Informatics for Integrating Biology and the Bedside

i2b2 Design Document

Project Management (PM) Cell

Table of Contents

DOCUMENT MANAGEMENT

Revision Number	Date	Author	Description of change
1.6.1	07/22/10	Janice Donahoe	Created 1.6 version of document.
1.6.2	09/22/11	Mike Mendis	Changed ADMIN roles

1. INTRODUCTION

The i2b2 design document describes the requirements, technical functionality, and the intended capabilities of the project management routines that reside in the i2b2 Web Client. This document is to be used as a guideline and continuing reference point as the developers write the code and quality assurance writes the test plans.

2. PROJECT MANAGEMENT CELL CONCEPTS

2.1 Objectives of Project Management Cell

Information in the project management cell is related to the setup and maintenance of the hive, projects, users and security.

2.2 Identification of Users

2.2.1 Clinical Researcher

- Member of the research team who is setup with access to the project in i2b2.
- Their access role is "**USER**".
- Can edit their user profile.

2.2.2 Manager of Clinical Researcher

- Manager of the research team.
- Their access role is "**MANAGER**".
- Can create and edit users associated to their project.
- Can create and edit project related information.

2.2.3 Administrator

- May or may not be part of the research team.
- Responsible for the administrative tasks related to the i2b2 Workbench.
- Their access role is "**ADMIN**" and Project ID is '**@**'
- Can create and edit users associated to any project.
- Can create and edit all projects.
- Can create and edit hive information.

3. REQUIREMENTS

3.1 Design Requirements

The project management module resides within the i2b2 Web Application. The following section outlines some of the basic design requirements.

3.1.1 Pages

Individual pages will be used to capture information throughout the project management module. These pages will be specific to the data that is collected.

3.1.1.1 NAVIGATION BAR

A **navigation bar** will remain on the left side of all the pages. This can be used if a user wants to quickly access a particular page.
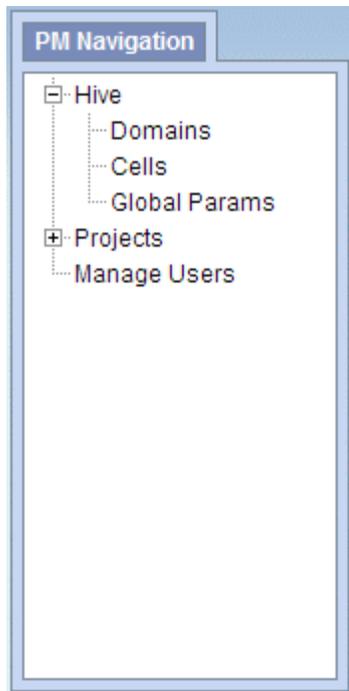

Figure 1 Navigation Bar

3.1.1.2 PAGE HEADING

At the top of each page will be a tab that contains the name of the page and its location.

In the example shown below, the heading tells us the user is on the **project parameters page** for the "demo" project.
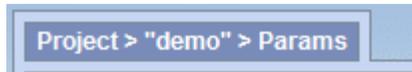
Figure 2 Page Heading

3.2 Functional Requirements

To assist with workflow and overall ease of use, individuals will now be able to easily save data and parameters for the hive, projects and users. An outline of what can be stored in the PM database and the related functionality is listed below.

3.2.1 Global Setup

In i2b2 Project Management, **global data** refers to information that is <u>not</u> specific to any one hive, project, or user. The information setup here is used by all.

3.2.1.1 GLOBAL PARAMETERS

- Parameters that are to be used by all hives and projects will be defined in the *global parameters page*.
- The data will be stored in the **PM_GLOBAL_PARAMS** table.
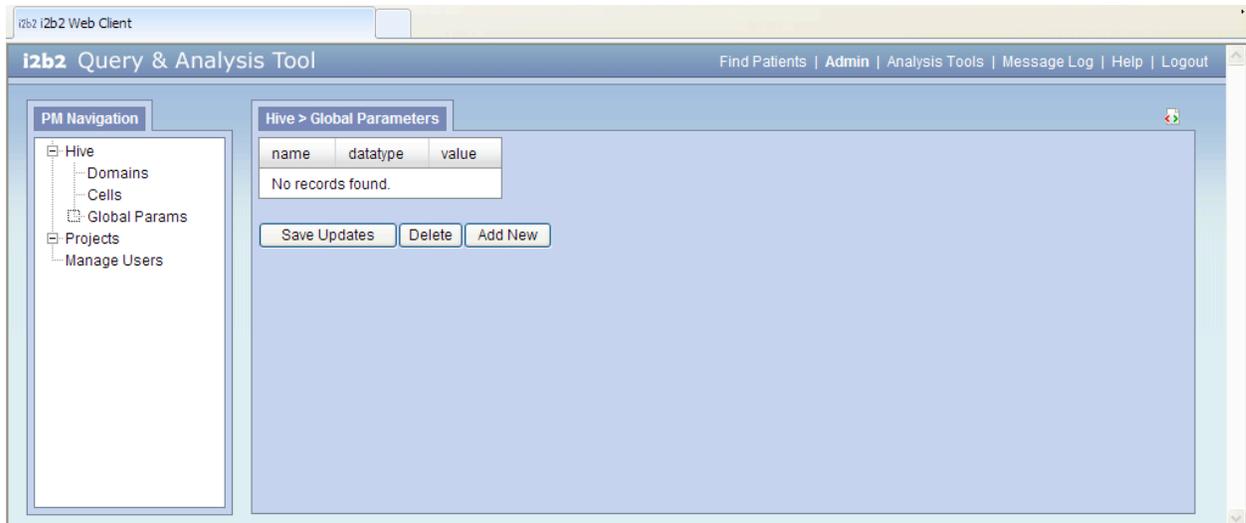
Figure 3 Global Parameters Page

3.2.2 Users Setup

In order to use the i2b2 application a user has to be setup in project management and given access to one or many projects. In addition, variables can be setup for a user that can be used across multiple projects or specific to one project.

3.2.2.1 USER DATA

- General information about the user will be stored in the **PM_USER_DATA** table.
- The **user_id** will be referenced in other tables such as *PM_USER_PARAMS* and *PM_PROJECT_USER_ROLES*.
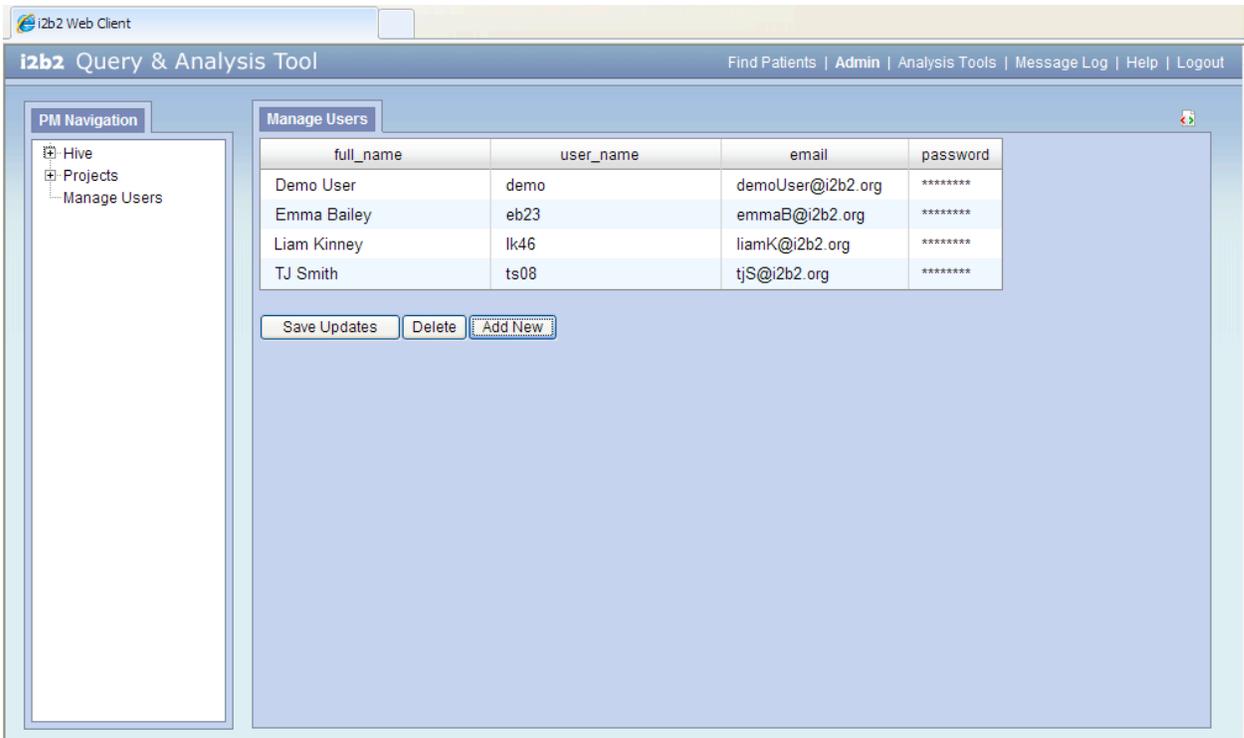
Figure 4 User Data Page

3.2.2.2 USER PARAMETERS

- Parameters entered here are *specific to the user* and are not specific to any one project.
- The information will be stored in the **PM_USER_PARAMS** table.
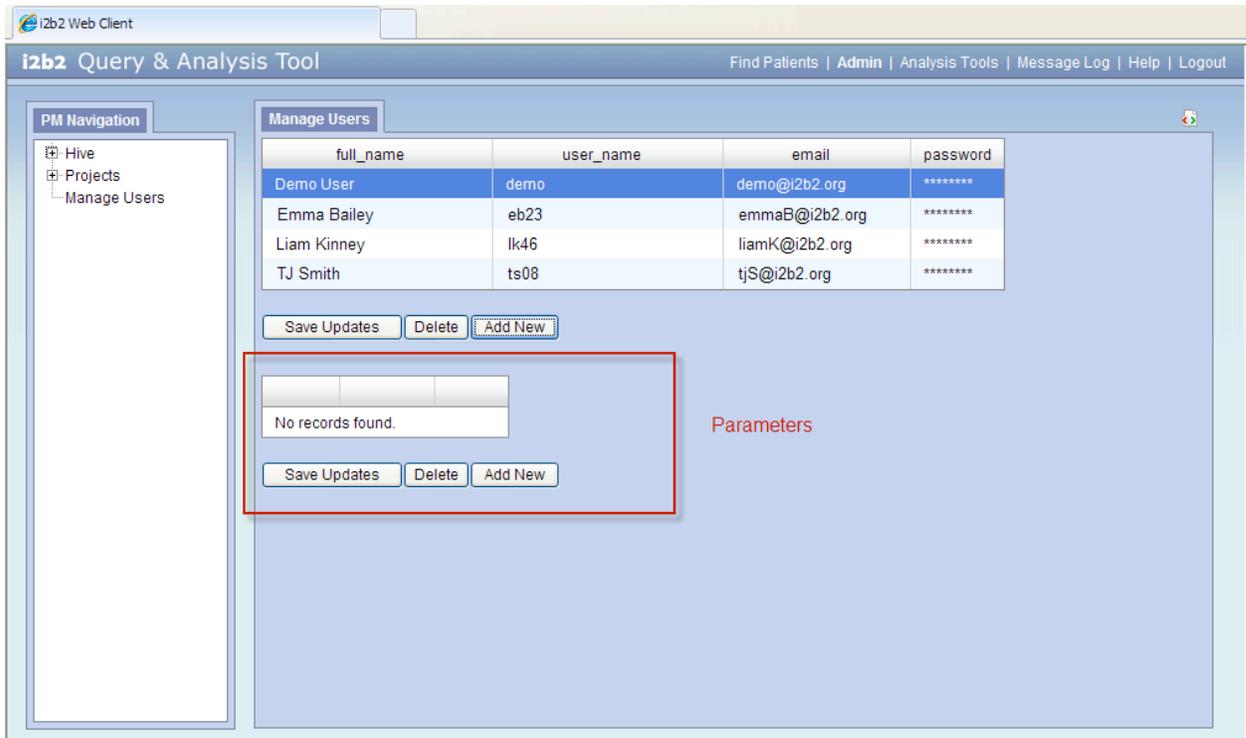
Figure 5 User Parameters Section

3.2.2.3 USER PROJECT ROLES

- The **project user page** captures information regarding which users have access to the project and their role.
- The information will be stored in the **PM_PROJECT_USER_ROLES** table.

Detailed information about user roles can be found in the project section of this document.

3.2.3 Hive Setup

3.2.3.1 HIVE DATA

- The domain page captures general information about the hive.
- Information captured includes the *domain id*, *domain name*, *environment*, and *help URL*.
- The information will be stored in the **PM_HIVE_DATA** table.
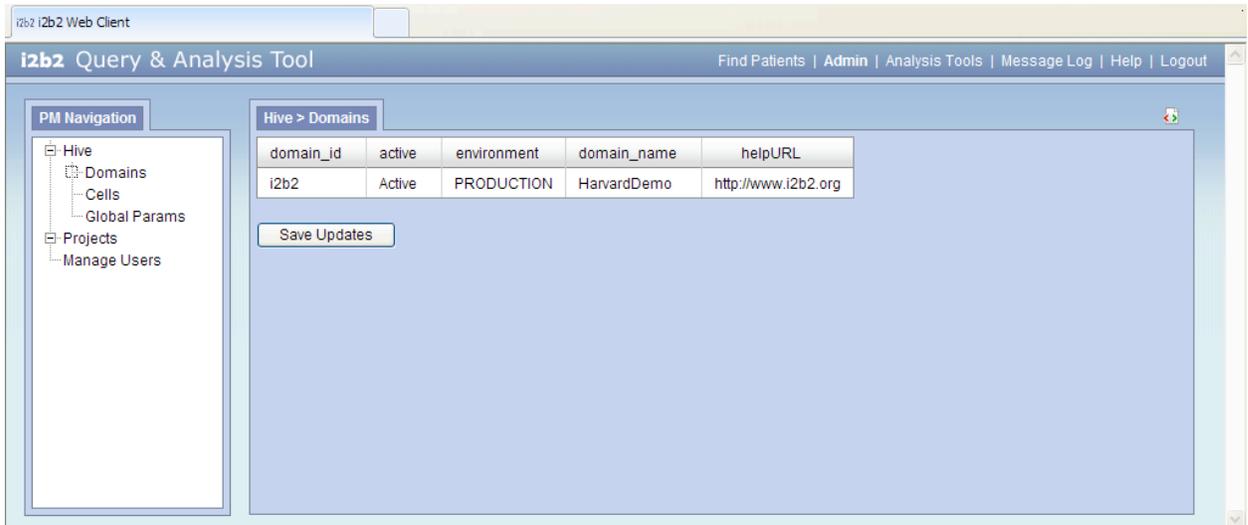
Figure 6 Hive Data (Domain) Page

3.2.3.2 HIVE PARAMETERS

- Parameters entered here are *specific to the hive (domain)* and are not specific to any one project or user.

- The hive parameters can be added from the **hive (domain) data page**. Once you click on the domain an additional section for entering the parameters will appear at the bottom of the page.

- The information will be stored in the **PM_HIVE_PARAMS** table.
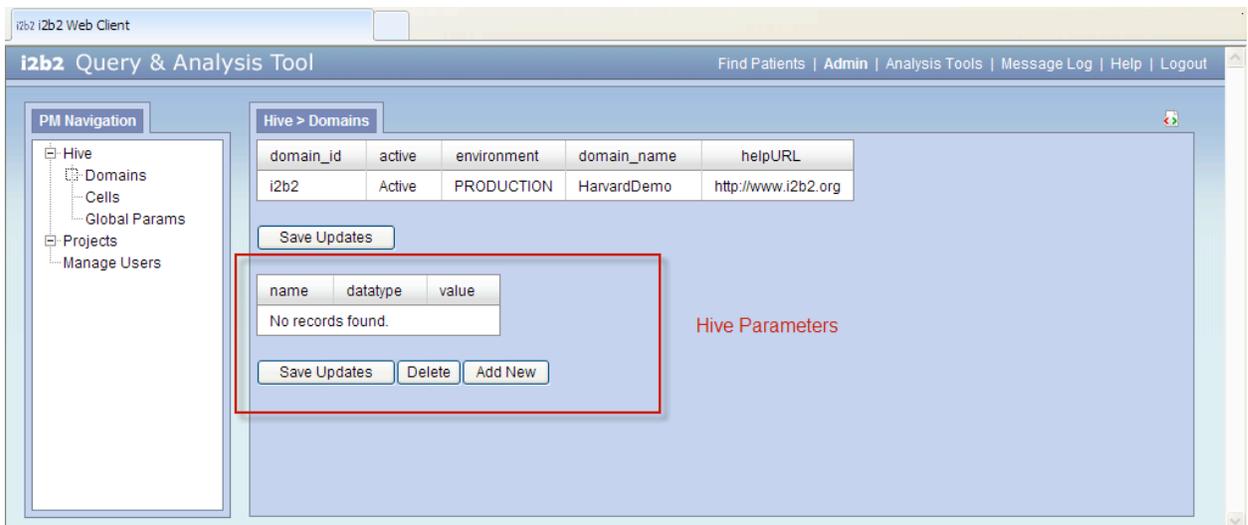
Figure 7 Hive Parameters Section

3.2.3.3 CELL DATA

- ▪ The **cell data page** captures information and registers the cells associated to the hive.

- ▪ The information will be stored in the **PM_CELL_DATA** table.
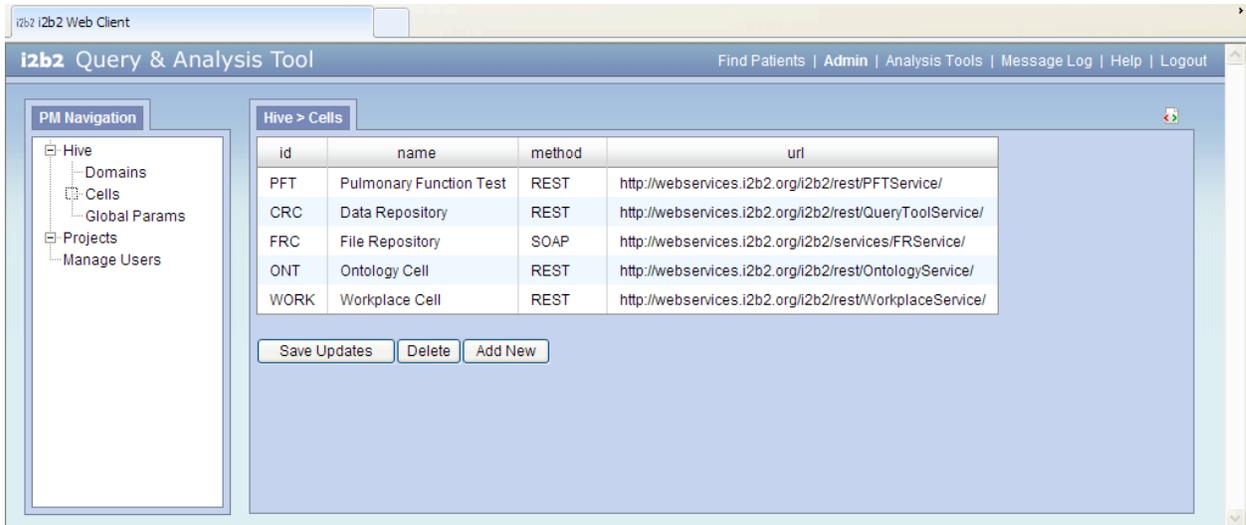
Figure 8 Cells Data Page

3.2.3.4 CELL PARAMETERS

- ▪ Parameters entered here are *specific to the cell* and are not specific to any one project or user.

- ▪ The cell parameters can be added from the **cell data page**. Once you click on a cell an additional section for entering the parameters will appear at the bottom of the page.

- ▪ The information will be stored in the **PM_CELL_PARAMS** table.
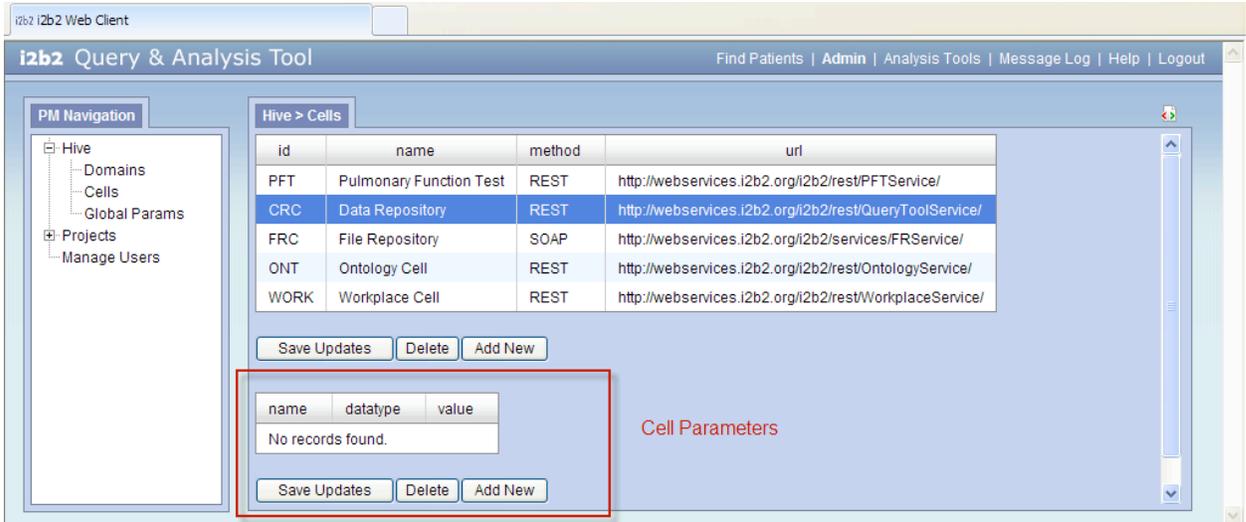
Figure 9 Cell Parameters Section

3.2.4 Project Setup

A hive can have multiple projects setup.

3.2.4.1 PROJECT DATA

- The **project data page** captures general information about the project.
- The information will be stored in the **PM_PROJECT_DATA** table.
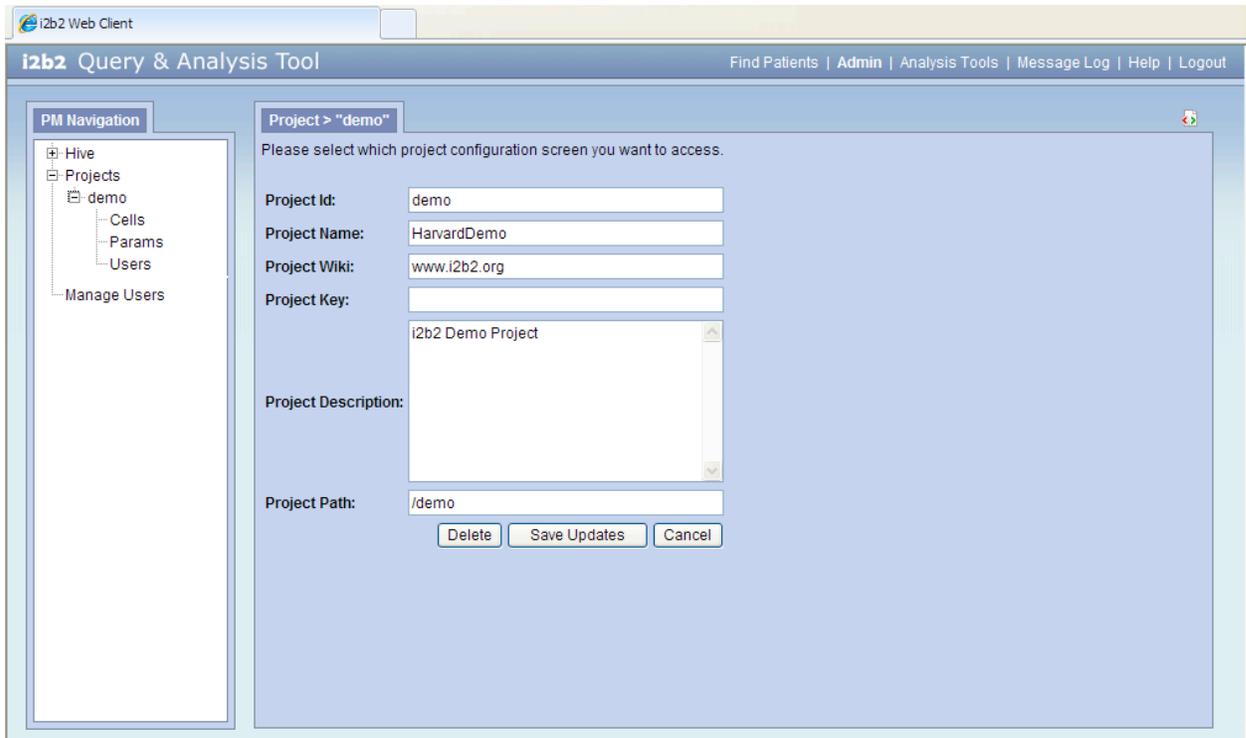
Figure 10 Project Data Page

3.2.4.2 PROJECT PARAMETERS

- Parameters entered here are *specific to the project*.
- The project parameters can be added from the **project parameters page**.
- The information will be stored in the **PM_PROJECT_PARAMS** table.
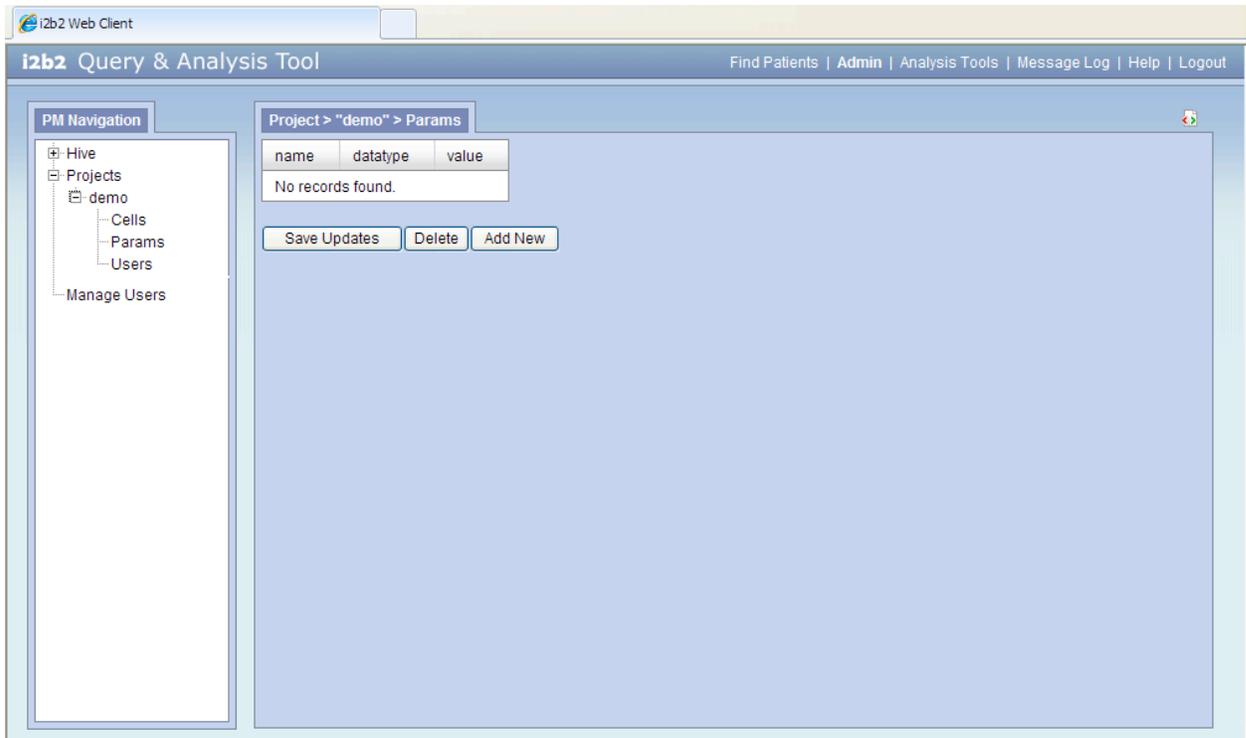
Figure 11 Project Parameters Page

3.2.4.3 PROJECT USERS

- The **project users' page** captures information regarding which users have access to the project and their role. *(see the next section for detailed information regarding roles)*

- The information will be stored in the **PM_PROJECT_USER_ROLES** table.
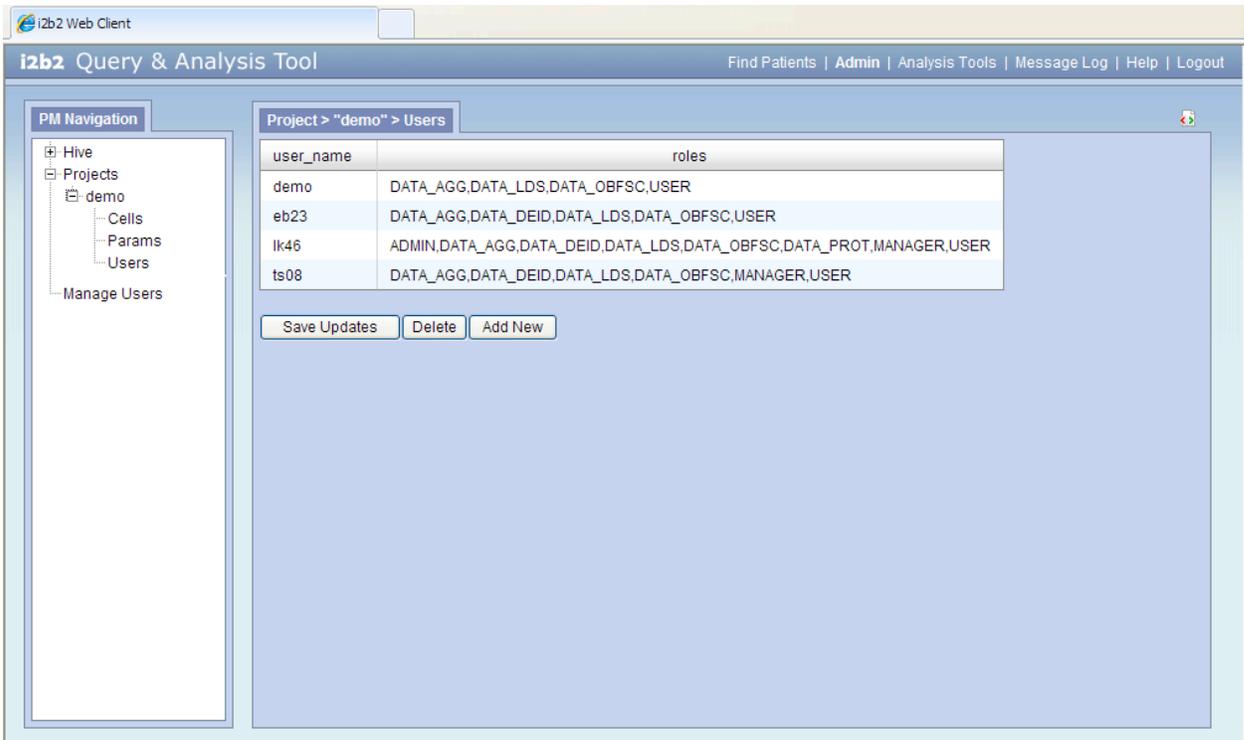
Figure 12 Project Users Page

3.2.4.3.1 Project User Roles

Each user will have at least two roles per user_id and product_id combination. These two roles can be further defined as a **Data Protection role** and a **Hive Management role**.

The data protection role/path establishes the detail of data the user can see while the hive management role/path defines their level of functionality the user has in a project.

Data Protection Track	
Role	**Access Description**
DATA_OBFSC	OBFSC = Obfuscated ▪ The user can see aggregated results that are obfuscated (example: patient count). ▪ The user is limited on the number of times they can run the same query within a specified time period. If the user exceeds the maximum number of times then their account will be locked and only the Admin user can unlock it.

DATA_AGG	AGG = Aggregated • The user can see aggregated results like the patient count. • The results are <u>not</u> obfuscated and the user is <u>not</u> limited to the number of times they can run the same query.
DATA_LDS	LDS = Limited Data Set • The user can see all fields except for those that are encrypted. • An example of an encrypted field is the *blob fields* in the *fact* and *dimension tables*.
DATA_DEID	DEID = De-identified Data • The user can see all fields including those that are encrypted. • An example of an encrypted field is the *blob fields* in the *fact* and *dimension tables*.
DATA_PROT	PROT = Protected • The user can see all data, including the identified data that resides in the Identity Management Cell.

Hive Management Track	
Role	**Access Description**
USER	Can create queries and access them if he/she is the owner of the query.
MANAGER	Can create queries and can access queries created by different users within the project.

☞ *Additional roles can be added to the* **PM_PROJECT_USER_ROLES** *table but there will not be any recognized hierarchy to those roles.*

3.2.4.4 PROJECT USER PARAMETERS

▪ Parameters entered here are *specific to the user **and** the project*.

▪ The project's user parameters can be added from the **project users' page**. Once you click on a project an additional section for entering the user parameters will appear at the bottom of the page.

▪ The information will be stored in the **PM_PROJECT_USER_PARAMS** table.

Figure 13 Project Users Parameters Section

4. TABLES

4.1 Global Tables

4.1.1 PM_GLOBAL_PARAMS

- The PM_GLOBAL_PARAMS table is commonly used in a production system to specify default values for various users.

- Parameters are _not_ specific to a hive or project.

- There is one table per project management cell.

PM_GLOBAL_PARAMETERS		
PK	**id**	**int**
	param_name	**varchar(50)**
	project_path	**varchar(50)**
	value	varchar(255)
	datatype_cd	varchar(255)
	can_override	int

4.1.1.1 DATATYPE CODE

All the parameter tables contain a column called datatype_cd. The value (code) entered in this column is used to determine what the object is. The following is a list of datatype codes.

Datatype_cd	Datatype_Desc
T	Text that is less than 2000 characters
M	Reference to Text file (>2000Characters)
C	Reference to binary file
N	Numeric (Float)
D	Date (yyyy-MM-ddTHH:mm:ss)
I	Integer
B	Boolean (T/F)

IP	(Reserved)
EP	(Reserved)
RTF	Reference to RTF File
XLS	Reference to Microsoft Excel
XML	Reference to XML file or blob
DOC	Reference to Microsoft Word Document

4.1.2 PM_USER_DATA

- The **PM_USER_DATA** table contains general Information about the user
- There is one row per user; data with special tags can occur once per user is also in that row.
- The **user_id** is associated with a humanly legible name in the format that they user prefers.
- The **password** is the MD5 hash for the user's password.
- The **email** is the users preferred email address.

PM_USER_DATA		
PK	**user_id**	**varchar(50)**
	full_name	varchar(255)
	password	varchar(255)
	email	varchar(255)

4.1.3 PM_USER_PARAMS

- The **PM_USER_PARAMS** table contains parameters that are specific to the user.
- The **id** is an auto generated number
- The **user_id** is associated with a humanly legible name in a format preferred by the user.
- The **param_name_cd** is associated with the object in the **code_lookup** table

PM_USER_PARAMS		
PK	**id**	**int**
	user_id	**varchar(50)**
	param_name_cd	**varchar(50)**
	value	varchar(255)
	datatype_cd	varchar(50)

☞ *A "_params" table contains name-value pairs associated with a user that can occur more than once or vary from user to user. An "@" in place of a user_id means that the parameter applies to all users not explicitly listed.*

4.2 Hive Tables

4.2.1 PM_HIVE_DATA

- The **PM_HIVE_DATA** table contains general information about the hive.
- The **domain_id** column is expected to be highly unique (at least 20 characters in a random sequence).

☞ *The domain_id needs to be unique across multiple enterprises which is why it should be at least 20 characters in a random sequence.*

- The name in the **domain_name** column represents the hive's domain identifier.
- Each **domain_id** row contains the **domain_name** for the hive and the **environment_cd**.
- The **environment_cd** can be easily switched by changing the active configuration.

PM_HIVE_DATA		
PK	**domain_id**	**varchar(50)**

PM_HIVE_DATA	
environment_cd	**varchar(255)**
domain_name	**varchar(255)**
helpurl	varchar(255)
active	int

The enumerated values that represent the environment are:

Value	Description
PRODUCTION	The operational system that is used by all i2b2 users. This environment contains actual / real data.
TEST	A testing or "staging" environment. Used for testing changes before they are moved to production. The data may or may not be real.
DEVELOPMENT	The development environment is used for developing and upgrading the systems. Only test data resides in this environment.
STOPPED	Temporarily down
INACTIVE	Undefined down period
ARCHIVED	Not active, not anticipated to be restarted in the future.

4.2.2 PM_HIVE_PARAMS

- The **PM_HIVE_PARAMS** table is one of several "_params" tables in the PM cell.
- These parameters are associated with the various "**domain_id's**" from the **PM_HIVE_DATA** table.
- This table allows users to specify *name-value pairs* associated with various **PM_HIVE_DATA** configurations.
- These parameters are not specific to any project or user.

PM_HIVE_PARAMS		
PK	**id**	**int**
	domain_id	**varchar(50)**
	param_name_cd	**varchar(50)**
	value	varchar(255)
	datatype_cd	varchar(50)

4.2.3 PM_CELL_DATA

- The **PM_CELL_DATA** table contains general information about the cells.
- There is one row for each cell_ID.
- A hive may have several cells of the same type, but they will be distributed to the projects according to their project path.
- In version 1.x of the Hive software, the XML will only return the cell which is the most specific for that project.

EXAMPLE:

If 3 ONT cells exist:

1. one with project "/hive"
2. one with project "/hive/asthma"
3. one with project "/hive/asthma/snm0"

Only the ONT cell specified by "/hive/asthma/snmo" will be returned for project "snm0". It essentially allows certain projects to be "diverted" from the mainstream cells.

PM_CELL_DATA		
PK	**cell_id**	**varchar(50)**
PK	**project_path**	**varchar(255)**
	name	varchar(255)
	URL	varchar(255)
	method_cd	varchar(255)
	can_override	int

4.2.4 PM_CELL_PARAMS

- The **PM_CELL_PARAMS** table contains cell specific parameters.

- Follows the same rules for the return of specific cell-associated *name-value* pairs.

- Commonly used in a production system to specify default values for various users.

PM_CELL_PARAMS		
PK	**id**	**int**
	cell_id	**varchar(50)**
	project_path	**varchar(255)**
	param_name_cd	**varchar(50)**
	value	varchar(255)
	datatype_cd	varchar(50)
	can_override	int

4.3 Project Tables

4.3.1 PM_PROJECT_DATA

- The **PM_PROJECT_DATA** table contains general information about the project.

- There is only one row per **project_id**; data with special tags that can occur once per project is also in that row.

- The **project_ID** is a unique identifier.

- The **project_name** is a short human legible name for the project.

- The **project_key** is the MD5 hash for a project key used to encrypt data.

- The **project_wiki** contains a URL for the project wiki.

- The **project_path** is used in queries to find the value for cell and global parameters. Below is a more detailed explanation and example.

- The **project_description** is a long description that can be searched.

pm_project_data		
PK	**project_id**	**varchar(50)**
	project_name	varchar(255)
	project_key	varchar(255)
	project_wiki	varchar(255)
	project_path	varchar(255)
	project_description	varchar(2000)

project_path

- Represents a special string in the format /"Project_name"/"sub-project_name"/ etc.

- Used primarily to manage default parameters that the users will see in regards to the interactions they have with a cell.

- Will be used in queries to find the value for cell and global parameters.

- The hive is initialized with default values for all parameters with a blank/null project, following which the params may be overridden by adding a project.

- When the query returns, only one value for each param is returned, but that is the most specific available.

EXAMPLE:

project_path	Value
/	Overall hive default
/ASTH	Asthma default
/HTN	Hypertension default
/ASTH/SNM0	Sub project for Asthma

If the above table was queried by a member of the asthma project who is not a member of the SNM0 sub project then "Asthma default" would be obtained. If the project "MDD" was to query the table as a member of the major depression project,

the value of "Overall hive default" will be obtained because a more specific entry does not exist for the user.

4.3.2 PM_PROJECT_PARAMS

- Project specific parameters
- Each project can have its own set of parameters

pm_project_params		
PK	**id**	**int**
	project_id	**varchar(50)**
	param_name_cd	**varchar(50)**
	value	varchar(255)
	datatype_cd	varchar(50)

4.3.3 PM_PROJECT_USER_ROLES

- Users associated to the project and their respective role.
- Data is specific to the project.
- The **PM_PROJECT_USER_ROLES** table will have at least two roles per *user_id* and *product_id* combination.
- "Data Protection Track" role establishes the detail of data that may be seen by the user.
- "Hive Management Track" role establishes the amount of control a user has in a project.
- "Custom Track" role establishes individual roles which might be needed for a specific application
- The roles are in hierarchical order, similar to the Data Protection Track, so that the roles on top gain the permissions from the ones below.
- Although the table will only contain the role for the highest level of detail the user can see, other roles to see less detailed data will also be automatically granted.
- If a cell requires other unique roles for a user, these can be added to the **PM_PROJECT_USER_ROLES** table, but there will be no recognized hierarchy to those roles.

- For roles that span across all projects, the **project_ID** column in the **PM_PROJECT_USER_ROLES** table gets an "@" sign.

- For roles that span across all users (for a project), the **user_ID** column in the **PM_PROJECT_USER_ROLES** table gets an "@" sign.

- At a minimum, a user/project combination will have a role from at least each of the two tracks listed below:

Data Protection Track	Hive Management Track	Custom Track
DATA_PROT	ADMIN	EDITOR
DATA_DEID	MANAGER	
DATA_LDS	USER	
DATA_AGG		
DATA_OBFSC		

PM_PROJECT_USER_ROLES		
PK	**project_id**	**varchar(50)**
PK	**user_id**	**varchar(50)**
PK	**user_role_cd**	**varchar(255)**

4.3.4 PM_PROJECT_USER_PARAMS

- User parameters that are specific to the project

PM_PROJECT_USER_PARAMS		
PK	**id**	**int**
	project_id	**varchar(50)**
	user_id	**varchar(50)**
	param_name_cd	**varchar(50)**
	value	varchar(255)

PM_PROJECT_USER_PARAMS	
datatype_cd	varchar(50)

4.4   Access/Restriction Tables

4.4.1 PM_ROLE_REQUIREMENT

- In order to assign the permissions to edit the PM tables, entries are made into the **PM_ROLE_REQUIREMENT** table as shows below.

- There is a column to record "read" and a column to record "write" permissions.

- The "**Manager**" permission is only allowed in tables that have a *project_id* or a *project_path*.

- Each *table name* and *column name* are specified in the table.

- If there is an @ in the *table_cd* or the *column_cd* column for a specific code, that is similar to a wild card character and implies that the code can be used in any column with a similar name in any table, or any column in the table.

- *name_char* can be used for a description of the purpose and function of a column (varchar datatype for lookup/searching capabilities).

PM_ROLE_REQUIREMENT		
PK	**table_cd**	**varchar(50)**
PK	**column_cd**	**varchar(50)**
PK	**read_hivemgmt_cd**	**varchar(50)**
PK	**write_hivemgmt_cd**	**varchar(50)**
	name_char	varchar(2000)

4.5   Audit Tables

Each table in the project management database has a limited auditing of its own management through the change management columns. These columns record the date-time of the initial entry, the date-time of the latest change, the type of change (delete, update, create), and the user_id of the person who performed the change.

The columns are filled in when changes are made by the web services. There is no guarantee that the columns are filled in during direct database access.

Transaction Columns	
entry_date	datetime
change_date	datetime
changeBy_char	varchar(50)
status_cd	varchar(50)

The codes in the status_cd column are:

Code	Description
D	Delete
U	Update
C	Create

☞ *The transaction columns can be made into a complete audit trail by adding "read" to the changes recorded, and by creating a new record every time a CRUD transaction is performed. The primary key should be changed to include all of these columns, and only the latest change should be read.*

4.6   Future Miscellaneous Tables

4.6.1 PM_CODE_LOOKUP

- Any of the "_cd" columns have descriptions available in the **PM_CODE_LOOKUP** table.

- This table is available for client applications to obtain the list of codes that may be entered by the user for the column of a specific table.

- The **name_char** has the descriptive name of the code.

- The **lookup_keys_char** is a string with a bar-type-delimiter that allows strings to be used to lookup subsets of codes.

- If there is an "@" in the *table_cd* or the *column_cd* column for a specific code, that is similar to a wild card character and implies that the code can be used in any column with a similar name in any table, or any column in the table.

pm_code_lookup		
PK	**table_cd**	**varchar(50)**
PK	**column_cd**	**varchar(50)**
PK	**code_cd**	**varchar(50)**
	name_char	varchar(2000)
	lookup_keys_char	varchar(2000)