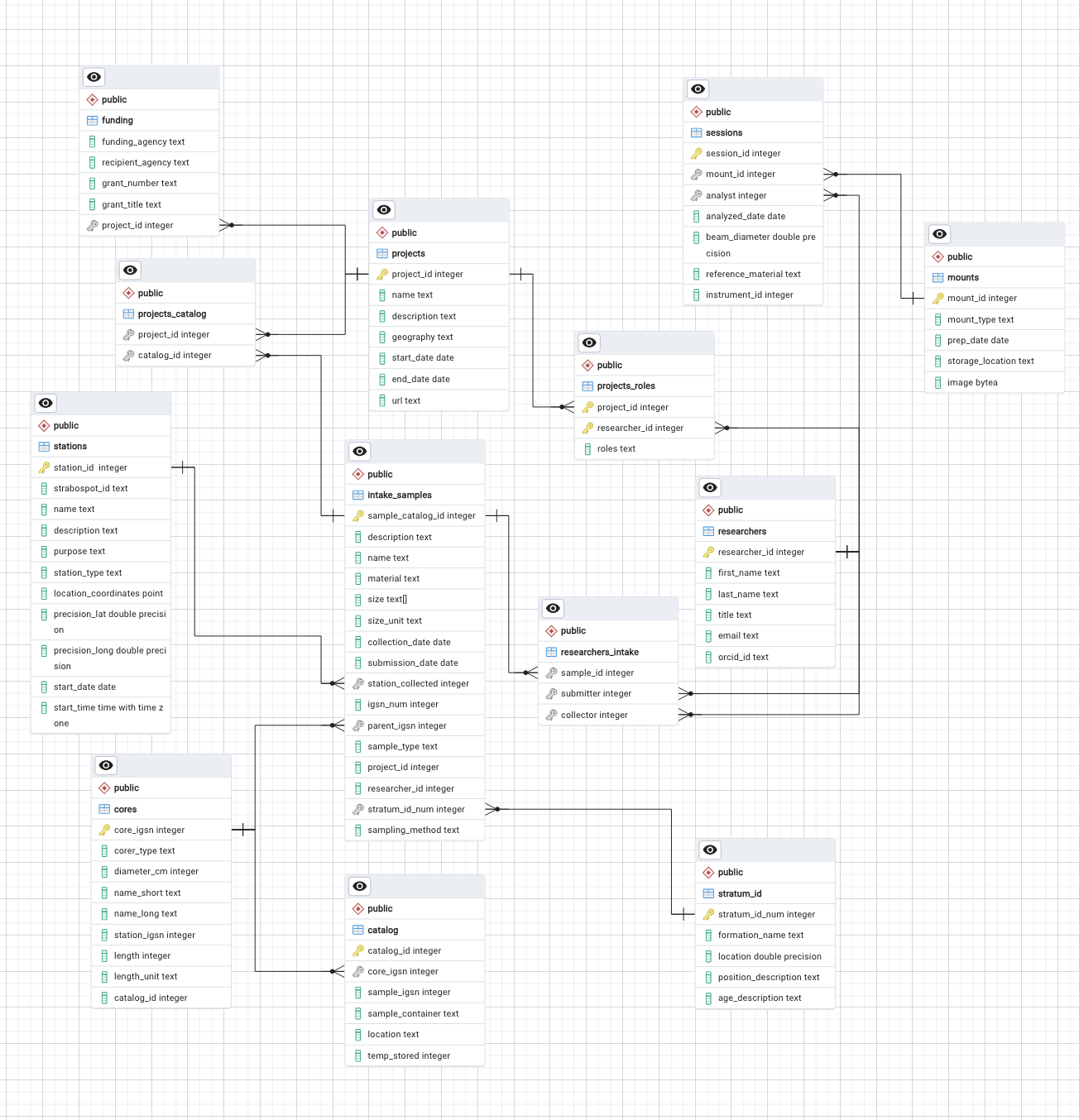
Foundation Tephra Schema

Prepared by Concord Undergraduate students based on spreadsheet outline by PI Kuehn

[**First Draft:**](#_50pa4b9i7m40) **2**

[**Second Draft:**](#_1n96pcx28fzs) **13**

# First Draft:



Notes:

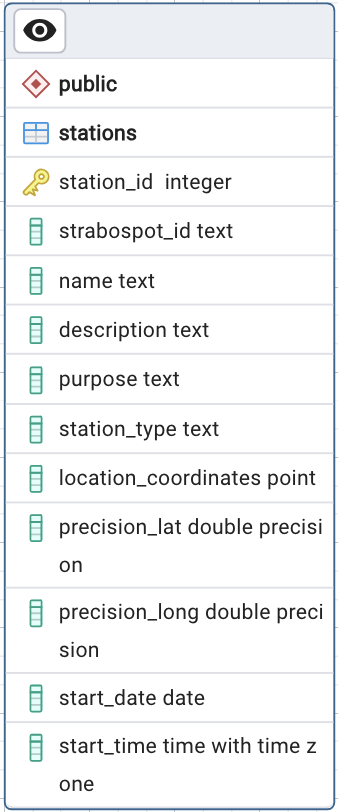
* This schema is meant to be the foundation of any future version of the tephra schema. The main focus topics are:
  + People
  + Projects
  + Stations
  + Samples - ingest
  + Catalog of stored objects (sample and their containers, locations)
  + Mounts/targets (with samples in them)
  + Funding/grants ( in project)

Revision Notes:

* + IGSN is alphanumeric (text)
    - Changed from an integer to a text datatype
  + Core drives, input samples, etc. could have images and/or documents - could have both fields with file links (list or array in field??)
    - Added a media table that connects to the catalog table, and holds all of the urls for images and documents connected to each core/drive/sample within the catalog
  + Funder’s grant ID (text), URL
    - Changed the grant\_id data type to text, and added a column for a url
  + URLs - possibly many, List or Array? - Could be useful to have the URL associated with something about what each URL is for - like a file, a website, a picture
    - Within the new media table, there is a column for a description of the url and a column for the type (file, image, website, etc.)
  + How best to handle access permissions? - public, admin, specific people or projects, embargo date, on/off switch - account-based with a “public” user?
  + Current institution. People can change institutions - How to handle a history - Start, End dates - Text comment field?
    - Added a history table that holds the start and end dates of past institutions a researcher has been a part of. For the relationship to work, another table just for institutions was created. It holds all of the possible institutions the researcher could be from.
  + Mounts - physical object in inventory, also context for chemistry data; Contains specific sample fractions which connect to the sample inventory (foreign key to specific container ID), also foreign key to sample; Most mounts have four positions, but there are other types of mounts too
    - Created a foreign key inside the mounts table that connects to the sample\_catalog\_id in the intake\_sample table.
  + Sample catalog IDs - CU\*\*\*\*, UA\*\*\*\*\*, C\*\*\*\*\*\* - So alphanumeric (text)
    - Changed the datatype to text in the sample, core, catalog and drive tables.
  + Cores have stations
    - Created a stations column and made it a foreign key to connect to the stations table
  + Core drives table, child of station, samples can be child of core drive; Projects can have cores, drives just like they have samples; Cores, drives have projects the other way too
    - Created a drives table, and changed the relationships so that sample, core, and drives are all connected to stations, catalog, and each other.
  + Stations, cores, drives, samples can all have an IGSN; Can have hierarchy of Station->Core->Drive->Sample or Station->Sample or sometimes just sample with no parent(s)
  + Intake samples - Mostly use mass in grams as unit for amount, sometimes have volume in cubic centimeters (CC), some have both measurements to preserve, also must support wet versus dry
    - Created a column column along with the mass column. There are unit columns for each. Also added a state column which will describe if it is wet or dry
  + Catalog (inventory) also has mass in each container
    - Added a mass column and unit column
  + Mass in container is usually the original, but could change; Can also send some of a sample to another lab, nice to record that
    - Created a sample\_history table that the user can add entries for each time an amount of a sample is removed from its original container. This table to connect to the catalog table
  + Associate sample possibly with a specific volcano and specific eruption - Volcano name, Eruption name, Smithsonian GVP volcano and event IDs - Put in stratum. Sample depth - top, bottom, thickness - Put in stratum
    - Added more specific columns into the stratum table

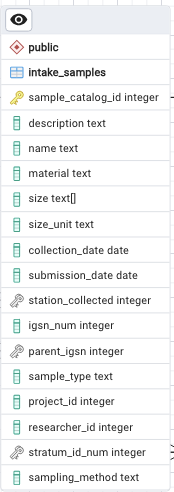
Tables:

“stations”:



* This table holds the information of the station where the samples are collected.
* Primary Key: for now is the station\_id which is a generated unique value as the stations are entered. If every station handling samples has a Strabospot ID, then the Strabospot ID column can be made as the primary key.
* Columns “name”, “purpose” and “description” hold the more specific information needed to help identify each station. The datatype is text to make this flexible for the user to enter.
* As of now the “station\_type” column is using a text datatype. Looking down the road this could potentially be a drop down that the user can pick between different types. This creates a more cohesive table.
* Instead of needing separate columns for latitude and longitude, there only needs to be one column “location\_coordinates''. This column uses the point data type which stores a pair of float decimal values. This will hold the latitude and longitude in decimal form.
* “precision\_lat” and “precision\_long” use the double precision datatype. This allows the column to hold a wide range of decimal values that should account for typical precision values used.
* “start\_date” holds the day that the station was created, while “start\_time” holds the specific time the station was created. This time also includes the time zone just to provide a more specific description of when it was truly created.

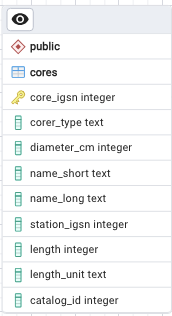
“intake\_samples”:



The idea is for this table to be seen as the main table of the database. This is where the most used information will be kept, and where the most information will be inputted.

* Primary Key: “sample\_catalog\_id” will hold the unique ID ex. “CU1429” of each sample since samples can have several names at once. This ID is the most convenient way to identify the sample compared to the other names. This will also make sure this table connects to the catalog table.
* “description” and “material” are text data type columns used to provide more specific attributes of the sample
* “size” is an integer and there is an added column “size\_unit” to have the unit entered.
* “collection\_date” and “submission\_date” will be used based on which is applicable to the sample. But this way we have the date it was entered into the system either way.
* For now, the “name” column is there to provide the user a way to provide a descriptive name of the sample that is not a numerical ID
* “isgn\_num” is the IGSN number. The integer data type will hold enough numbers to input the whole IGSN number.
* “sample\_type” and “sampling\_method” are descriptive columns with a text datatype
* Included the project\_id so that at a first glance you can easily identify the project the sample is a part of. The datatype is text so several projects can be listed here.
* Foreign Keys:
  + station\_collected: connects each sample to which station it was collected from inside the “stations” table.
  + parent\_igsn: this is the IGSN number of the core which the sample was taken from if applicable. This connects the table to the “cores” table
  + stratum\_id\_num: the unique id used for the stratum of the sample. This connects the table to the “stratum\_id” table that goes into more depth of the id

“cores”:



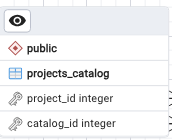
This table provides detailed information regarding the core that a sample possibly came from. This will help us keep track of how big the core was to start out, and where the inputted samples come from.

“catalog”:

This table is meant to act as an inventory or “catalog” of all inputted cores and samples in the database.

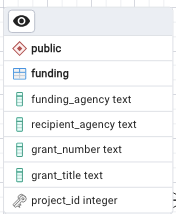
* Primary Key: A unique generated catalog\_id
* Columns in this table include the type of container the sample/core is in and the location of said container. Either the sample\_igsn or core\_igsn number will be inputted depending on what is applicable.
* Foreign Key/s:
  + For now, only “core\_igsn” is a foreign key to connect the table to the cores table. It could be more efficient to create an “igsn” reference table that holds both sample and core igsn numbers then use that to relate instead. Not quite sure yet.

“projects\_catalog”:



This is a reference table that holds the “project\_id” and “catalog\_id”. This is used for relations because a sample can be apart of several projects and a project can have several

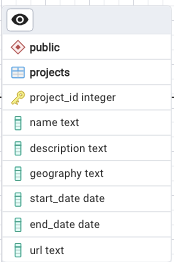
“funding”:



This table holds the information about the funding behind the different projects.

* Foreign Key:
  + “project\_id” connects to the reference table which then connects to the “projects” table

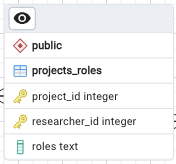
“projects”:



This is the main table for all the information regarding specific projects.

* Primary Key: “project\_id” which is a unique integer.
* The rest of the columns are descriptive information to give the user a broad picture of what each project is meant for. This includes start and end date, name, description, geography, and the url.

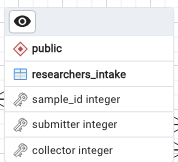
“project\_roles”:



This table is meant to showcase the roles each researcher plays in each project

* Primary Keys: “project\_id” and “researcher\_id” are both primary keys
* The column “roles” details what roles is being achieved

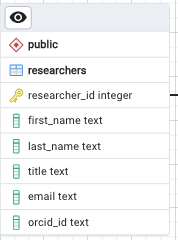
“researchers\_intake”:



This is a reference table used to establish the relationship between the sample, and the analyst that submitted and/or collected it.

* The sample\_id is a foreign key connected to the “intake\_samples” table, then “submitter” and “collector” are foreign keys that connect to the “researchers” table.

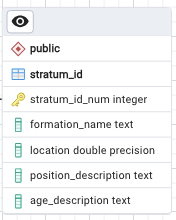
“researchers”:



Holds the information about any researcher inside the database.

* Primary Key: “researcher\_id” which is a generated unique number
* Includes first and last name, their title, contact email, and orcid\_id

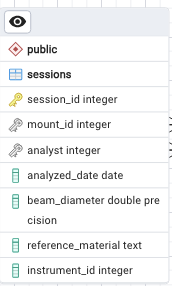
“stratum\_id”:



This table is meant to hold the information that makes up the stratum id of a collected sample.

* Primary Key: a generated unique integer “stratum\_id”
* Includes the formation name, location coordinates(double precision datatype), position text description, and the age description

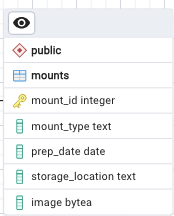
“sessions”:



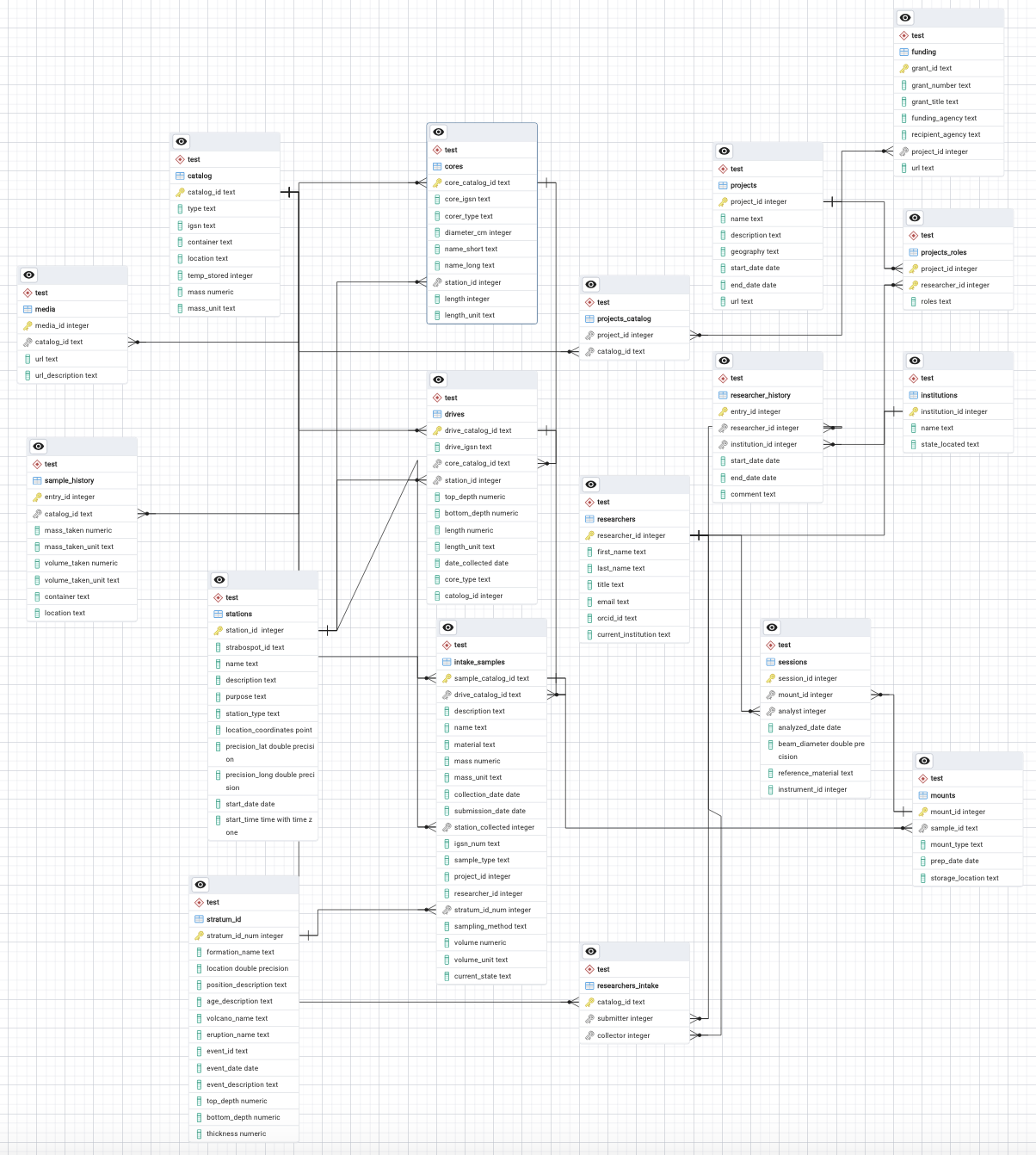
This table holds the information about a session in which the microprobe is used.

* Primary Key: session\_id which will be a unique generated integer
* Foreign Keys: “mount\_id” connects to the “mounts” table, and “analyst” connects to the “project\_roles” table.

“mounts”:



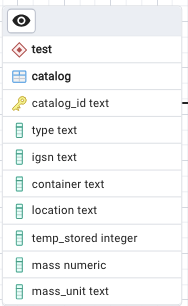
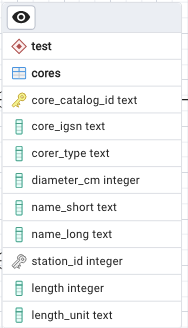
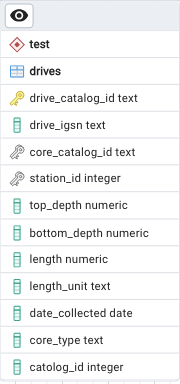
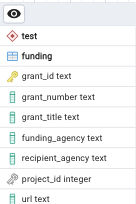
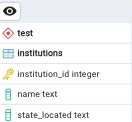
# Second Draft:



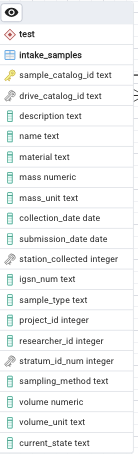
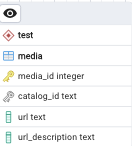
Notes:

* This is the second draft of the database. Most revisions came from the notes above from the first draft, and some reorganization took place as well.
* Instead of specific samples being the main table, the three tables intake\_samples, cores, and drives are the focus here.
* The catalog table serves a better inventory role, as it will hold the information of everything physically kept in storage
* Columns can easily be switched between “null” and “not null” values.

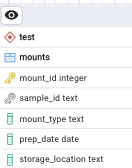
Tables

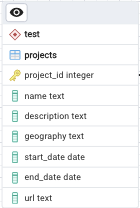
* “catalog”
* 
* The catalog table acts as an overall inventory for samples, cores, and drives. The primary key is the catalog\_id “CU23…” which is a text field. The mass column holds the decimal value while the mass\_unit column holds the unit abbreviation. Location is referring to the location in which the container has been stored.
* “cores”
* 
* The cores table holds all specific information that is needed for different cores in the system. The primary key is the core\_catalog\_id, which also acts as a foreign key to the catalog table. The other foreign key is the station\_id which creates a relationship to the station table. The diameter\_cm column is an integer value that will be assumed to be in cm.
* “drives”
* 
* The drives table holds the specific information on the different drives. The primary key is also the drive\_catalog\_id “CU45…” which connects to the catalog table. The core\_catalog\_id acts as a foreign key to demonstrate the relationship between the core and the drive. The second foreign key is the “station\_id” which connects it to the specific station in case there is no known core for the drive and we still want to know the station it came from.
* “funding”
* 
* The funding table holds all information on specific funding that is used for all of the different ongoing projects. The primary key is the grant\_id which will be an auto-generated value. The foreign key project\_id connects this table to the projects table.
* “institutions”
* 

This table holds the basic information of the different institutions researchers could potentially reside. The primary key is the institution\_id and that is a auto-generated value. It connects to the researcher table.

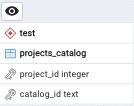
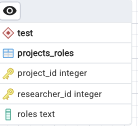
* “intake\_samples”
* 
* This table holds any information that will be needed on the different samples in the database. The primary key is the sample\_catalog\_id “CU34…” which establishes the connection to the catalog table. There are three foreign keys. Drive\_catalog connects the sample to the parent drive, station\_collected connects it to the station it was collected in, and stratum\_id\_num creates a conection to the stratum details in the stratum table. Both mass and volume are in this table with the intent that one of the other can be used here. Current\_state is meant to hold if the sample is dry or wet.
* “media”
* 

This table holds all the urls that will send the user to any documents attached to a sample, drive, or core. The primary key is an auto-generated value, while the foreign key “catalog\_id” connects this table to the catalog. There is also a column for a url description so the user can describe what type of file the url is for.

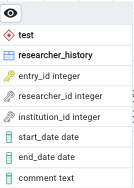
* “mounts”
* 
* The primary key mount\_id is an auto-generated value, and the sample\_id foreign key connects the mount to the used sample.

* “projects”
* 

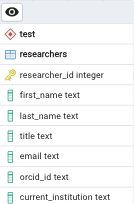
Projects holds the basic information regarding the different projects that the samples are being used in. The primary key here is an auto-generated value that will act as the project id.

* “projects\_catalog”
* 
* This is a reference table which holds the project ids and catalog ids as both foreign keys so that several projects can be connected to several samples, cores, or drives at the same time.
* “project\_roles”
* 

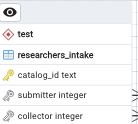
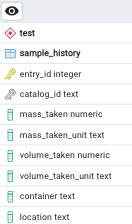
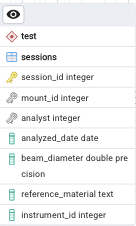
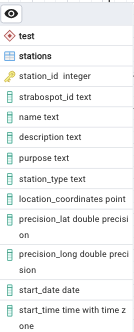
This table is meant to establish a relationship between the researchers and the different roles they have on different projects. This is done by using two primary keys, and the roles column is there to provide a description of their current role.

* “researcher\_history”
* 

This table holds the different institutions a researcher has been apart of over the years, since people can move to different places quite a bit. The primary key is an auto-generated value, by entry it will increment. The researcher\_id and institution\_id act as foreign keys to connect the researcher and institution tables. With each entry a start and end date will need to be entered. This gives the flexibility to enter the same researcher several times if they move institution more than once.

* “researchers”
* 

This holds the basic information on the researchers. The primary key is their researcher\_id which will be an auto-generated value.

* “researchers\_intake”
* 
* This table acts as a reference table in order to differentiate if a researcher was a submitter or a collector on a given sample. The primary key is the catalog\_id which connects this table straight to the main inventory.
* “sample\_history”
* 
* This table is meant to document when a sample is split into different places and possibly sent to a different lab. The primary key is the entry\_id, which will auto-generate with each entry. This way the same sample can have several entries if it is split into lots of different parts. This will help keep track of the different locations of specific samples. The foreign key catalog\_id connects each entry to the entered sample/core/drive to the main catalog.
* “sessions”
* 
* The primary key here is the session\_id, with the mount\_id and analyst\_id acting as foreign keys to connect the mount used and the analyst who preformed the session.
* “stations”
* 
* This table holds all information needed to be known on any entered station. This includes the strabospot\_id, name, description, coordinates, and the start date/time. The primary key is an auto-generated value since every station may or may not have a strabospot\_id.
* “stratum\_id”
* 

This table holds more in-depth information on a given sample. The primary key is the stratum\_id which connects this table to the intake\_sample table.