ADAPTABLE User Documentation

Table of contents

Section I: Accessing the New Features and Tool

Introduction

The functionality developed for the ADAPTABLE Supplement project is open-source and available as a part of the PopMedNet (PMN) source code. There are two options for accessing the new features. The first option is through installing the source code found at the PopMedNet GitHub here: https://github.com/PopMedNet-Team/ADAPTABLE. The public is welcome to download the source code and create a personal instance to utilize the tool. To assist with setting up a new instance, open-source documentation can be found here: PopMedNet Open Source Resources. Second, if organizations and/or data partners prefer not to set-up their own instance, they are welcome to contact PopMedNet through our service desk and request a project space within a PopMedNet-hosted instance: PopMedNet Service Desk: Request a New Project. Additionally, in the upcoming months, PCORI-funded projects will be able to access the functionality via the PCORnet Query Tool on PopMedNet. For more information see the figure below or contact the PopMedNet team via our Service Desk.

1.

Open Source Download
- Available now for download on GitHub.
- Organization is responsible for installing, setting-up, and hosting the instance.
- The PopMedNet team is available to assist and provide guidance to any open-source partners.

2.

PopMedNet Hosted
- PopMedNet can set-up and host an instance for an organization interested in accessing the new functionality.
- Available now.
- Includes support and hosting costs.

PCORI-funded projects will have access to the functionality on the PCORnet Query Tool in the upcoming months.
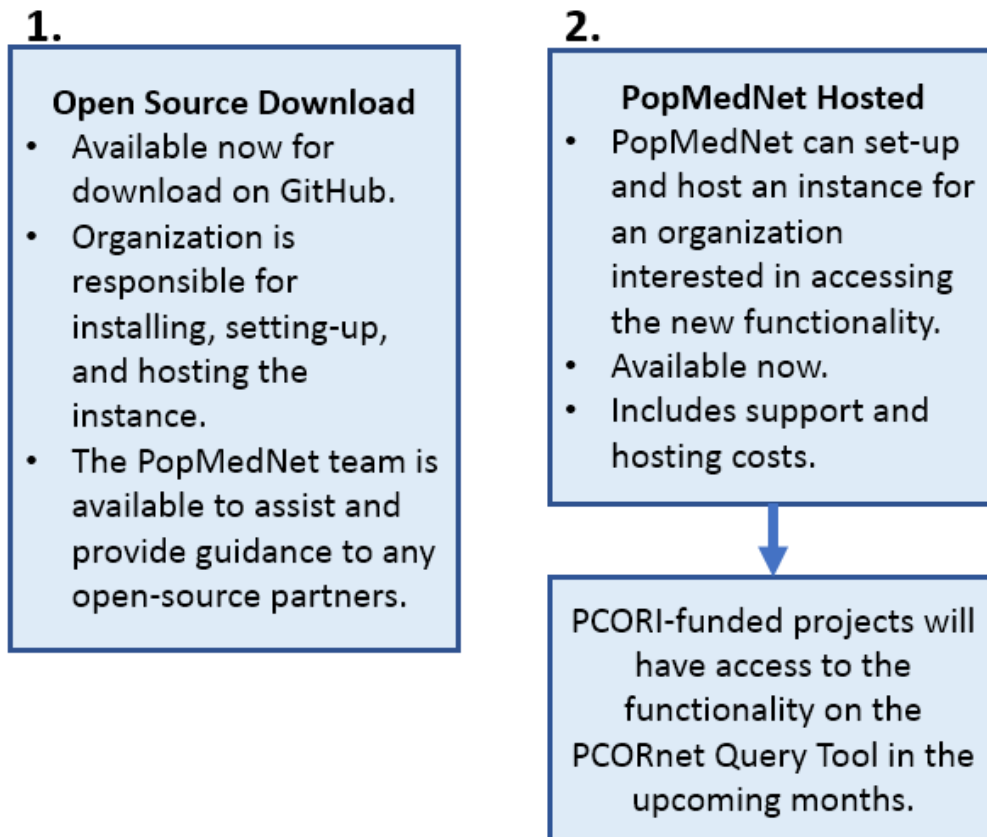
Figure 1. Two options for Accessing the New MDQ Functionality

Account and DataMart Client Set-up

Requests are initiated at a PopMedNet site (or open-source equivalent) and distributed to data partners. Data partners respond to the requests via the DataMart Client (DMC) application. In order to respond to requests, users must first download, install, and configure the DataMart Client locally.
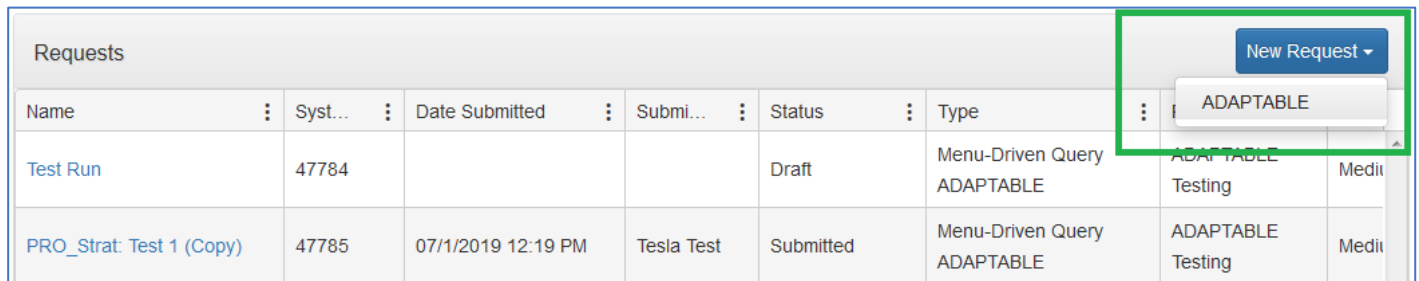
DataMart Client Installation

- Any user that will be executing and responding to queries with the new functionality will need the most recent version of the DataMart Client. Find the download link for **version 2019.2.A** (NOT 2019.2) here: [DataMart Client Installer.](#)

- For more information regarding the set-up and configuration of the DataMart Client please see the [PopMedNet User Guide](#)**.**

Section II: Creating the Menu-Driven Query (MDQ) Request

Create a Menu-Driven Query (MDQ) Request

1. The investigator must first login to their instance of PopMedNet with a username and password.

2. On the homepage of the site, click the **New Request** button above the Request panel. A drop-down will appear with the project(s) in which you have permission to create requests. Choose the project that is configured to include the updated MDQ tool. The screenshot below shows the **ADAPTABLE** project.

Figure 2. ADAPTABLE Project

3. The request type list appears after selecting a project. Your permissions govern the projects and associated request types you see. Select the following request type: **Menu-Driven Query**

4. After selecting the request type, the user will be brought to the request creation screen. Enter the metadata in the request metadata pop-up. In the **Overview** tab, include a

unique name and fill-in the metadata fields as appropriate. In the **Description** tab, fill-in
the request purpose and description. Save the metadata.

All metadata is distributed to the DataMarts with the request and should be detailed
enough to allow data partners to identify the request and determine how to
respond. See screenshot of the metadata box below in Figure 3:

Figure 3. Metadata

5. Once the metadata is saved, the request will default to the **Task: New Request** tab. In
    the Task tab, specify the request specific parameters and settings depending on the
    research question. [ADAPTABLE-specific queries and parameters are defined in MDQs 3-5
    in Section IV.]

Figure 4. Task Tab in a New Request

Section III: MDQ Parameters Defined

Menu-Driven Queries (MDQs) utilize a simple point-and-click interface to add terms to the request criteria that are joined through logical operations (and/or), associations, and then grouped according to the chosen stratification(s). A major benefit is that users do not need programming expertise to create and execute MDQs.

Terms

Terms are criteria that define the patient cohort of interest. Each term corresponds to one or more fields in the PCORnet Common Data Model v4.1. While terms define the results, the term will not appear as a column in the output unless the related stratification is also selected. The table below shows which fields in the database **some** of the available terms query:

MDQ Term in the PMN User Interface	Field(s) and [Table] in PCORnet CDM
Trial	TRIALID [PCORNET_TRIAL]

Patient Reported Outcome (PRO)	PRO_ITEM_NAME and PRO_RESPONSE_TEXT [PRO_CM]
Sex	SEX [DEMOGRAPHICS]
Observation Period	ADMIT_DATE [ENCOUNTER]
Diagnosis	DX and DX_TYPE [DIAGNOSIS]

Table 1. Terms

Stratifications

Similar to the terms, the stratifications correspond to fields in the PCORnet Common Data Model (and therefore the data partner's database). The stratification will stratify the results by the chosen fields—meaning that the stratifications will appear in the result output. The inclusion of stratifications in the query provides additional information by adding columns to the output. Table 2 below details the affect that some stratifications have on the output.

MDQ Stratification in the PMN User Interface	How the Output is Affected
Trial	TRIALID and PARTICIPANTID columns added.
Patient Reported Outcome (PRO)	PRO_ITEM_NAME, PRO_RESPONSE_TEXT, PRO_RESPONSE_NUM, and PRO_MEASURE_SEQ columns added.
Sex	Stratification by the SEX variables present in the database.
Observation Period	Yearly or monthly stratification options available.
Patient Reported Outcome Encounters	PRO_ITEM_NAME, PRO_RESPONSE_TEXT, PRO_RESPONSE_NUM, PRO_MEASURE_SEQ, ENCOUNTERID, ENC_TYPE, ADMIT_DATE, DX, DX_TYPE, PX, and PX_TYPE columns added. This stratification indicates the addition of a temporal relationship between the PRO and Encounter information. This expands on the PRO columns included in the PRO stratification above. This stratification and the PRO stratification should not be included in the same query as it would be redundant.

Table 2. Stratifications

Temporal Events

The Temporal Event feature in MDQs query the PRO_CM table, specifically the PRO_ITEM_NAME field, in the database for the index event. For each patient with at least one index event, the request will then query that patient's ADMIT_DATEs within the ENCOUNTER table and return all encounters that fall within the specified time window. If a patient has multiple Index Events, the query will apply the time window to each one separately.

To add the time window functionality to the query, the investigator will need to add the stratification: **Patient Reported Outcome Encounters**. By adding this stratification, a new panel, **Temporal Events**, will appear.

Figure 5. Patient Reported Outcome Encounters Stratification and Temporal Events Panel

The **Patient Reported Outcome Encounters** stratification creates additional columns in the output that pulls data from the following tables: PRO_CM, ENCOUNTER, DIAGNOSIS, and PROCEDURE (see Table 2 for the exact fields). In the Temporal Events panel, there is the **Index Event Identifier**. The Index Event Identifier queries the PRO_ITEM_NAME field in the database. The query will then look for a SAS date in the same row as the Index Event in the PRO_RESPONSE_NUM field. For example, if HOSPITALIZATION_DATE was entered into the PRO_ITEM_NAME field, the user would enter "HOSPITALIZATION_DATE" as the Index Event (see Figure 5). The query will then search for data within the same row as each instance of HOSPITALIZATION_DATE and use the SAS date in the PRO_RESPONSE_NUM to calculate the specified time window.

Figure 6. Example of PRO Information in the Database

The time window can be specified by number of days before and number of days after the Index Event. The days can be changed and customized for each query. For example, another query that has a time window of "0 Days Before and 0 Days After" will search for ADMIT_DATEs that are exact matches with the Index Event dates. The calculation includes querying the ADMIT_DATEs (from the ENCOUNTER table) that fall within the specified time window around the HOSPITALIZATION_DATE for each of the patients that match the query criteria. All encounters that match both the criteria and time window will be returned in the output.

For more information on the definition of each field listed above, please refer to the PCORnet Common Data Model v4.1.

Section IV: Example Menu-Driven Queries and ADAPTABLE Use Cases

Menu-Driven Queries (MDQs) are primarily a cohort-defining tool. The more basic queries return a count of patients that match the given criteria to inform investigators of the potential study population. In the case of the ADAPTABLE Supplement project, the tool functionality was expanded to return detailed encounter and EHR information on an already chosen study population in a trial. With regard to either purpose for the queries, MDQs are most optimal when a series of queries are run, each query building upon the previous one. This process allows the investigators to observe how the addition of more criteria affects the output. The recommended first step is for investigators to define the cohort of patients more broadly in a baseline query. Each subsequent query will then contain additional criteria and continue to build upon the previous queries. For example, in the case of a new study, a good baseline query would be to first determine the total number of patients in the Trial of interest, e.g. ADAPTABLE. The second query may then ask "of those patients in the trial, how many reported a hospitalization event?". More information on these initial queries are below. The MDQs defined in #3, #4, and #5 below are based on the ADAPTABLE Supplement's use cases and utilize all of the new

functionality, including the ability to query both the Trial and Patient Reported Outcomes (PRO) tables in the database.

Multi-MDQ Example Set

MDQ 1.

As a baseline query, find the total number of patients in the study cohort. To find that number, create a MDQ in which there is only one term in the criteria: the **Trial** term. Investigators would enter the Trial ID(s) of interest. Next, remove the stratifications and execute against the database. The result will be a count of the unique patients in the Trial. See Figure 7 for an example of this query.

Figure 7. MDQ to Find Count of Patients in the ADAPTABLE Trial

MDQ 2.

If the study is interested in a particular patient-reported event, it may be useful to know the number of patients that reported that event and/or the total number of records for that event. For example, the ADAPTABLE Supplement study was interested in the patient-reported hospitalization dates. To find the total number of patient-reported hospitalization dates in the trial, the Investigator would add the **Patient Reported Outcome** term to the criteria. In the **Item Name** section for this term, enter the event of interest as it appears in the database, e.g. HOSPITALIZATION_DATE. Next, ensure that the query has both the Trial and Patient Reported Outcome stratifications (see Figure 8). The output of this query will result in a list of every instance of HOSPITALIZATION_DATE for participants in the trial. Investigators could then use the output to determine both the total number of records for this particular item, as well as the number of unique participants that have reported this item by looking at the PARTICIPANTID column.

Request Details

Criteria Groups

Criteria Group: Group 1

Group Name

Group 1

Trial	ADAPTABLE
	And
Patient Reported Outcome (PRO)	**Item Name:** HOSPITALIZATION_DATE
	Item Response:

Report Selector

Selected Stratification Fields:

Trial
Patient Reported Outcome (PRO)

Figure 8. MDQ for All Records with HOSPITALIZATION_DATE

MDQ 3. (ADAPTABLE Use Case 1)

Once some basic information has been collected regarding the patients in the Trial, a more complicated and specific use case query can be sent. The ADAPTABLE Supplement project had the following Use Case 1: For patients in the ADAPTABLE trial, find all EHR-based encounters occurring within a defined time window of a patient reported hospitalization.

In order to create a query to answer the use case above, create a Menu-Driven Query (MDQ) request as described in Section II. Then follow the steps below to enter the correct parameters:

1. Add the **Trial** term. Enter TrialID exactly as it appears in the database, e.g. ADAPTABLE.

2. Add the **Trial** stratification (if it is not already present).

3. Add the **Patient Reported Outcome Encounters** stratification. Do **not** add the **Patient Reported Outcome** stratification.

Figure 9. Stratifications for Use Case 1

4. In the **Temporal Events** panel, ensure that the Index Event matches what is present in the database for the hospitalization date, e.g. HOSPITALIZATION_DATE.

5. Specify desired time window. See Figure 5 for an example of a time window of 7 days before and 7 days after.

6. Select the DataMart that will be executing the query, i.e. Vanderbilt University Medical Center DataMart.

7. Click **Submit.**

Screenshot of submitted MDQ for Use Case 1:

In this example, we would expect results for patients in the ADAPTABLE trial, with an EHR-based visit date that is 7 days before or 7 days after the patient reported hospitalization date, which is stored in the PRO_CM table.

Figure 10. Use Case 1

MDQ 4.

In the case of MDQs that contain time windows, it is highly recommended that at least one other query with the same criteria, but a different time window be distributed. The previous query had a time window of "7 days before and 7 days after" the index event. A follow-up query could have a time window of "30 days before and 30 days after" the index event, or some other range. All of the steps would be the same as MDQ 3 above, the only exception being that the investigator would enter 30 days before and 30 days after in the Temporal Events panel. A larger time frame is likely to return more in the output.

MDQ 5. (ADAPTABLE Use Case 2)

Building upon the information in Use Case 1, the ADAPTABLE Supplement team was also interested in the patient-reported hospitalization events. The next MDQ aimed to answer the following: For patients in the ADAPTABLE trial, find all EHR-based encounters occurring within a defined time window of a patient reported hospitalization, as well as all of the patient reported hospitalization events for those patients.

In order to create a query to answer this question, create a Menu-Driven Query (MDQ) request as described in Section II. Then follow the steps below to enter the correct parameters:

1. Add the **Trial** term. Enter TrialID exactly as it appears in the database, e.g. ADAPTABLE.

2. Add the **Patient Reported Outcome** term. In **Item Name**, enter how the hospitalization event data appears in the database, e.g. HOSPITALIZATION_EVENT (see Figure 11).

Figure 11. PRO Term

3. Add the **Trial** stratification (if it is not already present).

4. If the **Patient Reported Outcome** stratification is present, **remove it**.

5. Add the **Patient Reported Outcome Encounters** stratification. See Figure 9 for reference on how the stratifications section should appear.
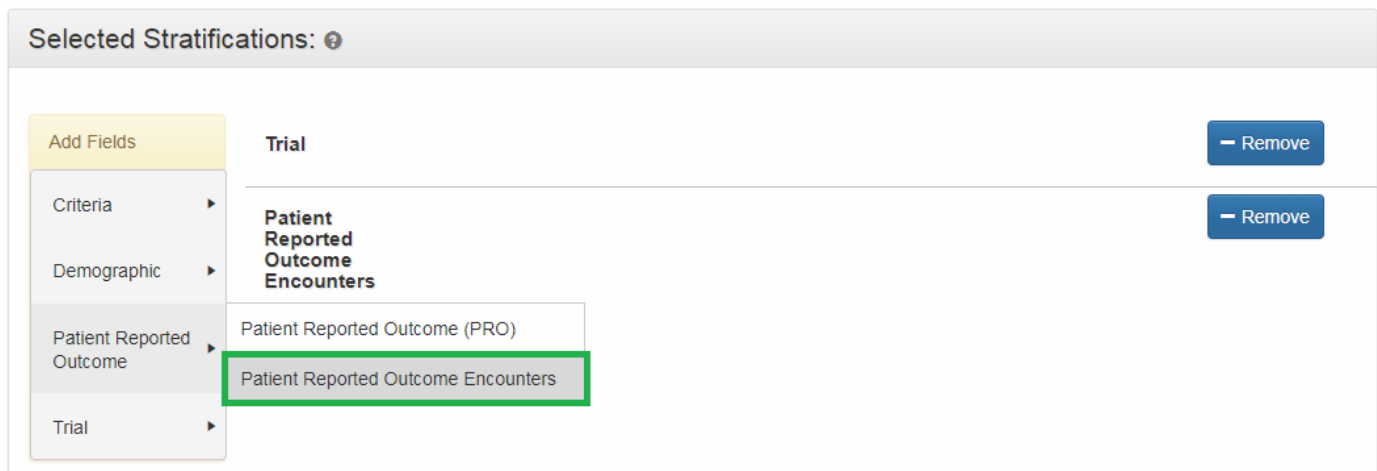
6. In the **Temporal Events** panel, ensure that the Index Event matches what is present in the database for the hospitalization date, e.g. HOSPITALIZATION_DATE.

7. Specify desired time window.

8. Select your DataMart that will be executing the query.

9. Click **Submit.**

Screenshot of submitted MDQ for Use Case 2:

In this example, we would expect results for patients in the ADAPTABLE trial, with an EHR-based visit date that is 7 days before or 7 days after the patient reported hospitalization dates AND have at least one record for PRO_ITEM_NAME = HOSPITALIZATION_EVENT. For each

patient that returned in Use Case 1, all instances of hospitalization events will also return. If a patient in Use Case 1 does not have a hospitalization event, they will not return.

Figure 12. Use Case 2

Note: If a participant was in the results for Use Case 1, but is absent from the results of Use Case 2, then the patient either has no entry of hospitalization event or the row containing the event has a typo in the PRO_ITEM_NAME field. If the query is looking for all rows with HOSPITALIZATION_EVENT, any instances with a typo or difference in capitalization will be excluded.

Other MDQs

Since MDQs contain nearly limitless possible combinations of terms and stratifications, other queries can be created depending on the investigators' specific interests.

1. If interested in viewing all **patient-reported items and responses** within the database, create a query without time windows. Build upon the baseline query by adding the Trial and Patient Reported Outcome (PRO) stratifications. The results will include the Trial and PRO information for every participant in the trial (see Table 2 for the exact fields). This may also assist in seeing if there are any typos or missing rows. See screenshot below for an example of this query.

Figure 13. MDQ to View all PRO Information for Trial Participants

2. Additionally, if interested in seeing rows that may have been excluded for one reason or another, **create a query using exclusion terms.** To add exclusion criteria while creating a MDQ, select the **New Group** button along the top of the criteria group. A second criteria group will appear. Enter a name for the criteria group. The name can be anything from "Group 2" to " Exclude HOSPITALIZATION_DATE", etc. Select the check box labelled **Exclusion Criteria.**

Figure 14. Create New Criteria Group

The query below in [Figure 15](#) will show any participants along with their PRO information that were excluded from Use Case 1 due to not having a row with HOSPITALIZATION_DATE in the PRO_ITEM_NAME field.

Figure 15. MDQ with Exclusion Criteria

The output from this query may provide the analyst an opportunity to spot a typo or missing value. See an example of the output below:

Query criteria: Exclude PRO Item Name = HOSPITALIZATION_DATE.

TRIALID	PARTICIPANTID	PRO_ITEM_NAME	PRO_RESPONSE_TEXT	PRO_RESPONSE_NUM	PRO_MEASURE_SEQ	Encounters
ADAPTABLE_1	Gil_A	HOSPITALIZATION_EVENT	MI	<< NULL >>	7.02	1
ADAPTABLE_1	Gil_A	LOCATION	VUMC	<< NULL >>	7.02	1
ADAPTABLE_1	Gil_A	hospitalization_date	5/11/2018	21315	7.02	1

Results show a patient that was excluded from the Use Cases due to a capitalization typo. The PRO Item Name field must be populated exactly as stated for the patient to return in the output.

Figure 16. Exclusion MDQ Output

If the PRO stratification is removed from this query, the output will simply show the list of participant IDs and Trial IDs for those that were excluded for the same reason.

To find patients excluded from Use Case 2 for not having a row with HOSPITALIZATION_EVENT, simply replace HOSPITALIZATION_DATE in the exclusion criteria group.

Section V: Responding to Queries in the DataMart Client (DMC)

In order to respond to MDQs, the analyst must have a DataMart Client installed. Instructions for installing and setting-up the DataMart Client can be found in Section I. Once the set-up is complete, follow the steps below to execute the queries.

1. Open the DataMart Client. Go to the **About** link in bottom right corner. Ensure that the version is 2019.2.A or higher.

2. In the same pop-up as above, select **Packages**. Select all of the adapter processor packages listed. Click **Delete Packages**, as shown in Figure 17. This will ensure that the query is being executed with the most recent code and not outdated code from past queries.
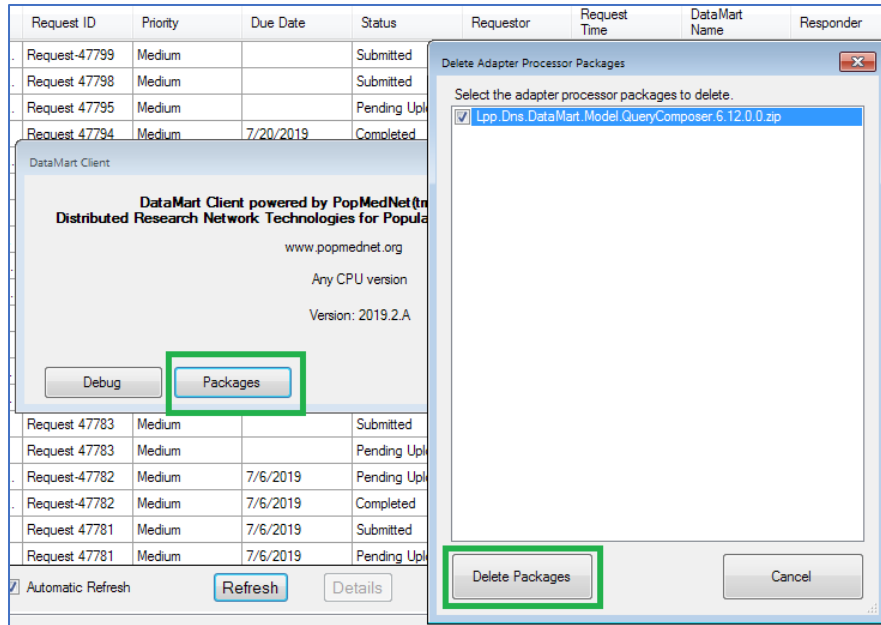
Figure 17. Delete Packages in DMC

3. Open query of interest. Navigate to the **Request Details** tab. Select **Run**.
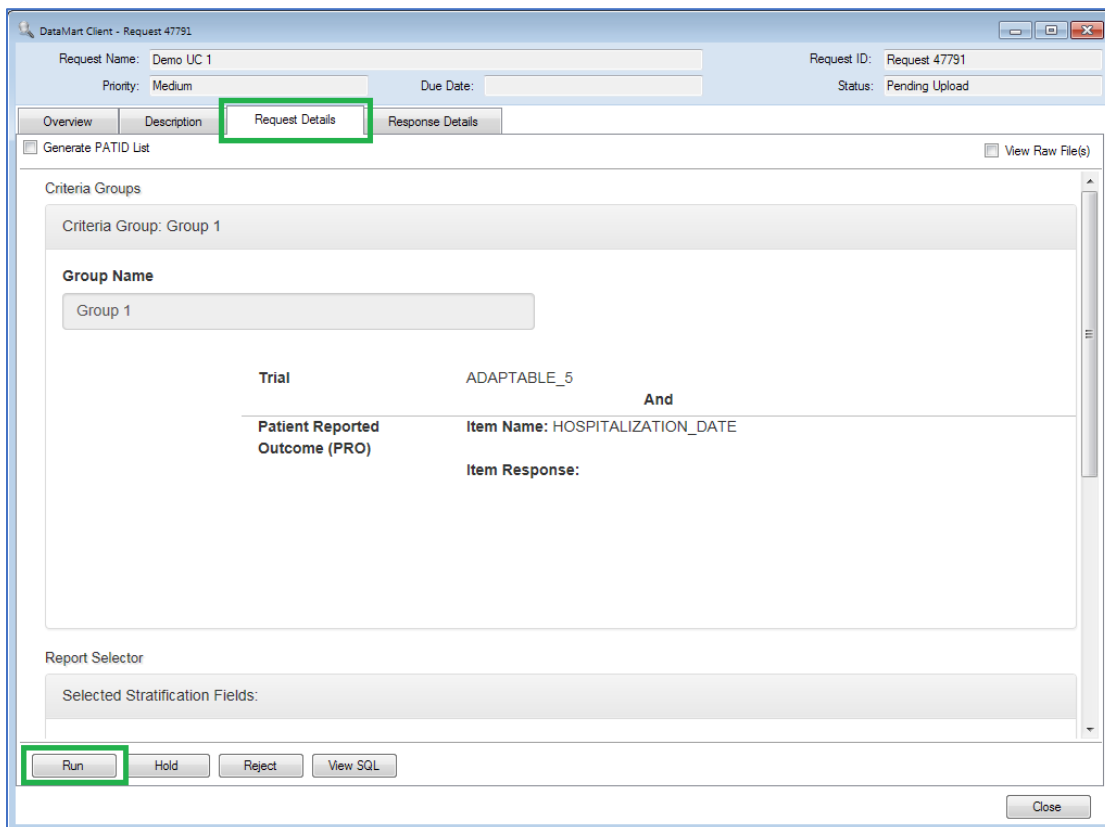
Figure 18. Request Details

4. After a few minutes, the **Response Details** tab should be populated with your results.

5. In the **Response Details** tab, click the **Export Results** button to export to **CSV**. It is important to export results to CSV in order to ensure that the dates can be re-formatted correctly.

Tip: Set the DataMart Client to cache results. This will save the results locally so that the analyst does not need to re-run the query each time they are interested in viewing the results again. Instructions for Caching Results are here: PMN User Guide: Unattended Operations and Cache Settings

Section VI: Viewing the Results in CSV

1. Once the results have been successfully exported to CSV, our first recommendation is to add color to the top row to differentiate between the fields/column names and the results.

2. Next, highlight the ADMIT_DATE field. This field will have a timestamp of "12:00:00 AM" or "00:00". This timestamp is a result of the time window calculation that includes a default time of midnight on all dates. There is likely no timestamps present in your data in the ADMIT_DATE field. Therefore, we recommend removing the time stamp by re-formatting the dates in the field. To re-format, highlight all of the dates in the field and then select the "Short Date" format in the CSV. This will retain the dates, but remove the time.
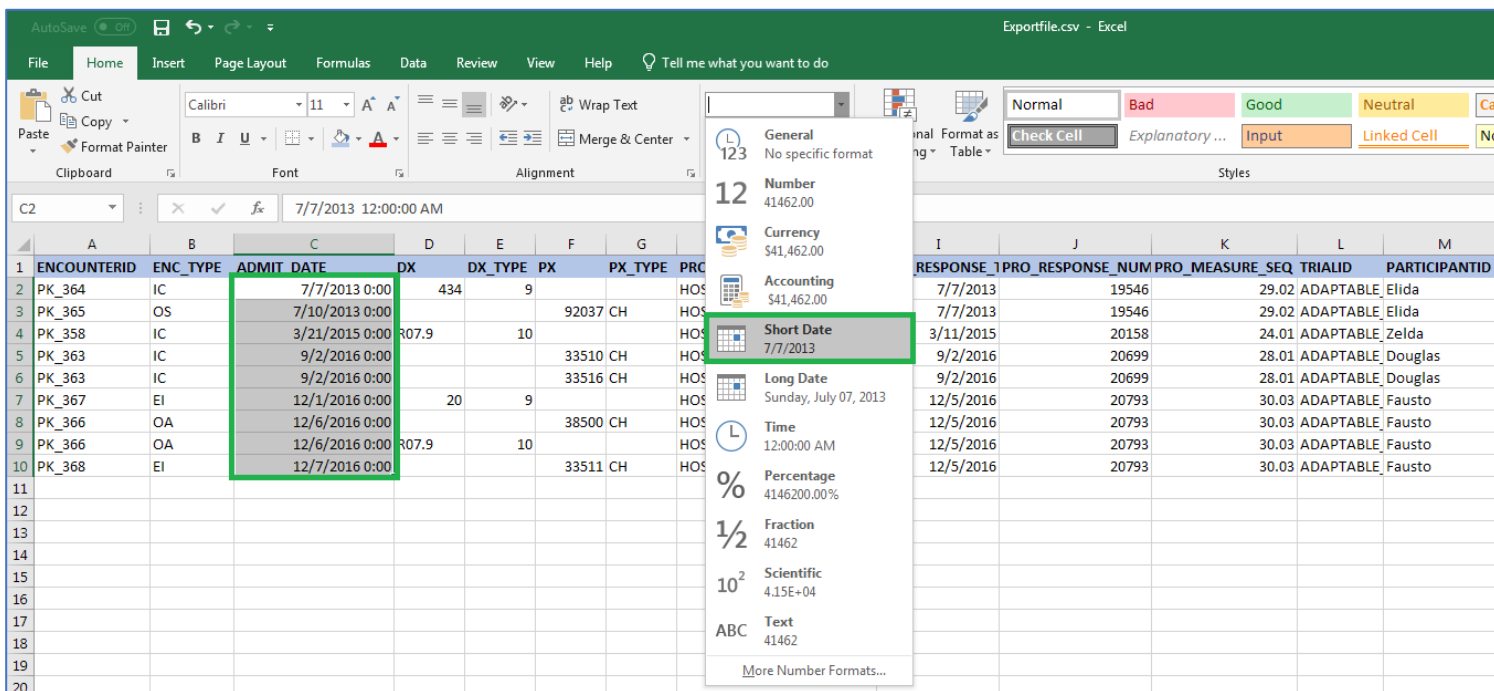
Figure 19. Re-format the ADMIT_DATEs

3. If you intend to review the results in CSV, we would suggest adding the filter option to each field as well. It may be most useful to filter by PARTICIPANTID to view all of the patient reported hospitalizations and encounters for each individual participant.

Section VII: The Result Output

General Guidelines

The MDQ output is shaped by the query criteria and the chosen stratifications. If there are no stratifications, the query will simply return a count of patients in the database that match the criteria, e.g. 405 patients. By adding stratifications based on demographic fields, e.g. Sex and Race, the query will return the patient count stratified by that chosen field. See an example of a MDQ stratified by Sex in Figure 20 below. Figure 20 illustrates that the query resulted in 28 patients that fit the criteria, and of those 28 patients, two have no SEX value present in the database, four have a SEX value of A (Ambiguous), 12 have F (Female), and 10 have the value M (Male).

Overview	Description	Request Details	Response Details	
Sex				Patients
<< NULL >>				2
A				4
F				12
M				10

Figure 20. MDQ Results Stratified by Sex

Trial

As described in Table 2, the Trial stratification will return both the TrialID and the ParticipantID in the results.   Each row will be a unique combination of the two fields.

Patient Reported Outcomes (PRO)

The output can also include more detailed information on the PRO terms of interest to the study team. The **Patient Reported Outcome** stratification will result in four additional fields in the output, as described in Table 2. Each unique combination of those four fields will be listed in the output as its own row. In the case of there being multiple instances of the same combination of fields, the Encounters column in the output indicates the number of records that match the row. In Figure 21 you can see that each row has a count of 1, indicating that there is only one instance of each unique combination in the database. A count of 2 would indicate that there are two rows with the exact same combination for

those four fields. Please note, that capitalization is treated as a difference between fields in Oracle databases, e.g. "CHEST PAIN" is not the same as "chest pain".

PRO_ITEM_NAME	PRO_RESPONSE_TEXT	PRO_RESPONSE_NUM	PRO_MEASURE_SEQ	Encounters
HOSPITALIZATION_DATE	12/1/2013	19693	15.01	1
HOSPITALIZATION_DATE	12/1/2013	19693	15.02	1
HOSPITALIZATION_DATE	2/1/2010	18294	1.01	1
HOSPITALIZATION_DATE	2/20/2019	21600	6.02	1
HOSPITALIZATION_DATE	4/17/2000	14717	2.01	1
HOSPITALIZATION_DATE	5/1/2015	20209	3.01	1
hospitalization_date	5/11/2018	21315	7.02	1
HOSPITALIZATION_DATE	6/7/2019	21707	6.01	1
HOSPITALIZATION_DATE	7/12/2016	20647	14.01	1
HOSPITALIZATION_DATE	7/13/2015	20282	25.03	1
HOSPITALIZATION_DATE	8/15/2013	19585	9.02	1
HOSPITALIZATION_EVENT	CABG	<< NULL >>	8.01	1
HOSPITALIZATION_EVENT	CHEST PAIN	<< NULL >>	9.02	1
HOSPITALIZATION_EVENT	MAJOR BLEEDING	<< NULL >>	2.01	1
HOSPITALIZATION_EVENT	MI	<< NULL >>	1.01	1
HOSPITALIZATION_EVENT	mi	<< NULL >>	25.03	1
HOSPITALIZATION_EVENT	MI	<< NULL >>	7.02	1
HOSPITALIZATION_EVENT	TIA	<< NULL >>	15.01	1
LOCATION	Vanderbilt Medical	<< NULL >>	9.02	1
LOCATION	VUMC	<< NULL >>	7.02	1

Figure 21. Encounters Column Counts the Number of Instances for Each Unique Combination of the Four Fields Listed

For more information on each of the columns returned by the Patient Reported Outcome stratification, please see #3 Patient Reported Outcome Fields in the section below.

Patient Reported Outcome Encounters

The **Patient Reported Outcome Encounters** stratification indicates the temporal event functionality and that the output will include both PRO and EHR information, as described in Section III. When the stratification is added, the temporal event functionality is automatically added to the query as well. As a result of the query structure, **each row will be unique** and therefore there is no "Count" column present in this case. If a query has Patient Reported Outcome Encounters as the only stratification, the results will be grouped by 11 different columns. As illustrated in Figure 22, there are three fields from the ENCOUNTER table, two fields from each the Diagnosis and Procedures tables, and four fields from the PRO_CM table.

ENCOUNTERID	ENC_TYPE	ADMIT_DATE	DX	DX_TYPE	PX	PX_TYPE	PRO_ITEM_NAME	PRO_RESPONSE_TEXT	PRO_RESPONSE_NUM	PRO_MEASURE_SEQ
PK_364	IC	7/7/2013	434	9			HOSPITALIZATION_DATE	7/7/2013	19546	29.02
PK_365	OS	7/10/2013			92037	CH	HOSPITALIZATION_DATE	7/7/2013	19546	29.02

Encounter   Diagnosis and Procedures   Patient Reported Outcomes (PRO_CM)

Figure 22. Tables and Fields Represented by the Patient Reported Outcome Encounters Stratification

1. **Encounter columns:** The three Encounter columns depict the encounter ID (EncounterID), encounter setting (ENC_TYPE) and admit date (ADMIT_DATE) for each encounter match with the patient reported index event.

2. **Diagnosis and Procedure columns:** The DX and DX Type columns include the diagnosis code and diagnosis code type, whereas the PX and PX Type columns include the procedure code and code type associated with each matching encounter. There will only be one code per row—this means that if an encounter has three codes, then there will be three rows for that encounter.

3. **Patient Reported Outcome fields:** The PRO fields in the output detail the patient reported events.   All of the below columns are also present when the Patient Reported Outcome stratification is present.
    - The PRO_ITEM_NAME column describes the type of PRO information that is being queried, e.g. hospitalization dates, hospitalization events, facility, vaccination date, etc.

    - The PRO_RESPONSE_TEXT field will be populated with the patient reported response to the item described in the PRO_ITEM_NAME field. If the PRO_ITEM_NAME field is populated with HOSPITALIZATION_DATE or Vaccination date, then the response will include the patient reported date, e.g. "7/7/2013", "February 2018", etc.

    - The PRO_RESPONSE_NUM column will contain the corresponding SAS date for the patient's reported date. This SAS date should be equivalent to the date populated in the PRO_RESPONSE_TEXT field. It is important to note that the SAS date must be populated in order to calculate the time windows. If a participant is missing a SAS date, then they will not be returned in the results that include temporal events.

    - The PRO_MEASURE_SEQ column is an arbitrary ID or sequence that is used to link item responses that are associated to the same hospitalization record. In studies, a participant will likely report various pieced of information. For example, in the ADAPTABLE study participants were asked to report aspects of a hospitalization, e.g. date, reason (event), hospital location, etc. All related

information should have the same PRO_MEASURE_SEQ to indicate the association to the same patient reported hospitalization.

When both the Trial and Patient Reported Outcome Encounters stratifications are a part of the query, there will be 13 columns.

ENCOUNTERID	ENC_TYPE	ADMIT_DATE	DX	DX_TYPE	PX	PX_TYPE	PRO_ITEM_NAME	PRO_RESPONSE_TEXT	PRO_RESPONSE_NUM	PRO_MEASURE_SEQ	TRIALID	PARTICIPANTID
PK_364	IC	7/7/2013	434	9			HOSPITALIZATION_DATE	7/7/2013	19546	29.02	ADAPTABLE_5	Elida
PK_365	OS	7/10/2013			92037	CH	HOSPITALIZATION_DATE	7/7/2013	19546	29.02	ADAPTABLE_5	Elida

Encounter   Diagnosis and Procedures   Patient Reported Outcomes (PRO_CM)   Trials (PCORNET_Trial)

Figure 23. Tables and Fields Represented by the Patient Reported Outcomes Encounters and Trial Stratifications

Section VIII: Interpreting the Results

The quickest way to determine what PRO information the MDQ is querying is to review the PRO_ITEM_NAME field. The PRO_ITEM_NAME field will indicate the PRO information of interest in the query, e.g. the HOSPITALIZATION_DATE or the HOSPITALIZATION_EVENT (see Figure 24).

DX	DX_TYPE	PX	PX_TYPE	PRO_ITEM_NAME	PRO_RESPONSE_TEXT	PRO_RESPONSE_NUM	PRO_MEASURE_SEQ	TRIALID
<< NULL >>	<< NULL >>	<< NULL >>	<< NULL >>	HOSPITALIZATION_EVENT	CABG	<< NULL >>	28.01	ADAPTABLE_5
<< NULL >>	<< NULL >>	<< NULL >>	<< NULL >>	HOSPITALIZATION_EVENT	NONSURGICAL PCI	<< NULL >>	29.03	ADAPTABLE_5
<< NULL >>	<< NULL >>	<< NULL >>	<< NULL >>	HOSPITALIZATION_EVENT	STROKE	<< NULL >>	29.01	ADAPTABLE_5
<< NULL >>	<< NULL >>	<< NULL >>	<< NULL >>	HOSPITALIZATION_EVENT	CABG	<< NULL >>	30.03	ADAPTABLE_5
434	09	<< NULL >>	<< NULL >>	HOSPITALIZATION_DATE	7/7/2013	19546	29.02	ADAPTABLE_5
<< NULL >>	<< NULL >>	92037	CH	HOSPITALIZATION_DATE	7/7/2013	19546	29.02	ADAPTABLE_5
<< NULL >>	<< NULL >>	33510	CH	HOSPITALIZATION_DATE	9/2/2016	20699	28.01	ADAPTABLE_5
<< NULL >>	<< NULL >>	33516	CH	HOSPITALIZATION_DATE	9/2/2016	20699	28.01	ADAPTABLE_5
<< NULL >>	<< NULL >>	38500	CH	HOSPITALIZATION_DATE	12/5/2016	20793	30.03	ADAPTABLE_5
R07.9	10	<< NULL >>	<< NULL >>	HOSPITALIZATION_DATE	12/5/2016	20793	30.03	ADAPTABLE_5
<< NULL >>	<< NULL >>	33511	CH	HOSPITALIZATION_DATE	12/5/2016	20793	30.03	ADAPTABLE_5

Figure 24. PRO_ITEM_NAME Column determines the PRO Information of Interest

The Temporal Event panel specifies an Index Event. The output will include rows that represent matches between the Index Event and Encounters. Only the Index Event, i.e. the patient-reported index date, will be linked to the Encounter, Diagnosis, and Procedure information in the output. Any other rows that contain other patient-reported outcomes, will have NULL values in those fields. Each row in the output for the ADAPTABLE Supplement Use Cases in Section IV represents either 1) a patient reported index event (e.g. hospitalization date) and Encounter match or 2) a patient reported hospitalization event. Use Case 1 queries will only

have rows for the first type, whereas Use Case 2 queries will include both the hospitalization date matches and the hospitalization events.

Rows with Index Events and Encounters

Each row is a unique result that represents a match between a patient-reported index date and an encounter that falls within the specified time window in the query. The output also includes the diagnosis and procedure codes associated with each encounter. Each of these rows include both the PRO information as well as the EHR information. In some instances there will be multiple rows for one index event, or in the case of ADAPTABLE, multiple rows for one patient reported hospitalization. More than one row for a single index event indicates that either A) more than one encounter is matched **OR** B) the encounter contains more than one diagnosis/procedure code. In order to determine whether a single index event has more than one row, review the PRO_MEASURE_SEQ column. As described in Section VII , the PRO_MEASURE_SEQ field should be populated in a way that links all of the details for a single patient-reported index event, e.g. the hospitalization date, event/reason, location, etc. Two patient reported index events **should not** have the same PRO_MEASURE_SEQ. Since each event has a unique PRO_MEASURE_SEQ, if there are multiple rows with the same PRO_MEASURE_SEQ it is an indicator that they all belong to the same patient-reported index event. Figure 25 illustrates both scenarios mentioned. First, the green text highlights a scenario where one patient-reported hospitalization is matched with two separate encounters. Next, the red text demonstrates a scenario where one hospitalization is matched to one encounter. There are two rows in the red box due to the encounter containing two codes, one diagnosis and one procedure code. As described in Section VII, each code will have its own row.

Matching numbers indicated that the rows belong to the same patient reported hospitalization

Two different encounters (PK_364 and PK_365) match the same patient reported hospitalization (as denoted by PRO_MEASURE_SEQ=29.02) on 7/7/2013

ENCOUNTERID	ENC_TYPE	ADMIT_DATE	DX	DX_TYPE	PX	PX_TYPE	PRO_ITEM_NAME	PRO_RESPONSE_TEXT	PRO_RESPONSE_NUM	PRO_MEASURE_SEQ	TRIALID	PARTICIPANTID
PK_363	IC	9/2/2016 12:00:00 AM	<< NULL >>	<< NULL >>	33510	CH	HOSPITALIZATION_DATE	9/2/2016	20699	28.01	ADAPTABLE_5	Douglas
PK_363	IC	9/2/2016 12:00:00 AM	<< NULL >>	<< NULL >>	33516	CH	HOSPITALIZATION_DATE	9/2/2016	20699	28.01	ADAPTABLE_5	Douglas
PK_364	IC	7/7/2013 12:00:00 AM	434	09	<< NULL >>	<< NULL >>	HOSPITALIZATION_DATE	7/7/2013	19546	29.02	ADAPTABLE_5	Elda
PK_365	OS	7/10/2013 12:00:00 AM	<< NULL >>	<< NULL >>	92037	CH	HOSPITALIZATION_DATE	7/7/2013	19546	29.02	ADAPTABLE_5	Elda
PK_367	EI	12/1/2016 12:00:00 AM	0020	09	<< NULL >>	<< NULL >>	HOSPITALIZATION_DATE	12/5/2016	20793	30.03	ADAPTABLE_5	Fausto
PK_366	OA	12/6/2016 12:00:00 AM	<< NULL >>	<< NULL >>	38500	CH	HOSPITALIZATION_DATE	12/5/2016	20793	30.03	ADAPTABLE_5	Fausto
PK_366	OA	12/6/2016 12:00:00 AM	R07.9	10	<< NULL >>	<< NULL >>	HOSPITALIZATION_DATE	12/5/2016	20793	30.03	ADAPTABLE_5	Fausto
PK_368	EI	12/7/2016 12:00:00 AM	<< NULL >>	<< NULL >>	33511	CH	HOSPITALIZATION_DATE	12/5/2016	20793	30.03	ADAPTABLE_5	Fausto
PK_358	IC	3/21/2015 12:00:00 AM	R07.9	10	<< NULL >>	<< NULL >>	HOSPITALIZATION_DATE	3/11/2015	20158	24.01	ADAPTABLE_5	Zelda

For the same hospitalization (as denoted by PRO_MEASURE_SEQ 30.3), there are two rows for the same Encounter (PK_366). Encounter PK_366 has two rows because there is both a procedure and diagnosis code present in this encounter. Each code has its own row

Figure 25. Multiple Rows for One Patient Reported Hospitalization Date

Rows With Non-Index Event Patient Reported Outcomes

In the case of other patient-reported information that is not the index date in the temporal event, e.g. the patient reported hospitalization event/reason in Use Case 2 in Section IV, there is no link to the EHR data. Any PRO term entered in the query criteria outside of the temporal events panel are only linked to the corresponding Trial and ParticipantID in the output. The remaining EHR fields will be NULL for these rows.

The PRO fields detail the patient reported event in a similar way as the patient reported dates described in Section VII.

- The PRO_ITEM_NAME field will indicate that the row is for a different item that is not the Index Event. For example, the ADAPTABLE Supplement study had hospitalization dates as the Index, and hospitalization events as non-index PRO information. See Figure 24 for an output that includes both events and dates.

- The patient reported response for the non-index PRO item can be found in the PRO_RESPONSE_TEXT field, e.g. Chest Pain, MI, etc.

- As mentioned in Section VII, the PRO_MEASURE_SEQ field is used for indicating a single reported index event, e.g. hospitalization. The various components of each index event (date, reason, location) should have a matching PRO_MEASURE_SEQ. Once the PRO_MEASURE_SEQ field is populated correctly, this field can be used to determine which non-index items are linked to which

index dates. In the screenshot below ([Figure 26](#)), the rows with the same PRO_MEASURE_SEQ can be grouped as a single patient-reported hospitalization. The participant has two separate PRO_MEASURE_SEQs, as indicated by 22.01 in red and 22.02 in blue. In each of these two groups, it is clear which hospitalization event is linked to which hospitalization date by matching the PRO_MEASURE_SEQ.

ENCOUNTERID	ENC_TYPE	ADMIT_DATE	DX	DX_TYPE	PX	PX_TYPE	PRO_ITEM_NAME	PRO_RESPONSE_TEXT	PRO_RESPONSE_NUM	PRO_MEASURE_SEQ	TRIALID	PARTICIPANTID
							HOSPITALIZATION_EVENT	STROKE		22.01	ADAPTABLE	W12
PK_190	OS	9/1/2018	434.91	9			HOSPITALIZATION_DATE	9/1/2018	21428	22.01	ADAPTABLE	W12
							HOSPITALIZATION_EVENT	CABG		22.02	ADAPTABLE	W12
PX_234	IP	9/15/2018			33511	CH	HOSPITALIZATION_DATE	9/14/2018	21441	22.02	ADAPTABLE	W12

Figure 26. PRO_MEASURE_SEQ Field

In the red box, the participant reported that they were hospitalized on 9/1/2018 for a STROKE. The Encounter that was matched with the hospitalization date is for the same day 9/1/2018 and has a diagnosis code associated with it. The diagnosis code is the ICD-9 code for Stroke. In this particular instance, it appears that there is complete concordance between what the patient reported and what is recorded in the EHR.

Contact Us

If there are any issues, please enter a service desk ticket here: https://popmednet.atlassian.net/servicedesk/customer/portal/1

Disclaimer