



HERON Training Manual

Center for Medical Informatics and Enterprise Analytics
University of Kansas Medical Center

3901 Rainbow Blvd.
Kansas City, KS 66160

p. 913-988-7688

Heron-admin@kumc.edu
<https://heron.kumc.edu/heron>

Contents

I.	General HERON Information	4
	HERON Overview.....	4
	Reasons to use HERON	4
	De-Identification Process	4
II.	Logging into HERON	5
	Credentials.....	5
III.	Hierarchical Structure	6
IV.	Running a Query / User Interface.....	7
	Navigate Terms	7
	Find	7
	Running a Query.....	8
	Previous Queries	8
	Workplace	9
V.	Searching Techniques	10
	Logical Condition “OR”	10
	Logical Condition “AND”	10
	Using both logical conditions “OR” and “AND”.....	11
	Exclusion.....	11
	Treat Groups Independently.....	11
	Same Financial Encounter.....	11
	Independent AND Same Financial Encounter.....	12
	Date Ranges	12
	Items Occur More than Once	13
VI.	Finalizing Queries	14
	Cohort Formation.....	14
	Finalizing Query with Shopping Cart of variables.....	14
VII.	Requesting Data.....	15
	Identified Data Request	15
	De-Identified Data Request	15
	Submitting a Data Request.....	15
	Data Request Review.....	15
VIII.	Receiving Data	16
	REDCap.....	16

	Raw Data.....	17
	Patient Table.....	17
	Data Table.....	18
	Code-Info.....	20
	Variable.....	21
IX.	Uploading Data into SQLite	22
	Retrieve Data & Database Schema:	22
	Upload De-identified Database Schema:	22
	Upload Identified Database Schema:.....	22
	Upload Data into Database:.....	22
X.	Basic Data Manipulation Techniques in SQL	23
	Browsing Data in SQLite:.....	23
	Basic SQL Queries:.....	23
	Joining Tables in SQLite:.....	24
	Tonsillectomy Example	24
XI.	Data in HERON	28
	Alerts	28
	Allergy.....	28
	Cancer Cases (Lay-Person Hierarchy)	28
	Cancer Cases (NAACCR Hierarchy).....	28
	Cardiology Lab Results	28
	Cardiovascular Registry (NCDR)	28
	Cystic Fibrosis Registry.....	29
	Demographics	29
	Diagnoses	30
	Flowsheets.....	32
	History	32
	Laboratory Tests (KUH Hierarchy).....	33
	Laboratory Tests (LOINC Hierarchy).....	33
	Medications.....	33
	Microbiology Negative Results.....	34
	Microbiology Positive Results.....	34
	Microbiology.....	34
	Microbiology tests that require growth monitoring. These may be positive or negative results.....	34
	Orders	34
	Procedures.....	34
	REDCap.....	35

Research Enrollment.....	35
Specimens.....	35
Fluid and tissue specimen information is pulled from the KUMC Biospecimen Repository.....	35
Trauma Registry (Draft).....	35
Includes information entered into the national trauma registry (NTDS) from KUMC	35
Visit Details.....	35
Provides information regarding each inpatient and outpatient visit including patient vitals	35
Visit Notes	36
Vizient (formerly UHC).....	36

General HERON Information

HERON Overview

HERON (Healthcare Enterprise Repository for Ontological Narration) is a search discovery tool that allows you to search de-identified data from various hospital and medical center sources that include but are not limited to Epic/O2 (the hospital electronic medical record), IDX (the clinical billing system), KU Hospital Cancer Registry, KU Biospecimen Repository, REDCap (selected projects), Social Security Death Index, and University HealthSystem Consortium (Quality Measure Data). By combining the various data sources, researchers can look at the data in new ways that are not available when viewing data one source at a time.

Reasons to use HERON

1. Request de-identified datasets which do not require IRB approval
2. Request identified data to find a cohort of patients who meet inclusion/exclusion criteria and receive any additional variables of interest.
 - o Will require IRB approval or QI letter of determination from the IRB
3. Receive data on patients in a research study
 - o Please contact heron-admin@kumc.edu to learn how to get data elements on a specific list of MRNs
4. Receive contact information for patients in FRONTIERS registry who are eligible for your study

De-Identification Process

HERON does not include any identifying information, such as a Medical Record Number (MRN), name, address, or birth date. In an effort to further de-identify the data, we shift dates by 1-365 days prior to the actual date. In the example below, you will see two example patients. The date offset remains consistent for a patient. However, note that patient #1 and patient #2 have different date offsets.

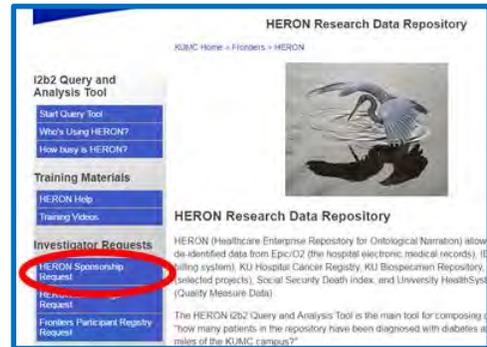
Date Shifting Example

Patient #	Variable	Real Date	Date Offset	HERON Date
1	Date of Birth	1/1/1950	-60 days	11/3/1949
1	Asthma Diagnosis	4/1/1965	-60 days	2/1/1965
2	Date of Birth	5/21/1991	-20 days	5/1/1991
2	Hospital Admission	6/30/2015	-20 days	6/10/2015

Logging into HERON

Credentials

1. Faculty or sponsored by a faculty member
 - To be sponsored by a faculty member have the faculty member proceed to <https://heron.kumc.edu/heron/> and click on HERON sponsorship request
2. CITI Training must be current:
 - If you have completed CITI training at an outside institution, please email your transcript to heron-admin@kumc.edu
3. Sign System Use Agreement



Once you have completed all of the requirements you should see 3 check marks on the right hand side of <https://heron.kumc.edu/heron/>. The check marks indicate that you have access to HERON. To open HERON click on “Start Query Tool” on the left hand side.



Hierarchical Structure

HERON is organized with a hierarchical structure. You are able to search very broad categories or very specific terms when constructing a query. In the example below you will see the contents for the base folder demographics.

1. Base Folders

- Open up specific folders to find what is inside
- Demographics is one of the base folders. The base folders are too broad to pull over into the query, because they will return too much data. Open up the folders to find specific contents within the folder.

2. Demographic Folder

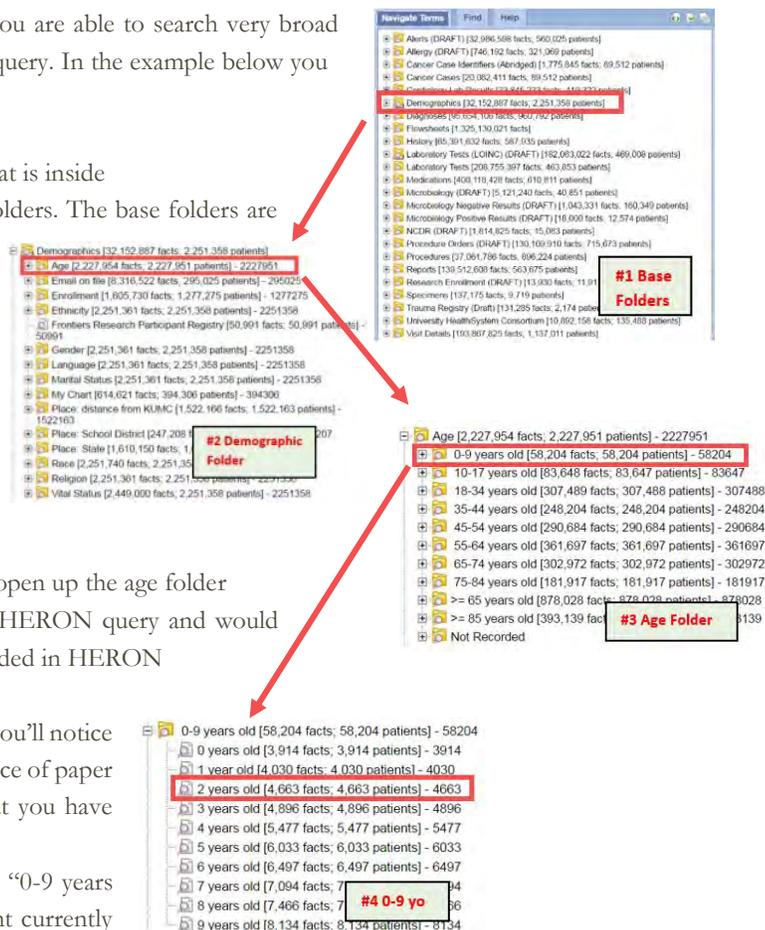
- Open up this folder to find many different demographic terms

3. Age Folder

- Within demographic folder we can open up the age folder
- This folder could be pulled into a HERON query and would return anyone who has an age recorded in HERON

4. 0-9 Years old

- Within this folder are specific ages, you'll notice that next to each age the icon is a piece of paper instead of a folder. This means that you have drilled down as specific as possible.
- You can pull over the whole folder "0-9 years old" which would return any patient currently between 0-9 years of age. Alternatively, pull over a specific age, i.e. 2 years old.

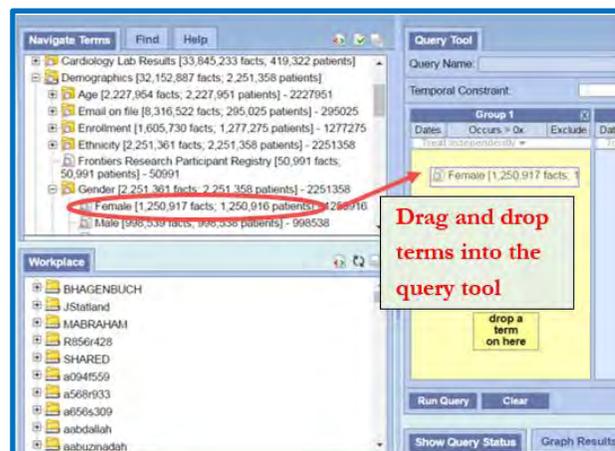


Running a Query / User Interface

The basic idea behind HERON is to pull several terms or folders (groups of terms) into the query tool to find patients who meet your eligibility criteria. Once you have found patients who meet your eligibility criteria you may add additional variables that you are interested in reviewing.

Navigate Terms

Terms are organized in the navigate terms hierarchy on the left hand side. See “Data in HERON” section to learn what is in each folder. When you find a term that you are interested in searching click on the term and drag it into the query tool.

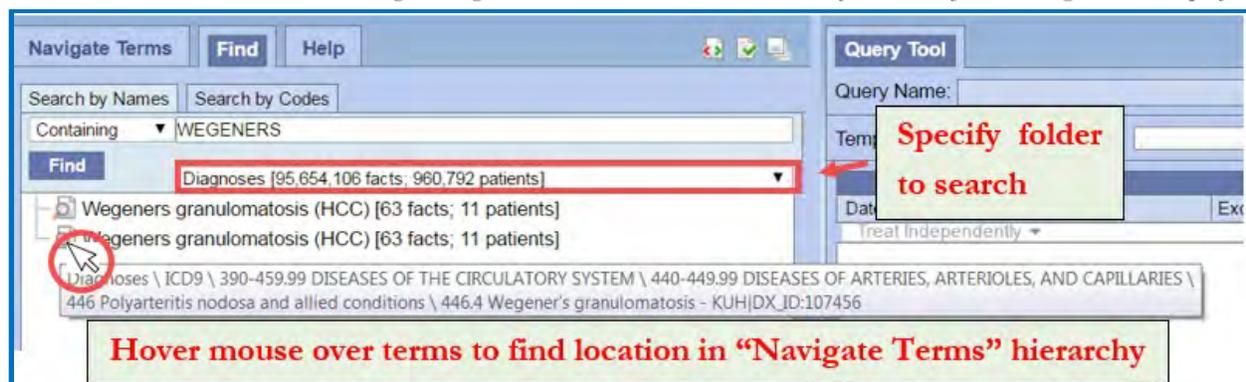


Find

To find terms within HERON you can take advantage of the find feature. If you click on find, you may search by names or by codes.

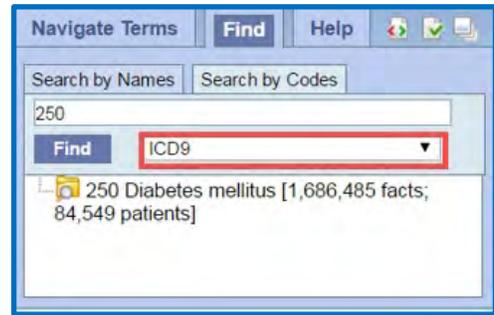
To search by names:

1. Click on the “Search by Names” tab found within Find
2. “Containing” is the default, but you may also choose exact, starting with, or ending with from the drop down menu
3. Type in name of term you are looking for, i.e. Wegeners
4. Limit your search to a specific folder, i.e. medications, diagnoses, etc.
5. Click find
6. Hover over each term to find where they are in the navigate terms hierarchy. Once you find where the term is in the hierarchy pull it into your search. It is typically better to go back into navigate terms rather than pull the name from the search, because sometimes you will find similar items that you are interested in nearby in the hierarchy.
 - o In the example below the folders you would open up under Navigate Terms would be as follows:
 1. Diagnoses
 2. ICD9
 3. 390-459.99 Diseases of the Circulatory System
 4. 440-449.99 Diseases of the Arteries, Arterioles, and Capillaries
 5. 446 Polyarteritis nodosa and allied conditions
 6. 446.4 Wegener’s granulomatosis → *Pull over this whole folder unless you want to get even more specific*



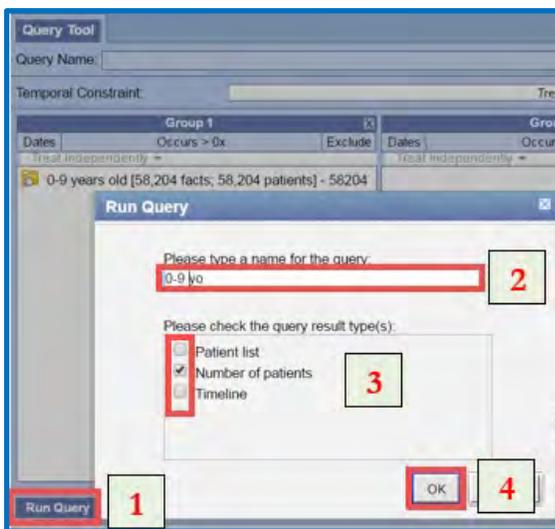
To search by codes:

1. Click on the “Search by Codes” tab found within Find
2. Type in the code you are searching for (i.e. 250 which is the ICD 9 code for diabetes)
3. Limit your search to a specific code type (i.e. ICD 9)
 - o The most common folders to search are ICD9, ICD10, and CPT
4. Click find
5. Pull the item into your query
 - o Alternatively, you may hover over each term to find where they are in the navigate terms hierarchy. Once you find where the term is in the hierarchy pull it into your search.



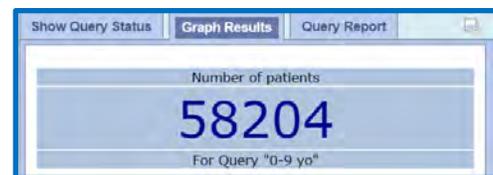
Running a Query

Once you have pulled terms into the query tool you can run the query.

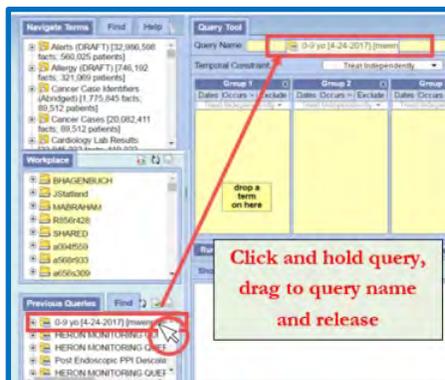


1. Click on “Run Query”
2. Type a query name or leave the pre-populated query name
3. Check what you would like to run
 - a. Patient List – saves the patient numbers in the query. This is only needed if you would like to use the demographic or timeline tools. Otherwise, leave this unchecked
 - b. Number of Patients – **always check this**
 - c. Timeline – check this to see a plot of all patients who meet your search criteria. For each patient, a tick mark will appear for every time they have a fact recorded about a term in your search.
4. Click OK

Once your query finishes, you will be able to see the results at the bottom half of the screen. For this query, we see that there are 58,204 patients who are 0-9 years old.



Previous Queries

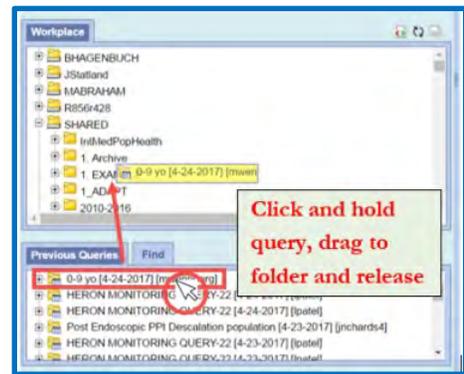


After your query runs, you can find it in the previous queries section in the bottom left section of HERON. You can click and drag the query to the query tool to pull up any previous queries.

If you cannot find your previous query, click on the button with the green check mark () next to previous queries and increase the maximum number of queries to display from 20 to 200 (or 2,000). The number you have to increase it dependent on how many HERON queries you have run! 😊

Workplace

The workplace will contain your individual folder and the shared folder. You are able to drop a previous query into your folder to find it more easily in the future or into a folder within the shared folder. When adding to the shared folder any other HERON user will have access to the query. Right click on “SHARED” to add another folder specific to your project or with your username (i.e. EXAMPLE). The picture on the right shows how to move a previous query to the EXAMPLE folder found within the shared folder.



Searching Techniques

In the next several sections you will learn the basics of how to search HERON. While a lot of the examples show you how to narrow by demographics, the same concepts apply for finding patients with specific diagnoses or who have been prescribed certain medications.

Logical Condition “OR”

All terms within one group have the logic of “or” For example listing both “female” and “Asian” in group 1, means that patients can be Female AND/OR Asian. This could include females who are Caucasian and males who are Asian. Using an example:

- Number of patients who are Female: 1,250,916
- Number of patients who are Asian: 13,866
- Number of patients who are females and/or Asian: 1,256,917

The screenshot shows the Query Tool interface with the Query Name set to "Female". The Temporal Constraint is "Treat all groups independently". Under Group 1, the search criteria are "Female [1,250,917 facts; 1,250,916 patients] - 1250916". A green box highlights the text "one or more of these" and a yellow box highlights "drop a term on here". The "Run Query" button is visible. Below the interface, a large blue box displays the result: "Number of patients 1250916 For Query 'Female'".

The screenshot shows the Query Tool interface with the Query Name set to "Asian". The Temporal Constraint is "Treat all groups independently". Under Group 1, the search criteria are "Asian [13,866 facts; 13,866 patients] - 13866". A green box highlights the text "one or more of these" and a yellow box highlights "drop a term on here". The "Run Query" button is visible. Below the interface, a large blue box displays the result: "Number of patients 13866 For Query 'Asian'".

The screenshot shows the Query Tool interface with the Query Name set to "Female or Asian". The Temporal Constraint is "Treat all groups independently". Under Group 1, the search criteria are "Female [1,250,917 facts; 1,250,916 patients] - 1250916". Under Group 2, the search criteria are "Asian [13,866 facts; 13,866 patients] - 13866". A green box highlights the text "one or more of these" for Group 1 and another green box highlights "one or more of these" for Group 2. A yellow box highlights "drop a term on here". The "Run Query" button is visible. Below the interface, a large blue box displays the result: "Number of patients 1256917 For Query 'Female or Asian'".

Logical Condition “AND”

Between groups there is the logical condition “and.” For example, “female” in group 1 and “Asian” in group 2, means that patients must be both female and Asian.

- Number of patients who are Female: 1,250,916
- Number of patients who are Asian: 13,866
- Number of patients who are Female and Asian: 7,865

The screenshot shows the Query Tool interface with the Query Name set to "Female". The Temporal Constraint is "Treat all groups independently". Under Group 1, the search criteria are "Female [1,250,917 facts; 1,250,916 patients] - 1250916". A green box highlights the text "one or more of these" and a yellow box highlights "drop a term on here". The "Run Query" button is visible. Below the interface, a large blue box displays the result: "Number of patients 1250916 For Query 'Female'".

The screenshot shows the Query Tool interface with the Query Name set to "Asian". The Temporal Constraint is "Treat all groups independently". Under Group 1, the search criteria are "Asian [13,866 facts; 13,866 patients] - 13866". A green box highlights the text "one or more of these" and a yellow box highlights "drop a term on here". The "Run Query" button is visible. Below the interface, a large blue box displays the result: "Number of patients 13866 For Query 'Asian'".

The screenshot shows the Query Tool interface with the Query Name set to "Female and Asian". The Temporal Constraint is "Treat all groups independently". Under Group 1, the search criteria are "Female [1,250,917 facts; 1,250,916 patients] - 1250916". Under Group 2, the search criteria are "Asian [13,866 facts; 13,866 patients] - 13866". A green box highlights the text "one or more of these" for Group 1 and another green box highlights "one or more of these" for Group 2. A yellow box highlights "drop a term on here". The "Run Query" button is visible. Below the interface, a large blue box displays the result: "Number of patients 7865 For Query 'Female and Asian'".

Using both logical conditions “OR” and “AND”

Between groups there is the logical condition “and” and within a group there is a logical condition “or.” We have found that there are 1,256,917 patients who are female or Asian. If we are only interested in patients who are 0-9 years old who are female or Asian we can use both “or” and “and” to create the search. Group 1 is “Female” or “Asian” and Group 2 is the age 0-9.

- Number of Patients who are female and/or asian: 1,256,917
- Number of Patients who are 0-9 years old: 58,204
- Number of Patients who are 0-9 years old AND female and/or Asian: 27,729

The first screenshot shows a query named 'Female or Asian' with two groups. Group 1 contains 'Female' (1,250,917 facts; 1,250,916 patients) and 'Asian' (13,806 facts; 13,806 patients). A green box highlights 'one or more of these' and a yellow box highlights 'drop a term on here'. The result is 1,256,917 patients.

The second screenshot shows a query named '0-9 yo' with one group containing '0-9 years old' (58,204 facts; 58,204 patients). A green box highlights 'one or more of these' and a yellow box highlights 'drop a term on here'. The result is 58,204 patients.

The third screenshot shows a query named '0-9 yo who are female OR Asian' with two groups. Group 1 contains 'Female' and 'Asian' (1,250,917 facts; 1,250,916 patients). Group 2 contains '0-9 years old' (58,204 facts; 58,204 patients). A green box highlights 'one or more of these' and a yellow box highlights 'drop a term on here'. The result is 27,729 patients.

Exclusion

Often times you will want to exclude patients. A common example of this is excluding patients who have passed away. To do this, pull the terms into your group that you want to exclude and click the button “exclude.” Once exclude is activated for that group you will see “NOT” prior to each term that the search is excluding. To exclude patients who have passed away exclude both deceased per SSA (social security administration) and the deceased folder (from O2 data). These are both found under Demographics \ Vital Status.

The screenshot shows a query named 'Group 1' with one group containing 'Deceased per SSA' (197,639 facts; 197,639 patients) and 'Deceased' (51,276 facts; 51,276 patients). A red circle highlights the 'Exclude' button. The result is 'none of these'.

Treat Groups Independently

Often researchers are interested in looking at events that occur anytime over the course of a patient’s life. To look at terms that occur any time in a patient’s medical record, use the default temporal constraint “Treat Independently.”

Please note: treat independently should ALWAYS be used for terms that will only appear once in the EMR (i.e. age, death, gender).

Same Financial Encounter

Sometimes researchers are interested in looking at events that happen at the same time. For example, you may be interested in looking at patients who have a diagnosis of diabetes recorded in their medical record at the same time as they are in the hospital. Use the temporal constraint “Selected groups occur in the same financial encounter” to find patients who meet this criteria.

The screenshot shows a query named 'Same Financial Encounter' with three groups. The temporal constraint is set to 'Selected groups occur in the same financial encounter'. The result is 'Occurs in Same Encounter'.

- Diabetes and hospital LOS treated independently: 29,749 patients
- Diabetes and hospital LOS in the same financial encounter: 25,521

Query Name: Independent - DM and LOS

Temporal Constraint: Treat all groups independently

Group 1: 250 Diabetes mellitus [1,686,485 facts; 84,549 patients] - 84549
E08-E13 Diabetes mellitus (E08-E13) [1,123,729 facts; 71,099 patients] - 71099

Group 2: Hospital LOS [246,146 facts; 135,488 patients]

one or more of these AND one or more of these

Number of patients: **29749**

For Query "Independent - DM and LOS"

Query Name: Same Encounter - DM and LOS

Temporal Constraint: Selected groups occur in the same financial encounter

Group 1: 250 Diabetes mellitus [1,686,485 facts; 84,549 patients] - 84549
E08-E13 Diabetes mellitus (E08-E13) [1,123,729 facts; 71,099 patients] - 71099

Group 2: Hospital LOS [246,146 facts; 135,488 patients]

Occurs in Same Encounter AND Occurs in Same Encounter

Number of patients: **25521**

For Query "Same Encounter - DM and LOS"

Independent AND Same Financial Encounter

For more complex queries, you may need to use both treat independently and same financial encounter. To achieve this you will need to use the small dropdown label in each group. Two examples are as follows:

1. You are interested in patients who have diabetes recorded in the hospital who have passed away. Since they may have died after their hospitalization, you will want to treat death independently
2. You are interested in patients who have diabetes recorded in the hospital who have a history of hypertension. You do not care if the hypertension was recorded during the sample hospitalization that diabetes was recorded during. For this, you should treat the diagnosis of hypertension independently.

Query Name: Mixed Temporal Constraints

Temporal Constraint: Selected groups occur in the same financial encounter

Group 1: 250 Diabetes mellitus [1,686,485 facts; 84,549 patients] - 84549
E08-E13 Diabetes mellitus (E08-E13) [1,123,729 facts; 71,099 patients]

Group 2: Hospital LOS [246,146 facts; 135,488 patients]

Group 3: Deceased per SSA [197,639 facts; 197,639 patients] - 197639
Deceased [51,276 facts; 51,276 patients] - 51276

Occurs in Same Encounter AND Occurs in Same Encounter AND Treat Independently

Number of patients: **6069**

For Query "Mixed Temporal Constraints"

Query Name: Mixed Temporal

Temporal Constraint: Selected groups occur in the same financial encounter

Group 1: 250 Diabetes mellitus [1,686,485 facts; 84,549 patients] - 84549
E08-E13 Diabetes mellitus (E08-E13) [1,123,729 facts; 71,099 patients]

Group 2: Hospital LOS [246,146 facts; 135,488 patients] - 135488

Group 3: I110 Essential (primary) hypertension [1,950,079 facts; 177,549 patients] - 177549
401 Essential hypertension [2,323,321 facts; 200,196 patients] - 200196

Occurs in Same Encounter AND Occurs in Same Encounter AND Treat Independently

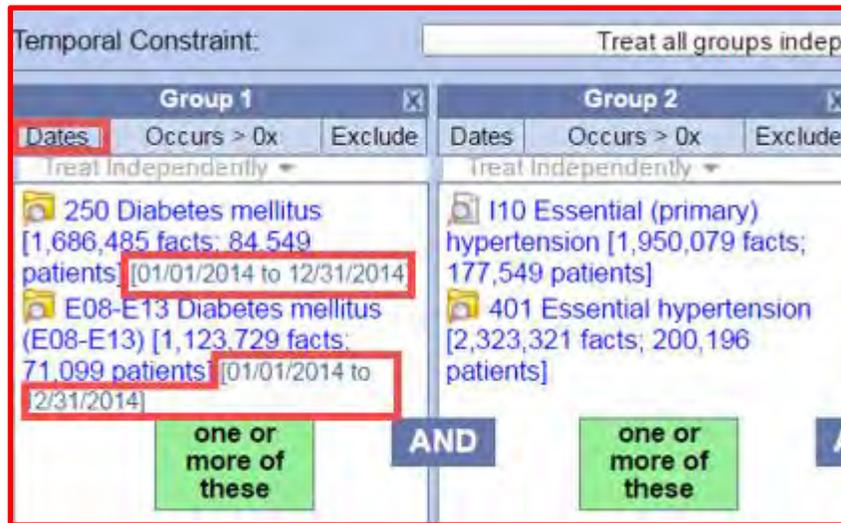
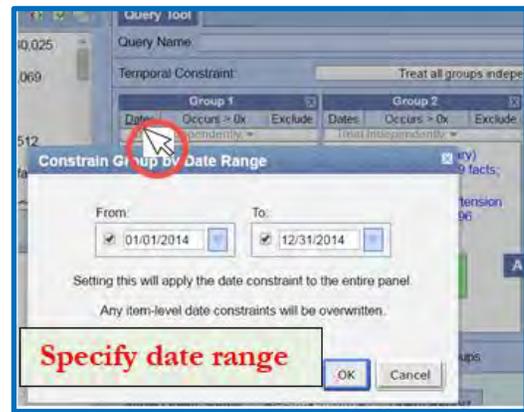
Number of patients: **21805**

For Query "Mixed Temporal"

Date Ranges

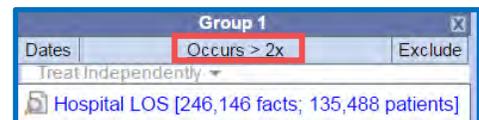
The date range in HERON allows you to specify that a patient must have had an item recorded during the date range.

- In the example below, the date range is on group 1 from 1/1/2014 – 12/31/2014
 - Patients in the cohort MUST have a recording of diabetes between 1/1/2014 and 12/31/2014
 - Patients in the cohort MAY have recordings of diabetes before or after the date range
- Patients in group 2 can have the diagnosis of hypertension recorded at any time
- **Please note:** ALL diabetes and hypertension data will be provided on the patients who meet the criteria. If you do not want any data outside of those date ranges, please state that in your data request.



Items Occur More than Once

Sometimes researchers are interested in items that occur >x number of times. For instance, you may be interested in patients who have had a hospital length of stay more than twice. Click on the default “occurs >0x” to adjust the number. This will adjust it for all items within a group.



- Examples:
 - Hospital LOS occurs >2x. This means patients have been admitted to KUH at least 3 times.
 - ALS diagnoses occurs >1x.
 - Sometimes patients may have ALS recorded once as a mistake, but physicians think that a patient is more likely to have a true diagnosis of ALS if it is recorded at least twice.

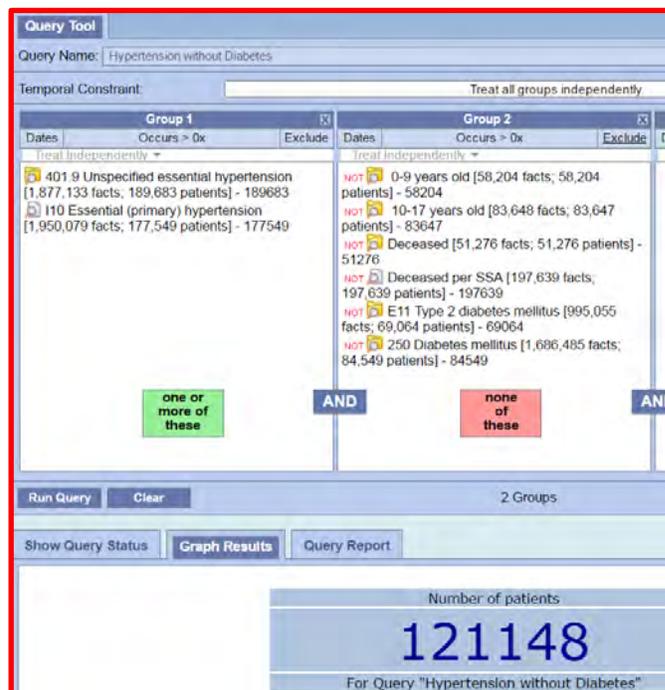
Finalizing Queries

Cohort Formation

Create a query in HERON that meets your inclusion/exclusion. In the following example, we want to look at patients who are over 18 who have hypertension, but do not have diabetes. These patients also must be living currently. The inclusion/exclusion criteria are as follows:

1. Inclusion Criteria:
 - a. Patients currently 18 years old or older
 - b. Living
 - c. Hypertension (ICD9: 401.9 or ICD10: I10)
2. Exclusion Criteria:
 - a. Deceased
 - b. Patients 0-17 years old
 - c. Diagnosis of diabetes (ICD9: 250 or ICD10: E11)

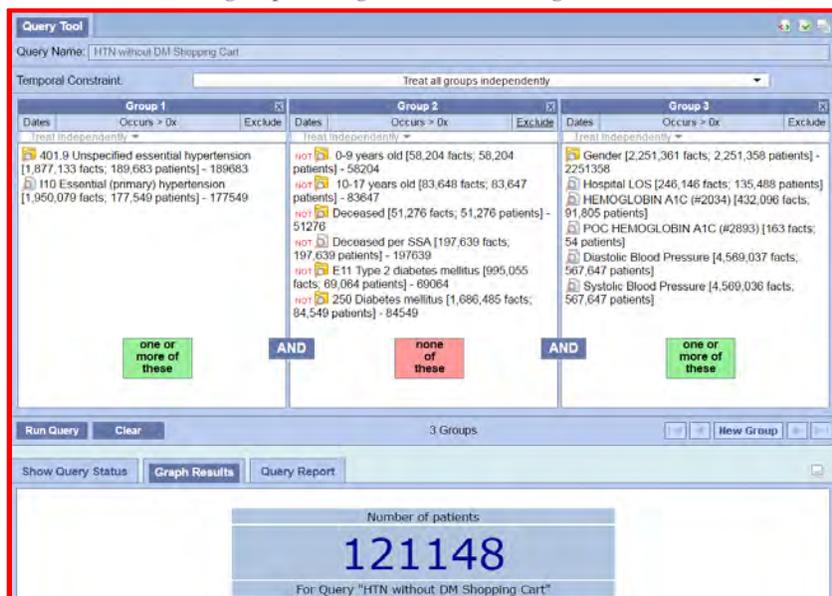
To translate this into HERON, use the basic construction of and/or. Since there are several exclusion criteria, all of these can be placed in one group and excluded.



Finalizing Query with Shopping Cart of variables

1. Define your cohort
2. Add a final group with every variable you are interested in analyzing and the variable gender found under the demographic folder.
 - a. Without the gender variable, the number of patients your search returns may have been reduced.

For the example above, the researcher is also interested in receiving data on hospital length of stays, hemoglobin A1C and blood pressure results. Add these to group 3 along with the variable gender.



Requesting Data

A data request must be submitted for identified and de-identified data requests.

Identified Data Request

Identified requests require IRB approval or a QI letter of determination from the IRB. With the request you will receive the patients MRN and all dates will be the true date.

De-Identified Data Request

De-identified requests do not require IRB approval, because it is considered non-human subjects research. The HERON database itself is under IRB approval, with waiver of consent and HIPAA authorization, because some members of the HERON team have access to identifiable data. However, end users do not need IRB approval.

The HERON staff act as ‘honest brokers.’ The honest brokers take data that were collected for clinical purposes and deliver it to the end user in a de-identified format. Because the end user cannot ascertain individual identities, the end user is not doing human subjects research.

Submitting a Data Request

1. Create a finalized query in HERON
 - a. Run the query and give it a useful name
 - b. Include all of the variables you are interested in by creating a “shopping cart” of variables. You will only receive data on terms found in your query
2. Faculty members are the only people who have access to submit a data request. Proceed to <https://heron.kumc.edu> using any web browser while on kumc’s network.
3. Select the "HERON Data Usage Request" link on the left hand side under Investigator Requests
4. Enter the first and last name of all individuals who need access to the data
 - a. You may also search for individuals on an IRB protocol by searching the IRB number
 - b. For individuals not at KUMC, please enter them manually on the next page
5. Answer all questions on the form and click submit



Data Request Review

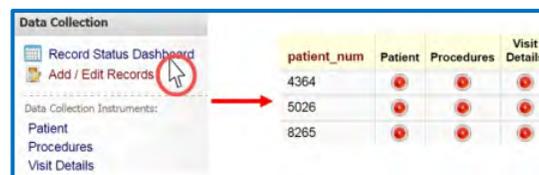
Data requests are reviewed by the hospital, the university, and the physicians group. All three entities need to approve the data request before data will be released. The approval process typically takes one week. After the data request is approved it takes an additional week for the data to be pulled.

Receiving Data

It takes about 2 weeks from data request submission until fulfillment of data. You will receive a condensed version of the data uploaded into a REDCap project. The raw data (all of the data) will be provided in the file repository of REDCap or via securefiles.

REDCap

When your request has been fulfilled, you will receive a notification email stating that you have been given access to a new REDCap project. Click on the “Record Status Dashboard” to get an overview of the patients in your cohort. You will see the patient_numbers and forms that have been completed for each patient. To view a form click on the red circle.



The patient form comes standard with all data requests. For de-identified requests, the dates will be shifted. Identified requests will include non-date shifted dates and will include the Medical Record Number.

Editing existing patient_num 4364

patient_num 4364

patient_num 4364

Case Summary

A data extract consists of a set of cases (either patients or encounters) and a collection of variables for each case. Since variables such as pulse may be observed any number of times for a single case, we give a count of the number of observations and, for the last observation, the date and value. For further detail, see the raw data file attachments.

For variables such as "diabetes diagnosis" the value is a term for the specific diagnosis such as "Diabetes mellitus type 1" and a corresponding code such as "ICD9:260.61". Numeric values with units are observed for variables such as a glucose lab. Text values are relatively uncommon.

Demographics

vital_status n

birth_date 1996-12-09 00:00:00

death_date

age 19

sex m

language english

race white

marital_status s

religion none

Form Status

Complete? Incomplete

All terms included in your HERON query will be on the additional forms (i.e. procedures, visit details, etc). It is important to note, those forms include only a summary of the data. Please look at the raw data for more in-depth analysis. The data will be shown as following:

- Variable: count = The number of times the variable is recorded in the medical record
- Variable: first date = The first date the variable was recorded in the medical record
- Variable: last date = The last date the variable was recorded in the medical record
- Variable: last name = Variable name
- Variable: last text val =
 - @: Null,
 - E: Equal for numerical variables
- Variable: last num val = The numerical value, if applicable
- Variable: last units = units, if applicable or available
- Variable: last modifier = last modifier that was recorded

Editing existing patient_num 4364

patient_num 4364

Height (cm): Count and Last Observation

Height (cm): count 15

Height (cm): first date 2008-02-08 12:03:00

Height (cm): last date 2016-05-15 12:03:00

Height (cm): last name Height (cm)

Height (cm): last code KUH*PAT_ENC HEIGHT

Height (cm): last text val E

Height (cm): last num val 165

Height (cm): last units

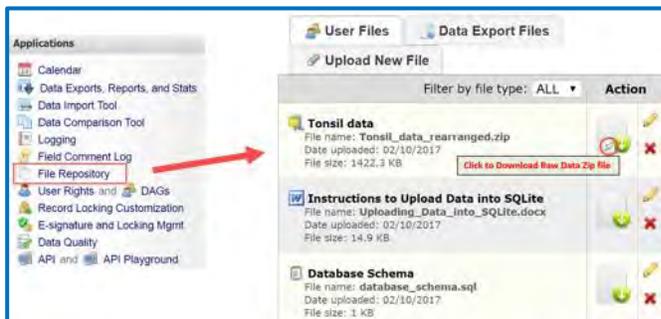
Height (cm): last modifier

Raw Data

The raw data will typically be uploaded into the file repository on the left hand side of the REDCap project. Once you open the file repository click on the download button to download the zip file.

If the data is too large, then it will be emailed via securefiles.

The raw data will contain four csv files: patient, data, code-info, and variable.



Patient Table

The patient table has one record for each patient in the patient set you created with your i2b2 query.

- **patient_num**: De-identified patient number which can be used to link information between the patient and data table
 - *Please note*: Patient_num's change between HERON releases. If you need to link two patient sets provided from different releases, please contact medical informatics: heron-admin@kumc.edu.
- **vital_status**: n = not deceased, y = deceased
- **birth_date**
 - De-identified datasets will include a date shift
 - Identified datasets will include the real birthday
- **death_date**
 - De-identified datasets will include a date shift
 - Identified datasets will include the real death date
- **Age**: Patient's current age. If the patient is deceased, the age will be the age at which they passed away. Ages over 89 are considered PHI. For patients over 89 their age will be masked to 88 years old.
- **Sex**: f = female, m = male, o = other
- **language**: patient's language from the EMR
- **race**: race of the patient
- **marital_status**: s = single m= married w = widow d = divorce x = separated p = life partner u = unknown
- religion
- **mrn**: only provided if identified dataset
- **last_date**: Last Encounter Visit date or Last medical record updated date.

patient_num	vital_status	birth_date	death_date	age	sex	language	race	marital_status	religion	last_date
4364	n	12/9/1996		19	m	english	white	s	none	4/27/2013
5026	n	10/1/2000		16	m	english	black	s	baptist	11/30/2016
8265	n	5/2/1999		17	f	english	white	s	none	8/14/2016

Data Table

All of the variable data is in one large table. Note that Heart Rhythm appears multiple times for the same patient. Raw data sets include all facts matching the variables and patients you request; constraints on dates and modifiers are not supported. A summary of the data, where only a count of the Heart Rhythms and the first and last values appear, is available in REDCap CRFs. The REDCap summary is sufficient for some forms of analysis. If you need further post-processing consider contacting heron-admin@kumc.edu for consulting possibilities.

- patient_num: De-identified patient number which can be used to link information between the patient and data table
- encounter_num: Things that happen in the same encounter (i.e. visit) have a unique de-identified number.
 - Encounter examples:
 - Office visit – height, weight, diagnostic codes, etc. will all be given the same encounter number
 - Inpatient Hospitalization – vitals, diagnoses, procedures, hospitalization dates, etc. will all have the same encounter number.
- valtype: example: (N)Number (T)Text (D) Date,
- tval: text value
 - @: Null,
 - When valtype = N (number), E is for Equal
- nval: numerical value, example: 120
 - This could be the nval for a blood pressure
- units: Example: Tab, mg
- code: the internal i2b2 concept code (e.g. DEM|AGEATV:10 or CPT:42820)
- modifier: Distinguish data source such as
- instance: this column will likely be unused in analysis. An instance is an event which typically connects modifiers (i.e. medication modifiers). This is useful, because you see one instance number per medication order which can link multiple modifiers associated with the medication. There can be multiple instance numbers per encounter. This number is deidentified in datasets.
- start_date: the start date for the observation
 - Deidentified datasets will have dateshifting
- end_date: the end date for the observation
 - Deidentified datasets will have dateshifting
- variable: the name of the query item that this fact matched (e.g. Gender)
- variable_index: index into the variable table
- code_label: e.g. Tonsillectomy and adenoidectomy; younger than 12
- modifier_label: Description for data source e.g. **DiagObs:Primary is the Primary billing diagnosis from IDX or UHC**

patient num	encounter num	valtype	tval	nval	units	code	modifier	instance	start_date	end_date	sourcesystem_cd	sub_encou	variable	variable index	code_label	modifier_label
4364	10449688	@	@		0	CPT:42820	@	8.13E+17	5/8/2008	5/8/2008	IDX@kuphysicians.com	8.26E+10	Tonsillectomy and adenoidectomy; younger than age 12	6	Tonsillectomy and adenoidectomy; younger than age 12	
4364	10449688	@	@		0	CPT:42820	@	8.13E+17	5/8/2008	5/8/2008	IDX@kuphysicians.com	8.26E+10	Tonsillectomy and adenoidectomy; younger than age 12	7	Tonsillectomy and adenoidectomy; younger than age 12	
4364	10417642	@			0	DEM AGEATV:10	@	1.04E+19	1/22/2008	1/22/2008	Epic@kumed.com	7.8E+09	10-17 years old at visit	1	10 years old at visit	
4364	10449688	@			0	DEM AGEATV:11	@	6.01E+18	5/8/2008	5/8/2008	Epic@kumed.com	1.72E+11	10-17 years old at visit	1	11 years old at visit	
4364	10449688	@			0	DEM AGEATV:11	@	6.2E+18	5/8/2008	5/8/2008	Epic@kumed.com	2.47E+11	10-17 years old at visit	1	11 years old at visit	
4364	10879549	N	E		142.4178	KUH PAT_ENC:HEIGHT	@	3.06E+18	6/16/2011	6/16/2011	Epic@kumed.com	1.27E+10	Height (cm)	3	Height (cm)	
4364	10879549	N	E		2392.32	KUH PAT_ENC:WEIGHT	@	3.06E+18	6/16/2011	6/16/2011	Epic@kumed.com	1.27E+10	Weight (oz)	14	Weight (oz)	
4364	10922466	N	E		2910.656	KUH PAT_ENC:WEIGHT	@	1.45E+19	9/1/2011	9/1/2011	Epic@kumed.com	1.6E+11	Weight (oz)	14	Weight (oz)	
5026	9475155	@	@		0	CPT:42820	@	1.14E+19	9/1/2010	9/1/2010	IDX@kuphysicians.com	1.35E+11	Tonsillectomy and adenoidectomy; younger than age 12	7	Tonsillectomy and adenoidectomy; younger than age 12	
5026	9475155	@			0	DEM AGEATV:9	@	1.54E+19	9/1/2010	9/1/2010	Epic@kumed.com	3.15E+11	0-9 years old at visit	0	9 years old at visit	
5026	9963324	N	E		153.8813	KUH PAT_ENC:HEIGHT	@	4.23E+18	9/24/2015	9/24/2015	Epic@kumed.com	1.02E+11	Height (cm)	3	Height (cm)	
5026	5626474	N	E		2407.886	KUH PAT_ENC:WEIGHT	@	1.22E+18	12/26/2014	12/26/2014	Epic@kumed.com	2.65E+11	Weight (oz)	14	Weight (oz)	
8265	10943149	@	@		0	CPT:42821	@	6.95E+18	3/21/2012	3/21/2012	IDX@kuphysicians.com	1.56E+11	Tonsillectomy and adenoidectomy; age 12 or over	4	Tonsillectomy and adenoidectomy; age 12 or over	
8265	10943149	@	@		0	CPT:42821	@	6.95E+18	3/21/2012	3/21/2012	IDX@kuphysicians.com	1.56E+11	Tonsillectomy and adenoidectomy; age 12 or over	5	Tonsillectomy and adenoidectomy; age 12 or over	
8265	12985534	N	E		144.6784	KUH PAT_ENC:HEIGHT	@	6.89E+18	1/21/2017	1/21/2017	Epic@kumed.com	7.92E+10	Height (cm)	3	Height (cm)	
8265	12985534	N	E		2492	KUH PAT_ENC:WEIGHT	@	6.89E+18	1/21/2017	1/21/2017	Epic@kumed.com	7.92E+10	Weight (oz)	14	Weight (oz)	

Code-Info

Some analysis uses not only relationships between variables and code values, but between the codes themselves and other parts of the term hierarchy. The code-info table enumerates all of the codes relevant to the variables in your query. This is particularly useful for flowsheet terms which may have a code_label such as “01. Regular” it is important to then look at the code_path or tooltip to determine what “01. Regular” is in reference to. An example is hearth rhythm.

- variable_index: Correlates with the variable and data tables
- variable: Correlates with the variable table and data tables
- variable_path:
- code is either
 - a code located under the variable
 - a modifier that may be used with the variable
- code_label
- code_path: is either
 - a concept path starting with the variable_path
 - a modifier path

variable_index	variable	variable_path	code	code_label	code_path
0	0-9 years old at visit	\\2b2\Visit Details\Age at visit\0-9 years old\	DEM AGEATV:9	9 years old at visit	\\2b2\Visit Details\Age at visit\0-9 years old\9 years old\
1	10-17 years old at visit	\\2b2\Visit Details\Age at visit\10-17 years old\	DEM AGEATV:10	10 years old at visit	\\2b2\Visit Details\Age at visit\10-17 years old\10 years old\
1	10-17 years old at visit	\\2b2\Visit Details\Age at visit\10-17 years old\	DEM AGEATV:11	11 years old at visit	\\2b2\Visit Details\Age at visit\10-17 years old\11 years old\
3	Height (cm)	\\2b2\Visit Details\Vitals\HEIGHT\	KUH PAT_ENC:HEIGHT	Height (cm)	\\2b2\Visit Details\Vitals\HEIGHT\
4	Tonsillectomy and adenoidectomy	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; age 12 or over-CPT:42821\	CPT:42821	Tonsillectomy and adenoidectomy; age 12 or over	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; age 12 or over-CPT:42821\
5	Tonsillectomy and adenoidectomy	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; age 12 or over-CPT:42821\	CPT:42821	Tonsillectomy and adenoidectomy; age 12 or over	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; age 12 or over-CPT:42821\
6	Tonsillectomy and adenoidectomy	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; younger than age 12-CPT:42820\	CPT:42820	Tonsillectomy and adenoidectomy; younger than age 12	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; younger than age 12-CPT:42820\
7	Tonsillectomy and adenoidectomy	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; younger than age 12-CPT:42820\	CPT:42820	Tonsillectomy and adenoidectomy; younger than age 12	\\2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; younger than age 12-CPT:42820\
14	Weight (oz)	\\2b2\Visit Details\Vitals\WEIGHT\	KUH PAT_ENC:WEIGHT	Weight (oz)	\\2b2\Visit Details\Vitals\WEIGHT\

Variable

The variable table has one record for each of the i2b2 "concepts" you requested:

- path: the internal i2b2 concept path. \a\b\ is under \a\ and so on. Can use this to find the concept in HERON again
- name: name of the variable as seen in the i2b2 query tool user interface but with the count part removed
- counts: as seen in the i2b2 query tool user interface
- index: links to the data table

path	name	counts	index
\i2b2\Visit Details\Age at visit\0-9 years old\	0-9 years old at visit	0-9 years old at visit [1,182,118 facts; 86,268 patients]	0
\i2b2\Visit Details\Age at visit\10-17 years old\	10-17 years old at visit	10-17 years old at visit [794,294 facts; 79,880 patients]	1
\i2b2\Visit Details\Vitals\HEIGHT\	Height (cm)	Height (cm) [3,537,760 facts; 483,116 patients]	3
\i2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; age 12 or over-CPT:42821\	Tonsillectomy and adenoidectomy; age 12 or over	Tonsillectomy and adenoidectomy; age 12 or over [434 facts; 424 patients]	4
\i2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; age 12 or over-CPT:42821\	Tonsillectomy and adenoidectomy; age 12 or over	Tonsillectomy and adenoidectomy; age 12 or over [434 facts; 424 patients]	5
\i2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; younger than age 12-CPT:42820\	Tonsillectomy and adenoidectomy; younger than age 12	Tonsillectomy and adenoidectomy; younger than age 12 [3,465 facts; 3,400 patients]	6
\i2b2\Procedures\PRC\Metathesaurus CPT Hierarchical Terms\Surgical Procedures\Surgical Procedures on the Digestive System\Surgical Procedures on the Pharynx, Adenoids, and Tonsils\Excision, Destruction of the Pharynx, Adenoids, and Tonsils\Tonsillectomy and adenoidectomy; younger than age 12-CPT:42820\	Tonsillectomy and adenoidectomy; younger than age 12	Tonsillectomy and adenoidectomy; younger than age 12 [3,465 facts; 3,400 patients]	7
\i2b2\Visit Details\Vitals\WEIGHT\	Weight (oz)	Weight (oz) [3,985,511 facts; 528,975 patients]	14

Uploading Data into SQLite

SQLite can be used to clean up the raw data to prepare it for analysis. SQLite is a free software which can be downloaded onto your computer. This can be downloaded from <http://sqlitebrowser.org/>. Once SQLite is installed, follow the steps below to upload the raw data. If you would like the database_schema.sql or dataset_import_identified.db, please contact heron-admin@kumc.edu.

Retrieve Data & Database Schema:

- Download raw data zip file from file repository in REDCap or send secure email.
 - Unzip file (Right click on zip file and click extract all)
- Download database structure from file repository in REDCap or send secure email.

Upload De-identified Database Schema:

- Open SQLite browser
- File → Import → Database from SQL File → Choose database structure to import: database_schema.sql
- Once complete, you should see 4 tables:
 - code_info_view
 - data_view
 - patient_view
 - variable_view

Upload Identified Database Schema:

- Open SQLite browser
- File → Open Database → Database from SQL File → dataset_import_identified.db
- Once complete, you should see 4 tables:
 - code_info_view
 - data_view
 - patient_view
 - variable_view

Upload Data into Database:

1. File → Import → Table from CSV
2. Choose one of the four data tables from raw data and click open
3. Replace the table name to match one of the 4 table names from the database schema
 - a. Ex: if the raw data name is “Project-code.info” put code_info_view
 - b. Ex: if the raw data name is “Project-data” put data_view
4. Check the box next to “column names in first line”
5. Uncheck the box next to “trim fields”
6. If you uploaded the database schema, a prompt will ask: “There is already a table of that name. Do you want to import the data into it?” – choose Yes
7. Click on Browse Data Tab and data should be filled in for the table you have uploaded
8. Repeat these steps to upload the raw data for each of the four tables
9. “Write Changes” to save the database

Basic Data Manipulation Techniques in SQL

Browsing Data in SQLite:

1. Click browse to look at the tables you have imported.
 - a. Patient table contains one row per patient and houses all of the basic demographics.
 - b. Data table contains many rows per patient and houses all of the observations. Note: if a patient has diabetes, this could be listed >100 times.
 - c. Code-info
 - d. Variable

Basic SQL Queries:

1. Click execute SQL to start writing SQL codes. Be sure to save your SQL file!! To get started, let's look at the data in the data table:
 - a. To look at all of the data:

```
SELECT * FROM data_view;
```
 - b. To see specific columns (i.e. patient_num, tval, code, start_date, and end_date)

```
SELECT patient_num, tval, code, start_date, end_date
FROM data_view;
```
 - c. Let's look at just the columns listed above for the ICD9code "ICD9:335.20" by using a where clause. Please note that if you do not know the exact code label, you can find it in the code-info table OR you can use the % before or after the part you do know. The % acts as a wildcard.

```
SELECT patient_num, tval, code, start_date, end_date
FROM data_view
Where code like 'ICD9:335.20';
```
 - d. The minimum start_date for the ICD9 code in 2c.

```
SELECT patient_num, tval, code, min(start_date), end_date
FROM data_view
Where code like 'ICD9:335.20';
```

OH MY! We only got one answer! The code above gives the minimum date from the whole patient set.
 - e. If you wanted to look at the minimum date for each patient (i.e. diagnosis date) you'll need to add a group by statement after the where clause. If we group by our patient_num that will give us the minimum start_date for EACH patient_num

```
SELECT patient_num, tval, code, min(start_date), end_date
FROM data_view
Where code like 'ICD9:335.20'
Group by patient_num;
```
2. You love the output of step 2e to get the minimum diagnosis date for each patient. But, now you want to save that as its own table. Once you create a new table, be sure to save your SQL file and "write changes" to your database.
 - a. Use the CREATE TABLE _____ as to make a new table. You can use any table name that you want just don't have any spaces in the name. I'm going to call it dx_date

```
CREATE TABLE dx_date as
SELECT patient_num, tval, code, min(start_date), end_date
FROM data_view
Where code like 'ICD9:335.20'
Group by patient_num;
```
 - b. Wow! You have made your very first table. You can look at what is in the table by using

```
SELECT * FROM dx_date;
```
 - c. Now, let's look at how many patients are in that table:

```
SELECT count(distinct patient_num)
FROM dx_date;
```

Joining Tables in SQLite:

One of the most useful features of SQL is the ability to join multiple smaller tables together. This can help you transform your “long-skinny” table to a table that has one row per patient. You need to have a variable that is common across the tables you are joining to link them together – typically this is patient_num.

1. To get practice joining try joining everything from the data_view table to the patient_view table.

```
SELECT data_view.*, patient_view.* FROM data_view
Join patient_view
On data_view.patient_num = patient_view.patient_num;
```

You will notice that your patient_view information repeats, because the patient_num is repeated through many rows of the data_view table.

2. If you only want specific columns from either table then in the SELECT use table_name.variable_name, table_name.variable_name2, etc. to include all of the specific variables you are interested in.

```
SELECT data_view.patient_num, data_view.tval, data_view.code, data_view.start_date,
data_view.end_date, patient_view.vital_status, patient_view.race
FROM data_view
Join patient_view
On data_view.patient_num = patient_view.patient_num;
```

Tonsillectomy Example

Based on the sample data above, we have run some code to show some basic data manipulations:

```
/*The following query shows you how to look at all of the patients in the query ordered by their birthdate with the
earliest birthdate first. The "*" means we are selecting all columns in the patient_view table. Later you will learn how to
only select specific columns*/
```

```
SELECT * FROM Patient_view
order by birth_date asc;
```

```
/*The data_view table is where all of the data for the variables in your query are found. The code for a specific variable
can be found in the code_info_view table */
```

```
SELECT patient_num, encounter_num, nval, code,
start_date, end_date, variable
FROM Data_view
where code like 'KUH|PAT_ENC:HEIGHT';
```

```
/*Finds the minimum height in the whole data set - will only return 1 number */
```

```
SELECT patient_num, encounter_num, min(nval), code,
start_date, end_date, variable
FROM Data_view
where code like 'KUH|PAT_ENC:HEIGHT';
```

```
/*Finds the minimum height per patient (due to group by patient_num). Also creates a table which can be referenced in
subsequent SQL code*/
```

```
CREATE TABLE min_height as
SELECT patient_num, encounter_num, min(nval) as min_height, code,
start_date, end_date, variable
FROM Data_view
where code like 'KUH|PAT_ENC:HEIGHT'
group by patient_num;
```

```
/*Selecting everything from the table we just made (min_height)*/
```

```
SELECT * FROM min_height;
```

```
/*Selecting everything from data_view table*/
```

```
SELECT * FROM DATA_VIEW
```

/*Looking in code_info_view table to find the CPT codes used in the query*/

```
SELECT distinct(code), code_label FROM code_info_view
where code like '%CPT%';
```

/*Created a table of the first tonsil surgery date based on CPT codes for each patient. This is a common way to make a table of diagnosis dates (ie hypertension diagnosis)*/

```
DROP TABLE tonsil_date; -- Only necessary if you need to delete a table and re-make it
CREATE TABLE tonsil_date as
SELECT patient_num, encounter_num, code, min(start_date) as surgery_date, variable
FROM DATA_VIEW
where code in ('CPT:42821', 'CPT:42820', 'CPT:42826', 'CPT:42825')
group by patient_num;
```

/*Looking at everything in our newly created table*/

```
SELECT * FROM tonsil_date;
```

/*Here we use our first join to join two tables together. We can see data from the tonsil_date table and the patient_view table. This is necessary to help us figure out the age at the time of surgery. Surgery date was in the data_view table (also found in tonsil_date table we created)and birth_date is in the patient_view table.

*/

```
SELECT tonsil_date.patient_num, tonsil_date.surgery_date, patient_view.patient_num,
patient_view.birth_date
FROM tonsil_date
join patient_view
on tonsil_date.patient_Num = patient_view.patient_num;
```

/*We created a table of each patient's age at the time of the surgery. julianday() allows us to look at a date in days, with Julian day number 0 assigned to the day starting at noon on January 1, 4713 BC. When we subtract the difference of the juliandays of two dates we get the days between the dates. We can divide by 365 to get the number of years between the dates. In this case the patients' age.

Again, you can find the code in the code_info_view or variable table

*/

```
CREATE Table surgery_age as
SELECT tonsil_date.patient_num, tonsil_date.surgery_date,
patient_view.birth_date,
(julianday(tonsil_date.surgery_date)- julianday(patient_view.birth_date))/365 as age_in_years
FROM tonsil_date
join patient_view
on tonsil_date.patient_Num = patient_view.patient_num;
CREATE TABLE weights as
SELECT patient_num, encounter_num, nval, code, start_date, variable from data_view
where code = 'KUH|PAT_ENC:WEIGHT';
```

/*We are interested in weights within plus or minus 30 days of the surgery date. Use a where clause and the statement "between" to find weight dates that are +/- 30 days from the surgery date.

*/

```
CREATE TABLE weights_surgery as
SELECT weights.patient_num, weights.nval, weights.start_date as weight_date, weights.variable,
surgery_age.surgery_date, surgery_age.age_in_years
FROM weights
join surgery_age
on surgery_age.patient_num = weights.patient_num
where weights.start_date between date(surgery_age.surgery_date, '-30 day')
and date(surgery_age.surgery_date, '+30 day');
```

/*Taking the earliest weight from all of the weights in our weights_surgery table (weights within +/- 30 days of surgery date).Using the group by patient_num returns the minimum weight per patient_num*/

```
CREATE TABLE earliest_weight_pre_surgery as
```

```

SELECT patient_num, nval, min(weight_date) as earliest_weight_date, variable,
surgery_date, age_in_years
FROM weights_surgery
group by patient_Num;

```

/*Counting the number of distinct patient_nums in the patient_view table
This can help you see how many patients you are looking for in your final data set*/

```

SELECT count(distinct patient_num)
from patient_view;

```

/*Joined the patient_view table with the earliest weight +/-30 days from surgery. Whenever you use join, you have to say what columns are the same in both tables. Typically you will want to join on the patient_num, but you may want to join on encounter_num or both*/

```

SELECT *
FROM Patient_view
left join earliest_weight_pre_surgery
on patient_view.patient_num = earliest_weight_pre_surgery.patient_num;

```

/*Creating a table to look at tonsillectomy over or under 12*/

```

CREATE TABLE tonsils as
SELECT patient_num, encounter_Num, code, start_date, variable
FROM Data_view
where variable like 'Tonsillectomy and adenoidectomy; age 12 or over'
or variable like 'Tonsillectomy, primary or secondary; age 12 or over' ;

```

/*Looking at the table we just made*/

```

SELECT * FROM tonsils;

```

/*Adding a column to the table tonsils, which will be a Y/N column to indicate if the patient was over or under 12. This same technique can be used to make a column Y/N for different comorbidities*/

```

ALTER TABLE tonsils
ADD under_12 text;

```

/*Updating the table to say Y when the variable says that the Tonsillectomy occurred younger than age 12*/

```

UPDATE tonsils
SET under_12 = 'Y'
where variable like 'Tonsillectomy and adenoidectomy; younger than age 12';

```

/*Updating the table to say Y when the variable says that the Tonsillectomy occurred younger than age 12*/

```

UPDATE tonsils
SET under_12 = 'Y'
where variable like 'Tonsillectomy, primary or secondary; younger than age 12';

```

/*Updating the table to say Y when the variable says that the Tonsillectomy occurred older than age 12. Note this time we combined two variables in our where clause*/

```

UPDATE tonsils
SET under_12 = 'N'
where variable like 'Tonsillectomy and adenoidectomy; age 12 or over'
or variable like 'Tonsillectomy, primary or secondary; age 12 or over';

```

/*We want to join our tables together to create our final data set that is ready for analysis. In the select statement, we chose which columns we wanted from each table*/

```

SELECT surgery_age.patient_num, surgery_age.age_in_years, surgery_age.surgery_date,
min_height.min_height, min_height.start_date as min_height_date,
earliest_weight_pre_surgery.nval as earliest_weight, earliest_weight_pre_surgery.earliest_weight_date,
under_12.under_12
FROM surgery_age
left join min_height
on surgery_age.patient_num = min_height.patient_num
left join earliest_weight_pre_surgery

```

```
on surgery_age.patient_num = earliest_weight_pre_surgery.patient_num
left join under_12
on surgery_age.patient_num = under_12.patient_num;
```

Data in HERON

Below are the high-level folders in HERON and explanation on some HERON terms.

Alerts

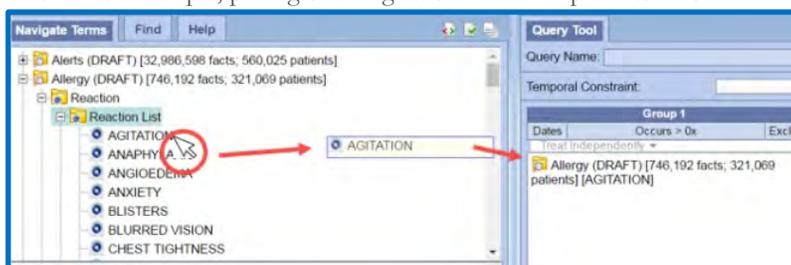
Alerts come from EPIC. There are modifiers at each level that allow you to limit your search to only alerts that had a specific status.

- **Alert Status (Modifiers):** Choice of alert status

Allergy

Common allergens are organized into folders beneath the parent folder “Allergy.” Researchers can choose a modifier to search at any level.

- **Reaction (Modifiers):** Choice of reaction or if no reaction was documented.
 - **Reaction List:** Pulling over a modifier from this folder would indicate that the patient has had that reaction recorded in their medical record. For example, pulling over “agitation” will find patients who have ever had agitation recorded for any allergen. Navigate to a specific allergen folder (i.e. animal) and pull over this modifier to find patients who only have agitation reaction to animals.
 - **Reaction not Documented**



- **Animal**
- **Chemical**
- **Drug Class**
- **Drug Ingredient**
- **Environmental**
- **Food**
- **Opt**
- **Plant**
- **Systemic**
- **Unclassified**

Cancer Cases (Lay-Person Hierarchy)

Data comes from NAACCR (North American Association of Central Cancer Registries) and is organized in a simplified manner for researchers not as familiar with the NAACCR data.

Cancer Cases (NAACCR Hierarchy)

Data comes from NAACCR (North American Association of Central Cancer Registries).

Cardiology Lab Results

Alerts come from EPIC. There are modifiers at each level that allow you to limit your search to only alerts that had a specific status.

Cardiovascular Registry (NCDR)

Information comes from the National Cardiovascular Data Registry.

- Age at visit

- Cath lab visit
- Coronary anatomy
- Demographics
- Diagnostic cath
- Discharge
- History and Risk Factors
- Intra and Post Procedure Events
- Labs
- Lesion and Devices
- Medications
- PCI Procedure

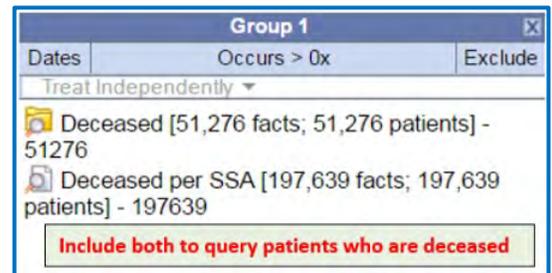
Cystic Fibrosis Registry

Includes information that has been entered into the national CF patient registry for patients seen at KUH.

Demographics

- **Age**
 - Represents the patients current age
 - **Note:** If you are looking for age at the time of a diagnosis (or some other event), use age at visit found under the Visit Details folder and specify that these must occur in the same encounter
- **Email on file**
 - **Note:** Does not provide you with the patient's email address
- **Enrollment**
 - **Catchment Area:** Includes patients living in the following counties:
 - Bates County, MO
 - Caldwell County, MO
 - Cass County, MO
 - Clay County, MO
 - Clinton County, MO
 - Jackson County, MO
 - Lafayette County, MO
 - Platte County, MO
 - Ray County, MO
 - Wyandotte County, KS
 - Franklin County, KS
 - Leavenworth County, KS
 - Linn County, KS
 - Miami County, KS
 - Johnson County, KS
 - **GPC Enrollment:** Patients must have 2 visits (>30 days apart) in the past 3 years
- **Ethnicity**
 - Ethnicity does not come with the standard HERON demographics. Pull over the entire folder into your "shopping cart" if you are interested in studying ethnicity.
- **Frontiers Research Participant Registry**
 - Patients who have been seen at KUH and have agreed to be contacted for future research studies.
 - Requires approval from the RRRC, but allows researchers to receive contact information for patients who meet study eligibility requirements.
 - **Note:** GREAT resource for recruitment, but always to remember to exclude those who are deceased.

- **Gender**
- **Language**
- **Marital Status**
- **My Chart**
 - Accessed MyChart
 - Chose to receive E-mail
 - MyChart Status
- **Place: distance from KUMC**
- **Place: School District**
- **Place: State**
- **Race**
- **Religion**
- **Vital Status**
 - **Deceased (folder):** death recorded in O2
 - Deceased (death date unknown): deceased per O2 records, but we are missing the death date.
 - **Note:** When you receive the data, it will be populated with a date. Please be mindful of this in your analysis and exclude the death dates where the death date is unknown.
 - **Deceased per SSA**
 - Records from the social security administration indicate that the patient has died. This information can only be updated if the patient's social security number is on file.
 - **Deferred**
 - **Living**
 - **Not Recorded**
 - Vital Status has not been recorded



Diagnoses

- **Modifiers**
 - **Billing (IDX, UHC)**
 - Clinic (IDX)
 - Hospital diagnosis (UHC)
 - Hospital procedure (UHC)
 - Primary billing diagnosis (IDX, UHC)
 - **Clinical (O2 EMR)**
 - Encounter diagnosis
 - Hospital problem
 - Medical History diagnosis
 - Primary diagnosis
 - Principal problem
 - Problem List

o GPC Modifiers: This folder is extremely useful for finding inpatient billing diagnoses

- Billing Diagnosis – Admit
 - Billing Diagnosis – Admit Non-Primary
 - o In EPIC there can be multiple final ICD diagnoses stored for each patient. Each diagnosis will have a unique line number.
 - o Any record with line not equal to 1 is coded as non-primary
 - Billing Diagnosis – Admit Primary
 - o In EPIC there can be multiple final ICD diagnoses stored for each patient. Each diagnosis will have a unique line number.
 - o The record associated with line equal to 1 represents the principal final coded diagnosis.
- Billing Diagnosis – Discharge
 - Billing Diagnosis – Discharge Non-Primary
 - o In EPIC there can be multiple final ICD diagnoses stored for each patient. Each diagnosis will have a unique line number.
 - o Any record with line not equal to 1 is coded as non-primary
 - Billing Diagnosis – Discharge Primary
 - o In EPIC there can be multiple final ICD diagnoses stored for each patient. Each diagnosis will have a unique line number.
 - o The record associated with line = 1 represents the principal final coded diagnosis.
 - Billing Diagnosis – Present on Admission
 - o This represents if a diagnosis was present on admission.
- Billing Diagnosis – Professional
 - Billing Diagnosis – Professional Non-Primary
 - Billing Diagnosis – Professional Primary

o Problem Status

- Active
- Deleted
- Resolved

o Trauma Registry: Having one of these modifiers indicates that a trauma registrar recorded this diagnostic code

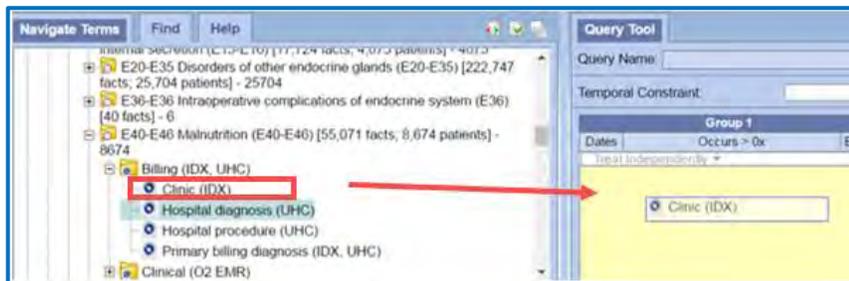
- Additional E-Code
- Injury Diagnosis
- Primary E-Code

o ICD10

o ICD10 is the diagnostic coding system that replaced ICD9. ICD10 data begins in XXX

o If you know an ICD10 diagnostic code, navigate through the terms using the letter and numbers associated with the ICD10 code. The tree structure is organized in alphabetical order and numerical order.

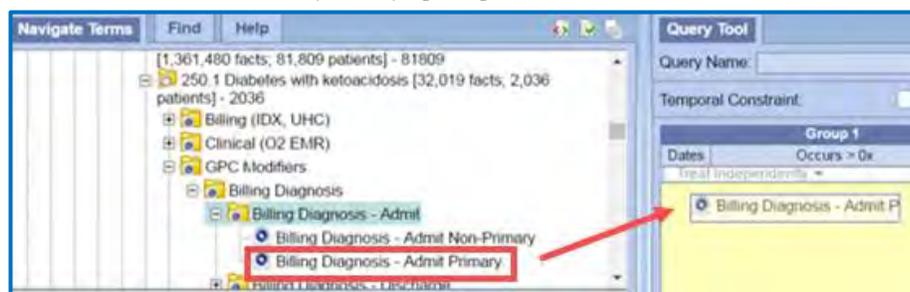
o You may pull over the entire diagnostic folder or a specific modifier within a diagnosis. The example above uses the modifier clinic



(IDX) billing which will only search for patients who had E40-E46 Malnutrition recorded in the clinic IDX billing system.

- **ICD9**

- ICD9 is the diagnostic coding system that came before ICD10. It is important to pull over the ICD9 and ICD10 diagnostic codes into your search.
- You may pull over diagnoses at many different levels. The simplest way to navigate through the ICD9 tree is by using the ICD9 code. For example, the ICD9 code for diabetes is 250. Start by opening the folder “240-279.99 Endocrine, Nutritional...,” because 250 falls within the range of numbers. Continue navigating until you find the specific folder for 250. If you want a diagnosis more specific than 250, you may open up the folder for ICD9 250 and choose 250.1 “Diabetes with ketoacidosis.”



- If you choose to use a specific modifier from above (i.e. billing diagnosis – admit primary) pull that modifier over from within the diagnosis folder you are interested in (i.e. 250.1)

- **Other Diagnoses Concepts**

- This folder is not the typical folder to find diagnoses in. It includes diagnostic terms that are not mapped to an ICD9 or ICD10 code. The patient counts for the terms located in this folder are typically very small.

Flowsheets

Flowsheets are organized by the hierarchy at KU hospital. There are several folders to sort through. The best way to navigate the hierarchy is to use the search by names feature and limit the search to the flowsheet folder. Once you find terms you are interested in, hover over the term and navigate to the folder in the hierarchy.

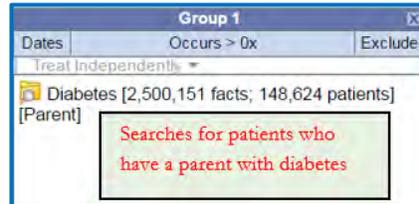
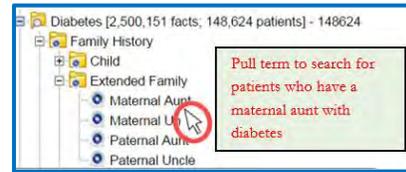
- **ED:** Folder contains information recorded in the ED. Currently, only two folders are populated
 - Consult
 - FLACC Scale: Face, Legs, Activity, Cry, Consolability scale
- **KU**
- **MODEL**
- **OB:** Folder contains obstetric information
- **OPHTH**
- **SAMPLE**
- **UKP**
- **XXX**
- **Z**

History

- **Family History Diagnosis**

- Modifiers

- Child
 - Daughter or Son
- Extended Family
 - Maternal Aunt, Maternal Uncle, Paternal Aunt, or Paternal Uncle
- GrandParent
 - Maternal Grandfather, Maternal Grandmother, Paternal Grandfather, or Paternal Grandmother
- Grandchild
- Other
- Parent
 - Father or Mother
- Sibling
 - Brother, Half Brother, Half Sister, or Sister
- Neg Hx = No recorded family history of specific condition



- Common conditions

- Pull over a specific folder to find patients who have any family history. Otherwise, limit your search to a modifier to find patients who have a specific family member with the condition.

- **Social History**

- Sexually Active
- Tobacco Usage
 - Smokeless Tobacco Use
 - Smoking Tobacco Use

- **Surgical History**

Laboratory Tests (KUH Hierarchy)

Terms are organized by KUH hierarchy.

Laboratory Tests (LOINC Hierarchy)

Terms are organized based on the national LOINC hierarchy

Medications

- **Modifiers:**

- Cumulative Daily Dose of Single Order
- Dispensed Medications
- Historical Medications: Recorded in a patients chart as a medication. These are typically reported by the patient (perhaps prescribed by an outside physician or an over-the-counter medication).
- Inpatient Medication Orders
- Medication Administration Record (MAR) Dose
- Medication Administration Record (MAR) Result Type
 - Within this folder you will find if the nurse or doctor gave the medication and how it was given. If you would like to find medications that were given you will need to pull over several (i.e. bolus, downtime given, and given all indicate that the patient received the medication)

- Other Medication Orders
- Outpatient Dispensed Medications (Surescripts):
 - Surescripts acts as a broker between doctor, pharmacy, and insurance to keep track of claim and fill information since 2014. Important to note: Surescripts information is not available for patients who do not use insurance to cover their prescriptions or if they do not have an upcoming scheduled appointment.
 - Amount:
 - Days Supply:
 - Dispense Status
 - Claim: Means that the pharmacy sent a claim to the insurance company, but does not mean that the patient picked up the prescription. Essentially means that the physician sent an order to the pharmacy.
 - Fill: Means that the patient picked up the prescription from the pharmacy
 - Unknown
- Outpatient Medication Orders
- PRN Inpatient Order
- RX Days Supply
- RX Frequency
- RX Quantity
- RX Refills
- Medications are organized by VA class. Use the Find to search by names. Once you find a medication you are interested in, hover over the term to find where it is located in the hierarchy. It is best to search a specific medication in the find so that it does not return thousands of results. Once found, you may pull over a broader medication folder.

Microbiology Negative Results

Microbiology tests that were immediately negative (don't require growth monitoring).

Microbiology Positive Results

Microbiology test results that were immediately positive (don't require growth monitoring).

Microbiology

Microbiology tests that require growth monitoring. These may be positive or negative results.

Orders

This folder includes tests, procedures, and consults (i.e., physical therapy) that have been ordered. Even if a patient has an order, it does not mean that the test was performed or that the test was performed at KUH.

Procedures

- **Modifiers**
 - Trauma Registry: Hospital Procedure
 - If this modifier is pulled over, it means that the hospital procedure was recorded by a trauma registrar.
- **ICD10 (Inpatient)**
- **ICD9 (Inpatient)**
- **Metathesaurus CPT Hierarchical Terms (Outpatient):** Outpatient procedures are billed using CPT codes. Navigate through the hierarchy or use the search by codes feature to search by a specific CPT code. If using the search by code feature, it is often helpful to navigate to the term in the hierarchy to see if there are any similar procedures.

REDCap

Users are able to request that their study specific REDCap be brought into HERON. This allows researchers to search for patient cohorts for patients already in their study who have certain conditions. If you would like a REDCap project brought in to HERON please contact heron-admin@kumc.edu.

Research Enrollment

A list of clinical trials pulled from CRIS that patients may be enrolled in. When searching you have the option to search by a specific status in regards to any research study or a specific research study.

- **Modifiers:**
 - Active
 - Beacon Treatment Plan – Not Active for Research
 - Inactive
 - Pre-Enrolled

Specimens

Fluid and tissue specimen information is pulled from the KUMC Biospecimen Repository

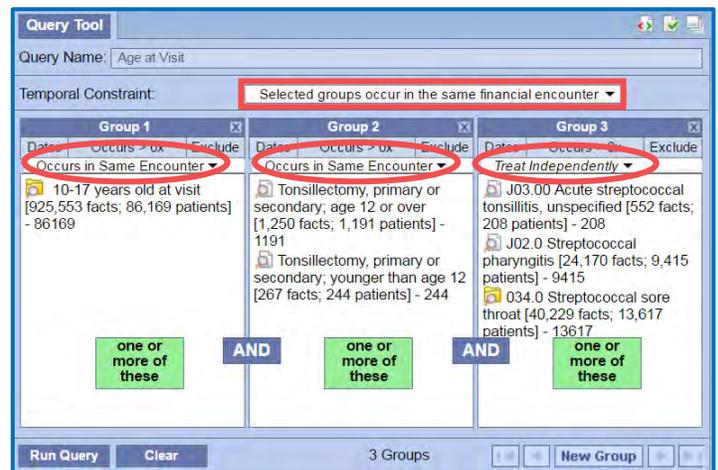
Trauma Registry (Draft)

Includes information entered into the national trauma registry (NTDS) from KUMC

Visit Details

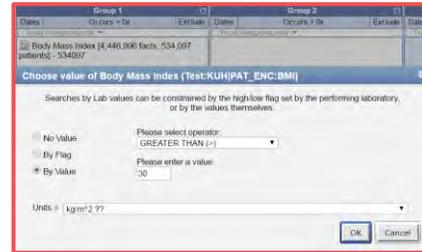
Provides information regarding each inpatient and outpatient visit including patient vitals

- **Age at Visit**
 - The age at visit allows users to search for a procedure or diagnosis that occurred for patients of a certain age. To use this change the temporal constraint to “selected groups occur in the same financial encounter”
 - The example to the right shows patients who were 10-17 years old at the time of a tonsillectomy procedure. These patients also must have a diagnosis of strep throat recorded by ICD9 or ICD10 at any point in their life. Since this is the age at the visit, their current age could be much higher. The strep throat in group 3 is treated independent of groups 1 and 2, which means it could occur before, during, or after the tonsillectomy procedure.



- **Clinical Service Department per O2**
 - Search this folder to find departments that patients were seen in per O2. The folder is organized by inpatient and outpatient departments.
- **Clinical Service Department per IDX**
 - Search this folder to find departments that patients were seen in per the IDX billing system. The folder is organized by inpatient and outpatient departments.
- **Discharge Disposition Codes**
 - Discharge disposition codes are found in this folder. This includes things such as “acute care facility (another hospital).”
- **Encounter Type**

- Visits have been grouped by encounter type based on the PCORnet common data model specifications.
- **Place of Service (IDX)**
- **Provider**
 - Searches for billing provider, primary care physician, or service provider. This data is not available for de-identified data requests.
- **Visit Vitals**
 - All vitals recorded during a visit are found in the visit vitals.
 - Users can specify a value for items such as BMI (i.e. inclusion criteria is BMI >30 or can choose “no value”



Visit Notes

Physician notes are found under note types. Some common note types include progress notes and operative reports. All note types are available for identified requests. For de-identified requests, specific note types may or may not be available. Please contact heron-admin@kumc.edu if you would like to use the de-identified notes. Prior to release of de-identified notes, notes are scrubbed of all 18 HIPAA personal health identifiers. A subset of each note type is reviewed by privacy specialists to confirm that the note type is scrubbed of PHI. Two privacy specialists must approve the release of the note type.

- **Visit Notes**
 - Note Concepts
 - Note Types
 - Use this folder to find things such as progress notes or operative notes.
 - Pulling notes through HERON may save time on having to go back into EPIC to review the medical records. They also allow users to have access to important information that would otherwise not be available with a de-identified request.

Vizient (formerly UHC)

Vizient (formerly United HealthSystem Consortium) includes inpatient quality data.

- **UHC Agency for Healthcare Research and Quality**
- **UHC Core Measures**
- **UHC Demographics**
- **UHC Diagnosis**
- **UHC Procedures**
- **UHC Visit Details**
 - Length of Stay is found within the visit detail folder. If interested in inpatient hospitalizations pull “hospital LOS” and “ICU days” (if interested in number of days in the ICU).
 - Note: The expected length of stay (LOS) is also found within the folder “Length of Stay.” The expected LOS includes how long Vizient would anticipate a patient with similar factors to be admitted to the hospital. Do not pull this subfolder over if you are only interested in studying how long the patient was actually in the hospital. Only pull over the terms highlighted on the right.

