

NIH Collaboratory Distributed Research Network: PopMedNet-i2b2 Integration Proof of Concept Report

PROJECT TITLE: Health Care Systems Research Collaboratory Coordinating Center

PIs: Lesley Curtis, PhD and Adrian Hernandez, MD

GRANT: U54AT007748

PROJECT DATES: 09.30.2012 – 08.31.2017

Prepared by: Jeffrey Brown, Jeffrey Klan, Shawn Murphy, Bruce Swan, and the NIH Collaboratory Electronic Health Records Core

Report Summary

Background

The Electronic Health Records (EHR) Core has created the NIH Collaboratory Distributed Research Network (DRN), a new resource that enables investigators to collaborate in the use of electronic health data, while also safeguarding protected health information and proprietary health system data. It supports both single- and multi-site research programs. The Network's distributed querying capabilities reduce the need to share confidential or proprietary data by enabling authorized researchers to send queries to health system collaborators. Queries are typically in the form of computer programs that a data partner can execute on a pre-existing dataset. The data partner can review and return the query results, rather than the data itself. This form of remote querying reduces legal, regulatory, privacy, proprietary, and technical barriers associated with data sharing for research. The NIH Collaboratory DRN uses an open-source networking software application – PopMedNet™ – to manage network operations and governance, distribute queries, and display results.¹

Proof of Concept Overview

To enhance current capabilities for distributed querying, the EHR Core conducted a pilot project to evaluate how to enable distributed querying of organizations using Informatics for Integrating Biology and the Bedside (i2b2) as their internal and querying data resource.² i2b2 is widely-used by academic medical centers and others for a range of data querying activities. Integration of i2b2 with PopMedNet could substantially expand the data resources available within the NIH Collaboratory DRN. A pilot project conducted as part of the ONC Standards & Interoperability Framework Query Health Initiative illustrated the feasibility of integrating i2b2 and PopMednet,^{3,4} this pilot extends that work.

Researchers at Partners Healthcare and Mass General Hospital in Boston collaborated closely with the EHR Core and our technology vendor, LincolnPeak Partners, to design and implement this Proof of Concept (POC). The Partners Healthcare investigators are the leading experts in the i2b2 software development, data model, and implementations.

The POC was conducted in two phases: 1) a design phase; and 2) an implementation phase. The design phase included identification of NIH Collaboratory DRN system and querying requirements and an assessment of approaches for querying i2b2 “nodes” within the constraints of the querying and system requirements. Based on the design requirements, the POC implementation used an existing PopMedNet query interface (the ESPnet Query Builder⁵) to create a query and distribute it to both an ESPnet⁶ site

¹ www.popmednet.org

² www.i2b2.org/

³ www.youtube.com/watch?v=sqDAo6E-b1o&feature=youtu.be

⁴ Klann JG, Buck MD, Brown JS, et al. Query Health: standards-based, cross-platform population health surveillance. *Journal of the American Medical Informatics Association* : JAMIA. 2014;21(4):650-656.

⁵ Vogel J, Brown JS, Land T, et al. MDPHnet: Secure, Distributed Sharing of Electronic Health Record Data for Public Health Surveillance, Evaluation, and Planning. *American Journal of Public Health*: December 2014, Vol. 104, No. 12, pp. 2265-2270.

⁶ esphealth.org

and an i2b2 site for local execution and response. Test data were used for the POC. This approach used a menu-driven interface to create a simple query for distribution, and a query adapter that transformed the query to execute against the two different data models (i2b2 and ESPnet) and return results in a standard format.

The query was successfully distributed to both POC sites, translated using newly developed “model adapters” to execute against the local data resource (i2b2 and ESPnet), and results were returned so that individual site results and aggregated results were available to the requester.

Phase 1 – Design Summary

The design phase involved requirements gathering to explore various options, use cases and technical approaches for querying i2b2 sites. Phase 1 activities were to:

- ✓ Explore options for querying i2b2 sites, including existing approaches such as SHRINE⁷
- ✓ Describe current work related to integration of i2b2 and PopMedNet
- ✓ Gather requirements from key stakeholders
- ✓ Research the i2b2 data schema to determine feasibility for using PopMedNet native queries against the i2b2 data model
- ✓ Document the potential queries for the POC
- ✓ Determine final list of requirements
- ✓ Describe technical design and most efficient approach for the POC
- ✓ Identify use cases
- ✓ Create a final design document for the implementation approach and POC plan

We began with high-level assumption that the integration should allow simple, menu-driven querying using an existing query interface, be minimally intrusive to data partners, require minimal software development, and not require any special analytic software. A straw-man reference model based on use of a native PopMedNet query (the ESPnet Query Builder) was developed, and several different approaches to distributed querying using that reference model were explored. We decided on a design that involved 1) building a “PopMedNet-i2b2 Model Adapter” to translate the ESPnet Query Builder query into a form that could be executed against an i2b2 schema, 2) querying directly against the i2b2 database rather than using the i2b2 software hive for query execution, and 3) illustrate how a single query could be executed against 2 different data models (i2b2 and ESPnet) and return results in an identical format.

Phase 2 – Implementation Summary

Phase 2 implementation activities were to:

- ✓ Implement an i2b2 model adapter that executes queries against the i2b2 data model

⁷ Weber GM, Murphy SN, McMurry AJ, *et al.* The Shared Health Research Information Network (SHRINE): A Prototype Federated Query Tool for Clinical Data Repositories. *J Am Med Inform Assoc* 2009;**16**:624–30.

- ✓ Modify the existing PopMedNet request composer that will package the request and serialize it into the appropriate XML for translation
- ✓ Implement a PopMedNet model processor responsible for executing the PopMedNet query against the i2b2 database
- ✓ Deploy the POC to the NIH Collaboratory DRN staging environment
- ✓ Create an i2b2 testing environment for executing the POC queries
- ✓ Evaluate POC results, document lessons learned and open issues

The POC was successfully completed, showing how a single menu-driven query could generate a serialized query for distribution to 2 different sites with different data models, have the query translated and processed locally, and compatible results returned for analysis.

General Issues and Challenges

The POC illustrated how a native PopMedNet query built using a simple menu-driven query interface can be distributed to 2 sites, transformed to execute against the local data model, and return identically formatted results.

Use of a common data model for multi-site distributed querying is a necessary but not sufficient condition. During discussion with our design partners (see Acknowledgements), it was noted that, although the existing i2b2 sites use virtually the same i2b2 schema, there is likely substantial variation in how sites populate and use the schema. In particular, i2b2 installations can use:

- ✓ Different ontology maps
- ✓ Different value sets (eg, sex, race, care setting, medical specialty)
- ✓ Different database managers/vendors for their i2b2 databases
- ✓ Different decisions regarding what data from the local clinical data warehouse to include in the i2b2 database

The POC required normalization of all relevant query variables across the two data models (i2b2 and ESPnet). Specifically, mappings were needed for race, gender, date formats, ICD9 names/ descriptions, and age ranges. Additional mappings would be needed to incorporate additional query data elements. The mappings must be completed for each site, and site data must be monitored on an ongoing basis to account for changes in data capture. Any large-scale implementation of an i2b2-based (or any other data model) network will need to consider solutions for ongoing data curation and that can more easily resolve variation across i2b2 sites. Further, the POC focused on a SQL server based i2b2 instance. Use of other systems such as Oracle with i2b2 (instead of SQL Server), will require additional modifications to create an Oracle-specific translator (i.e., PopMedNet Model Adapter).

Future Work

Any approach to cross-site i2b2 queries requires standardization of data, so that meaning is consistent at all locations. The approach in this work was to create several local mapping tables within an i2b2 node. In the classic i2b2 implementation, the i2b2 ontology services provide natural mechanisms for

local mapping. i2b2 natively uses a concept path as its primary key for all ontology elements to uniquely identify queryable elements.⁸ The local meaning of that concept path (such as the coded value it represents) can be defined by the local node -- only the concept path needs to remain constant across sites for multisite queries. If a mapping between sites is to be undertaken, it is by mapping the equivalent paths at the two sites.⁹ This is the foundation for cross-site querying via SHRINE networks,^{10,11} several of which exist across the country including a new initiative to develop a national network for clinical trial recruitment (ACT). Also, nearly two dozen non-SHRINE i2b2 nodes are presently adopting a standard ontology based on creating paths for the PCORnet common data model. Harnessing these multiple networks of standard terminologies is a powerful opportunity.

Additionally, simple queries could be executed using the existing i2b2 API, rather than through direct database access. To support interoperability, our native query format could be translated into i2b2 query format. We previously used this approach in Query Health.^{12,13} This provides several advantages that are more scalable and less invasive than the method in this work. This would allow immediate use of other querying capabilities of i2b2 without additional database changes; specifically, modifiers (e.g., medication route) and values (e.g., specific weight ranges) with automatic unit normalization.

⁸ Murphy SN, Weber G, Mendis M, *et al.* Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *J Am Med Inform Assoc* 2010;**17**:124–30.

⁹ Murphy S. Data warehousing for clinical research. In: *Encyclopedia of Database Systems*. Springer Publishing Company, Incorporated 2009.

¹⁰ McMurry AJ, Murphy SN, MacFadden D, *et al.* SHRINE: Enabling Nationally Scalable Multi-Site Disease Studies. *PLoS ONE* 2013;**8**:e55811.

¹¹ Weber GM, Murphy SN, McMurry AJ, *et al.* The Shared Health Research Information Network (SHRINE): A Prototype Federated Query Tool for Clinical Data Repositories. *J Am Med Inform Assoc* 2009;**16**:624–30.

¹² Klann JG, Buck MD, Brown JS, *et al.* Query Health: standards-based, cross-platform population health surveillance. *J Am Med Inform Assoc* 2014;**21**(4):650-656.

¹³ Klann GJ, Murphy NS. Computing Health Quality Measures Using Informatics for Integrating Biology and the Bedside. *J Med Internet Res* 2013;**15**:e75.

DETAILED REPORT OF FINDINGS

Phase 1: Design

The key activities for Phase 1 included selection of a query composer (ie, a menu-driven interface to build a request) and an approach for request execution. PopMedNet™ (PMN) has a set of query composers that could be used for the POC, or a new composer could be built, if needed.

Query Composer

PMN has a number of query composers that could be used for the POC. Each is described at www.popmednet.org. Currently available PMN query composers are:

- ✓ FDA Mini-Sentinel Summary Table Queries
- ✓ ESPnet Query Builder (ICD-9-CM Diagnosis codes)
- ✓ ESPnet Query Composer
- ✓ SPAN Query Builder

Based on conversations with the EHR Core and members of the i2b2 design team, the **ESPnet ICD-9-CM Diagnosis Code Query Builder** query composer was selected to formulate queries that can be routed to i2b2 DataMarts for execution (a “DataMart” is the PMN term used to denote the local i2b2 database). The ESPnet ICD-9-CM Diagnosis Code Query Builder is used by the MDPHnet¹⁴ project and is based on an adaptation of the ESP data model that makes it more consistent with the FDA Mini-Sentinel Common Data Model.¹⁵ The ESPnet data model was designed to capture EHR data and contains similar data elements as a typical i2b2 installation.

Request Execution

Once a query composed by the ESPnet ICD-9-CM Diagnosis Code Query Builder is received by the local DataMart Client (the DataMart Client is part of the PMN software that handles routing of requests from the secure portal to the local DataMart), there are two ways the query can be executed against the local data source:

- ✓ Convert the query settings to an i2b2 data server XML message and pass it to the i2b2 data server hive for execution (ie, use i2b2 software to execute the query).
- ✓ Convert the query setting to a SQL statement using the i2b2 schema and execute the SQL query directly against the i2b2 database. This approach leverages the PopMedNet modular design by creating a “PMN i2b2 Model Adaptor” that can be updated and managed independently of the i2b2 hive software. This mitigates risk related to changes on the i2b2 hive.

After initial discussion, it was determined the best approach for the POC would be to execute the query directly against the i2b2 data. This approach requires creation of a PMN i2b2 Model Adaptor that

¹⁴ mehi.masstech.org/what-we-do/hie/mdphnet

¹⁵ http://mini-sentinel.org/data_activities/distributed_db_and_data/details.aspx?ID=105

converts the request settings that are passed to the DataMart Client as an XML file to SQL for execution directly against the i2b2 database. By using an existing query composer, this approach allowed us to extend the POC to investigate how to issue a single query and translate it for execution against 2 different data models (i2b2 and ESPnet). Finally, this approach represented a new mechanism for facilitating distributed querying with i2b2, making it an optimal target for a POC.

Phase 2: Implementation

As described above, the implementation phase required development of a PMN i2b2 Model Adapter used to translate an ESPnet ICD-9-CM Diagnosis Code Query Builder query into SQL for execution against i2b2 DataMarts. The Appendix contains details of the query interface and response process.

Ontology Mapping

Use of the PMN i2b2 Model Adapter bypasses the i2b2 ontology and generates SQL to execute directly against the main i2b2 data tables:

- ✓ OBSERVATION_FACT – table used to record an observation within an encounter and contains the codes denoting the observation type, in our case ICD9 diagnosis.
- ✓ PATIENT_DIMENSION – table used to provide the patient demographics used to filter patient encounters and stratify results.
- ✓ VISIT_DIMENSION – table used to provide record encounters containing one or more observations recorded within a visit. This table also has location information and in-patient/out-patient status.

The native i2b2 query composer uses an ontology tree to formulate an i2b2 request. The ontology manifests itself as a string field containing a path describing the problem or condition that is being queried. The ontology path value is used to find a “concept code” used by the observation fact table to record encounters, for instance an encounter that results in a diabetes diagnosis using a 250xx ICD-9-CM diagnosis code.

The PMN i2b2 Model Adapter does not use the CONCEPT_DIMENSION table and instead executes directly against the OBSERVATION_FACT table to find all instances of ICD9 encounters based on the CONCEPT_CD format that identifies ICD9 diagnosis codes.

It is possible that some sites may use a different prefix or other coding structure to represent ICD9 diagnosis codes. If this is encountered, the existing PMN model adapter will be revised or a new adapter developed to accommodate alternate schemes. Additionally, as described in the next few sections, a set of PMN code look up tables could be deployed to map between PMN and coding structures found in various i2b2 installations.

PMN – i2b2 Race Code Mapping

ESP Query Builder and other tools use different codes sets to denote race. The model adapter needs to generate SQL that can be used by a specific i2b2 site, and, as such, needs to map between the code set

used by PMN to denote race and the underlying data source codes used to denote race. In our POC, the ESPnet query uses the following race categories:

- ✓ 0 – Unknown,
- ✓ 1 – American Indian or Alaska Native
- ✓ 2 – Asian
- ✓ 3 – Black or African American
- ✓ 4 – Native Hawaiian or Other Pacific Islander (NHOPI)
- ✓ 5 – White

In order to use the existing PMN query composer, the race codes found in the i2b2 installation need to be mapped to the ESPnet query code set. There are several approaches to resolve this issue. We could use an inline SQL case statement to translate the PMN race codes to those used in i2b2, or perform the translation using a local mapping table. The latter approach was chosen given its efficiency and flexibility. A new i2b2 table, called PMN_I2B2_RACE_CODE_LOOKUP was added to the i2b2 installation that contains entries used to map race codes used by a specific i2b2 instance to the PopMedNet race schema. The table has the following fields:

Field	Type	Description
RACE_CD (PK)	Varchar(50), not null	i2b2 race codes
RACE_CODE	int, not null	PMN ESP ICD9 race codes

Part of the implementation was to develop the data definition language (DDL) needed to add this table to SQL Server and SQL script used to load the table.

i2b2 - PMN Race Code Mapping

The above table handles translating PMN race codes into i2b2 race codes used by a site; however we also need to map an i2b2 race code in the result to the corresponding PMN race code when results are stratified by race. This requires another look up table, called “I2B2_PMN_RACE_CODE_LOOKUP” which maps codes from i2b2 to PMN ESPnet ICD-9-CM Query Builder race strata. The table has the following fields:

Field	Type	Description
RACE_CD (PK)	Varchar(50) , not null	i2b2 race codes
RACE_CHAR	iVarchar(50) , not null	PMN ESPnet ICD9 race result set values

Part of the implementation was to develop the DDL needed to add this table to SQL Server and SQL script used to load the table.

Observation Date Mapping

The ESPnet Query Builder uses SAS® dates to represent the observation periods. These dates are day offsets from the base date, January 1, 1960. These dates need to be converted to the date type defined in the i2b2 OBSERVATION_FACT table that records the encounter date. There are several ways to perform this conversion. These dates can be converted to the SQL statement using date functions, or could map each SAS date to the corresponding date type used by the i2b2 deployment. The former approach was taken in the POC implementation. The SQL Server DATE function is used to compute the date for each of the encounters in the OBSERVATION_FACT table. The following is a sample of that function: `"CONVERT(date, f.START_DATE) >= DATEADD(DAY, 14610, CONVERT(date, '1/1/1960'))"`

Sex Code Set Mapping

The ESPnet query and other tools use different code sets to denote sex. We needed to map the ESPnet sex code set to the code set used by the i2b2 deployment. There are several approaches to resolve this, such as adding a lookup table similar to the Race code lookup table above, or translation in the SQL statement. No action was necessary in POC i2b2 sample database given both databases used "M" and "F" to represent sex. Additional mapping would be needed to handle other possible values, such as unknown, blank, and transgender.

ICD9 Code Names/Description and Precision Mapping Support

The ESPnet query allows the user to stratify the results by ICD9 code, including mapping lower level (more granular) codes to higher level codes. For instance the user may specify a set of 3, 4, and/or 5 digit ICD9 diagnosis codes and specify the results be stratified by the corresponding 3 digit code ("250.5", "250.50", "250.51", etc. are aggregated as "250"). To enable stratification of results from i2b2 DataMarts, we need a way to map the higher precision codes to lower precision codes.

Secondly, code descriptions returned from all DataMarts should match in order to compare and aggregate results correctly. This allows the query to be federated across sites that use various code sets that in many cases use similar to but the exact same code names/descriptions.

These requirements were resolved through the use of a new PMN I2B2 ICD9 mapping table, called "PMN_ICD9_CODE_LOOKUP", which would need to be deployed at each PMN i2b2 site. The table has the following fields:

Field	Type	Description
CONCEPT_CD (PK)	Varchar(50)	i2b2 ICD9 codes recorded in the CONCEPT_CD field of the OBSERVATION_FACT table
CODE_3DIG	Varchar(50), null	PMN 3 digit code value
NAME_3DIG	Varchar(500), null	PMN 3 digit code name
CODE_4DIG	Varchar(50), null	PMN 4 digit code value
NAME_4DIG	Varchar(500), null	PMN 4 digit code name
CODE_5DIG	Varchar(50), null	PMN 5 digit code value

NAME_5DIG	Varchar(500), null	PMN 5 digit code name
-----------	--------------------	-----------------------

Part of the implementation was to develop the DDL required to add this table to SQL Server and SQL script used to load the table.

Age Range Mapping Support

Displaying results using age range stratifications requires mapping an age at encounter in a result record to an ESPnet ICD9 Diagnosis Age Range value. The ESPnet ICD9 Diagnosis Query Builder uses two different age range sets: 5 year age ranges, and 10 year age ranges.

There are two ways to implement the mapping; use a mapping table that maps an encounter age to the value in the rage set specified in the query, or perform the mapping as a case statement within the SQL code. Before we can perform the mapping, we first need to compute the patient's age at the time of the encounter based on the encounter date "START_DATE" in the OBSERVATION_FACT table and the patient's birth date "BIRTH_DATE" recorded in the PATIENT_DIMENSION table. This was performed using an inline SQL function: "DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766".

Next we used another inline function to determine the age range as follows:

```
CASE
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 0 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 5 THEN '00-04'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 5 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 10 THEN '05-09'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 10 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 15 THEN '10-14'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 15 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 20 THEN '15-19'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 20 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 25 THEN '20-24'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 25 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 30 THEN '25-29'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 30 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 35 THEN '30-34'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 35 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 40 THEN '35-39'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 40 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 45 THEN '40-44'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 45 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 50 THEN '45-49'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 50 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 55 THEN '50-54'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 55 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 60 THEN '55-59'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 60 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 65 THEN '60-64'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= 65 AND
      DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 < 70 THEN '65-69'
```

```

WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 70 AND
    DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 75 THEN '70-74'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 75 THEN
    '75+'
END as "5 Year Age Group"

```

Another approach to implement this function is to use a mapping table. A challenge with this approach is that it requires a nested query to first produce the encounter age which is then used to JOIN with the mapping table. During implementation, the mapping table called “PMN_AGE_GROUP_LOOKUP”, was developed but not used to perform the function until the relative performance of one approach over the other using large data sets can be determined. The table is as follows:

Field	Type	Description
AGE_AT_ENCOUNTER (PK)	int, not null, (PK)	Encounter age computed in inner query based on the START_DATE in the OBSERVATION_FACT table and the BIRTH_DATE in the PATIENT_DIMENSION
AGE_GROUP_5_YEARS	Varchar(50), null	5 year age group range value
AGE_GROUP_10_YEARS	Varchar(50), null	10 year age group range value

Part of the implementation was to develop the DDL required to add this table to SQL Server and SQL script used to load the table.

Observation Period Stratification

The ESPnet ICD9 Diagnosis Query Builder queries can be stratified by monthly or yearly observation periods. This was handled using inline functions to format the monthly or yearly period. The following shows the SQL generated for monthly period stratification:

```

“LEFT(CONVERT(VARCHAR, f.START_DATE, 102), 4) + '-' + LEFT(CONVERT(VARCHAR,
f.START_DATE, 101), 2) AS "Observation Period"”

```

Varying Database Manager Deployments

PMN queries will need to execute at sites that use a variety of database managers (eg, ODBC, etc.). As such, executing complex SQL queries that use functions native to a specific database manager is problematic. ODBC may be used to resolve this problem; however, this approach restricts the use of SQL extensions native to a specific database manager often required to process the query efficiently. There are several approaches to resolving this:

- ✓ Develop custom SQL transforms for each database manager that uses the syntax for the given database manager
- ✓ Use ODBC and, if necessary, post process results to achieve the desired result

The first approach was used in the POC implementation. The use of SQL functions was limited to Date functions and Conversion functions, so implementing corresponding function in other SQL extensions should be straightforward revisions to the existing XML transformation.

Given the implementation approach, the current ESP model adapter has been revised to include settings that allow the user to configure the adapter to the data source by choosing two additional settings as follows:

- ✓ Data Source – a drop-down control used to identify the database manager to be used by the DataMart
- ✓ Translator – a drop-down control used to identify the XML transform used to generate the SQL for a specific schema

Conclusion

To enhance current capabilities for distributed querying, the EHR Core conducted a pilot project to evaluate how to enable distributed querying of organizations using i2b2 as their internal data resource. Based on the design requirements, the POC used an existing PopMedNet query interface (ESPnet ICD9 Diagnosis Query Builder based on the ESPnet data model) to create a query and distribute it to an ESPnet site and an i2b2 site for local execution and response. The query ran successfully, and individual site results and aggregated results were generated.

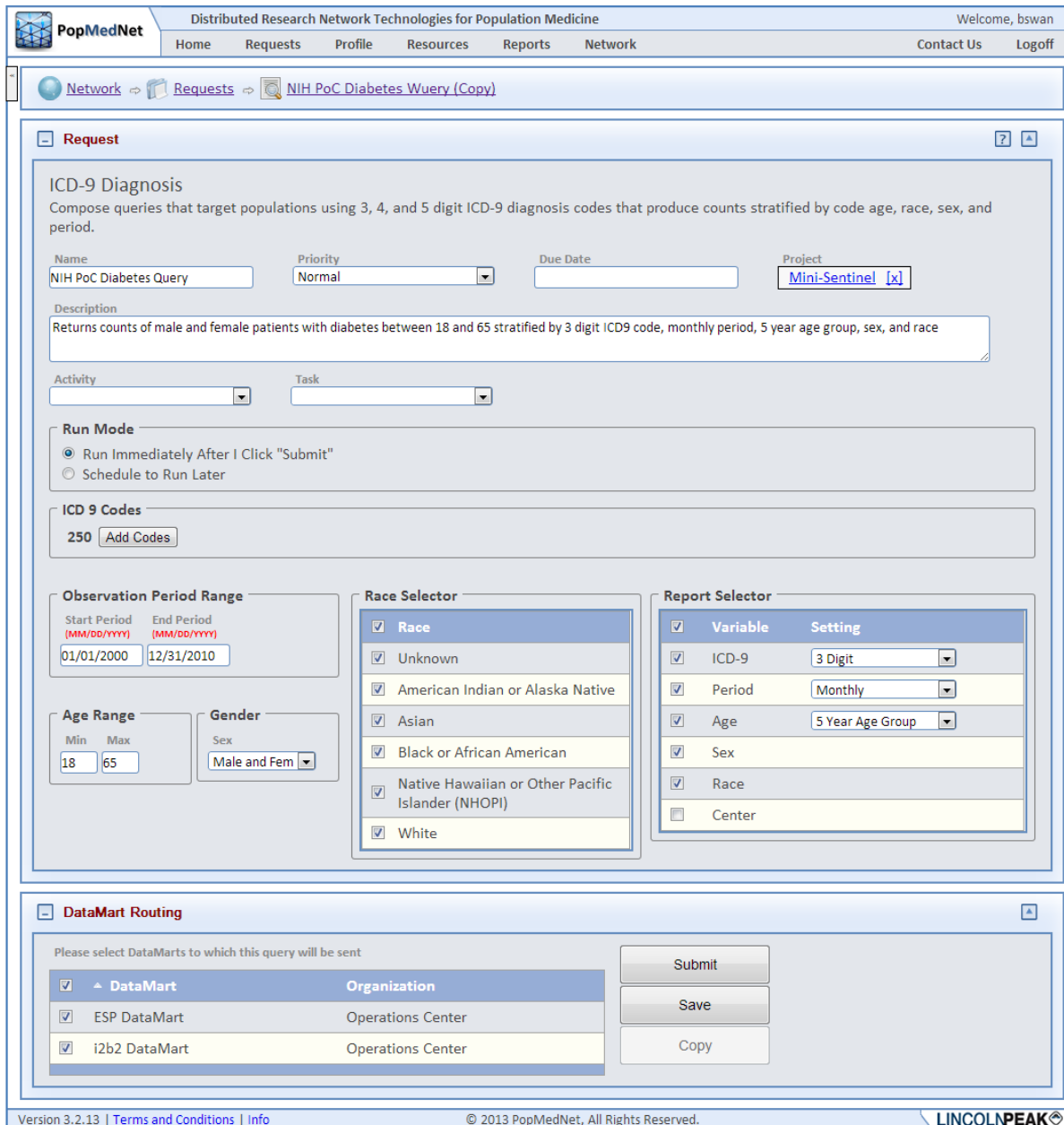
Acknowledgements

We acknowledge the invaluable insights and support provided by Drs. Jeff Klann and Shawn Murphy at Partners Healthcare. Their active participation, collaboration, and guidance substantially informed the design and implementation of the POC.

APPENDIX

ESPnet ICD9 Query Builder Composition Page

The following query is a sample query that was composed with the ESPnet ICD9 Diagnosis Query Builder and routed to an i2b2 DataMart supported by SQL Server and an ESPnet DataMart supported by PostgreSQL. The query returned counts of male and female patients with diabetes between 18 and 65 years of age stratified by 3 digit ICD9 code, month, 5 year age group, sex, and race.



PopMedNet Distributed Research Network Technologies for Population Medicine Welcome, bswan

Home Requests Profile Resources Reports Network Contact Us Logoff

Network Requests NIH PoC Diabetes Query (Copy)

Request

ICD-9 Diagnosis
Compose queries that target populations using 3, 4, and 5 digit ICD-9 diagnosis codes that produce counts stratified by code age, race, sex, and period.

Name: NIH PoC Diabetes Query Priority: Normal Due Date: Project: Mini-Sentinel [x]

Description: Returns counts of male and female patients with diabetes between 18 and 65 stratified by 3 digit ICD9 code, monthly period, 5 year age group, sex, and race

Activity: Task:

Run Mode
 Run Immediately After I Click "Submit"
 Schedule to Run Later

ICD 9 Codes
250 [Add Codes](#)

Observation Period Range
 Start Period (MM/DD/YYYY): 01/01/2000 End Period (MM/DD/YYYY): 12/31/2010

Age Range
 Min: 18 Max: 65

Gender
 Sex: Male and Fem

Race Selector

- Race
- Unknown
- American Indian or Alaska Native
- Asian
- Black or African American
- Native Hawaiian or Other Pacific Islander (NHOPI)
- White

Report Selector

Variable	Setting
<input checked="" type="checkbox"/> ICD-9	3 Digit
<input checked="" type="checkbox"/> Period	Monthly
<input checked="" type="checkbox"/> Age	5 Year Age Group
<input checked="" type="checkbox"/> Sex	
<input checked="" type="checkbox"/> Race	
<input type="checkbox"/> Center	

DataMart Routing

Please select DataMarts to which this query will be sent

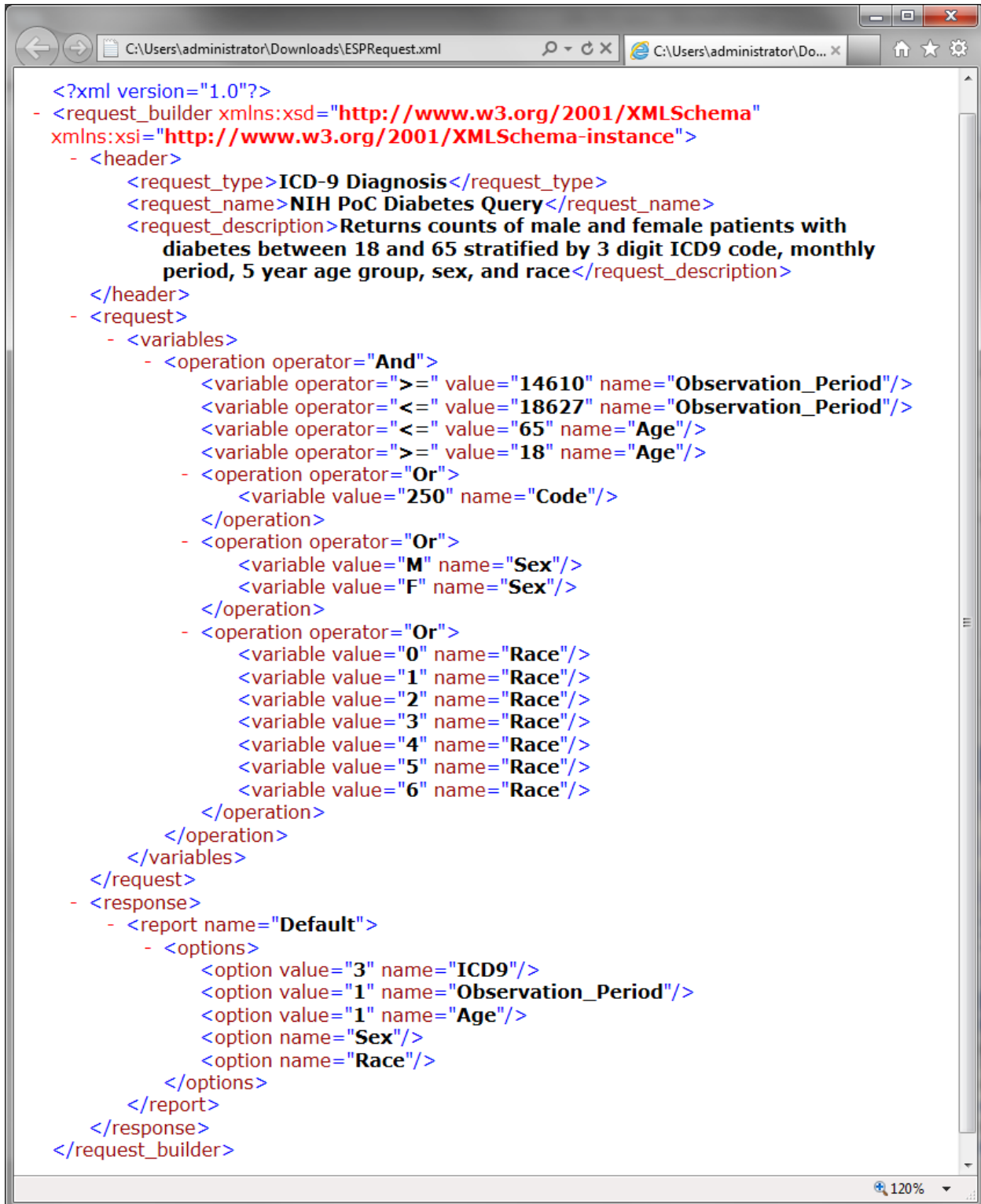
DataMart	Organization
<input checked="" type="checkbox"/> ESP DataMart	Operations Center
<input checked="" type="checkbox"/> i2b2 DataMart	Operations Center

Submit Save Copy

Version 3.2.13 | [Terms and Conditions](#) | [Info](#) © 2013 PopMedNet, All Rights Reserved. LINCOLN PEAK

Query Request XML

The following is the request serialized into XML that was routed to the DataMarts.



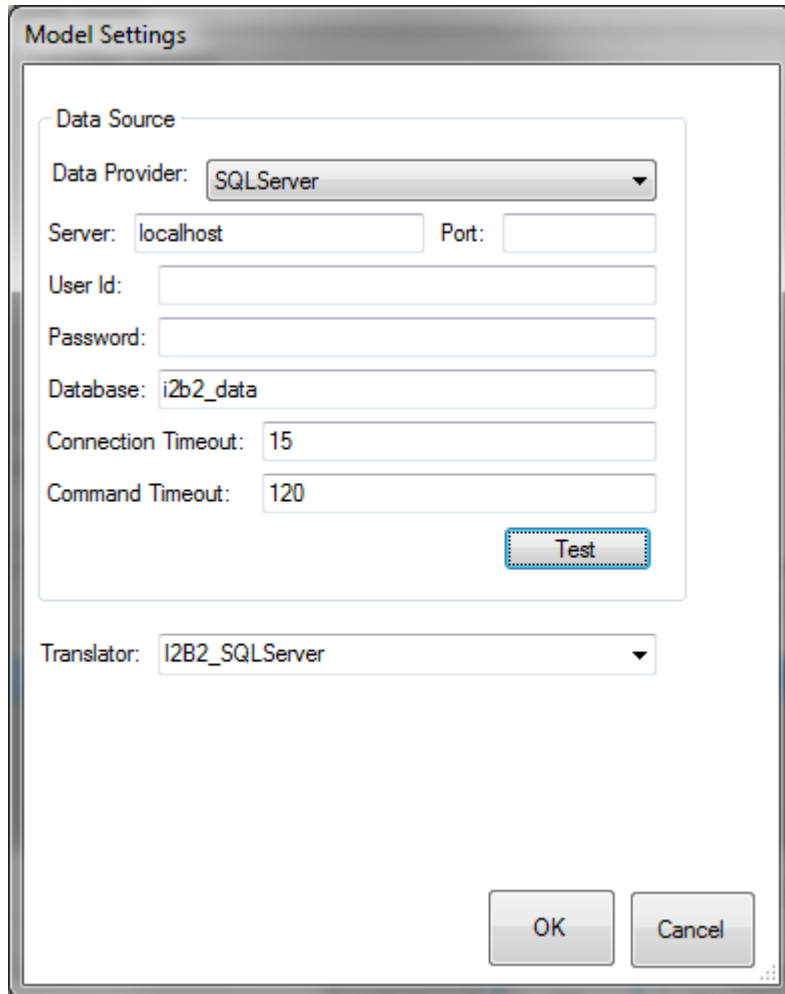
```

<?xml version="1.0"?>
- <request_builder xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <header>
    <request_type>ICD-9 Diagnosis</request_type>
    <request_name>NIH PoC Diabetes Query</request_name>
    <request_description>Returns counts of male and female patients with
      diabetes between 18 and 65 stratified by 3 digit ICD9 code, monthly
      period, 5 year age group, sex, and race</request_description>
  </header>
  - <request>
    - <variables>
      - <operation operator="And">
        <variable operator=">=" value="14610" name="Observation_Period"/>
        <variable operator="<=" value="18627" name="Observation_Period"/>
        <variable operator="<=" value="65" name="Age"/>
        <variable operator=">=" value="18" name="Age"/>
      - <operation operator="Or">
        <variable value="250" name="Code"/>
      </operation>
      - <operation operator="Or">
        <variable value="M" name="Sex"/>
        <variable value="F" name="Sex"/>
      </operation>
      - <operation operator="Or">
        <variable value="0" name="Race"/>
        <variable value="1" name="Race"/>
        <variable value="2" name="Race"/>
        <variable value="3" name="Race"/>
        <variable value="4" name="Race"/>
        <variable value="5" name="Race"/>
        <variable value="6" name="Race"/>
      </operation>
    </operation>
  </variables>
</request>
  - <response>
    - <report name="Default">
      - <options>
        <option value="3" name="ICD9"/>
        <option value="1" name="Observation_Period"/>
        <option value="1" name="Age"/>
        <option name="Sex"/>
        <option name="Race"/>
      </options>
    </report>
  </response>
</request_builder>

```

i2b2 and ESPnet Model Adapters

The following images show the DataMart Client Application model adapters for the i2b2 DataMart and the ESP DataMart.



The screenshot shows a "Model Settings" dialog box with the following fields and controls:

- Data Source** (grouped box):
 - Data Provider:** SQLServer (dropdown menu)
 - Server:** localhost (text input)
 - Port:** (empty text input)
 - User Id:** (empty text input)
 - Password:** (empty text input)
 - Database:** i2b2_data (text input)
 - Connection Timeout:** 15 (text input)
 - Command Timeout:** 120 (text input)
 - Test:** (button)
- Translator:** I2B2_SQLServer (dropdown menu)
- OK** and **Cancel** buttons at the bottom.

Model Settings

Data Source

Data Provider: PostgreSQL

Server: localhost Port: 5432

User Id: esp_mdphnet

Password: *****

Database: ESP_7_3

Connection Timeout: 15

Command Timeout: 120

Test

Translator: ESP_PostgreSQL

OK Cancel

i2b2 ICD9 Diagnosis SQL

The following SQL was generated by the i2b2 model adapter transform.

```
SELECT
  "Code", "Description",
  "Observation Period",
  "5 Year Age Group",
  "Sex",
  "Race"
  count("Patients") as "Patients"
FROM (
  SELECT DISTINCT
    l.CODE_3DIG AS "Code",
    l.NAME_3DIG AS "Description",
    LEFT(CONVERT(VARCHAR, f.START_DATE, 102),4) + '-' +
    LEFT(CONVERT(VARCHAR, f.START_DATE, 101),2) AS "Observation Period",
  CASE
```



```

WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 0 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 5
THEN '00-04'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 5 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 10
THEN '05-09'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 10 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 15
THEN '10-14'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 15 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 20
THEN '15-19'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 20 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 25
THEN '20-24'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 25 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 30
THEN '25-29'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 30 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 35
THEN '30-34'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 35 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 40
THEN '35-39'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 40 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 45
THEN '40-44'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 45 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 50
THEN '45-49'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 50 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 55
THEN '50-54'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 55 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 60
THEN '55-59'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 60 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 65
THEN '60-64'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 65 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 70
THEN '65-69'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 70 AND
     DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 < 75
THEN '70-74'
WHEN DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE) / 8766 >= 75
THEN '75+'
END as "5 Year Age Group",
p.SEX_CD AS "Sex",
rp.RACE_CHAR AS "Race",
f.PATIENT_NUM AS "Patients"
FROM OBSERVATION_FACT f
JOIN PATIENT_DIMENSION p on f.PATIENT_NUM = p.PATIENT_NUM
JOIN VISIT_DIMENSION v on f.ENCOUNTER_NUM = v.ENCOUNTER_NUM
JOIN ICD9_CODE_LOOKUP l on f.CONCEPT_CD = l.CONCEPT_CD

```

```

JOIN PMN_I2B2_RACE_CODE_LOOKUP pr on pr.RACE_CD = p.RACE_CD
JOIN I2B2_PMN_RACE_CODE_LOOKUP rp on p.RACE_CD = rp.RACE_CD
WHERE (CONVERT(date, f.START_DATE) >= DATEADD(DAY, 14610, CONVERT(date,
'1/1/1960')) And
CONVERT(date, f.START_DATE) <= DATEADD(DAY, 18627, CONVERT(date,
'1/1/1960')) And
DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 <= '65' And
DATEDIFF(hour, p.BIRTH_DATE, f.START_DATE)/8766 >= '18' And
(f.concept_cd like upper('ICD9:250%')) And
(p.SEX_CD = 'M' Or p.SEX_CD = 'F') And
(pr.RACE_CODE = 0 Or pr.RACE_CODE = 1 Or pr.RACE_CODE = 2 Or
pr.RACE_CODE = 3 Or pr.RACE_CODE = 4 Or pr.RACE_CODE = 5)
)
) i
GROUP BY "Code", "Description", "Observation Period", "5 Year Age
Group", "Sex", "Race"
ORDER BY "Code", "Description", "Observation Period", "5 Year Age
Group", "Sex", "Race"

```

i2b2 DataMart Client Query Detail Form

The following is the PopMedNet DataMart Client (DMC) query detail form after it has executed the query against the i2b2 DataMart. This is the mechanism that data partners use to review query results and securely upload the results to the NIH Collaboratory DRN secure portal.

DataMart Client - Request Detail

Network: Local Project: Mini-Sentinel DataMart: i2b2 DataMart
 Request Name: NIH PoC Diabetes Query Request Type: NIH PoC Diabetes Query Request Id: 32
 Priority: Medium Due Date: Status: AwaitingResponseApproval
 Requestor: bswan Request Time: 09/04/2013 09:37 AM Email: bswan@lincolnpeak.com
 Submitted To: ESP DataMart, i2b2 DataMart
 Description: Returns counts of male and female patients with diabetes between 18 and 65 stratified by 3 digit ICD9 code, monthly period, 5 year age group, sex, and race
 Note:

Request: File View


StartPeriod	2000-01-01T00:00:00
EndPeriod	2010-12-31T00:00:00
Codes	250
MinAge	18
MaxAge	65
MinVisits	0
Genders	Male and Female

Response: File View

Code	Description	Observation Period	5 Year Age Group	Sex	Race	Patients
250	Diabetes mellitus	2003-08	65-69	F	White	1
250	Diabetes mellitus	2004-04	45-49	M	Asian	1
250	Diabetes mellitus	2004-06	55-59	F	Black or African American	1
250	Diabetes mellitus	2004-06	55-59	M	Hispanic	1
250	Diabetes mellitus	2004-07	55-59	F	Black or African American	1
250	Diabetes mellitus	2006-04	30-34	M	Black or African American	1
250	Diabetes mellitus	2006-11	30-34	M	Black or African American	1

Completed Diagnosis Query

The following image shows the completed query in the NIH Collaboratory DRN secure portal request status page. This is how the requestor views the query results.

 PopMedNet
Distributed Research Network Technologies for Population Medicine
Welcome, bswan

Home Requests Profile Resources Reports Network
Contact Us Logoff

Network > Requests > NIH PoC Diabetes Query

Request
▲

ICD-9 Diagnosis
Compose queries that target populations using 3, 4, and 5 digit ICD-9 diagnosis codes that produce counts stratified by code age, race, sex, and period.

Name

Priority

Due Date

Purpose of use

Level of PHI Disclosure

Project

Project Description

Description
Returns counts of male and female patients with diabetes between 18 and 65 stratified by 3 digit ICD9 code, monthly period, 5 year age group, sex, and race

Activity

Task

Run Mode

Request was submitted immediately.

[Show Documents List](#)

ICD 9 Codes

Code	Description
250	DIABETES MELLITUS

Observation Period Range

Start Period	End Period
01/01/2000	12/31/2010

Races Selected

- Unknown
- American Indian or Alaska Native
- Asian
- Black or African American
- Native Hawaiian or Other Pacific Islander (NHOPI)
- White

Reports Selected

- ICD-9 (3 Digit)
- Period (Monthly)
- Age (5 Year Age Group)
- Sex
- Race

Age Range

Min	Max
18	65


Gender

Sex

Male and Female


Received Responses
▲

<input checked="" type="checkbox"/>	DataMart	Last Response	Status	Message
<input checked="" type="checkbox"/>	i2b2 DataMart	9/4/2013 9:58:08 AM	Completed	[history]
<input checked="" type="checkbox"/>	ESP DataMart	9/4/2013 9:45:12 AM	Completed	[history]

Version 3.2.13 | [Terms and Conditions](#) | [Info](#)
© 2013 PopMedNet, All Rights Reserved.


Completed Diagnosis Query Results

The following image shows the completed query in the NIH Collaboratory DRN secure portal request results page containing an individual site results view.


PopMedNet

Distributed Research Network Technologies for Population Medicine

Welcome, bswan
[Home](#) [Requests](#) [Profile](#) [Resources](#) [Reports](#) [Network](#) [Contact Us](#) [Logoff](#)

Network Requests NIH PoC Diabetes Query

ICD-9 Diagnosis
 Compose queries that target populations using 3, 4, and 5 digit ICD-9 diagnosis codes that produce counts stratified by code age, race, sex, and period.

Name

Priority

Due Date

Purpose of use

Level of PHI Disclosure

Project

Project Description

Description

Activity

Task

Source DataMarts:

[Show Documents List](#)

DataMart	Code	Description	Observation Period	5 Year Age Group	Sex	Race	Patients
i2b2 DataMart	250	Diabetes mellitus	2003-08	65-69	F	White	1
i2b2 DataMart	250	Diabetes mellitus	2004-04	45-49	M	Asian	1
i2b2 DataMart	250	Diabetes mellitus	2004-06	55-59	F	Black or African American	1
i2b2 DataMart	250	Diabetes mellitus	2004-06	55-59	M	Hispanic	1
i2b2 DataMart	250	Diabetes mellitus	2004-07	55-59	F	Black or African American	1
i2b2 DataMart	250	Diabetes mellitus	2006-04	30-34	M	Black or African American	1
i2b2 DataMart	250	Diabetes mellitus	2006-11	30-34	M	Black or African American	1
i2b2 DataMart	250	Diabetes mellitus	2007-07	20-24	M	Hispanic	1
i2b2 DataMart	250	Diabetes mellitus	2008-04	55-59	F	White	1
i2b2 DataMart	250	Diabetes mellitus	2009-04	65-69	M	Asian	1
DataMart	Code	Description	Observation_Period	5 Year Age Group	Sex	Race	Patients
ESP DataMart	250	Diabetes mellitus	2009-07	15-19	F	Unknown	3
ESP DataMart	250	Diabetes mellitus	2009-07	20-24	F	Unknown	1
ESP DataMart	250	Diabetes mellitus	2009-07	20-24	M	Unknown	2
ESP DataMart	250	Diabetes mellitus	2009-07	25-29	F	Asian	1
ESP DataMart	250	Diabetes mellitus	2009-07	25-29	F	Unknown	1
ESP DataMart	250	Diabetes mellitus	2009-07	25-29	M	Asian	1
ESP DataMart	250	Diabetes mellitus	2009-07	30-34	F	Unknown	3
ESP DataMart	250	Diabetes mellitus	2009-07	30-34	M	Unknown	3
ESP DataMart	250	Diabetes mellitus	2009-07	35-39	F	Unknown	1
ESP DataMart	250	Diabetes mellitus	2009-07	35-39	M	Unknown	1
ESP DataMart	250	Diabetes mellitus	2009-07	40-44	F	Unknown	4

Version 3.2.13 | [Terms and Conditions](#) | [Info](#)
© 2013 PopMedNet, All Rights Reserved.
