ENCODE — Bridging the gap in Ancient Writing Cultures
Oslo 2022-10-10

Introduction to data modeling for the Humanities

Gioele Barabucci

NTNU | Norwegian University of Science and Technology

Data modelling

1. Theory: What is (not) data modelling
2. Turning research objects into tables
3. Establishig relations between entities
4. Extending and refining models
5. (?) Recording medatadata (e.g., provenance, time, context)

1. Theory: What is (not) data modelling

2. Turning research objects into tables
3. Establishing relations between entities
4. Extending and refining models
5. (?) Recording metadata (e.g., provenance, time, context)

Why do we create databases?

Why do we create databases?

So that we can

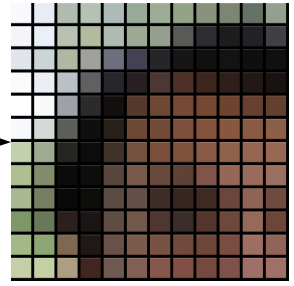
- store “things”
- in databases
- in a persistent way
- and later query them.

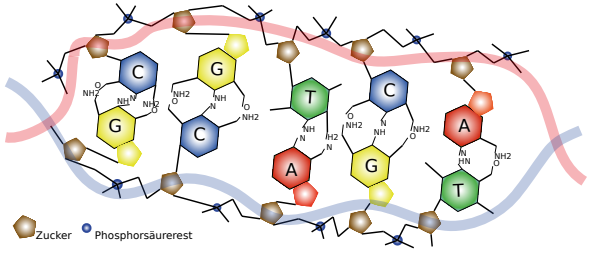
How can we put things <u>into</u> a computer?

?

Gioele's surname is Barabucci.
He lives in Europe and is 178 cm tall

```
name: Gioele Barabucci
height: 178
```





Non-, semi-, fully-structured information

Machine-readable != Usable by a machine.

Gioele's surname is Barabucci.
He lived in Cologne for 6 years
and now lives in Norway.

```
person {
    name: "Gioele"
    surname: "Barabucci"
    residence {
        country: &DE
        from: 2014, to: 2020
    }
    residence {
        country: &NO
        from: 2020
    }
}
```

<name>Gioele</name>'s surname
is <surname>Barabucci</surname>. He has been
living in <place country="DE">Cologne</place> for <timespan since="2014"
to="2020">6 years</timespan> and <timespan since="2020">now</timespan>
lives in <place country="NO">Norway</place>.

Is this a database?

Every

collection of data

is a database

Different trade-offs

PDF with scans

+ looks like source material
+ compatible with all computers
- no search function
- not editable

OCR'd txt files

+ content searchable
- no difference between data and metadata

RDF dataset

+ extreme flexibility
+ allows interlinking multiple datasets
- hard to maintain consistency
- slow queries

TIFF scans in folders

+ easy to add/remove entries
~ searchable, if names are smart
- content not searchable

BaseX XML-DB

+ rich vocabulary to model concepts
+ accepts XML files used by researchers
~ separation data/metadata possible
- requires creation of interface to see data

SQLite DB

+ guarantees consistency of data (ACID)
- all concepts must be modelled as tables and relations between rows

A database provides (at least)...

- A way to declare how the data should look like (headers, schema, TBox, ...)
- A way to add data
- A way to store data
- A way to query existing data

CSV + Excel/Librecalc

- **Header names**: A way to declare how the data should look like
- **Add row**: A way to add data
- **Excel/Librecalc**: A way to store data
- **CTRL-F**: A way to query existing data

XML + BaseX

- **DTD/XML Schema**: A way to declare how the data should look like
- **Add XML file**: A way to add data
- **BaseX**: A way to store data
- **XQuery**: A way to query existing data

Today SQL

Today ~~SQL~~ ER

Today ~~SQL~~ ~~ER~~

RDBMS

Today ~~SQL~~ ~~ER~~ ~~RDBMS~~

Today
relational
databases

Relational database

- **Relational model (~ER, Entity-Relationship)**: A way to describe how the pieces of data relate to each other
- **SQL (DDL)**: A way to declare how the data should look like
- **SQL INSERT**: A way to add data
- **RDBMS**: A way to store data
- **SQL (DQL)**: A way to query existing data

(lossy) Equivalence

```
<name>Gioele</name>'s surname
is <surname>Barabucci</surname>.
He has been living in <place
country="DE">Cologne</place> for
<timespan since="2014"
to="2020">6 years</timespan>
and <timespan since="2020">now
</timespan>  lives in <place
country="NO">Norway</place>.
```

```
person {
    name: “Gioele”
    surname: “Barabucci”
    residence {
        country: &DE
        from: 2014, to: 2020
    }
    residence {
        country: &NO
        from: 2020
    }
}
```

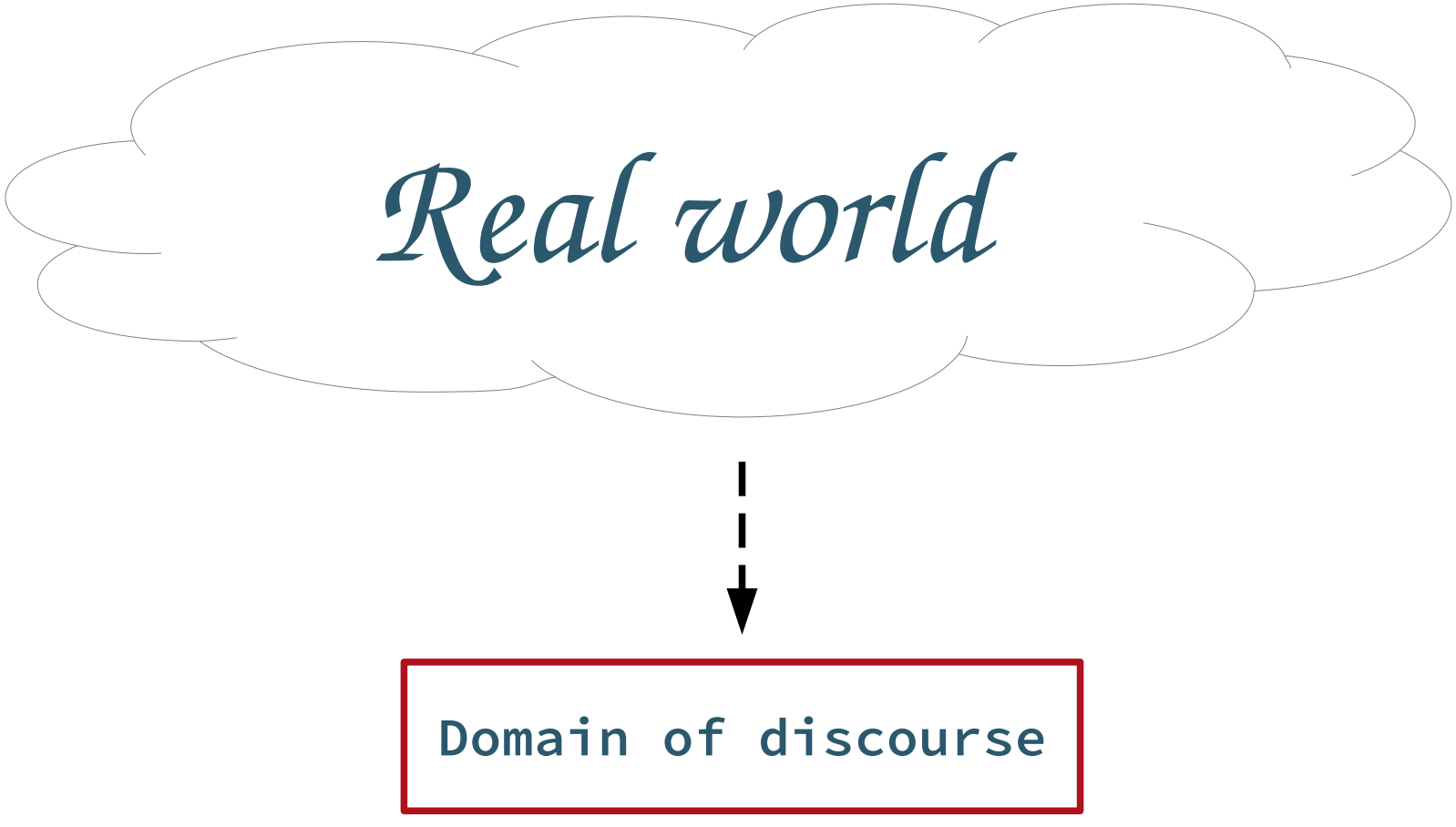
Residence				
Name	Surname	Country	from	to
Gioele	Barabucci	DE	2014	2022
Gioele	Barabucci	NO	2000	

Relational database

- **Relational model (~ER, Entity-Relationship)**: A way to describe how the pieces of data relate to each other
- **SQL (DDL)**: A way to declare how the data should look like
- **SQL (INSERT)**: A way to add data
- **RDBMS**: A way to store data
- **SQL (DQL)**: A way to query existing data

What does modelling mean?

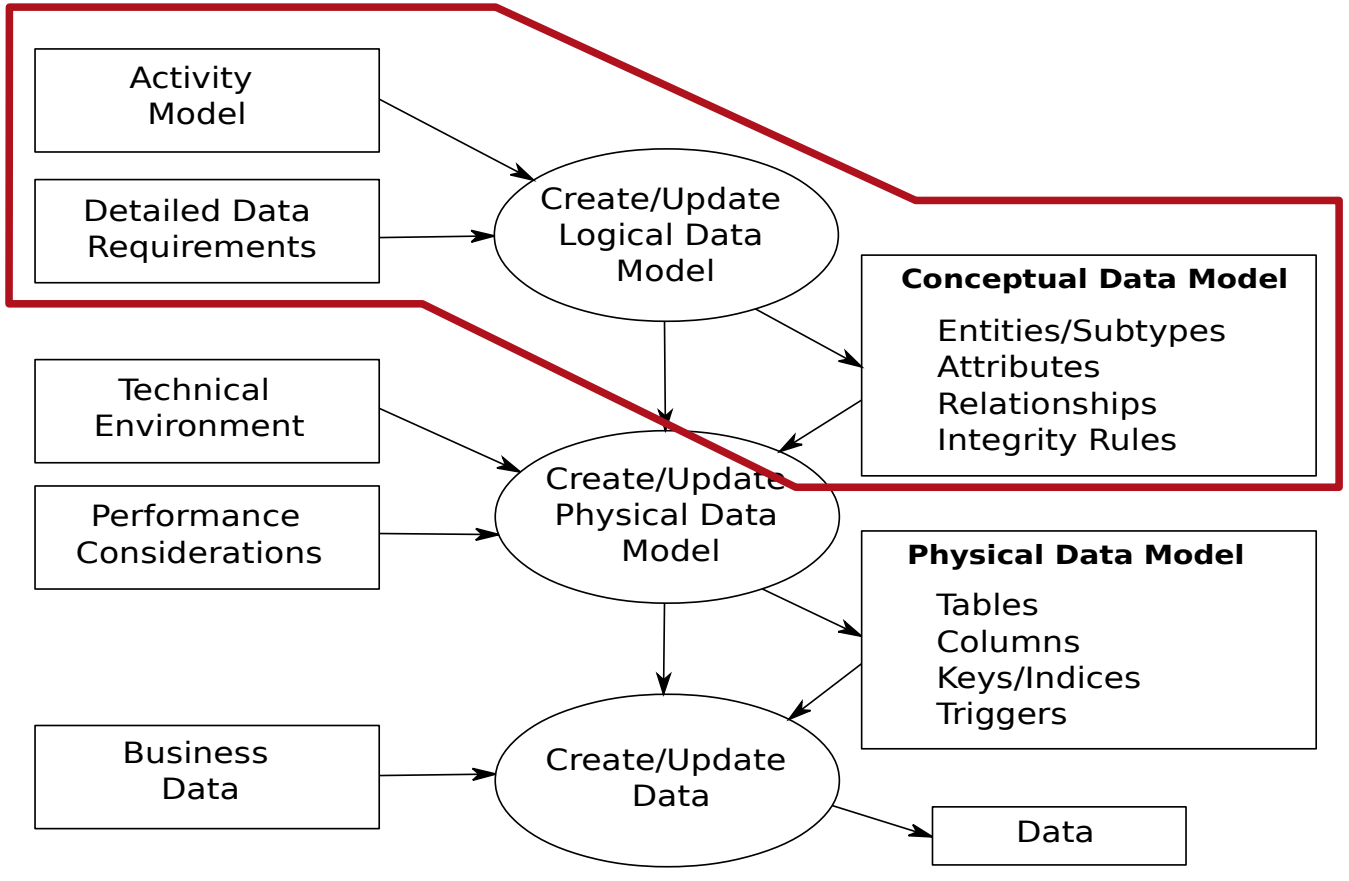
Modeling means choosing what's in and what's out



Modeling issues

- How to name things?
  - Person? Student? Students? A231?
- How to aggregate things?
  - Name? Name + Surname? Middle? Nickname?
- How to deal with changes? Repetitions? Alternative?
  - Maiden name? Remarriage? Name in other language?
- How to define identity? Equality? Equivalence?
  - Hello 2000 years of philosophy

Modeling means choosing what's in and what's out



ANSI/SPARK/EPISTLE model

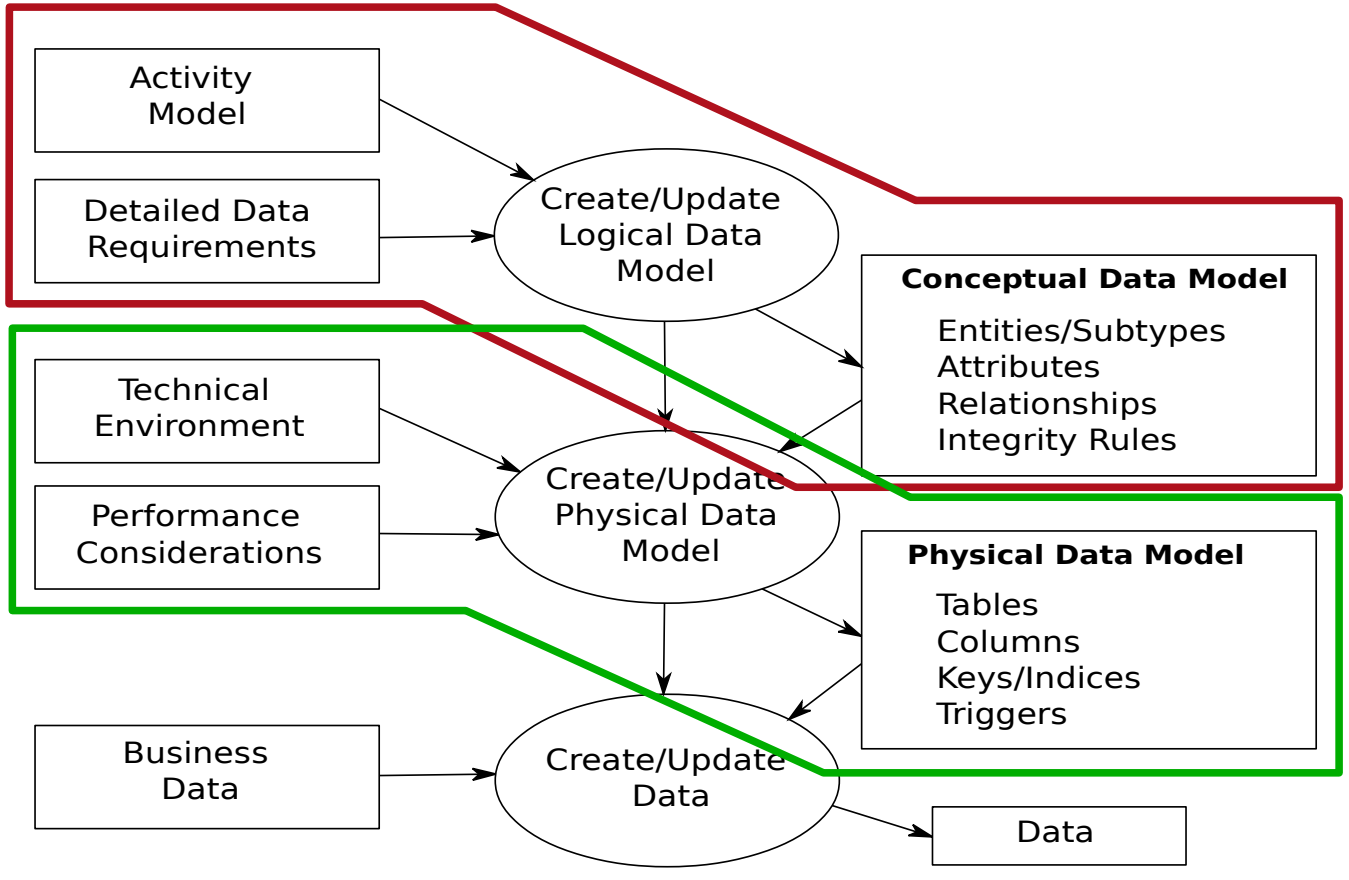
Semplification

+

Formalization

Developing High Quality Data Models, Matthew West and Julian Fowler. 1996

Modeling means choosing what's in and what's out



ANSI/SPARK/EPISTLE model

Semplification

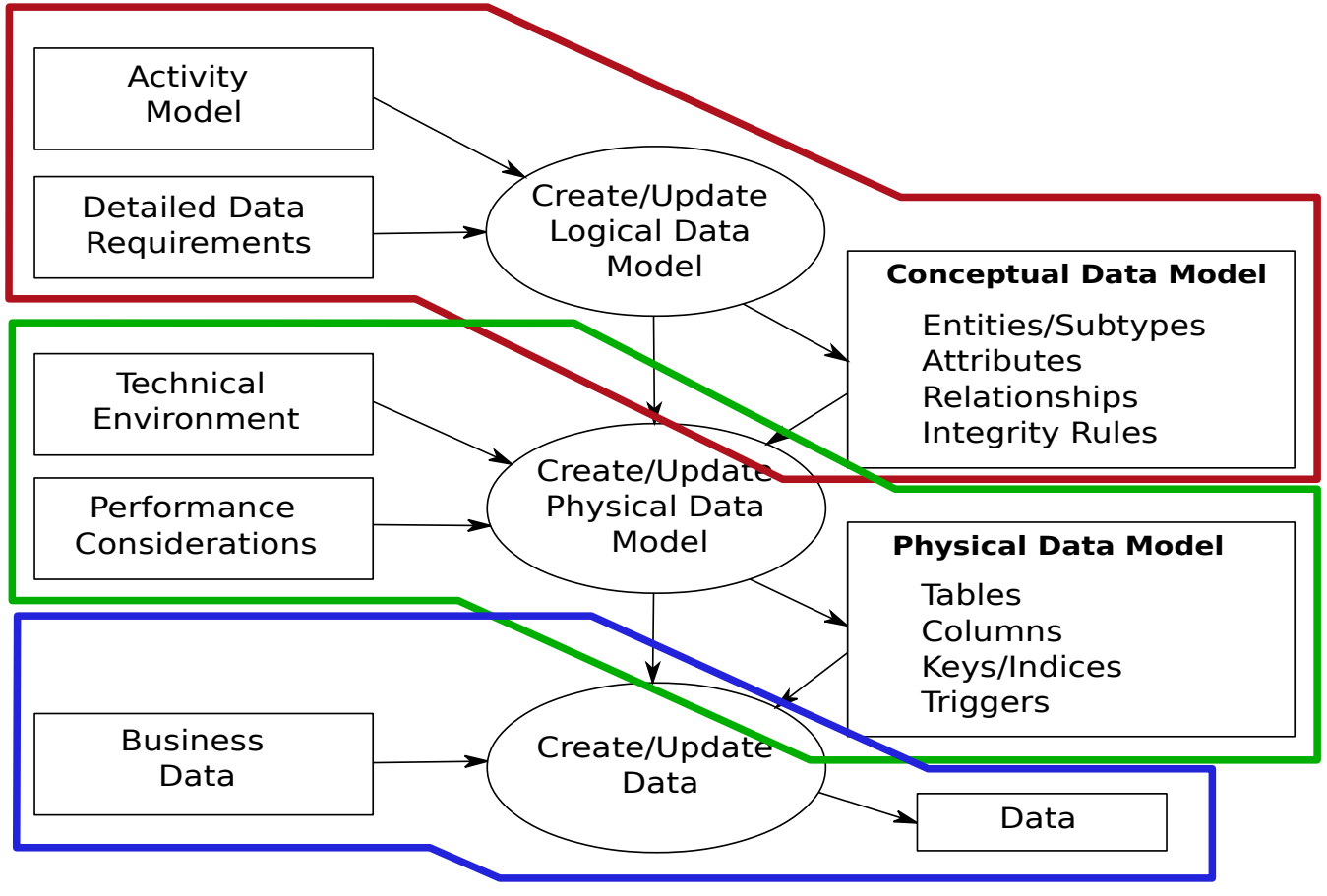
+

Formalization

Implementation

Developing High Quality Data Models, Matthew West and Julian Fowler. 1996

Modeling means choosing what's in and what's out



ANSI/SPARK/EPISTLE model

Semplification

+

Formalization

Implementation

Storage

Developing High Quality Data Models, Matthew West and Julian Fowler. 1996

No perfect models

On Exactitude in Science

[...] the Art of Cartography attained such Perfection that the map of a single Province occupied the entirety of a City, and the map of the Empire, the entirety of a Province. In time, those Unconscionable Maps no longer satisfied, and the Cartographers Guilds struck a Map of the Empire whose size was that of the Empire, and which coincided point for point with it. [...]

— *Jorge Luis Borges*

No perfect models

On Exactitude in Science

[...] the Art of Cartography attained such Perfection that the map of a single Province occupied the entirety of a City, and the map of the Empire, the entirety of a Province. In time, those Unconscionable Maps no longer satisfied, and the Cartographers Guilds struck a Map of the Empire whose size was that of the Empire, and which coincided point for point with it. [...]

— *Jorge Luis Borges*

No perfect models

A map is not the territory it represents, but, if correct, it has a similar structure to the territory, which accounts for its usefulness.

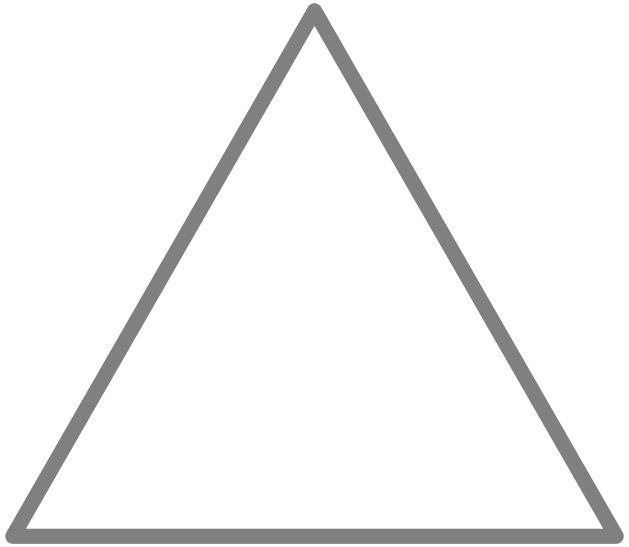
— *Alfred Korzybski*

All models are wrong, but some models are useful.

— *Georg Box*

CAP theorem for modeling technologies

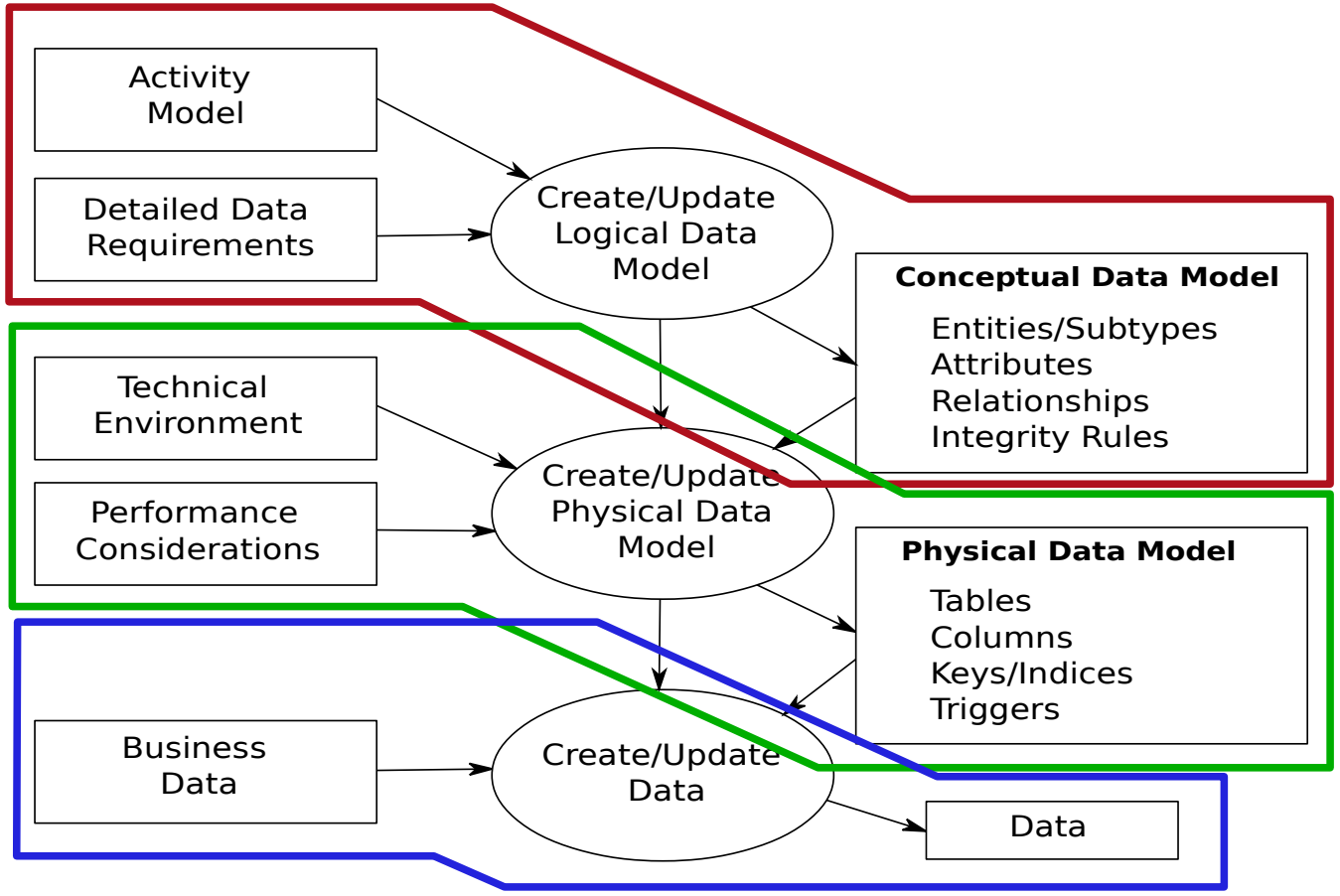
Expressivity
(= faithfulness)



Ease of
modeling

Speed of
inference/query

Who does the modeling?



ANSI/SPARK/EPISTLE model

Semplification

+

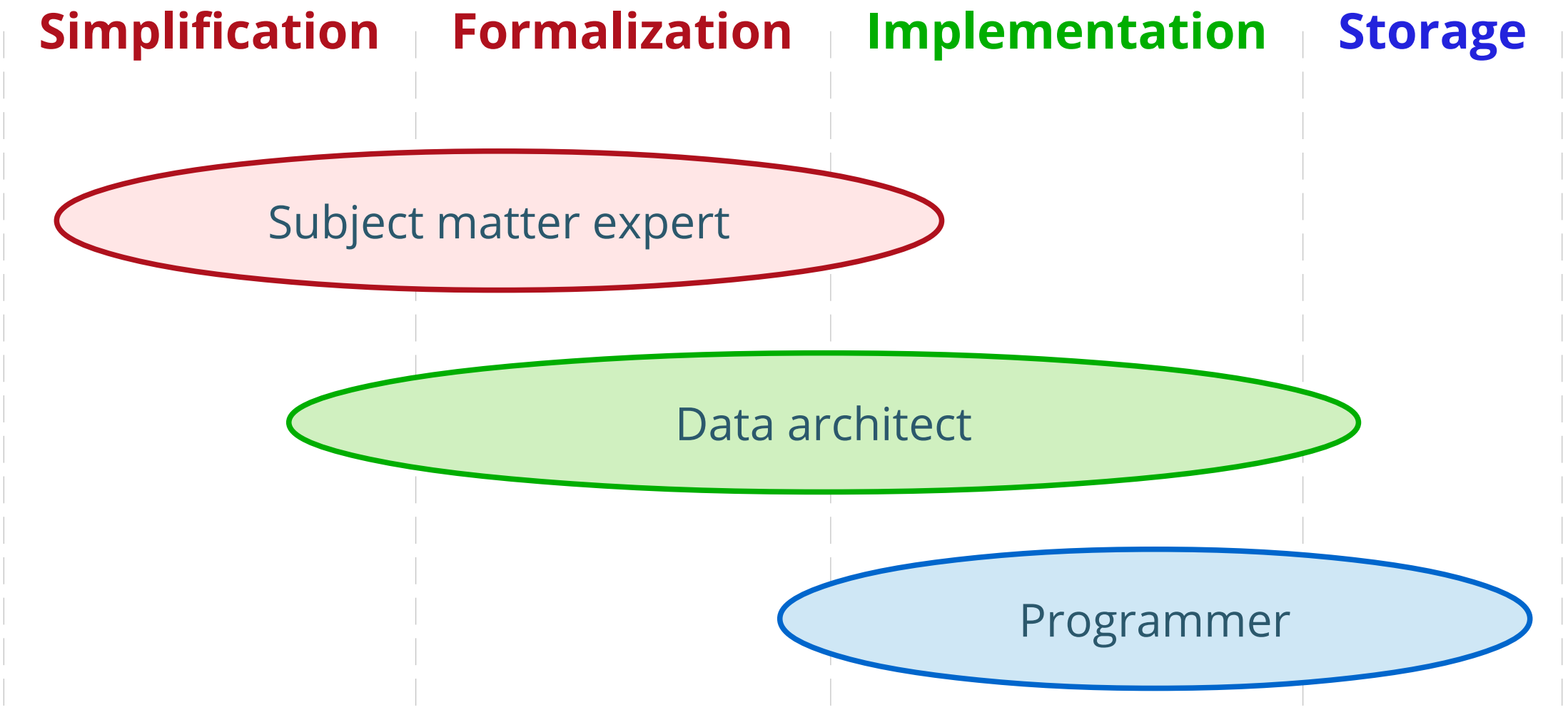
Formalization

Implementation

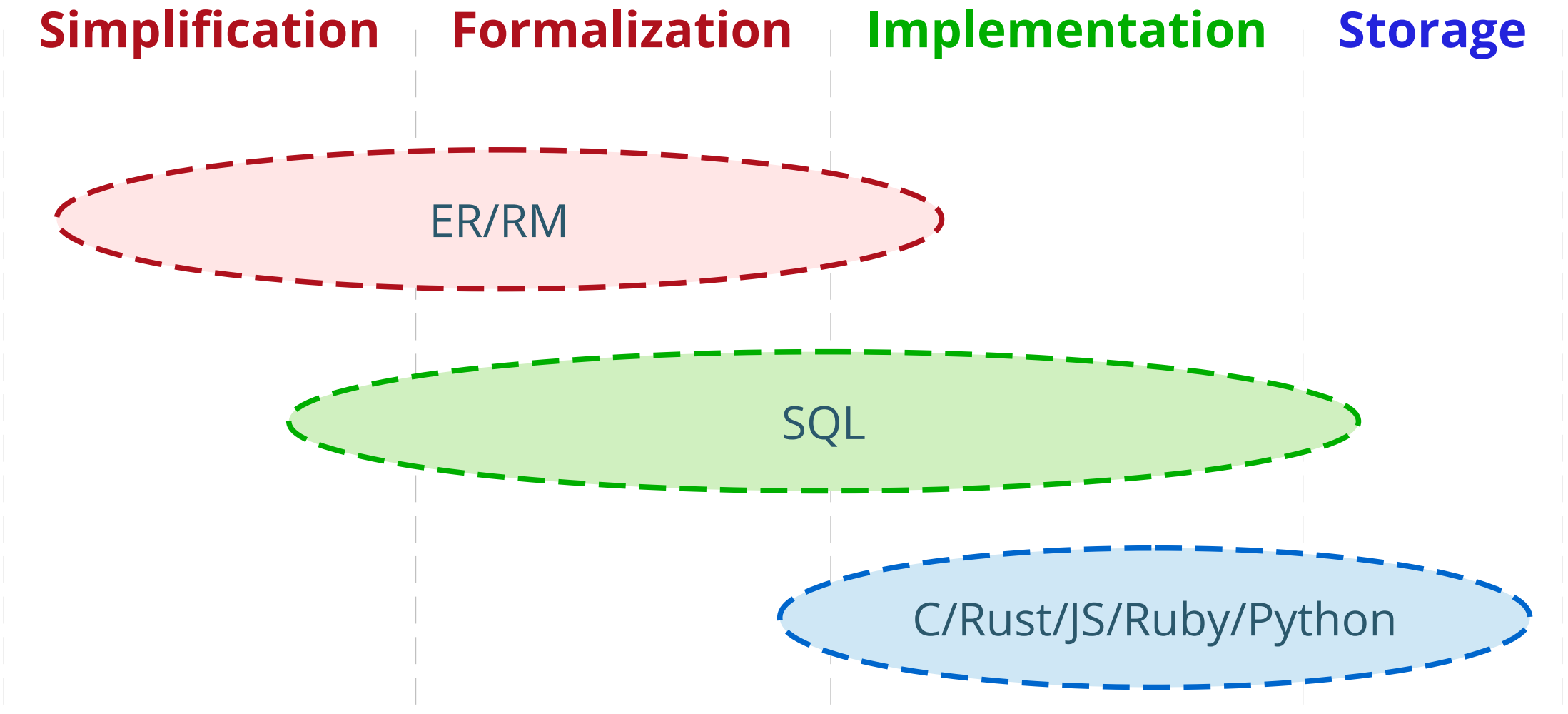
Storage

Developing High Quality Data Models, Matthew West and Julian Fowler. 1996

Who does the modeling?



How is the model described?



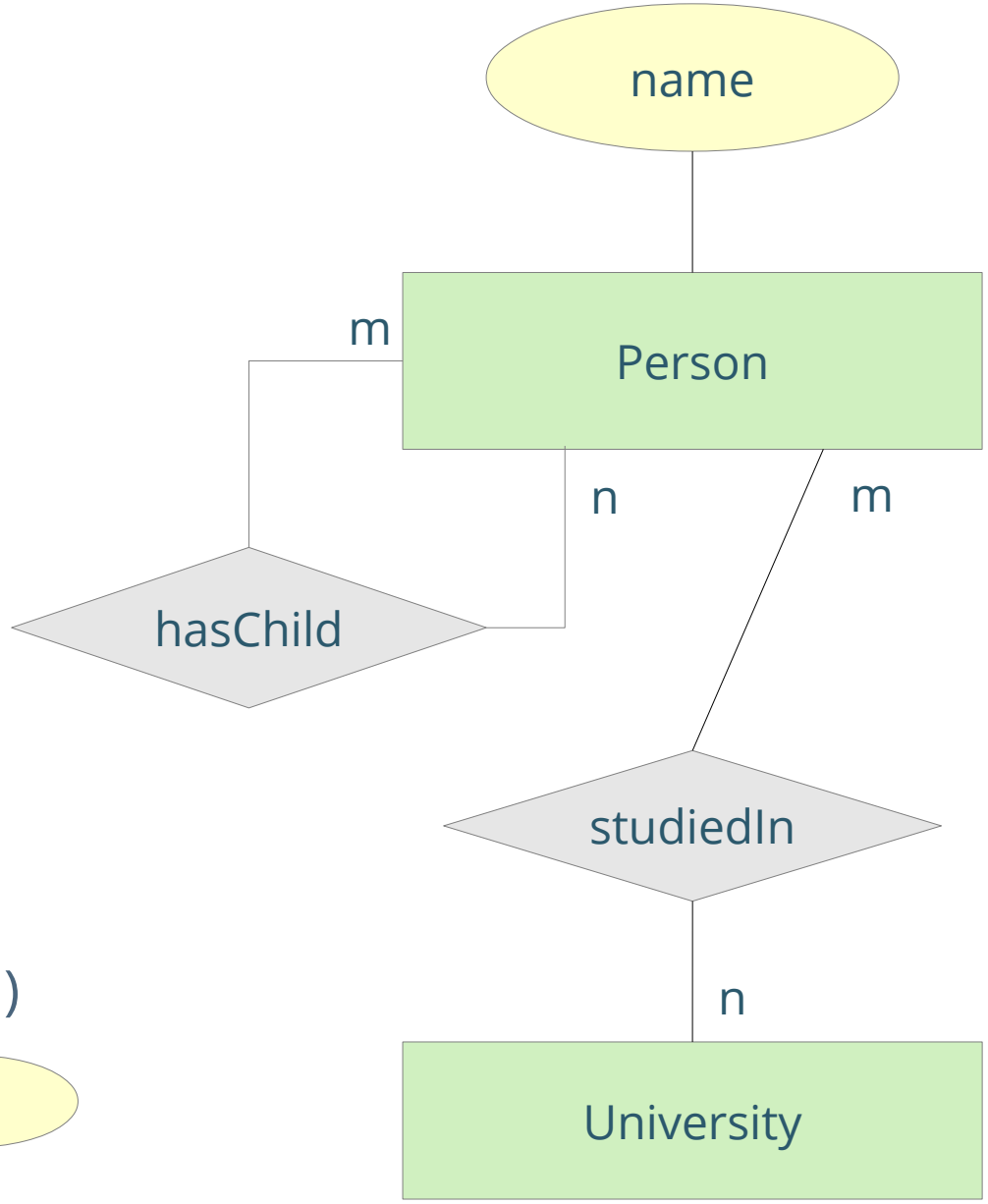
Modelling in ER / SQL

ER: The basics

Two types:

- Entities
  - have IDs (keys, PK)
  - e.g. **`Person`**
- Relationships
  - relate entities (via FK)
  - **`hasChild`**, **`studiedIn`**
  - various cardinalities (1:1, 1:n, n:m)

Both may have attributes



(rough) Translation into Relational Model

- Entities = Tables
- Relationships = Tables
- Fields = Columns in tables

Persons	
ID	name
1	Jack
2	Mel
3	Anne

Universities	
ID	name
1	Bologna
2	Sorbonne
3	Oxford

studiedIn	
personID	uniID
1	1
1	3
2	1
3	7

(rough) Translation into SQL (DDL)

```
CREATE TABLE “Persons” (
    “ID” INTEGER PRIMARY KEY,
    “Name” VARCHAR(128),
);
CREATE TABLE “studiedIn” (
   “PersonID” INTEGER REFERENCES Persons(ID),
   “UniID” INTEGER REFERENCES Universities(ID),
)
```

Concepts vs Entities vs Datatypes

- Concepts = Classifications of things that exist in the domain
- Entities/Relations/Fields/Attributes
  = Formalization of stand-alone concepts
  - Gioele is a Person
- Datatypes = Type of the data in the fields/attributes
  - (The value in the field) Name is a piece of text
  - (The value in the field) Height is a number
  - (The value in the field) Day of birth is a point in time

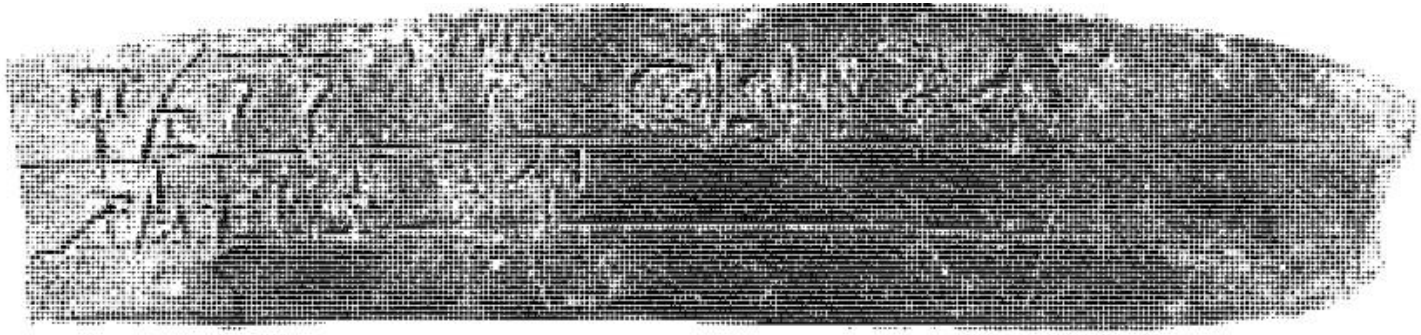
Life is hard, use

MySQL Workbench

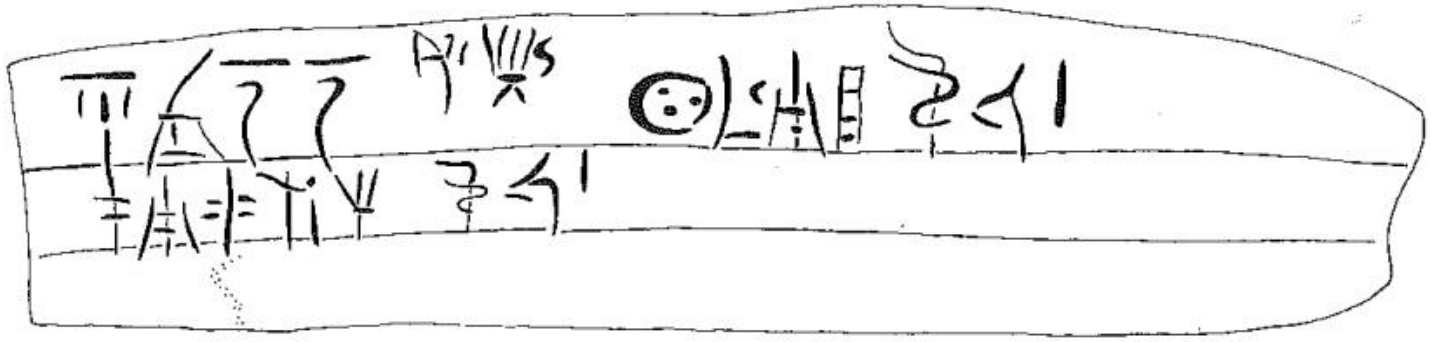
1. Theory: What is (not) data modelling

2. Turning research objects into tables

3. Establishing relations between entities
4. Extending and refining models
5. (?) Recording metadata (e.g., provenance, time, context)



Our source (COMIK)



Fp(1) 5 A 138

.1 di-wi-jo-jo 'me-no' qe-ra-si-ja OLE S 1

.2 pa-si-te-o-i OLE S 1

.3 *vacat*

Cut at left.

Top-down modelling

Study the domain

"Smith (1978) classifies the artifacts in three kinds"

The model

The model

"the city is almost always recorded", "oh, many entries have a date!"

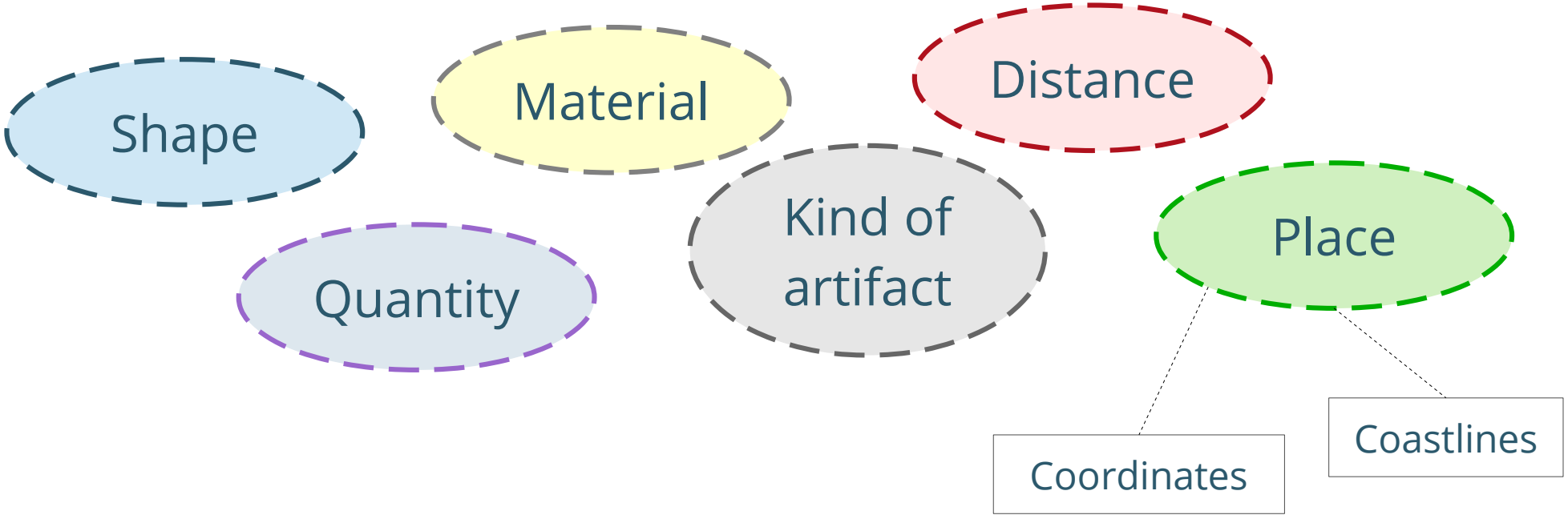
Study the data

Bottom-up modelling

Query-oriented modelling

Are round gold earrings found more frequently near lakes?

Research question



Concepts required in model

Extra data

Research Q for COMIK data?

Our first query

How many tablets contain the syllable "ja"?

Research question

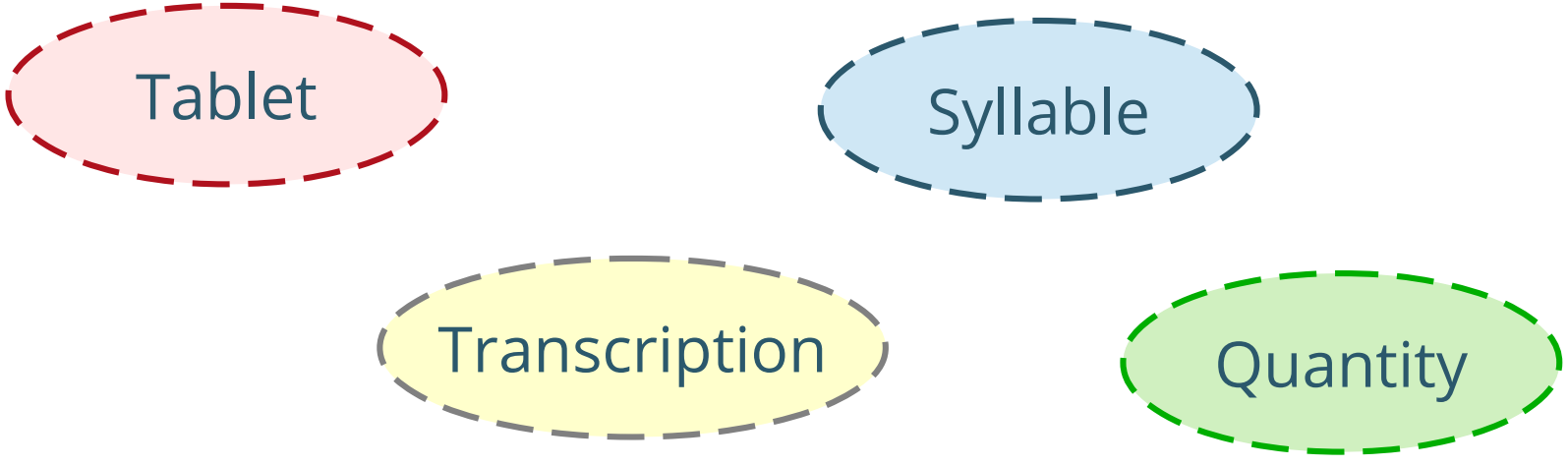
Which concepts do we need?

How many tablets contain the syllable "ja"?

Our first query

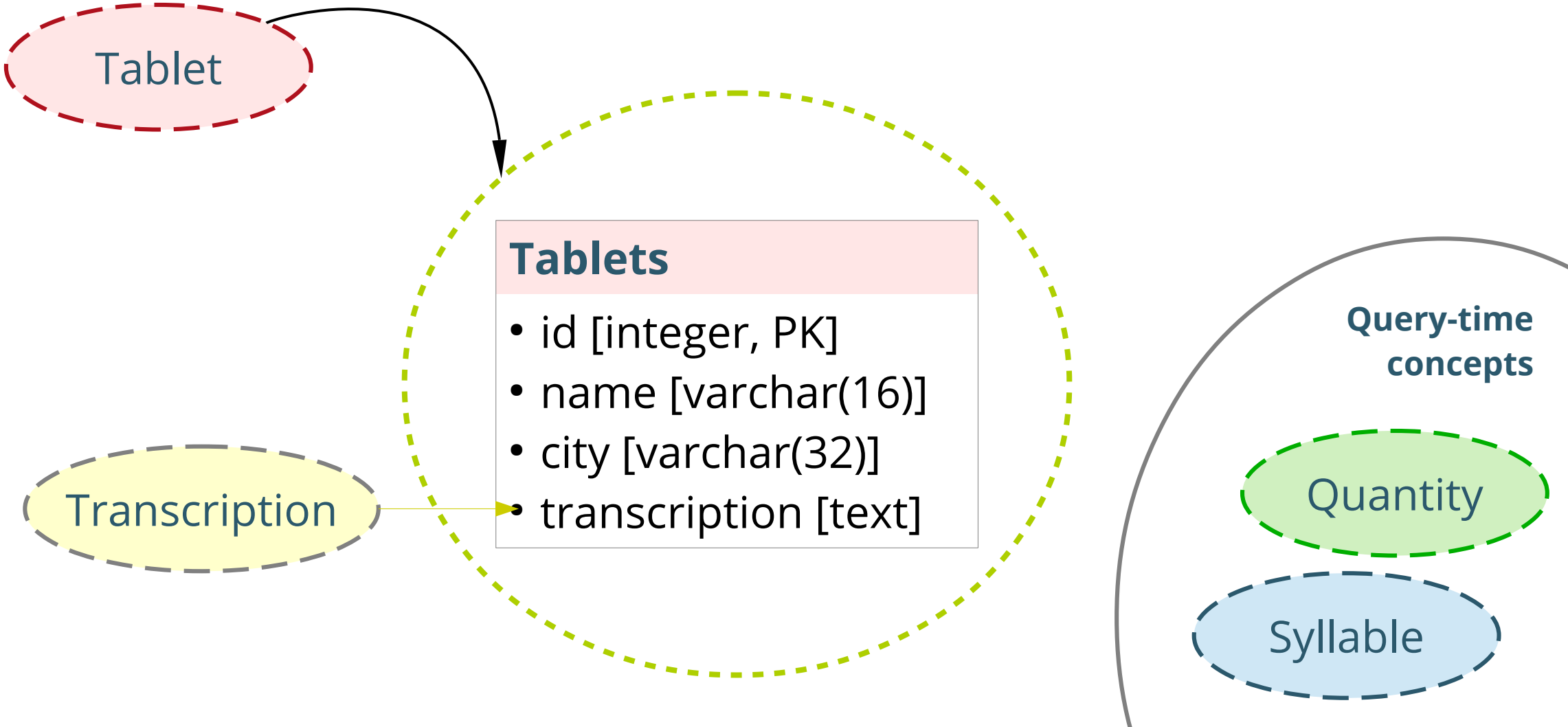
How many tablets contain the syllable "ja"?

Research question



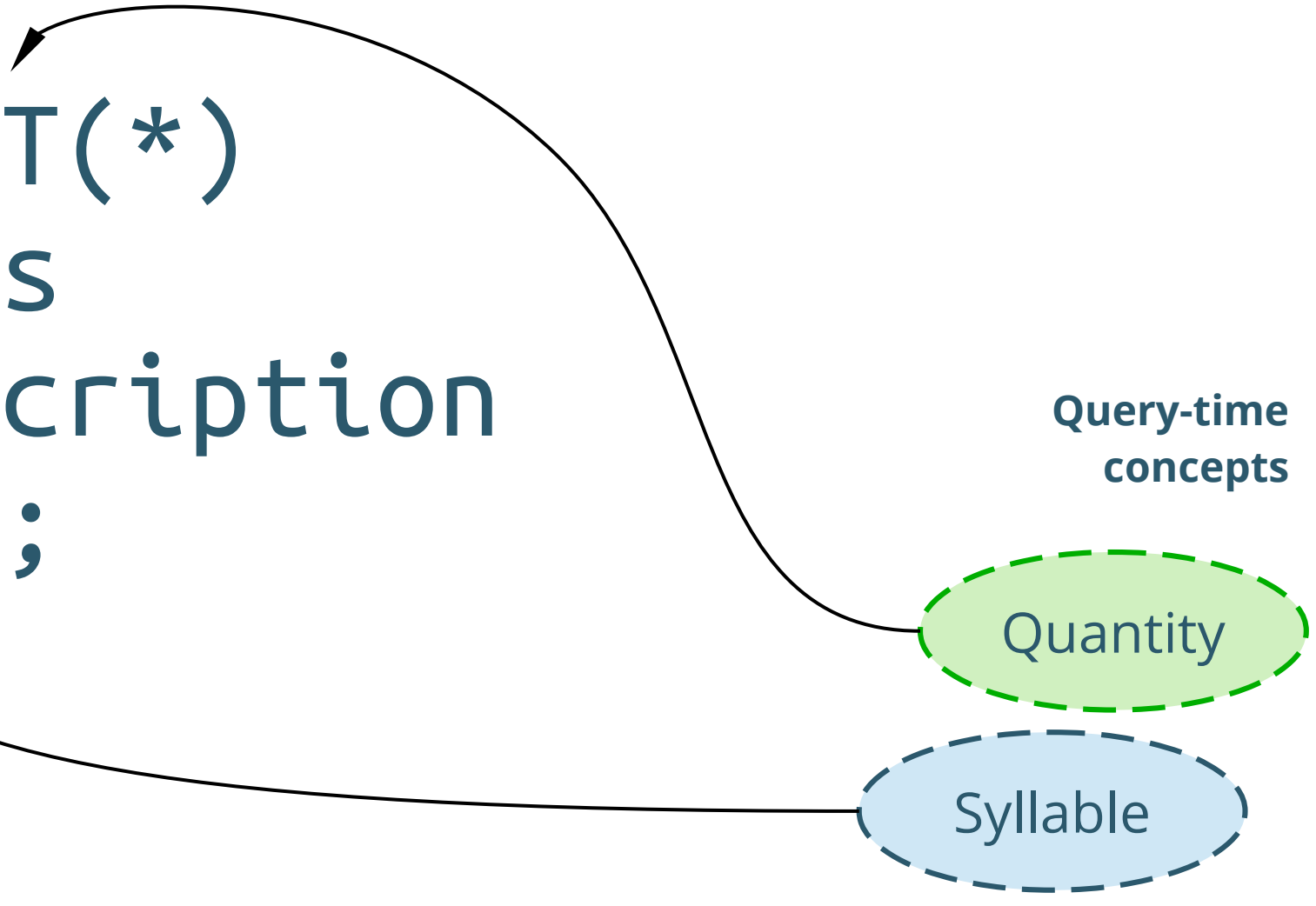
Concepts required in model

Concept → RM


Tablet
Tablets
• id [integer, PK]
• name [varchar(16)]
• city [varchar(32)]
• transcription [text]
Transcription
Query-time concepts
Quantity
Syllable

Query

```
SELECT COUNT(*)
FROM tablets
WHERE transcription
LIKE "%ja%";
```



Concept → RM → Table

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]
- transcription [text]

Tablets			
ID	name	city	transcription
1	Eb 297	Pylos	.1 i-je-re-ja , e-ke-qe
2	KN As <4493>	Knossos	.1]e-pi-ko-wo , e-qe-ta ,
3	KN Fp 5	Knossos	.1 di-wi-jo-jo 'me-no' q

Let's try!

1) Add the "Tablets" table using the Workbench

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]
- transcription [text]

2) Add data from the source

3) Run the query

```
SELECT COUNT(*) FROM tablets WHERE transcription LIKE "%ja%";
```

1. Theory: What is (not) data modelling
2. Turning research objects into tables

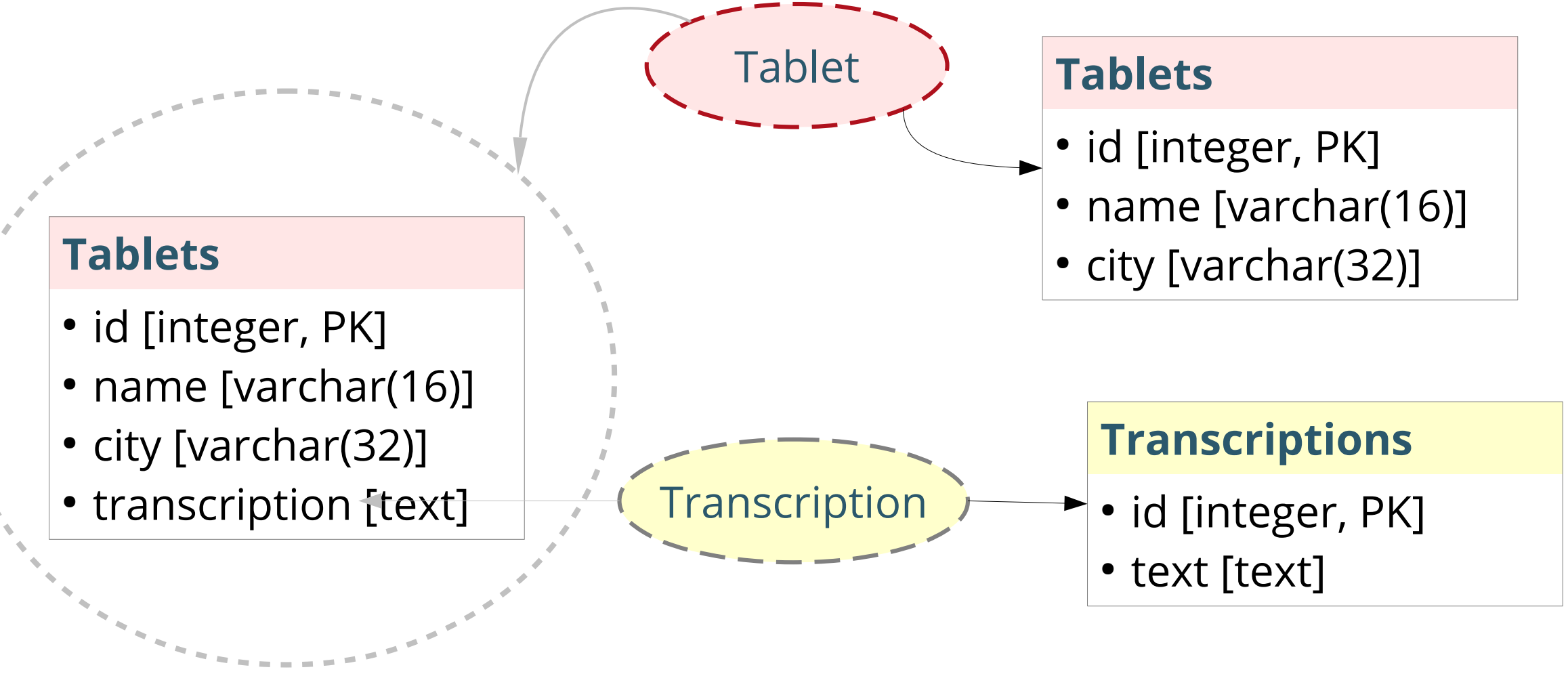
3. Establishing relations between entities

4. Extending and refining models
5. (?) Recording metadata (e.g., provenance, time, context)

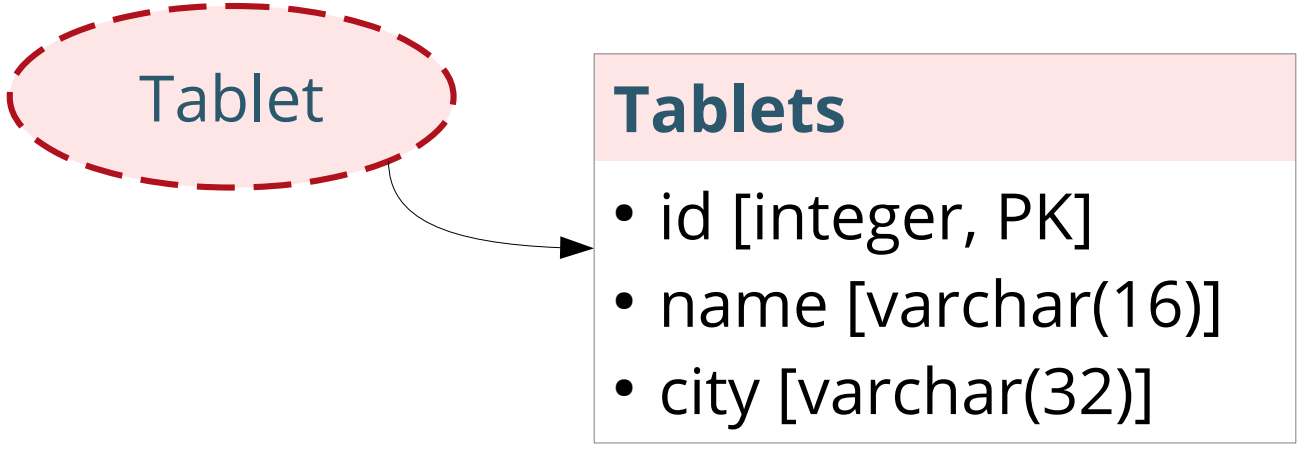
Some tablets have two transcriptions!

What should we do?

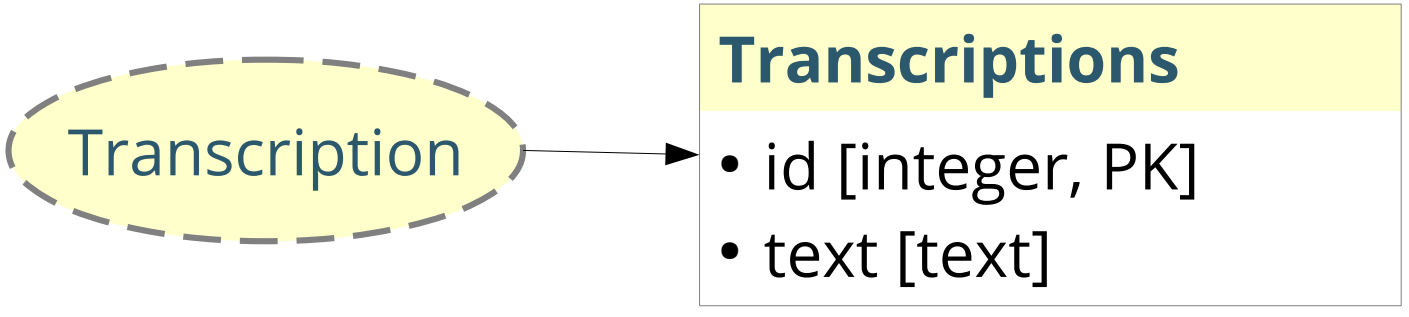
Split the Tablets table



Split the Tablets table



Issues?



References between entities

Primary key
the ID of this entity

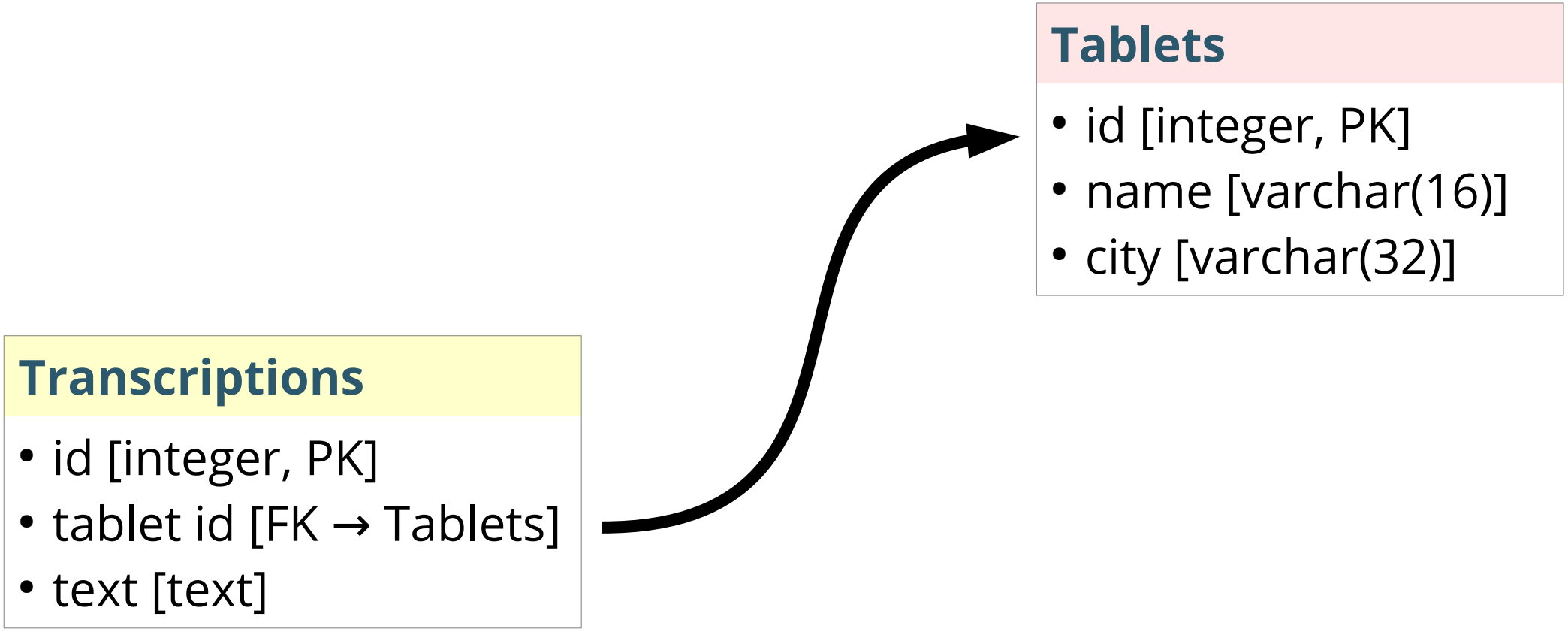
Foreign key
the ID of the other entity

Surrogate/artificial key
PK is a meaningless field

Natural key
PK is one of the field of the entity

Composite key
PK = N fields

Relationship via PK/FK keys



Query

```
SELECT COUNT(*)
FROM tablets
INNER JOIN transcriptions AS tr
ON tablets.id = tr.tablet_id
WHERE tr.text like "%ja%"
```

Let's try!

1) Add the "Transcriptions" table using the Workbench
2) Move data from Tablets
3) Create links to Tablets
4) Run the query

```
SELECT COUNT(*) FROM tablets
INNER JOIN transcriptions AS tr
ON tablets.id = tr.tablet_id
WHERE tr.text like "%na%"
```

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

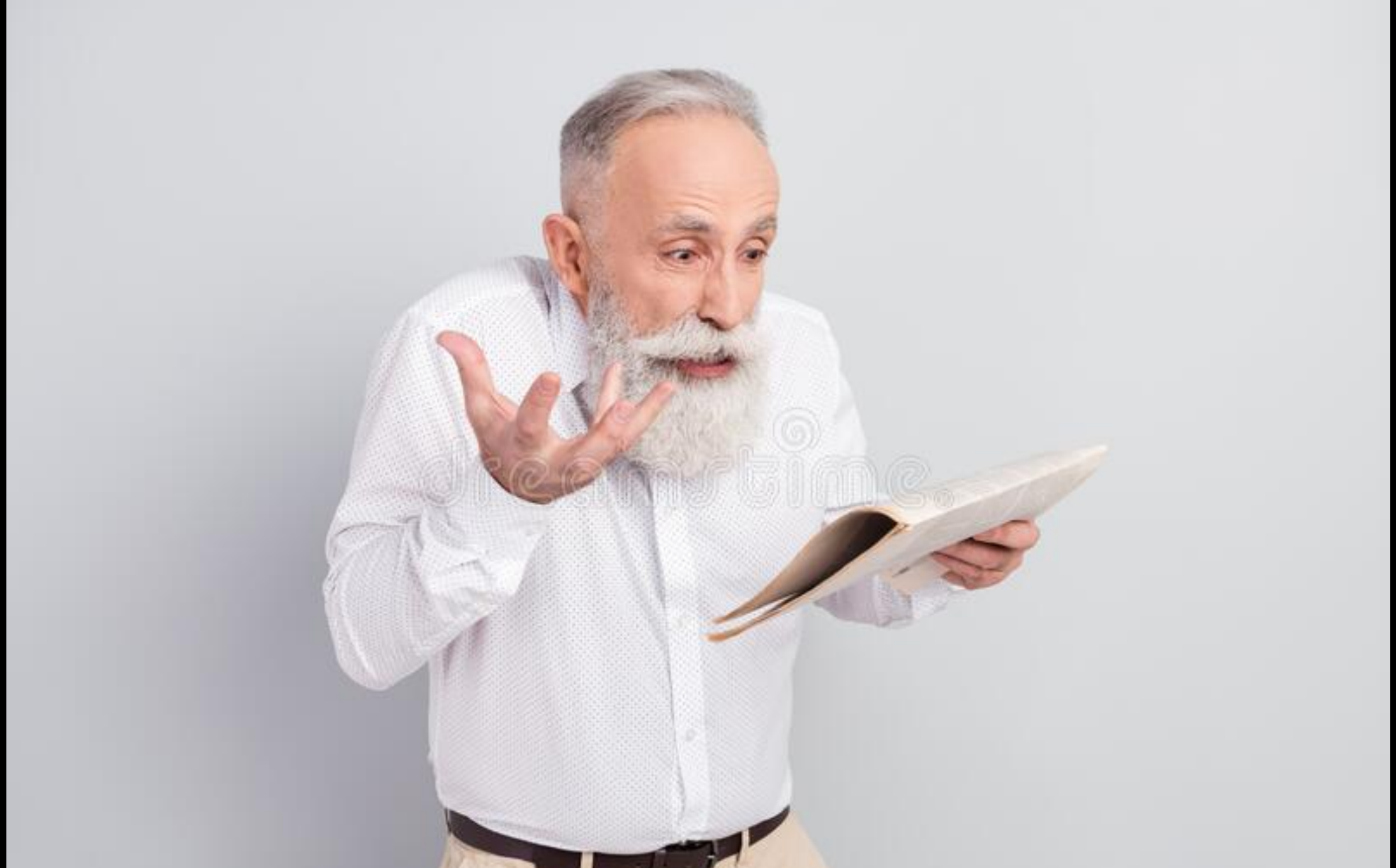
Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]
- text [text]

1. Theory: What is (not) data modelling
2. Turning research objects into tables
3. Establishing relations between entities

4. Extending and refining models

5. (?) Recording metadata (e.g., provenance, time, context)


dreamstime

Which tablets contain the syllable "ko" in more than one line?

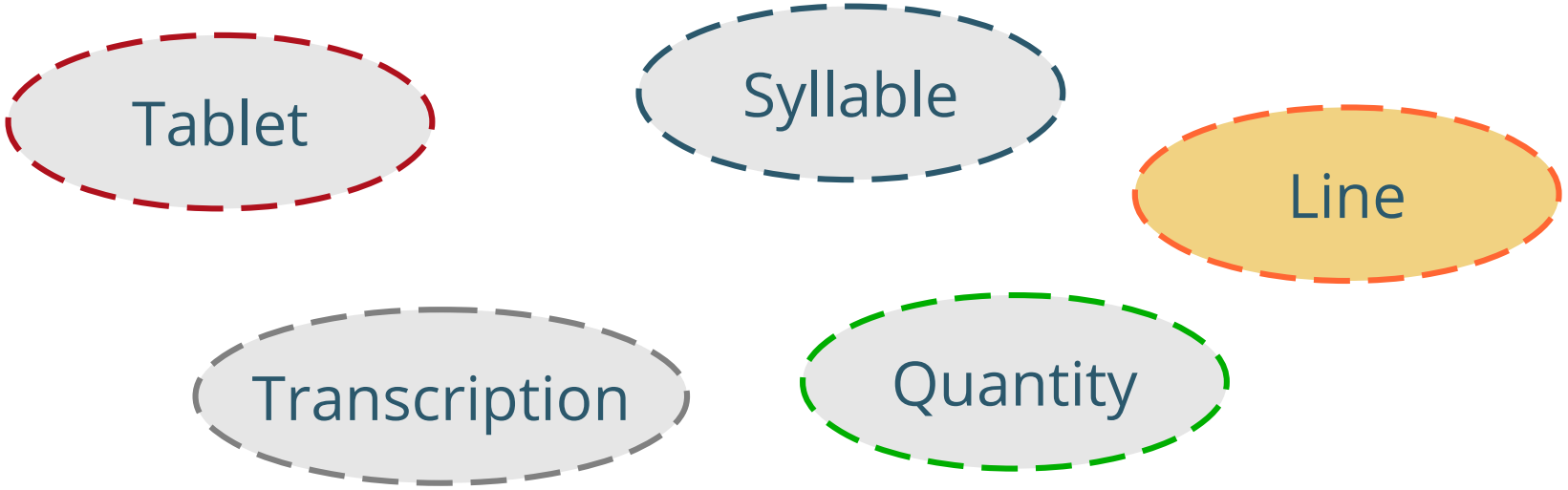
Which concepts do we need?

Which tablets contain the syllable "ko" in more than one line?

New concept: Line

Which tablets contain the syllable "ko" in more than one line?

Research question



Concepts required in model

A possible approach: Lines in Trascriptions

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]
- line 1 [text]
- line 2 [text]
- line 3 [text]
- line 4 [text]
- ...
- line 20 [text]

A possible approach: Lines in Trascriptions

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]
- line 1 [text]
- line 2 [text]
- line 3 [text]
- line 4 [text]
- ...
- line 20 [text]

Issues?

Transcriptions								
ID	Tablet ID	Line 1	Line 2	Line 3	Line 4	Line 5	...	Line 20
1	1	i-je-re-ja	e-ke-qe	NULL	NULL	NULL	...	NULL
2	1	]e-pi-ko-wo	e-qe-ta	NULL	NULL	NULL	...	NULL
3	2	di-wi-jo-jo	'me-no' q	de-ja-no-ko	NULL	NULL	...	NULL

NULLs = no value

NULLs = no value

Transcriptions								
ID	Tablet ID	Line 1	Line 2	Line 3	Line 4	Line 5	...	Line 20
1	1	i-je-re-ja	e-ke-qe	NULL	NULL	NULL	...	NULL
2	1	]e-pi-ko-wo	e-qe-ta	NULL	NULL	NULL	...	NULL
3	2	di-wi-jo-jo	'me-no' q	de-ja-no-ko	NULL	NULL	...	NULL

Transcription is *denormalized*

Better approach: Lines as separate table

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]

Lines

- id [integer, PK]
- text [text]

Better approach: Lines as separate table

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]

Issues?

Lines

- id [integer, PK]
- text [text]

Better approach: Lines as separate table

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

Lines

- id [integer, PK]
- transcription id [FK → Tr]
- text [text]

Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]

Better approach: Lines as separate table

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

Lines

- id [integer, PK]
- transcription id [FK → Tr]
- text [text]

Issues?

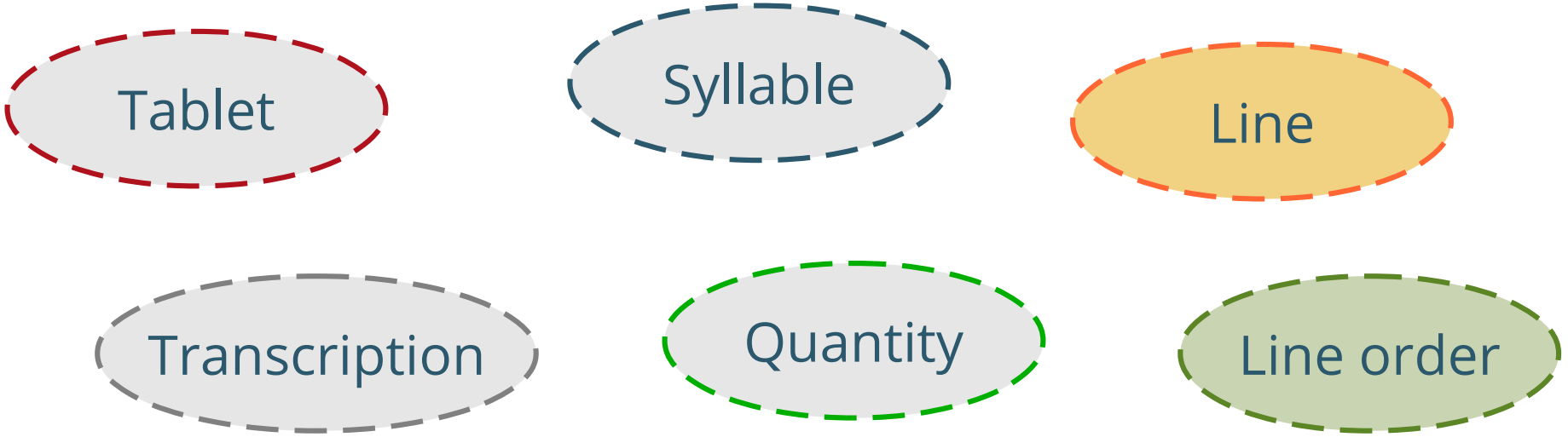
Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]

Implicit comcept: Line order

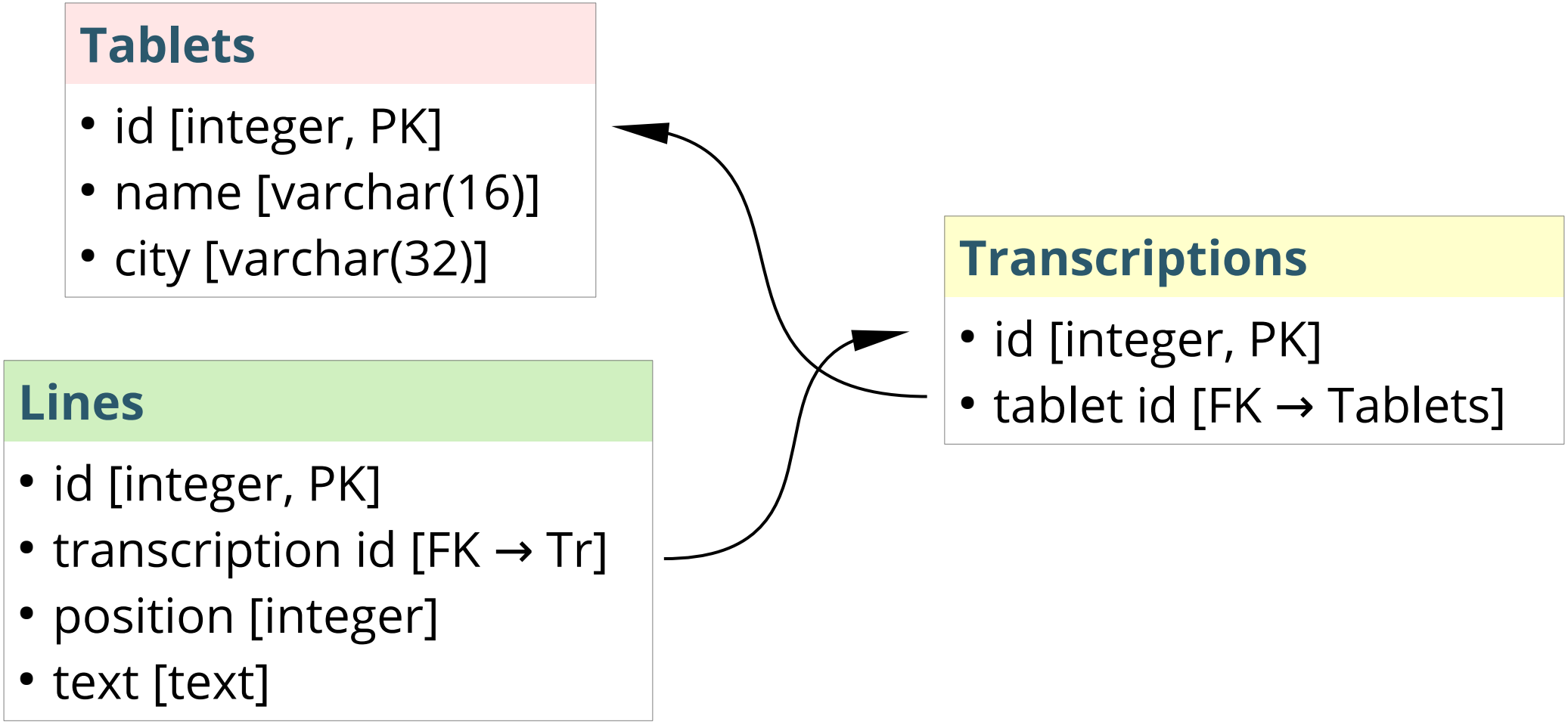
Which tablets contain the syllable "ko" in more than one line?

Research question



Concepts required in model

Better approach: Lines as separate table



Query

```
SELECT *
FROM tablets
INNER JOIN transcriptions AS tr
ON tablets.id = tr.tablet_id
INNER JOIN lines
ON tr.id = lines.trascription_id
WHERE lines.text like "%ko%"
GROUP BY lines.transcription_id
HAVING COUNT(lines.text) >= 2;
```

Normalization forms

1NF: No tabular data in fields

- "One field, one piece of data"
- Lines in Transcriptions.text

2NF: No repetition of field-related data

- "Don't repeat the metadata"
- City in Tablets if we add additional data

3NF: No data unrelated to the Primary Key

- "Don't merge two tables in one"
- tablet_name, tablet_city, line_position, line_text

Modeling guidelines

OntoClean

- Nicola Guarino and Chris Welty. 2002. Evaluating Ontological Decisions with OntoClean

Kimball Lifecycle

- Ralph Kimball et al. (1998). The Data Warehouse Lifecycle Toolkit.

Database normalization forms

Let's try!

1) Add the "Lines"

2) Move data from Transcriptions

3) Create links to Transcriptions

4) Run the query

```
SELECT *
FROM tablets
INNER JOIN transcriptions AS tr
ON tablets.id = tr.tablet_id
INNER JOIN lines
ON tr.id = lines.trascription_id
WHERE lines.text like "%ko%"
GROUP BY lines.transcription_id
HAVING COUNT(lines.text) >= 2;
```

Tablets

- id [integer, PK]
- name [varchar(16)]
- city [varchar(32)]

Transcriptions

- id [integer, PK]
- tablet id [FK → Tablets]

Lines

- id [integer, PK]
- transcription id [FK → Tr]
- position [integer]
- text [text]

1. Theory: What is (not) data modelling
2. Turning research objects into tables
3. Establishing relations between entities
4. Extending and refining models

5. **Recording metadata** (e.g., provenance, time, context)

ENCODE — Bridging the gap in Ancient Writing Cultures
Oslo 2022-10-10

Introduction to data modeling for the Humanities

Gioele Barabucci

NTNU | Norwegian University of Science and Technology