HR Employee 360

Table of Contents

[Intro 1](#_Toc66956292)

[Executive Summary 2](#_Toc66956293)

[Proof 2](#_Toc66956294)

[1. Connect - Model Business Entities and Relationships 3](#_Toc66956295)

[2. Integrate – Load As-Is 9](#_Toc66956296)

[3. Explore Immediately 11](#_Toc66956297)

[4. Integrate - Mapping 12](#_Toc66956298)

[5. Explore 14](#_Toc66956299)

[6. Integrate - Mastering 20](#_Toc66956300)

[7. More on Knowledge Graphs 27](#_Toc66956301)

[8. Export 30](#_Toc66956302)

[9. Query Console - SQL Views 30](#_Toc66956303)

[Conclusion 31](#_Toc66956304)

[APPENDIX 32](#_Toc66956305)

# Intro

You’ve seen the corporate overview, and know that MarkLogic is Data Integration. Simplified. But in this walkthrough we’re going to show you what this looks like in action with real data.

This walkthrough uses synthetic data created based on our experiences in the field. It’s simple enough to help convey the concepts. We know it’s always more meaningful to people when they see this in action with their own data, and now with this cookbook data hub example, you have a getting started kit for integrating your own data. After going through the walkthrough we hope you’re empowered to quickly edit the model and steps accordingly to use this data hub with your own data.

Our walkthrough is presented in the terms of a common use case, presented with fictional companies and data.

**Use the Data Hub Central Community Edition with MarkLogic Server for this walkthrough.**

# Executive Summary

MountainTop is a large US property and casualty insurance provider that is growing rapidly through acquisition. They have recently acquired Coastal insurance. To this point both companies have operated as independent business entities that used distinct applications and data repositories.

Mountaintop is expanding into new markets and to staff projects for new business initiatives they need to be able to staff projects with their existing employee base given their prior in-house experience. They want to have a clear, integrated view of their organization to meet this need as well as to reduce the time and cost of future acquisitions and to establish a clear work from home policy.

Mountaintop is struggling currently with the staffing process as the data pertaining to employees’ current role, performance appraisals, career goals etc. is housed and managed by different applications (like Workday, SAP, and Oracle) that operate in silos. As a result, the staffing team members spend a significant amount of effort in matching the new job requirements with the available candidate pool. The CFO is also upset as the time to integrate the silos is impeding them from making further acquisitions.

##### HR Business Needs

In order to make the staffing process efficient and help achieve its objective, Mountaintop wants to build rich, unified view of employees from both the subsidiaries that will make it incredibly easy for staffing team to:

1. Perform complex searches across the full set of employees in real-time — something that wasn’t possible before (e.g. skills, work experience, performance rating etc.)
2. Get automated recommendations when a candidate matches a job requisition
3. Support HR analysts, who want a live dashboard that depicts departments (and locations) of employees
4. Ensure that company maintains tight security controls on this sensitive employee data

##### HR IT challenge

What MountainTop needs to build is an employee 360 application based on HR Data Hub that integrates employee data from MountainTop and its acquisitions.

We can easily achieve this goal using MarkLogic Data Hub Platform. And that’s not all, with all the data in HR Data Hub, we’d also be able to overlay a skills-based graph that matches job requisitions to most suitable employee profiles.

# Proof

You’ve likely seen Load, Curate, Deliver on slides before in MarkLogic overviews and MarkLogic World presentations. A goal of this walkthrough is to show you what this looks like in action.

We’re going to integrate data from two different insurance companies and display that data for our HR rep in a unified view.

For this walkthrough, we’re going to take you through the MarkLogic Data Hub process which allows you to:

1. Load your data as-is
2. Curate the data into a unified view
   1. We will model the key business entities and relationships we want to be able to query across
   2. We will map our data sources to our model
   3. We will harmonize the data into a consistent and unified view
   4. We will master Employee entities to disambiguate and merge similar Employees
3. Deliver the results of the unified view to the call center rep
   1. As well as to different consumers through different lenses on a single, integrated set of information in the data hub

It turns out, Load and Curate can and often do occur in parallel. This is the flexibility MarkLogic provides. And when getting started, we start the process by defining the business problem we want to address first before actually loading any data.

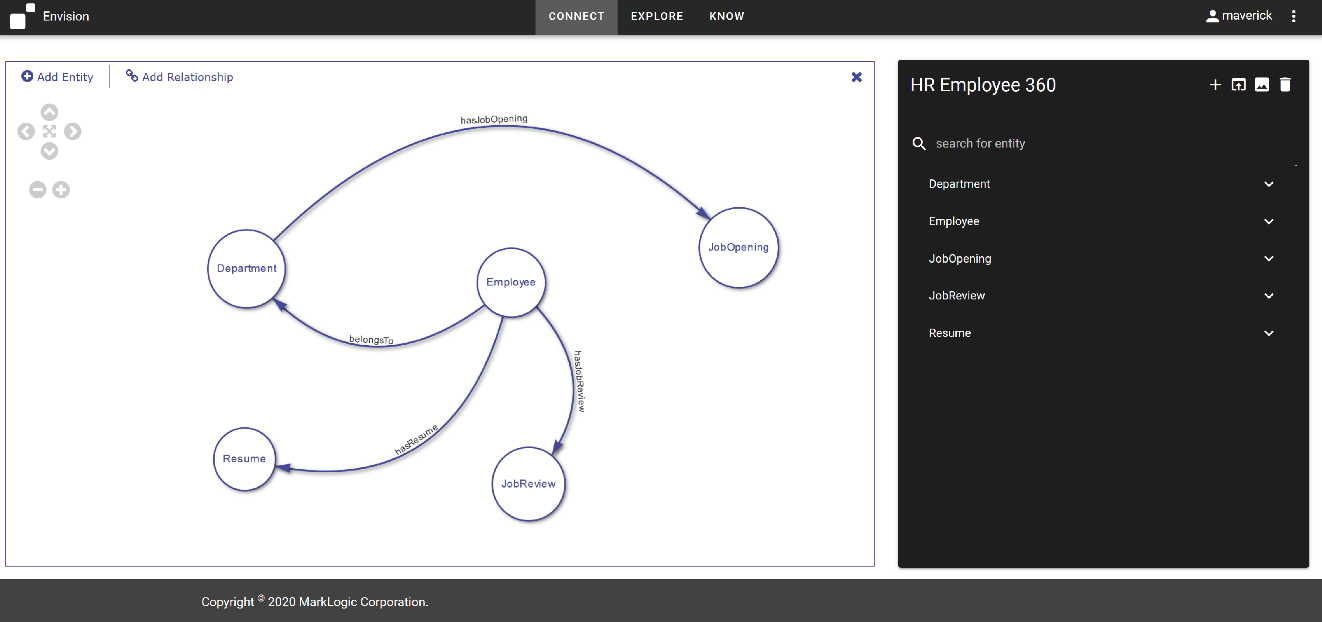
# 1. Connect - Model Business Entities and Relationships

You have the data. It’s in silos. We know it’s all over the place in your organization. At MarkLogic we deal with this all the time. The real questions we need to start with are:

* How does your business want to capture insight into your data? Another way of answering this question is to ask:
  + What key business concepts, expressed as entities and relationships, do we want to be able to connect in the data hub that will answer the questions and solve the problems our business is currently facing?

We make our beginning by modeling a solution to our problem using our Connect application.

* Action: Login
* Action: Load the HR Employee 360 Model

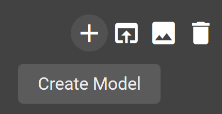


MarkLogic allows us to create a data model, decoupled from the underlying sources, and that’s what we’re going to do here. The conceptual model we create will be persisted in the database and drive our requirements for data integration and delivery.

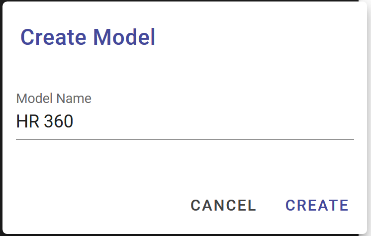
It’s an incremental approach to data integration. So we start with the minimal set of entities and attributes we want to connect to create our model. We can always come back and add more to the model and more data to our integration. That’s the beauty of it.

By default, we’ve pre-loaded a model for use based on the familiar use case of HR360 we’ve seen over the years. You may create additional models as well. The first available model is loaded for display. We’re going to come back to this. But for now, let’s start with a blank canvas so we can see what creating a model looks like.

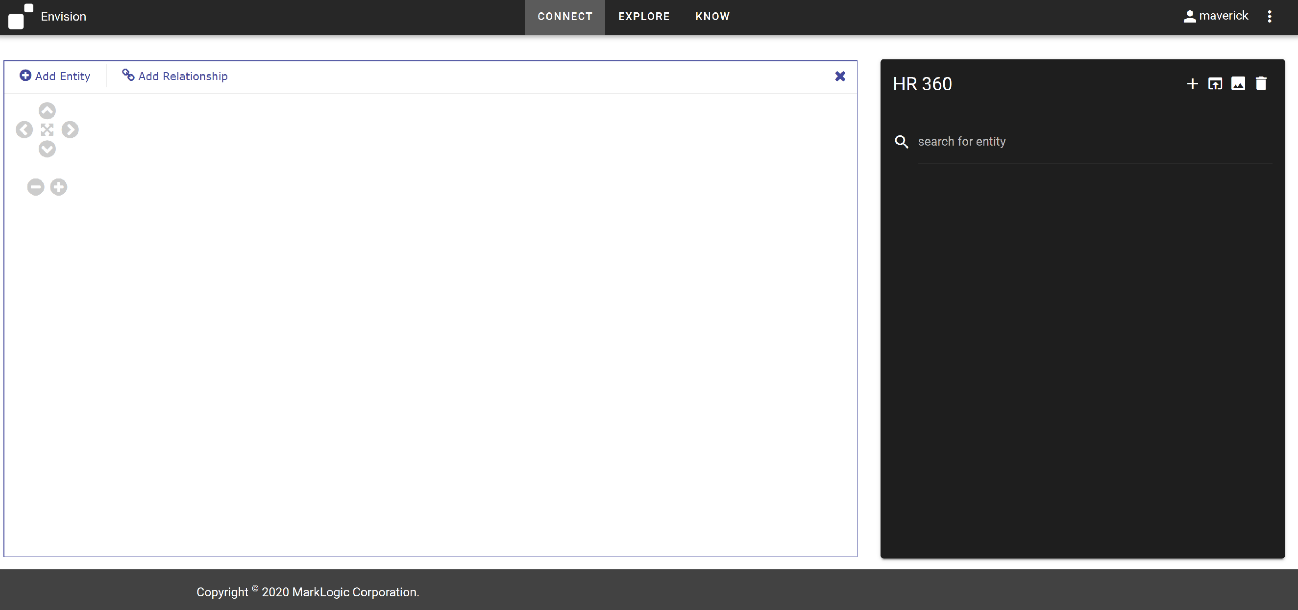
* Action: Click the ‘+’ sign in the upper right hand corner



* Action: Enter ‘My HR 360’ as the Model Name



* Action: Click CREATE

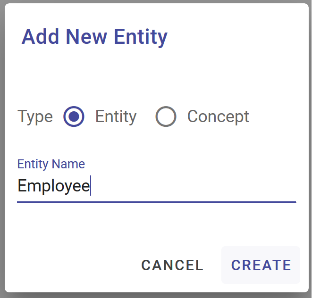


This is our canvas. Everyone understands a whiteboard or a back of the napkin diagram. In this application we simply add circles for entities and drag lines between entities for relationships to define our model for the data. In this way, we create our model. This helps us to understand our business while capturing our requirements for how we wish to express and work with our data in a unified view.

In the upper right we see the name of the model we’re working on: ‘My HR 360’.

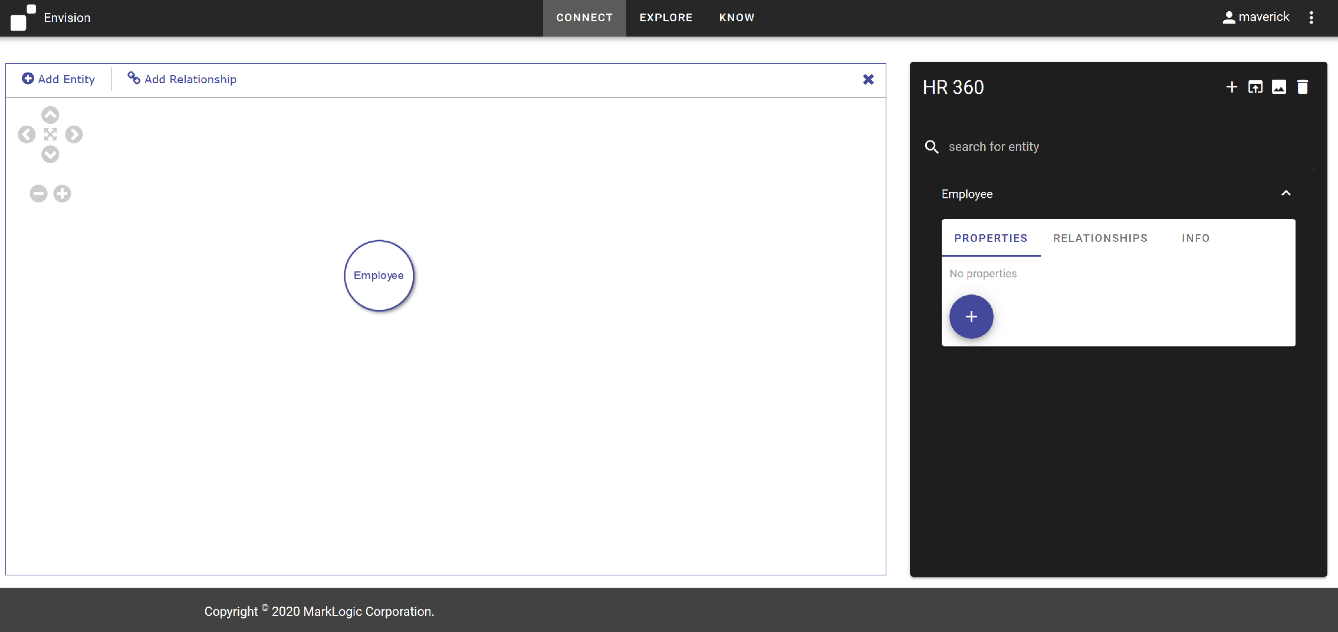
Now, we’ve been told for this HR 360 solution we have Employees and those employees have PerformanceReviews and Resumes. Employees report to a Department, and Departments have JobOpenings. So let’s go ahead and start to create those entities and relationships.

* Action: Click ‘Add Entity’ in the upper left of the graph canvas
* Action: Click somewhere on the graph canvas



We simply add the name of our new entity when prompted.

* Action: Enter Entity Name: Employee
* Action: Click ‘Create’ to create the Employee entity



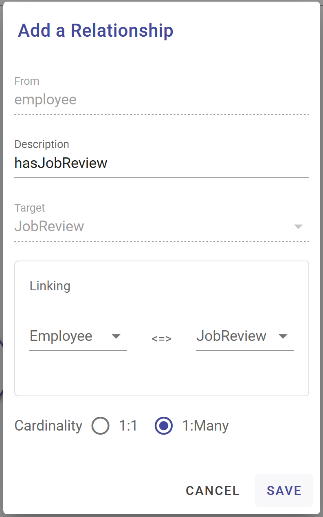
We now see our Employee entity has been added to the canvas. On the right we also see our entity listed along with the ability to define the properties we care about capturing in our unified view for this particular entity. We’ll come back to that later.

Let’s add our JobReview entity similar to how we did our Employee.

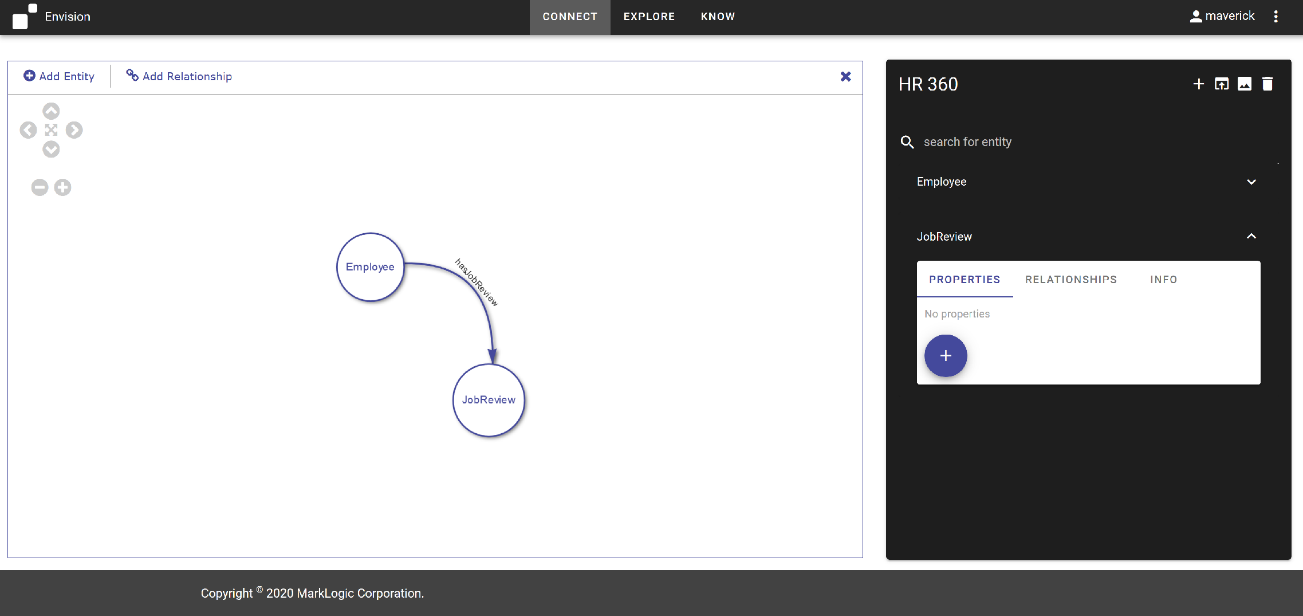
* Action: Repeat steps above taken for ‘Employee entity to add ‘JobReview’ entity.

Next, you’ve informed us that Employees ‘have’ JobReviews we’re interested in connecting, so let’s just add that relationship.

* Action: Click ‘Add Relationship’
* Action: Mouse down on ‘Employee entity
* Action: Drag Relationship from ‘Employee’ entity to ‘JobReview’ entity
* Action: Release mouse button on ‘JobReview’ entity



* Action: Enter ‘hasJobReview’ in the Description field to name the relationship
* Action: Select 1:Many
* Action: Click ‘Save’ to save the relationship



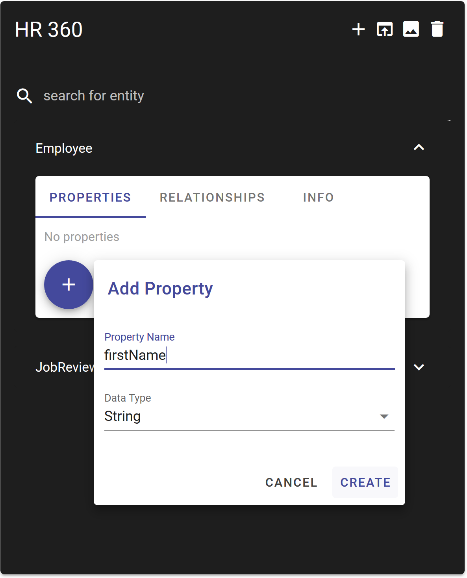
Creating our model can be as simple as drawing circles and lines. We simply label them with nouns and verbs.

* *Optional Actions:* 
  + *Add ‘Department’ entity*
  + *Add ‘belongsTo’ relationship between ‘Employee’ and ‘Department’*
  + *Add ‘Resume’ entity*
  + *Add ‘hasResume’ relationship between ‘Employee’ and ‘Resume’*
  + *Add ‘JobOpening’ entity*
  + *Add ‘hasJobOpening’ relationships between ‘Department’ and ‘JobOpening’*

For our Employee, we can also define the properties we want to be able to query in the unified view of our data.

In MarkLogic, we model what we need, as we need it. Connect allows us to quickly capture the key attributes we need to answer the immediate business problem. We can always come back and append and enrich our model later.

* Action: Click the’+’ button on the Properties tab for the ‘Employee’ entity.



* Action: Enter ‘firstName’ for the property Name
* Action: Click ‘CREATE’

We see firstName added to the Properties definitions for our Employee entity.

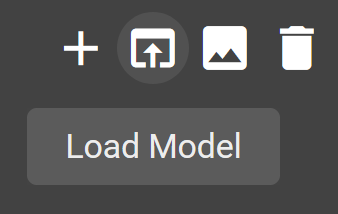
We can of course add other things like lastName, and id, and maybe there’s some key attribute like hireDate that’s a Date type. We can select the data type while defining the properties as either Boolean, String, Integer, Decimal, Date, or even an Array list.

* Action: Enter ‘employeeId for a property name.
* Action: Display drop down of available types

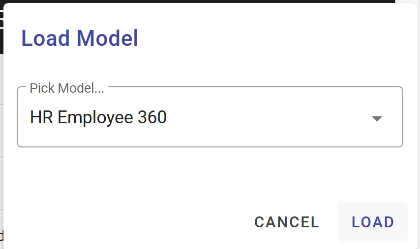
As we’re doing this, our model is being saved to the MarkLogic Data Hub using Entity Services. MarkLogic provides a model driven approach to data integration. But these models aren’t like relational models set in concrete. They inform the shape the shape and characteristics of our data like security, while allowing the data that doesn’t fit the model to still exist as-is in the system and remain available for search and query and future curation.

But we’re not going to make you sit through creating everything here. We’ve already started a model for us. So we’ll just open that up and continue from there.

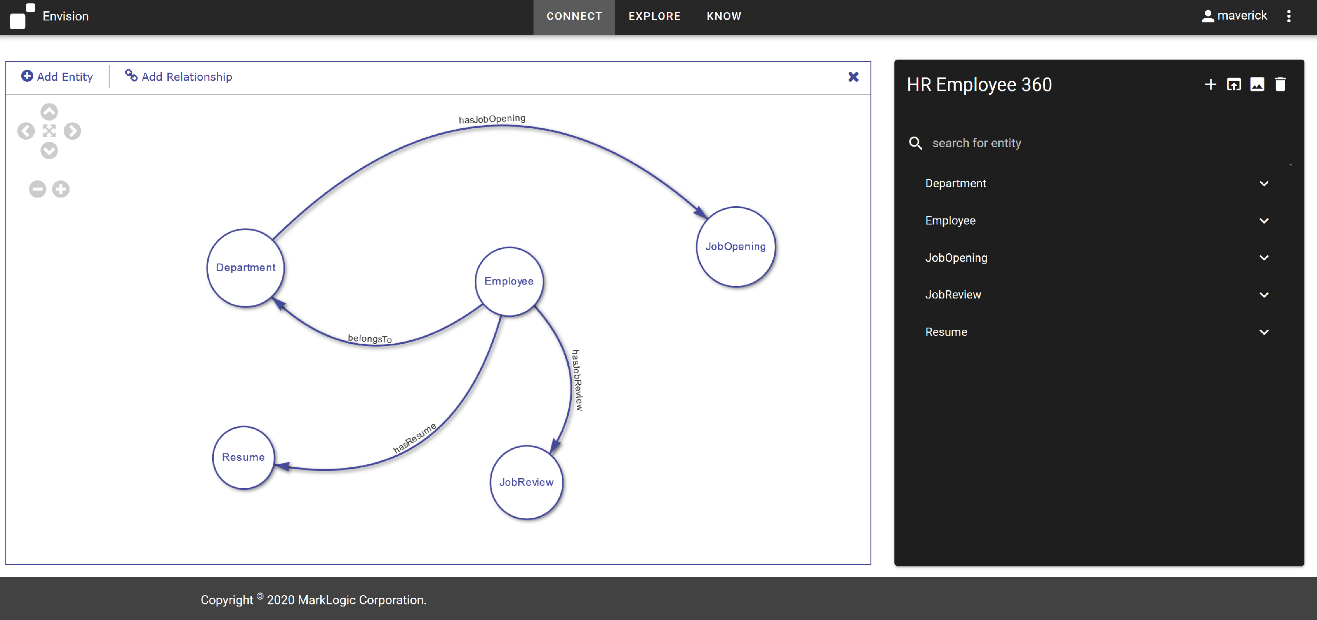
* Action: Click the ‘load model’ icon in the upper right hand corner



* Action: Select ‘HR Employee 360’ from the ‘Pick a Model’ dropdown menu.



* Action: Click ‘LOAD’



When you look at the business problem on the canvas, it appears to be really simple. We’ve got Employees. They have Resumes and JobReviews. Employees belong to Departments which have JobOpenings. With a handful of attributes from each, we can provide the HR Analyst a single pane of glass to better serve Employees and meet our business objective of aligning resources to the work required by the business.

The real challenge is the fact that we have multiple silos of data for each of these entities. But MarkLogic allows us to simply map our sources to the model to create our unified view of data and we’ll look at that next.

And that’s it, we now have our model for how we wish to express our data to meet our HR Employee 360 business requirement. You can click around the entities and relationships to inspect their properties.

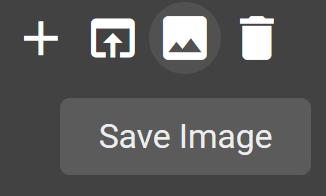
* Action: Click different entities. Display and examine some of the properties.

When we loaded our saved model, it was actually persisted in the data hub as MarkLogic Entity Services. Again, Entity Services provides the backbone for how we drive data integration in MarkLogic.

And we can always come back and update a model with new attributes and relationships and even more entities. What’s really interesting is since our model is decoupled from the actual sources and we can connect data using it incrementally, we can actually be aspirational in our modeling.

*OPTIONAL: We’ve seen this type of whiteboard modeling be very useful for bridging the gap between business and IT. Let’s say Business wants to add a Supplier entity to the model with a relationship to the Department (Supplier providesSuppliesTo Department). A stakeholder in the room may correctly point out: “We don’t have that data”. And you may not, yet, but you’ve captured something important to the business. A business stakeholder may inform us that they have a plan to acquire the data because if they better understood which Suppliers were supplying which Departments they’d be better equipped at negotiation time to leverage their insights to get the company better Supplier rates in the coming year. With this simple discussion and addition to the model, IT now has some insight into what types of data Business wants import in the future and why and now can prepare for it and look for opportunities to supply it and populate it in the data hub in the future. There’s no harm in capturing additional requirements in the model now.*

*OPTIONAL: After we create our model, we can export an image of the model to share with our colleagues as well.*

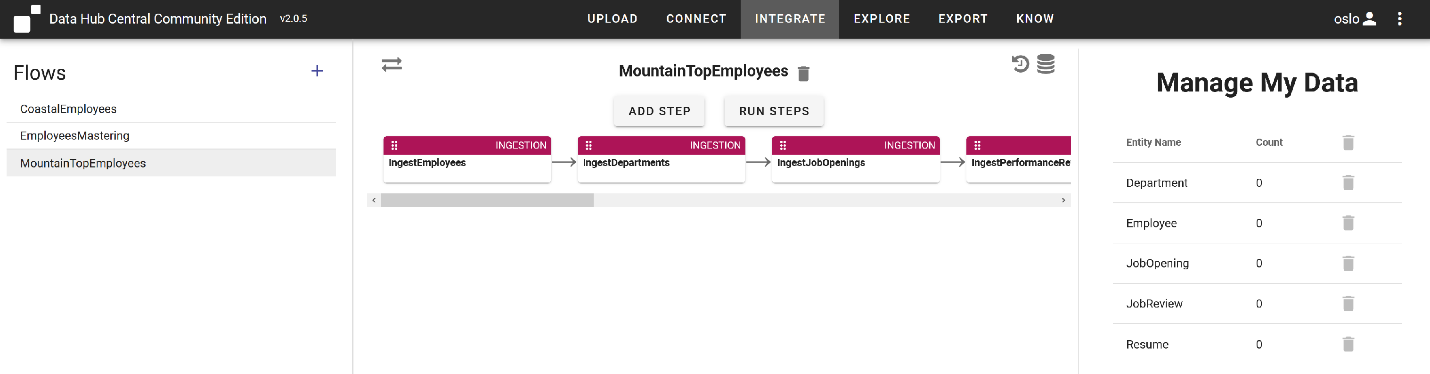


The next step is to map sources to our model. And for that we’ll go to our mapping and mastering component in the integrate tab.

# 2. Integrate – Load As-Is

In the MarkLogic Data Hub we create flows for ingesting and curating data. A flow can contain multiple steps and these steps can be for ingest, mapping, mastering, and custom operations on our data.

What you see here are 3 flows that we’ve created for 2 different companies we’re integrating as part of our HR Employee 360 solution.



Let’s take a closer look at our MountainTopEmployees flow.

For our MountainTopEmployees, we’ve defined a single flow with 5 ingestion steps and 5 mapping steps.

*Note: Data Hub Community Edition does not allow you to create Ingestion steps. You can create them using QuickStart though and DHCCE can then use them.*

*Note: DHCCE provides an Upload tab, which allows you to load your data through drag and drop. Your flows can then start with a mapping step and don’t require an ingest step.*

The ingestion steps are where we load as-is. So let’s take a closer look at these.

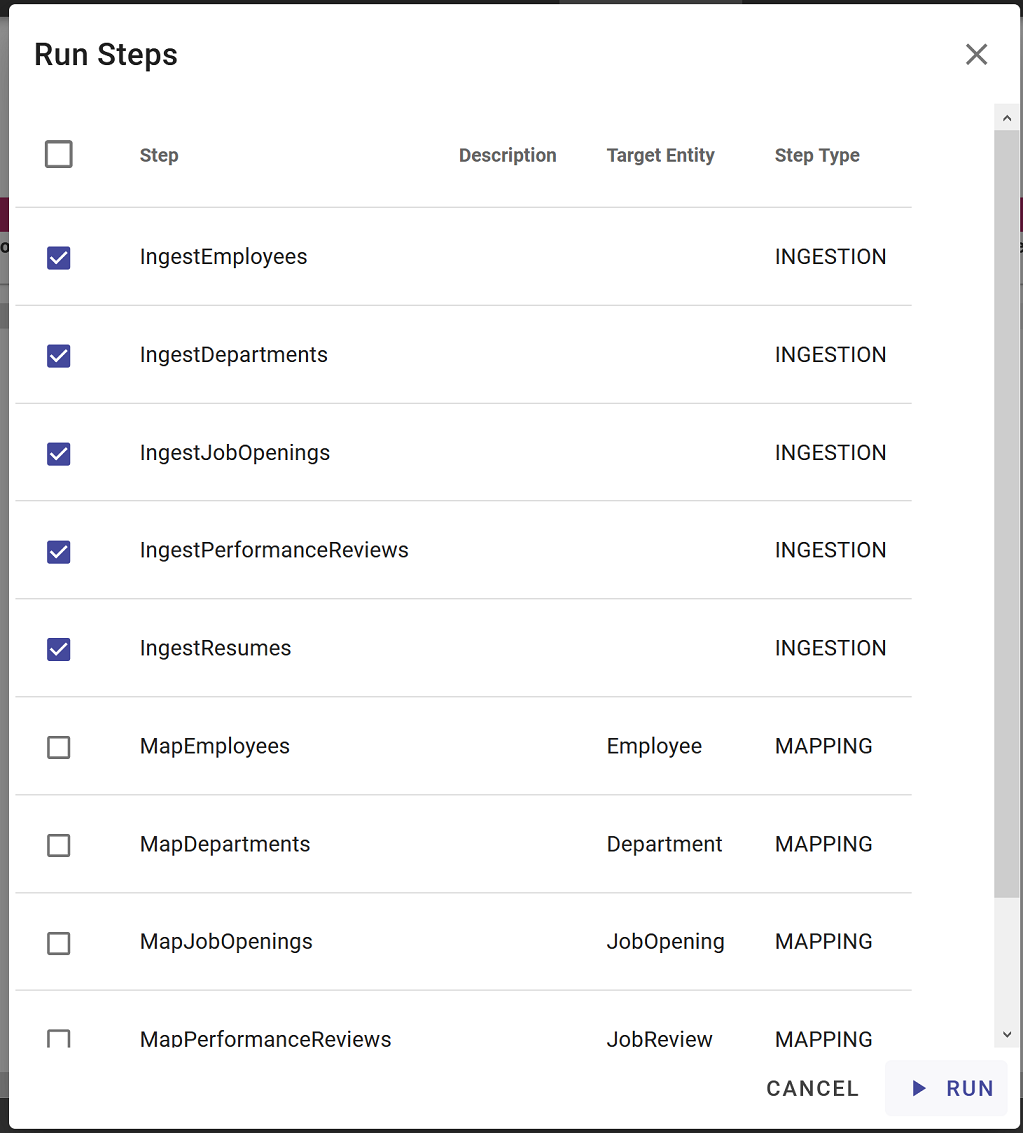
There are five here, one for each source of Employee, Department, JobOpening, PerformanceReview and Resume data.

Let’s open the Employee one by clicking on it and take a look. Here we see the location of our data to be loaded. Remember to update this accordingly if your data is in a different location.

Feel free to inspect the data and you’ll see for MountainTop the majority of these are .csv files.

We load the raw data by running the ingestion steps and we have our data in our hub.

* Action: With MountainTopEmployees flow selected, Click Run Steps
* Action: Select the checkboxes for the ingest steps
* Action: Click Run to run the ingest steps for MountainTopEmployees



# 3. Explore Immediately

The next step is to map those raw data sources we just loaded to our model. For that, we create mapping steps. But we don’t have to do any mapping to start getting insights into our data!

When we load data into MarkLogic, all text and structure was indexed for us. We can immediately start to explore this data from the Explore tab.

The Explore tab defaults to the Final database. In a data hub, data is first loaded to a staging database for curation. After harmonization, data will be available in the final database.

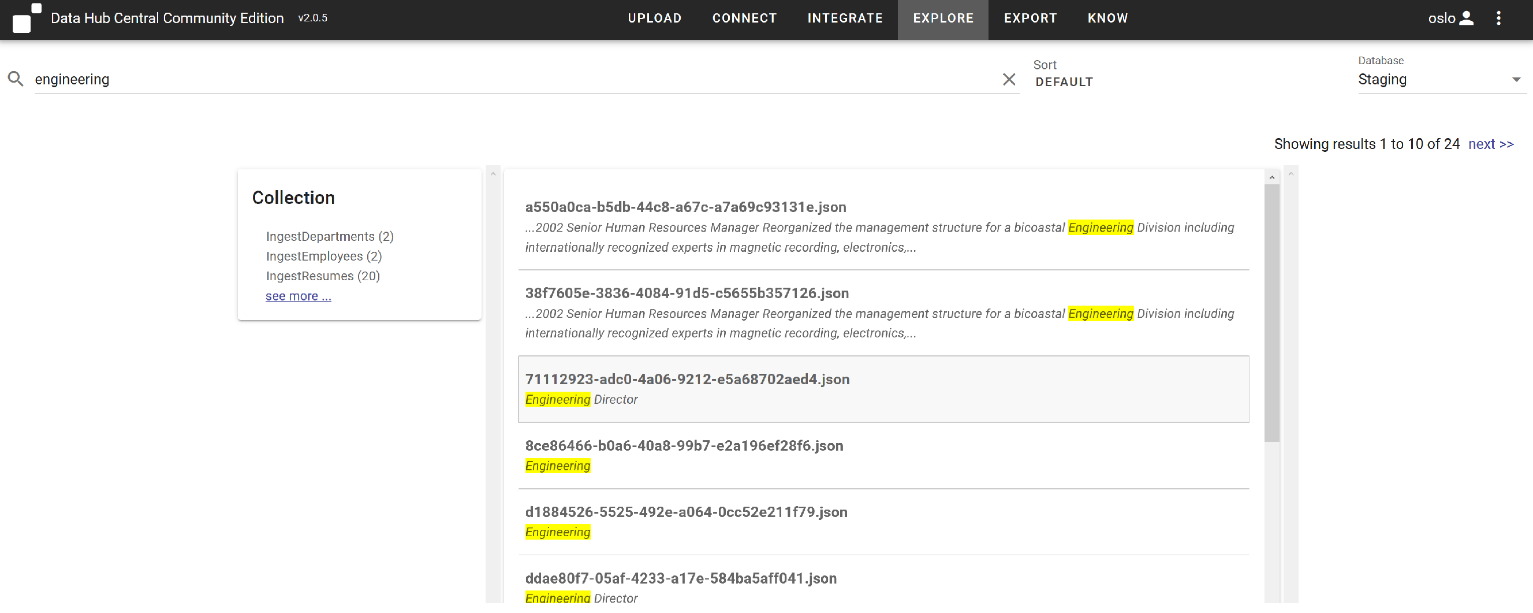
* Action: In the upper right select “Staging” from the Database dropdown menu

On the left we see the collection names for the sets of data we’ve ingested for our flow along with counts for how many records of each have been loaded. The default collection names come from the names of our ingestion steps.

On the right we see a list of the records, identified by their identifier in MarkLogic, along with a snippet of text and some metadata for each.

Now we can start to do some discovery and look at what we have in the hub so far that has to do with engineering.

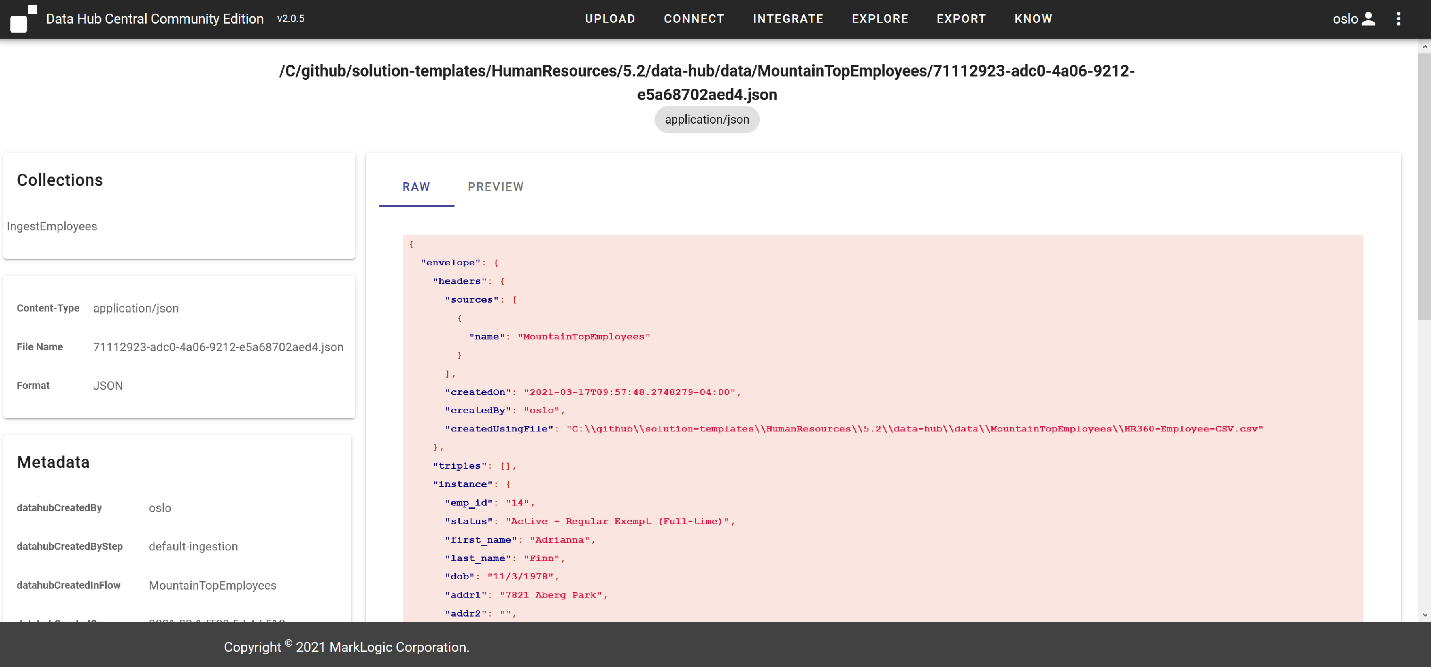
* Action: Search on ‘engineering’



This can be very useful for validation, and getting answers immediately before we’ve mapped anything to the entities we’ve defined. Here we see we have Department, Employee, and Resume data all that has ‘engineering’ somewhere in their attribute values.

If we click on the record identifier, we can even explore the shape of the data, which can be useful for informing how we map and model data.

* Action: Click on the record displaying text snippet “Engineering Director”



On the right, we see a raw document preview for one of the Employee records we’ve loaded. This is how MarkLogic stores the document internally. To the left of the document we see all metadata associated with this record. At the top, the gobbledygook string ending in .json is the name of the record in MarkLogic.

So with MarkLogic’s powerful indexing, we can start to search and query our data as is, but we know we’ll want a common vocabulary across these unstructured , disparate sources to enable structured queries and search so let’s go clean this raw data up by mapping it to our Connect model.

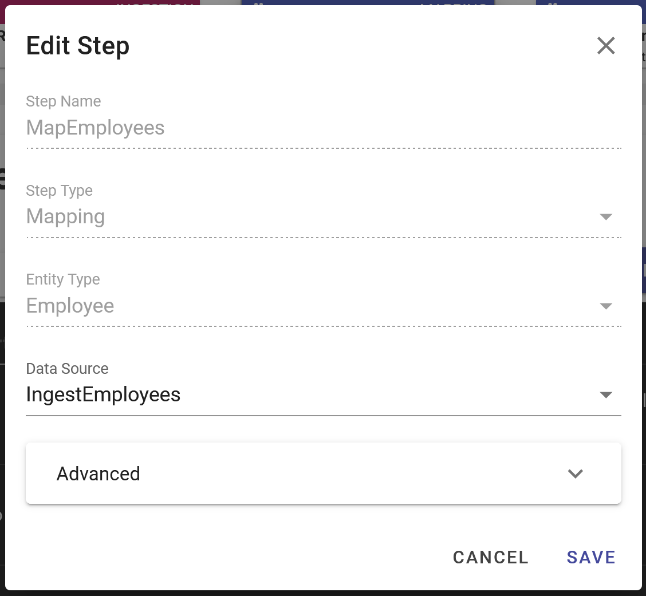
# 4. Integrate - Mapping

When we create a mapping step, we get to choose our source data and the entity we want to map it to.

When we loaded our data, it was actually loaded into collections named after the ingestion step. So we just define our Source as the IngestEmployees collection and choose Employee as our target entity.

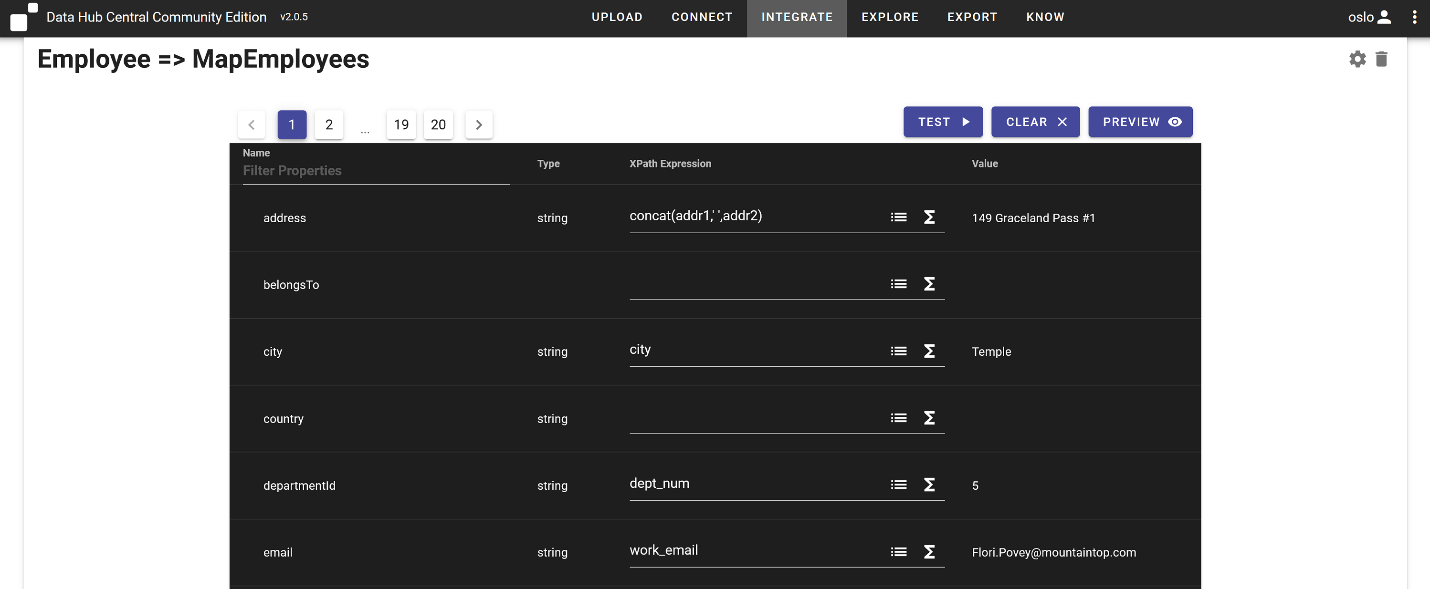
You can add mapping steps by clicking Add Steps and selecting the mapping option. For us though, let’s examine what we’ve already created.

* Action: Select the MountainTopEmployees flow
* Action: Select the MapEmployees step
* Action: Click the cog icon to edit the step configuration. This displays the following



* Action: Click Cancel

In DHCCE we just map our source properties from the raw records to our target properties we’ve defined on our Connect model to create our mapping.



On the left we see the target properties for our Connect model as defined on our Employee entity. On the right we see fields where we’ve created our mappings. When we select the dropdown on the left of the f(x) for the target entity, we see the attributes available for mapping from our source data. When we select one, we’ll be provided a preview of that data for a randomly selected record. It’s easy for MarkLogic to provide this as all text and structure of records was indexed during ingest.

Mapping is where overcome the headache of firstName being fName in one system, givenNameOne in another, etc. We see here that we’ve mapped emp\_id to employeeId.

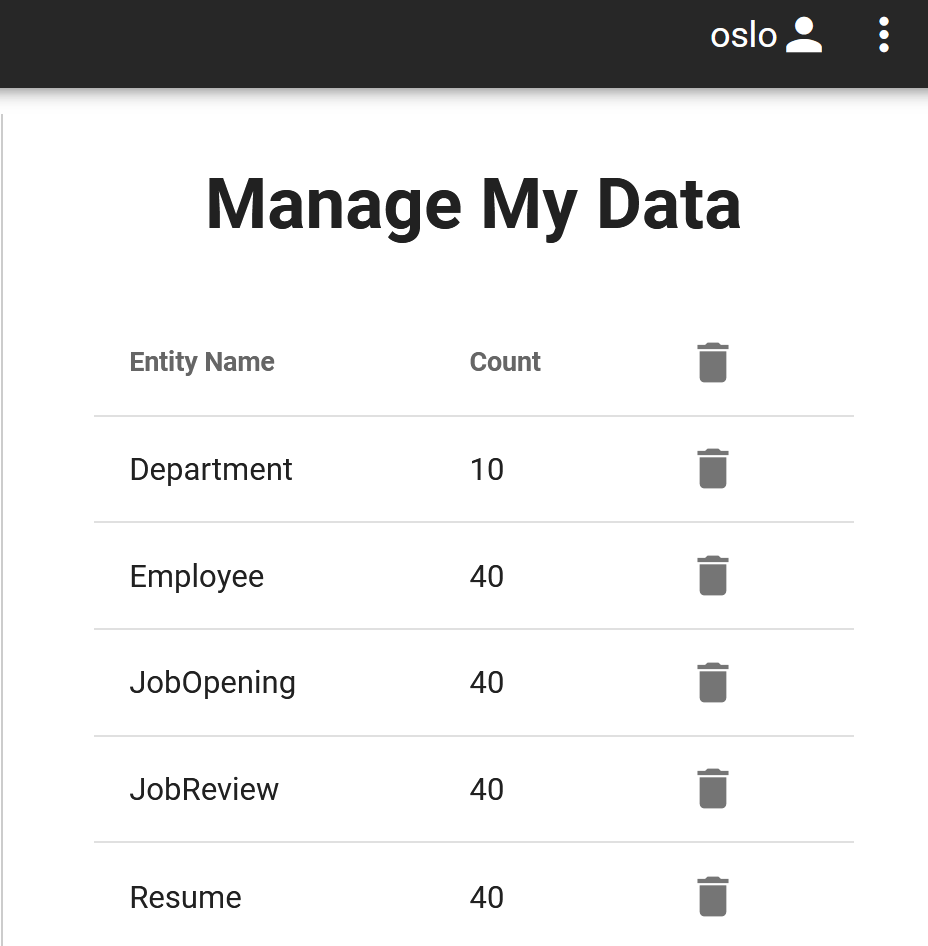
The f(x) allows us to leverage the Data Hub’s built in functions to help cleanse and rationalize our data. We can see we’re concatenating our address values from two attributes in the source and we’re also rationalizing our date so that we’ll be able to leverage dates in searches and queries for ranges and to perform date math.

You notice that not all values are mapped. That’s ok. The target properties can get populated from a different source later. We’re mapping what we have available now to answer the business problem in front of us.

And after our mapping is we created, we just run the mapping step to harmonize our data. We created 5 mappings, one for each of our entity types and data sources.

* Action: Run the mapping steps for MountainTopEmployees

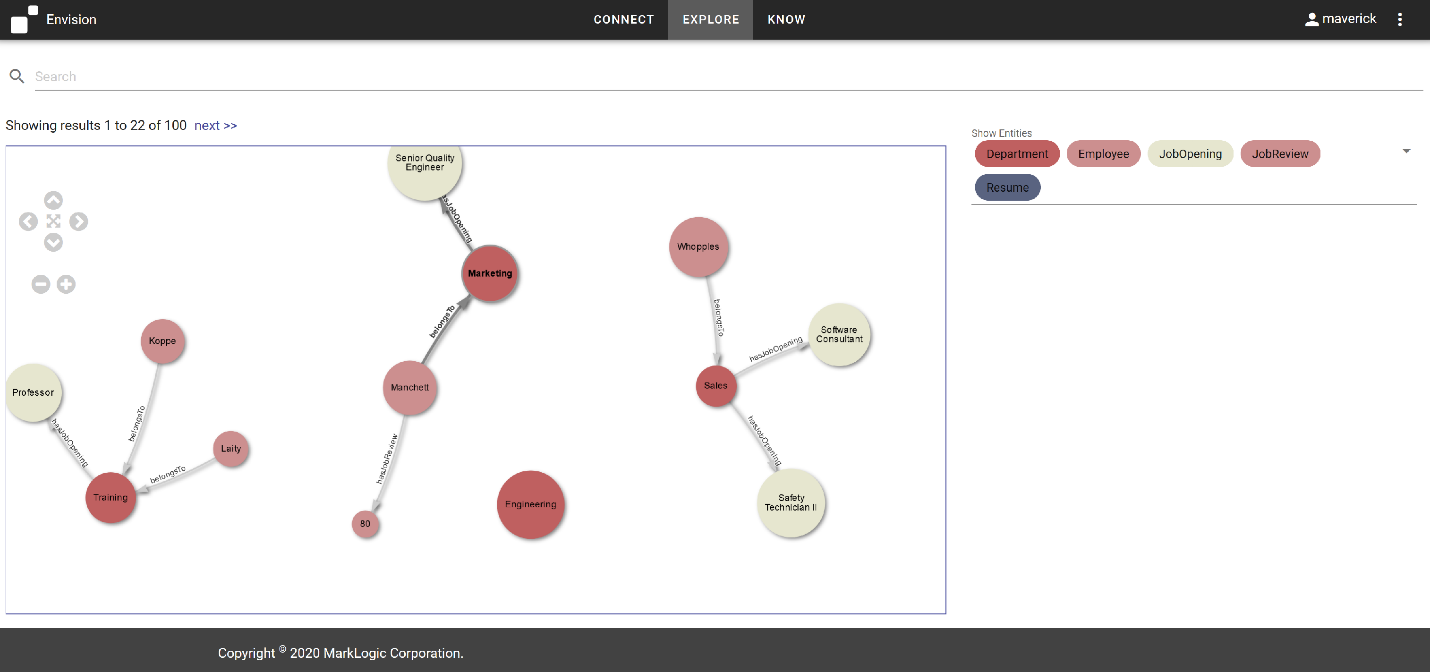
As data is mapped from source to model and harmonized, entity instances are created in the final database. We can see their counts in the “Manage My Data” pane on the right.



And that’s it. We can now take a look at our data in a unified view. For that, we’ll take a look at Explore again.

# 5. Explore

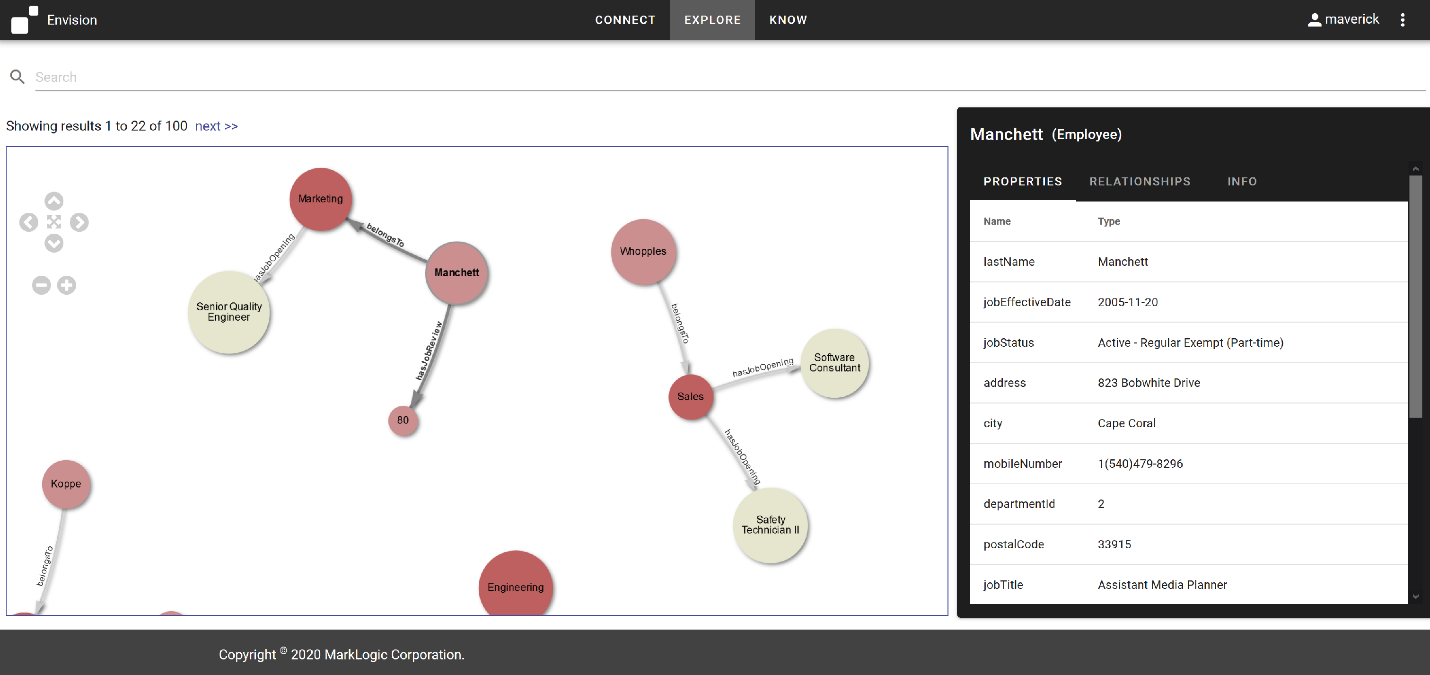
Now that we’ve harmonized our data, we can examine it in a unified view in Explore. Just that simple.



Recall, we had 5 sources of data we harmonized. And now we look at our populated model, and we can see that our data looks just like our model. The conceptual and physical representations of data in MarkLogic are closely aligned, which makes it a lot simple for people to understand and interact with their data.

And we haven’t really done in any integration yet. We’ll revisit this after we load our CoastalEmployees. But this is to show you we already have a way to interact and get value of this data while we continue to enrich and append to it.

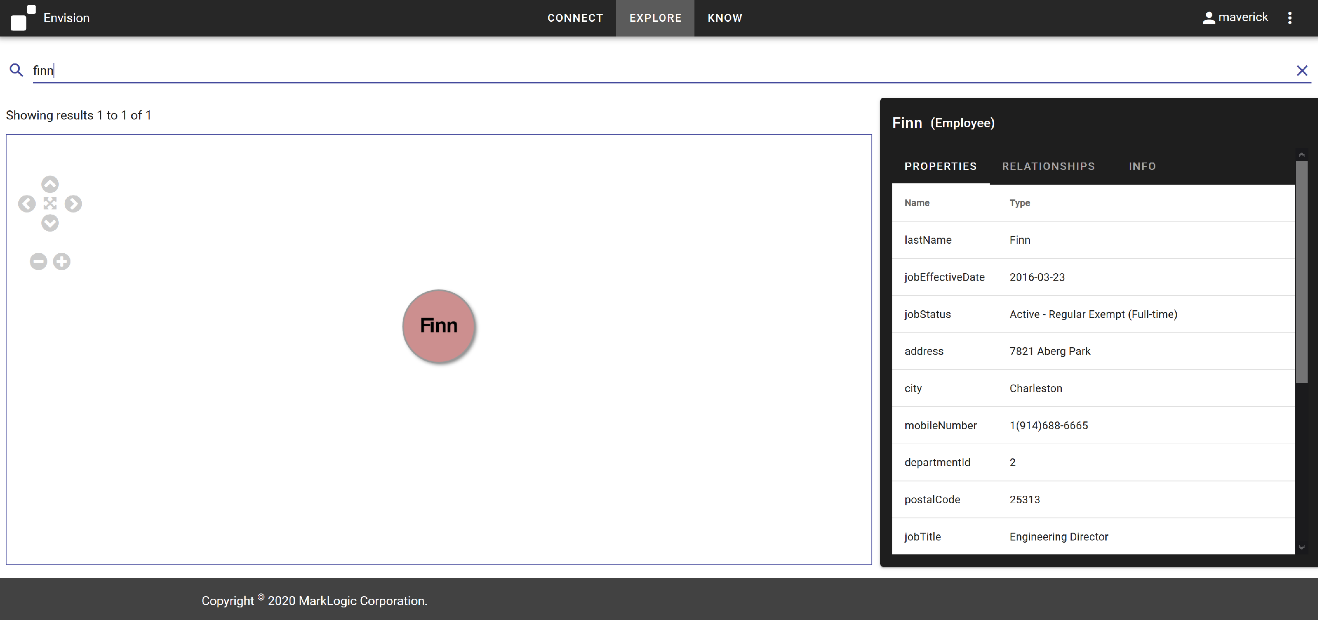
If we click on any entity, we can see the data that was populated from the source.



*OPTIONAL: The labels on the Employee nodes are currently displaying the lastName property. You can actually configure the labels for all nodes in the Connect info tab.*

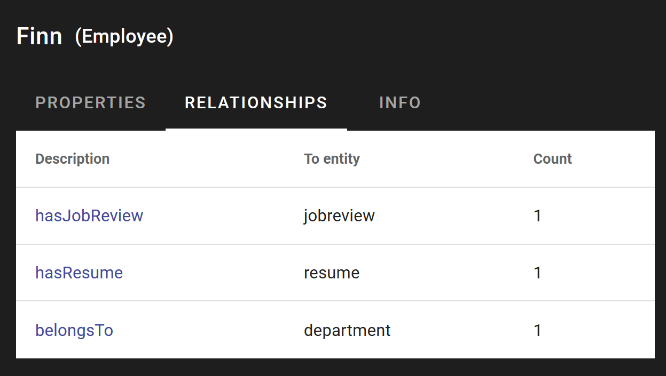
Let’s think about our HR Employee 360 scenario. Adrianna Finn has contacted about new opportunities available after the merger with Coastal. Since MarkLogic indexes all text and structure on index, we can simply search on the last name to find our Employee.

* Action: search on ‘finn’
* Action: Click on the Finn entity to display properties



We can click on the Relationships tab to see what relationships Gael has to other entities.

* Action: Click on RELATIONSHIPS tab to display relationships

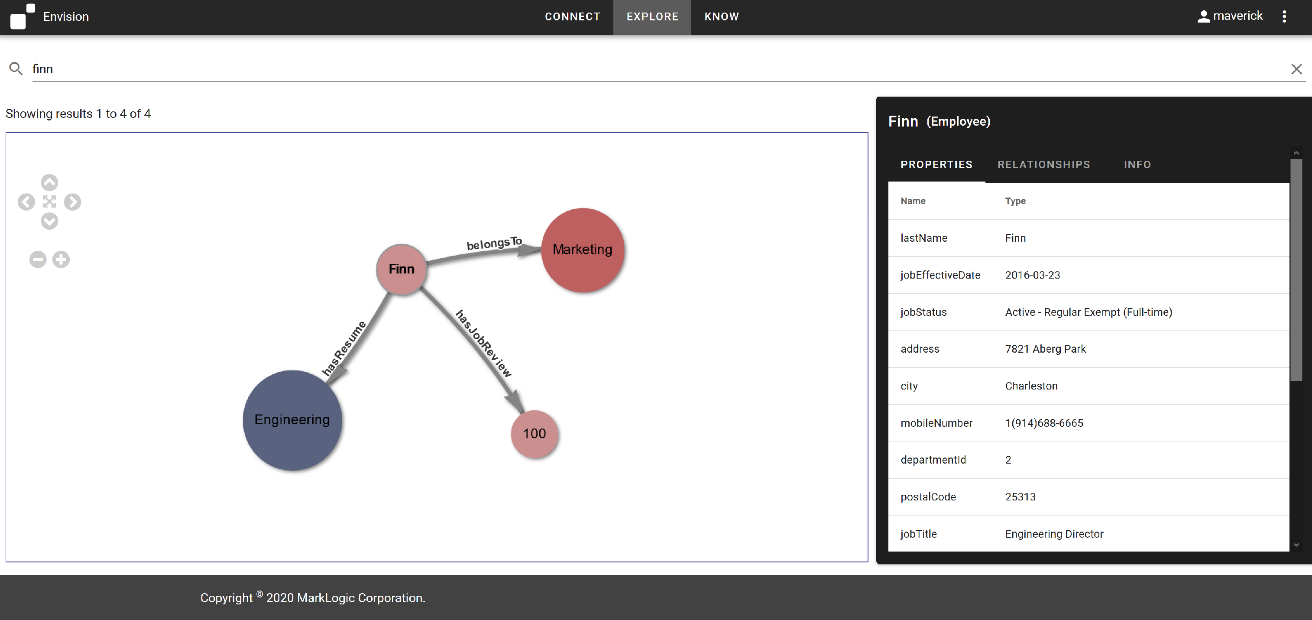


We see we currently have 1 jobreview, 1 resume, and 1 department associated with Adrianna. We can just click the ‘hasJobReview’ description to expand this relationship in the graph.

* Action: Click ‘hasJobReview’ relationship to expand in graph view

We can similarly inspect the associated Resume and Department relationships tab to expand those relationships as well.

* Action: Expand relationships for Finn
  + *OPTIONAL: Expand relationships by right-clicking on Finn entity in graph view*



* Action: Select Finn entity
* Click on ‘Properties’ tab

We can click around the entities to learn more about Finn. We can see she received an exemplary 100 on her last review. We can also inspect her resume. If we wanted to see what other opportunities are available in Marketing, we can expand the hasJobOpening relationship for Marketing to see what positions are available in her current organization.

This will get more interesting when we have data from Coastal in here. But we’re already getting use out of this data.

We know Finn worked at Coastal in the past, prior to coming to MountainTop, so we’re interested in analyzing this data as well to see what her performance in the past looked like and what her experience looked like there.

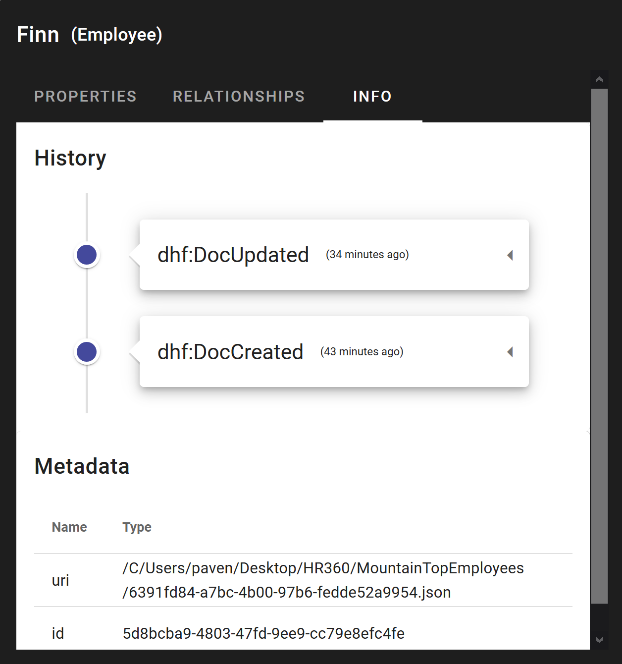
#### Governed by Default

We saw that when we loaded data into MarkLogic we ingested information with security applied automatically on ingest.

Another requirement for governance is understanding, where did this data come from?

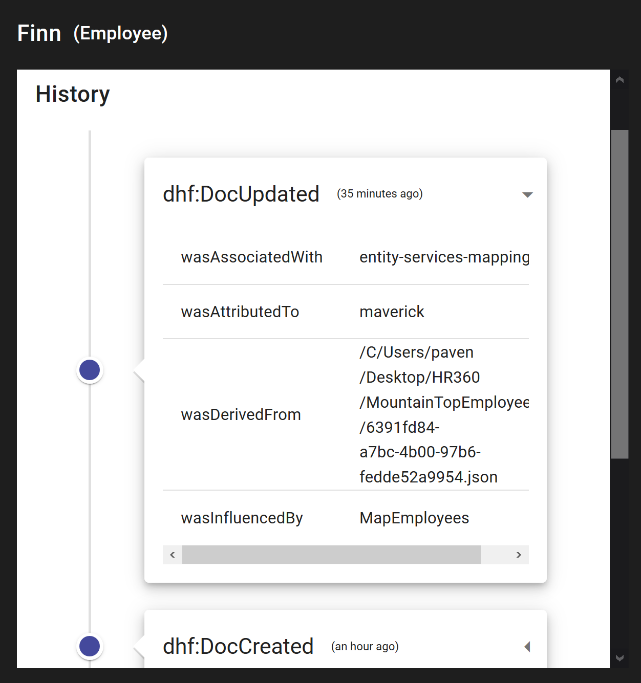
By default, the MarkLogic tracks where information comes from to provide provenance and lineage on data out of the box.

* Action: Select ‘Finn’ Employee.
* Action: Click ‘INFO’ tab



We see an event history of the entity with the latest action applied to the entity at the top of the list. We can click on any event to expand its detail and see just where the source data from the entity came from and came to be.

* Action: Click event dropdown to expand its detail



By default information is saved about where this entity came from, what was the last flow step applied, and who was the user that applied it.

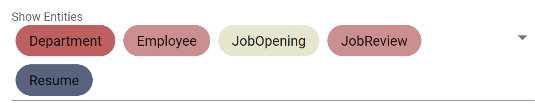
#### The Power of Relationships

One last thing to highlight here is how important relationships are to data integration.

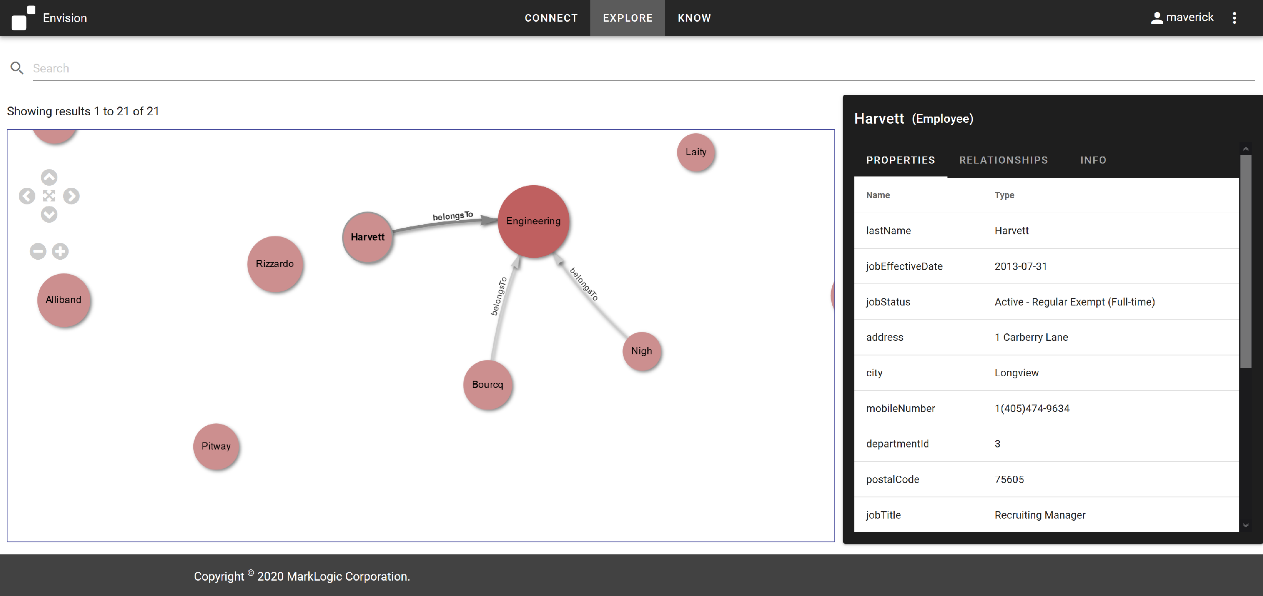
Let’s take a look at just our Employee entities for a moment.

* Action: Clear search field and refresh page

At the bottom of our pane we see an area that tell us what entities, based on the chosen model in Connect, are available for search and use in our Explore pane. The color coding also provides a simple legend for discovery for us.

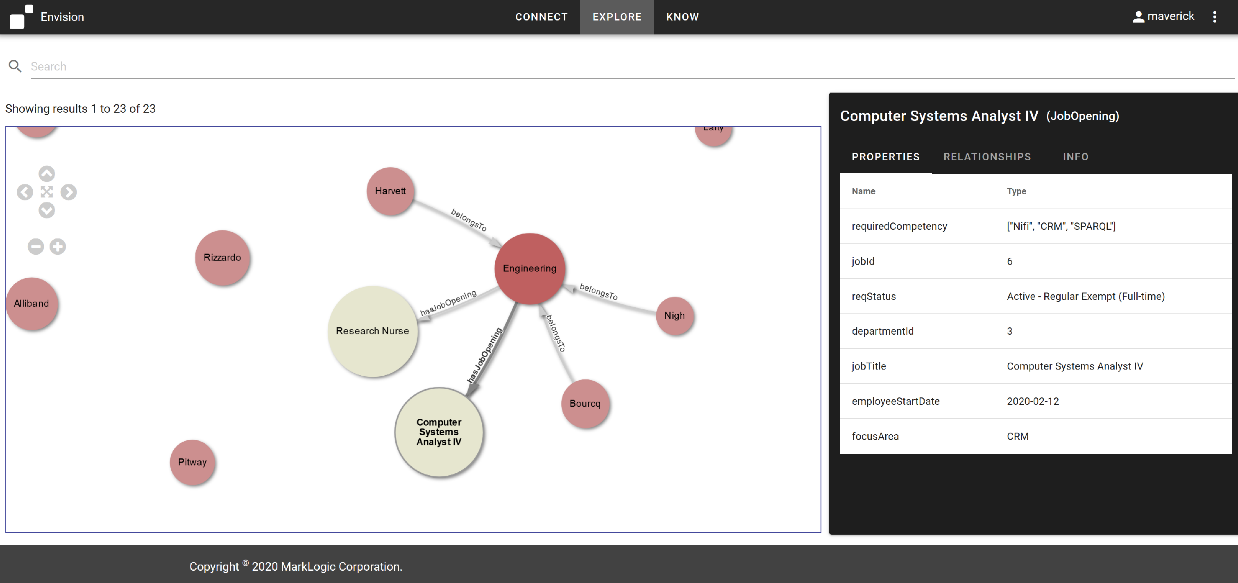


* Action: Expand the ‘belongsTo’ relationships for any of the Employees.



We can easily see and understand the relationship some of our entities have to a certain Department.

We can then expand the ‘hasJobOpening’ relationship from Department to see what JobOpenings are currently available.



#### Knowledge Graphs created by Default

Relationships are a first class citizen in MarkLogic and a graph of integrated entities and relationships is created by default when we integrate data in the data hub. Because of this, we can ‘follow our nose’ to explore data and relationships without having to know up front what the data model is or what the integrated data looks like. We can just follow the relationships to expand the nodes to learn more.

These are Employees that may only be familiar with each other maybe not even know each other at all in real life. But they share a relationship based on their Department. Illuminating these relationships enhances our understanding of our Employee, providing the opportunity for us to meet the requirements for graph use cases such as recommendation engines, product recommendation systems, asset management, social network graphs, knowledge graphs, and fraud detection and analysis solutions. With a greater understanding of Employee connections to other business entities we can start to create future opportunities in our HR Employee 360 solution for job recommendations and asset alignment during the minutes that were previously wasted hunting through silos just to match a single Employee with the right job requirement.

*Note: A deeper demonstration of graph capabilities is available at #7,* ***More on Knowledge Graphs****, below.*

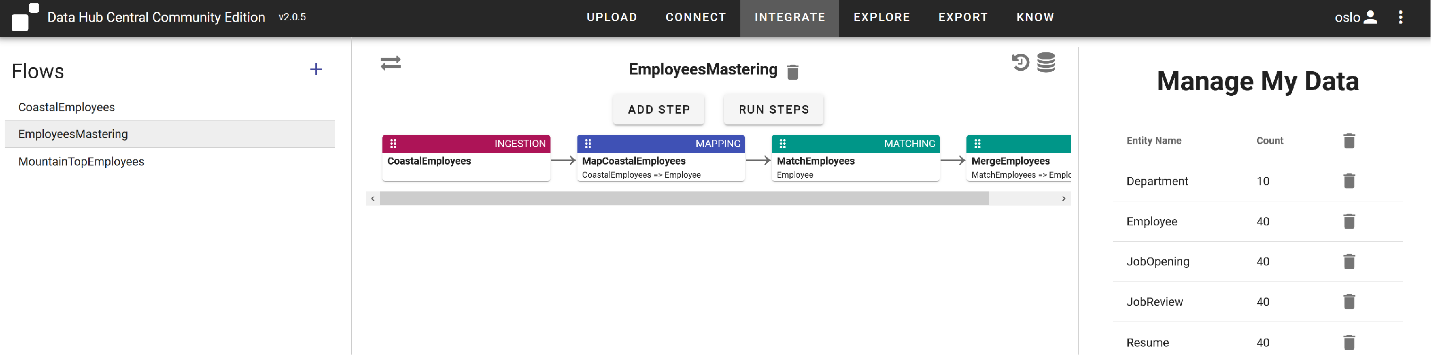
But this isn’t the end of the story. We have more data to load. We want to incrementally and iteratively enrich and append the data in the hub. So let’s continue by integrating the data for our Coastal Employees into the data hub.

# 6. Integrate - Mastering

We see that there are 3 flows defined in our data hub.

The MountainTopEmployees flow brought in data from MountainTop and harmonized it for us. We just explored that data in Explore. Now we’re going to bring in the data for Coastal. However, some of the Employees that exist in Coastal also exist in MountainTop. We’re going to want to merge similar Employees for entity disambiguation and to eliminate duplicates. For that we’re going to look at Smart Mastering in the Data Hub. Let’s take a look at the EmployeesMastering flow.

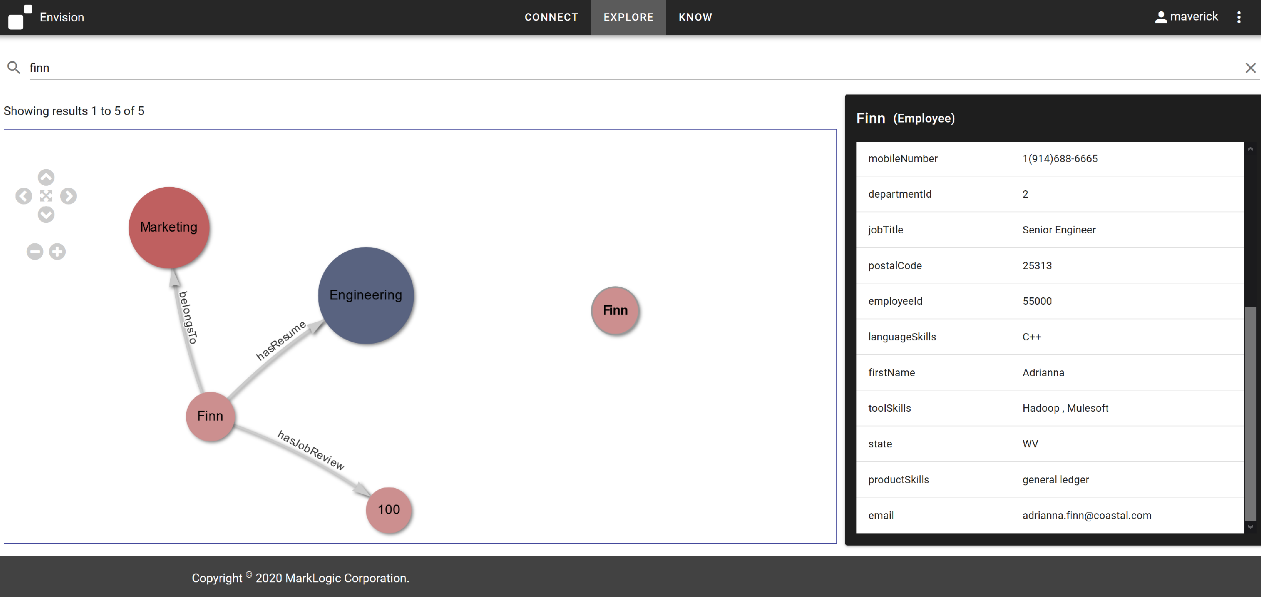
The first two steps of the flow look similar to our employee data for MountainTopEmployees. We have an ingest step and a mapping step for our CoastalEmployee data. We’ll just run those



* Action: Explore the CoastalEmployees ingest configuration.
  + Where the MountainTopEmployee source data was a .csv file, here we say we have a directory of .json documents.
* Action: Explore the MapCoastalEmployees mapping configuration.
  + We see we’re getting some similar, and some different attributes for this data. And some column names are different. But we’re mapping them to the entity model we created at the start. Here we use the string-join() function as well.
* Action: Run the ingestion and mapping steps for EmployeesMastering flow.

Before we run the mastering steps, let’s look at our Employee “Finn” and her properties in the Explore tab.

* Action: Search on ‘finn’, expand relationships, and select the Employee entity with no relationships.

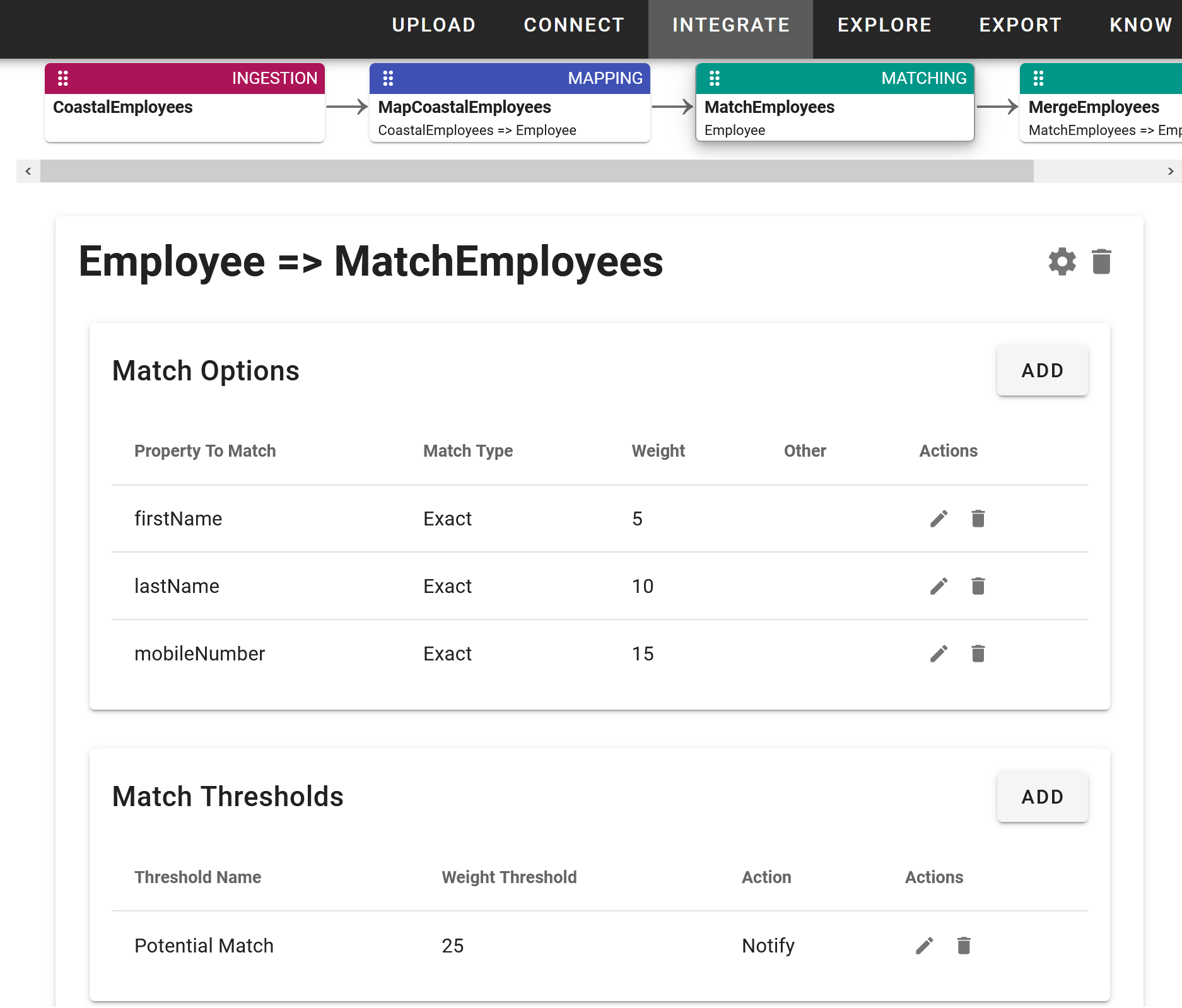


We see there are two of her in there now. Previously there was just one. We can see the new one has a coastal email address. But some of the values for this newly populated Finn are the same as the ones in our MountainTop Finn. This is where smart mastering helps us.

The business is expecting duplicate Employees and has informed us that if the first name, last name, and mobile number of an Employee in MountainTop is the same as an Employee from Coastal, we know they’re the same Employee. We can use MarkLogic’s Smart Mastering steps in the EmployeesMastering flow to master our Employees from Coastal. Mastering in the data hub requires two steps: a Match step for finding matching records and a Merge step, for defining additional rules for how to cleans and rationalize data when similar entities are merged.

* Action: Select MatchEmployees Step

This step will master these Employees using its Match configuration:



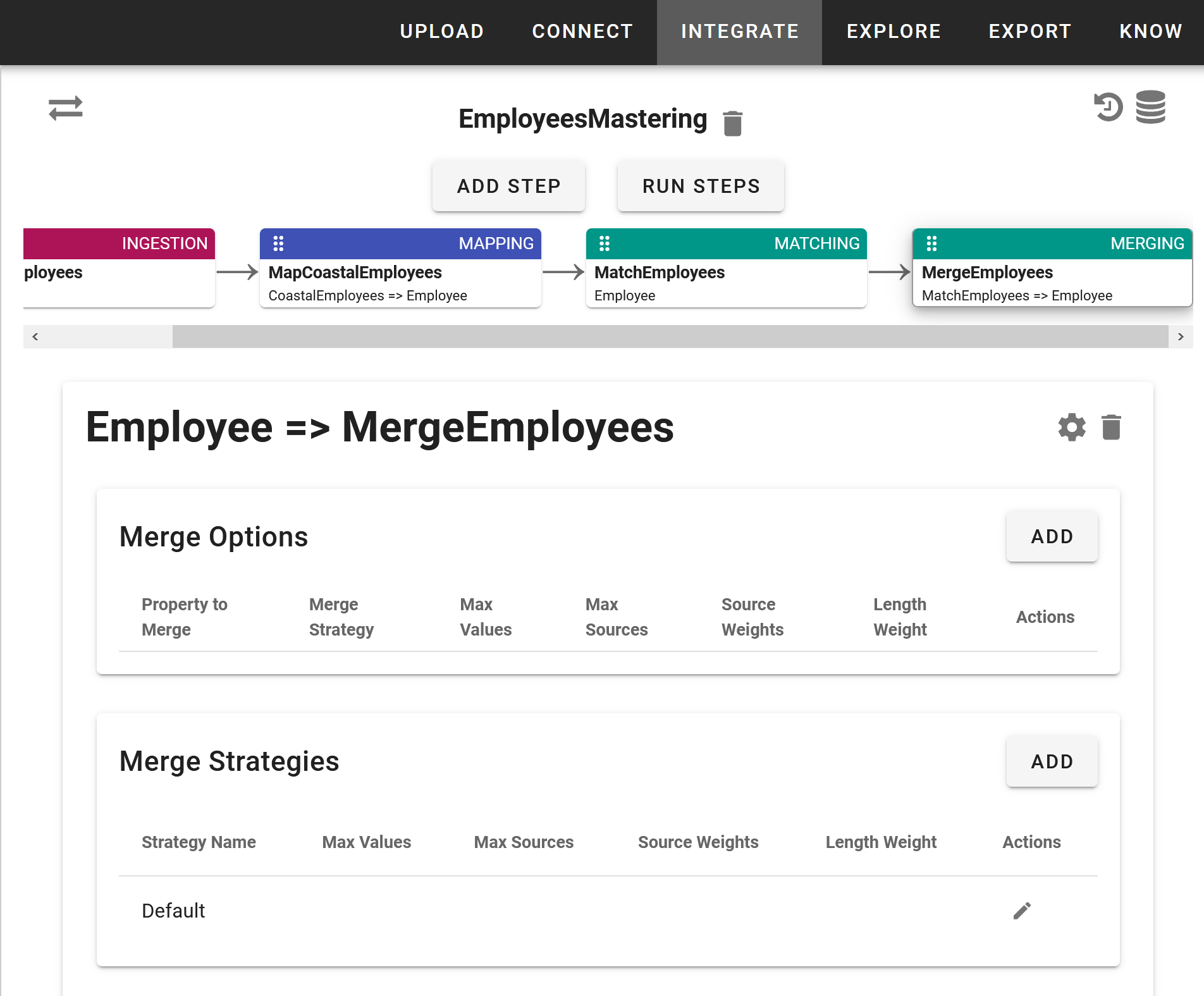
In the matching options, we define the criteria for a match. Here we’re just using exact values, but MarkLogic Smart Mastering provides all sorts of capabilities for us to disambiguate entities using built-in algorithms or even letting you define your own custom rules.

We defined the attributes to inspect, and associate a weight with each. We then set a threshold weight and an action. If the sum of the values for matches reaches the threshold, the action will be taken upon similar entities. Here we’ve defined a notify action. If we choose the merge action, then similar entities will automatically be merged for us. But often we require Human In The Loop (HITL) curation. So with notify, we’ll be able to use DHCCE’s Mastering workspace to review similar records and then decide if we want to merge them or not.

The capabilities for entity disambiguation come out of the box with MarkLogic Smart Mastering. But what do we do if we have a match? Let’s look at the Merge step configuration.

* Action: Select MergeEmployees step

Merge Configuration:



With these options we can configure how two or more records in the data would be merged together. A new record is created with the combined contents of the original records, according to the merging rules you create. The original records stay in the database and are tagged as archived. We’re just using the default merge options here. So we’ll see what this looks in the data after mastering.

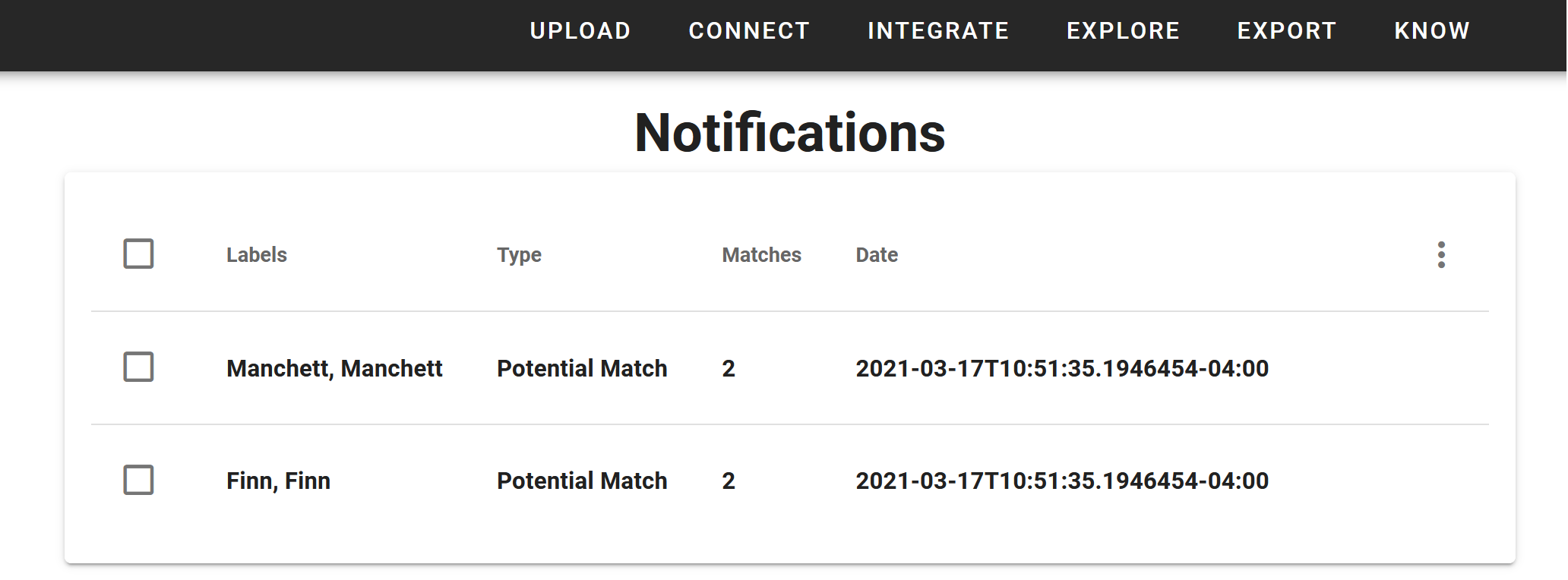
Let’s run our mastering step

* Action: Run the EmployeesMastering flow Match and Merge steps
* Action: Go to the Explore pane and take a look at Finn to see she is still in 2 separate entities

After running, in the upper right we receive a notification we have 2 possible matches to review.



* Action: Click the 3 dots and select “Notifications’ from the dropdown menu.



Here is our notification inbox. We see 2 matches.

* Action: Click the Potential Match labeled Finn to review her matches.



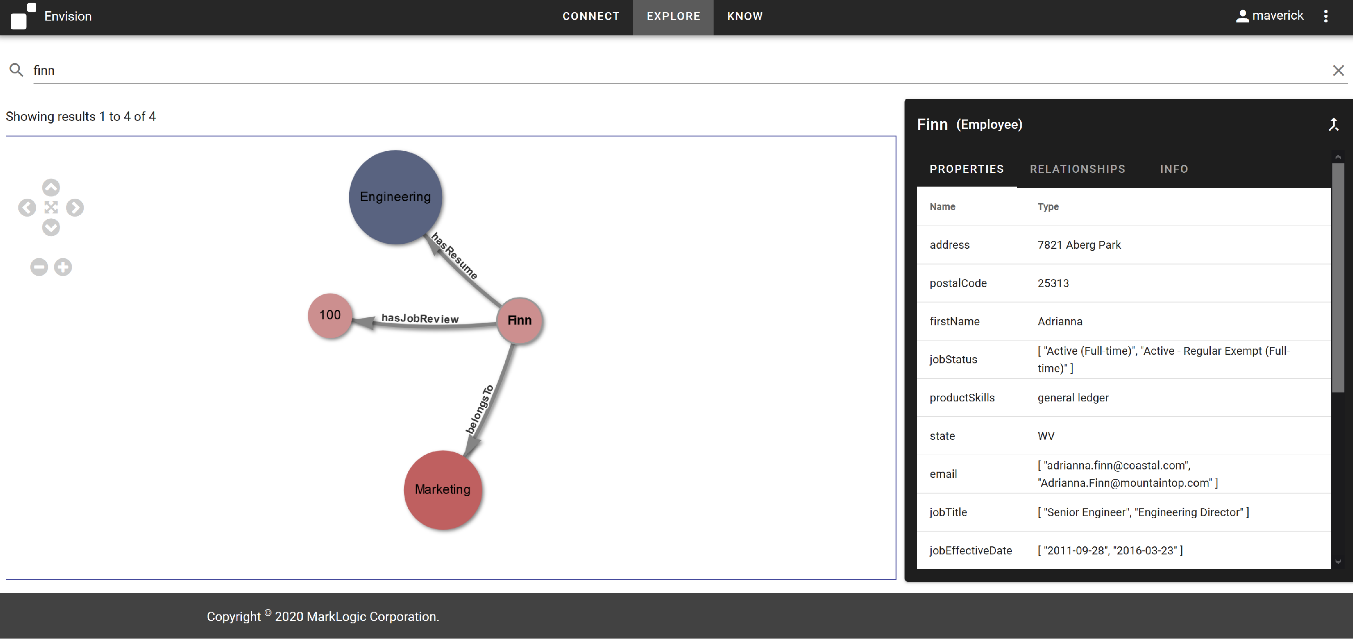
Starting from the left, we see the records that matched for Finn given our mastering rules. On the far right, we see a merged preview of what her entity record will look like if we merge the similar entities.

* Action: Click the merge button

If we select “Block” then merging will be blocked for subsequent of any mastering flows where these records match the match criteria.

Also, while Finn only has 2 matching records in this case, it’s not uncommon to have several records for a single potential match.

* Action: Goto Explore and Refresh search page for Finn



We see we now have a single Finn Employee entity, that’s been merged from 2 disparate Employee records coming from 2 different systems. Her entity information now contains lists of information. This is the default merge behavior of Smart Mastering. The merge options can be used to set rules as to what’s ultimately stored here. But the arrays of information can now be used as keys to join with Resume and JobReviews from Coastal as well. We also see we have the ability to ‘unmerge’ entities. We’ll revisit this in a moment.

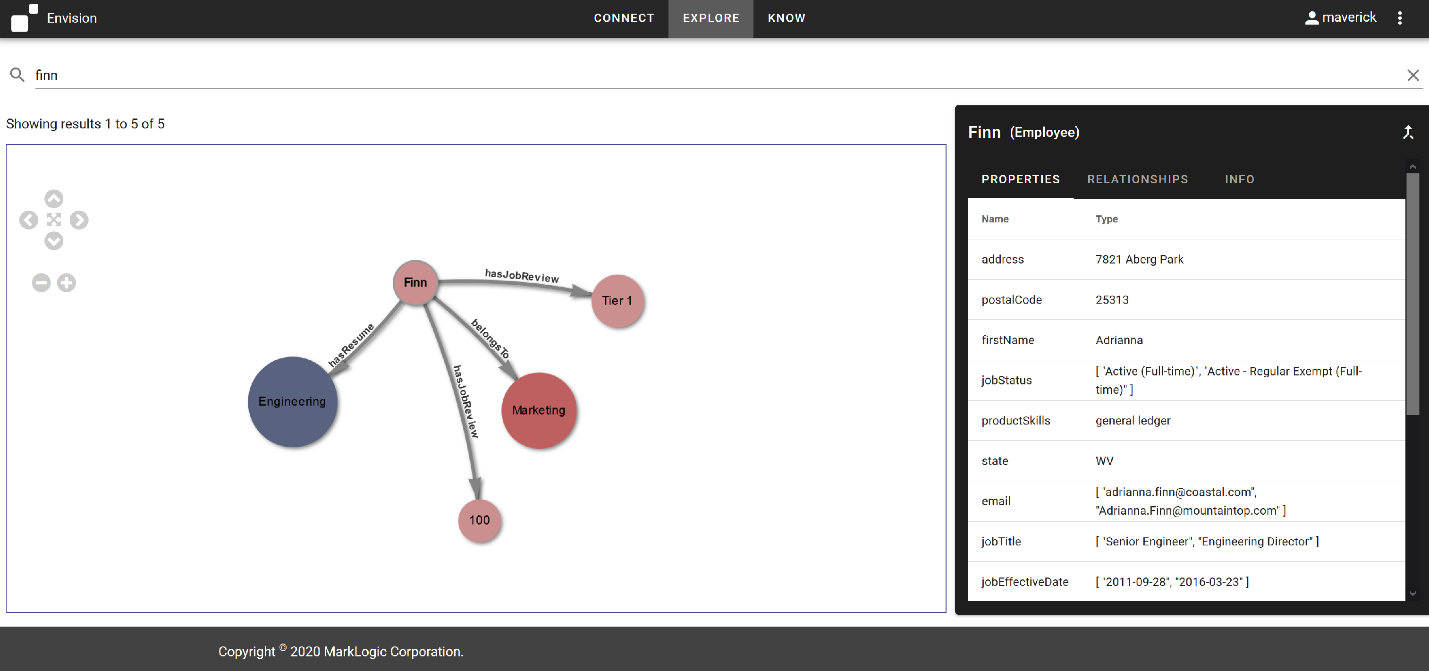
Let’s finish loading our second brand. We’ll add Resumes and Job Reviews using the 3rd, CoastalEmployees flow.

This flow just has 2 ingest steps and 2 mapping steps to map the resume source data to our Resume entity and the performance review source data to the JobReview entity.

* Action: run the CoastalEmployees flow in Integrate.

Let’s look at Finn one last time.

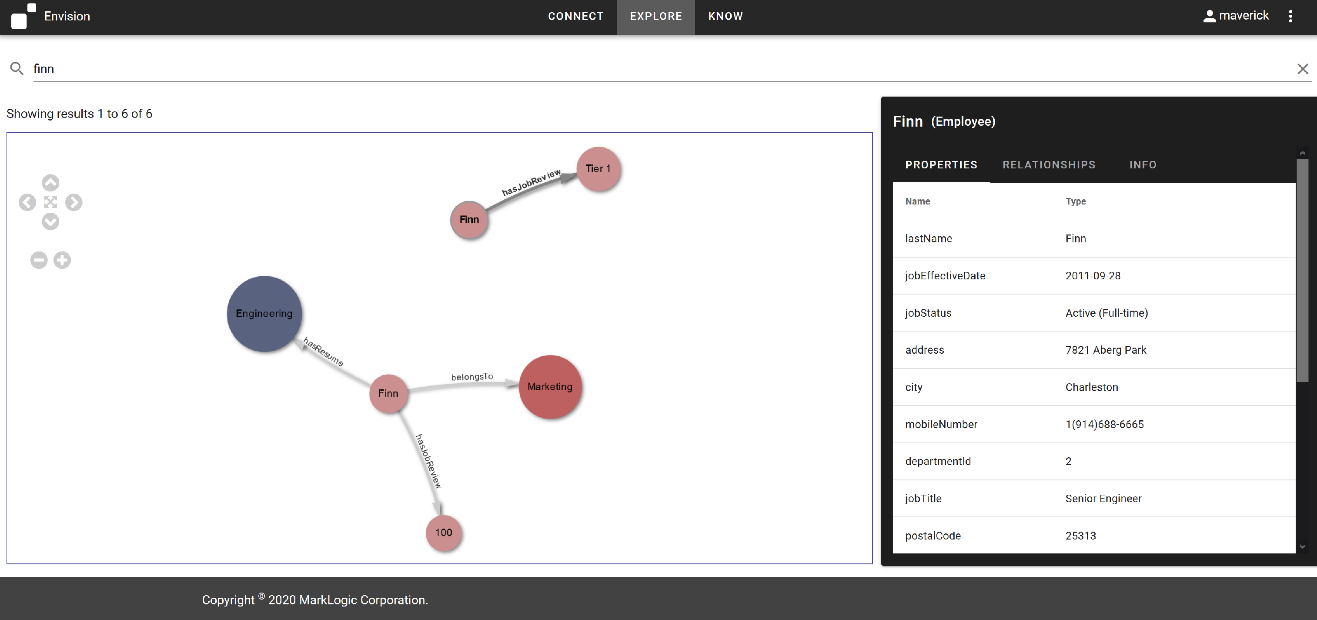
* Action: search on ‘finn’
* Action: Expand all relationships



We see the single, mastered Finn entity now references JobReviews from both MountainTop and Coastal. The HR Analyst now has a single pane of glass to explore Employee data from both companies. We’ve provided a solution for the business problem while demonstrating how we iteratively and incrementally integrate data in the MarkLogic Data Hub.

Finally. Let’s say upon further inspection, the business decides that these Finns are actually separate entities. Well, we can just unmerge them.

* Action: Click Unmerge for ‘Finn’ entity



And as we can see. The relevant relationships are maintained and the data remains usable, and now more accurate that the business has validated the properties of this entity. So it’s very easy to merge, unmerge, and connect data in the data hub. And we always maintain the provenance and lineage for each entity by default.

# 7. More on Knowledge Graphs

#### Expressing Concepts in our Model to Analyze New Relationships, Enrich our Graph, and Connect with Linked Open Data in the Data Hub

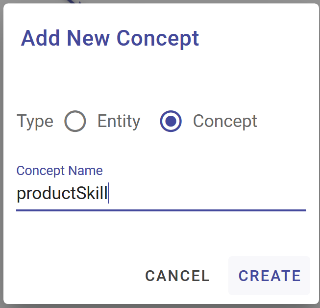
With MarkLogic, it’s simple to add new relationships between entities to explore and analyze new connections between entities as well as to enable them to connect to a larger graph of data stored in the Data Hub. With our Connect whiteboard model, we can illustrate what this looks like in action.

We know our HR analysts want to be able to quickly align Employees to JobOpenings. Enabling this visualization in Explore would make this very simple for them to quickly see who may be a good fit where.

In Connect, we can create a special type of Entity, we refer to here as a Concept that will project the values for and Entity property as nodes in the graph.

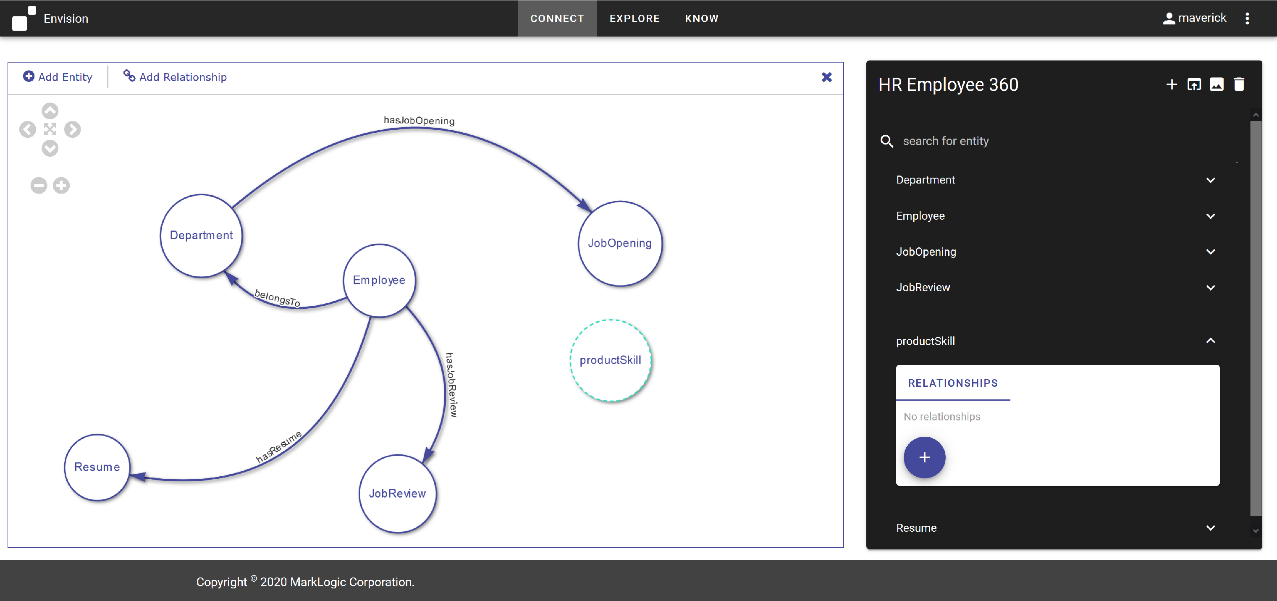
Let’s add a Concept, productSkill, which will create productSkill nodes for our graph based on values for the productSkill property we defined in our Employee entity and the focusArea property we defined in our JobOpening entity.

* Action: Click ‘Connect’ tab
* Action: Click ‘Add Entity’
* Action: Click the ‘Concept’ radio button
* Action: Enter the name ‘specialty for our new Concept

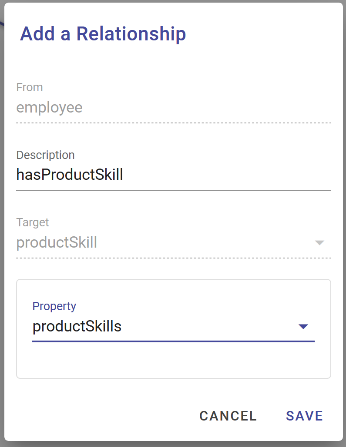


* Action: Click ‘CREATE’

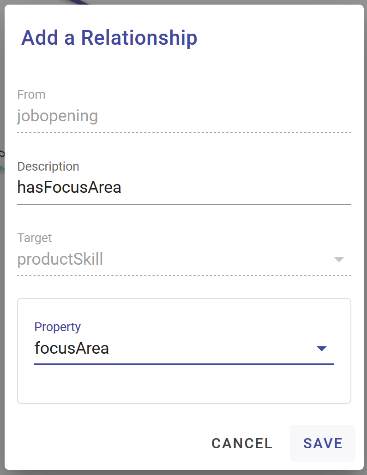
We see special type of Entity added to our canvas. Solid line circles are Entities (Documents). Dashed circles are Concepts (IRIs).



* Action: Click ‘Add Relationship’
* Drag the relationship from Employee to productSkill

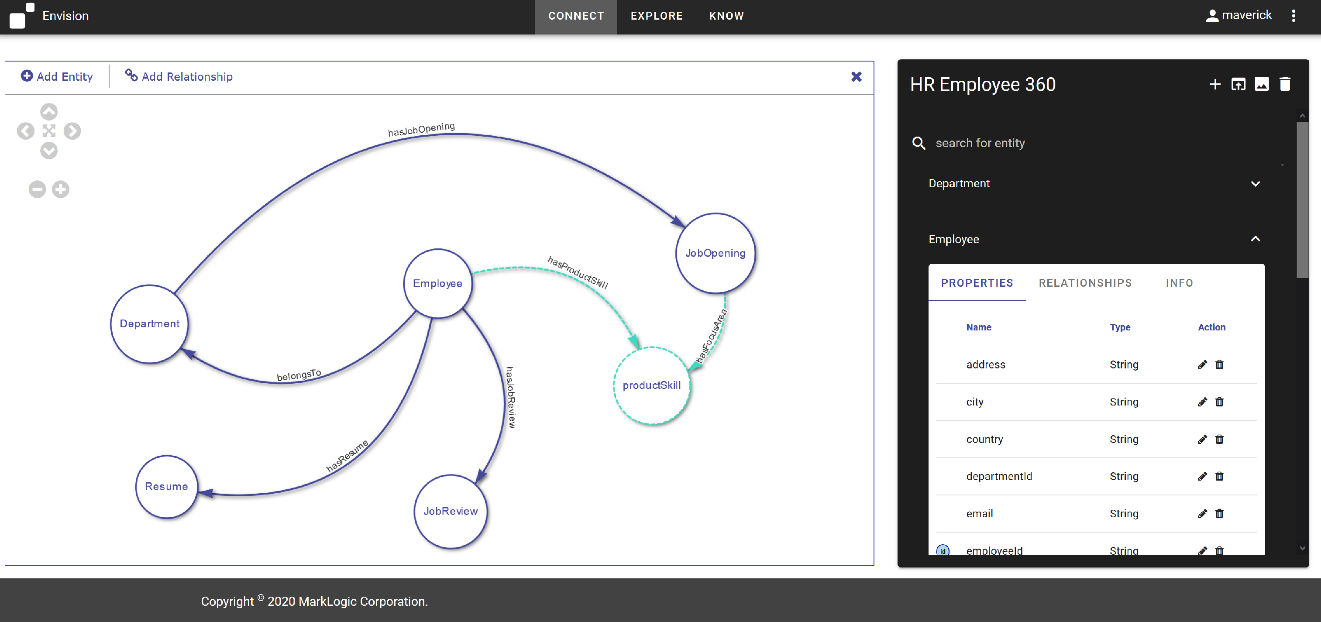


* Action: Enter ‘hasProductSkill’ for Description of the relationship name.
* Action: Select the productSkill property from the list of Employee properties available for the Target in the relationship
* Click ‘SAVE’
* Action: Click ‘Add Relationship’
* Drag the relationship from JobOpening to productSkill



* Action: Enter ‘hasFocusArea’ for Description of the relationship name.
* Action: Select the focusArea property from the list of JobOpening properties available for the Target in the relationship
* Click ‘SAVE’

We now see our productSkill concept related to Employee and JobOpening defined on the canvas.

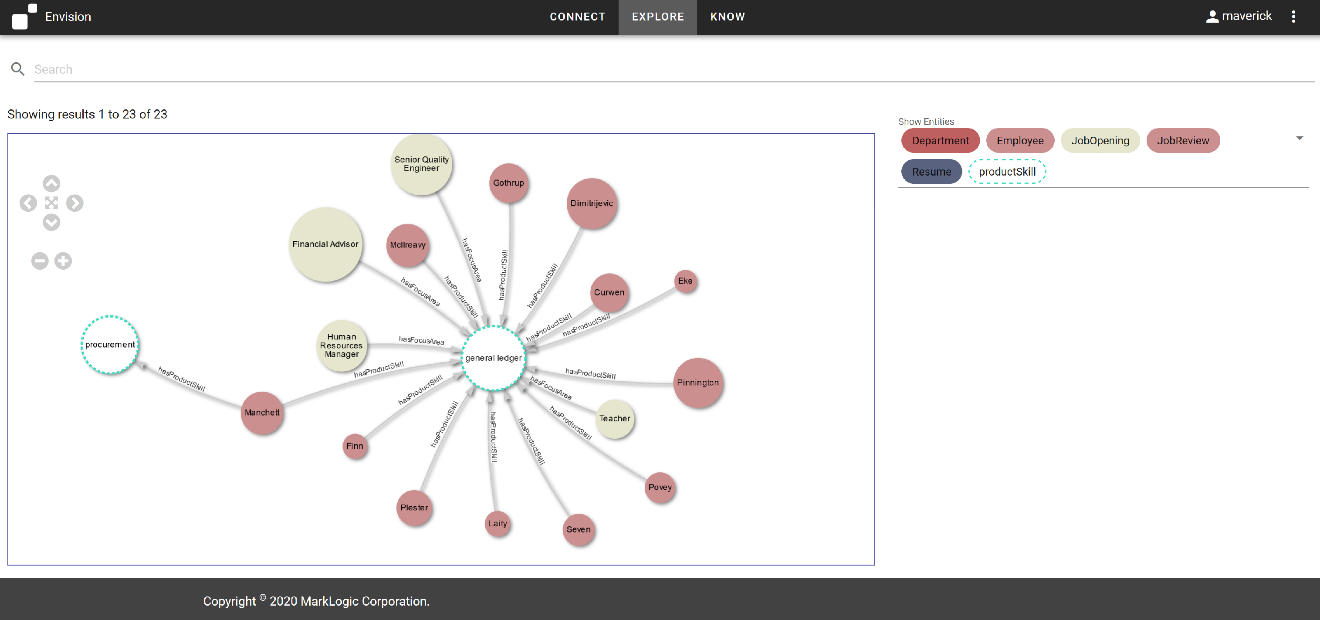


Let’s now go look at the Explore tab.

* Action: Click the ‘EXPLORE’ tab

We now see our entities connected by the concepts we’ve expressed through our model. The values for the Employee projectSkills property and JobOpening focusArea properties are projected as Concepts that allow us to visualize the relationships between entities based on these values.

If our HR Analyst now expands the relationships for ‘general ledger’ in Explore , they can now quickly see what JobOpenings are available that require this particular skill and who in the company has this experience.



In addition to connecting the existing data, we can leverage these concepts to connect our integrated data with other ontologies and linked open data we’ve loaded into the hub to allow us a richer view and level of accessibility to our data.

*Note: Connections can only be made from Entity to Concept to express a property’s value.* ***Connections can NOT be made from Concept to Concept.*** *That’s the work of an ontology editor. The feature here is to make it so Entity values can be given the ability to plug the harmonized model into a graph*

# 8. Export

We know that many of our consumers will still want to interact with the data we’ve integrated in the way they’re familiar with. NoSQL in MarkLogic means Not Only SQL. But MarkLogic can definitely do SQL

From the Export tab, we can export .csv files of our harmonized entities and then open these in our BI Tool of choice.

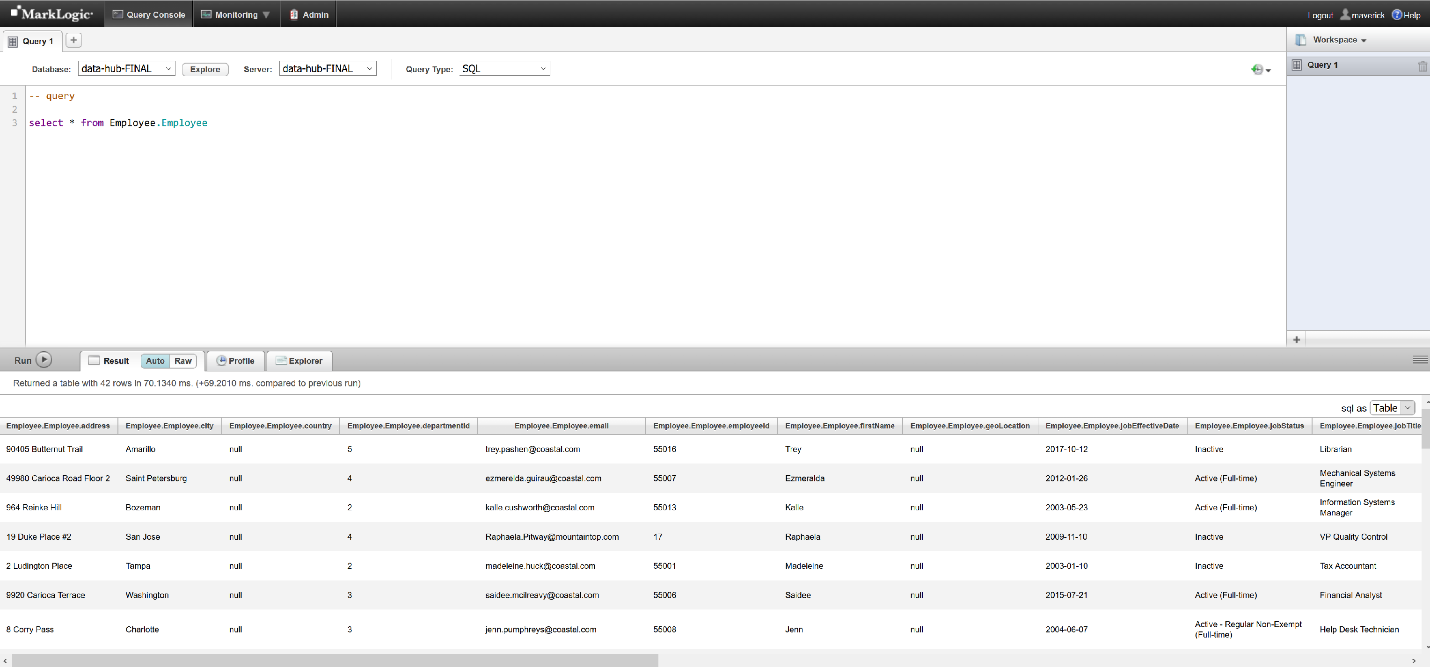
MarkLogic also has connectors for Tableau and PowerBi that can also be used with the following default views created in section 10 below.

# 9. Query Console - SQL Views

When we created our Connect Model, we actually created a tabular view of each entity based on the entities, properties, and relationships defined. This happens by default in MarkLogic for the entities we’ve defined. The data hub allows us to create Employee SQL views using a declarative template as well.

What you’re looking at here is Query Console (Available by default at localhost:8001 ). This is MarkLogic’s query and development environment. Here I select the database for our harmonized data, and just issue a query to show me what we have in the Employee view.

This is the same data we were looking at in the graph view, just in table form. One of the things MarkLogic is REALLY good at is providing different types of lenses on a single set of data so people and applications can consume it in the formats they’re familiar with.



You can also select \* from sys\_tables; for viewing all the views currently available.

# Conclusion

And that’s what integrating data in MarkLogic looks like in action. When we say Load, Curate, Access, you’ve seen that we don’t necessarily start by loading data first.

1. We start by defining the business problem we want to solve and model our solution as the key business entities and relationships we want to be able to interrogate to provide our answers. We created this model using **Connect**.
2. In the MarkLogic Data Hub we define flows for ingesting data, mapping our sources to our model, and mastering our entities for disambiguation. We can also define custom steps. We used **Integrate** for this.
3. We explored our data and saw it looked exactly like our model using **Explore**. We could search and expand relationships as well as view the provenance and lineage associated with each entity.
4. Throughout this process we saw how we can iteratively and incrementally append and enrich our data in the hub, as well as unmerge mastered entities.
5. And finally, we saw the **MarkLogic Data Hub** provides the ability to easily create multiple lenses on the same set of data so that users can access their data via custom web applications, SQL, or a BI tool such as Power BI or Tableau.

MarkLogic allows you to integrate what you need as you need it and has data in use in ¼ to 1/5 the time of integrating data and delivering results using traditional tools and applications.

We’ve shown you today using this sample data. But we’d like you to experience this with yours. We hope this guide helps accelerate your success!

# APPENDIX

*For Information on how to use Connect and Explore not covered in this walkthrough, please refer to the guides for the individual components on the wiki and on github.*