# Query timing - ANYtime and SAME visit queries

Query Timing: ANYtime in Patient History and In SAME Patient Visit

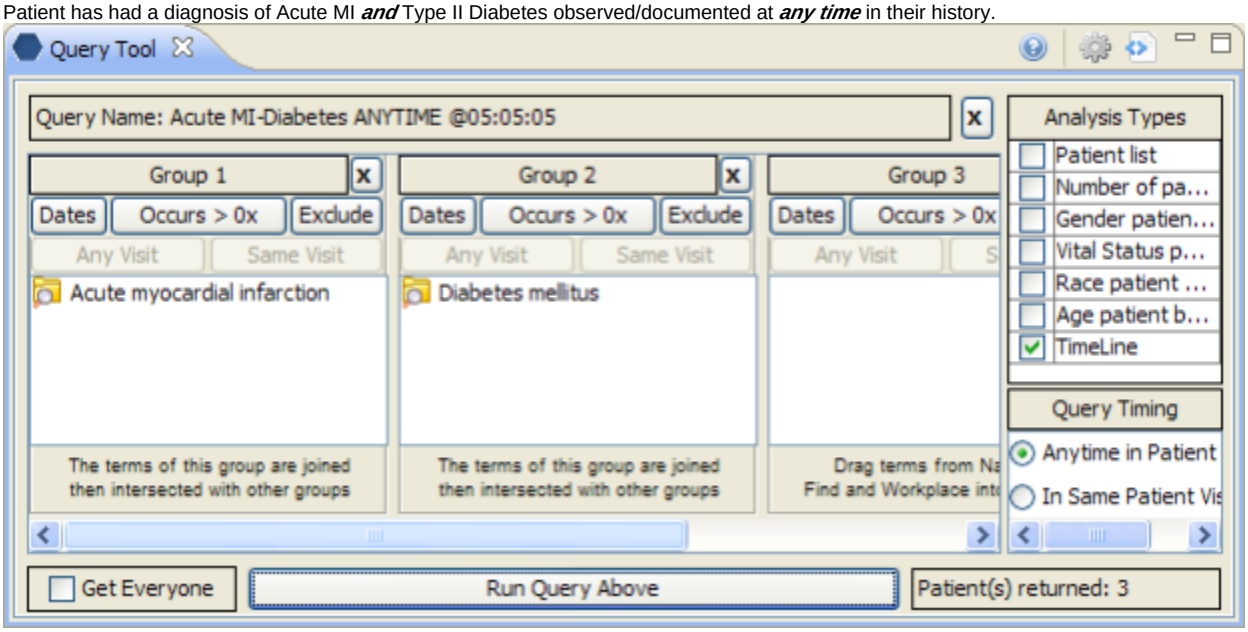
Introduced in Core i2b2 Version 1.6

In the 1.6 Release of i2b2 we introduce the idea of "Query Timing". Query Timing is a way of specifying co-occurring events. In the Query Timing used in the i2b2 User Interface up to this release, concepts were assumed to be co-occurring in the same patient. In 1.6, the idea gets extended to concepts that co-occur in one visit.

The way it works is that in the data repository is a visit dimension which is part of the star schema and is populated by rows of visits, each that are tied to patients and observation_facts in the fact table through the encounter_num column. When a query is requested to run and be constrained by the "In Same Patient Visit" option, the data will be tied together by the encounter_num and patients will only be found that have all of the specified concepts occurring together in one of their visits.

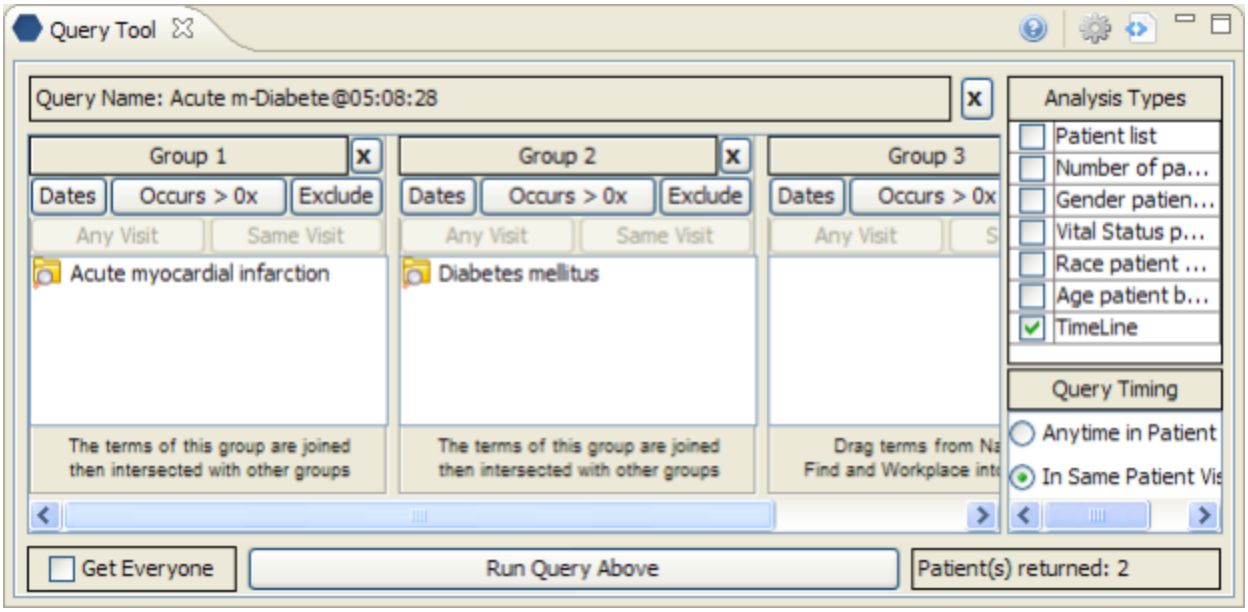
The visit_dimension table is intended to be a record of actual visits that the patient has at a hospital or outpatient care center. It may also be used as a proxy for a specimen (an encounter with a lab) or as a proxy for an assay (an encounter with a microarray). Although this works for many cases, a more elegant solution may be implemented in the future to support configurable types of "encounter_dimensions" each with their own observation_fact table column.

**Example 1:**
Patient has had a diagnosis of Acute MI ***and*** Type II Diabetes observed/documented at ***any time*** in their history.



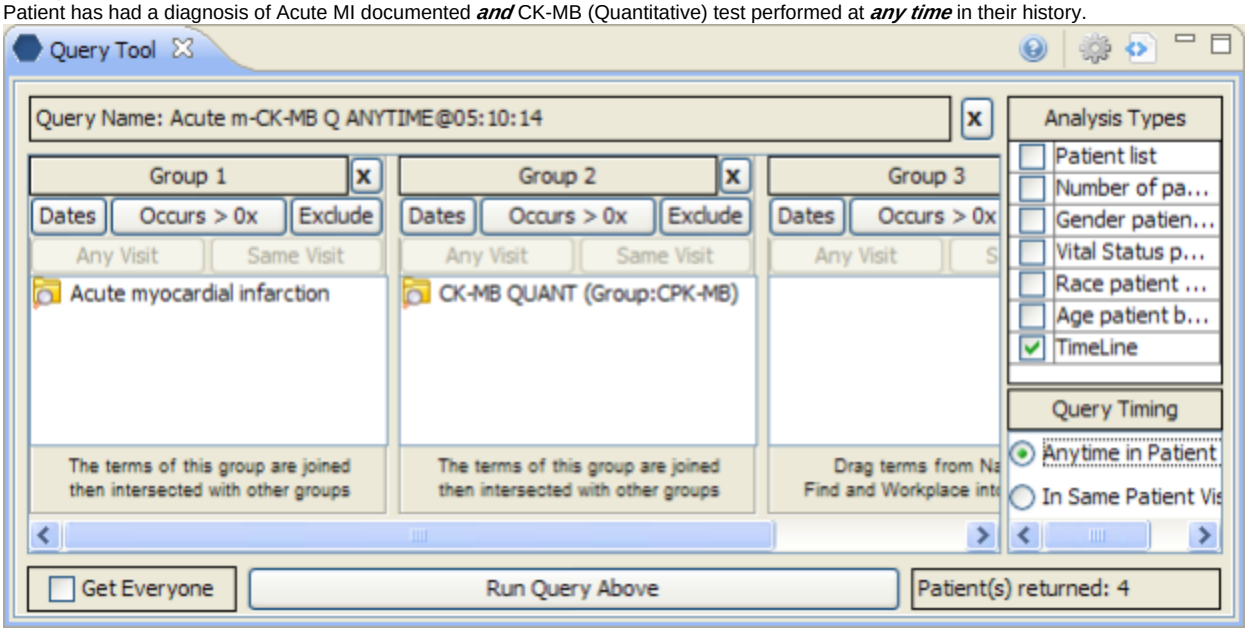

**Results**: there are 3 patients who have had a diagnosis of both an acute MI and type II diabetes entered as a fact in the observation_fact table, where acute MI and type II diabetes have the same patient_num.

**Example 2:**
Patient has had a diagnosis of Acute MI and Type II Diabetes observed/documented during the same visit.



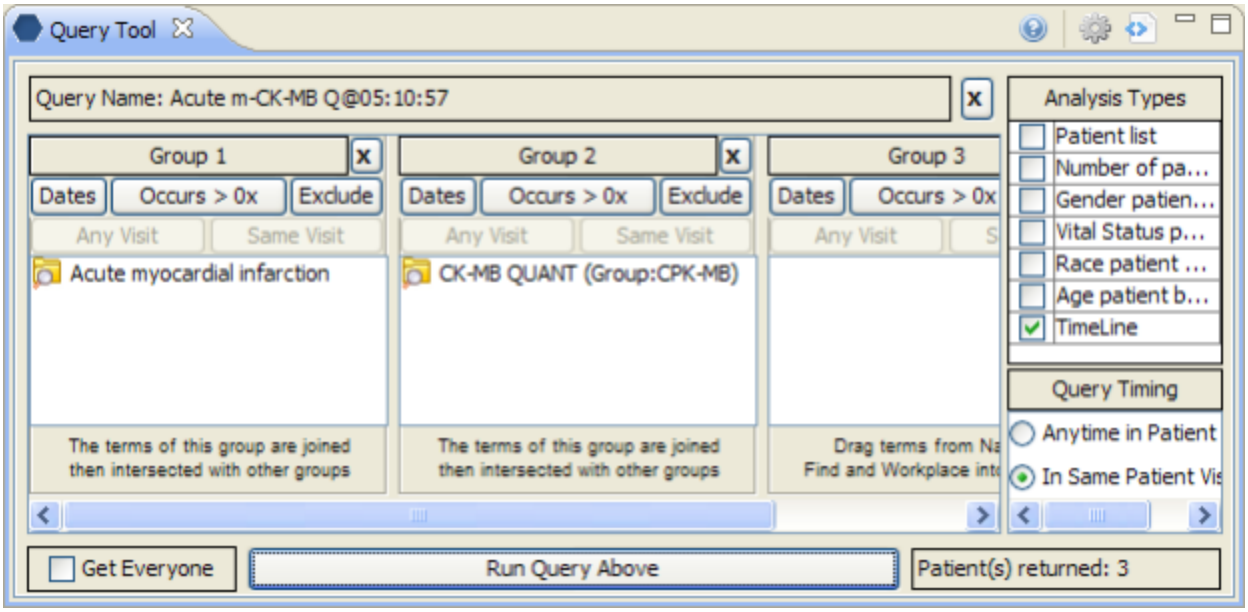

**Results**: there are 2 patients who have had a diagnosis of both an acute MI and type II diabetes entered as a fact in the observation_fact table where acute MI and type II diabetes have the same encounter_num.

**Example 3:**
Patient has had a diagnosis of Acute MI documented ***and*** CK-MB (Quantitative) test performed at ***any time*** in their history.




**Results**: there are 4 patients who have had a diagnosis of both an acute MI and a CK-MB test entered as a fact in the observation_fact table where acute MI and the CK-MB test have the same patient_num.

**Example 4:**
Patient has had a diagnosis of Acute MI documented ***and*** CK-MB (Quantitative) test performed during the same visit.




**Results**: there are 3 patients who have had a diagnosis of both an acute MI and a CK-MB test entered as a fact in the observation_fact table where acute MI and the CK-MB test have the same encounter_num..

The ability also exists to obtain the list of encounter_nums (or Visits) that are positive "hits"

The visit_dimension can use metadata tables in the ontology cell with similar options to the way the patient_dimension can use metadata tables, allowing complex formulas behind the query items. For example, an age of the patient at the visit can be represented as an ontology item, as the visit_dimension table has the start and end date of all visits.

For example, this query gives all patients that had a visit before they were three years old:

(SQL Server)

```
Declare @age_of_patient_that_visit_should_proceed integer
set @age_of_patient_that_visit_should_proceed = 3
select encounter_num from visit_dimension
where start_date <
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25*@age_of_patient_that_visit_should_proceed))
```

For example, this query gives those patients that had a visit when they were 40-45 years old:

(SQL Server)

```
Declare @age_of_patient_that_visit_should_start_on integer
Declare @age_of_patient_that_visit_should_end_on integer
set @age_of_patient_that_visit_should_start_on = 40
set @age_of_patient_that_visit_should_end_on = 45
select encounter_num from visit_dimension
where start_date BETWEEN
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * (@age_of_patient_that_visit_should_start_on)))
AND
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * (@age_of_patient_that_visit_should_end_on+1)))
```

Both of these queries can result from routine metadata definitions. The items in the ontology tables would look like:
The ontology cell provides instructions on how the data repository cell should perform its query for any given term or set of terms. Each row in an ontology metadata table(s) represents a term or item, such as a diagnosis like "diabetes", or a laboratory test like "white blood cell count". In order to find these items in the data repository, the data repository cell needs to know which table to find them in and how to identify them. The information contained in the metadata tables is utilized to build a metadata SELECT SQL statement as follows:

select [c_facttablecolumnname] from [c_tablename] where [c_columnname] [c_operator] [c_dimcode]
In the ontology table, the items that define encounters as shown in the SQL queries above would look like:
Encounters for all patients that had a visit before they were three years old:

(SQL Server)

```
select encounter_num from visit_dimension
where start_date <
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * 3))
```

(Ontology table entries)

```
[c_name] = patients that had a visit before they were three years old
[c_fullpath] =
encounter detail\ age at visit\ < 3 y/o
[c_facttablecolumnname] = encounter_num
[c_tablename] = visit_dimension
[c_columnname] = start_date
[c_operator] = <
[c_dimcode] =
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * 3))
```

Encounters for all patients that had a visit when they were 40-45 years old:

(SQL Server)

```
select encounter_num from visit_dimension
where start_date BETWEEN
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * 40))
AND
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * 45))
```

(Ontology table entries)

```
[c_name] = patients that had a visit when they were 40-45 years old
[c_fullpath] =
 encounter detail\ age at visit\ 40 – 45 y/o
[c_facttablecolumnname] = encounter_num
[c_tablename] = visit_dimension
[c_columnname] = start_date
[c_operator] = BETWEEN
[c_dimcode] =
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * 40))
AND
((select birth_date from patient_dimension
where patient_num = visit_dimension.patient_num)
+ (365.25 * 45))
```

© Shawn Murphy 2010