

**i2b2**

---

**Software Documentation**

**i2b2 Design Document**  
**Project Management (PM) Cell**

*Document Version:* 1.7.00-002  
*i2b2 Software Version:* 1.7.00

# TABLE OF CONTENTS

- DOCUMENT MANAGEMENT ..... 5**
- 1 INTRODUCTION ..... 6**
- 2 PROJECT MANAGEMENT CELL CONCEPTS ..... 7**
  - 2.1 OBJECTIVES OF PROJECT MANAGEMENT CELL ..... 7
  - 2.2 IDENTIFICATION OF USERS ..... 7
    - 2.2.1 *Clinical Researcher* ..... 7
    - 2.2.2 *Manager of Clinical Researcher* ..... 7
    - 2.2.3 *Administrator* ..... 7
- 3 REQUIREMENTS ..... 8**
  - 3.1 DESIGN REQUIREMENTS ..... 8
    - 3.1.1 *Pages* ..... 8
      - 3.1.1.1 Navigation Bar ..... 8
      - 3.1.1.2 Page Heading ..... 8
  - 3.2 FUNCTIONAL REQUIREMENTS ..... 9
    - 3.2.1 *Global Setup* ..... 9
      - 3.2.1.1 Global Parameters ..... 9
    - 3.2.2 *Users Setup* ..... 10
      - 3.2.2.1 User Data ..... 10
      - 3.2.2.2 User Parameters ..... 11
      - 3.2.2.3 User Project Roles ..... 12
    - 3.2.3 *Hive Setup* ..... 12
      - 3.2.3.1 Hive Data ..... 12
      - 3.2.3.2 Hive Parameters ..... 13
      - 3.2.3.3 Cell Data ..... 14
      - 3.2.3.4 Cell Parameters ..... 15
    - 3.2.4 *Project Setup* ..... 15
      - 3.2.4.1 Project Data ..... 15
      - 3.2.4.2 Project Parameters ..... 16
      - 3.2.4.3 Project Users ..... 17
      - 3.2.4.4 Project User Roles ..... 18
      - 3.2.4.5 Project User Parameters ..... 19
- 4 TABLES ..... 21**
  - 4.1 GLOBAL TABLES ..... 21
    - 4.1.1 *PM\_GLOBAL\_PARAMS Table* ..... 21
      - 4.1.1.1 Datatype Code ..... 21
    - 4.1.2 *PM\_USER\_DATA Table* ..... 22
    - 4.1.3 *PM\_USER\_PARAMS Table* ..... 23
  - 4.2 HIVE TABLES ..... 23
    - 4.2.1 *PM\_HIVE\_DATA Table* ..... 23
    - 4.2.2 *PM\_HIVE\_PARAMS Table* ..... 25
    - 4.2.3 *PM\_CELL\_DATA Table* ..... 25
    - 4.2.4 *PM\_CELL\_PARAMS Table* ..... 26
  - 4.3 PROJECT TABLES ..... 27
    - 4.3.1 *PM\_PROJECT\_DATA Table* ..... 27
    - 4.3.2 *PM\_PROJECT\_PARAMS Table* ..... 29

- 4.3.3 *PM\_PROJECT\_USER\_ROLES Table* .....29
- 4.3.4 *PM\_PROJECT\_USER\_PARAMS Table*.....30
- 4.4 ACCESS / RESTRICTION TABLES .....31
  - 4.4.1 *PM\_ROLE\_REQUIREMENT Table* .....31
- 4.5 AUDIT TABLES .....32
- 4.6 FUTURE MISCELLANEOUS TABLES.....33
  - 4.6.1 *PM\_CODE\_LOOKUP Table* .....33

# DOCUMENT MANAGEMENT

| <b>Revision Number</b> | <b>Date</b> | <b>Author</b>  | <b>Description of change</b>                      |
|------------------------|-------------|----------------|---|
| 1.7.1                  | 11/07/12    | Janice Donahoe | Created the 1.7 version of the document           |
| 1.7.00-002             | 08/12/2015  | Janice Donahoe | Fixed some minor spelling and grammatical errors. |

# 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**.
- They can edit their user profile.

#### 2.2.2 Manager of Clinical Researcher

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

#### 2.2.3 Administrator

- They may or may not be part of the research team.
- They are responsible for the administrative tasks related to the i2b2.
- Their access role is **ADMIN** and the project id is @.
- They can create and edit users associated to any project.
- They can create and edit all projects.
- They can create and edit hive information.

## 3 REQUIREMENTS

### 3.1 Design Requirements

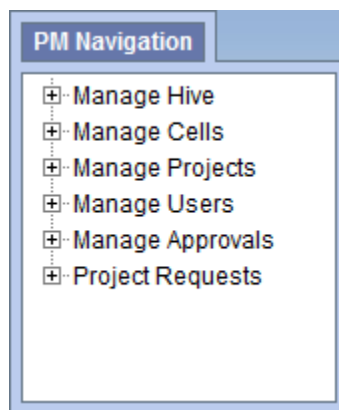
The **Project Management Module** is also referred to as the **Admin Module** and it resides within the i2b2 Web Application. The following section outlines some of the basic design requirements.

#### 3.1.1 Pages

Individual pages will be used to capture information throughout the Admin 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.



##### 3.1.1.2 Page Heading

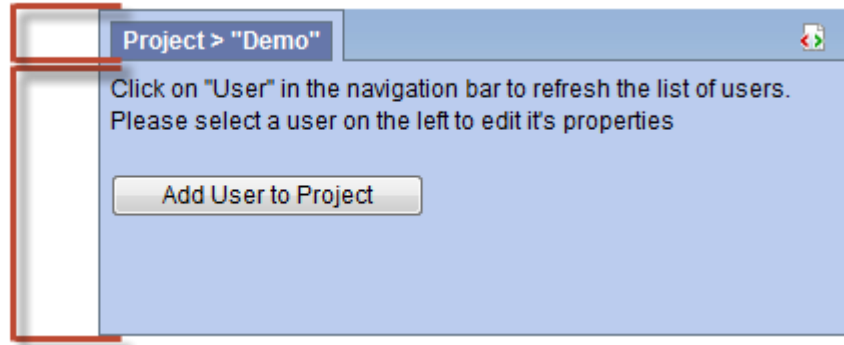
At the top of each page will be a *tab* that contains the name of the page and its location.

In the example shown below, the heading tells us the user is in *Manage Projects* for the *demo project*. The page itself tells us they are on the *User page*



Page Heading

Page Information



## 3.2 Functional Requirements

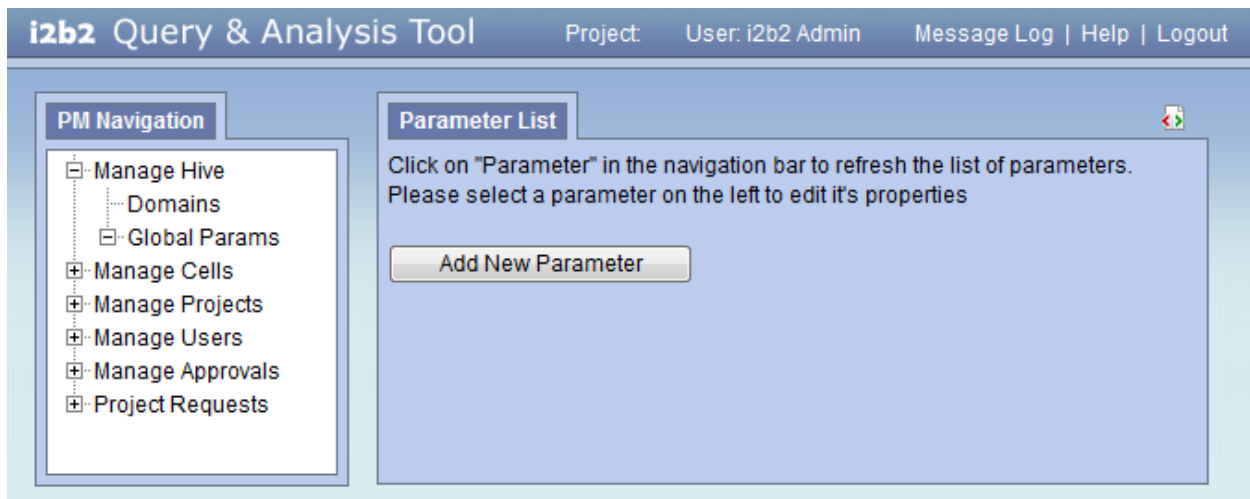
To assist with the workflow and overall ease of use, individuals will now be able to easily save data and parameters for the hive, projects and users. An outline of what can be stored in the PM database and the related functionality is listed below.

### 3.2.1 Global Setup

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

#### 3.2.1.1 Global Parameters

- Any 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.

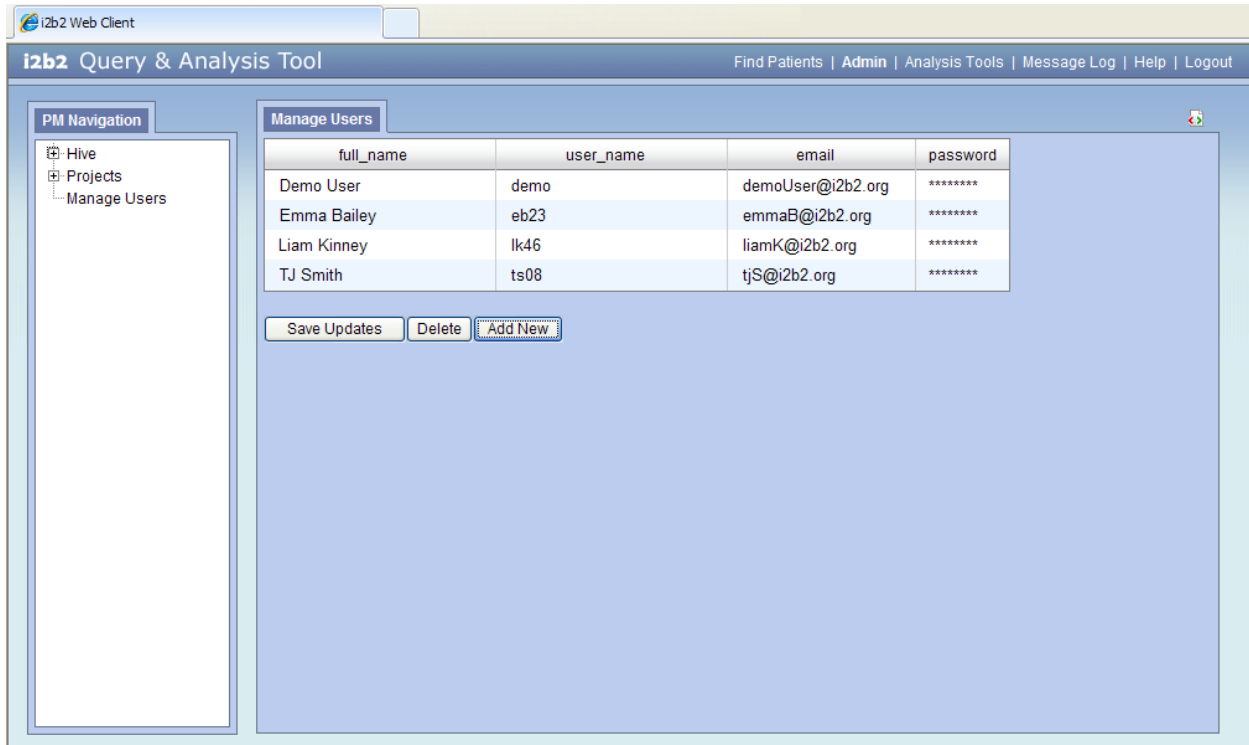


## 3.2.2 Users Setup

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

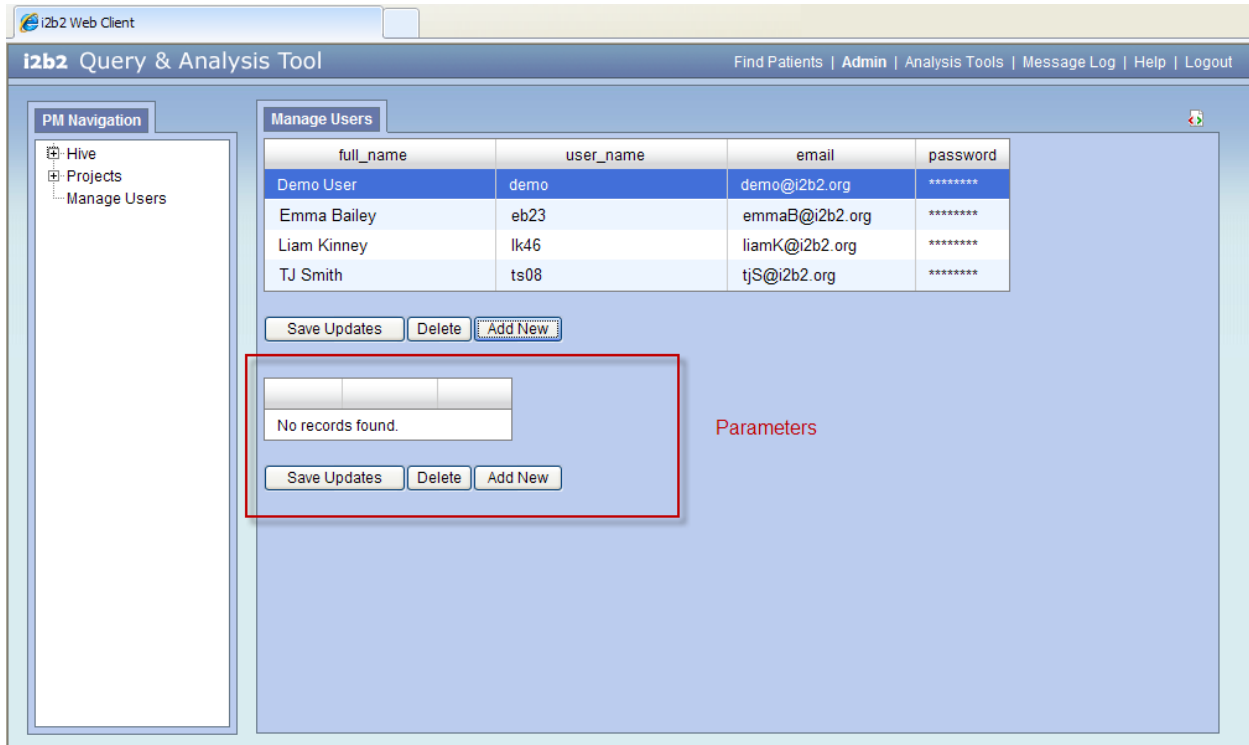
### 3.2.2.1 User Data

- 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*.



### 3.2.2.2 User Parameters

- Parameters entered here are specific to the user and are not specific to any on project.
- The information will be stored in the **PM\_USER\_PARAMS** table.



### 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.

#### **Note**

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

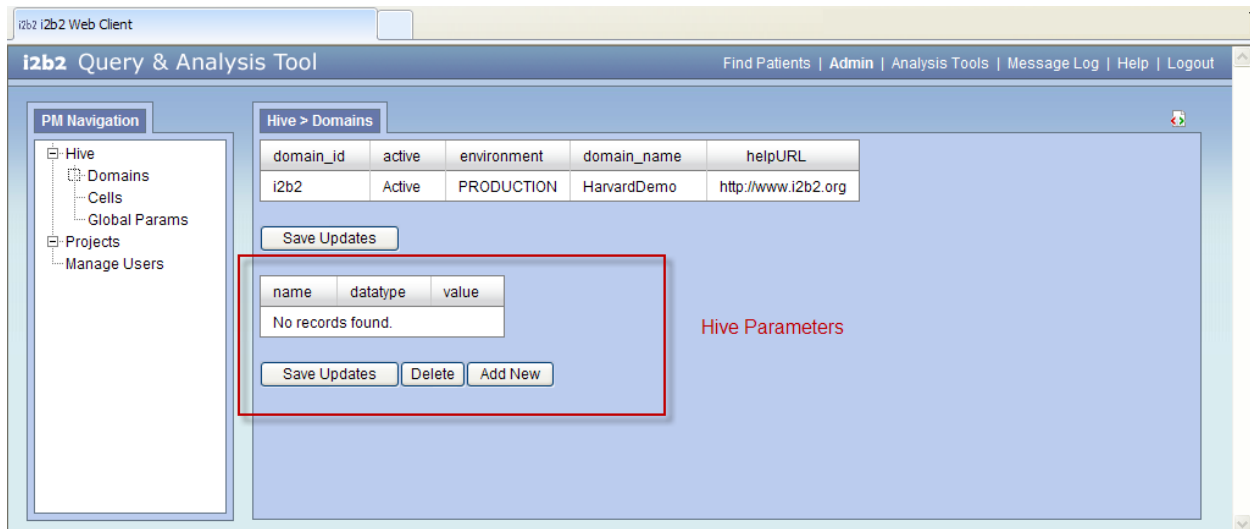
- The domain page captures general information about the hive.

- The information captured includes the *domain id*, *domain name*, *environment*, and *help URL*.
- The information will be stored in the **PM\_HIVE\_DATA** table.

The screenshot displays the 'i2b2 Query & Analysis Tool' interface. At the top, there is a navigation bar with 'Project', 'User: i2b2 Admin', 'Message Log | Help | Logout'. On the left, a 'PM Navigation' sidebar lists: Manage Hive, Domains, Global Params, Manage Cells, Manage Projects, Manage Users, Manage Approvals, and Project Requests. The main area is titled 'Hive Overview' and contains the text 'Please update hive information.' Below this are four input fields: 'Domain Id' (text: i2b2), 'Help URL' (text: http://www.i2b2.org), 'Domain Name' (text: HarvardDemo), and 'Environment' (dropdown menu: Production). At the bottom right of the form are 'Save' and 'Cancel' buttons.

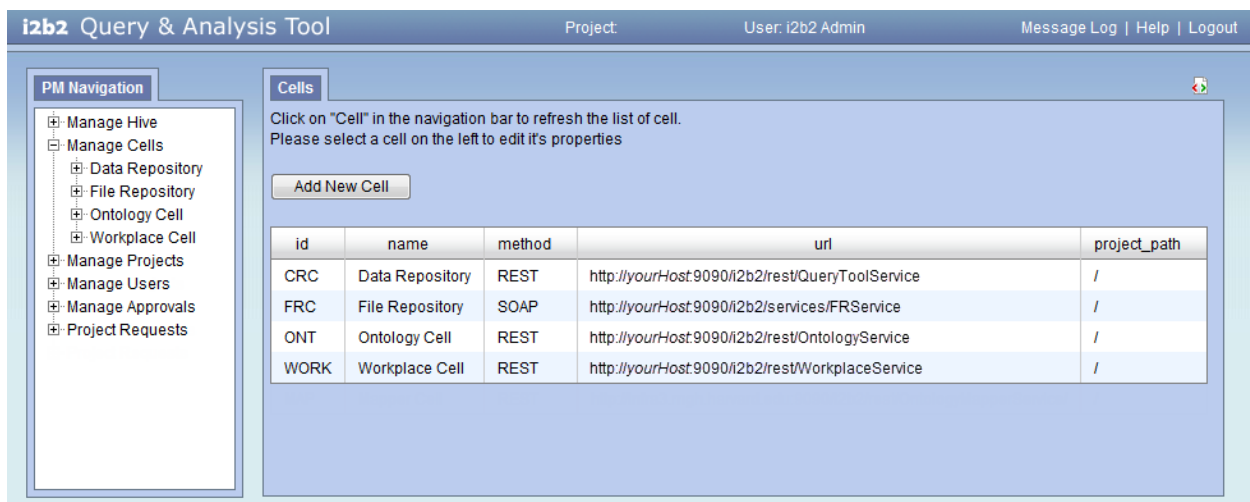
### 3.2.3.2 Hive Parameters

- 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.



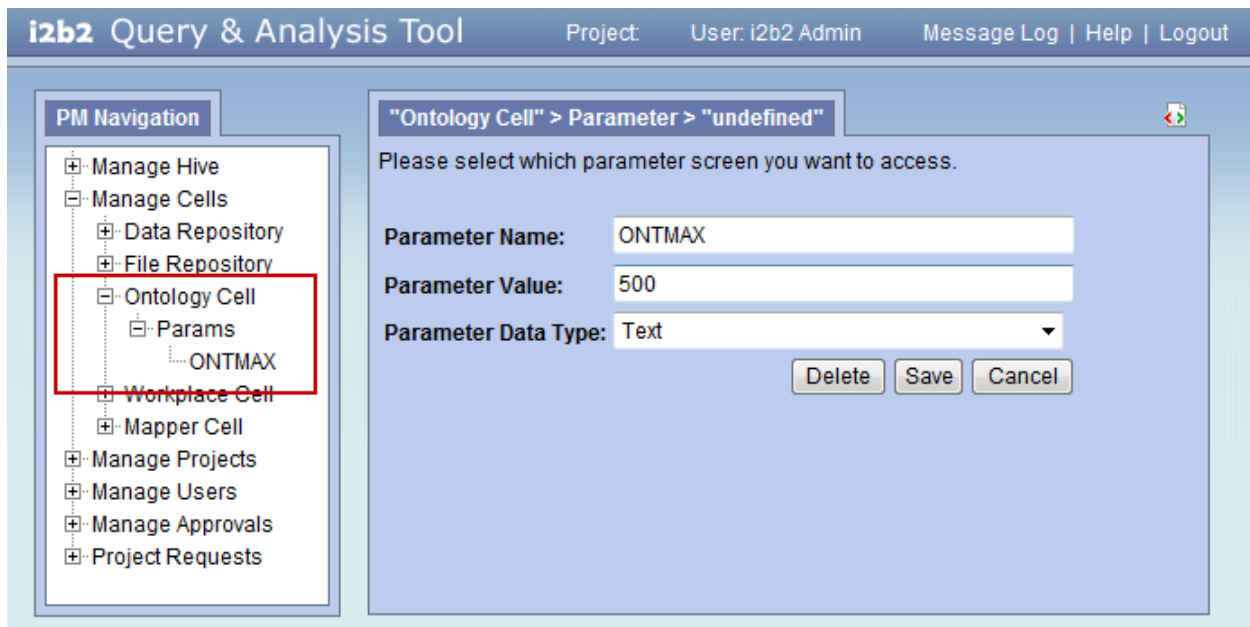
### 3.2.3.3 Cell Data

- 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.



### 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.



### 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.

**i2b2 Query & Analysis Tool** Project: User: i2b2 Admin Message Log | Help | Logout

**PM Navigation**

- Manage Hive
- Manage Cells
- Manage Projects
  - i2b2 Demo
    - Params
    - Users
- Manage Users
- Manage Approvals
- Project Requests

**Project > "undefined"**

Please select which project configuration screen you want to access.

**Project Id:** Demo

**Project Name:** i2b2 Demo

**Project Wiki:** http://www.i2b2.org

**Project Key:**

**Project Description:**

**Project Path:** /Demo

Delete Save Updates Cancel

### 3.2.4.2 Project Parameters

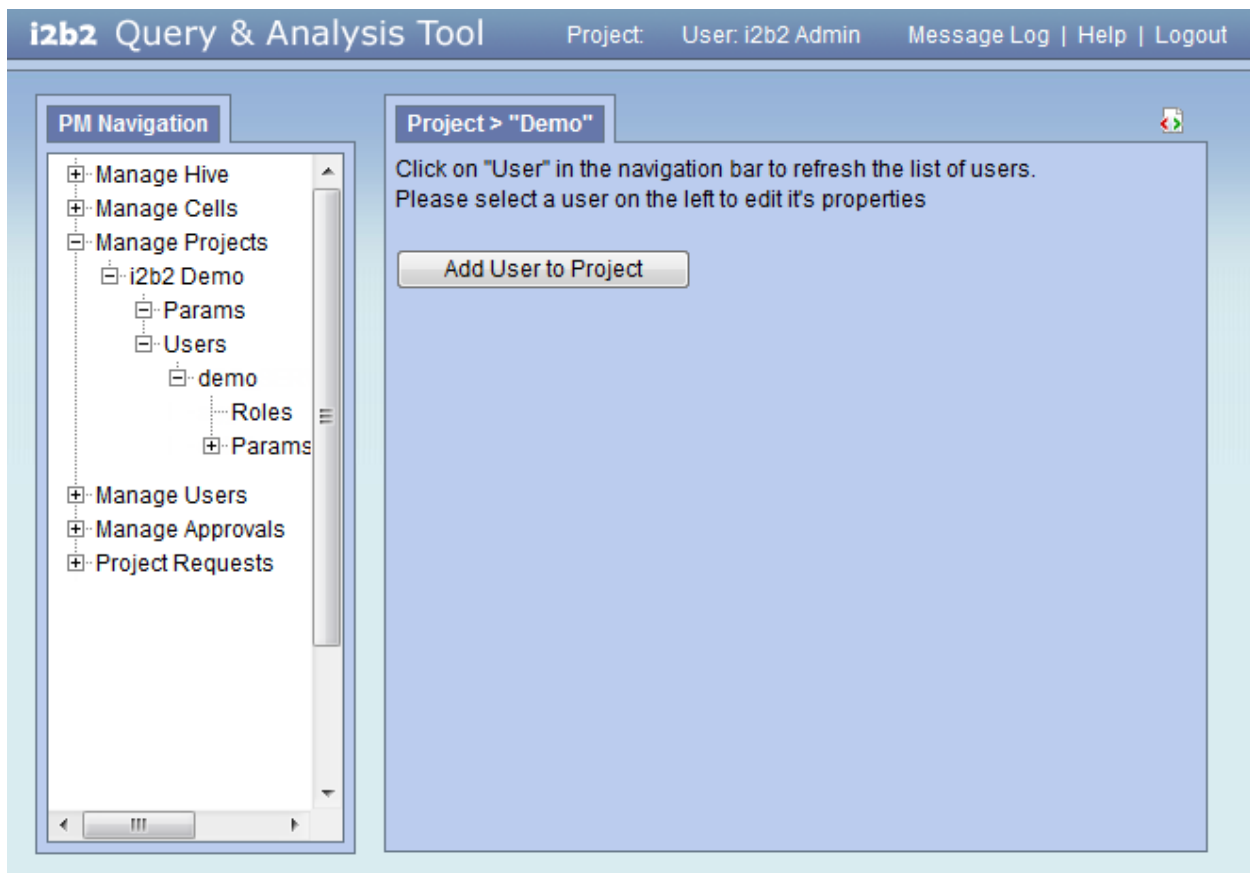
- Parameters entered here are *specific to the project*.
- The project parameters can be added from the **project parameters page**.
- The information will be stored in the **PM\_PROJECT\_PARAMS** table.



The screenshot displays the i2b2 Query & Analysis Tool interface. On the left is a 'PM Navigation' sidebar with a tree view containing the following items: Manage Hive, Manage Cells, Manage Projects (expanded to show i2b2 Demo, Params, and Users), Manage Users, Manage Approvals, and Project Requests. The main area is titled 'Parameter List' and contains a message: 'Click on "Parameter" in the navigation bar to refresh the list of parameters. Please select a parameter on the left to edit it's properties'. Below the message is a button labeled 'Add New Parameter'.

### 3.2.4.3 Project Users

- The **project users' page** captures information regarding which users have access to the project and their role. *(see next section for detailed information regarding roles)*
- The information will be stored in the **PM\_PROJECT\_USER\_ROLES** table.



### 3.2.4.4 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 establishes the detail of data the user can see while the hive management role defines the level of functionality the user has in a project. The following tables summarize the roles in a hierarchical order of least to most access.

| Data Protection Track |   |
|-----------------------|---|
| Role                  | Access Description  |
| DATA_OBFSC            | OBFSC = Obfuscated <ul style="list-style-type: none"> <li>▪ The user can see aggregated results that are obfuscated (example: patient count).</li> <li>▪ 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.</li> </ul> |

|           |   |
|-----------|---|
| DATA_AGG  | <p>AGG = Aggregated</p> <ul style="list-style-type: none"> <li>The user can see aggregated results like the patient count.</li> <li>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.</li> </ul> |
| DATA_LDS  | <p>LDS = Limited Data Set</p> <ul style="list-style-type: none"> <li>The user can see all fields except for those that are encrypted.</li> <li>An example of an encrypted field is the <i>blob fields</i> in the <i>fact</i> and <i>dimension tables</i>.</li> </ul>        |
| DATA_DEID | <p>DEID = De-identified Data</p> <ul style="list-style-type: none"> <li>The user can see all fields including those that are encrypted.</li> <li>An example of an encrypted field is the <i>blob fields</i> in the <i>fact</i> and <i>dimension tables</i>.</li> </ul>      |
| DATA_PROT | <p>PROT = Protected</p> <ul style="list-style-type: none"> <li>The user can see all data, including the identified data that resides in the Identity Management Cell.</li> </ul>  |

| 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 as well as access queries created by different users within the project |

**Note**

Additional roles can be added to the PM\_PROJECT\_USER\_ROLES table but there will not be any recognized hierarchy to those roles.

### 3.2.4.5 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.

PM Navigation

- ⊕ Manage Hive
- ⊕ Manage Cells
- ⊖ Manage Projects
  - ⊖ i2b2 Demo
    - ⊕ Params
    - ⊖ Users
      - ⊖ demo
        - Roles
        - Params
- ⊕ Manage Users
- ⊕ Manage Approvals
- ⊕ Project Requests

Parameter List

Click on "Parameter" in the navigation bar to refresh the list of parameters.  
Please select a parameter on the left to edit it's properties

Add New Parameter

## 4 TABLES

### 4.1 Global Tables

#### 4.1.1 PM\_GLOBAL\_PARAMS Table

- The *PM\_GLOBAL\_PARAMS* table is commonly used in a production system to specify the default values for various users.
- These 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 DATA\_TYPE\_CD. The value (code) entered in this column is used to determine what the object is. The following is a list of data type codes.

| DATATYPE_CD | Description                                  |
|-------------|--|
| T           | Text that is less than 2000 characters       |
| M           | Reference to a Text file (> 2000 characters) |
| C           | Reference to a 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 a RTF File                |
| XLS | Reference to a Microsoft Excel file    |
| XML | Reference to a XML file or blob        |
| DOC | Reference to a Microsoft Word document |

## 4.1.2 PM\_USER\_DATA Table

- 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 the user prefers.
- The **PASSWORD** is the MD5 has for the user's password.
- The **EMAIL** is the users preferred email address.

| PM_USER_DATA |                |                    |
|--------------|----------------|--------------------|
| <b>PK</b>    | <b>USER_ID</b> | <b>varchar(50)</b> |
|              | FULL_NAME      | varchar(255)       |
|              | PASSWORD       | varchar(255)       |
|              | EMAIL          | varchar(255)       |

### 4.1.3 PM\_USER\_PARAMS Table

- 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 the format that the user prefers.
- The **PARAM\_NAME\_CD** is associated with the object in the *CODE\_LOOKUP* table.

| PM_USER_PARAMS |                      |                    |
|----------------|----------------------|--------------------|
| PK             | ID                   | int                |
|                | <b>USER_ID</b>       | <b>varchar(50)</b> |
|                | <b>PARAM_NAME_CD</b> | <b>varchar(50)</b> |
|                | VALUE                | varchar(255)       |
|                | DATATYPE_CD          | varchar(50)        |

#### Note

A “\_PARAMS” table contains **name-value pairs** associated with a user; the parameters can occur more than once or vary from user to user.

A “@” in place of a USER\_ID means that the parameter(s) apply to all users not explicitly listed.

## 4.2 Hive Tables

### 4.2.1 PM\_HIVE\_DATA Table

- 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).

**Note**

The DOMAIN\_ID needs to be unique across multiple enterprises, which is why it should be at least 20 characters in 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 |                       |                     |
|--------------|-----------------------|---------------------|
| <b>PK</b>    | <b>DOMAIN_ID</b>      | <b>varchar(50)</b>  |
|              | <b>ENVIRONMENT_CD</b> | <b>varchar(255)</b> |
|              | <b>DOMAIN_NAME</b>    | <b>varchar(255)</b> |
|              | HELPPURL              | 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 Table

- The *PM\_HIVE\_PARAMS* table is one of several “\_PARAM” tables in the PM cell.
- These parameters are associated with the various **DOMAIN\_IDs** 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                |
|                | <b>DOMAIN_ID</b>     | <b>varchar(50)</b> |
|                | <b>PARAM_NAME_CD</b> | <b>varchar(50)</b> |
|                | VALUE                | varchar(255)       |
|                | DATATYPE_CD          | varchar(50)        |

## 4.2.3 PM\_CELL\_DATA Table

- 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 Ontology (ONT) cells exist with the following PROJECT\_PATHS:

1. One with project “/hive”
2. One with project “/hive/asthma”
3. One with project “/hive/asthma/snm0”

Then only the ONT cell specified by “/hive/asthma/snm0” will be returned for project “snm0”. It essentially allows certain projects to be “diverted” from the mainstream cells.

| PM_CELL_DATA |                     |                     |
|--------------|---------------------|---------------------|
| <b>PK</b>    | <b>CELL_ID</b>      | <b>varchar(50)</b>  |
| <b>PK</b>    | <b>PROJECT_PATH</b> | <b>varchar(255)</b> |
|              | NAME                | varchar(255)        |
|              | URL                 | varchar(255)        |
|              | METHOD_CD           | varchar(255)        |
|              | CAN_OVERRIDE        | int                 |

#### 4.2.4 PM\_CELL\_PARAMS Table

- 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 Table

- 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 the 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 both 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 following 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.
- It will be used in queries to find the value for both cell and global parameters.
- The hive is initialized with default values for all parameters with a blank / null project, following which the parameters may be overridden by adding a project.
- When the query returns, only one value for each parameter 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 Table

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

| PM_PROJECT_PARAMS |                      |                    |
|-------------------|----------------------|--------------------|
| PK                | ID                   | int                |
|                   | <b>PROJECT_ID</b>    | <b>varchar(50)</b> |
|                   | <b>PARAM_NAME_CD</b> | <b>varchar(50)</b> |
|                   | VALUE                | varchar(255)       |
|                   | DATATYPE_CD          | varchar(50)        |

### 4.3.3 PM\_PROJECT\_USER\_ROLES Table

- 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.
- The “Hive Management Track” role establishes the amount of control a user has in a project.
- The “Data Protection Track” role establishes the detail of data that may be seen by the user.
- The “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 not be any recognized hierarchy to those roles.
- For roles that span across all projects, the **PROJECT\_ID** column in the *PM\_PROJECT\_USER\_ROLES* table will have an “@” sign.
- For roles that span across all users (for a project), the **USER\_ID** column in the *PM\_PROJECT\_USER\_ROLES* table will have an “@” sign.
- At a minimum, a user / project combination will have a role from the Data Protection Track **and** the Hive Management Track (Custom Track is optional).

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

| PM_PROJECT_USER_ROLES |                     |                     |
|-----------------------|---------------------|---------------------|
| <b>PK</b>             | <b>PROJECT_ID</b>   | <b>varchar(50)</b>  |
| <b>PK</b>             | <b>USER_ID</b>      | <b>varchar(50)</b>  |
| <b>PK</b>             | <b>USER_ROLE_CD</b> | <b>varchar(255)</b> |
|                       |                     |                     |

#### 4.3.4 PM\_PROJECT\_USER\_PARAMS Table

- User parameter(s) that is specific to the project.

| PM_PROJECT_USER_PARAMS |                      |                    |
|------------------------|----------------------|--------------------|
| <b>PK</b>              | <b>ID</b>            | <b>int</b>         |
|                        | <b>PROJECT_ID</b>    | <b>varchar(50)</b> |
|                        | <b>PARAM_NAME_CD</b> | <b>varchar(50)</b> |
|                        | VALUE                | varchar(255)       |
|                        | DATATYPE_CD          | varchar(50)        |

## 4.4 Access / Restriction Tables

### 4.4.1 PM\_ROLE\_REQUIREMENT Table

- In order to assign the permissions to edit the PM tables, entries are made into the *PM\_ROLE\_REQUIREMENT* table as shown 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.
- An @ in the *TABLE\_CD* or the *COLUMN\_CD* columns for a specific code 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 as a description for the purpose and function of a column (varchar datatype for lookup / searching capabilities).

| PM_ROLE_REQUIREMENT |                          |                     |
|---------------------|--------------------------|---------------------|
| <b>PK</b>           | <b>TABLE_CD</b>          | <b>varchar(50)</b>  |
| <b>PK</b>           | <b>COLUMN_CD</b>         | <b>varchar(50)</b>  |
| <b>PK</b>           | <b>READ_HIVEMGMT_CD</b>  | <b>varchar(50)</b>  |
| <b>PK</b>           | <b>WRITE_HIVEMGMT_CD</b> | <b>varchar(255)</b> |
|                     | NAME_CHAR                | varchar(2000)       |

## 4.5 Audit Tables

Each table in the project management database has limited auditing of its own management through the change management columns. These columns record the following:

- The date-time of the initial entry
- The date-time of the latest change
- 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      |



**Note**

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 Table

- 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.
- An @ in the *TABLE\_CD* or the *COLUMN\_CD* columns for a specific code 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 |                  |                    |
|----------------|------------------|--------------------|
| <b>PK</b>      | <b>TABLE_CD</b>  | <b>varchar(50)</b> |
| <b>PK</b>      | <b>COLUMN_CD</b> | <b>varchar(50)</b> |
| <b>PK</b>      | <b>CODE_CD</b>   | <b>varchar(50)</b> |
|                | NAME_CHAR        | varchar(2000)      |
|                | LOOKUP_KEYS_CHAR | varchar(2000)      |