ActivityInfo

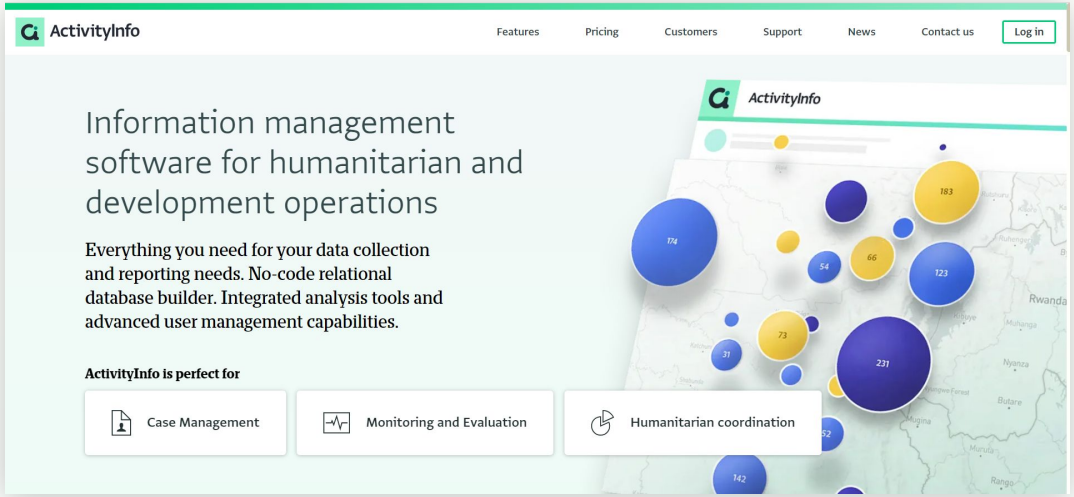
Fundamentals of data modelling in humanitarian and development contexts

Starting shortly, Please wait!

Presented by the ActivityInfo Team

All in one information management software for humanitarian and development operations

- Track activities, outcomes
- Beneficiary management
- Surveys
- Work offline/online



Meet your instructor

Victoria Manya
Customer Education Specialist
BeDataDriven

ActivityInfo

Webinar Series

The series title

July 10

SESSION 1

Fundamentals of data modelling in humanitarian and development contexts

July 24

SESSION 2

Data modelling in practice

ActivityInfo

Outline

01 Overview of data modelling

02 The data modelling process

03 Introduction to data modelling in practice

04 Q/A session

Overview of data modelling

QUIZ?

Question:

What has been your biggest challenge when working with data models in your projects?

A: Ensuring data integrity and consistency.

B: Managing relationships between complex entities.

C: Avoiding data redundancy and optimizing storage.

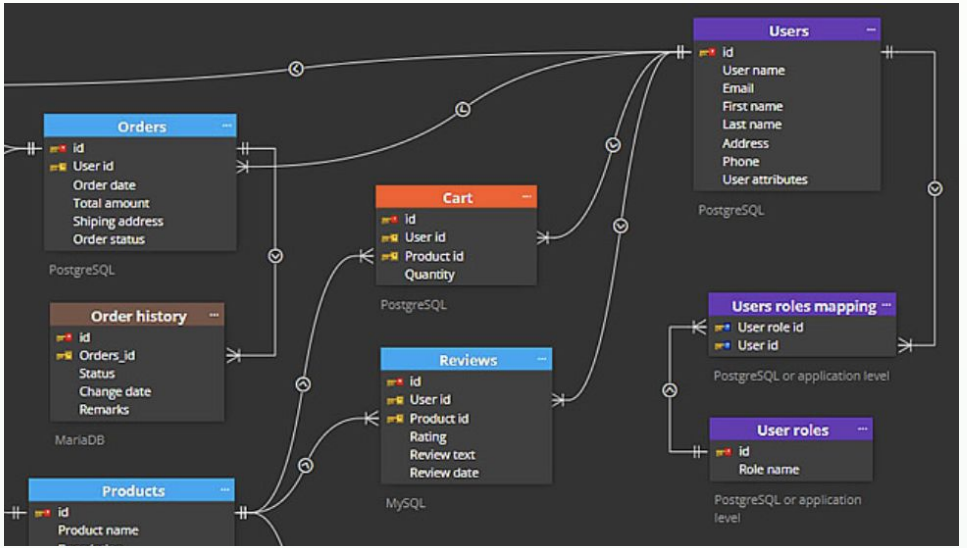
D: Understanding and implementing normalization rules.

E: Communicating data model designs to non-technical stakeholders.

What is a data model

A data model is a **visual representation** of a conceptual framework that organizes and defines data elements and shows how they interact with each other.

By mapping out data structures and their relationships in a visual format, it provides a method by which data is stored, organized, and retrieved.



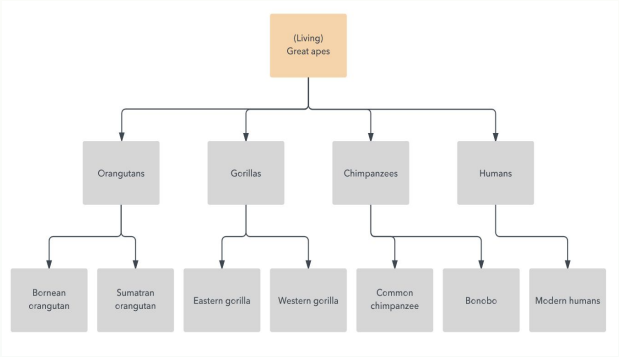
Datensen:2024

Types of data models

Hierarchical database model

Relational model

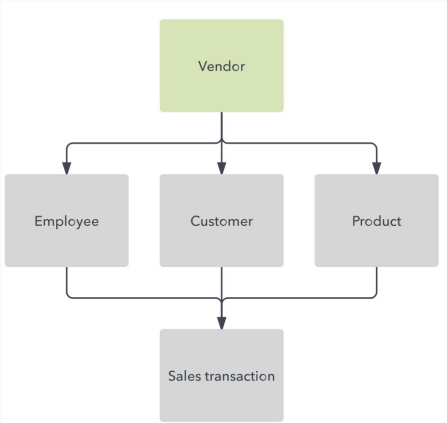
Network model



Student ID	First name	Last name
52-743965	Charles	Peters
48-209689	Anthony	Sondrup
14-204968	Rebecca	Phillips

ProviderID	Provider name
156-983	UnitedHealth
146-823	Blue Shield
447-784	Carefirst Inc.

Student ID	ProviderID	Type of plan	Start date
52-743965	156-983	HSA	04/01/2016
48-209689	146-823	HMO	12/01/2015
14-204968	447-784	HSA	03/14/2016

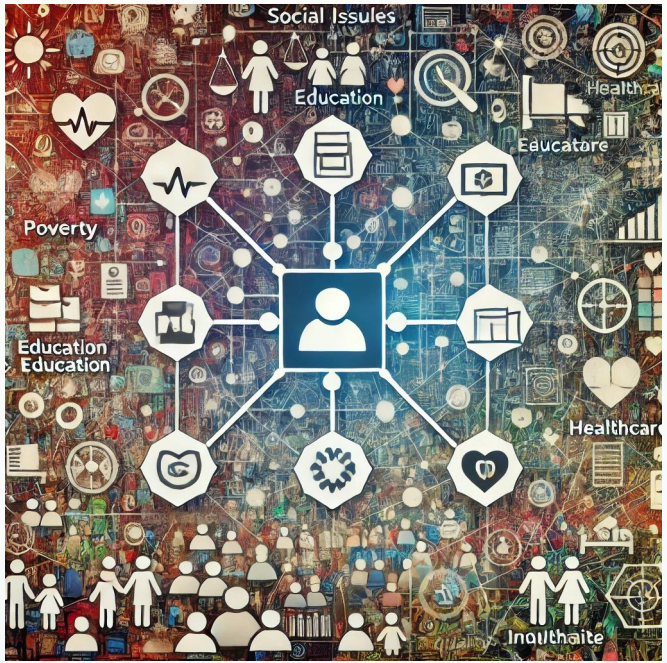


ActivityInfo

[Lucid chart]

Designing a data model: importance

The complexity of social issues make imperative, the need for a clear data model.



ActivityInfo

Why create a data model

01	Data collection, organization,integrity and analysis	• structure data, thus easier to store, retrieve, and manipulate • data integrity by defining rules and constraints • Facilitates analysis by outlining possible analytical approaches based on existing data relationships.	Accuracy, reliability, consistency, efficiency
02	Security	• Who needs to access data and why • Policy requirements	Data protection
03	Scalability and documentation	• Blueprint on how data can be expanded • Provide a clear and structured representation of the data,	Maintenance and improvements
04	Data integration	• Organizations have multiple data sources • Define how different systems relate	Facilitate data consolidation
05	Communication	• Common language for communication between different stakeholders	Shared understanding

ActivityInfo

Fundamentals principles and frameworks

Entity-Relationship Principle

Entities: Represent real-world objects or concepts involved in humanitarian work. For example, entities could include beneficiaries,relief Items, donors, and distribution centers.

Relationships: Define how these entities interact with each other. For instance, beneficiaries receive relief items or donors fund distribution centers.

Attributes: Characteristics of these entities, such as beneficiary ID, name, age, and location for "Beneficiaries" or item type, quantity, and expiration date for "Relief Items."

Example:

- **Entity**: Beneficiary
  - **Attributes**: ID, Name, Age, Gender, Location
- **Entity**: Relief Item
  - **Attributes**: Item ID, Item Name, Quantity, Expiration Date
- **Relationship**: Beneficiaries receive Relief Items

Fundamentals principles and frameworks

Normalization

Purpose: Reduces redundancy and ensures data integrity by organizing data into smaller, related tables. This is crucial in humanitarian databases to avoid duplicating beneficiary information or relief items, ensuring accurate and up-to-date records..

Process: Involves dividing a large table into smaller tables and defining relationships between them

Example:

- **1st Normal Form (1NF)**: Ensure all fields contain atomic (indivisible) values. E.g., separate full name into first name and last name.
- **2nd Normal Form (2NF)**: Remove subsets of data that apply to multiple rows and place them in separate tables. E.g., create separate tables for "Locations" linked to "Beneficiaries" by location ID.
- **3rd Normal Form (3NF)**: Ensure all fields depend only on the primary key. E.g., remove fields that do not directly describe the beneficiary, such as item attributes.

Fundamentals principles and frameworks

Data Integrity

Ensuring Accuracy: Use constraints like primary keys (unique identifiers for each record) and foreign keys (to maintain relationships between tables) to enforce data integrity.

Validation: Implement validation rules to ensure data entered into the system meets predefined criteria, such as valid age ranges or location codes.

Example:

- **Primary Key**: Beneficiary ID in the "Beneficiaries" table.
- **Foreign Key**: Location ID in the "Beneficiaries" table linking to the "Locations" table.
- **Validation Rule**: Ensure age is within a reasonable range (e.g., 0-20 years).

Fundamentals principles and frameworks

Scalability and Flexibility

Scalability: Design data models that can handle increasing volumes of data, such as more beneficiaries or additional relief items, without significant redesign.

Flexibility: Allow for modifications to accommodate changing requirements, such as new types of aid or additional data attributes.

Example:

- **Flexible Schema**: Add new attributes to the "Beneficiaries" table (e.g., health status) without affecting existing data structures.
- **Scalable Model**: Use partitioned tables to manage large volumes of data efficiently.

Fundamentals principles and frameworks

Simplicity and Clarity

User-Friendly Models: Design data models that are easy to understand and use by both technical and non-technical staff. This simplicity helps in training field workers and ensuring accurate data entry.

Clear Documentation: Maintain comprehensive documentation for the data model, including entity definitions, relationships, and attributes.

Example:

- **Simple Relationships**: Clearly defined relationships between tables, such as a straightforward link between "Beneficiaries" and "Relief Items."
- **Documentation**: Detailed explanations of each entity and attribute, including data types and constraints.
    a. **Data Dictionary**: A document listing all entities, attributes, and their definitions.
    b. **ER Diagram**: A visual representation of the entity-relationship model showing entities and their relationships.
    c.

Data modelling process

Designing a data model: An analogy

Understanding Requirements: Gathering Needs

House Blueprint: Meeting with homeowners to understand the number of rooms, style, layout preferences, and functional requirements.

Data Model: Meeting with stakeholders to understand data requirements, key entities, relationships, and the end goals of the data system.

Understand the purpose and objectives: Theory of change

- Which is the objective of my intervention?
- What is the pathway of change?

Understand data requirements: MEAL plan

- Which is the indicators that help me monitor and evaluate?
- Which the the way of calculation?
- Which the the data source for the calculation? Which is the format?
- How will I use the information

Gather requirements from stakeholders: Data flow

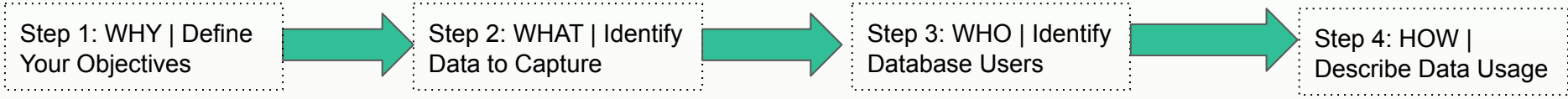
- Who collects the information?And how often?
- Who access the information?And how often?
- Who analyzes the information? And how often?

ActivityInfo

Image source:Ozimmer

Designing a data model: A checklist

Understanding Requirements: Gathering Needs



ActivityInfo

Designing a data model: An analogy

Entities: Rooms and Spaces

Entities in a data model represent the primary components or objects, similar to the rooms and spaces in a house blueprint.

House Blueprint: Rooms like the kitchen, bedroom, bathroom, and living room.

Data Model: Entities such as Beneficiaries, Relief Items, Donors, and Distribution Centers.

A discrete data object, the basic building block of your database.

Beneficiaries

It is important to identify your focal entity

ActivityInfo

Image source:HOA

Designing a data model: An analogy

Attributes: Room Features and Details

Attributes provide details about each entity, akin to the specific features and dimensions of each room in a house

House Blueprint: Room dimensions, window locations, door placements, and types of flooring.

Data Model: Attributes like beneficiary ID, name, age, gender, and location for the beneficiary entity.

The characteristic that describes your entity in some way

Beneficiaries
• Name -String data • Date of birth -Date data • Sex -text • Age-Integer

ActivityInfo

Image source:FH

Designing a data model: An analogy

Relationships: Hallways and Connections part

Relationships define how entities are related to each other, similar to how hallways, doors, and stairs connect different rooms in a house.

House Blueprint: Hallways connecting the living room to the kitchen, stairs connecting floors.

Data Model: Relationships like beneficiaries receive relief Items or donors fund distribution centers.

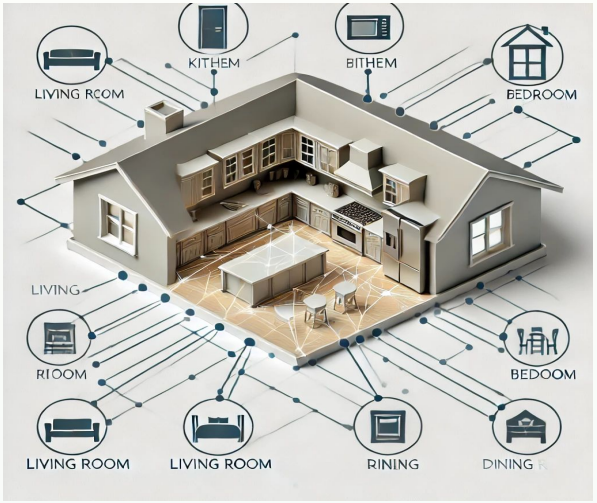
This guarantees seamless data integration across various tables and reports.

ActivityInfo

Image source:Pinterest

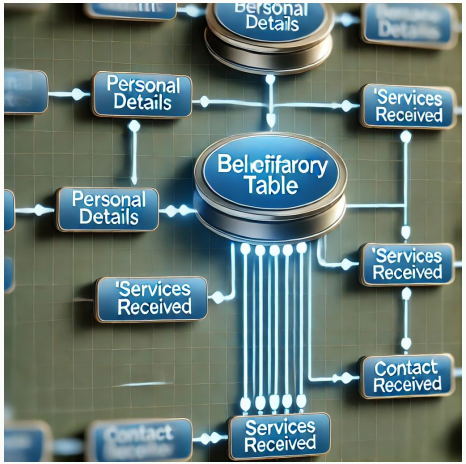
Designing a data model: An analogy

Relationships: Hallways and Connections-example



The house floor plan highlights the kitchen as the focal area, with lines connecting it to other rooms such as the living room, bedroom, and dining room. This emphasizes the central role of the kitchen in the layout.

The connections illustrate the relationships between the focal entity and other entities.



The database schema highlights the 'beneficiary' table as the central entity, with connections to related tables like 'personal details', 'services received', and 'contact information'. This illustrates the relationships between the focal entity and other entities in the database.

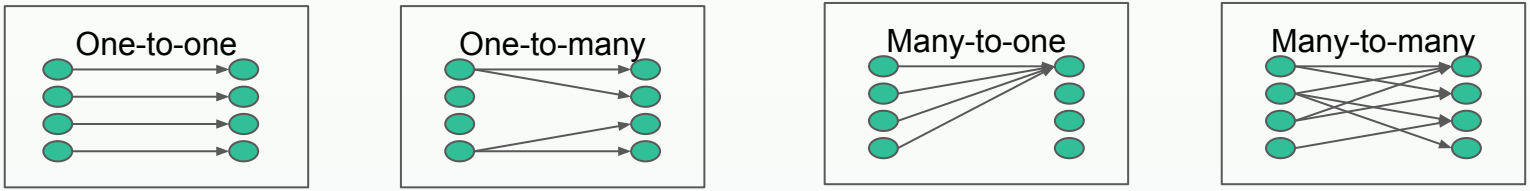
ActivityInfo

Image source:Pinterest

Designing a data model: An analogy

Relationships: Hallways and Connections

Relationships between entities are defined using cardinality (one-to-one, one-to-many, many-to-many) and referential integrity constraints (foreign keys). These relationships ensure data consistency



ActivityInfo

Designing a data model: An analogy

Normalization: Structural Integrity and Optimization

Normalization in a data model ensures data is organized efficiently and without redundancy, similar to how an architect ensures the structural integrity and optimal layout of a house.

House Blueprint: Ensuring each room serves a distinct purpose and optimizing space to avoid unnecessary duplication (e.g., one central pantry instead of multiple small storage areas).

Data Model: Organizing data into tables to avoid redundancy, ensuring each piece of data is stored only once, and linked appropriately through keys.

ActivityInfo

Image source:Frontiersin

Designing a data model: An analogy

How to effectively employ normalization

First Rule

Each attribute (column) in a table must contain only atomic (indivisible) values. This means that each cell of the table should hold a single, non-repeating value. Attributes should not contain lists, arrays, or nested structures

Name	GBV follow up dates
James	18/02
Maria	23/04,18/02, 29/10

Name	Bene ID	Session ID	GBV follow up dates
James	01	02	18/02
Maria	02	05	29/10

ActivityInfo

Designing a data model: An analogy

How to effectively employ normalization

Second Rule

All other values must be functionally dependent on the whole primary key

Name	**Bene ID**	**Session ID**	**GBV follow up dates**
James	01	02	18/02
Maria	02	05	29/10

Name	**Bene ID (primary key)**
James	01
Maria	02

Bene ID	**Session ID (primary key)**	**GBV follow up dates**
01	02	18/02
02	05	29/10

ActivityInfo

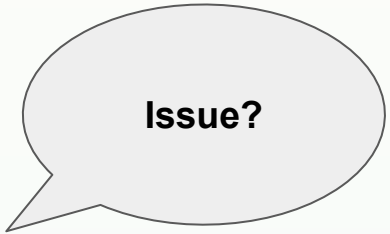
Designing a data model: An analogy

How to effectively employ normalization

Third Rule

It should not have any transitive dependencies. A transitive dependency occurs when a non-key attribute depends on another non-key attribute, rather than directly on the primary key.

Name	Bene ID	Session ID	SW ID	GBV follow up dates
James	01	02	001	18/02
James	01	04	001	27/10
Maria	02	05	003	29/10
Maria	02	06	003	23/04



ActivityInfo

Designing a data model: An analogy

How to effectively employ normalization

Third Rule

It should not have any transitive dependencies. A transitive dependency occurs when a non-key attribute depends on another non-key attribute, rather than directly on the primary key.

Name	**Bene ID** (primary key)
James	01
Maria	02

SW ID	**Bene ID** (primary key)
001	01
002	02

Bene ID	**Session ID** (primary key)	**GBV follow up dates**
01	02	18/02
02	05	29/10

ActivityInfo

Designing a data model: An analogy

Constraints and Rules: Building Codes and Regulations

Constraints and rules in a data model ensure data integrity and accuracy, just as building codes and regulations ensure the safety and compliance of a house.

House Blueprint: Adhering to building codes, zoning laws, and safety regulations.

Data Model: Implementing primary keys, foreign keys, and validation rules to maintain data integrity.

ActivityInfo

Image source:Frontiersin

Designing a data model: An analogy

Documentation: Detailed Plans and Specifications

Comprehensive documentation is essential in both processes to guide implementation and ensure clarity.

House Blueprint: Detailed architectural plans, specifications, and material lists.

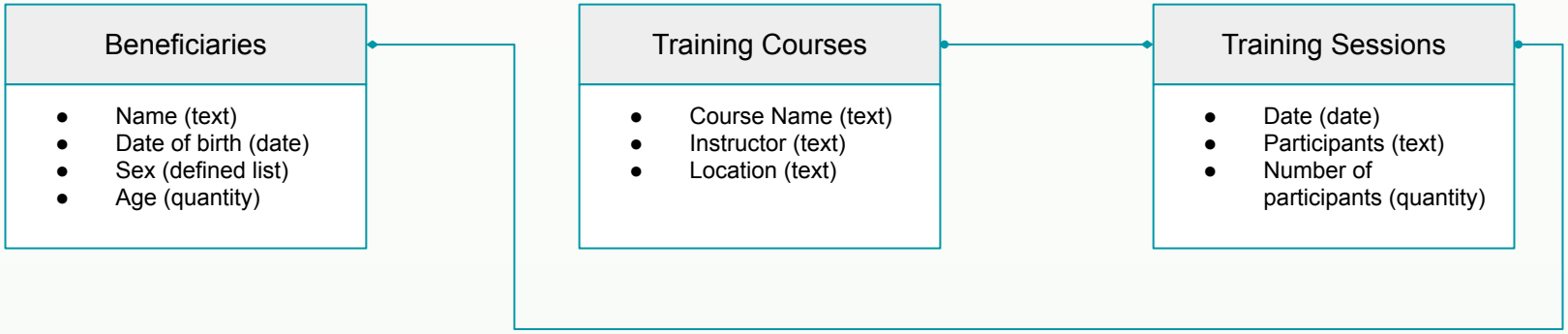
Data Model: Data dictionaries, entity-relationship diagrams, and documentation of all data attributes and relationships.

ActivityInfo

Image source:Frontiersin

Introduction to Data modelling in practice

Designing a Data model



Next Webinar

Join us in the next session

July 24

SESSION 2

Data modelling in practice

ActivityInfo

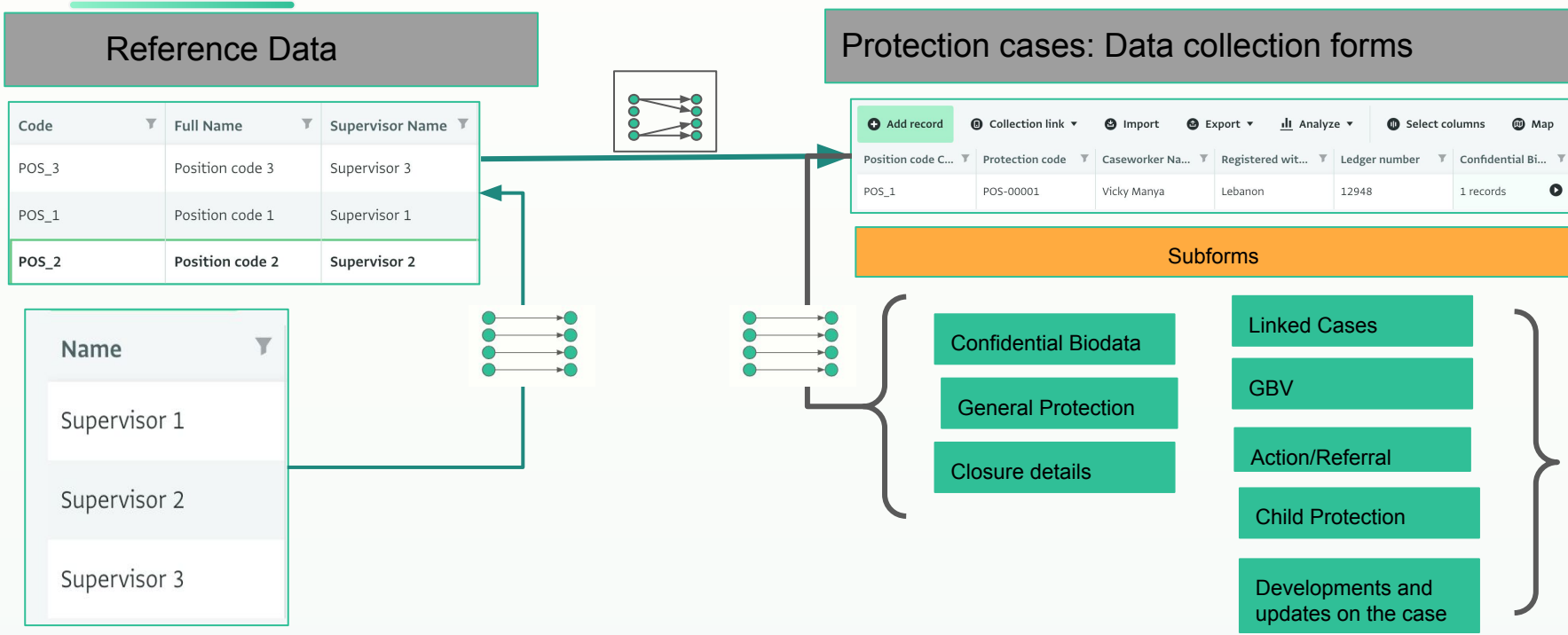
Questions?

Follow us:

LinkedIn page: https://www.linkedin.com/showcase/activityinfo/
LinkedIn group: https://www.linkedin.com/groups/5098257/
Twitter: https://twitter.com/activityinfo

Some common data models in humanitarian and development contexts

Define Relationships



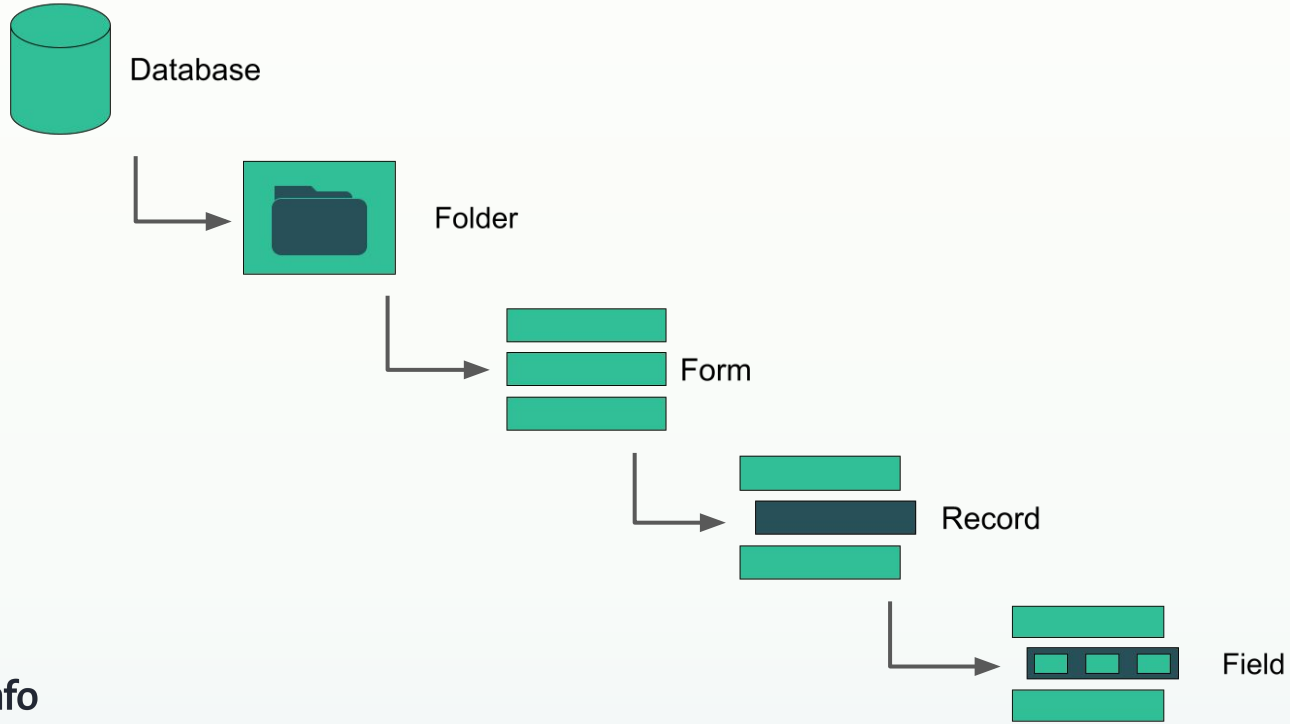
ActivityInfo

Table definition and association supports flexibility,efficient user role definition, ability to edit, export and import data

Example: ActivityInfo Case management database template for Protection

The relational nature of ActivityInfo Databases

An intuitive hierarchy for organizing data



ActivityInfo

Subforms versus Reference forms

Subform

Use when:

- you want to create a **parent-child** relationship between a two records
- related data is **dependent** on another, in other words, the subrecord would not make sense without being attached to the parent record

Reference forms

Use when:

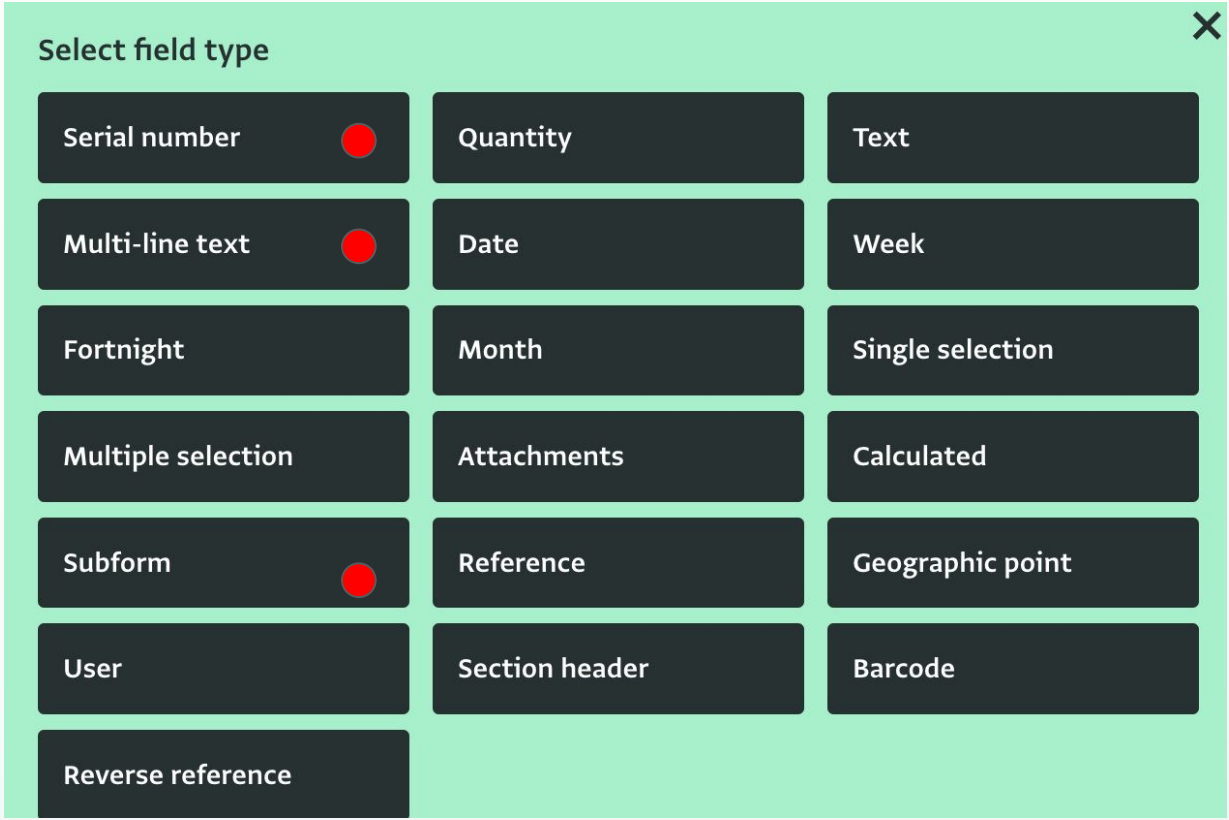
- data is known in **advance** and managed **centrally**
- data is **shared** across multiple forms
- you want to better manage a **long list** of options
- you need to make **updates** to the reference data that will be reflected in all forms using the reference
- you want to manage data to be used for **categorization** of other information e.g. geographic locations

The most common data models in humanitarian and development contexts

Relational databases are flexible

- Can handle various types of data, making them suitable for a wide range of use cases
- Improve data consistency

Used in the template



ActivityInfo

Best practices

Best Practices	Best Practices
Understand the requirements	Design the data model with performance in mind
Normalize data	Anticipate future changes and scalability requirements
Use clear and consistent naming conventions	Validate and Iterate
Document extensively	Incorporate security measures into the data model design
Enforce data integrity constraints	Provide training and knowledge sharing sessions

Designing a data model: Glossary of terms

Entity: a discrete data object, the basic building block of your database

Attribute: a characteristic that describes your entity in some way

Relationship: how entities relate to each other

Cardinality: how many on one side of the relationship relate to how many on the other side of the relationship

Key: an attribute or combination of attributes used to uniquely identify an entity

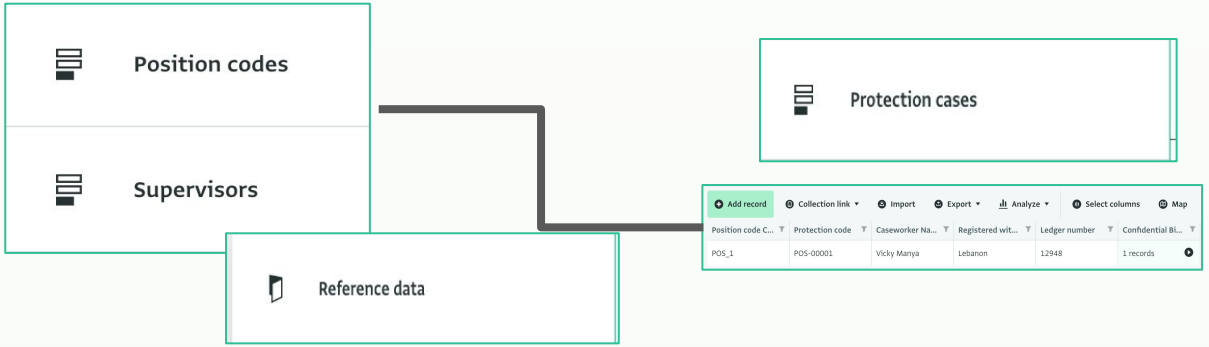
Normalization: the process of organizing your data in your database more efficiently

Some common data models in humanitarian and development contexts

Relational model:Protection, Child Protection, and Gender-Based Violence (GBV)

Organization: it's organized around individual cases or beneficiaries in a humanitarian context. It includes fields for personal details, assistance provided, location, and case status.

Rationale: In humanitarian development, this data model helps organizations and agencies manage and track assistance and support provided to individuals or communities affected by disasters, conflicts, or other crises. It enables efficient allocation of resources and ensures accountability for aid delivery.



ActivityInfo

Example: [ActivityInfo Case management database template for Protection]

Key components that make the data model efficient: Summary

Tables and Subforms	The database utilizes tables and subforms to organize and store data related to cases.It is designed to minimize data redundancy and improve data integrity through normalization.
Relationships	The database uses reference forms to link related information. Referential integrity, or the ability to maintain data consistency through referencing, is a key feature of relational databases.Primary and Foreign keys like names, serial numbers, position codes and supervisors names etc aid this relationship
Flexibility	Architecture is flexible and can handle various types of data, making them suitable for a wide range of cases. The ability to customize forms and fields aid flexibility
User Roles	This role-based access control is a common feature in relational databases to restrict access and manage permissions.
Data Entry and Editing	Users can easily add, edit, and update records in the database.
Integrity constraints	The use of relevance and validation rules as unique and check constraints, ensuring that the data is accurate and consistent.
Import and Export	Which allows us connect to other platforms either through API integration or just the use interface that allows data export and import.