

Published in final edited form as:

*J Biomed Inform.* 2014 December ; 52: 72–77. doi:10.1016/j.jbi.2014.02.010.

## Using Patient Lists to Add Value to Integrated Data Repositories

Ted D. Wade<sup>a</sup>, Pearlanne T. Zelarney<sup>a</sup>, Richard C. Hum<sup>a</sup>, Sylvia McGee<sup>a</sup>, and Deborah H. Batson<sup>b</sup>

<sup>a</sup>Division of Biostatistics and Bioinformatics, National Jewish Health, Denver, Colorado 80206, USA

<sup>b</sup>Department of Research Informatics, Children's Hospital Colorado Research Institute, Aurora, Colorado 80045, USA

### Abstract

Patient lists are project-specific sets of patients that can be queried in integrated data repositories (IDR's). By allowing a set of patients to be an addition to the qualifying conditions of a query, returned results will refer to, and only to, that set of patients. We report a variety of use cases for such lists, including: restricting retrospective chart review to a defined set of patients; following a set of patients for practice management purposes; distributing "honest-brokered" (deidentified) data; adding phenotypes to biosamples; and enhancing the content of study or registry data.

Among the capabilities needed to implement patient lists in an IDR are: capture of patient identifiers from a query and feedback of these into the IDR; the existence of a permanent internal identifier in the IDR that is mappable to external identifiers; the ability to add queryable attributes to the IDR; the ability to merge data from multiple queries; and suitable control over user access and de-identification of results. We implemented patient lists in a custom IDR of our own design. We reviewed capabilities of other published IDRs for focusing on sets of patients. The widely used i2b2 IDR platform has various ways to address patient sets, and it could be modified to add the low-overhead version of patient lists that we describe.

### Keywords

integrated data repository; patient registry; honest broker; bio-repository; i2b2; meaningful use

## 1. Introduction

We are well into the era of "secondary use" of health data [1], of which an important part is reusing clinical data for both publishable research and quality improvement [2]. A 2010 survey [3] defined "integrated data repository" (IDR) as a data warehouse integrating

---

© 2014 Elsevier Inc. All rights reserved.

Corresponding author: Ted D. Wade, Division of Biostatistics and Bioinformatics, National Jewish Health - Room M222, 1400 Jackson Street, Denver, CO 80206-2761, wadet@njhealth.org, +1-303-398-1877.

**Publisher's Disclaimer:** This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

various sources of clinical data to support queries for a range of research-like functions. IDRs for research are usually designed to allow “attribute-centric” queries [4]: that is, queries for the set of patients who meet some criteria based on values of clinical observations or characteristics (also called “attributes” in the common Entity-Attribute-Value (EAV) model [5, 6]). An example might be, “Give me all the patients who have more than one admission and are on inhaled corticosteroids.” Dinu and Nadkarni [7] said that clinical care systems are, in contrast, optimized to retrieve many or all attribute values for a particular patient. An example would be displaying the entire electronic chart for a selected patient. They also noted an occasional need to do both at once - to retrieve a specified set of attributes for a specified set of patients. They called this “bulk data extraction,” a name suggestive of exceptions and custom programming. We will call this a patient list query. An example might be “For my pre-defined set of patients, filter them by those who have had a chest CT and return their current meds and pulmonary function test results.” IDRs designed to query for specified sets of patients in this way seem to be rare or limited in the capability. In our own IDR [6] we have discovered a variety of uses for such a function. Primarily these involve query for detailed, patient-level data rather than aggregations. We shall illustrate the value of patient lists with a number of use cases, describe methods for addressing related technical and governance issues, and then discuss the concept with regard to i2b2.

### 1.1 IDR Query Capabilities

IDRs are complex systems that have to solve a variety of problems, including: identity management, semantic and syntactic comparability of data from different sources, protection of confidentiality, convenient and flexible query, and the performance and usability issues arising from the sheer volumes and varieties of medical data [6,8]. The ability of such systems to support patient-list query varies.

One approach to an IDR is to simplify governance by irreversibly de-identifying all data, as in Vanderbilt’s BioVU DNA Biobank [9]. This would necessarily preclude query about a known (i.e., identified) set of patients.

The Enterprise Data Trust at Mayo Clinic [10] emphasizes industrial-scale data modeling of healthcare concepts to a relationally normalized database. From this they extract a dimensional model [11] to support ad hoc query by using standard commercial tools. Dimensional models are used for queries that aggregate data across selected dimensions, such as time or location, that cut across all records. Other attribute values pertaining to individual patients, such as specific drugs, conditions or procedures, would not be available for selecting data in a dimensional query.

The National Institutes of Health has an intramural IDR called BTRIS [12] to hold data from numerous research studies. BTRIS supports query by providing an interface with templates in which users can select values of clinical and demographic attributes for filtering returned data. Because protocols are a high-level concept in this system, an investigator can focus a query on the subjects in their own study. They also offer “list reports” to create lists of patients “that can be used as filters for other reports”. This seems like a form of query by patient list.

The recent IDR survey [3] noted, by comparison to a previous survey, trends for substantial and increasing use of flexible, ad hoc user query interfaces (as opposed to custom programming for data extraction) and automatically de-identified data (during either data loading or output). One such system is Stanford's STRIDE IDR [13]. It supports a clinical data warehouse, individual research project data management, and a biorepository. The underlying data model is EAV, accessed by two graphical user query interfaces with granular access control and with the ability to release data that has varying levels of de-identification. All are features that it shares with the widely used i2b2 platform[14]. STRIDE allows users to discover and save patient cohorts, which sounds like the internally-defined patient lists that we describe later. i2b2's suitability for patient list query is discussed in detail later.

This paper will illustrate patient list uses and concepts at National Jewish Health in a custom-designed IDR [6], known to us simply as the RDB (Research Database). The RDB uses a unique data model called a "dimensional bus", implemented on a SQL Server 2005 database platform. It has a custom query user interface, programmed using the ColdFusion language, that compiles T-SQL queries at run time. This interface gives highly expressive, ad hoc and simultaneous query of any attribute of clinical, research and biorepository data. Data in the RDB come from an Allscripts EHR (electronic health record), a Freezerworks sample management system, and our custom-programmed clinical data management system, Study Design by Metadata (SDM).

All researchers are allowed to query the RDB for record frequencies, obscured to protect privacy. Query for individual-level data is controlled by issuing to individual research projects customized "Access Tickets." The Tickets implement the institutional review board's (IRB) permissions specifying access to data sources, level of de-identification, and even patients's consents for studies of a certain type. The RDB can act as an automated honest broker [15], by maintaining a link between a random internal identifier, called the ASID (Anonymous Subject Identifier) and direct identifiers such as patient name. All queries expose the ASID automatically, but a study being allowed access to direct identifiers will obtain those through a separate action by our database curator.

## 2. Background

While the patient list capabilities of the RDB evolved incrementally, with use cases and requirements in a feedback loop, we shall present all our use cases (Table 1) first as "motivation" for the requirements.

### 2.1 List from IDR Query

(Table 1 case 1). Most of our lists are generated from internal queries of our IDR. The initial query is usually one authorized by the IRB for cohort discovery. In some cases the curator adds personal identifiers to that query to support subject recruitment. In other cases the cohort is being used in retrospective research, and so its composition is frozen because the IRB has restricted the protocol to data existing at the time of protocol approval. In either the prospective or retrospective model, a patient list allows the researcher to learn more about the population: prospectively, as new EHR data are acquired, or retrospectively, when

different types of observations are needed. Our IRB treats this as a safer form of “chart review”. Compared to unrestricted review of the medical record, the IRB prefers queries by patient list in our IDR because: it restricts the scope of review, it can de-identify the data, and it leaves a detailed audit trail of accesses.

## 2.2 Add to Existing Study Database

(Table 1 case 2). The patient list can be used to help with a very common problem: acquiring data on regular clinical care for an existing study sample [16]. We do this by constructing a list based on the study’s identified patients. The study then uses the list to query the repository’s EHR data for those patients, and links the query results back into the study database, using one of the mechanisms described in section 3.5.

## 2.3 EHR-derived List

(Table 1 case 3). Users needing data to support population health management might supply us a list of clinical patients that also included, for example, data on the stage of a patient’s treatment or disease, or the projected date of some follow-up visit. These variables can be used in IDR queries returning information on selected subsets of the list. An important source of patient lists could be the lists that are part of the CMS (Centers for Medicare and Medicaid Services) EHR *Meaningful Use* objectives [17]. The objective asks eligible professionals to “Generate lists of patients by specific conditions to use for quality improvement, reduction of disparities, research, or outreach” [18]. Encouraging the export of such lists from EHRs to a linked IDR would open new possibilities for making their use more “meaningful”. We currently have one EHR-generated patient list, part of an Alpha-1 antitrypsin deficiency registry, that we update with each data load. The clinician uses the list to monitor if the follow-up visits have occurred and to check on population level lab results.

## 2.4 Honest Brokering

(Table 1 case 4). Our IDR [6], like others [14] can act as an automated “honest broker” [15] because it can de-identify data and maintain the coded key (the ASID) to it. Our IDR can, by using a patient list, also be helpful when our Honest Broker Service de-identifies data from sources outside the IDR. Brokers from our Service can create a patient list containing data that they have de-identified. By giving the researcher an Access Ticket that allows query of that list, we “deliver” the brokered data, but also can allow them to make queries for additional, de-identified, IDR data on their patients.

## 2.5 Biobanking

(Table 1 case 5). Researchers can use our IDR to search for institutional biobank specimens whose donors have specific clinical attributes. We have created patient lists derived from such searches so that investigators can query to find out more about the donors for stratified randomization or other research purposes. For example, one study identified biosamples based on whether the patient had certain allergy testing done, but needed to be blinded to test status until all their samples were assayed as part of the study protocol. Once they had their results, we created a patient list so they could query for clinical information such as diagnosis, skin test results, and IgE levels for each sample/subject.

Honest brokering is used for another biobank, our Live Cell Core. The Core is chartered to operate as non-human subjects research, because honest brokers de-identify the cell specimens right after collection, and donors are not consented for particular studies. This severely limits the phenotypic information available. However, a researcher can apply to the Honest Broker Service to obtain more detailed phenotypes. The Service can implement this by making a patient list of the donors and then making IDR queries from the list.

## 2.6 Access to Study/Registry Data

(Table 1 case 6). There are a few large or long-term studies that have found it useful to load their entire study databases into our IDR. If the study data are periodically updated in the IDR, this arrangement has some similarity to a “registry” [19] (see also the discussion of i2b2 later). Having a study database in the IDR has two advantages. First, the study can use the IDR’s powerful query capabilities to explore their own data. Second, an Access Ticket can be issued to allow a new collaborator with the study to examine study data or to link it to other data in the IDR that was not part of the original study.

In data from a complex study or registry there might not be any variable that can be used in a condition to select all subjects in the study. To meet this need we add a patient list of all the subjects in a study. Other attributes in a list, which can hold patient characteristics or study status (Sections 3.1 and 3.5 explain how list attributes are added), can be used synergistically with study data to provide easily defined subsets of subjects for registry, study or collaboration purposes. We use three overlapping patient lists to track processes in our automated cancer registry (which is partially imbedded in our IDR) for state reporting.

## 2.7 Promoting Collaboration

(Table 1 case 7). Our system allows query for privacy-safe frequencies (patient counts) to any local researcher, without needing protocol approval. We decided to allow our patient lists to appear in the catalog of data items allowed in frequency queries. We reasoned that it might encourage research collaboration for researchers to be aware of the existence and sample size (revealable by a frequency query) of each other’s lists. One researcher with a large collection of bio-samples created a patient list so that he could direct colleagues to the IDR when they asked about available samples and about overlap with their own study criteria.

## 3.0 Material and Methods

We developed the ability to query data from our IDR where the query is restricted to a particular subset, called a *patient list*, of the patients in the IDR. In Table 2 we define the level of need for various functional capabilities when using patient lists from two types of sources. In the first instance the patient list gets defined *internally* by a query of the IDR. In the second instance the list is supplied *externally* from a source outside the IDR. These requirements guided our implementation. We shall note general implementation issues, but detailed guidance for other IDRs could only be based on deeper knowledge of those systems than we have. We note below when a requirement has specific relevance to particular use

cases. All but one of the use cases (Table 1, case 7) assume that IDR query can return individual-level data, not just frequencies.

### 3.1 Feedback from Internal Query

(Table 2 row 1). After their initial queries of our IDR we found that users often wanted to come back for more data on the same sample of patients. Our users could save the logic of their discovery query and modify it to return more types of observations. However, as our IDR always gained more data and more patients every month, the same query qualifying conditions executed later would usually return more patients than were originally discovered, thus creating a moving target for the researcher.

In the case of retrospective research, IRB approval might have been given to study the discovered, point-in-time, sample only. In the case of prospective research the researcher might want to follow only the initially discovered sample over time, extracting new observations, but not new patients, from the IDR. In either case the use of stored queries based on attribute values is problematic.

To solve this problem we added a new workflow. After our users save an attribute-based query, they can request that the set of patients that it found be used to create a patient list. The list consists of a set of tuples of the form, {ASID, ListX.member}, where the attribute, ListX.member, has a value meaning “is a member of list X”. With the list added as queryable data to the IDR, queries can include ListX.member in a query condition to restrict any returned data to refer to only list members. An example is given in Figure 1. This requirement enables use cases 1, 5 and 6.

### 3.2 Permanent Identifier

(Table 2 row 2). IDRs logically need to have an internal, permanent identifier to designate distinct individual patients and to merge information on individuals from various sources. Some of them define governance circumstances under which the identifier, or some relatively enduring surrogate identifier that is mapped 1:1 to the permanent one, can be revealed to users, thus allowing longitudinal followup information to be acquired later. We believe this to be a requirement for any realistic application of patient lists. The IDR at National Jewish Health can reveal such an identifier [6], a randomly chosen number that we call the ASID (Anonymous Subject Identifier). The i2b2 system can expose such an identifier [14], but for governance reasons some other IDRs [9, 20] do not.

### 3.3 Adding Attributes to IDR

(Table 2 row 3). We implemented our patient lists by creating new attributes (see explanations related to “tuples” in Sections 3.1 and 3.5) to query in our IDR. This is easiest in repositories, such as ours [6] or i2b2 [14], that add attributes using a modifiable data dictionary.

### 3.4 Convert External ID

(Table 2 row 4). If a list of patients comes from a source external to the IDR, they will have to be mapped to the internal permanent identifier, thus allowing them to be used as data in

the IDR. By policy our internal identifier, even though it is a part of the results of any query, is never to be used for re-identification of a patient. Therefore, an external source can only designate a list patient by supplying direct identifiers, such as name or medical record number. Mapping of a direct identifier to the internal ID can only be done if the IDR maintains such a link. Some IDRs, like ours, maintain a link from real identities to internal identities [6, 14] and some either do not [20] or may forbid its use by policy [9]. This requirement enables use cases 2, 3, 4 and 5.

We allow users to request creation of a patient list using any direct identifier that is tracked in our master patient index, including full name, medical record number, and study identifier (for patients who also have study data in our IDR).

### 3.5 Merge Multiple Queries

(Table 2 row 5). Use of a patient list always results in adding to existing information that is held by the researcher on those patients. If the list is created by an internal query, subsequent queries on the list are meant to add to the data gained by the initial query. Users can merge these data using the exposed internal identifier (in our case, the ASID).

If the list is based on an external source, the subsequent list query results need to be merged to the data in the external source. One solution is to add to the list tuples a surrogate identifier such as a study id. So a tuple would look something like {ASID, ListY.member, studyID}. Queries of the list can therefore return the studyID for use in merging. This means that access to the studyID attribute must only be allowed for the researchers originating the list, who by definition already know the patients' real identities. Another solution is for the IDR curator to give the list user a table that crosswalks the ASID (the ASID is automatically included in query results) and whatever direct identifier was originally used to create the list. In either solution, note that the list user eventually sees an association of the ASID with the direct identifier. This is sometimes allowed because researchers agree in their protocol that protected health information, which includes knowledge of this association, can only be used within their project. In other cases the IDR curator does the queries for the researcher, delivering query results that expose only the direct identifier, already known to the researcher, instead of the ASID. These linking issues are most relevant to use case 2.

### 3.6 Modifying List Membership

(Table 2 row 6). Some projects have requested that the membership of a list be modified after its initial creation. As long as this does not conflict with their IRB-approved protocol, we will do this for them. We find that externally-sourced lists are more likely to request modifications of their lists, so this requirement relates most to use cases 2, 3 and 6

### 3.7 Policies on De-identification

(Table 2 row 7). A repository must be able to implement any level of de-identification [21] of data that is required by local governance (e.g., IRB or Privacy Office) to protect results of queries that use a particular patient list. In doing automated de-identification of queried data, an IDR is, in essence, acting as an honest broker [15]. In our IDR [6] we can issue a research project an Access Ticket that prescribes which data they can query (e.g., a particular patient



list and medical record data) and the level of deidentification (e.g., protected health information, HIPAA (Health Insurance Portability and Accountability Act) limited data set, or HIPAA de-identified). De-identification happens more often in use cases 4 and 5.

### 3.8 Restricted Use

(Table 2 row 8). Patient lists are intended to be used only by particular authorized projects when querying for patient-level data. Thus there needs to be a way to restrict a list's use to the researchers authorized to query for the project. We do this by assigning a project's Access Ticket to individual logins, and require that a person logging in should select only one Access Ticket from among any Tickets they have.

### 3.9 Avoid Inadvertent Exposure

(Table 2 row 9). Queries using a list should only return data about the patients on the list. This requires close attention to the workings of a query processor. In our case when an Access Ticket authorizes use of a patient list, software will force any query to include an attribute from the list in the query conditions. With such a Ticket, software also prohibits the use of a non-patient list attribute as a logical OR condition in the query. For example, a query for list membership OR patients with no mental health diagnosis would be prohibited. Both restrictions prevent exposure of data on non-list patients.

### 3.10 Data flow

Figure 1 shows examples of the sequence of flow of data in creating a list from an external source (steps A.1 and A.2) and from an internal query (steps B.1 and B.2). Steps A.3 and A.4 show a study using an external list to query the IDR for new information about the list patients. Note that the external list uses a reference to a Master Patient Index to translate medical record numbers into the internal patient ID code. No such translation is needed for the internally-sourced list, which already contains the internal code.

### 3.11 Applicability to a common IDR platform

Our lists are used in a custom IDR [6]. We analyzed the use of the popular i2b2 platform [14] at Children's Hospital Colorado (CHCO) to learn the extent to which similar functionality might exist, or be created, in that widely-used platform. We found that not all features of what we are calling a patient list exist in i2b2, but there is no barrier to implementing such a feature. i2b2 queries are built around a set of concepts, called its ontology, that are user-determined at a local installation. An i2b2 instance may reference an ontology combining standard terminologies like ICD-9-CM or LOINC with a locally developed set of terms. The local terminology might reference local laboratory test codes, or observations particular to a study. Therefore concepts defining data in multiple patient lists, which might contain both standard and local terms unique to a patient list, could be added readily.

An i2b2 query returns a de-identified set of patients and the clinical observations requested in the query. The logic of a query in i2b2 can be saved with a date range in its qualifying conditions, e.g., '*All patients with encounters in 2012 having a Dx of bronchiolitis and a PICU visit*'. Repeating the query later when the database has been updated should find the



same set of patients as before, except where the data loading process has retroactively dropped or added qualifying patients due to identity adjustments. Once a query has been “frozen” by date, one can re-run the query, adding other conditions that are not subject to the date range and thus get the same set of patients with additional observations from the underlying database. Our custom query engine at NJH [6] can be used in the same way.

There is no built-in method in i2b2 to define a patient list, defined as an independent set of queryable identities, without using a frozen query as described above. However, there is no obvious reason why such a method could not be developed, since i2b2 is fully extensible in both query tool and data model.

Working solely with the “out-of-the-box” i2b2 implementation, technical staff at CHCO may create subpopulations by loading specific sets of patients or specific observations into an instance of i2b2 called a “project”. The methods available include:

1. Building an entirely separate instance of i2b2 for the data from a study with its own ontology, or mapped to an existing i2b2 ontology.
2. Extracting a “data mart” that contains only the data of interest for a study. This has a similar effect as #1, but may be easier.
3. Loading the data of a study into i2b2 and describing it within its own branch of the ontology. At CHCO the production i2b2 instance has a terminology branch called “Registries” whose immediate sub-trees each identify the data from a separate registry. Terms in the sub-trees may be built to reflect the observations recorded solely for a particular registry. Registry terms may be supplemented with other terms in the non-registry portion of the terminology tree, leveraging single-source-of-truth data like demographics from the larger i2b2 instance.

## 4.0 Results and Discussion

To date, we have created 30 patient lists covering the various use cases in Table 1. One limitation to the applicability of a patient list occurs when the source data for the list does not facilitate matching to the identity directory of the IDR. Our Clinical Research Unit had a database of patients interested in being research subjects. We suggested that it would be valuable to query on a list of those candidates in order to discover more qualifying information about them in the IDR. However, the quality of identifying information in the candidate database was too poor to make enough confident matches, and the project was shelved pending improvements in the candidate database.

Another limitation is that the purpose of a patient list must fit within the approved mission of the IDR itself. If the IDR, like some, is not intended for use in direct patient care, then governance must not allow patient lists with that purpose, although population health/practice management purposes might be acceptable.

In our implementation, data on a patient list must be updated by IDR staff, and adding more types of observations to the list requires rebuilding the list with both new and old data included. We have prototyped a way, which we call “Active Observations Projects”, for a project to design and load multiple patient lists on its own and then manually update their

data at any time. The facility, taken with the query capabilities of the IDR, could provide functionality similar to many patient registries, while also having access to the much broader and deeper data of the IDR. Technical challenges for this functionality involve coordinating real-time data changes with the periodic loading cycle, and runtime matching of new records to the correct ASID based on user-entered identifiers. Data used in queries should display or conceal real identifiers, depending upon the application. This will complicate the programming and governance related to “who is allowed to see what identifiers and when.”

## 5. Conclusions

Research enterprises have no doubt resorted to a variety of programming solutions to accomplish the tasks that patient lists helped us solve. The patient list concept works – almost automatically -- with an increasingly common model [3]: research IDRs with graphical query interfaces for attribute-centered queries and automated data de-identification. The enabling concept of the patient list is simple: that a pre-defined set of patient identities can be thought of as just another source of data in an IDR. Further, patient lists are straightforward data structures which should not be burdensome to load into an IDR. Based on our experience, patient lists can add much value to an IDR. We think that the patient list concept can be adapted to different IDRs. The concept might even be applicable to federated query systems [22, 23] -- if these have some kind of overall identity management, so that a list of patient identities can be transformed into whatever the federation uses.

## Acknowledgments

Authors RCH, PTZ and SM conceived of and implemented the patient lists at NJH. Author TDW supervised the work and is the primary writer. Author DB contributed the account of i2b2's capabilities. For the NJH IDR, Rob Carey made the query engine modifications and Shannon Holck designed validation tests. Support for this work was from National Jewish Health, with partial support for TDW from Colorado Clinical and Translational Science Institute, grant UL1RR025780 from NIH.

## References

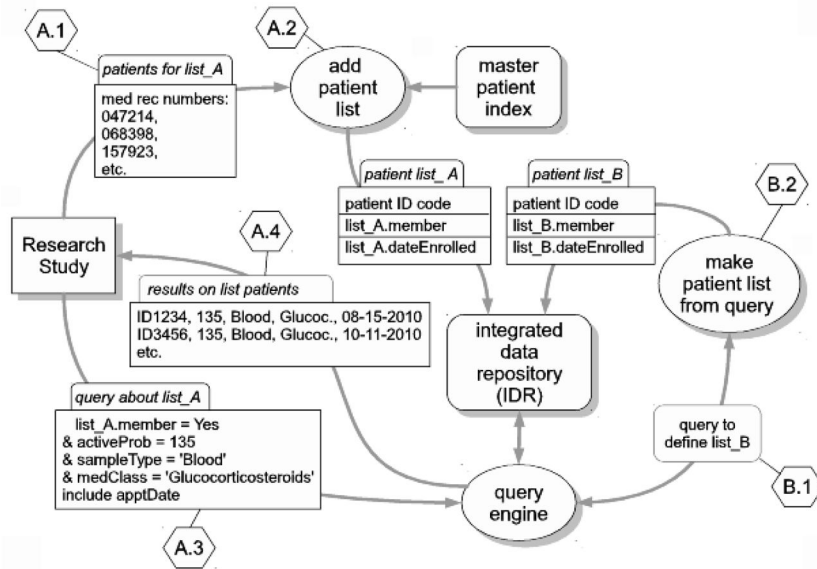
1. Safran C, Bloomrosen M, Hammond WE, Labkoff S, Markel-Fox S, Tang PC, Detmer DE. Toward a National Framework for the Secondary Use of Health Data: An American Medical Informatics Association White Paper. *J Amer Med Inform Assoc.* 2007; 14:1–9. [PubMed: 17077452]
2. Weiner MG, Embi PJ. Toward reuse of clinical data for research and quality improvement: the end of the beginning? *Ann Intern Med.* 2009; 151:359e60. [PubMed: 19638404]
3. MacKenzie S, Wyatt M, Schuff R, Tenenbaum J, Anderson N. Practices and Perspectives on Building Integrated Data Repositories: Results from a 2010 CTSA Survey. *J Amer Med Inform Assoc.* 2012; 19:e119ee124. [PubMed: 22437072]
4. Nadkarni PM, Brandt C. Data Extraction and Ad Hoc Query of an Entity–Attribute–Value Database. *J Amer Med Inform Assoc.* 1998; 5:511–527. [PubMed: 9824799]
5. Anhøj J. Generic design of web-based clinical databases. *J Med Internet Res.* 2003; 5:e27. [PubMed: 14713655]
6. Wade TD, Hum RC, Murphy JR. A Dimensional Bus model for integrating clinical and research data. *J Amer Med Inform Assoc.* 2011; 18(Suppl 1):18:i96–i102.
7. Dinu V, Nadkarni P. Guidelines for the effective use of entity-attribute-value modeling for biomedical databases. *Int J Med Inform.* 2007; 76:769e79. [PubMed: 17098467]
8. Murphy, SN. Data Warehousing for Clinical Research. In: Liu, L.; Tamer, OM., editors. *Encyclopedia of database systems.* Springer; 2009.

9. Roden DM, Pulley JM, Basford MA, Bernard GR, Clayton EW, Balsler JR, Masys DR. Development of a Large-Scale De-Identified DNA Biobank to Enable Personalized Medicine. *Clin Pharmacol Ther.* 2008; 4(3):362–369. [PubMed: 18500243]
10. Chute CG, Beck SA, Fisk TB, DN. The Enterprise Data Trust at Mayo Clinic: a semantically integrated warehouse of biomedical data. *J Am Med Inform Assoc.* 2010; 17(2):131–135. [PubMed: 20190054]
11. Kimball, R.; Reeves, L.; Ross, M.; Thornthwaite, W. *The Data Warehouse Lifecycle Toolkit.* Wiley; New York: 1998.
12. Cimino JJ, Ayres EJ. The clinical research data repository of the US National Institutes of Health. *Stud Health Technol Inform.* 2010; 160:1299–1303. [PubMed: 20841894]
13. Lowe HJ, Ferris TA, Hernandez PM, Weber SC. STRIDE – An Integrated Standards-Based Translational Research Informatics Platform. *AMIA 2009 Symp Proc.* 2009:391–395.
14. Murphy SN, Weber G, Mendis M, Gainer V, Chueh HC, Churchill S, Kohane I. Serving the enterprise and beyond with informatics for integrating biology and the bedside (I2B2). *J Amer Med Inform Assoc.* 2010; 17:124–130. [PubMed: 20190053]
15. Dhir R, Patel AA, Winters S, Bisceglia M, Swanson D, Aamodt R, Becich MJ. A multidisciplinary approach to honest broker services for tissue banks and clinical data: a pragmatic and practical model. *Cancer.* 2008; 113(7):1705–15. [PubMed: 18683217]
16. Kahn MG, Kaplan D, Sokol RJ, DiLaura RP. Configuration challenges: implementing translational research policies in electronic medical records. *Academic Medicine.* 2007; 82(7):661–669. [PubMed: 17595562]
17. Blumenthal D, Tavenner M. The “Meaningful Use” Regulation for Electronic Health Records. *N Engl J Med.* 2010; 363:501–504. [PubMed: 20647183]
18. Center for Medicare and Medicaid Services. [accessed 06 June, 2013] Stage 2 Eligible Professional Meaningful Use Core Measures: Measure 11 of 17. 2012. [http://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/downloads/Stage2\\_EPCore\\_11\\_PatientLists.pdf](http://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/downloads/Stage2_EPCore_11_PatientLists.pdf)
19. Gliklich, RE.; Dreyer, NA. *Registries for Evaluating Patient Outcomes: A User’s Guide.* Agency for Healthcare Research and Quality Publication; Rockville, Maryland: 2010. (AHRQ Publication; 10-EHC049)
20. Erdal BS, Liu J, Ding J, Chen J, Marsh CB, Kamal J, Clymer BD. A database de-identification framework to enable direct queries on medical data for secondary use. *Methods Inf Med.* 2012; 51(3):229–41. [PubMed: 22311158]
21. Murphy SN, Gainer V, Mendis M, Churchill S, Kohane I. Strategies for maintaining patient privacy in i2b2. *J Am Med Inform Assoc.* 2011; 18(Suppl):i103–8. [PubMed: 21169616]
22. Louie B, Mork P, Martin-Sanchez F, Halevy A, Tarczy-Hornoch P. Data integration and genomic medicine. *J Biomed Inform.* 2007; 40:5e16. [PubMed: 16574494]
23. Payne P, Ervin D, Dhaval R, Borlawsky T, Lai A. TRIAD: The Translational Research Informatics and Data Management Grid. *Appl Clin Inform.* 2011; 2(3):331–344. [PubMed: 23616879]

### Highlights for

*Using Patient Lists to Add Value to Integrated Data Repositories – T D Wade et al*

- Project-specific patient lists add value to integrated data repositories (IDRs).
- Query on a patient list restricts returned data to list members.
- List uses include chart review, practice management, phenotyping and data brokering.
- A patient list can be used for searching and linking to registry data in an IDR.
- Features needed to implement patient lists could be added to i2b2 and other IDRs.



**Figure 1. Data flow examples for patient lists in an integrated data repository**

There are two types of lists, one type (list\_A) generated externally to the IDR, and the other type (list\_B) generated from a query to the IDR itself. Hexagons number a sequence of events for each type of list. B.1 is a list-defining internal query, and B.2 is the creation of a list from that query. The A sequence shows creation of a list (A.1, A.2) from data external to the IDR, followed by a query about the list (A.3), and the return of results (A.4) from that query. The identities in the external list are coded by reference to data in the Master Patient Index. Coded ID's and list membership variables are fed into the IDR, and list data (e.g., the *list\_A.member* Boolean) can then be used by a study in queries to restrict returned data to only list members.

**Table 1**  
**Use cases for patient lists in an IDR**

These are cases we have implemented on our IDR.

List use case	Objectives	List origin
1. List from IDR Query	Internal query defining a study cohort. Researchers can return for additional data without cluttering results with original sample-defining values	Internal query
2. Add to Existing Study Database	Get EHR data for consented study participants	External (study)
3. EHR-derived List	Get population data on a clinically relevant group of patients for practice management or "Meaningful Use"	External (EHR)
4. Honest Brokering	Allows dissemination of brokered de-identified data and connection with IDR data	External
5. Biobanking	Add phenotypes to biosamples	Internal or external
6. Access to Study/Registry Data	Allows queries of study and EHR data together	Internal query
7. Promoting Collaboration	Share aggregate information on a population to promote collaborator interest	Internal or external

**Table 2**  
**Needs for specified capabilities in implementing patient lists**

The set of patients in a list can originate from an internal IDR query or from an external source such as a study or clinical database.

Functional capability	Origin of list	
	Internal query	External source
1. <b>Feedback from Internal Query:</b> Feed patient IDs from a query result back to the IDR	needed	not needed
2. <b>Permanent Identifier:</b> Exposability of a permanent ID in the IDR	needed	needed
3. <b>Adding Attributes to IDR:</b> Allow adding queryable attributes to IDR	needed	needed
4. <b>Convert External ID:</b> Ability to convert external ID to IDR's standard ID	not needed	needed
5. <b>Merge Multiple Queries:</b> Merge multiple accessions of a list on common identifiers	needed	needed
6. <b>Modifying List Membership:</b> Allow modification of list membership over time	less likely	more likely
7. <b>Policies on De-identification:</b> Implement policies on levels of de-identification on data returned from list-based query.	needed	needed
8. <b>Restricted Use:</b> Ability to restrict use of a list to authorized persons	needed	needed
9. <b>Avoid Inadvertent Exposure:</b> Use of a list in a query must not inadvertently expose unauthorized data.	needed	needed