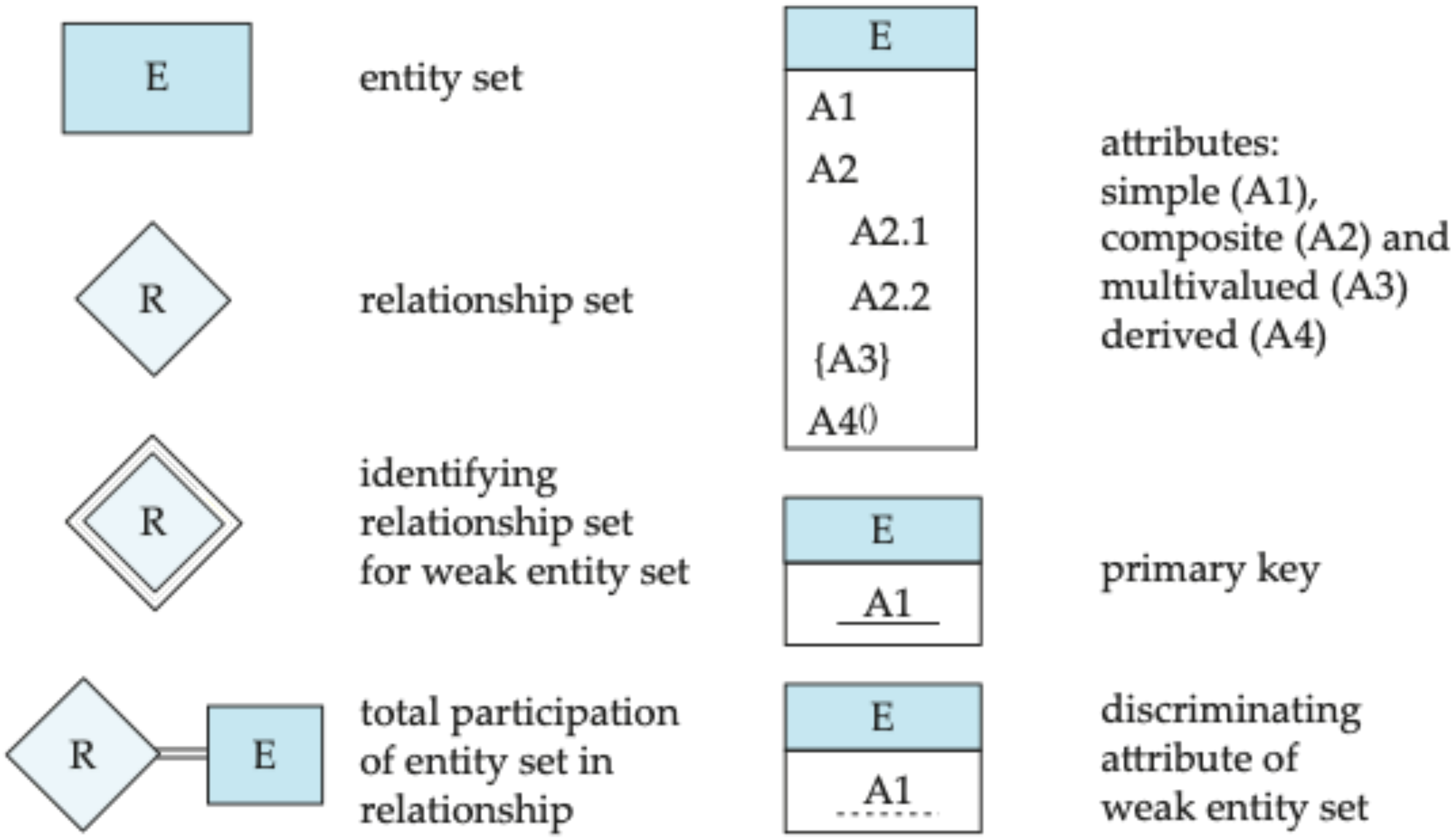
- One-to-one relationships. The primary key of either one of the participating entity sets forms a minimal superkey, and either one can be chosen as the primary key.
- One-to-Many relationships and Many-to-one relationships
  - The primary key of the “Many” side is a minimal superkey and is used as the primary key.
- Many-to-Many relationships:
  - The preceding union of the primary keys is a minimal superkey and is chosen as the primary key.

Weak Entity Sets

- A weak entity set is one whose existence is dependent on another entity, called its **identifying entity**

- A weak entity set is one whose existence is dependent on another entity, called its identifying entity

Summary of Symbols



entity set

relationship set

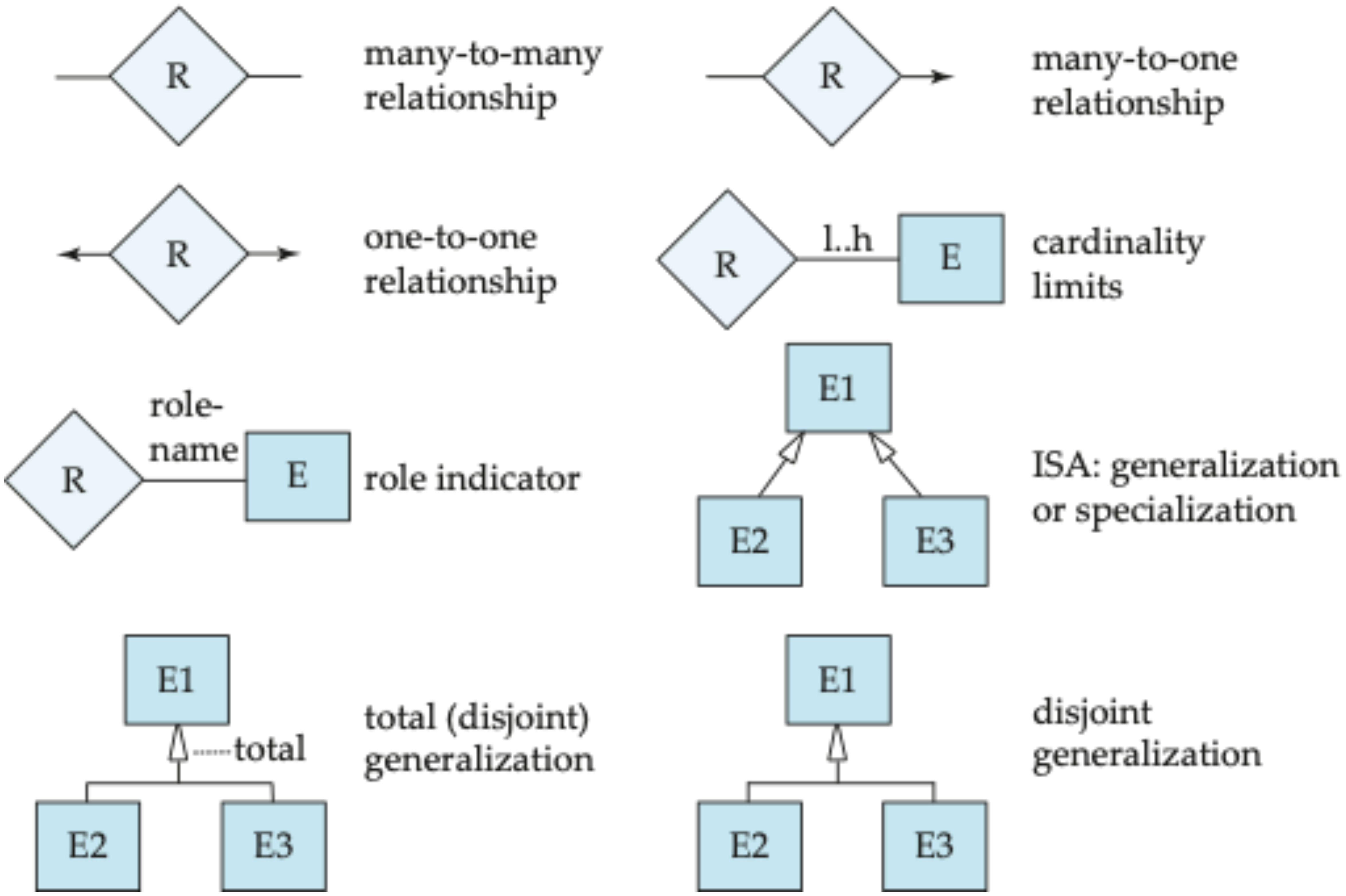
identifying relationship set for weak entity set

total participation of entity set in relationship

attributes:
simple (A1),
composite (A2) and
multivalued (A3)
derived (A4)

primary key

discriminating attribute of weak entity set


R
many-to-many relationship
R
many-to-one relationship
R
one-to-one relationship
R
l..h
E
cardinality limits
R
role-name
E
role indicator
E1
E2
E3
ISA: generalization or specialization
E1
total
E2
E3
total (disjoint) generalization
E1
E2
E3
disjoint generalization

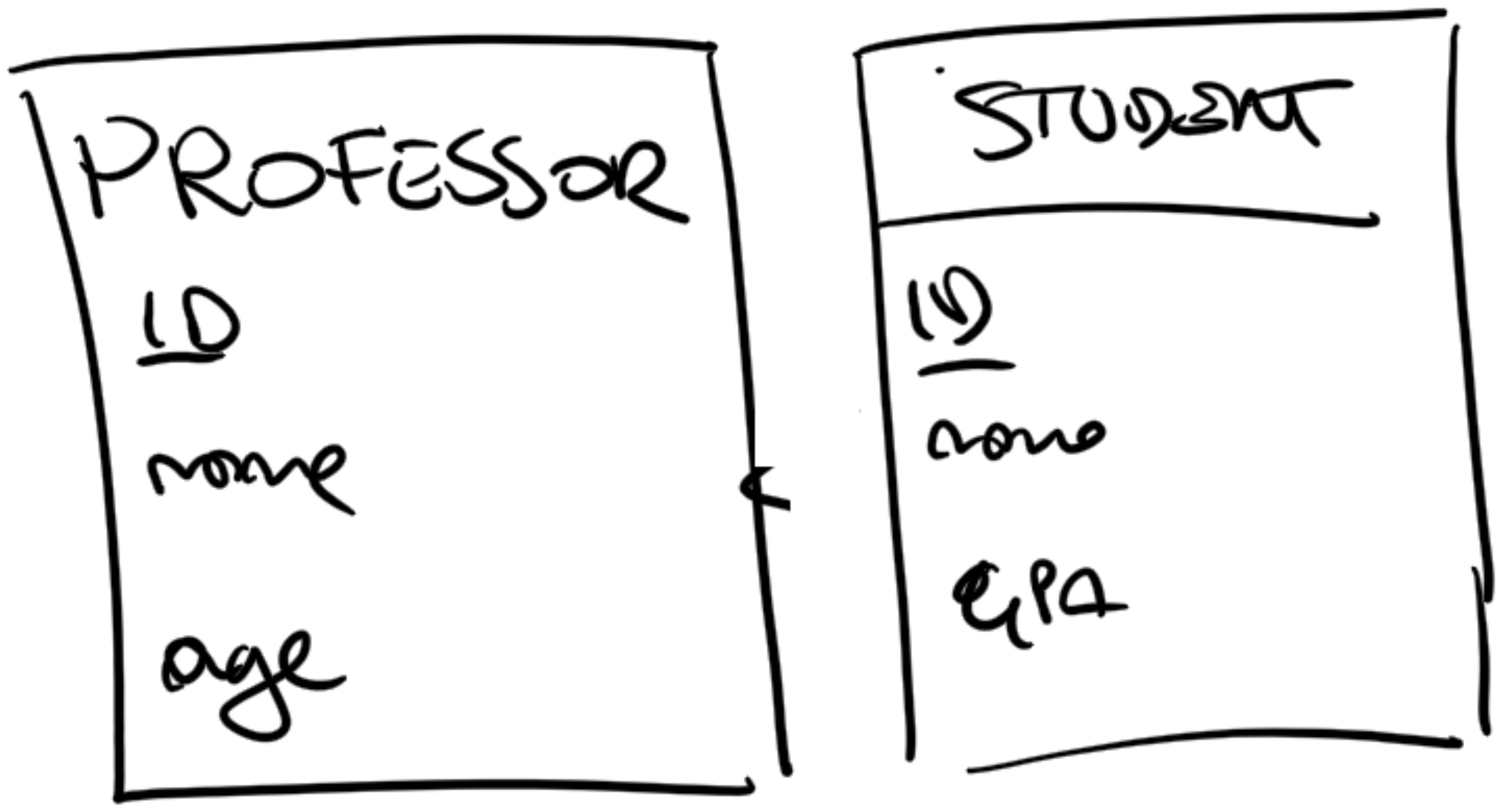
From ER to Relational Model

- Entity and relationship sets can be expressed as relation schemas that represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a
  collection of schemas.

Reduction of Entities

- For each **entity** set there is a unique schema with the same name
- Each schema has a number of columns (generally corresponding to attributes), which have unique names

Professor(<u>ID</u>,Name,Age)
Student(<u>ID</u>,Name,GPA)

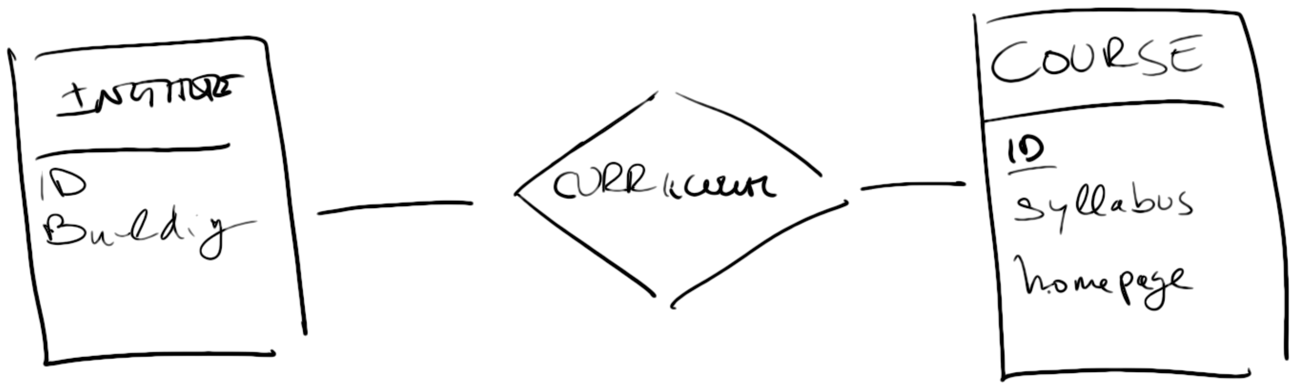


Weak entities set becomes a relation that includes a column for the primary
key of the identifying entity.

Reduction of Relationships

- For each **relationship** set there is a unique schema with the same name

- A **many-to-many** relationship (figure) is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.



Curriculum(<u>Institute_ID</u>,<u>Course_ID</u>)

Reduction of Relationships

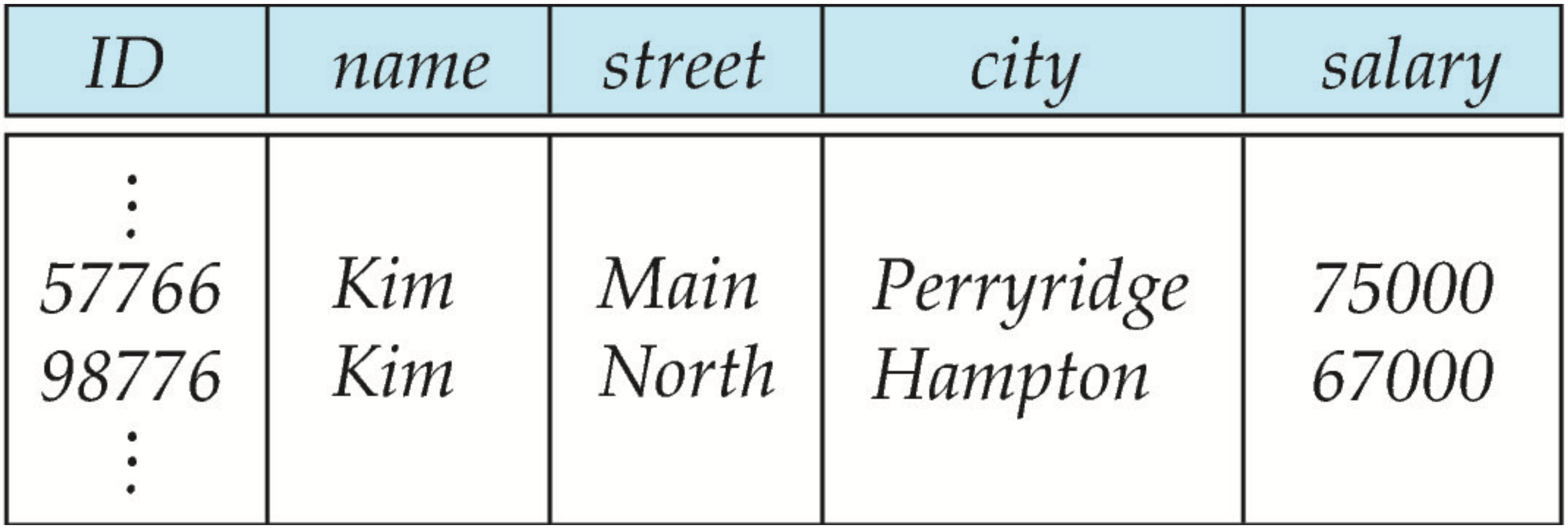
- **Many-to-one** and one-to-many** relationship can be represented by adding an extra attribute to the "many" side
- For **one-to-one** relationship, either side can be chosen to act as the "many" side

Normalisation

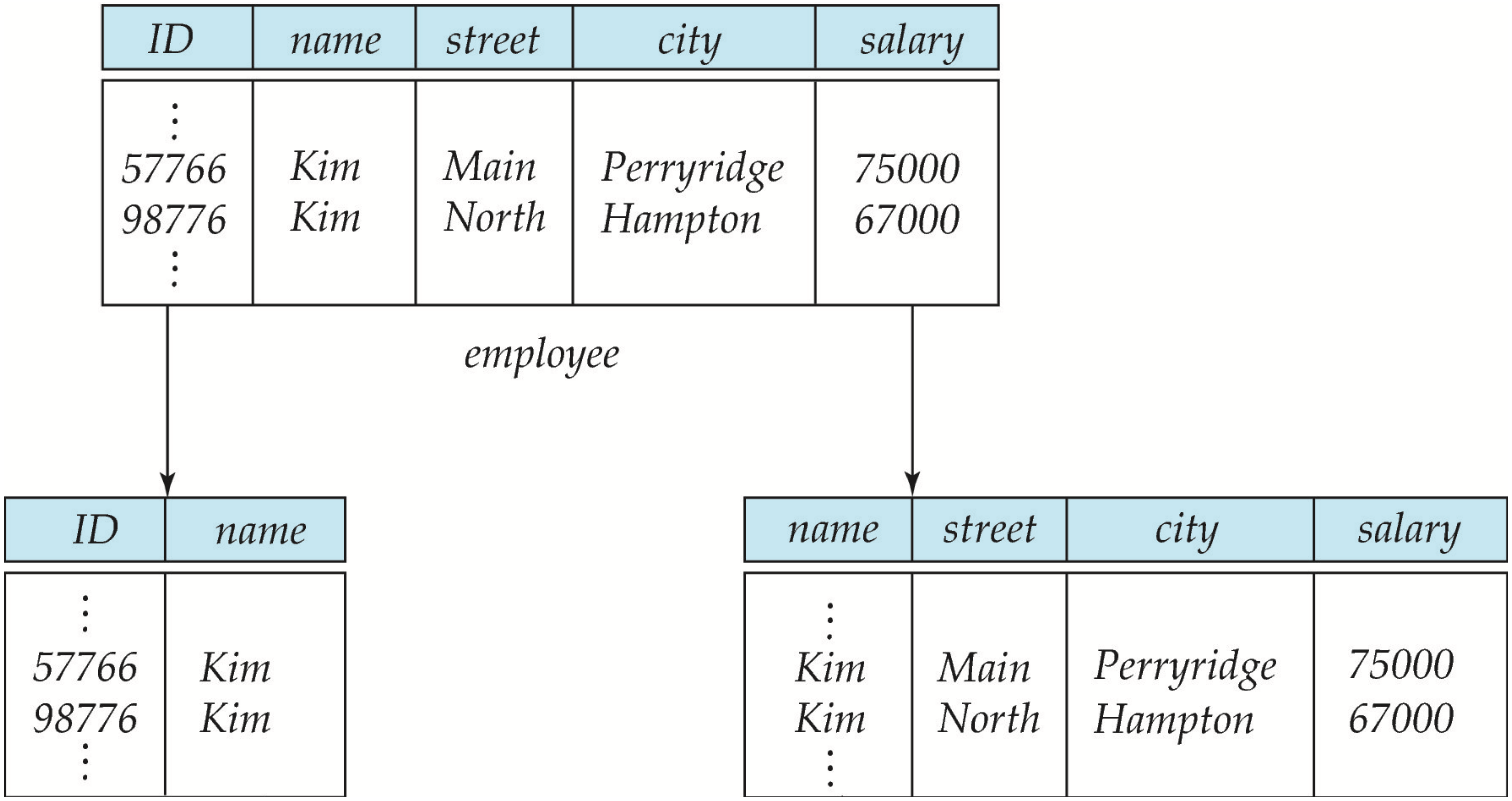
- Typically decomposes tables to avoid redundancy
- Spans both logical and physical database design
- Aims at **improving** the database design

Goals

- Make the schema informative
- Minimize information duplication
- Avoid modification anomalies
- Disallow spurious tuples



ID	*name*	*street*	*city*	*salary*
⋮				
57766	Kim	Main	Perryridge	75000
98776	Kim	North	Hampton	67000
⋮				

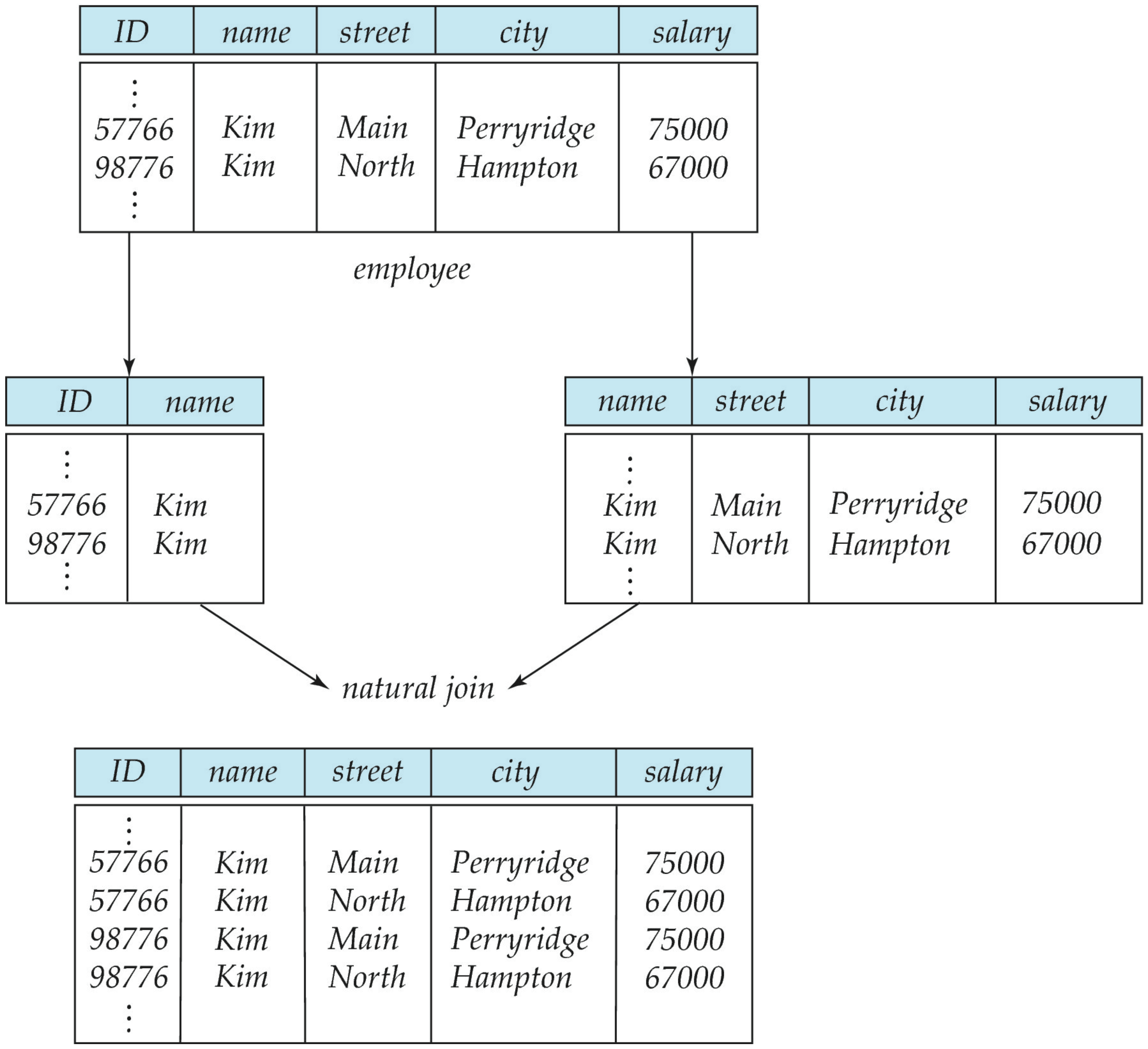


ID	name	street	city	salary
⋮				
57766	Kim	Main	Perryridge	75000
98776	Kim	North	Hampton	67000
⋮				

employee

ID	name
⋮	
57766	Kim
98776	Kim
⋮	

name	street	city	salary
⋮			
Kim	Main	Perryridge	75000
Kim	North	Hampton	67000
⋮			



ID	name	street	city	salary
⋮				
57766	Kim	Main	Perryridge	75000
98776	Kim	North	Hampton	67000
⋮				

employee

ID	name
⋮	
57766	Kim
98776	Kim
⋮	

name	street	city	salary
⋮			
Kim	Main	Perryridge	75000
Kim	North	Hampton	67000
⋮			

natural join

ID	name	street	city	salary
⋮				
57766	Kim	Main	Perryridge	75000
57766	Kim	North	Hampton	67000
98776	Kim	Main	Perryridge	75000
98776	Kim	North	Hampton	67000
⋮				

Normal Forms (Refresh)

- First Normal Form (1NF)
  - A table has only atomic valued clumns.
  - Values stored in a column should be of the same domain
  - All the columns in a table should have unique names.
  - And the order in which data is stored, does not matter.
- Second Normal Form (2NF)
  - A table is in the First Normal form and every non-prime attribute is fully functional dependent[^33] on the primary key
- Third Normal Form (3NF)
  - A table is in the Second Normal form and every non-prime attribute is non-transitively dependent on every key

  [^33]: $X \rightarrow Y, \forall A \in X((X - A) \nrightarrow Y)$

Modelling for Database: A note on Storage

- Storage is laid out in a row-oriented fashion
- For relational this is as close as the the tabular representation
- All the values from one row of a table are stored next to each other.
- This is true also for some NoSQL (we will see it again)
    - Document databases stores documents a contiguous bit sequence

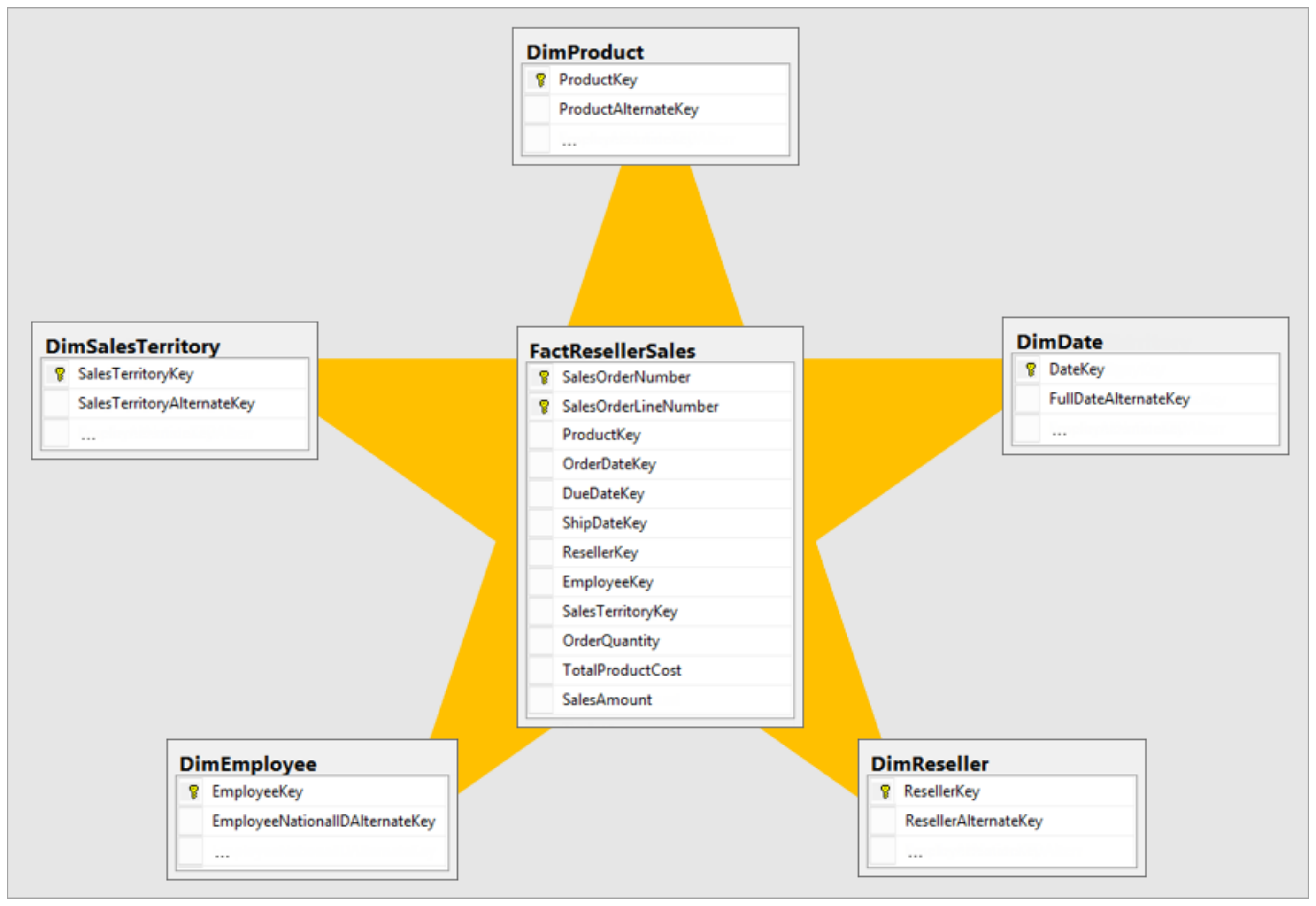
Data Modelling for Data Warehouses

- Works in phases related to the aforementioned levels of abstractions
- Less diversity in the data model, usually relational in the form of a star schema (also known as dimensional Modelling[41]).
- Redundancy and incompleteness are not avoided, fact tables often have over 100 columns, sometimes several hundreds.
- Optimized for OLAP

- The data model of a data warehouse is most commonly relational, because SQL is generally a good fit for analytic queries.
- Do not associate SQL with analytic, it depends on the data Modelling.

[41] Ralph Kimball and Margy Ross: The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modelling, 3rd edition. John Wiley & Sons, July 2013. ISBN: 978-1-118-53080-1

A Star is Born



DimProduct
- 🔑 ProductKey
- ProductAlternateKey
- ...

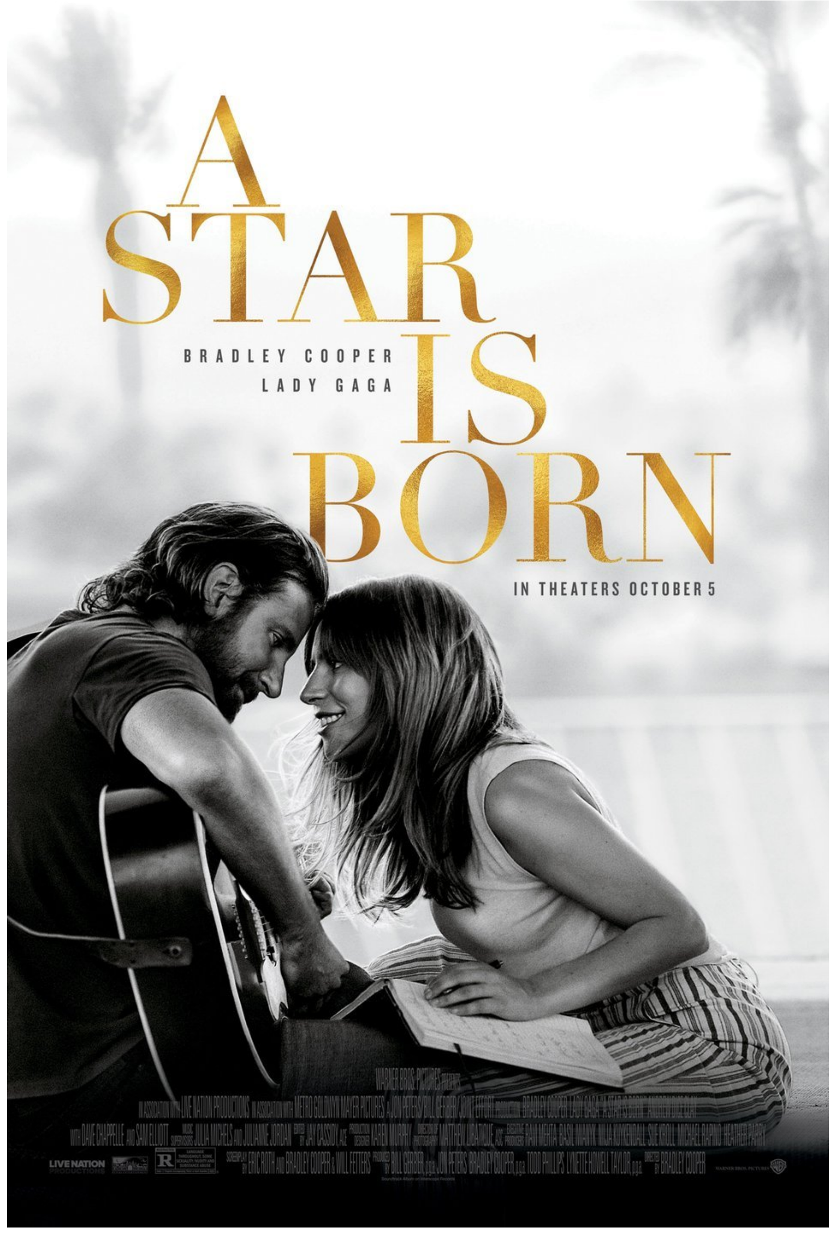
DimSalesTerritory
- 🔑 SalesTerritoryKey
- SalesTerritoryAlternateKey
- ...

FactResellerSales
- 🔑 SalesOrderNumber
- 🔑 SalesOrderLineNumber
- ProductKey
- OrderDateKey
- DueDateKey
- ShipDateKey
- ResellerKey
- EmployeeKey
- SalesTerritoryKey
- OrderQuantity
- TotalProductCost
- SalesAmount

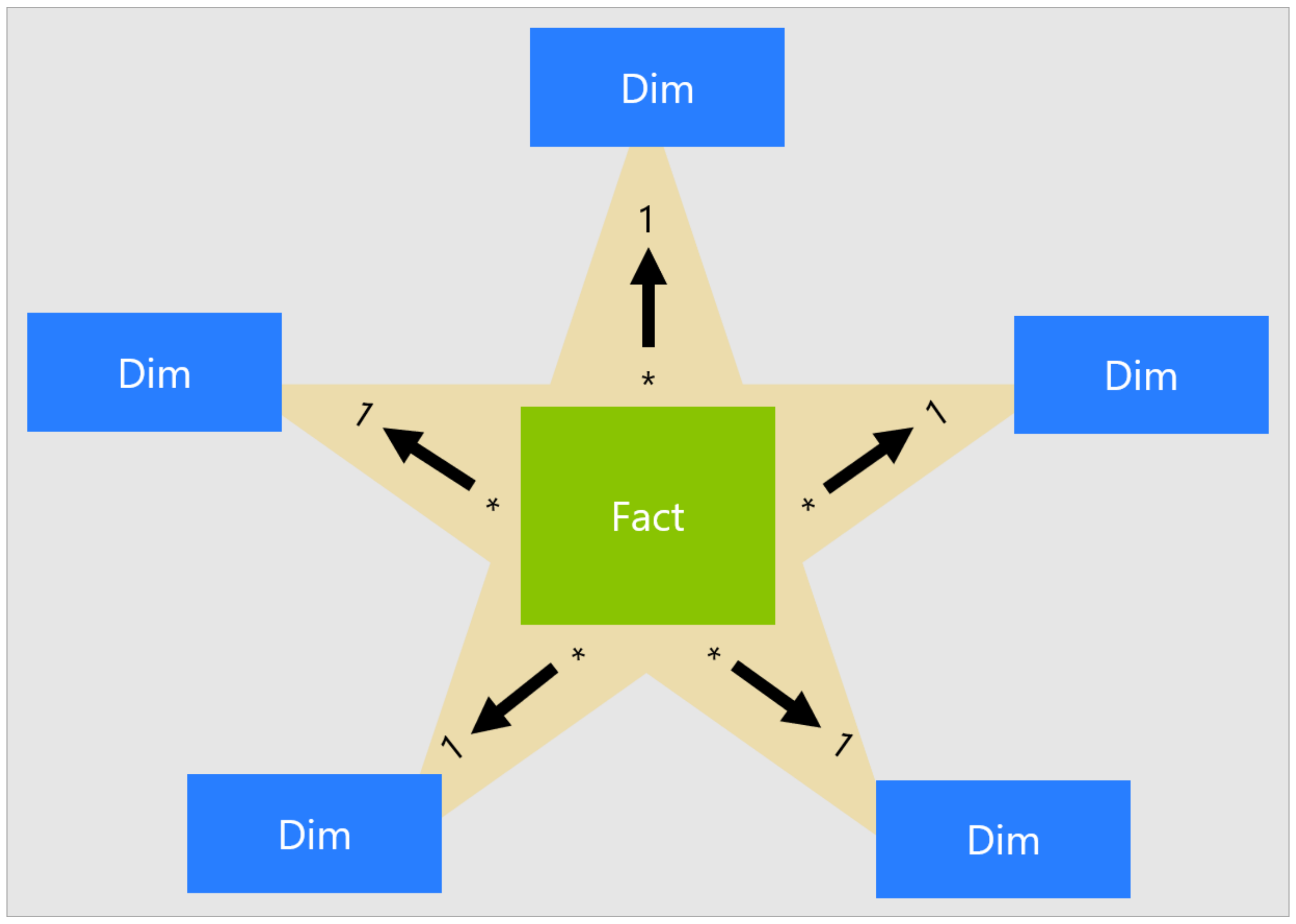
DimDate
- 🔑 DateKey
- FullDateAlternateKey
- ...

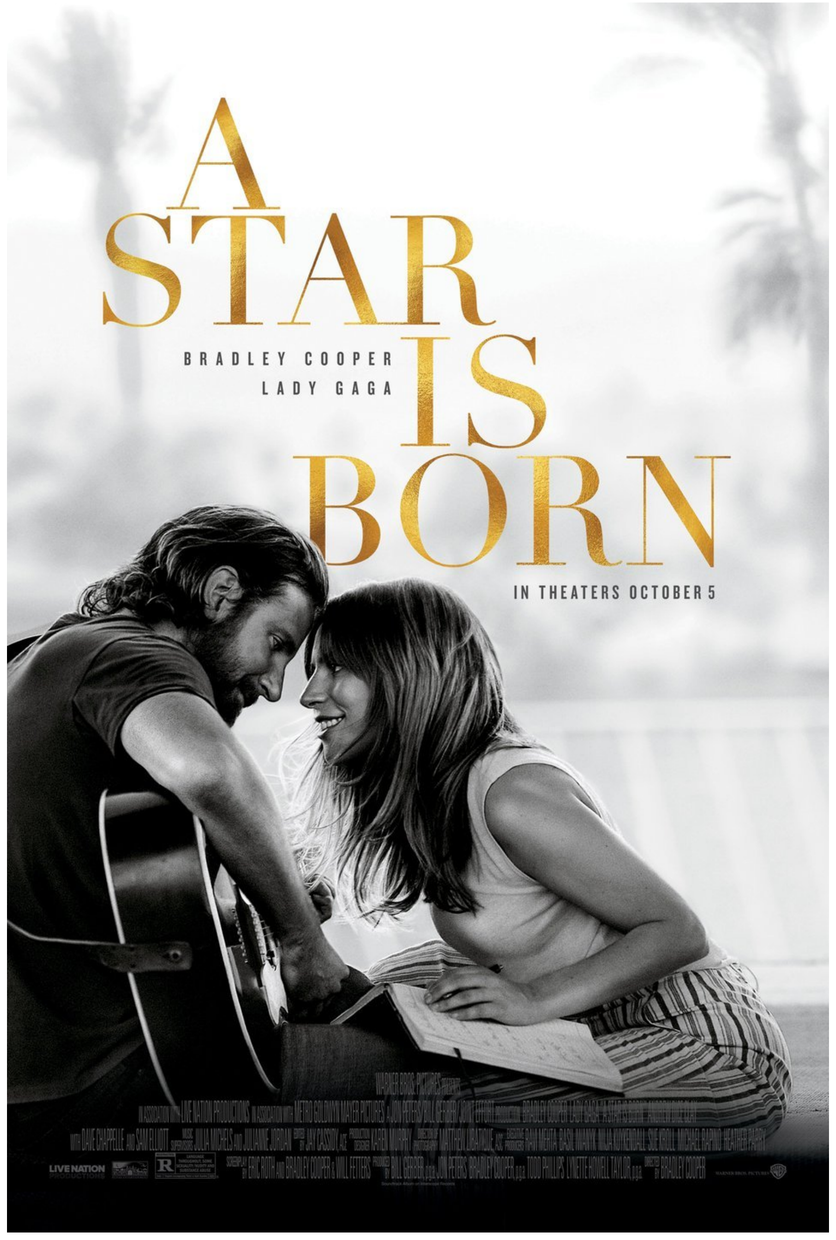
DimEmployee
- 🔑 EmployeeKey
- EmployeeNationalIDAlternateKey
- ...

DimReseller
- 🔑 ResellerKey
- ResellerAlternateKey
- ...



A Star is Born





Dimensional Modelling

A denormalized relational model
- made up of tables with attributes
- Relationships defined by keys and foreign keys

Four-Step Dimensional Design Process

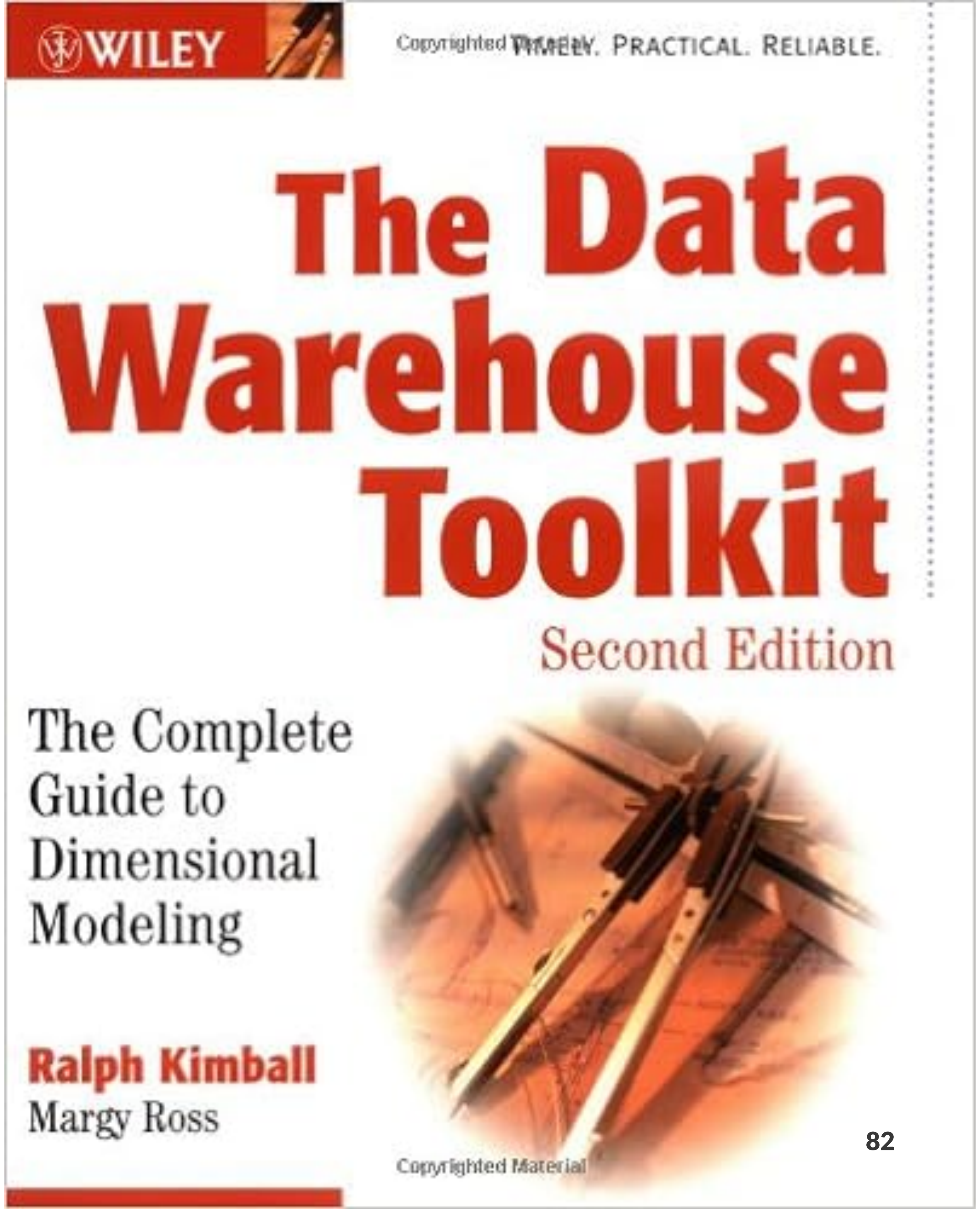
1. Select the business process.

2. Declare the grain.

3. Identify the dimensions.

4. Identify the facts.

Mandatory Read


WILEY
Copyrighted Material
TIMELY. PRACTICAL. RELIABLE.
The Data Warehouse Toolkit
Second Edition
The Complete Guide to Dimensional Modeling
Ralph Kimball
Margy Ross
Copyrighted Material

Business processes are crtical activities that your organization performs, e.g., registering students for a class.

The **grain** establishes exactly what a single fact table row represents. Three common grains categorize all fact tables: transactional, periodic snapshot, or accumulating snapshot.

Dimensions provide contex to business process events, e.g., who, what, where, when, why, and how.

:wq

Facts are the measurements that result from a business process event and are almost always numeric.

Comparison with DBMS

- One table per entity
- Minimise data redundancy
- Optimise update
- Processing Model: Transaction

- One fact table for data organization
- Maximise understandability
- Optimised for retrieval
- Processing Model: Analytical

Dimensional Modelling: Fact Table

A **fact table** contains the numeric measures produced by an operational measurement event in the real world.

A **single fact** table row has a one-to-one relationship to a measurement event as described by the fact table's grain.

A **surrogate key** is a unique identifier that you add to a table to support star schema Modelling. By definition, it's not defined or stored in the source data

Dimensional Modelling: Dimension Table

Dimension tables contain the descriptive attributes used by BI applications for filtering and grouping the facts.

1 in a 1-M relationship with Fact Table

Every dimension table has a single **primary key** column , which is embedded as a foreign key in any associated fact table.

Dimensional Modelling: Slowly/Rapidly Changing Dimensions

A **slowly changing dimension** (SCD) is a dimension "Dimension (data warehouse)") that stores data which, while generally stable, may change over time, often in an unpredictable manner.

Common examples of SCDs include geographical locations, customer details, or product attributes.

A **rapidly changing dimension**, such as transactional parameters like customer ID, product ID, quantity, and price, which undergo frequent updates.

The 5/10 Essential Rules of Dimensional Modelling (Read)[42]

1. Load detailed atomic data into dimensional structures.
2. Structure dimensional models around business processes.
3. Ensure that every fact table has an associated date dimension table.
4. Ensure that all facts in a single fact table are at the same grain or level of detail.
5. Resolve many-to-many relationships in fact tables.

[42] https://www.kimballgroup.com/2009/05/the-10-essential-rules-of-dimensional-Modelling/

The 10/10 Essential Rules of Dimensional Modelling (Read)[42]

1. Resolve many-to-one relationships in dimension tables.
2. Store report labels and filter domain values in dimension tables.
3. Make certain that dimension tables use a surrogate key.
4. Create conformed dimensions to integrate data across the enterprise.
5. Continuously balance requirements and realities to deliver a DW/BI solution that's accepted by business users and that supports their decision-making.

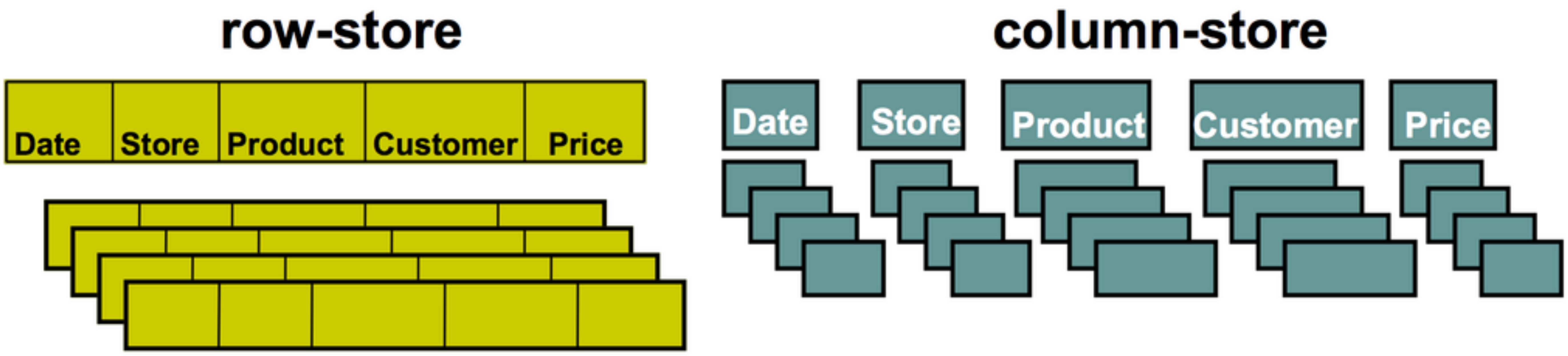
[42] https://www.kimballgroup.com/2009/05/the-10-essential-rules-of-dimensional-Modelling/

The Traditional RDBMS Wisdom Is (Almost Certainly) All Wrong[43]

[43] Source with slides: The Traditional RDBMS Wisdom Is (Almost Certainly) All Wrong," presentation at EPFL, May 2013

A note on Storage

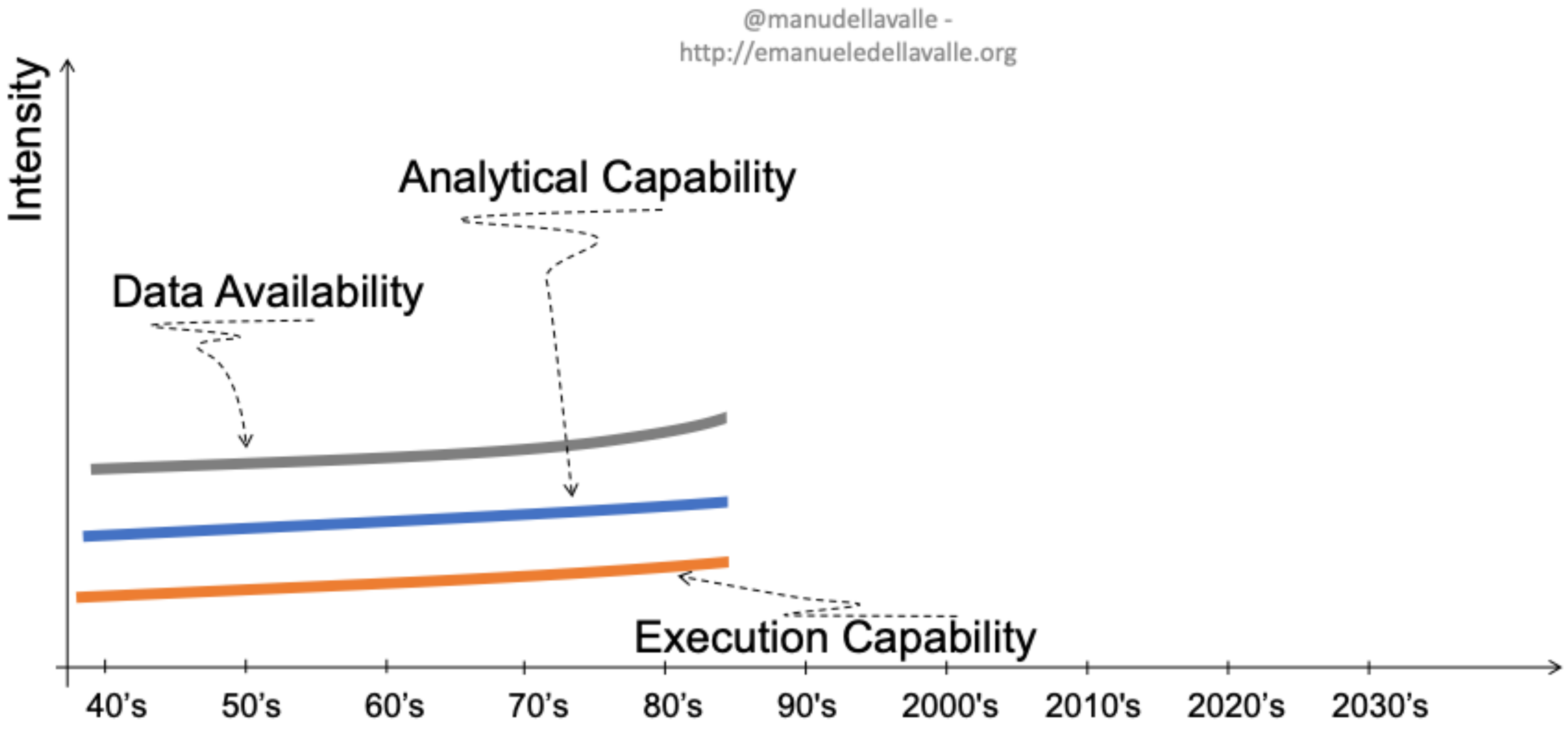
- Data warehouse typically interact with OLTP database to expose one or more OLAP system.
- Such OLAP system adopt storage optimized for analytics, i.e., Column Oriented
- The column-oriented storage layout relies on each column file containing the rows in the same order.
- Not just relational data, e.g., Apache Parquet



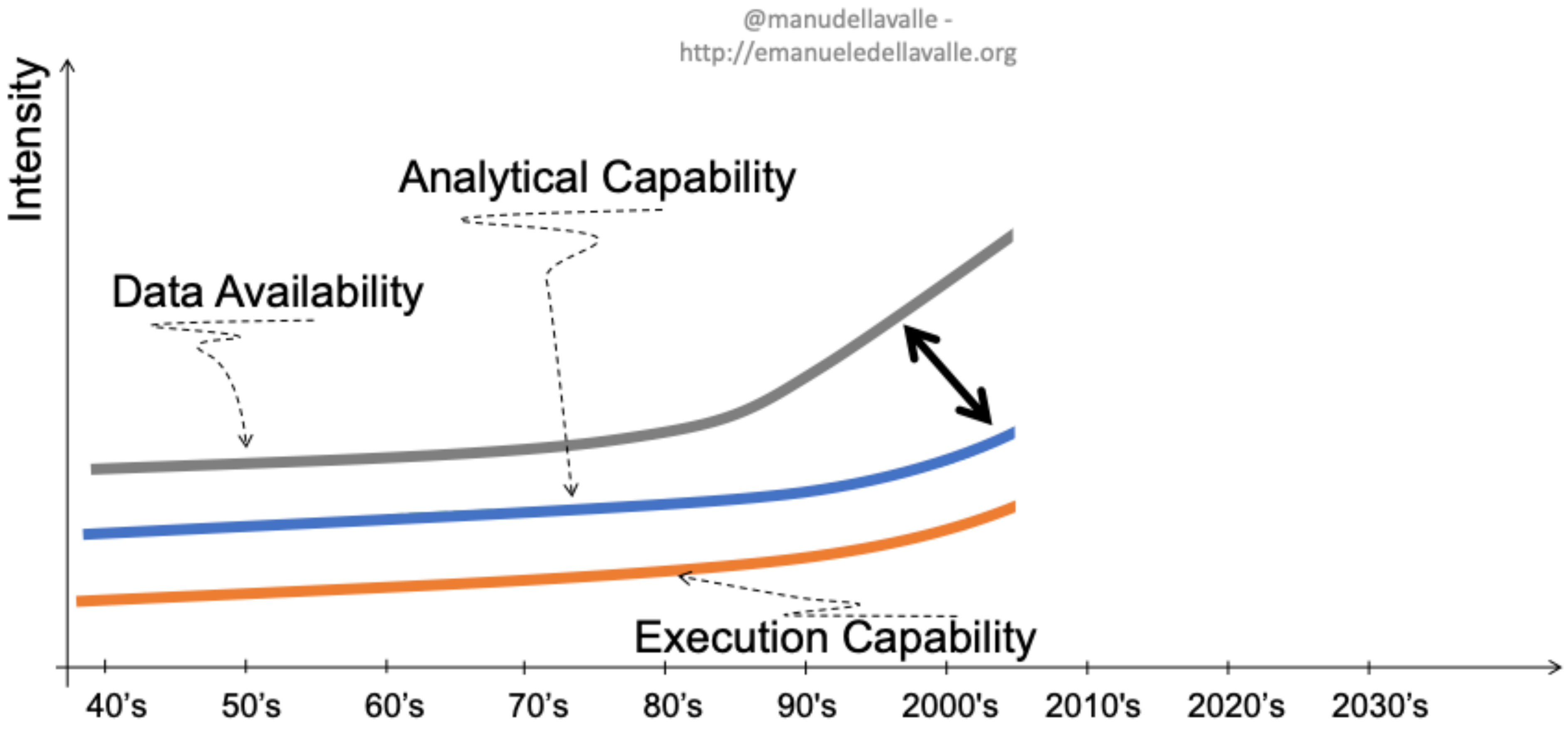
Data Modelling for Big Data

The Data Landscape: Variety is the Driver

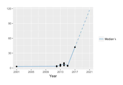
From data to analysis and execution



The appearance of the “Big Data”

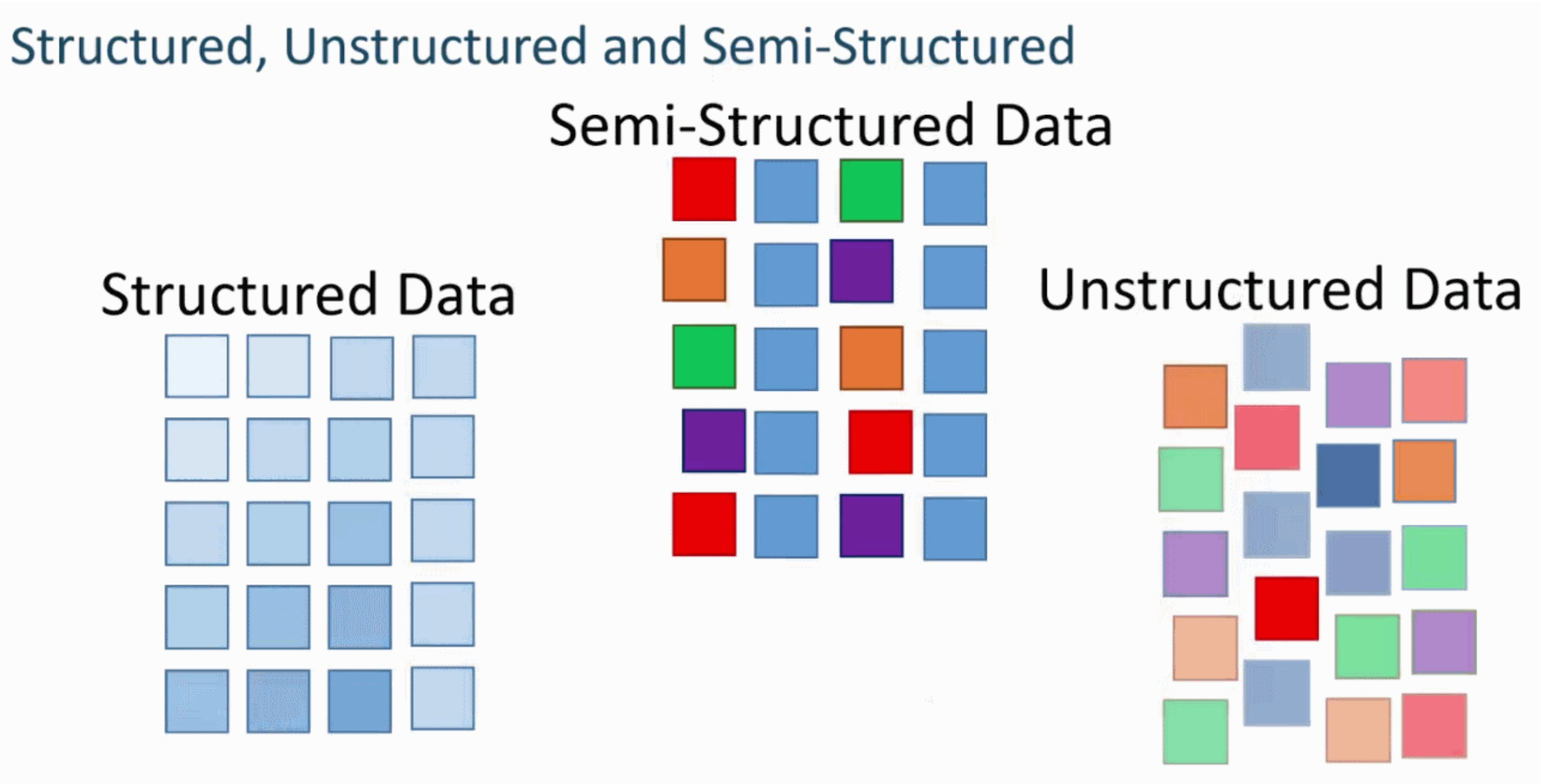


A Growing Trend



source

The Data Landscape



Structured data are organized and labeled according to a precise model (e.g., relational data)
^ Unstructured data, on the other hand, are not constrained (e.g., text, video, audio)
^ In between, there are many form of semi-structured data, e.g., JSON and XML, whose models do not impose a strict structure but provide means for validation.

Horizontal vs Vertical Scalability

Introduction

- "Traditional" SQL system scale **vertically** (scale up) - Adding data to a "traditional" SQL system may degrade its performances
  - When the machine, where the SQL system runs, no longer performs as required, the solution is to buy a better machine (with more RAM, more cores and more disk)
- Big Data solutions scale **horizontally** (scale out)
  - Adding data to a Big Data solution may degrade its performances
  - When the machines, where the big data solution runs, no longer performs as required, the solution is to add another machine

Curtesy of Emanuele Della Valle and Marco Brambilla

hardware

Commodity

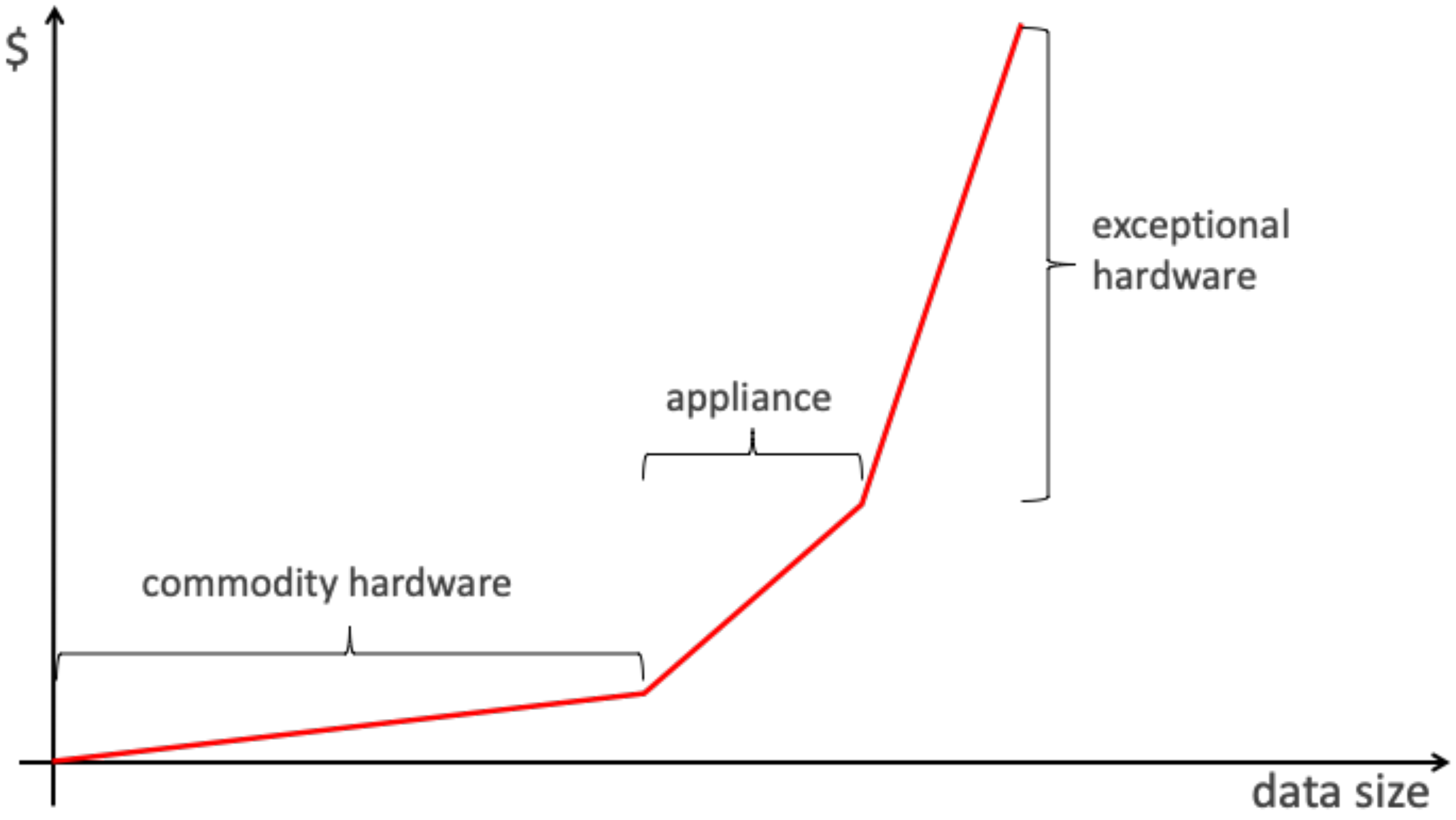
- CPU: 8-32 cores
- RAM: 16-64 GB
- Disk: 1-3 TB
- Network: 10 GE

Appliance

- CPU: 576 cores
- RAM: 24TB
- Disk: 360TB of SSD/rack
- Network: 40 Gb/second InfiniBand

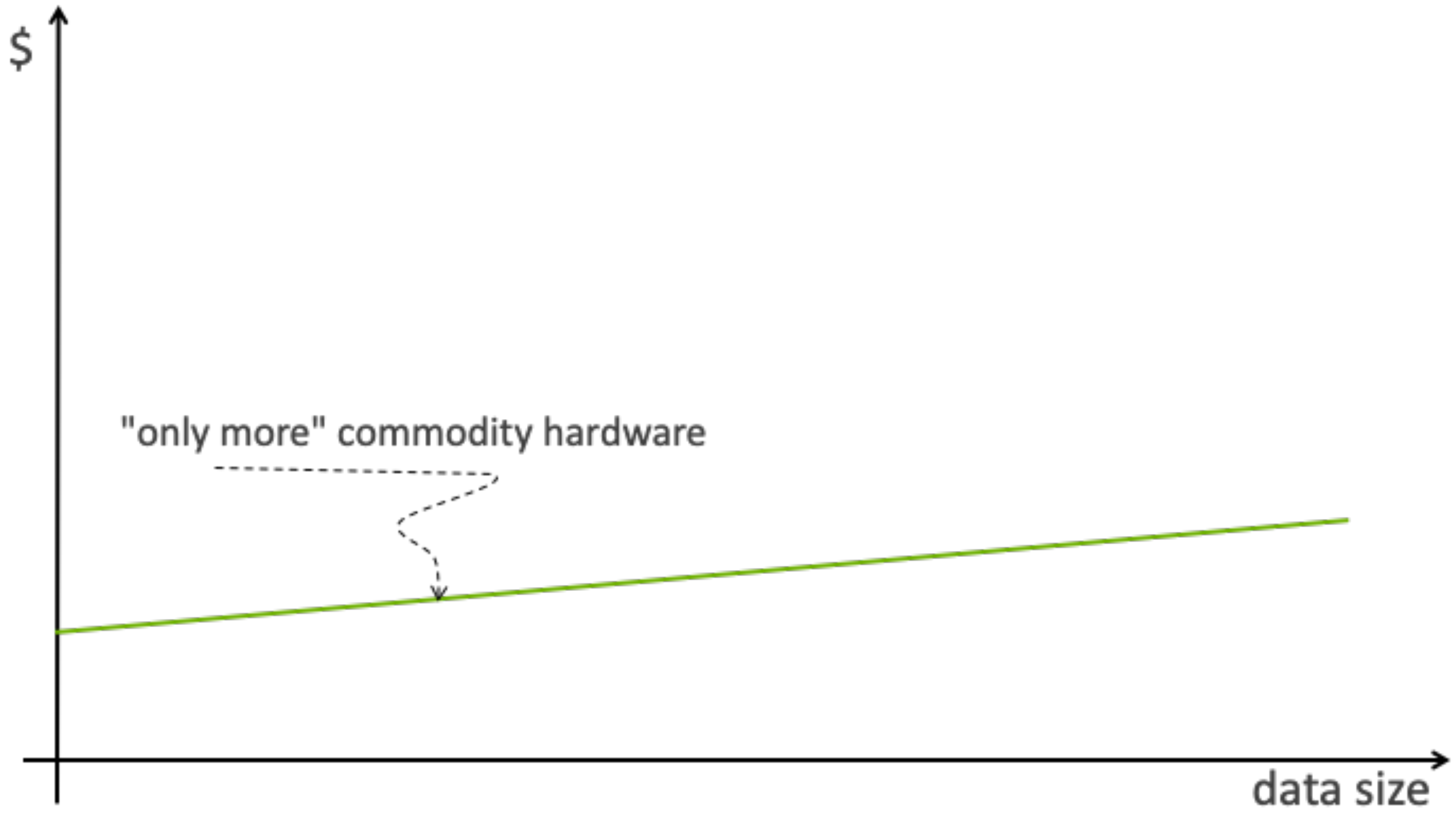
ORACLE EXADATA DATABASE MACHINE X6-8

Vertical Scalability



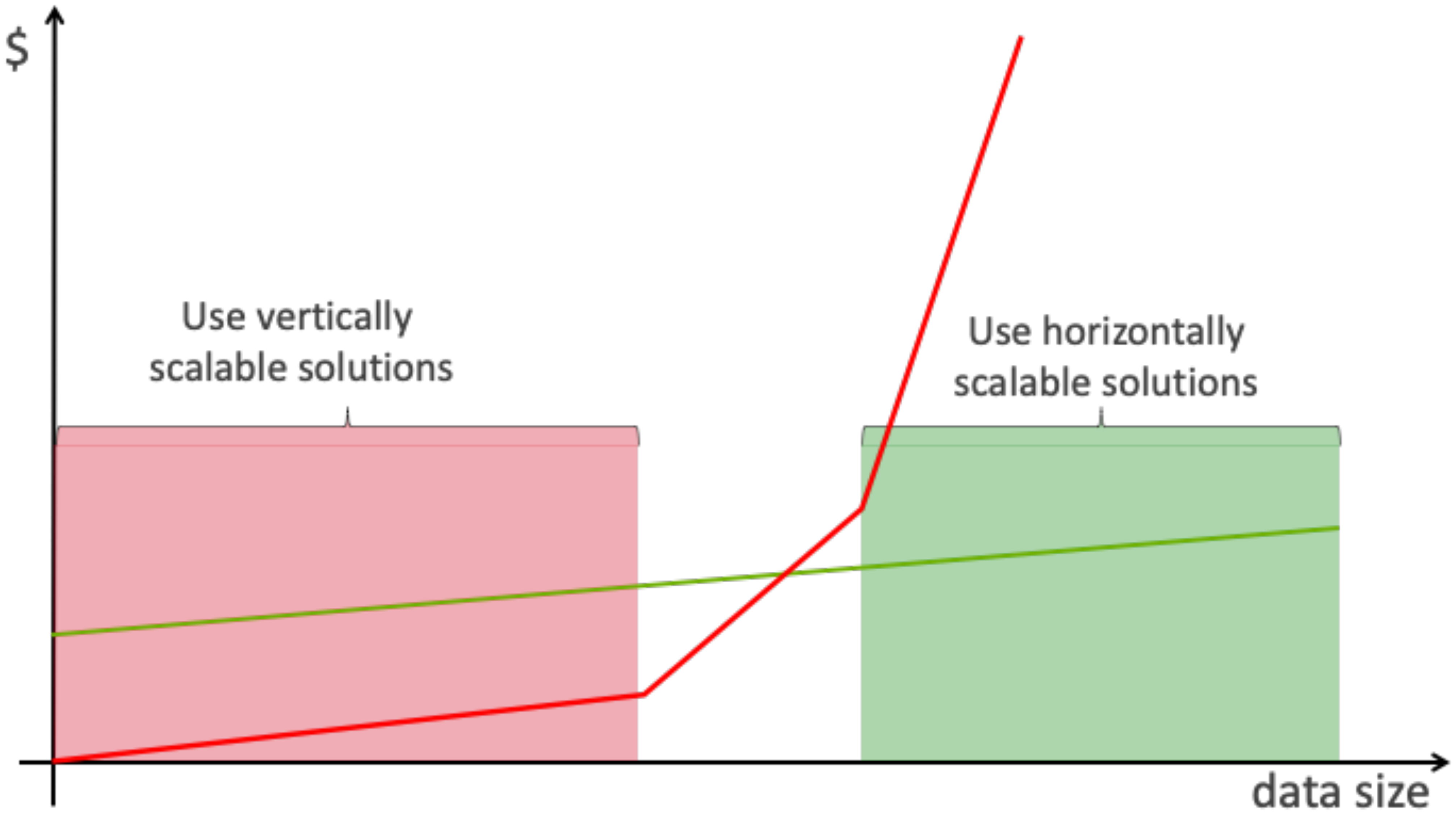
Curtesy of Emanuele Della Valle and Marco Brambilla

Horizontal Scalability



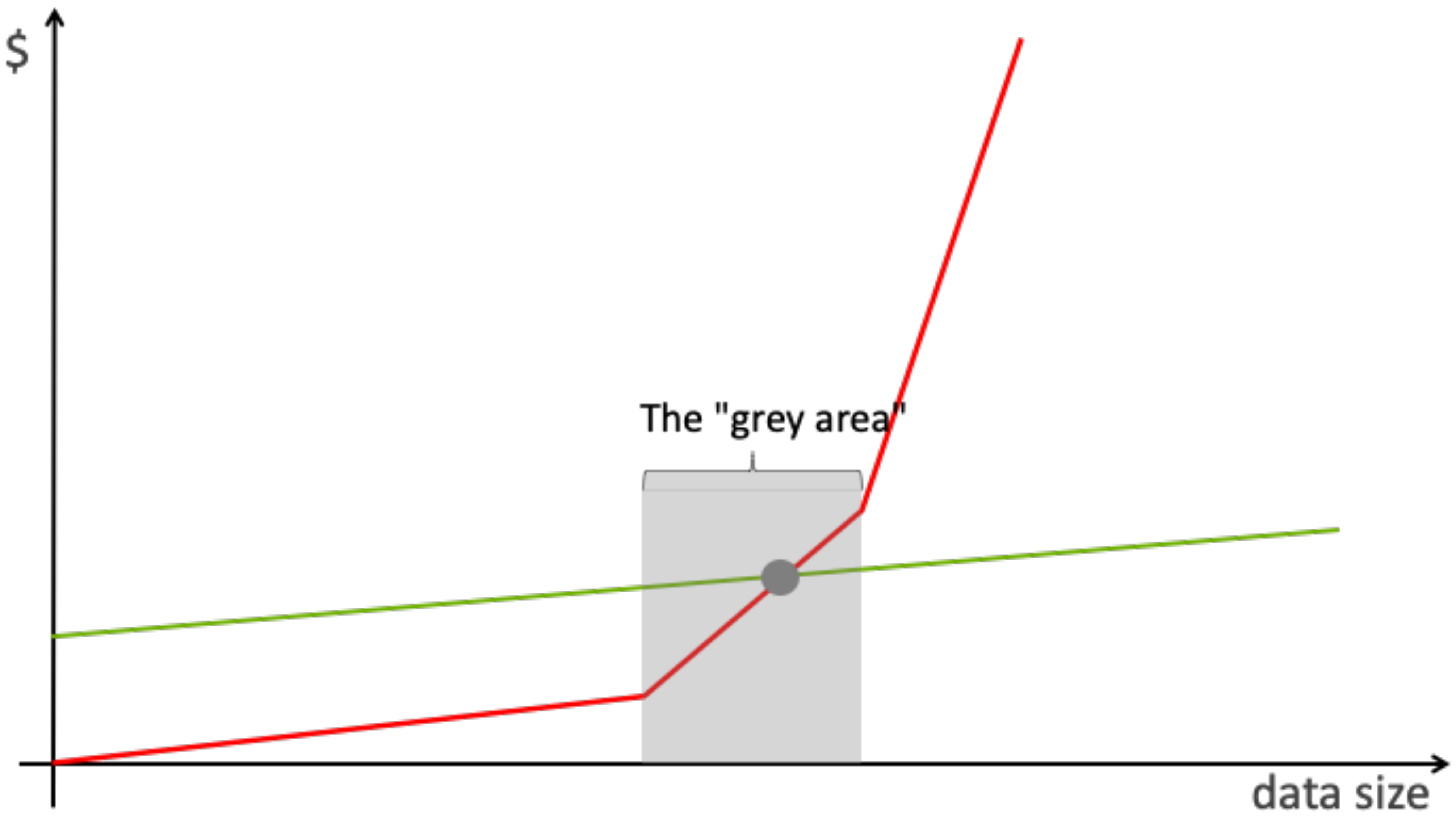
Curtesy of Emanuele Della Valle and Marco Brambilla

Vertical vs Horizontal Scalability



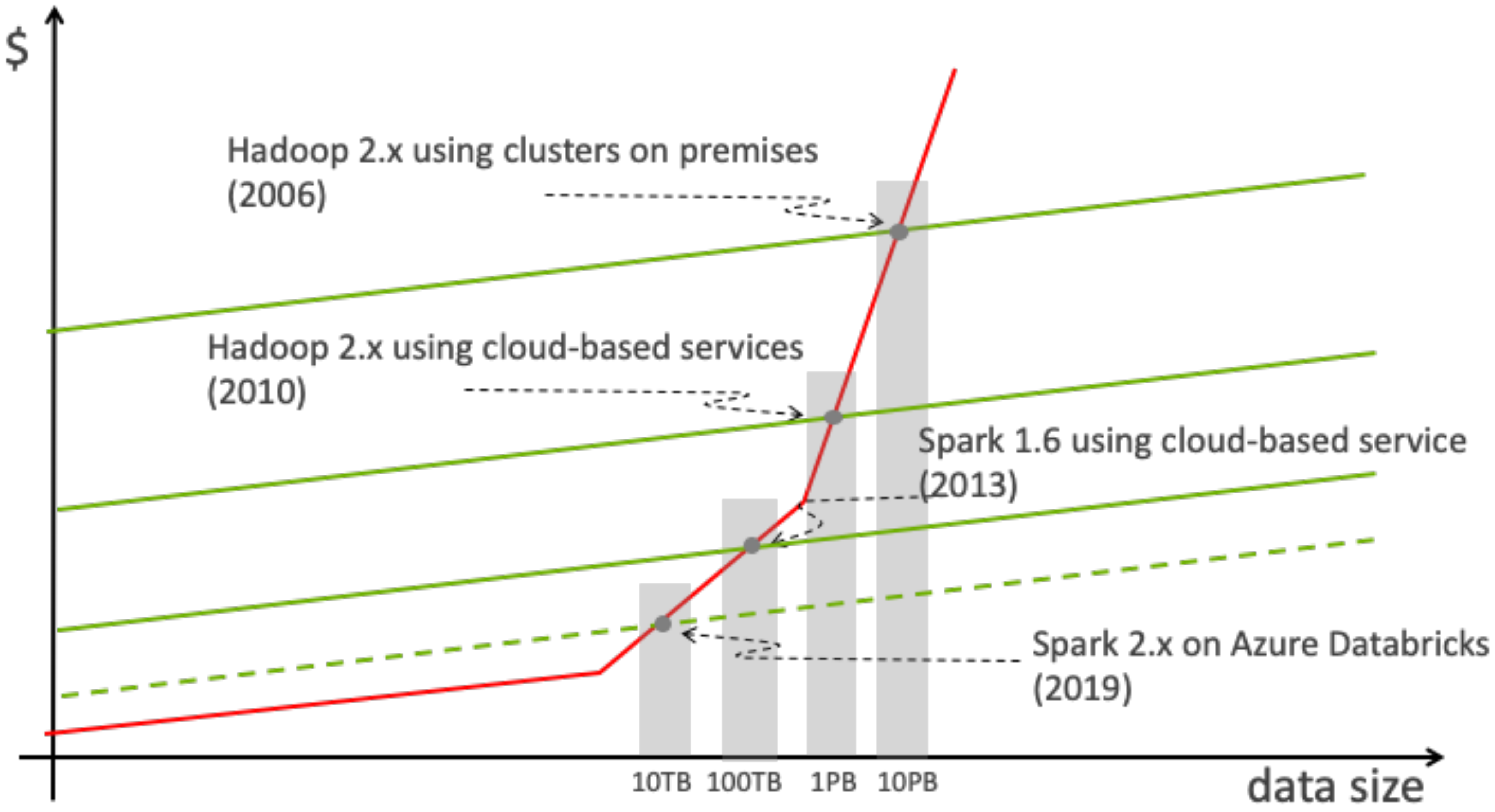
Curtesy of Emanuele Della Valle and Marco Brambilla

Vertical vs Horizontal Scalability



Curtesy of Emanuele Della Valle and Marco Brambilla

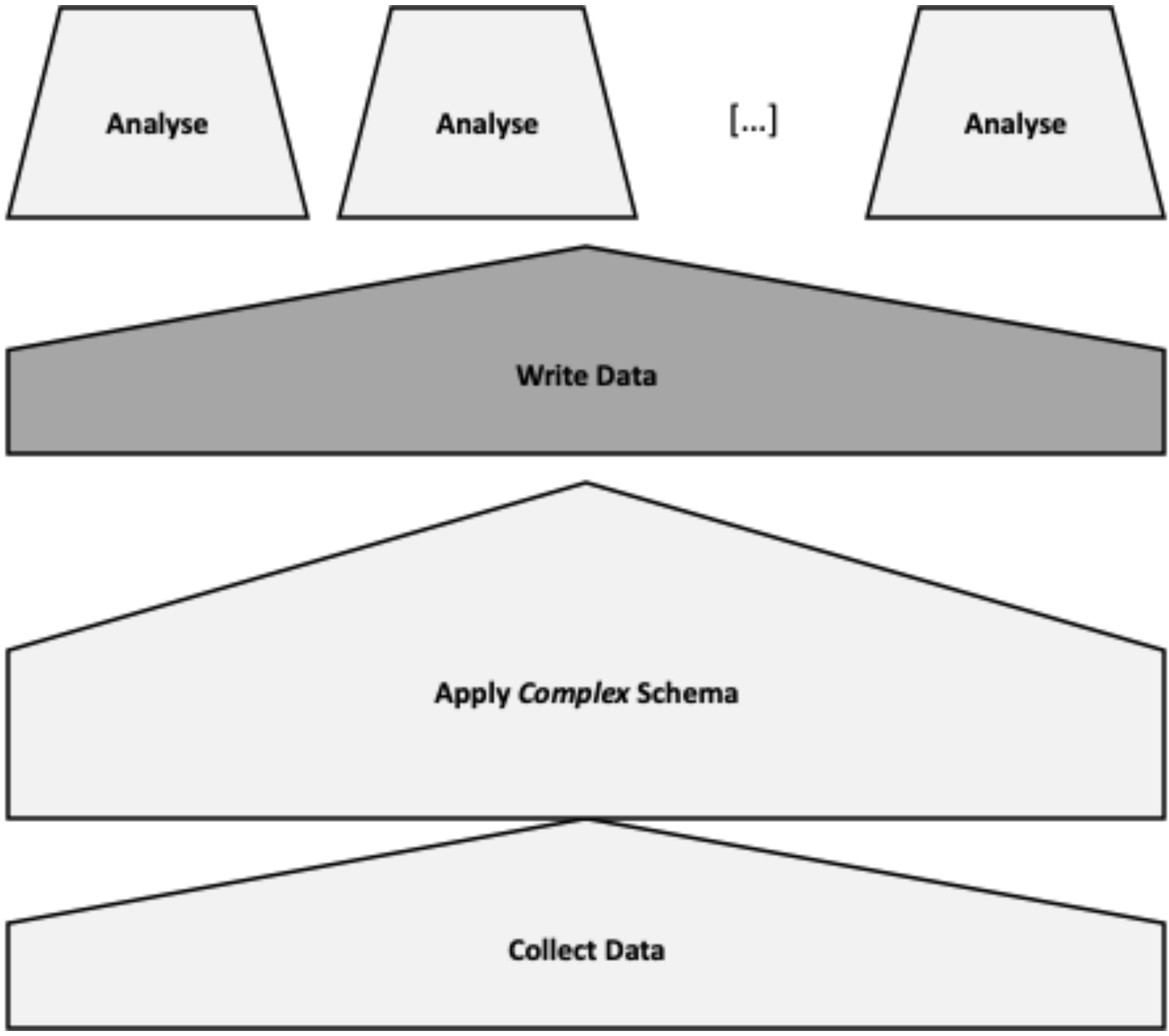
Grey Area is Time-Dependent



Curtesy of Emanuele Della Valle and Marco Brambilla

Traditional Data Modelling Workflow

- Known as Schema on Write
- Focus on the modelling a schema that can accommodate all needs
- Bad impact on those analysis that were not envisioned

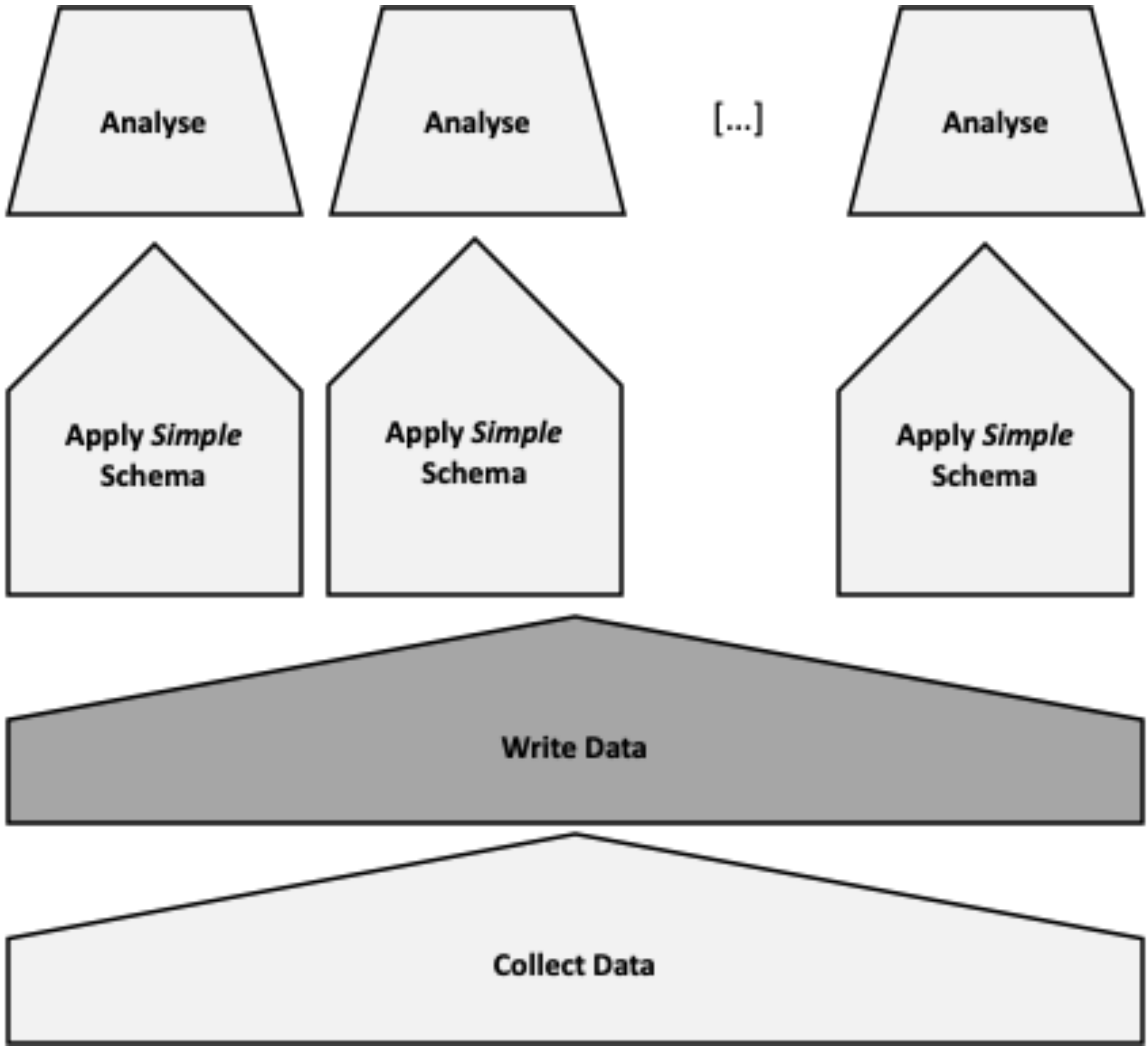


Extract Transform Load

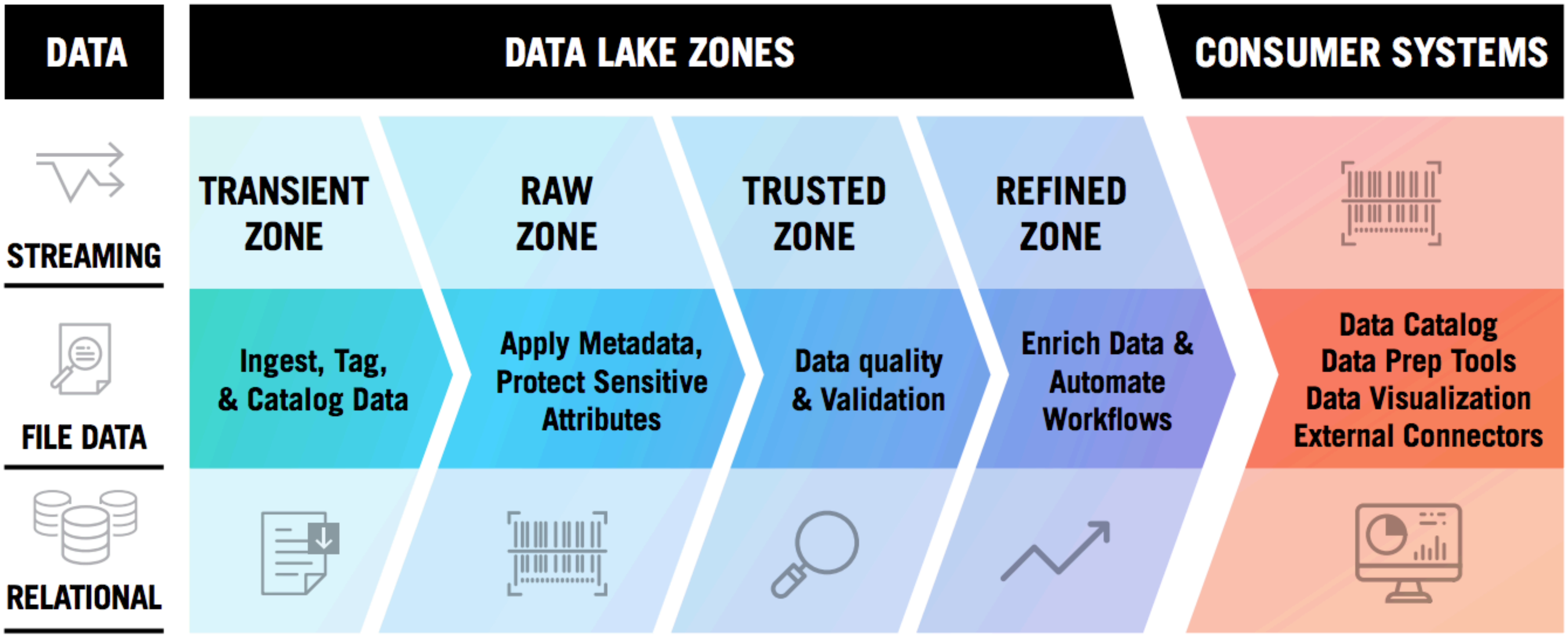
Some analyses may no longer be performed because the data were lost at writing time,

Schema on Read

- Load data first, ask question later
- All data are kept, the minimal schema need for an analysis is applied when needed
- New analyses can be introduced in any point in time



Data Lakes



DATA

STREAMING

FILE DATA

RELATIONAL

DATA LAKE ZONES

TRANSIENT ZONE

Ingest, Tag, & Catalog Data

RAW ZONE

Apply Metadata, Protect Sensitive Attributes

TRUSTED ZONE

Data quality & Validation

REFINED ZONE

Enrich Data & Automate Workflows

CONSUMER SYSTEMS

Data Catalog
Data Prep Tools
Data Visualization
External Connectors