MET CS 669 Database Design and Implementation for Business
Term Project Iteration 2

Name: Katherine Rein

Table of Contents

Project Direction Overview

For the past 2 years I have been involved in a research lab with Dr. Christoph Nolte in the Earth and Environment and Data Science departments. The work I've been doing involves validating a database of land transaction values for conservation work all over the US. One of the goals of this project is to create an economic model to predict land costs. Another reason that this project was taken up, was so that finding land transaction data was easier and more studies could be done using this data. Public data is notoriously awful and so creating a publicly available dataset would do wonders for the research world.

Building off of this work that I have been doing, I would love to use this project to create a way to help with understanding the data we collected better. This could mean lots of different things, but I see this as a way to help merge, visualize, and validate our data. I assume this will simplify the process for the end user. With a simplified process, people that don't know much about coding but know a lot about environmental science and economics can use this data efficiently. This means that the database will primarily be used by scientists and researchers but could be used by anyone wanting to understand this data.

The database will contain all the fields that we validated as a lab group (Appraised Value, Purchase Price, Acreage, Date of Transaction, Protection Type, Government Spending). I am unsure how much identification data I will need but I have multiple fields for that as well (CT ID, Municipality, County, State, Assessor Parcel Numbers). That is where I will start in terms of fields that the database has, however, that could change as I make it more complex. One way I could add data would be including GIS polygons for each transaction. This would allow us to manipulate the data spatially. I am unsure if our tools are set up for GIS data, but it could be a fun extension of the class. I am currently applying to jobs every day that utilize GIS daily to help with their environmental analysis.

I am interested in this topic not only because I want to help make my life easier as a research assistant, but because I want to feel like I've contributed to helping my lab group. While I have validated almost 700 transactions all over the country, I don't feel like I have done much of anything meaningful at the job. Being able to create a tool that will help beyond validation work sounds right up my alley. Hopefully this will be more than just something for me and people can use this for their research to make a difference in the climate crisis.

Use Cases and Fields

(1) Input Data Use Case

Data has already been inputted and we are adding to it or we are starting a brand new database of information.

1. Load in all CT IDs, polygons, and APNs for the whole of the US
2. Ensure data contains at least one of the above 3 identifiers for each row and that each is unique
3. Handle duplicate data (within a data input and with existing data)
    a. Create additional rows or replace rows
    b. Combine Assessor Parcel Numbers into one row based on CT IDs
4. Delineated which column of the inputted data goes in which of our column names

Field	What it Stores	Why it's Needed

Data_Column_Names	These are the column names on the inputted data that we want to use.	This will tell the computer which of the columns to look at in the data inputted to make sure the right data gets inputted in the right spot. This also helps with efficiency as these databases can be very large.
Our_Column_Names	These are the column names on the database that we want to update with the new data.	This will tell the computer exactly which column to line up the values from Data_Column_Names with.
CT_ID	This is the unique identifier number from the Conservation Almanac. This is most often used by the PLACES Lab and is considered the most specific identifier.	This unique identifier helps researchers add more information or update information.
Polygon	This stores the GIS data that tells us spatially where the data is.	This unique identifier helps researchers add more information or update information. It also helps in matching data without a ct_id to a row with a ct_id.
APN	This stores the Assessor Parcel Number identifier. This is an identifier that is most commonly used by the government and specifically the tax assessor when discussing properties.	This unique identifier helps researchers add more information or update information. It also helps in matching data without a ct_id to a row with a ct_id.

(2) Validate Data Use Case

Check the data from the Conservation Almanac to see how far off the data was.

1. Select a variable to validate
2. Load in data from The Trust for Public Land's Conservation Almanac for the selected variable
3. Calculate percent difference for the variable for all rows of our data that have a ct_id
4. Graph or map the percent difference to visualize the accuracy of the conservation almanac (and potentially locate input errors

Field	What it Stores	Why it's Needed
Validation_Variable	This stores the variable that we want to investigate in relation to the Conservation Almanac.	This is needed to limit how much data we have to load in and look at if we only need to look at one variable at a time.

Almanac_Data	This is the data from the conservation almanac for the Validation_Variable.	Without it, we would have nothing to compare our data to.
Our_Data	This is our data filtered down to just the Validation_Variable.	Without it, we would have nothing to validate.
Percent_Difference_Validation	This stores how much our data and the conservation almanac data differs.	This is how we can tell how needed our validation was. It can also help locate input errors.

(3) Under/Overpaying Use Case

Identify if each transaction was a deal or not based on the appraised value and the purchase price.

1. Identify all rows in which there is both appraised value and a purchase price
2. Calculate the percent difference between the appraised value and the purchase price
    a. A negative value means the buyers paid less and a positive value means they paid more
3. Graph or map the percent difference to visualize the areas where buyers were over or under paying for the land or easements

Field	What it Stores	Why it's Needed
Appraised_Value	This stores the appraised value of the transaction for that polygon, ct_id, or APN.	This will tell us how much the transaction was worth at or around the time of the transaction.
Purchase_Price	This stores the purchase price of the transaction for that polygon, ct_id, or APN.	This will tell us exactly how much was spent on the transaction.
Percent_Difference_Cost	This stores the percent amount that the purchase price is greater than the appraised value.	This will tell us if the buyer had to pay more or less than the Fair Market Value and by how much.

(4) Best Year Use Case

Given a span of years, identify the best year to buy land or an easement in a certain area.

1. Identify a range of years that you would like to study
2. Identify an area of interest that you would like to study (city, state, municipality, county, region, etc.)
3. Normalize the purchase price to 2024 dollars
4. Calculate the Price per acre for each transaction in that area during that time frame
5. Map/Graph the data to see if there is a year with the cheapest per acre cost

Field	What it Stores	Why it's Needed

Area_of_Interest	This is the city, state, municipality, county, region, etc. that we would like to study.	This sorts our data down to just the area of interest that we want to look at making it more efficient to compute and analyze.
Year_Range	This is range of years that we would like to study.	This sorts our data down to just the time frame that we want to look at making it more efficient to compute and analyze.
Purchase_Price	This stores the purchase price of the transaction for that polygon, ct_id, or APN.	This will tell us exactly how much was spent on the transaction.
Normalized_Purchase_Price	This stores the purchase price of the transaction for that polygon, ct_id, or APN while accounting for inflataion and normalizing it to 2024.	This will tell us how much was spent on the transaction while taking inflation into account so that the data isn't skewed towards older transactions.
Acreage	This stores the acreage of each transaction for that polygon, ct_id, or APN.	This will tell us how much land was bought or had an easement placed on it.
Price_per_Acre	This stores how much was spent per acre for each transaction.	This will create a more equal measure of how much is being spent on the land as transactions are of all different sizes but an acre is one size.

(5) Response Rate Use Case

See how different types of transactions are treated during the validation process in terms of response rate.

1. Filter rows by contact status using a binary method
    a. A transaction either has a response or no response
2. Select a value to test
    a. Ex) Local/State/Federal, Size of Transactions, States
3. Sort transactions into different buckets of said value
4. Graph or map to view trends of the response rate

Field	**What it Stores**	**Why it's Needed**
Contact_Status	This stores a value from 0 to 11 indicating where the transaction is in regards to contact status.	This is used to help us see where we are on the timeline of completing a transaction which can then be simplified into a binary variable.

	0. Not known 1. Not contacted 2. Initial email sent 3. Follow-up email sent 4. Voicemail left 5. Right person 6. Has responded 7. Spoke on the phone 8. Plans to share data 9. Will not share data **10. Shared some data** 11. Shared all data	
Responded	This stores a binary variable that is 1 if the Contact_Status is a 10 or 11 and stores a 0 if the Contact_Status is from 2-9.	This is important because now we have simplified our understanding of where each transaction is and can do analysis more easily and efficiently.
Value_of_Interest	This stores the column of interest with a limited number of buckets to ease the graphing/mapping process.	This is important because this is the value we want to look at and how response rate varies for it.

(6) City Center Variation Use Case

See how different variables vary as the distance from a city center varies.

1. Identify a value of interest
    a. Ex) Acreage, Price per Acre, Response Rate
2. Load in or calculate the center of each transaction
3. Identify the closest city to each transaction (center)
4. Calculate the distance from the center of the transaction to the closest city
5. Graph the value of interest vs the distance from city center

Field	What it Stores	Why it's Needed

Value_of_Interest	This stores the column of interest for seeing how city center distance varies for it.	This is important because this is the value we want to look at and how city center distance varies for it.
Center_of_Transaction	This stores the longitude and latitude of the center of each transaction.	This is needed so that we have one point to calculate the distance to the city from.
City_Centers	This stores the longitude and latitude of the center of the closest city to the transaction.	This is needed so that we can calculate Distance_from_City.
Distance_from_City	This stores the distance in meters from the center of the closest city.	This is needed so that we can understand the ways that the Value_of_Interest varies with Distance_from_City.

Structural Database Rules

(1) Input Data Use Case

1. Each ct_id may have zero to many APNs; each APN may have one ct_id.

The ct_ids have to have at least one APN but each APN does not have have a ct_id. The APN cannot be used by multiple ct_ids.

2. Each ct_id has one to many polygons; each polygon has exactly one ct_id.

The ct_ids can have area that is not all in one polygon meaning that it requires multiple polygons to draw the area. All of the polygons taken from The Trust for Public Land's Conservation Almanac are related to only one ct_id as there are no polygons that overlap.

3. Each APN has one to many polygons; each polygon has one or more APNs.

The APNs can have area that is not all in one polygon meaning that it requires multiple polygons to draw the area. Some of the polygons take up the area of more than one APN.

(2) Validate Data Use Case

4. Each data point from The Trust for Public Land's Conservation Almanac has a ct_id; each ct_id may have a data point from The Trust for Public Land's Conservation Almanac.

All data that has been inputted into The Trust for Public Land's Conservation Almanac has been organized by ct_id. Not all data fields are filled in on The Trust for Public Land's Conservation Almanac.

5. Each Percent_Difference_Validation has a ct_id; each ct_id may have a Percent_Difference_Validation.

The validation is done based on ct_ids so if we can compare two numbers then they have to have a ct_id. Not all ct_ids have data from either our labs validation or The Trust for Public Land's Conservation Almanac so we wouldn't be able to calculate a value for that.

6. Each data point from The Trust for Public Land's Conservation Almanac may have a Percent_Difference_Validation; each Percent_Difference_Validation has a data point from The Trust for Public Land's Conservation Almanac.

All data that has been inputted into The Trust for Public Land's Conservation Almanac has been organized by ct_id. Not all data fields are filled in on our data. Because of this there could be rows that don't have anything to compare The Trust for Public Land's Conservation Almanac with.

(3) Under/Overpaying Use Case

7. Each Appraised Value may have exactly one Purchase Price; each Purchase Price may have zero to many Appraised Values.

Sometimes we get data from multiple different places that is different but all are simply opinions of appraisers on what the appraised value should be. The purchase price can only be one number.

8. Each Appraised Value has one or more ct_ids; each ct_id may have one or more Appraised Values.

The only way an appraised value could have multiple ct_ids is if at least 2 different transactions have the same appraised value by chance. This means nothing about correlation between the transactions. Sometimes we get data from multiple different places that is different but all are simply opinions of appraisers on what the appraised value should be.

9. Each Appraised Value may have one or more Percent_Difference_Cost; each Percent_Difference_Cost has one or more Appraised Values.

Like before, both of these numbers aren't unique identifiers and so they could be the same across transactions which is why both are plural relationships. The Percent_Difference_Cost has to have an Appraised Value because there is no other way for the Percent_Difference_Cost to be calculated without it.

10. Each Percent_Difference_Cost has a ct_id; each ct_id may have a Percent_Difference_Cost.

The cost analysis is done based on ct_ids so if we can compare two numbers then they have to have a ct_id. Not all ct_ids have data from our lab's so we wouldn't be able to calculate a value for that.

11. Each Percent_Difference_Cost has a Purchase Price; each Purchase Price may have a Percent_Difference_Cost.

Like before, both of these numbers aren't unique identifiers and so they could be the same across transactions which is why both are plural relationships. The Percent_Difference_Cost has to have an

Appraised Value because there is no other way for the Percent_Difference_Cost to be calculated without it. There can not be multiple opinions on the Appraised Value.

12. Each ct_id may have exactly one Purchase Price; each purchase price has one or more ct_id.

Like before, the purchase price isn't a unique identifier and so it could be the same across transactions which is why it is a plural relationship. Not all ct_ids have purchase prices because we don't have a complete validation dataset yet.

(4) Best Year Use Case

13. Each Year_Range has one to many Years; each year may be in a Year_Range.

A year range has to have a least one year but not all years are going to be used in a year range.

14. Each Normalized_Purchase_Price has exactly one Purchase_Price; each Purchase_Price may have exactly one Normalized_Purchase_Price.

The normalized purchase price is calculated from the purchase price so there is only one value that can come from each purchase price. Depending on how fast we want to calculate the normalized purchase price, we might not calculate it for every purchase price.

15. Each Price_per_Acre has exactly one Acreage; each Acreage may have exactly one Price_per_Acre.

The price per acre is calculated from the acreage so there is only one value that can come from each acreage. Depending on how fast we want to calculate the price per acre, we might not calculate it for every acreage.

(5) Response Rate Use Case

16. Each Contact_Status may have zero to many ct_ids; each ct_id has exactly one Contact_Status.

When sorting into categories, the ct_ids have to have a contact_status but not all of the Contact_Statuses have to be used.

17. Each Responded has exactly one ct_id; each Responded has exactly one ct_id.

This is simply a Boolean flag that each row (uniquely labeled by the ct_id) has.

18. Each Contact_Status has exactly one Responded; each Responded has many Contact_Status'.

There is only one value (true or false) for each contact_status type. Responded doesn't tell us exactly which Contact_Status we are looking at because each value could be multiple different Contact_Status'.

(6) City Center Variation Use Case

19. Each ct_id has exactly one Center_of_Transaction; each Center_of_Transaction has one or more ct_id.

Each ct_id is completely unique but it is possible that the center of transaction is not unique. The center of transaction could be the same for two different transactions if drawn correctly. The center of transaction is calculated based on the polygons for each ct_id (and every ct_id has some sort of polygon) so every ct_id has a center of transaction.

20. Each ct_id has exactly one Distance_from_City; each Distance_from_City has one or more ct_id.

Each ct_id is completely unique but it is possible that the distance to city center is not unique. The distance from city could be the same for two different transactions if drawn correctly. The center of transaction is calculated based on the polygons for each ct_id (and every ct_id has some sort of polygon) so every ct_id has a center of transaction. This is used to calculate the ct_id distance from city so every ct_id has a distance from city.

21. Each Center_of_Transaction is related to exactly one City_Center; each City_Center may be related to zero or many Center_of_Transactions.

Unless dead center in between two cities (practically impossible), each transaction will be closer to one city than any other. A city may have no transactions closest to it or could have many.

22. Each Center_of_Transaction has one or more Distance_from_City; each Distance_from_City has one or more Center_of_Transaction.

Neither center of transaction nor distance from city are completely unique. The distance from city or the center of transaction could be the same for two different transactions if drawn correctly.

23. Each Distance_from_City is related to at least one City_Center; each City_Center may be related to zero or many Distance_from_City.

Distance from city is not completely unique. The distance from city or the center of transaction could be the same for two different transactions if drawn correctly. Therefore the same number could have multiple different cities that it is measuring to. The city center may have no transactions or many transactions measuring to it.

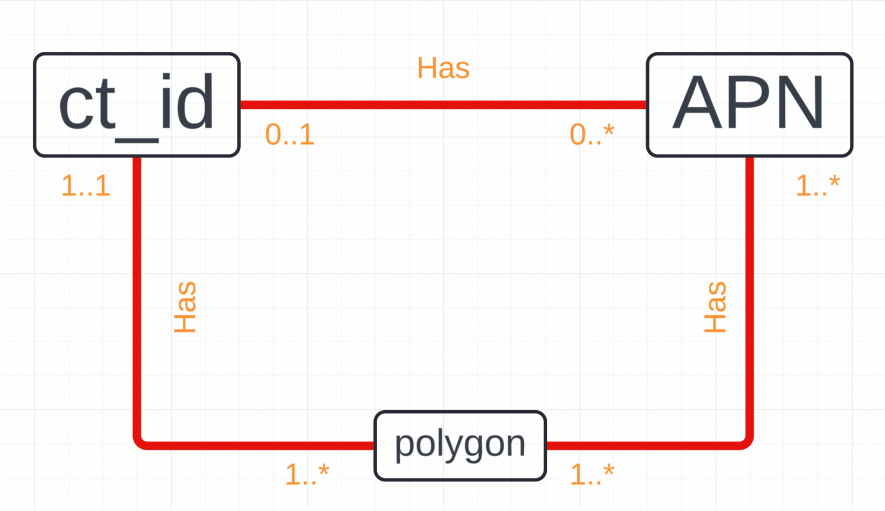
24. Each City_Center may be related to zero to many ct_ids; each ct_id is related to exactly one City_Center.

Unless dead center in between two cities (practically impossible), each transaction will be closer to one city than any other. A city may have no transactions closest to it or could have many.

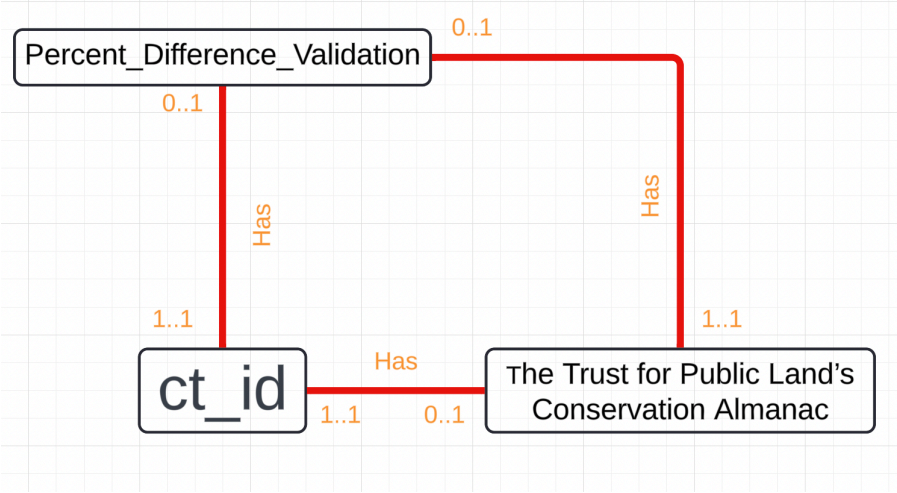
Conceptual Entity-Relationship Diagram

All explanations of the relationships are in the section above.

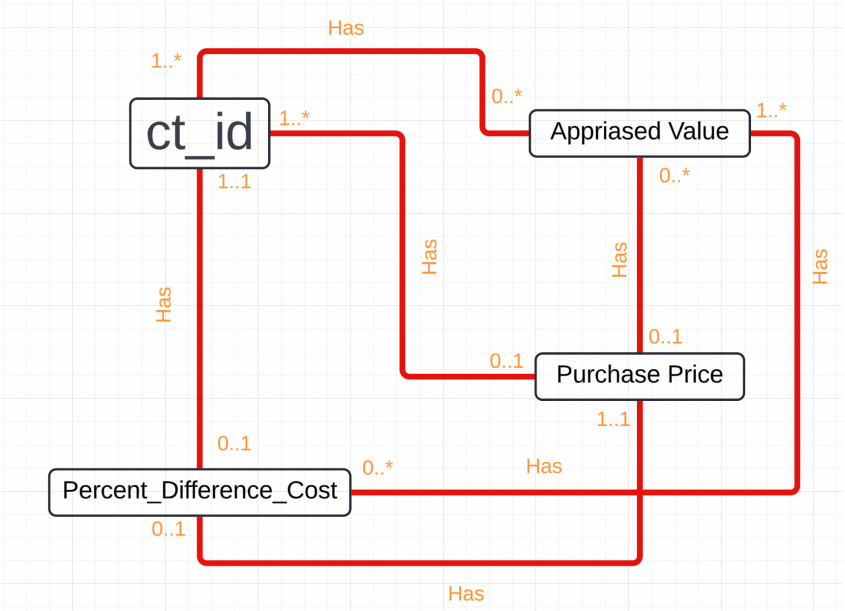
(1) Input Data Use Case



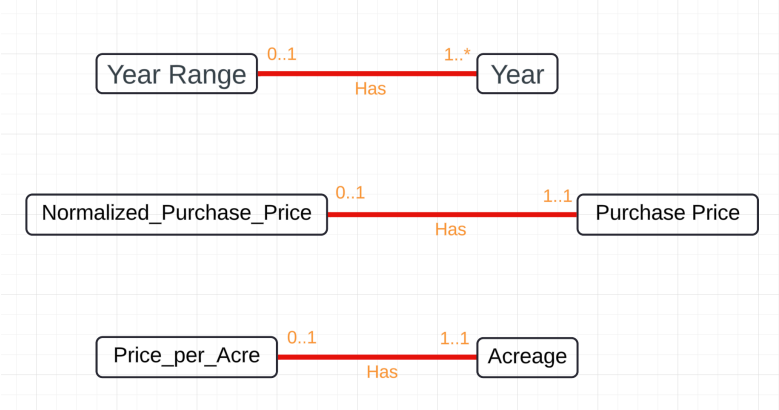
(2) Validate Data Use Case



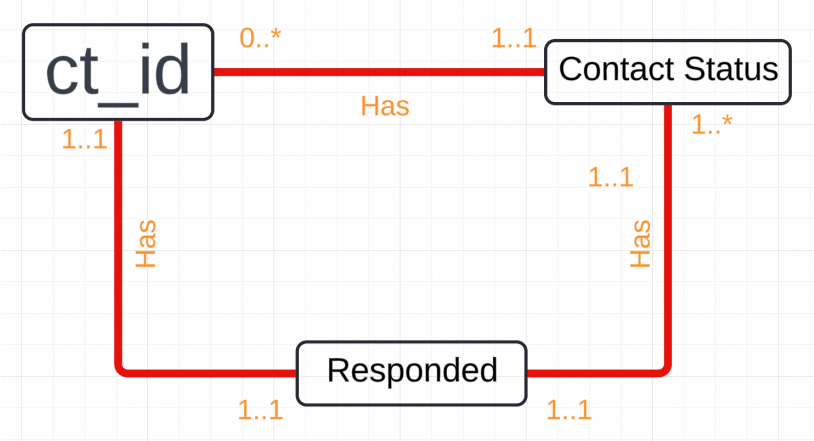
(3) Under/Overpaying Use Case



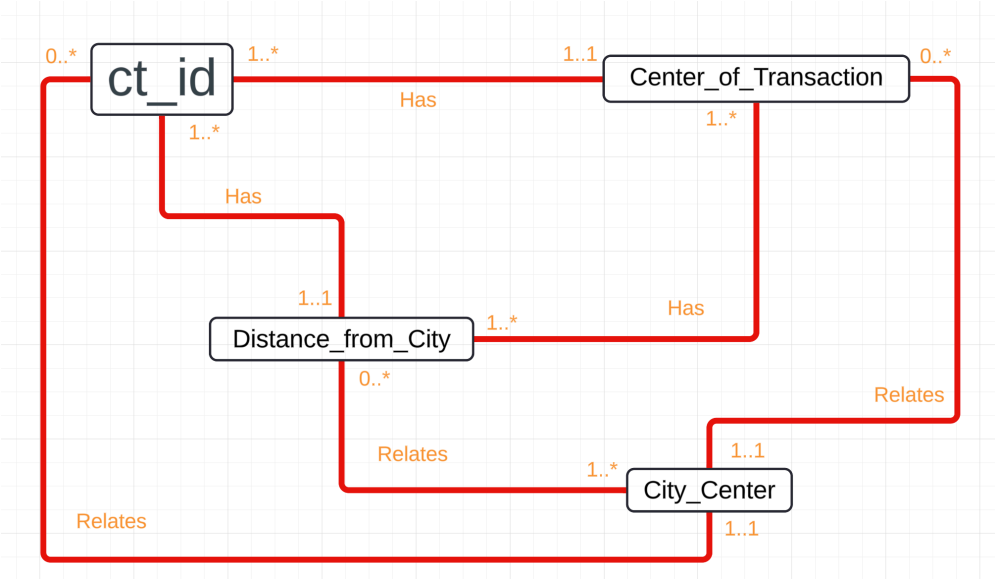
(4) Best Year Use Case



(5) Response Rate Use Case



(6) City Center Variation Use Case



Summary and Reflection

This database will be used to merge, visualize, compute, and analyze the data that is collected in the PLACES Lab at Boston University more efficiently. It will need to be able to merge and clean data very quickly and efficiently as that is the most common fault of working with data inputted by humans. Public data is notoriously dirty. There will also be a lot of working with GIS or spatial data. While possible to have use cases without a spatial component to the database, it would be nice to be able to include that.

My biggest question at the moment is: do all parts of the diagrams need to be connected to each other? If the relationship doesn't seem useful do we need to diagram it?