

Data Models and Databases 101

Berkeley Copwatch / WITNESS workshop
28 April 2018

Objectives

- Become familiar with what databases are, why they can be useful, and the steps to designing an effective database.
- Come to agreed understanding of purpose of next phase of BC database, what we want it to enable us to do.
- Collaboratively design new data model, schema, and data dictionary, building upon work that has already been done.

What is a database?

Datum or **data element**: a piece of factual information.

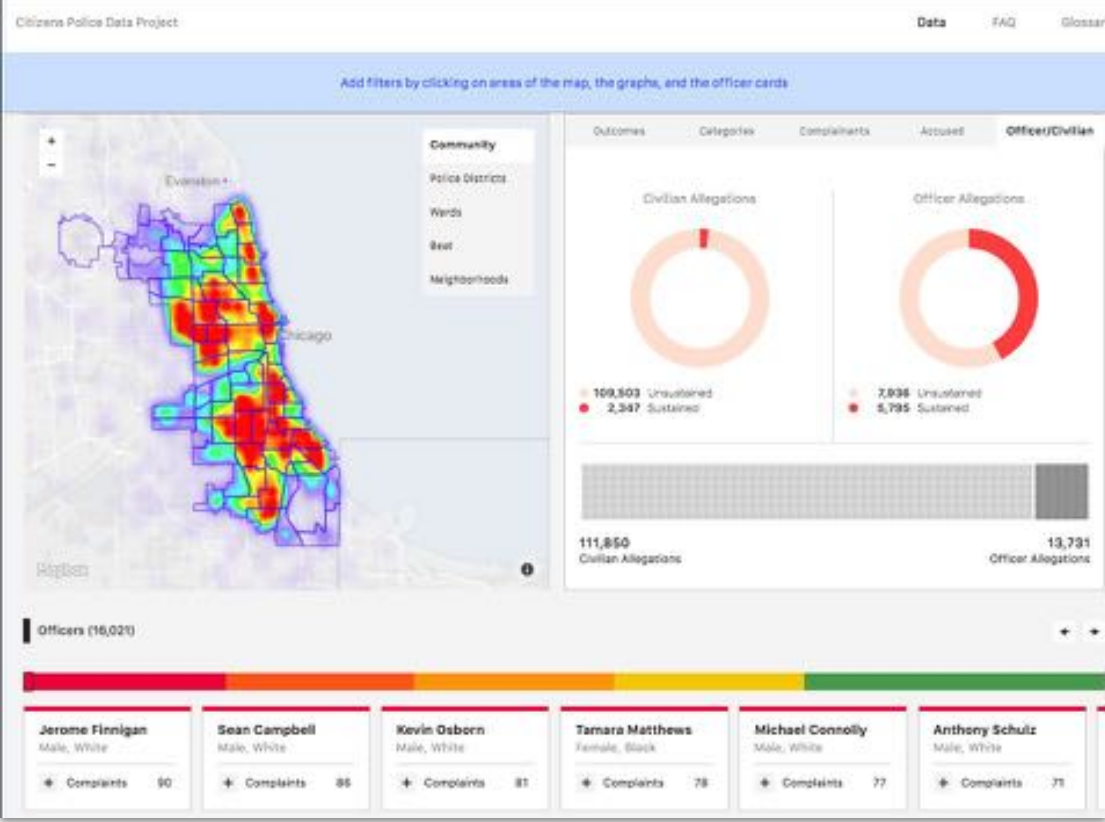
Database: a collection of related data that represents some aspect of the real world, and that is designed and populated to serve a specific purpose.

Typically implemented using **database management software** that allows users to retrieve selected/filtered data, sort, count or calculate data.

How can a database be useful to you?

- An accessible and centralized place to store information you want to hold on to.
- Structured data can harness computing power to sort, match, link, calculate, aggregate information, and see it in different ways.
- Structured data is easier to share and invite collaboration on.
- Enabling strategic use and exchange of information.

Some examples of databases



Beat	OfficerID	OfficerFirst	OfficerLast	AllegationCategory	Allegation	Recommend	Recommended
5038862	3051	Margaret	Birkenmayr	DOT	Operation/ Reports	NA	500
5069989	23420	Jenny	Santiago	OSC	Illegal Search	Search Of F UN	500
5088255	32365	Faul	Samara	OSD	False Arrest	Illegal Arrest UN	500
5079488	26445	Mario	Simonetti	OSB	Illegal Search	Improper S NA	500
5090241	31415	Michael	Youman	OSA	Use Of Force	Excessive Fi UN	500
5072292	4539	Michael	Chlebok	ODD	Operation/	Slow / No R NA	500
5049697	27981	Gregory	Sweney	ODU	Operation/	Inadequate UN	500
5053598	23448	Reginald	Randle	OSD	False Arrest	Illegal Arrest NA	500
504482	32155	Laurence	Stiles	OSL	Domestic	Domestic In NS	500
504163	7852	Joseph	Blayer	ODU	Operation/	Inadequate NS	500
507401	23793	Carmelo	Royce	OSI	Conduct Un	Association SU	505
507413	16337	Matthew	Little	OSC	Illegal Search	Search Of F UN	500
505567	9425	Alejandro	Gallejos	OSA	Use Of Force	Excessive Fi NS	500
299608	3897	Thomas	Carney	ODV	Operation/	Inventory F ZZ	500
5009902	11863	Edvard	Handewald	OSG	Illegal Search	Improper S NA	500
507006	25108	Christopher	Sackett	DOT	Operation/	Reports UN	500
5030778	9725	Joe	Gathings	ODU	Operation/	Inadequate NA	500
5082170	27962	John	Swarbrick	OSG	Use Of Force	Unnecessary ZZ	500
5071731	22692	Jeffery	Planey	OSA	Use Of Force	Excessive Fi NS	500
5033831	21785	Michael	Ryces	ODU	Operation/	Inadequate ER	500
5072296	24128	Joseph	Rizzi	OSD	False Arrest	Illegal Arrest NA	500
297230	15014	Paul	Kopacz	OSD	False Arrest	Illegal Arrest NS	500
280290	16495	Ricardo	Lopez	OSL	Domestic	Domestic In NS	500
5054267	6009	Brian	Cygnar	OSC	Illegal Search	Search Of F NA	500
5029664	10417	Matthew	Gordan	OSB	Use Of Force	Excessive Fi UN	500
507945	13946	Robert	Jurczykowi	ODF	Lockup	Pris Escape UN	500
5029715	15511	Michael	Lachance	ODV	Operation/	Inventory F NA	500
5009911	17572	Nensid	Markovitch	OSC	Illegal Search	Search Of F NA	500
5035213	29688	Jason	Vogler	ODU	Operation/	Inadequate UN	500
288621	9100	William	Frapoly	ODI	Operation/	Neglect Of F NS	500
513687	5467	Ernest	Corber	OSA	Supervision	State Civil T UN	500

Invisible Institute, Citizen Police Data Project

File a Complaint Using these officer details

Now that you've found the offending officer, it's time to submit a complaint to the relevant oversight body. For complaints regarding Chicago Police Department officers, complaints are accepted through the link below. Clicking below will open a new browser tab where you can copy the officer details from this page into the complaint form when requested.

Warning! The following link will open an external site in a new browser tab:

[File Complaint Online](#)



Officer name: Robert Cesario
Officer badge number: #10

```
13 class Department(db.Model):
14     __tablename__ = 'departments'
15     id = db.Column(db.Integer, primary_key=True)
16     name = db.Column(db.String(255), index=True, unique=True, nullable=False)
17     short_name = db.Column(db.String(100), unique=False, nullable=False)
18
19     def __repr__(self):
20         return '<Department ID {}: {}>'.format(self.id, self.name)
21
22
23 class Officer(db.Model):
24     __tablename__ = 'officers'
25
26     id = db.Column(db.Integer, primary_key=True)
27     last_name = db.Column(db.String(120), index=True, unique=False)
28     first_name = db.Column(db.String(120), index=True, unique=False)
29     middle_initial = db.Column(db.String(120), unique=False, nullable=True)
30     race = db.Column(db.String(120), index=True, unique=False)
31     gender = db.Column(db.String(120), index=True, unique=False)
32     employment_date = db.Column(db.DateTime, index=True, unique=False, nullable=True)
33     birth_year = db.Column(db.Integer, index=True, unique=False, nullable=True)
34     assignments = db.relationship('Assignment', backref='officer', lazy='dynamic')
35     face = db.relationship('Face', backref='officer', lazy='dynamic')
36     department_id = db.Column(db.Integer, db.ForeignKey('departments.id'))
37     department = db.relationship('Department', backref='officers')
38
39     def __repr__(self):
40         return '<Officer ID {}: {} {} {}>'.format(self.id,
41                                                     self.first_name,
42                                                     self.middle_initial,
43                                                     self.last_name)
44
45
```

Q I'm looking for...

Search for ... Go!

9 results found

⚠ Incidents 9 results Download results

General filters

Date

Start date

End date

Country

Mexico

Filters for Incidents

Violation type

Arbitrary Arrest

Illegal Detention

Start Date	End Date	Description	Violation types	Location	Perpetrators
17 March 2008	18 March 2008	According to Comisión Nacional de los Derechos Humanos: "El 18 de marzo de 2008, esta Comisión Nacional, recibió la queja formulada por Q1, en la	Torture Arbitrary Arrest Illegal Detention	Campo Militar Número 6-B, Mexico	33 Batallón de Infantería
1 April 2008	1 April 2008	According to Comisión Nacional de Derechos Humanos: "El 1° de abril de 2008, esta Comisión Nacional recibió la queja formulada por la señora Cinthia Noemí ...	Right To Liberty Personal Integrity Beating Arbitrary Arrest Detention Torture	installations of 20 Regimiento de Caballería Motorizado, Mexico	20 Regimiento de Caballería Motorizado PGR SIEDO
7 June 2008	7 June 2008	According to Comisión Nacional de los Derechos Humanos: "El 7 de junio de 2008, esta Comisión Nacional recibió la queja formulada por la señora Cinthia Noemí ...	Right To Liberty Personal Integrity Beating Arbitrary Arrest Detention Torture	installations of 3 Compañía de Infantería No Encuadrada, Mexico	3 Compañía de Infantería No Encuadrada

	A	B	C	D	E	F	G	H	I
1	violation_id	start_date	end_date	location_desc	admin_level	admin_level	osurname	osm_id	division
2	ba4ad418-64	3/17/08	3/18/08	Campo Militar Número 6-B					ocd-div
3	ba4ad418-64	3/17/08	3/18/08	Campo Militar Número 6-B					ocd-div
4	ba4ad418-64	3/17/08	3/18/08	Campo Militar Número 6-B					ocd-div
5	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
6	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
7	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
8	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
9	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
10	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
11	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
12	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
13	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
14	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
15	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
16	fddec299-b9i	4/1/08	4/1/08	installations of 20 Regimiento de Caballería Motorizado					ocd-div
17	85756bb3-8f	6/7/08	6/7/08	installations of the 3 Compañía de Infantería No Encuadrada					ocd-div
18	85756bb3-8f	6/7/08	6/7/08	installations of the 3 Compañía de Infantería No Encuadrada					ocd-div
19	85756bb3-8f	6/7/08	6/7/08	installations of the 3 Compañía de Infantería No Encuadrada					ocd-div
20	85756bb3-8f	6/7/08	6/7/08	installations of the 3 Compañía de Infantería No Encuadrada					ocd-div
21	85756bb3-8f	6/7/08	6/7/08	installations of the 3 Compañía de Infantería No Encuadrada					ocd-div
22	85756bb3-8f	6/7/08	6/7/08	installations of the 3 Compañía de Infantería No Encuadrada					ocd-div
23	ad80e08-9c	6/27/08	6/27/08	Campo Militar Número 5-C					ocd-div
24	ad80e08-9c	6/27/08	6/27/08	Campo Militar Número 5-C					ocd-div
25	ad80e08-9c	6/27/08	6/27/08	Campo Militar Número 5-C					ocd-div
26	ad80e08-9c	6/27/08	6/27/08	Campo Militar Número 5-C					ocd-div
27	ad80e08-9c	6/27/08	6/27/08	Campo Militar Número 5-C					ocd-div

Security Force Monitor, "Who was in Command" database

Form completed by: _____ Your Phone Number: _____
Berkeley COPWATCH File Creation Date: 4/12/2008
Case Interview Form (for 2nd hand accounts) Document I.D.: CW-456

Information on Complainant

First Name: _____ Last Name: _____
 Street Address: _____
 City: _____ State: _____ Zip: _____
 Phone: _____
 Relationship to Victim: _____

Information on Victim

First Name: _____ Last Name: _____
 Street Address: _____
 City: _____ State: _____ Zip: _____ Age: _____
 Phone: _____ Gender: _____
 Race/Ethnicity: _____ Victim's sexual orientation: _____
 Immigration status: _____ Date of birth: _____
 Criminal history related to incident: _____

Information on the Officers

badge#	officer l. name	officer r. name	sex	gender	dept.	use it

Nature of Complaint

Police violence
 Harassment
 Wrongful arrest
 Justice system complaint
 Policy or Other

Information on the incident

Location of incident: _____
 Incident date: _____ Incident time: _____
 What happened?

Manage Database for "Incident Reports Copy"

Tables Fields Relationships

The relationships graph provides access to data in one table from another. If a relationship is defined between two tables (even through another table), fields from one table can be accessed from the other.

*Case work data-base current Converted

- Action needed
- Action needed from COPWATCH
- Action taken by COPWATCH
- agency
- badge #
- Bring Justice?
- Call From
- Call To
- Changes
- Clere's plan
- Complainant's Age
- Complainant's City
- Complainant's First Name
- Complainant's Gender
- Complainant's Last Name
- Complainant's Race/Ethnicity
- Complainant's State
- Complainant's Street Address
- Complainant's Zip
- Complaint accepted?
- complaint sustained
- Confidentiality?
- Copwatcher
- Copwatcher Phone Number
- court date and location
- Criminal history related to incident
- date of birth
- Date of call
- dept
- Description of any injuries/damages to persons & properties
- discrimination
- Discussion
- Do you have copies of medical records
- document I.D.
- evidence:
- File Creation Date
- filed a complaint?
- first or second hand
- Form completed by:
- immigration status
- incident date
- incident time
- interested in organizing with people in similar situation
- law enforcement agency
- Location of incident
- misconduct
- Need help finding witnesses
- Need help getting them
- neighborhood codes
- officer l. name
- Officer gender

Tables / Relationships Arrange Tools Pages

Print... Cancel OK

Berkeley Copwatch, internal database

Database vs. archive

- **Archive:** An organization (systems & people) that preserves and provides access to information objects (e.g. files, videos, documents, artworks, etc).
- **Database:** Compiled and structured data that serves some purpose.
- Archives often have databases, to serve purpose of search and retrieval of objects. Archive databases contain a type of data called “**metadata**” - i.e. data that describes information object (e.g. data about a video, or data about a document).

Database vs. archive, Berkeley Copwatch

- BC has both a database and a video archive, that mostly function separately.
- Do we want them to be more integrated? The database can contain data about policing *and* metadata about videos.

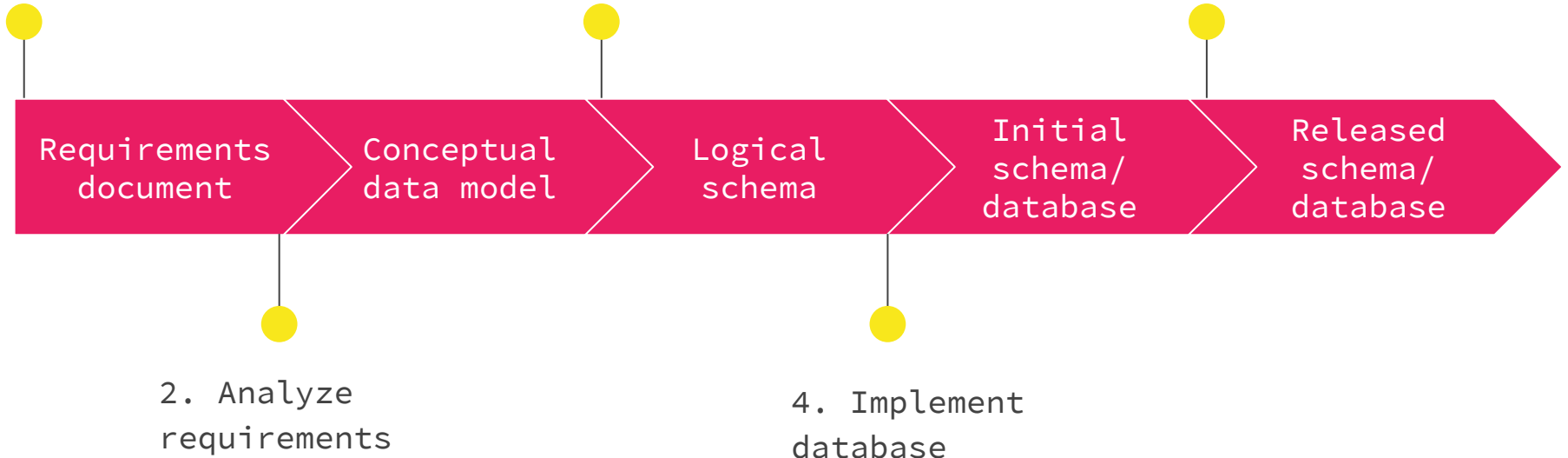
If so, let's make sure to include in data modeling, up next...

Database development process

1. Gather requirements

3. Design database

5. Test database



1. Gathering requirements

Creating a common understanding of why, who we are creating this database for, what those users need to be able to do.

Databases provide responses to queries.

- **What questions do we want to be able to answer using the data?**

Example: “Who Was In Command” database

What is WhoWasInCommand?

WhoWasInCommand answers key questions about the structure, behaviour and people in charge of security forces like the police and army:

- Who is in charge of the specialized anti-riot police unit?
- What army unit has jurisdiction over what areas and for how long?
- Where did this commander previously serve, and where did they go next?
- When was a particular police unit based in a specific city?
- What allegations have civil society groups made against a unit or commander?

WhoWasInCommand presents data from thousands of public sources to help human rights researchers, investigative journalists and anyone who wants security forces to be more accountable.

[Learn more →](#)

Berkeley Copwatch requirements: discussion

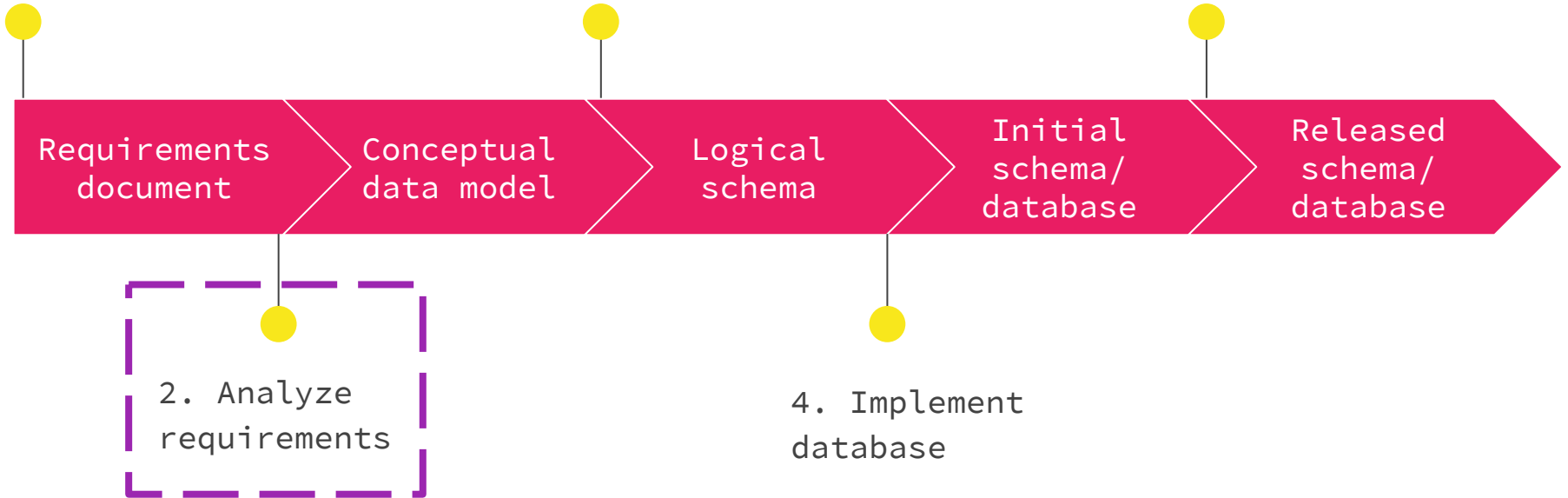
- What should the purpose of this next phase of the Berkeley Copwatch database be?
- Review [preliminary set of questions](#). Is this the information we want to know? Any additions / revisions?

Database development process

1. Gather requirements

3. Design database

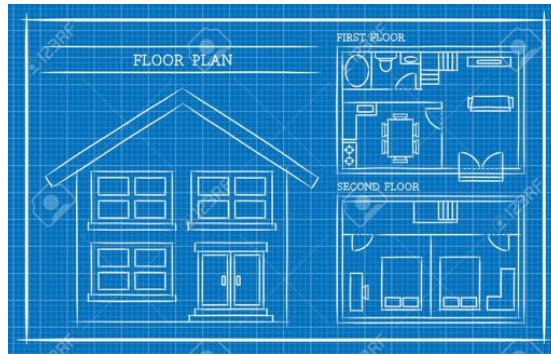
5. Testing



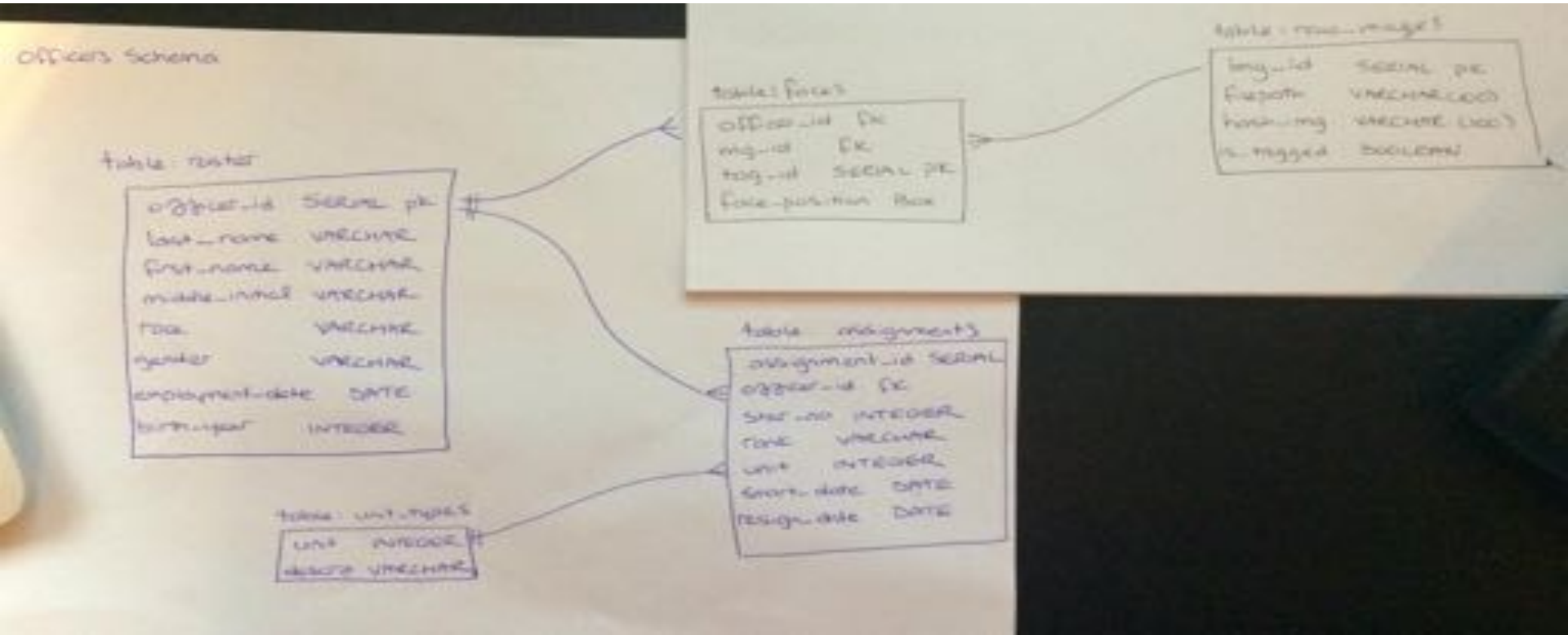
2. Analyzing requirements → Data model

Examining requirements and determining what data needs to be in the database, how data items need to be structured, in order to answer your questions.

→ High-level “blueprint” that outlines your data “house”.



Example: (early) data model from Open Oversight



Same (updated) data model, represented in another way

```
13 class Department(db.Model):
14     __tablename__ = 'departments'
15     id = db.Column(db.Integer, primary_key=True)
16     name = db.Column(db.String(255), index=True, unique=True, nullable=False)
17     short_name = db.Column(db.String(100), unique=False, nullable=False)
18
19     def __repr__(self):
20         return '<Department ID {}: {}>'.format(self.id, self.name)
21
22
23 class Officer(db.Model):
24     __tablename__ = 'officers'
25
26     id = db.Column(db.Integer, primary_key=True)
27     last_name = db.Column(db.String(120), index=True, unique=False)
28     first_name = db.Column(db.String(120), index=True, unique=False)
29     middle_initial = db.Column(db.String(120), unique=False, nullable=True)
30     race = db.Column(db.String(120), index=True, unique=False)
31     gender = db.Column(db.String(120), index=True, unique=False)
32     employment_date = db.Column(db.DateTime, index=True, unique=False, nullable=True)
33     birth_year = db.Column(db.Integer, index=True, unique=False, nullable=True)
34     assignments = db.relationship('Assignment', backref='officer', lazy='dynamic')
35     face = db.relationship('Face', backref='officer', lazy='dynamic')
36     department_id = db.Column(db.Integer, db.ForeignKey('departments.id'))
37     department = db.relationship('Department', backref='officers')
```

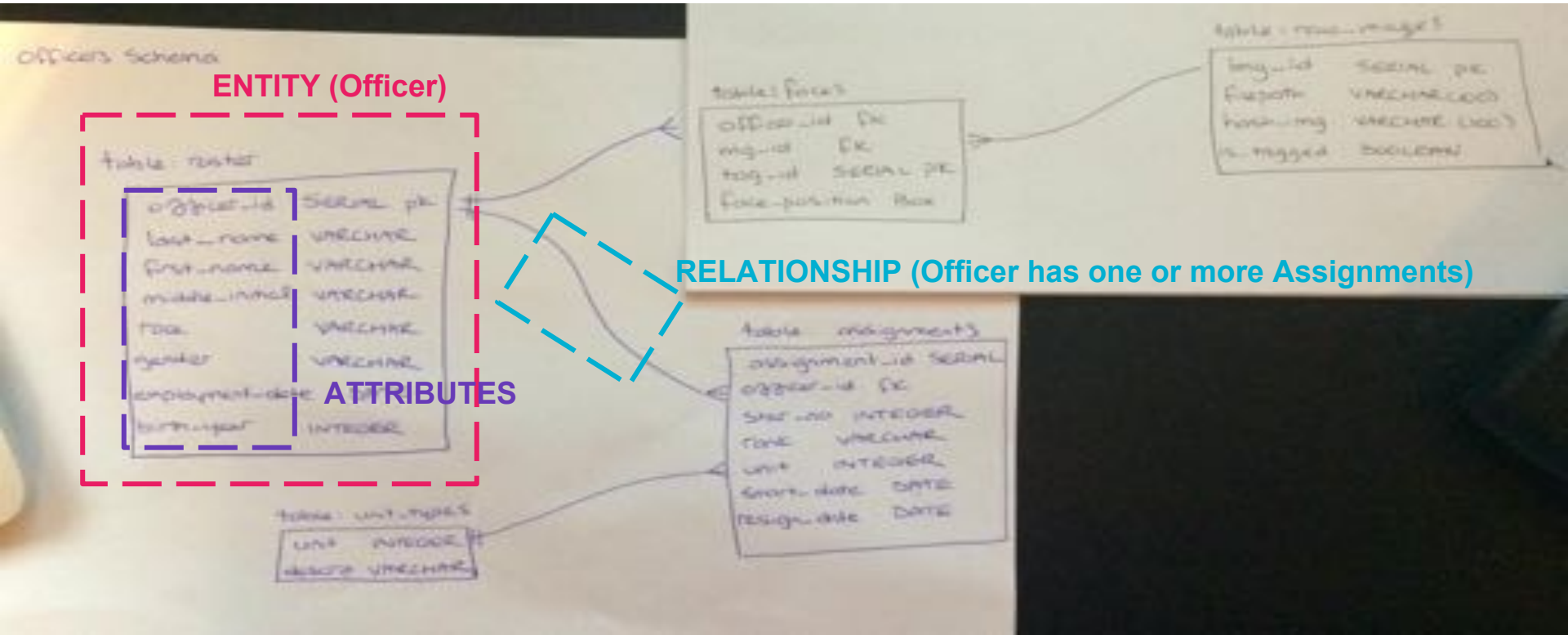
Components of a data model

Entities: The real-life things (people, objects, events, etc) that the database contains information *about*. E.g. officers, events, videos.

Attributes: Aspects of the entities, i.e. the fields within each entity. E.g. Officer: First name, last name, shield number.

Relationships: How different entities are associated with each other. E.g. Video *documents* an Event.

Example: Open Oversight



Example: Open Oversight

```
23 class Officer(db.Model):
24     __tablename__ = 'officers'
25
26     id = db.Column(db.Integer, primary_key=True)
27     last_name = db.Column(db.String(120), index=True, unique=False)
28     first_name = db.Column(db.String(120), index=True, unique=False)
29     middle_initial = db.Column(db.String(120), index=True, unique=False)
30     race = db.Column(db.String(120), index=True, unique=False)
31     gender = db.Column(db.String(120), index=True, unique=False)
32     employment_date = db.Column(db.DateTime, index=True, unique=False)
33     birth_year = db.Column(db.Integer, index=True, unique=False)
34     assignments = db.relationship('Assignment', backref='officer')
35     face = db.relationship('Face', backref='officer')
36     department_id = db.Column(db.Integer, db.ForeignKey('departments.id'))
37     department = db.relationship('Department', backref='officers')
```

ENTITY (Officer)

ATTRIBUTES

```
45
46 class Assignment(db.Model):
47     __tablename__ = 'assignments'
48
49     id = db.Column(db.Integer, primary_key=True)
50     officer_id = db.Column(db.Integer, db.ForeignKey('officers.id'))
51     baseofficer = db.relationship('Officer', backref='assignments')
52     star_no = db.Column(db.String(120), index=True, unique=False)
53     rank = db.Column(db.String(120), index=True, unique=False)
54     unit = db.Column(db.Integer, db.ForeignKey('unit_types.id'), nullable=True)
55     star_date = db.Column(db.DateTime, index=True, unique=False, nullable=True)
56     resign_date = db.Column(db.DateTime, index=True, unique=False, nullable=True)
57
```

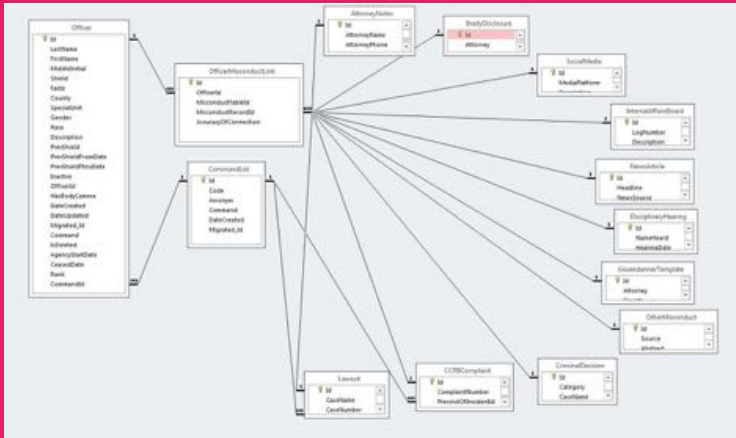
RELATIONSHIP

Let's Data Model

Based on the questions we want to answer

What we're going to do next:

- Identify entities.
- Identify attributes for those entities.
- Identify relationships between entities.



Identifying entities

Looking back at our questions, **what are the real-life things (people, objects, events) that we are asking questions *about*?**

Example: WhoWasInCommand's questions:

- Who is in charge of the specialized anti-riot police unit?
- What army unit has jurisdiction over what areas and for how long?
- Where did this commander previously serve, and where did they go next?
- When was a particular police unit based in a specific city?
- What allegations have civil society groups made against a unit or commander?

Example: Entities in WhoWasInCommand data model

Security Force Monitor researches and creates data about three things (or entities) related to security forces around the world:

- **Organizations** are official state or state-sanctioned organizations responsible for the internal or external security for a country, including police, army, navy, air force and other security bodies. Organizations refer to any any part of the hierarchy of a security force, ranging from a national defense ministry, to a police unit based in a small town. Organizations can also be groupings of organizations that occur, such as "operations", "joint task forces" or peacekeeping missions. The Monitor collects data about an organization's name, aliases, location, geographical areas of operation and relationships with other organizations.
- **Persons** are natural persons who are affiliated with, or hold positions of command over a specific organization at a particular point in time. The Monitor creates a dossier for each person, which includes their name, aliases, rank, title, role and the different organizations which they are affiliated with.
- **Events** are publicly-documented allegations of human rights violations committed by security forces. These include extrajudicial killings, rape, torture and other forms of violence. The Monitor does not make allegations itself, but rather compiles allegations made by governmental bodies, human rights organizations and other civil society actors. For each Event, the Monitor includes a description from the source, date(s), specific location(s), its perpetrators and the type of human rights violation.

Exercise: Identifying entities

From our [requirements/questions](#), what/who are we trying to answer questions *about*?

- Start with the explicit subjects & objects of questions.
- Entities might not be explicit, e.g. “Do BPD officers profile by race?” -- what are we counting in order to answer this question?
- Consider if some things can be generalized into a broader entity (e.g. “searches” and “raids” as types of “Events”).

Clarify: What does having (separate) entities mean?

EVERYTHING						
Incident Date	Incident Location	Victim Name	Victim Age	Victim Gender	Complainant Name	Complainant Address

VS.

INCIDENT	
Incident Date	Incident Location

VICTIM		
Victim Name	Victim Age	Victim Gender

COMPLAINANT	
Complainant Name	Complainant Address

Why have entities / relationships?: Examples

- Can create a record for Victim without necessarily creating a Incident record first. Or can delete an Incident record without losing data about Victim.
- If updating Complainant Address for complainant with multiple complaints, only need to do it once. Less chance of inconsistency/error.
- Can associate multiple Victims to a single Incident without creating sorting / counting problems.

Identifying attributes

- Next, what are the properties/aspects of each entity that will help us answer our questions?

E.g. “Person” entity might include: name, race, gender, immigration status, health status, etc. etc.

Example: Legal Aid database

Officer entity

(Note: as you can see, it's not 100% clear what some of these attributes mean. That's why we need a **data dictionary** -- coming up later!)



Exercise: Identifying attributes

Working in pairs(?), take one of the entities we've identified, and brainstorm a list of attributes on the [worksheets](#), which we'll post on the wall.

Refer to:

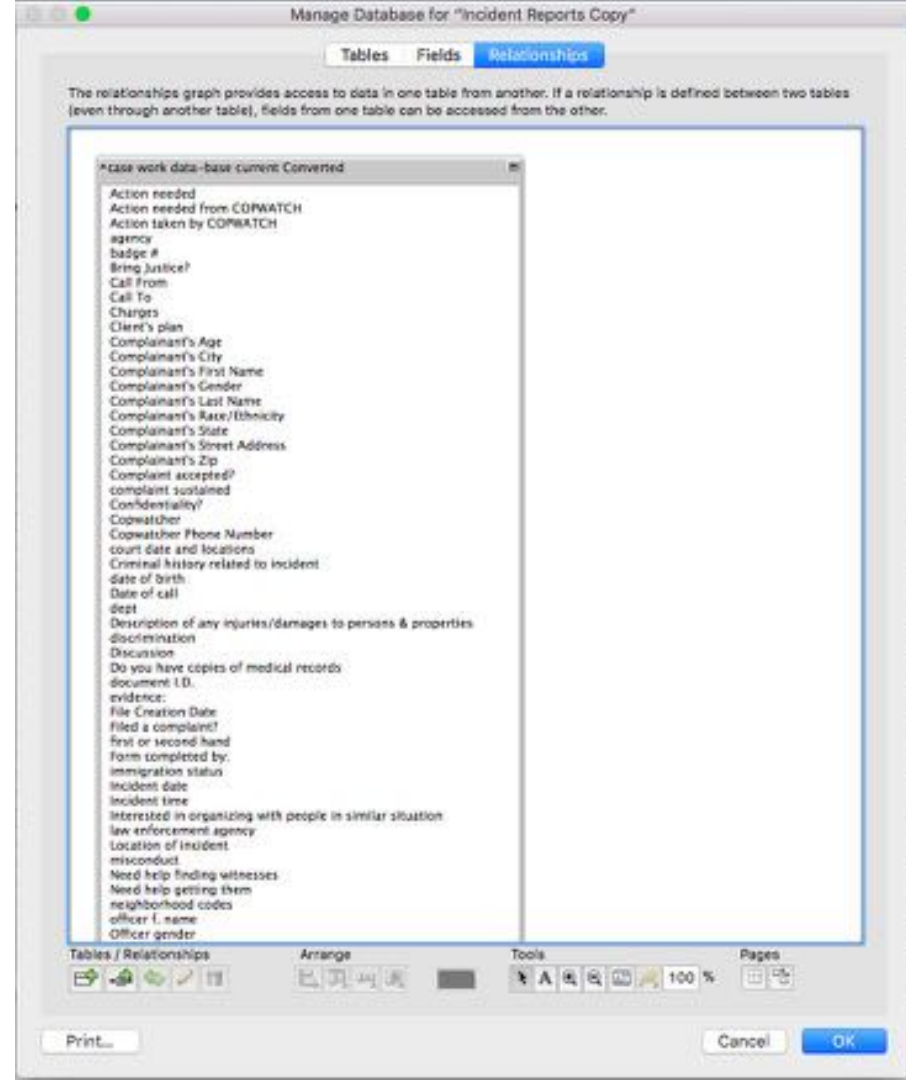
- Our questions / requirements list.
- Current Berkeley Copwatch database attributes (up next).

Tips for identifying attributes

- Any entity can have an infinite number of attributes. Choose ones that are important for identifying entity and answering questions.
- If there's an important attribute that doesn't belong to any of the entities we've identified, we will make a new entity. Don't shoehorn attribute into entity it doesn't describe.
- Consider what data you realistically have.

Current Berkeley Copwatch database

- [Attributes list](#) from DB
- Forms templates ([1](#),[2](#),[3](#))
- Currently in one entity (incidents). Let's use these attributes but store under new entities.



Last data modeling step! Identifying relationships

Drawing the relationships that we want to examine between the entities.

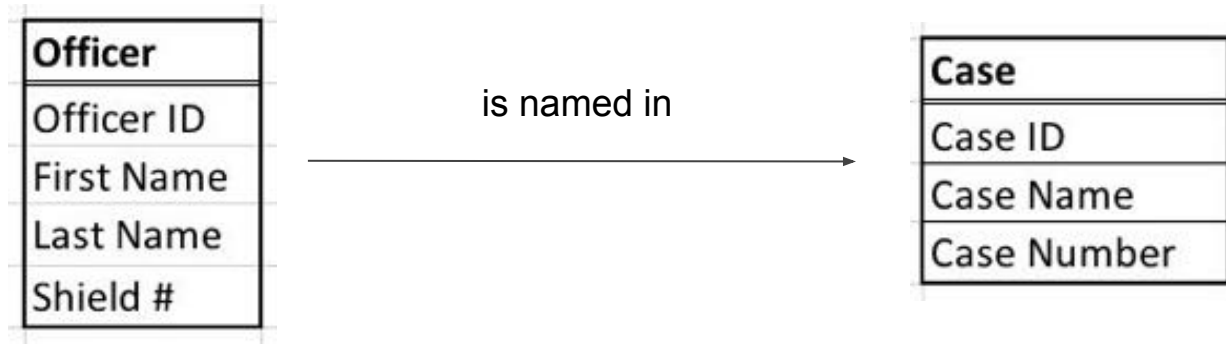
Many possible relationships, but create ones that:

- Reflect real-life relationships.
- Are as simple as possible, while allowing us to answer our questions.
- Tip: try stating the relationship in a “entity verb entity” sentence, e.g. “Videos show an Event”

Example: Identifying relationships

“Which officers have a history of lawsuits against them”?

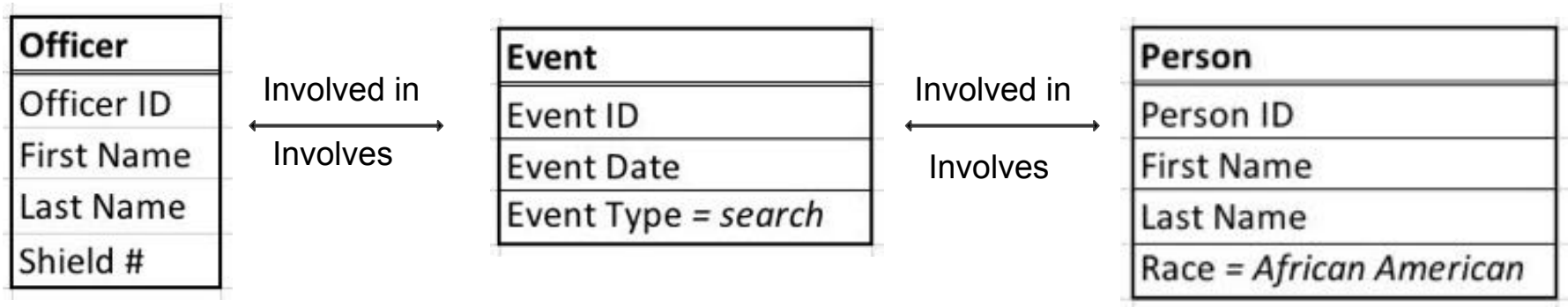
To answer this question, we should connect officer and lawsuit entities:



Example: Identifying relationships

“Do officers conduct searches based on people’s race?”

This question can involve 3 related entities.



Exercise: Identifying relationships

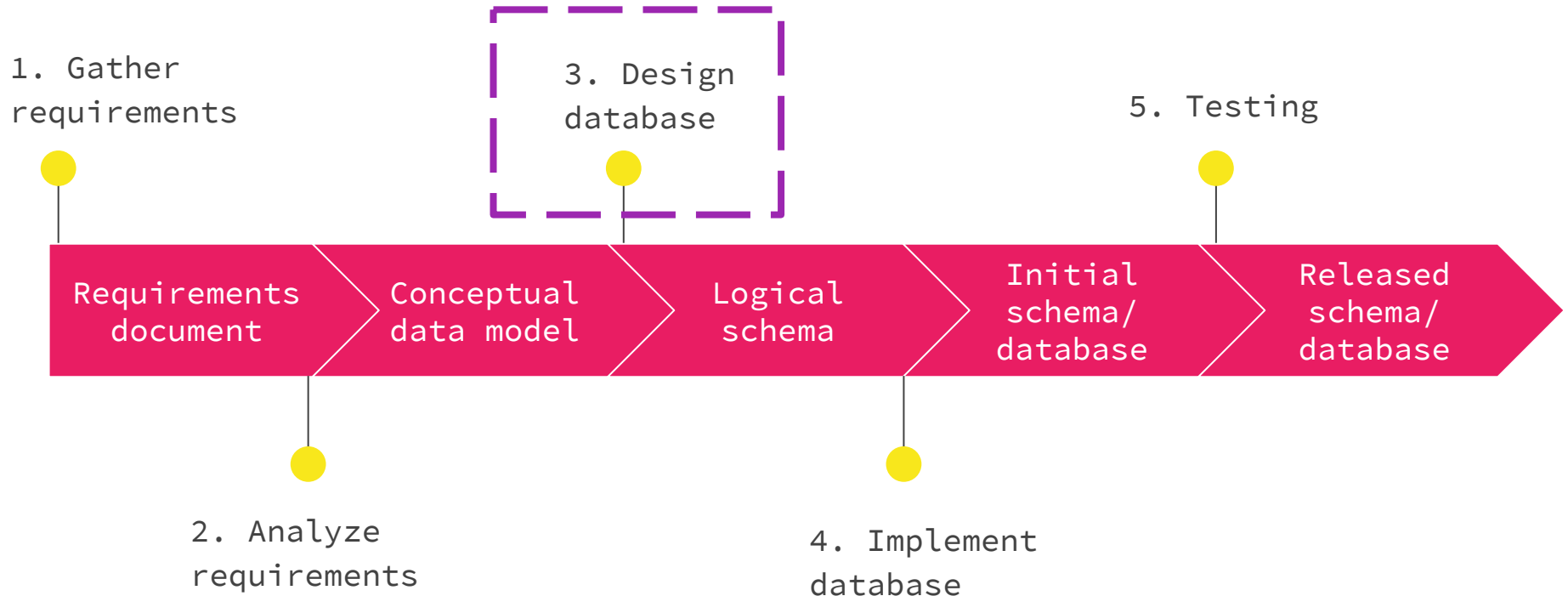
Using the string, let's map out the relationships we want to explore between entities.

Relationships should:

- Help us answer our questions.
- Reflect real-life association.
- As simple as possible, avoiding redundancy and loops.

Congratulations!
We have a conceptual data model!

Database development process - where we are



3. Design database: Elaborating the data model

Logical data model: Follows from conceptual model, but goes into more technical detail about the database structure (e.g. keys or ids, direction of relationship lines, additional tables to handle many-to-many relationships, etc).

Can use this to implement database in any database software.

Often visualized as an Entity Relationship Diagram.

Design database: setting data rules

Database design also involves setting rules for each data element:

- Definition of the field/attribute.
- Controlled vocabularies or defined value lists.
- Syntax rules for values.
- Whether a value is required or optional.

These rules can be documented in a **Data Dictionary**.

Example: Data dictionary

A	B	C	D	E	F
ELEMENT NAME	FFN ID	Source ID	Title(s)	Description	Tags
ELEMENT DEFINITION	Record reference number.	ID given to source media.	Cataloger-supplied title based on content.	Cataloger-supplied description based on viewing the content.	Cataloger-supplied keywords.
ELEMENT RULES	"F" followed by 5-digit sequential number.	Enter as written on source media.	Generally, enter text as given on the source (label), verbatim, unless correction or more elaboration is needed. Title on source is the text written on the spine of the tape, or the top text written on the DVD. Multiple titles should be separated with " "	Generally, enter text as given on the source (label), verbatim, unless correction or more elaboration is needed (e.g. names, locations, affiliations, etc). Multiple descriptions should be separated with " " in the same order as titles are entered.	Any words that describe the video that may help users search for content. Separate terms with commas.
ELEMENT REQUIREMENT	Required	Required.	Required	Optional	Optional
EXAMPLE ENTRY	F00001	S019	Black 14 Gathering / Rumah Panjang Jinjang Pongal	Anwar Ibrahim speech in Kelab Sultan Sulaiman.	Kampung Baru, press conference, Anwar Ibrahim, election, protest, Malaysia

Exercise: Data Dictionary

Let's focus on the attributes that would be helpful to define as a group.

For controlled vocabularies, we can use current [BC database value lists](#) as a start.

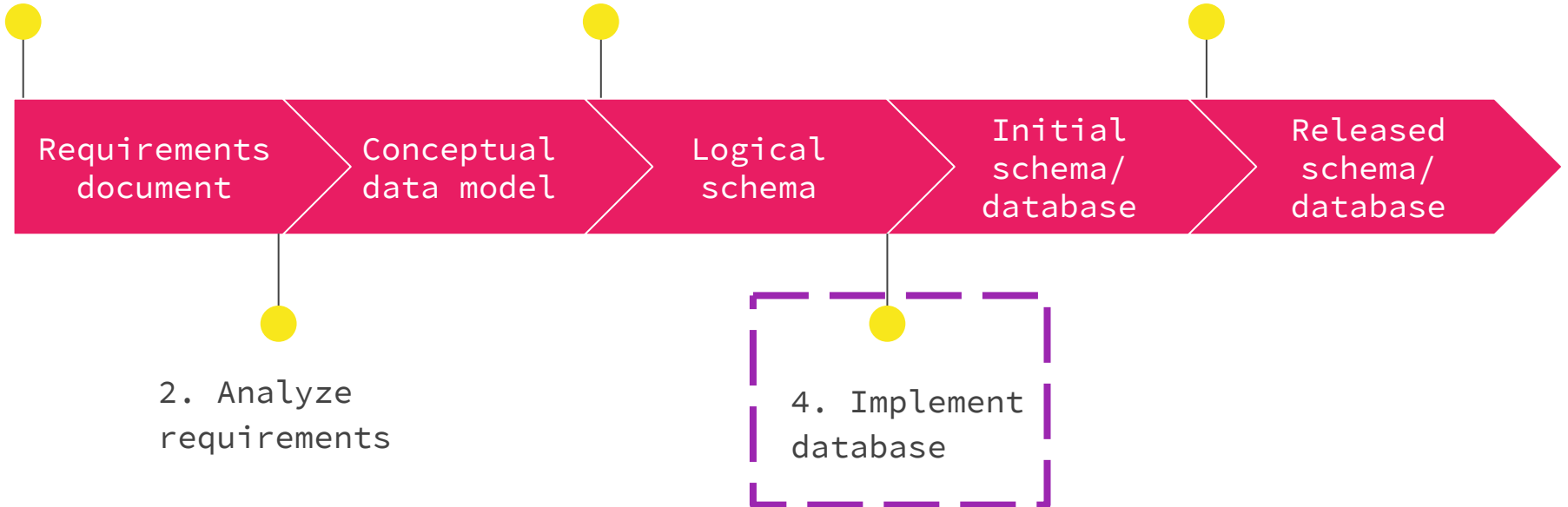
- Any attributes need to be more clearly defined?
- Any attributes that need a controlled vocabulary? Do value terms need to be defined?
- Do we need to apply specific rules to any elements?

Database development process - next steps

1. Gather requirements

3. Design database

5. Testing



Implementation: Next steps

- Implement the database for initial testing.
- Choose appropriate database management software / platform.
- Create views / presentation of the data (may have different views for different user types).