Name: Katherine Rein

**Table of Contents**

[Project Direction Overview 2](#_Toc169434084)

[Use Cases and Fields 2](#_Toc169434085)

[Structural Database Rules 7](#_Toc169434086)

[Conceptual Entity-Relationship Diagram 9](#_Toc169434087)

[Full DBMS Physical ERD 10](#_Toc169434088)

[Stored Procedure Execution and Explanations 11](#_Toc169434089)

[Question Identification and Explanations 13](#_Toc169434090)

[Query Executions and Explanations 13](#_Toc169434091)

[Index Identification and Creations 15](#_Toc169434092)

[History Table Demonstration 18](#_Toc169434093)

[Data Visualizations 19](#_Toc169434094)

[Summary and Reflection 20](#_Toc169434095)

# 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)
   1. Create additional rows or replace rows
   2. 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 (using either a bar chart, a histogram, a scatterplot, a line 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
   1. A negative value means the buyers paid less and a positive value means they paid more
3. Graph (using either a bar chart, a histogram, a scatterplot, a line 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 (using either a bar chart, a histogram, a scatterplot, a line 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
   1. A transaction either has a response or no response
2. Select a value to test
   1. Ex) Local/State/Federal, Size of Transactions, States
3. Sort transactions into different buckets of said value
4. Graph (using either a bar chart, a histogram, a scatterplot, a line 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.  A screenshot of a phone  Description automatically generated | 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. |
| 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
   1. 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. Each researcher may alter many transactions; each transaction may be altered by many researchers.

Both sides are optional plural as multiple researchers can be collaborating on all or a part of this project.

1. Each grantor may transfer many transactions; each transaction may be transferred by many grantors.

Transactions can be owned by many people and they can choose to keep these transactions or transfer them (but they don’t have to transfer them). When the transaction is transferred it can be done by the many people (and is typically required to be agreed upon by all of them).

1. Each grantor works with one to many grantees; each grantee works with one to many grantors.

For a grantor to be granting something, there must be a grantee so it is not optional. Like before, multiple grantees and grantors can own or purchase a transaction.

1. Each grantee may purchase many transactions; each transaction may be purchased by many grantees.

Transactions can be owned by many people and each person can own many transactions. However, a person doesn’t have to own a transaction.

1. Each researcher may input many validation data; each validation data may be input by one researcher.

Any researcher can input no or multiple validation data but if multiple researchers were to input the same validation datum, then we would have duplicate data and that would be an issue.

1. Each researcher may create many maps; each map may be created by many researchers.

As this is a collaborative project, the maps will be worked on by many people. However, every map that we think of doesn’t have to be created. Also, many researchers can make more than one maps.

1. Each researcher may create many graphs; each graph may be created by many researchers.

As this is a collaborative project, the maps will be worked on by many people. However, every map that we think of doesn’t have to be created. Also, many researchers can make more than one maps.

1. Each map may depict many transactions; each transaction may be depicted by multiple maps.

Most often we will be depicting more than one transaction. Not every transaction has to be depicted. Oftentimes the transactions will be on more than one of our maps.

1. Each graph may depict many transactions; each transaction may be depicted by multiple graphs.

Most often we will be depicting more than one transaction. Not every transaction has to be depicted. Oftentimes the transactions will be on more than one of our graphs.

1. Each researcher may advise one to many land conservation companies; each land conservation company may be advised by one to many land conservation companies.

The company may have a lot of advising that needs to be done and so they need more than one researcher. Some companies may be small so the researcher can advise more than one company.

1. A transaction is described by either a ct\_id, APN, polygon, or multiple of these.

If we don’t have any of these we can’t identify the transactions so it has to have at least one. Having multiple makes it easier to identify the transaction and can often get the others from each.

1. A graph is either a bar chart, a histogram, a scatterplot, a line graph, multiple of these, or none of these.

It’s possible to have multiple of these elements on one graph (however I rarely see this). There are also graph types that are less common that could be used.

# Conceptual Entity-Relationship Diagram

A diagram of a computer

Description automatically generated

# Full DBMS Physical ERD

A diagram of a flowchart

Description automatically generated

This follows 1NF as each transaction has a unique id so we don’t need to rely on row order for any information, everything has its own datatype with no mixing, each table has a primary key, and there is no way for repeating groups. The lack of repeating groups was fixed by having everything stem from the ct\_id which is completely unique (hard coded that way).

This follows 2NF because each non-key attribute in the table is dependent on the entire primary key. This follows 3NF/BCNF because each attribute in the table is dependent on the key, the whole key, and nothing but the key. These were achieved by splitting some of the attributes of the transaction table into side tables that reference attributes of transaction.

# Stored Procedure Execution and Explanations

A screenshot of a computer code

Description automatically generated

This adds city data manually while utilizing the cities\_seq. It takes the name of the city and the lat/long coordinates of the city.

A screenshot of a computer

Description automatically generated

This adds our researched data manually. There is only the ability to add one APN at the moment but more can be added on later on. The APN is not required but it can be added (same with the city).

A computer screen shot of a code

Description automatically generated

This correlates a researcher to a transaction. This could be done within the addourdata procedure but that felt like too much information to require.

A screenshot of a computer code

Description automatically generated

This is the data that we are wanting to check the accuracy of. We are able to add in the all of the data and the researcher here.

# Question Identification and Explanations

First Query: How many transactions and APNs has each researcher inputted that have at least a response?

Second Query: How many transactions have multiple APNs?

Third Query: Which transactions have over $200,000 of public spending?

# Query Executions and Explanations

First Query:

A screenshot of a computer

Description automatically generated

This counts the number of APNs and ct\_ids where the responded column is true. It then groups these all by the Researchers first name so that we can see each researchers progress.

Second Query:

A screenshot of a computer

Description automatically generated

This tells us which ct\_ids have multiple and how many APNs each has. It does so utilizing a having clause.

Third Query:

A screenshot of a computer

Description automatically generated

This looks for which ct\_ids have a public spending of more than $200,000. It then orders the data by spending and researcher first name.

# Index Identification and Creations

Primary Keys:

LandConservationCompany.LandConservationCompanyID

Researcher.ResearcherID

Contact\_Status.Contact\_Status\_ID

Cities.Cities\_ID

Transaction.ct\_id

Map.MapID

Graph.GraphID

Grantee.GranteeID

Grantor.GrantorID

Advise.AdviseID

ValidationData.ValidationDataID

Create2.Create2ID

Depict.DepictID

Alter.AlterID

Depict2.Depict2ID

Create.CreateID

Bar\_Chart.GraphID

Histogram.GraphID

Scatterplot.GraphID

Line\_Graph.GraphID

APN.APN\_id

Polygon.ct\_id

Purchase.PurchaseID

Valuation.Valuation\_id

WorksWith.WorksWithID

Transfer.TransferID

Foreign Keys:

|  |  |  |
| --- | --- | --- |
| Column | Unique? | Description |
| Transaction.Contact\_Status | Not unique | The foreign key in Transaction referencing Contact\_Status. Not unique because multiple transactions can have the same contact status. |
| Transaction.Closest\_City | Not unique | The foreign key in Transaction referencing Cities. Not unique because multiple transactions can be associated with the same city. |
| Advise.LandConservationCompanyID | Not unique | The foreign key in Advise referencing LandConservationCompany. Not unique because multiple advises can be related to the same company. |
| Advise.ResearcherID | Not unique | The foreign key in Advise referencing Researcher. Not unique because multiple advises can be associated with the same researcher. |
| ValidationData.ResearcherID | Not unique | The foreign key in ValidationData referencing Researcher. Not unique because multiple validation data entries can be associated with the same researcher. |
| ValidationData.ct\_id | Unique | The foreign key in ValidationData referencing Transaction. Unique because only one validation data entry can be related to the same transaction. |
| Create2.MapID | Not unique | The foreign key in Create2 referencing Map. Not unique because multiple create entries can be associated with the same map. |
| Create2.ResearcherID | Not unique | The foreign key in Create2 referencing Researcher. Not unique because multiple create entries can be associated with the same researcher. |
| Depict.MapID | Not unique | The foreign key in Depict referencing Map. Not unique because multiple depicts can be associated with the same map. |
| Depict.ct\_id | Not unique | The foreign key in Depict referencing Transaction. Not unique because multiple depicts can be related to the same transaction. |
| Alter.ct\_id | Not unique | The foreign key in Alter referencing Transaction. Not unique because multiple alterations can be related to the same transaction. |
| Alter.ResearcherID | Not unique | The foreign key in Alter referencing Researcher. Not unique because multiple alterations can be associated with the same researcher. |
| Depict2.GraphID | Not unique | The foreign key in Depict2 referencing Graph. Not unique because multiple depict2 entries can be associated with the same graph. |
| Depict2.ct\_id | Not unique | The foreign key in Depict2 referencing Transaction. Not unique because multiple depict2 entries can be related to the same transaction. |
| Create.GraphID | Not unique | The foreign key in Create referencing Graph. Not unique because multiple create entries can be associated with the same graph. |
| Create.ResearcherID | Not unique | The foreign key in Create referencing Researcher. Not unique because multiple create entries can be associated with the same researcher. |
| Bar\_Chart.GraphID | Unique | The foreign key in Bar\_Chart referencing Graph. Unique because each bar chart is associated with a specific graph. |
| Histogram.GraphID | Unique | The foreign key in Histogram referencing Graph. Unique because each histogram is associated with a specific graph. |
| Scatterplot.GraphID | Unique | The foreign key in Scatterplot referencing Graph. Unique because each scatterplot is associated with a specific graph. |
| Line\_Graph.GraphID | Unique | The foreign key in Line\_Graph referencing Graph. Unique because each line graph is associated with a specific graph. |
| APN.ct\_id | Not unique | The foreign key in APN referencing Transaction. Not unique because multiple APNs can be related to the same transaction. |
| Polygon.ct\_id | Unique | The foreign key in Polygon referencing Transaction. Unique because each polygon is associated with a specific transaction. |
| Purchase.GranteeID | Not unique | The foreign key in Purchase referencing Grantee. Not unique because multiple purchases can be associated with the same grantee. |
| Purchase.ct\_id | Unique | The foreign key in Purchase referencing Transaction. Unique because only one purchases can be related to the same transaction. |
| Valuation.ct\_id | Not unique | The foreign key in Valuation referencing Transaction. Not unique because multiple valuations can be related to the same transaction. |
| Valuation.PurchaseID | Not unique | The foreign key in Valuation referencing Purchase. Not unique because multiple valuations can be related to the same purchase. |
| WorksWith.GrantorID | Not unique | The foreign key in WorksWith referencing Grantor. Not unique because multiple works with entries can be associated with the same grantor. |
| WorksWith.GranteeID | Not unique | The foreign key in WorksWith referencing Grantee. Not unique because multiple works with entries can be associated with the same grantee. |
| Transfer.GrantorID | Not unique | The foreign key in Transfer referencing Grantor. Not unique because multiple transfers can be associated with the same grantor. |
| Transfer.ct\_id | Not unique | The foreign key in Transfer referencing Transaction. Not unique because multiple transfers can be related to the same transaction. |

Query:

"Transaction".public\_spending: This could be useful when trying to identify certain information about trasactions that are in a certain price range.

“Transaction".Acreage: This could be useful when trying to identify certain information about trasactions that are of a certain size.

"Transaction"."Date": This could be useful when trying to identify certain information about trasactions that are in a certain time frame.

# History Table Demonstration

The ResearcherChange table is used if we need to reallocate some of the transactions to a different researcher. This occurs when certain researchers leave, can work more hours, or need to switch up what they are focusing on.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

# Data Visualizations

To determine how companies and governments tend to spend their money when it comes to conservation work. This will help us understand where most of the money already goes when we want to talk to people about increasing funds.

When trying to predict how much a transaction might cost, it could be nice to be able to look at the trends of transactions of different sizes. Looking at the history of how much transactions cost would benefit our predictions.

# 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.

I don’t have any new questions at the moment.