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
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
Security Force Monitor, “Who was in Command” database
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
2. Analyze requirements
3. Design database
4. Implement 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
2. Analyze requirements
3. Design database
4. Implement database
5. Testing

- Requirements document
- Conceptual data model
- Logical schema
- Initial schema/database
- Released schema/database
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

```python
class Department(db.Model):
    __tablename__ = 'departments'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(255), index=True, unique=True, nullable=False)
    short_name = db.Column(db.String(100), unique=False, nullable=False)

    def __repr__(self):
        return '<Department ID {}> {}: {}>'.format(self.id, self.name)

class Officer(db.Model):
    __tablename__ = 'officers'
    id = db.Column(db.Integer, primary_key=True)
    last_name = db.Column(db.String(128), index=True, unique=False)
    first_name = db.Column(db.String(128), index=True, unique=False)
    middle_initial = db.Column(db.String(128), unique=False, nullable=True)
    race = db.Column(db.String(128), index=True, unique=False)
    gender = db.Column(db.String(128), index=True, unique=False)
    employment_date = db.Column(db.DateTime, index=True, unique=False, nullable=True)
    birth_year = db.Column(db.Integer, index=True, unique=False, nullable=True)
    assignments = db.relationship('Assignment', backref='officer', lazy='dynamic')
    face = db.relationship('Face', backref='officer', lazy='dynamic')
    department_id = db.Column(db.Integer, db.ForeignKey('departments.id'))
    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

ENTITY (Officer)

ATTRIBUTES

RELATIONSHIP (Officer has one or more Assignments)
Example: Open Oversight

class Officer(db.Model):
    __tablename__ = 'officers'

    id = db.Column(db.Integer, primary_key=True)
    last_name = db.Column(db.String(120), index=True, unique=False)
    first_name = db.Column(db.String(120), index=True, unique=False)
    middle_initial = db.Column(db.String(120), index=True, unique=False)
    race = db.Column(db.String(120), index=True, unique=False)
    gender = db.Column(db.String(120), index=True, unique=False)
    employment_date = db.Column(db.DateTime, index=True, unique=False)
    birth_year = db.Column(db.Integer, index=True, unique=False)
    department_id = db.Column(db.Integer, index=True, unique=False)
    department = db.relationship('Department', backref='officers')

class Assignment(db.Model):
    __tablename__ = 'assignments'

    id = db.Column(db.Integer, primary_key=True)
    officer_id = db.Column(db.Integer, db.ForeignKey('officers.id'))
    baseofficer = db.relationship('Officer', backref='officers')
    star_no = db.Column(db.String(120), index=True, unique=False)
    rank = db.Column(db.String(120), index=True, unique=False)
    unit = db.Column(db.String(120), db.ForeignKey('unit_types.id'), nullable=True)
    star_date = db.Column(db.DateTime, index=True, unique=False)
    resign_date = db.Column(db.DateTime, index=True, unique=False)
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 complies 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?

VS.
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

“How which officers have a history of lawsuits against them”?

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

“This question can involve 3 related entities.

<table>
<thead>
<tr>
<th>Officer</th>
<th>Involved in</th>
<th>Involved in</th>
<th>Involved in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Officer ID</td>
<td>First Name</td>
<td>Last Name</td>
<td>Shield #</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>Involved in</th>
</tr>
</thead>
</table>
| Event ID | Race = *African American*
| Event Date | |
| Event Type = *search* | |

<table>
<thead>
<tr>
<th>Person</th>
<th>Involved in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person ID</td>
<td>First Name</td>
</tr>
<tr>
<td>Last Name</td>
<td></td>
</tr>
<tr>
<td>Race = <em>African American</em></td>
<td></td>
</tr>
</tbody>
</table>
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

1. Gather requirements
2. Analyze requirements
3. Design database
4. Implement database
5. Testing

- Requirements document
- Conceptual data model
- Logical schema
- Initial schema/database
- Released schema/database
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.
Example: El Grito 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**

<table>
<thead>
<tr>
<th>ELEMENT NAME</th>
<th>FFN ID</th>
<th>Source ID</th>
<th>Title(s)</th>
<th>Description</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEMENT RULES</td>
<td>&quot;F&quot; followed by 5-digit sequential number.</td>
<td>Enter as written on source media.</td>
<td>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 &quot;/&quot; in the same order as titles are entered.</td>
<td>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 &quot;/&quot; in the same order as titles are entered.</td>
<td>Any words that describe the video that may help users search for content. Separate terms with commas.</td>
</tr>
<tr>
<td>ELEMENT REQUIREMENT</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Optional</td>
<td>Optional</td>
</tr>
</tbody>
</table>
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
2. Analyze requirements
3. Design database
4. Implement database
5. Testing

Requirements document → Conceptual data model → Logical schema → Initial schema/database → Released schema/database
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).