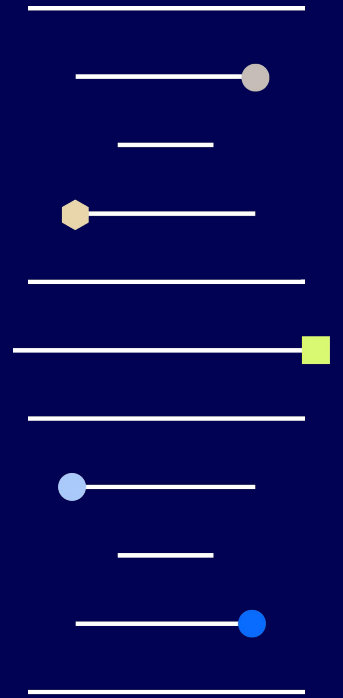
Benchling Best Practices

Data Modeling & Discovery

Benchling

Data Modeling Discovery

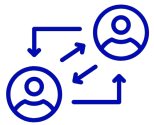


Discovery Process & Goals

Discovery is meant to uncover the important interactions, processes, and data needs of your organization. Discovery is foundational to building an effective data model. Without due diligence in discovery, your data model will not capture the intricacies of your scientific work and you will often spend extra time adding to and adjusting your data model.

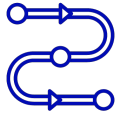
Benchling's Services teams often employs these strategies when working with your team to complete project implementations, but you can also use these questions to prompt your discovery.

Interactions



- What samples types are common between teams?
- How do you handle team handoffs?
- How are samples transferred?

Processes



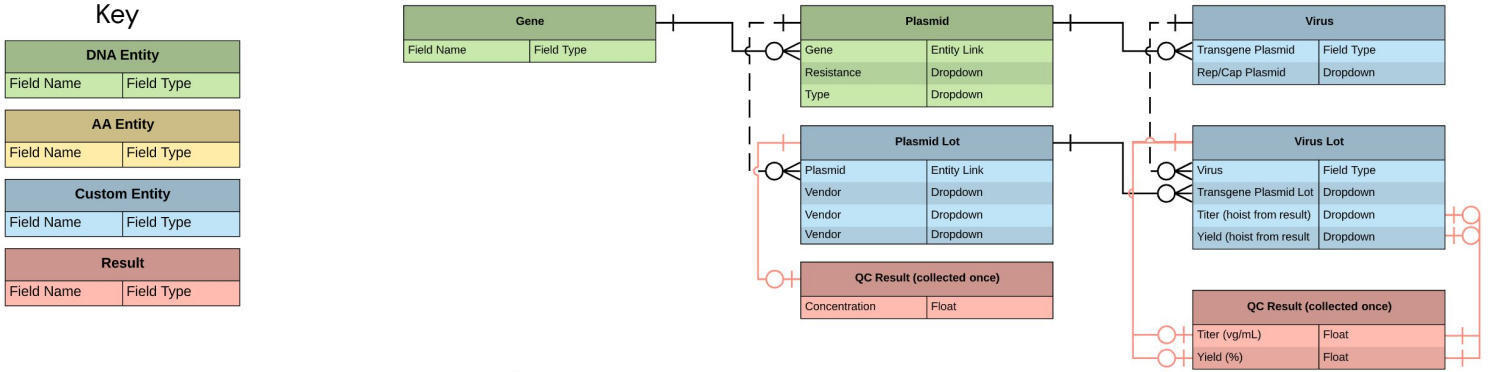
- What is produced, and what needs to be tracked?
- What data do you generate?
- What metadata is required?
- Can multiple data points be entered or just one?

Data Needs

- What types of reports need to be generated?
- What entities are needed to allow easy data capture

Discovery Tools & Purpose

Example of a LucidChart Data Model



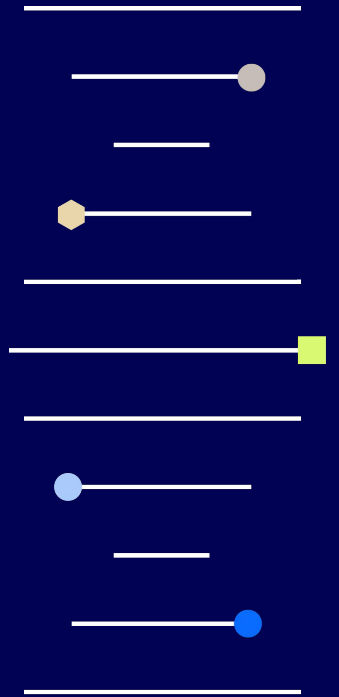
Data Modeling Tools

- Internally at Benchling we use a tool called Lucidchart for data modeling but Google Slides or Powerpoint can serve as alternatives

Having a Data Model with Alignment across all teams yields:

- Common schemas across departments
- Consistent use of controlled vocabulary
- Thoughtfully designed schemas that meet the needs of downstream consumers

Data Modeling in Benchling

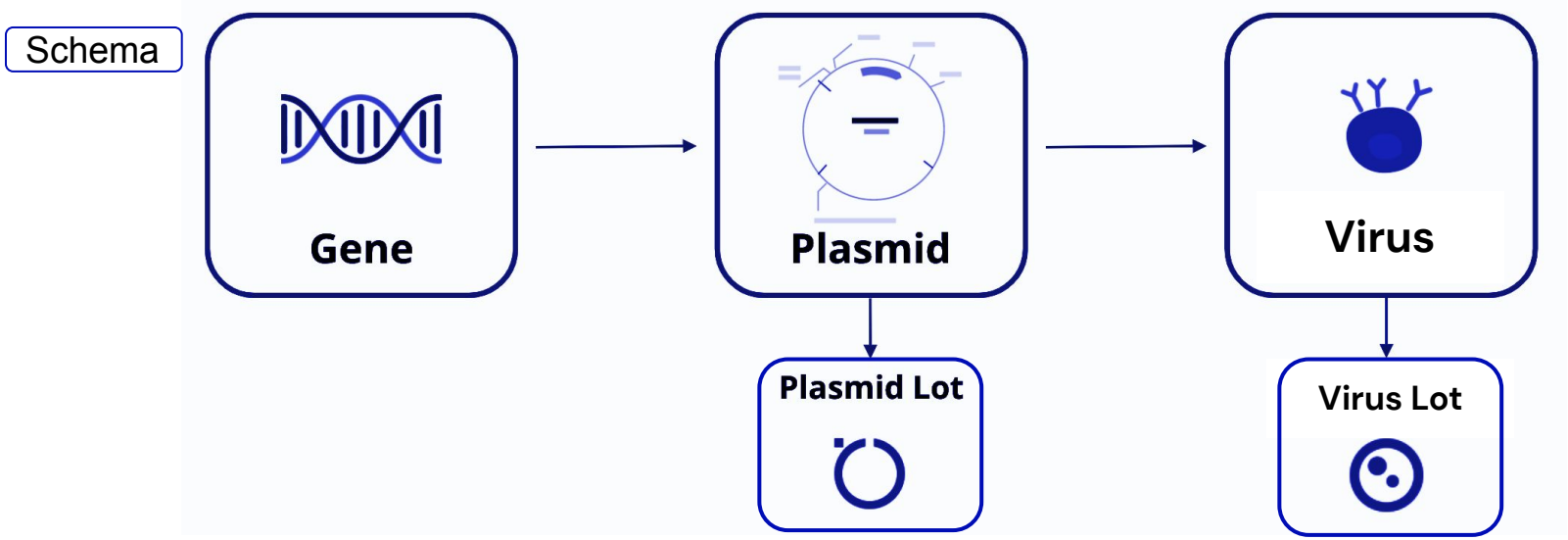


Registry Key Terms

Understand these to be able to clearly communicate about your data model

Benchling Term	Example
Schema • A unique material type or category • For each schema, a designated set of fields capture information about the material	**Antibody Schema** **Target:** Dropdown **Light Chain:** Entity Link **Heavy Chain:** Entity Link
Entity • A unique manifestation of a schema • Each entity of a schema is assigned with a unique ID • Fields represent data that are specific to each entity	**Antibody 1341** **Target:** CD19 **Light Chain:** Light Chain 3 **Heavy Chain:** Heavy Chain 1
Child Entity • Represents a “lot” or a derivative of an entity • Allows for additional fields to track information unique to that Child Entity	**Antibody 1341_001** **Parent Ab:** Antibody 1341 **Date Received:** 01/19/2021 **Supplier:** NEB

Data Modeling in Benchling



Our Schemas connect and pass data downstream. Benchling's biological awareness allows us to link sequences and other information from upstream entities to downstream entities such as genes to viruses and virus lots. In this example, each box represents a separate schema, and each arrow represents some sort of linkage that must be established.

Registry + Inventory + Results Comparison

Registry: Registry fields describe the characteristics of the entity

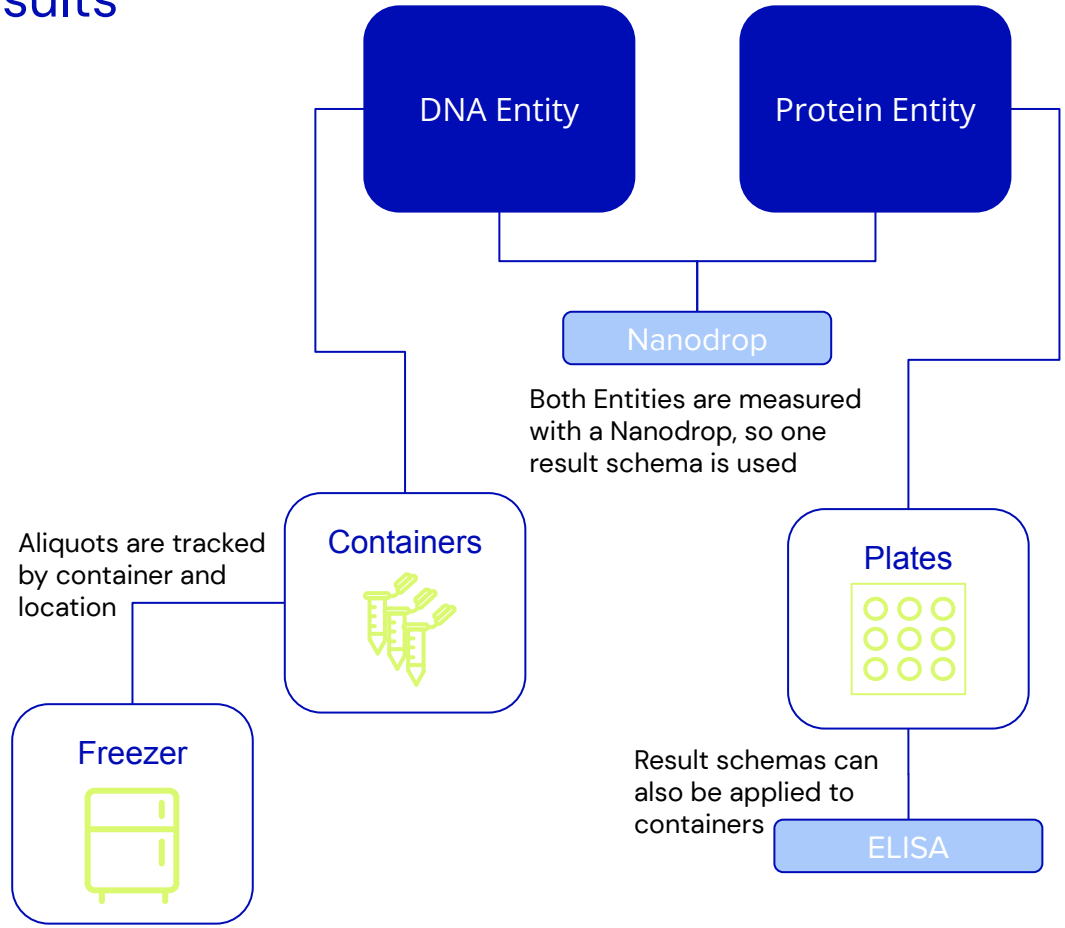
What is it? where did it come from?

Inventory: Tracks physical items in your fridges and freezers and some qualities such as concentration and amount

Where is it? How much do we have?

Results: Results fields store data produced/assay results after the creation of the item

How pure is it? What is the yield?



Moving from Spreadsheets to Data Models

Entity Name	Entity Link	Dropdown	Entity Link	Entity Link	Dropdown	Text Field	Entity Link
A	B	C	D	E	F	G	H
Plasmid Name	Backbone	Resistance	Promoter	Terminator	Cloning Type	Designer	Gene
Plasmid1	BB 123	Amp	Pro1	Ter2			GeneX
Plasmid1	BB 123	Ampicillin	Pro1	Ter2			GeneX
Plasmid1	BB 123	Amp	Promoter1	Ter2			GeneX
Plasmid2	BB 123	Kan	Pro2	Ter2			GeneX
Plasmid3	BB 123	Kan	Pro3	Ter2			GeneX
Plasmid4	BB 123	KN	Pro4	Ter2			GeneX
Plasmid5	BB 1	Ampicillin	Promoter3	Ter2			
Plasmid6	BB 1	Kan	Promoter4	Ter2			
Plasmid7	BB 123	Kan	Promoter5				
Plasmid8	BB 123	KN	Pro2				
Plasmid9	BB 123	Ampicillin					
Plasmid10	BB 123	Ampicillin					

Data you have in spreadsheets can be used to begin to develop your schemas and metadata. Each unique collection of samples maps to a schema of interest. Each unique header would map to metadata fields.

Examine the data as well; in this example, the resistance column has some recorded instances of Amp and Ampicillin. Your team might consider controlled dropdowns instead of allowing free text.

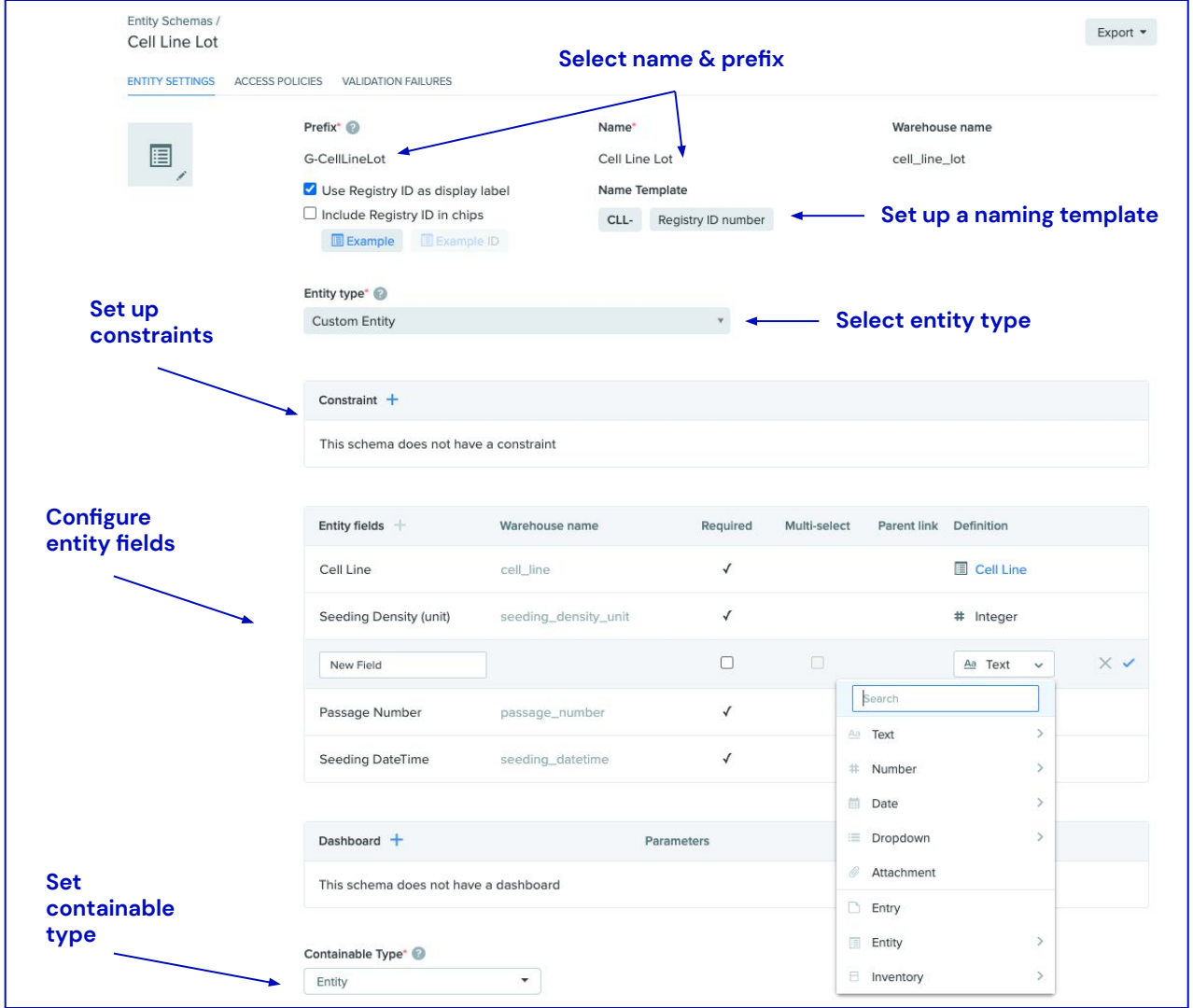
Comparing types of Entity Relationships

Establishing connections between entities is the core of structuring your data model and can give context to how your science is affected as it moves further downstream. These connections create bidirectional links between entities.

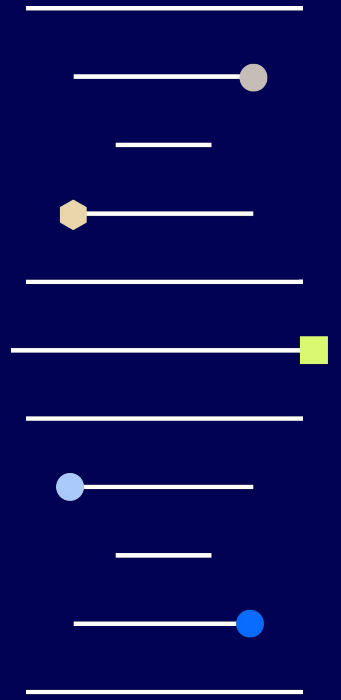
- **Entity Links:** This basic connection allows you to track link two entities and enables access to metadata across the entity link
- **Parent-Child Links:** Leverage this type of link to create derivatives of an entity, such as lots, dilutions or preps
- **Translation Links:** (DNA & RNA Schemas only) Translation links will search your DNA Sequence for compatible Amino Acid sequences registered in your Benchling
- **Part Links:** (DNA & RNA Schemas only) DNA parts allow you to create a library of short conserved sequences that will be regularly used in your lab and auto-link them to a larger construct

Data Modeling Configuration in Registry

After the data model is designed, it is configured in Benchling's Registry Settings



Data Modeling Best Practices



Best Practices for Handling Data Model Complexity

How Complex is too Complex?

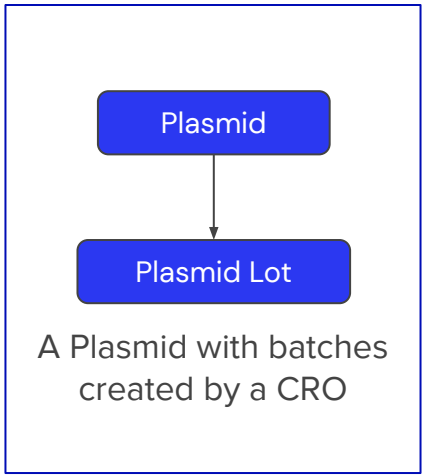
Balance the needs of your data requirements with the ease of use in the lab. Follow these best practices to ensure you don't overwhelm your scientists with documentation.

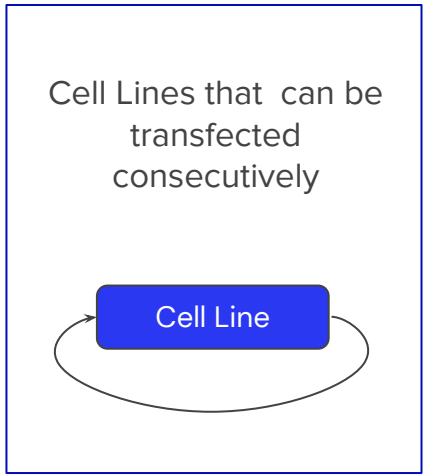
- **Limit the number of Schemas:** Condense schemas that are highly similar and use dropdowns to add descriptions of the differences.
  - **Ex:** "Biofluid" instead of "Plasma" and "Blood"
- **Consider Metadata Placement:** Registry, Inventory and Results are interconnected. Your scientific context will determine whether a field like "Concentration" is something is captured on a container, an entity, a result of an assay
- **Leverage Automated Hoisting:** Don't repeat metadata for downstream entities and make registration efficient with automation.
  - **Ex:** Use Hoisted Fields to get information about the Plasmid used to create a Protein

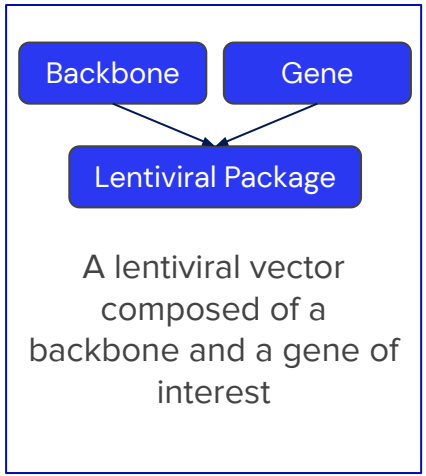
Best Practices for using Child Entities

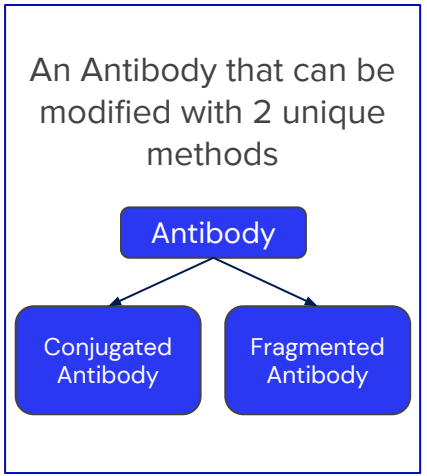
Parent Links allow you to track a child entity as a derivative of a parent entity. If you want to *repeatedly create samples* from an entity, leverage the Parent Link.

Parent Entities have all metadata regarding *what* a sample is, and a child entity will record metadata regarding its creation. How you want to structure your connections will depend on the type of relationship you want to represent:









Best Practices for Naming Strategies & Registry IDs

Naming Strategy	When to Use	Name Preview	Registry ID Preview	Alias* Preview
Generate new registry IDs	Maintain previous internal name and searability, gain additional layer of distinction with new ID	Benchling Plasmid	PL001	
Generate registry IDs based on entity names	Align Benchling identifier with existing naming convention	PLASMID322	PL322	
Generate new registry IDs and replace name with registry ID	Override previous naming convention and entirely adopt Benchling's new naming system	PLASMID001	PLASMID001	
Generate new registry IDs, rename according to name template, and keep old name as alias	Adopt a naming template, manintan searchability of previous entity name, and gain additional layer of distinction with new ID	PL-resistance-001	PL001	PL322
Generate new registry IDs, and replace name according to name template	Adopt a naming template with additional layer of distinction with new ID	PL-resistance-001	PL001	
Keep existing entity names as registry IDs	Override Benchling's identifier and default to internal naming system	PL322	PL322	

Registry IDs are unique Benchling identifiers of distinct entities that help ensure data integrity within your organization. When building a schema, you will need to select a naming convention option. Consider which one suits your needs and **turn OFF all the others to maintain consistency**.

**Aliases are searchable and @ mentionable fields in Benchling that allow you adopt a new naming system without disrupting previous naming conventions*

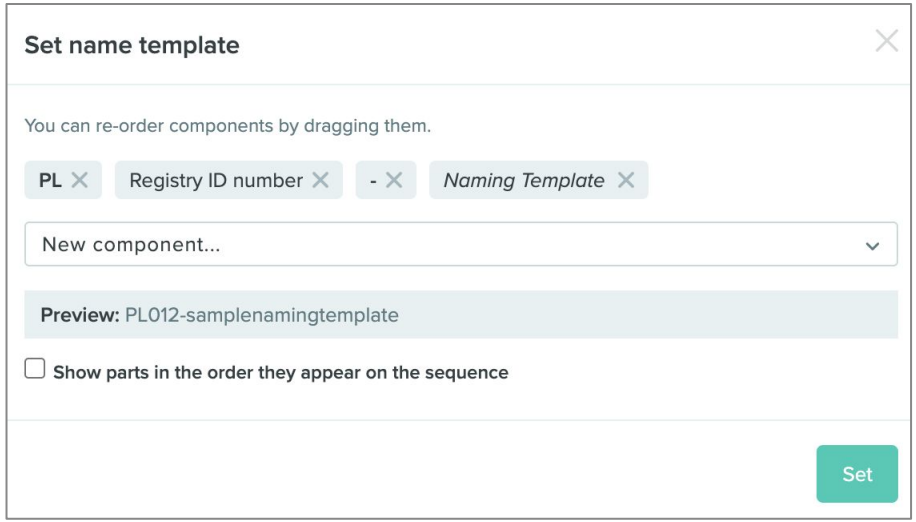
Best Practices for Naming Templates

If you select a naming strategy that uses Naming Templates, take advantage of elements such as:

- Parent entity & metadata hoisting
- Incrementing numeric IDs
- Parent Lot IDs (only for schemas with parent links)
- Fixed Text

Carefully consider what information is vital for your lab members to search and identify entities and set up a naming template on your schema.

In addition to a name, your entities will have a Registry ID and an optional Alias. All three components are searchable in Benchling.



To learn more about naming conventions, access this Help Article!

Benefits & Tradeoffs of Schema Design Strategies

Should you align on schemas and fields across your entire organization?

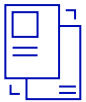
	Required Fields	Org Shared Schemas	Team-Specific Schemas
Benefits	+ Standardize the inputs of your registry schema + Define what "complete" metadata is + Commit to data quality up front	+ Lower admin lift to develop and maintain + Less confusion for users with fewer schemas to Choose + Cleaner UI Experience	+ Schemas designed to fit to one team's needs + Filter by schema to view team-specific metadata + Can utilize schema-specific permissions + Descriptive schema names
Trade-offs	- Too many required fields can make registration difficult, particularly if data is gathered at different times or if some fields only pertain to certain teams	- Team Specific fields are shown to everyone - Filtering is more complicated to identify specific teams' entities - Aggregated datasets may have missing metadata values	- Increase in number of schemas increases Admin Lift to maintain - Long list of schemas to choose from for users - Downstream entities may need to be developed for each team's outputs, increasing complexity - Possible redundancy or entity duplication if different teams are working with and registering the same samples

Resources for Registry & Data Modeling

Registry Configuration Product Guide

Import Legacy Data in Bulk

Managing Registry schema permissions



Schema basics