Data Mart Guide

The OpenClinica Data Mart generates SQL designed to provide easy and flexible access to your study data in the form of a well structured PostgreSQL database. The Data Mart is generated via an XSLT (more info on working with OpenClinica XSLTs) that runs on the CDISC ODM XML that OpenClinica natively produces. You can set-up, run, and access the Data Mart SQL in OpenClinica's Extract Data module.

Note: Datamart currently does not support non-ASCII characters in CRF name, Response Text, Multi-Select values, and Checkbox values.

Key Features of Data Mart

The OpenClinica Data Mart provides the following key features:

- The tool makes each CRF or group of repeating Items accessible as a separate table.
- The tool combines all versions of a CRF into one table. Fields that are not relevant to each specific record appear as null.
- The tool shows records from Groups with one row per entry, and a connection to the parent CRF.
- The tool stores date Items as the type DATE in the database.
- The user can select the Study, Events, and CRFs to view.
- The order of columns in the data table for a CRF matches the order of Items in the original CRF, with Items from new CRF versions appearing at the end.
- The tool provides continued reliable access to the OpenClinica data even if the OpenClinica database schema changes in the future.
- Single-select or radio button Items can be viewed in text or integer form.
- Each multi-select or checkbox Item appears as a set of Boolean columns, with one column per option.

1 Accessing Data Mart on OpenClinica Optimized Hosting

If you are an Optimized Hosting customer, we will set up the OpenClinica Data Mart for you. In this scenario we use an IP address/network with a single username and password to enable access your Data Mart database(s). The IP address/network is provided by the customer in your Configuration Worksheet. Once enabled you will be able to access the Data Mart via the following steps.

1. First, we will provide you with the following information:

Instance Number:
Password:

Your username will be "datamart_INSTANCE" where INSTANCE is your Instance Number we provided you.

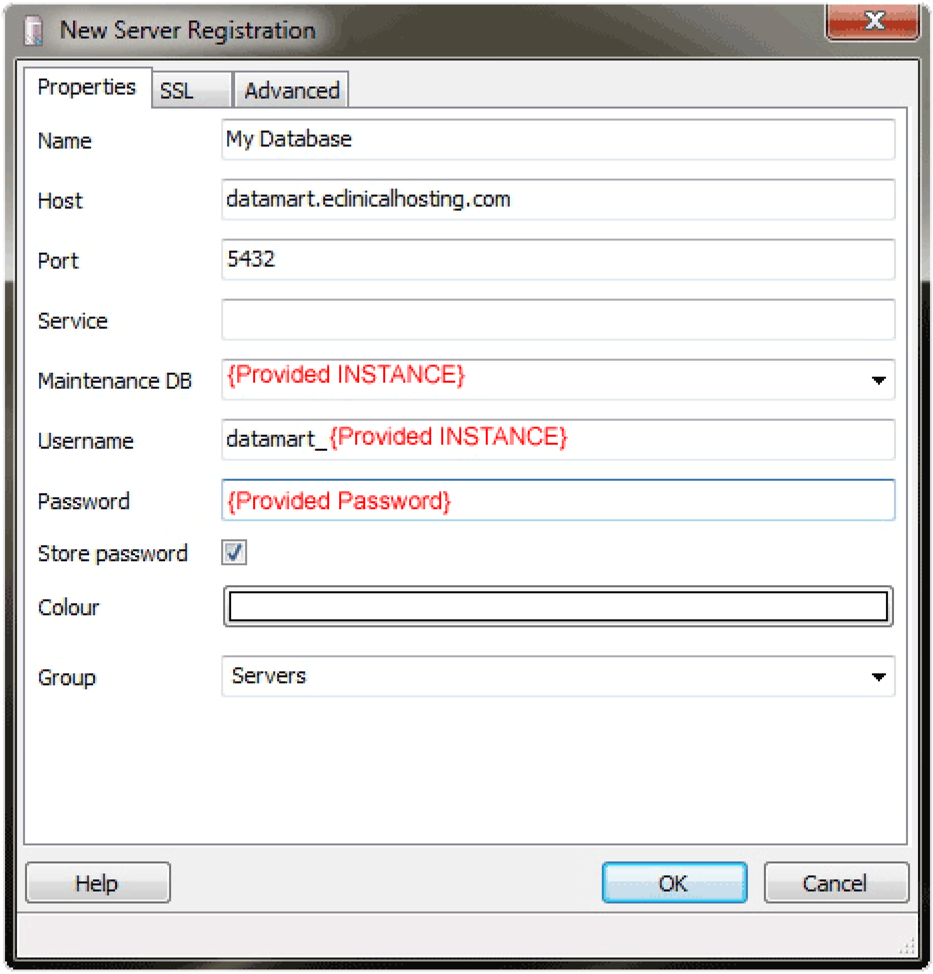
Your two databases will be named "INSTANCE" for your production Data Mart database name and "INSTANCE_test" for your test Data Mart database name.

2. Open pgAdmin

Once opened, click File --> Add Server.

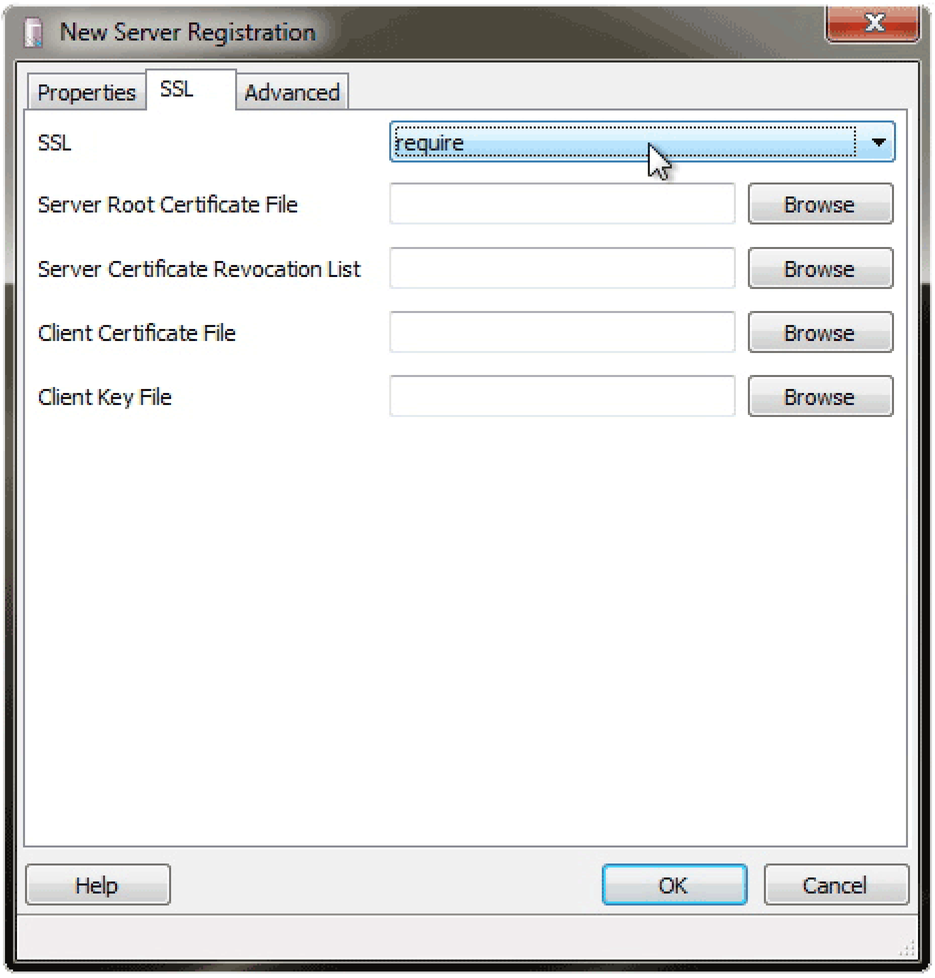
In the New Server Registration page configure the settings as follows:

Name: (This is a name for your organizational purposes and can be named whatever you would like.)
Host: datamart.eclinicalhosting.com
Port: 5432
Maintenance DB: INSTANCE
Username: datamart_INSTANCE Please note these INSTANCE names needs to be lower case when you enter.
Password: Provided password



Name: My Database
Host: datamart.eclinicalhosting.com
Port: 5432
Service:
Maintenance DB: {Provided INSTANCE}
Username: datamart_{Provided INSTANCE}
Password: {Provided Password}
Store password: ☑
Colour:
Group: Servers

SSL: require



SSL: require

Server Root Certificate File

Server Certificate Revocation List

Client Certificate File

Client Key File

Once done, click on "OK" to save the definition. Now, to access it double click on the "Name" you provided in the list with a red "X" next to it.

Note: The "Maintenance DB" cannot be set to "postgres" and must be set to one of your DB's (either "INSTANCE" or "INSTANCE_test").

You will be able to see numerous Data Mart databases listed on that server. However, you will be able to access your databases only.

This page is not approved for publication.

2 Accessing the Data Mart Local Deployment

The OpenClinica Enterprise Edition is configured to have the Data Mart SQL auto-loaded into a PostgreSQL database. You always have the ability to download the .sql file and load it into any database you would like.

By default, each time you run a particular Data Mart extract, the data in your database will be replaced.

There are two options for setting up your Data Mart (the option you have chosen is identified in your Configuration Worksheet):

- **Option 1:** We setup the Data Mart users and database on your current PostgreSQL server running OpenClinica and enable access to the Data Mart database from a designated IP address. We provide you with the username, password, hostname, and database names that will allow you to connect to the Data Mart. Since under this scenario your Data Mart is on a sever we manage, we limit access to the Data Mart to a single user account and IP in order to maintain a high level of data security and accountability. If you require more elaborate access, please select option 2.
- **Option 2:** You provide a PostgreSQL server, and we configure OpenClinica to load the Data Mart data into it. Since you provide the PostgreSQL server, you would need to supply us with the IP of the PostgreSQL server, the username and password as well as the database name for each OpenClinica instance that you would like the Data Mart to utilize.

Option 2 provides the greatest flexibility for accessing Data Mart. If you select option 1, you may wish to set-up a remote access solution in order to enable multiple people to access the Data Mart database. If you require separate usernames for your users you will have to go with option 2.

Tips for Working With Data Mart

- If you do not want your data to be overwritten on every extract (which is the default setting), you may download the .sql file (either manually or via a scheduled job) and create a script that creates a database called datamart-TIME (where time is current date-time, or any other variable chosen as a convention) and loads the sql file into that db. This way, each time this script is run, a new database would be created and your prior database would be preserved.
- Under option 1, if you want multiple users to access Data Mart from a single IP, you can setup a terminal server with pgAdmin and any other tools you may want to use to connect to the database. Then, provide us with the IP address of this server so we can authorize in the PostgreSQL database server. Now, as many staff as you allow can access the Data Mart through the terminal server.
- Another method for allowing multi-user access under option 1 is create a backup/copy the Data Mart database using the single account we provide, then restore this backup onto a PostgreSQL server that you manage elsewhere.

This page is not approved for publication.

3 How We Set Up Your Data Mart

When we configure your OpenClincia Data Mart, we make the following changes to your OpenClinica instance.

(Note: All changes are done in the extract.properties file. The extract.properties file is located at %tomcat_home%/OpenClinica/WEB-INF/classes/extract.properties.)

The following items are added under the "extract.10" set in the file:

```
extract.11.odmType=clinical_data
extract.11.file=ODMReportStylesheet.xsl
extract.11.fileDescription=Datamart
extract.11.linkText=Run Now
extract.11.helpText=Generate the Datamart
extract.11.location=$exportFilePath/$datasetName/DATAMART2
extract.11.exportname=sql_$datasetName$date.sql
extract.11.post=db1
extract.11.zip=true
extract.11.deleteOld=true
extract.11.success=Your extract job completed successfully. The file is available for download $linkURL.
extract.11.failure=The extract did not complete due to errors. Please contact your system administrator for details.
```

Additionally, we configure the following (based on your settings) in the "SQL Postprocessor Configurations" section of the extract.properties file.

```
db1.username=datamart
db1.password=datamart
db1.url=jdbc:postgresql://localhost:5432/datamart
db1.dataBase=postgres
```

This page is not approved for publication.

4 Setting Up a PostgreSQL Server for Data Mart

Instructions

This guide covers how you can setup PostgreSQL for use with your OpenClinica Data Mart. This is necessary if you would like to send the Data Mart data to an external database (Option 2 in the Customer Configuration Worksheet).

Setting up your PostgreSQL database involves the following steps:

1. Install a PostgreSQL server.
2. Create a role in PostgreSQL that OpenClinica can send the Data Mart with.
3. Create a DB for the Data Mart data to be loaded into.
4. Enable access in PostgreSQL so that you OpenClinica application server can communicate with PostgreSQL.

Step 1: Install a PostgreSQL Server.

If your server is running Linux please follow the instructions on our installation guide for Linux section "Postgres Install" located here.

If your server is running Windows please follow the instructions on our installation guide for Windows section "Install PostgreSQL" located here.

Step 2: Create a role.

If your server for PostgreSQL is running Linux, run the following command to create a role. This command will create a role with the username of "datamart" and a password of "datamart."

```
/opt/PostgreSQL/8.4/bin/psql -U postgres -c "CREATE ROLE datamart LOGIN ENCRYPTED
PASSWORD 'datamart' NOINHERIT NOCREATEDB NOCREATEROLE;"
```

If your server for PostgreSQL is running Windows please open up PGAdminIII (which is installed when you install PostgreSQL). In the PGAdminIII interface execute the following query on the "postgres" database. This will create a role with the username of "datamart" and a password of "datamart."

```
CREATE ROLE datamart LOGIN ENCRYPTED PASSWORD 'datamart' NOINHERIT NOCREATEDB
NOCREATEROLE;
```

Step 3: Create a DB.

If your server for PostgreSQL is running Linux, run the following command to create a DB. This command will create a DB with the name of "datamart" and owned by the role "datamart."

```
/opt/PostgreSQL/8.4/bin/psql -U postgres -c "CREATE DATABASE  datamart WITH
ENCODING='UTF8' OWNER=datamart;"
```

If your server for PostgreSQL is running Windows, please open up PGAdminIII. In the PGAdminIII interface execute the following query on the "postgres" DB. This will create a DB with the name of "datamart" and owned by the role "datamart."

```
CREATE DATABASE  datamart WITH ENCODING='UTF8' OWNER=datamart;
```

Step 4: Allow access to the DB from your application server.

By default, PostgreSQL will not allow the connection. Please follow this guide to enable access for the application server.

If you followed the instructions in steps 1-4, the entry for pg_hba.conf should look like the following. Replace $IP with the IP of your application server.

```
host datamart datamart $IP/32 md5
```

If more then one Data Mart DB is needed:

Most customers will have more then one Data Mart DB since they have more then one OpenClinica instance with Data Mart activated on each instance. In this scenario you have two options for setting up the additional Data Mart databases.

1. Use a separate DB server for each Data Mart DB
    - For this scenario you can re-run steps 1-4 on the other server.
2. Use the same DB server for all Data Mart DBs
    - For this scenario, re-run step 3 "Create a DB" while replacing the DB name with a unique name. e.g.:" datamart_test"
    - Additionally please complete step 4 "Allow access to the DB from your application server" for this new DB name. For example, if you used "datamart_test" as your new DB name an example entry for pg_hba.conf will look like the following:

```
host datamart_test datamart $IP/32 md5
```

What your OpenClinica Enterprise support team needs to configure this Data Mart option:

As described in you Customer Configuration Worksheet we will need the following information to complete the configuration of your Data Mart DB(s).

- Server IPHostname
  - The IPHostname of the PostgreSQL DB server you configured.
- Database Name
  - Using the examples provided in this guide this would be "datamart" and "datamart_test" for a two instance installation.
- Username
  - Using the examples provided in this guide the username would be "datamart" for both Data Mart DB(s)
- Password
  - Using the examples provided in this guide the password would be "datamart". *This should be changed for security purposes.*

The following is a depiction of the relevant section in teh Customer Configuration Worksheet. Based on this guide you would provide us the following information for your two instances. Replace $IP with the IPHostname of your database server.

Production Instance		**Test Instance**	
Server IPHostname:	$IP	Server IPHostname:	$IP
Database Name:	datamart	Database Name:	datamart_test
Username:	datamart	Username:	datamart
Password:	datamart	Password:	datamart

This page is not approved for publication.

5 Working with the Data Mart

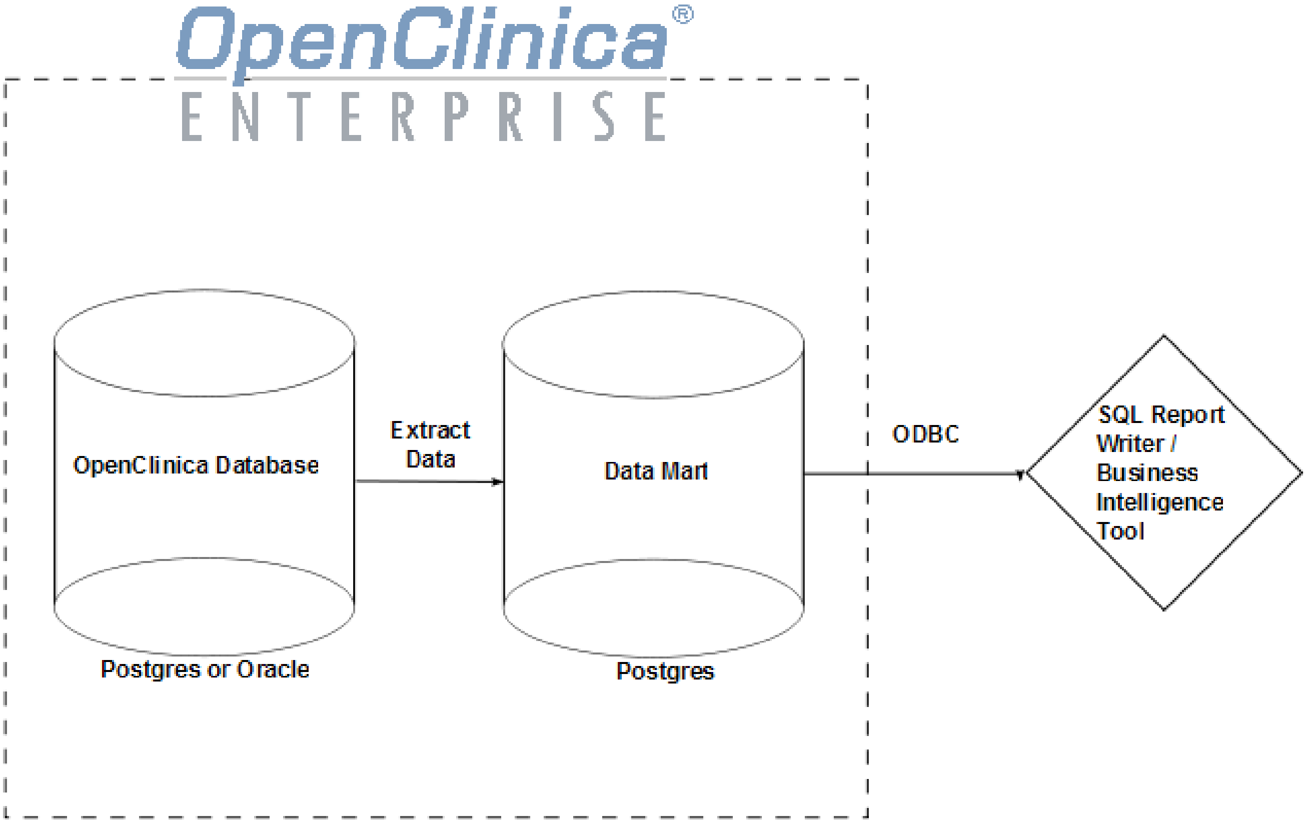
This page is not approved for publication.

5.1 System Architecture for Data Mart

The OpenClinica Data Mart takes data that is extracted from OpenClinica, converts it to SQL, and inserts the resulting data into a database for reporting purposes. The data flows in the following manner:

1. OpenClinica exports data in the CDISC ODM 1.3 with OpenClinica Extensions format.
2. OpenClinica converts the data via an XML style sheet to SQL, and loads the data into the Data Mart.
3. The data user accesses the data from the data warehouse using a business intelligence or SQL reporting tool. If these tools are based on a client computer, access to the data warehouse can be via Open Database Connectivity (ODBC).

Data Mart System Diagram:



With the Postgres client pgAdmin, you can make a direct connection to the Postgres database containing the Data Mart. An alternative solution is to set up an ODBC connection to the database, for use with desktop business intelligence or SQL report writer tools.

If your organization has selected the Postgres database output option for the Data Mart, the system administrator from the OpenClinica, LLC client services team provides your system administrator with the information to set up the ODBC connections and the direct database connections from pgAdmin. Please see your system administrator for access to the database.

This page is not approved for publication.

5.2 Loading Data into the Data Mart

For instructions, see Define Dataset in the OpenClinica User Documentation. Specific notes applicable to the Data Mart are:

- Select all Items in the Study.
- Select all Attributes.
- The dataset name will become part of the name of the schema in the data warehouse.

To perform a single, manual export, see instructions at Generate and Download Dataset in the OpenClinica User Documentation. Select the Data Mart format.

To set up a recurring scheduled job, see instructions at Scheduled Export Jobs in the OpenClinica User Documentation. Select the Data Mart format.

This page is not approved for publication.

5.3 Data Mart Schema for Study

Each OpenClinica dataset extracted from an OpenClinica Study creates its own database schema within the Data Mart. The schema name consists of the Study Protocol Name, plus the name of the OpenClinica dataset.

IMPORTANT:

Potential Conflicts with Table and Column Name Maximum Lengths

PostgreSQL has a maximum length of 63 characters for the names of columns and tables. The Datamart generates table and column names based a composite of the CRF, group, item, and (for multi-selects/checkboxes) the item's response options text. If you have long names for these objects within your CRF, Data Mart may have problems not creating the required tables/columns. For instance, before adding a column to a table, Postgres checks the column name length for validity. If the column name is too long, Postgres will truncate the proposed column name, and attempt to create the column using the truncated name. The problem is that if the truncated column name is found to already exist, instead of skipping adding the column gracefully, the SQL script may error out.

Essentially the combination of the CRF, group, item names and Response Options text can cause a column name to be long enough to be truncated. If two different columns get truncated to the same name the Datamart will not work as expected. We suggest using names that are a short as possible to avoid this conflict.

In OpenClinica 3.1.4 we have made some changes to the way that Data Mart handles very long column names for checkbox/multi select options this change only affects checkbox and multi-select options. To prevent the long column names in the SQL scripts from causing errors, we have added logic when building the column names which does some truncation if the column name would exceed 63 characters. The column names are built by concatenating the Item Name and the Response Options Text. If the Item Name exceeds 30 characters, then it will be truncated and a unique number will be added to it. If the Response Options Text exceeds 30 characters, then it will be truncated and a unique number will be added to it. For any columns that have been truncated, they will be added to a new mapping table. This mapping table has been built to make it easy to reference any items that have been truncated. The mapping table contains the Column name (as truncated), the original Item Name and original Response Options Text.

Additionally in 3.1.4, CRF table names will now be truncated to prevent long names from causing Data Mart errors. Currently table names can be comprised of the following elements concatenated together:

- CRF Name
- Item Group Name
- '_resp_opts'

CRF & Item Group Name concatenations exceeding 52 characters will be truncated with a unique number appended to prevent duplicate names. This limit will leave room in the table name for '_resp_opts', in case a separate table for multiselect items needs to be created.

From the PostgreSQL Manual Section 4.1.1, Identifiers and Key Words:

"SQL identifiers and key words must begin with a letter (a-z, but also letters with diacritical marks and non-Latin letters) or an underscore (_). Subsequent characters in an identifier or key word can be letters, underscores, digits (0-9), or dollar signs ($). Note that dollar signs are not allowed in identifiers according to the letter of the SQL standard, so their use may render applications less portable. The SQL standard will not define a key word that contains digits or starts or ends with an underscore, so identifiers of this form are safe against possible conflict with future extensions of the standard.

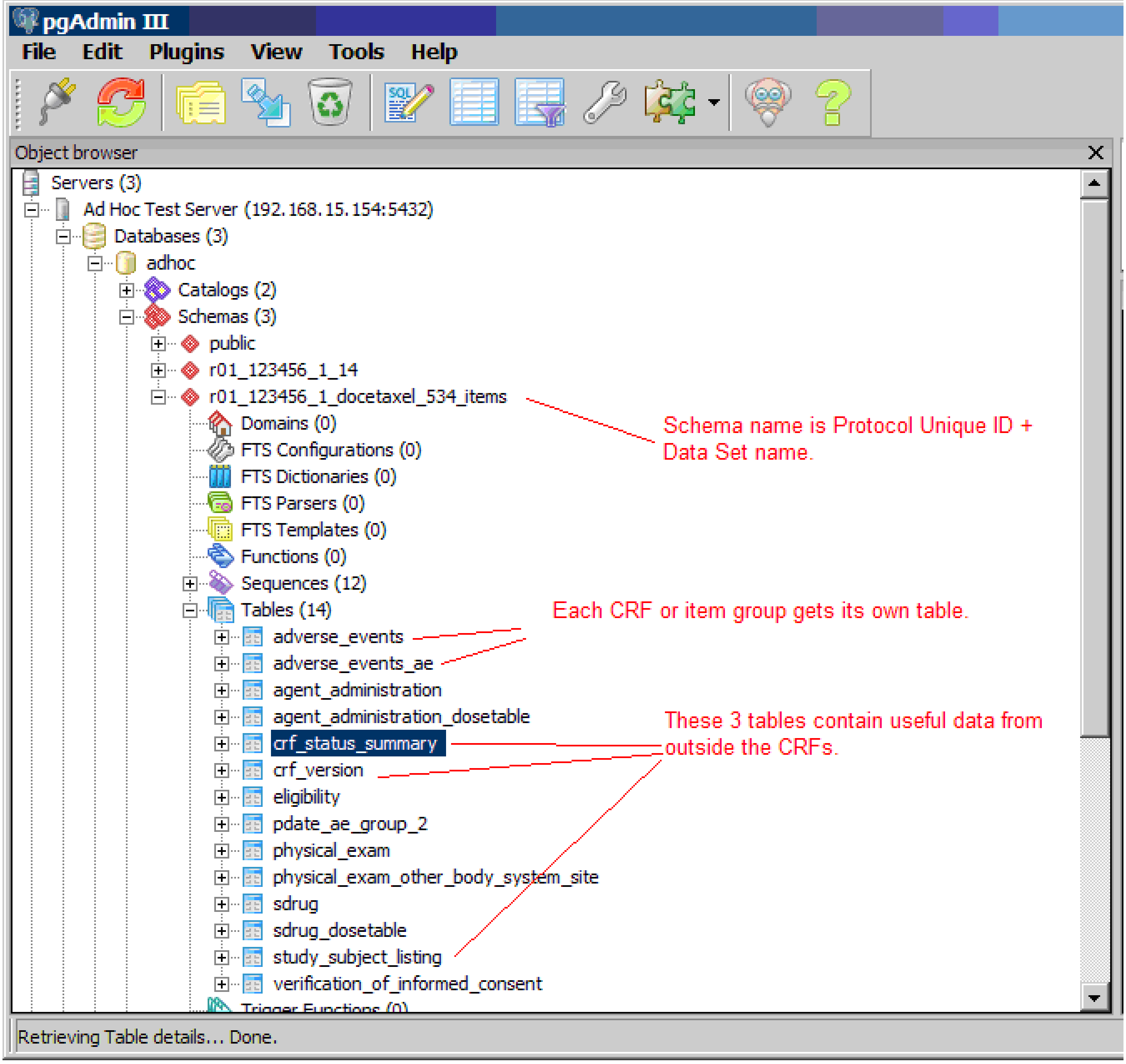
The system uses no more than NAMEDATALEN-1 characters of an identifier; longer names can be written in commands, but they will be truncated. By default, NAMEDATALEN is 64 so the maximum identifier length is 63. If this limit is problematic, it can be raised by changing the NAMEDATALEN constant in src/include/postgres_ext.h."

Reserved Words

PostgreSQL reserves certain words for creating objects within its database schema, which means that it will restrict you from creating tables named with these reserved words. If you have objects within a CRF (such as item_name, item_group) with these reserved words, Data Mart will not create the required tables/columns. Therefore, we advise that you refrain from using these words as your identifiers of an item during the CRF design. If you wish to use these reserved words for your items in the CRF, you may need to add additional characters in order to differentate it from a reserve word (e.g. name the data item INTEGER_123 instead of INTEGER). For a complete list of PostgreSQL reserved words, please refer to the following link:

http://www.postgresql.org/docs/8.4/static/sql-keywords-appendix.html

Data Mart Study Schema and CRF Tables (Example from pgAdmin):



- Servers (3)
  - Ad Hoc Test Server (192.168.15.154:5432)
    - Databases (3)
      - adhoc
        - Catalogs (2)
        - Schemas (3)
          - public
          - r01_123456_1_14
          - r01_123456_1_docetaxel_534_items
            - Domains (0)
            - FTS Configurations (0)
            - FTS Dictionaries (0)
            - FTS Parsers (0)
            - FTS Templates (0)
            - Functions (0)
            - Sequences (12)
            - Tables (14)
              - adverse_events
              - adverse_events_ae
              - agent_administration
              - agent_administration_dosetable
              - crf_status_summary
              - crf_version
              - eligibility
              - pdate_ae_group_2
              - physical_exam
              - physical_exam_other_body_system_site
              - sdrug
              - sdrug_dosetable
              - study_subject_listing
              - verification_of_informed_consent
            - Trigger Functions (0)

Schema name is Protocol Unique ID + Data Set name.

Each CRF or item group gets its own table.

These 3 tables contain useful data from outside the CRFs.

Retrieving Table details... Done.

To grant access to specific users for a subset of Study data (for example, only Subjects from a certain Site, or only certain CRFs and data Items) create a special dataset, and create a user Role in the database with access to only that specific schema.

When extracting data in Datamart format, Datamart creates one table for every non-grouped CRF and an additional table for grouped CRF. The maximum non-grouped CRF items count limit is 125. If non-grouped items count in the CRF exceeds the defined maximum limit; single-select labels and multi-select Booleans will be stored in another table.

This page is not approved for publication.

5.4 Data Types

- **Date**: OpenClinica dates are presented as dates in the Data Mart. This makes it easy to perform calculations and to transform the output format.

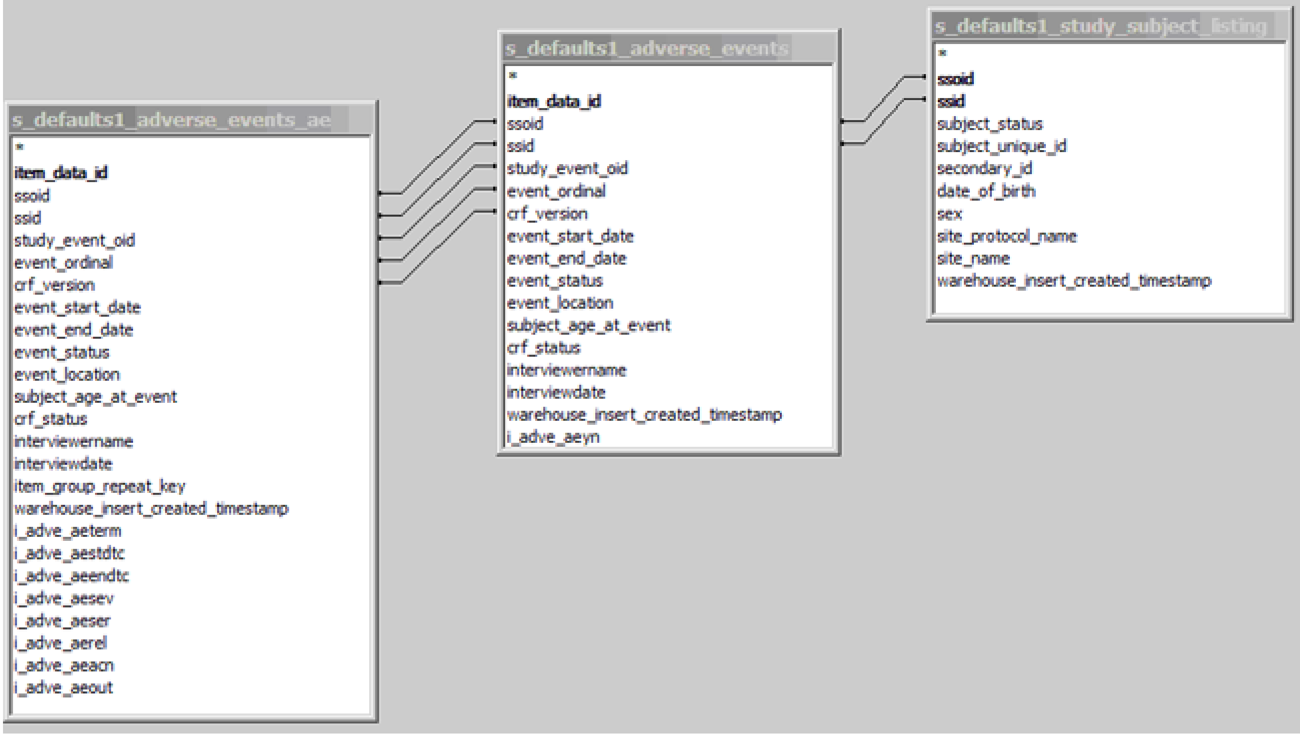
- **Single-Select or Radio Button**: Single-select Items allow selection of a single value from a list, using either a radio button or pulldown control. The possible values are stored as a list of underlying data values (normally integers) and corresponding labels. The Data Mart presents each single-select Item as two columns in the database. The first column stores the value, and the second column stores the label.

- **Multi-select or Checkbox**: For example, Race - Check all that apply. Multi-select or checkbox Items will appear as one Boolean column per possible selection. The column name will reflect the label, for example, i_dm00_race_asian.

This page is not approved for publication.

5.5 SQL Joins for Related Tables

CRF Table Relationships (Graphical Example from MS Access):



Each row in each CRF table and grouped Item table relates to a particular Study Subject, uniquely identified by Study Subject ID and Study Subject OID. Each CRF table in the Data Mart can be joined to the study_subject_listing table on ssid and ssoid.

Each Item in a group of repeating Items is related to the CRF by the Study Subject ID, Study Subject OID, Study Event OID, Event Ordinal, and CRF Version.

The WHERE clause in the following SELECT statement reflects these joins. When querying data from the database, use WHERE clauses of this form.

```
SELECT
 ssl.ssid,
 ae_ungr.event_start_date,
 ssl.date_of_birth,
 ae_ungr.i_adve_aeyn_label,
 ae_grouped.i_adve_aeterm,
 ae_grouped.i_adve_aesev
FROM
 r01_123456_1_docetaxel_534_items.study_subject_listing ssl,
 r01_123456_1_docetaxel_534_items.adverse_events ae_ungr,
 r01_123456_1_docetaxel_534_items.adverse_events_ae ae_grouped
WHERE
 ssl.ssoid =  ae_ungr.ssoid
 and
 ssl.ssid =  ae_ungr.ssid
 and
 ae_ungr.ssoid = ae_grouped.ssoid
 and
 ae_ungr.ssid = ae_grouped.ssid
 and
 ae_ungr.study_event_oid = ae_grouped.study_event_oid
 and
 ae_ungr.event_ordinal = ae_grouped.event_ordinal
 and
 ae_ungr.crf_version = ae_grouped.crf_version;
```

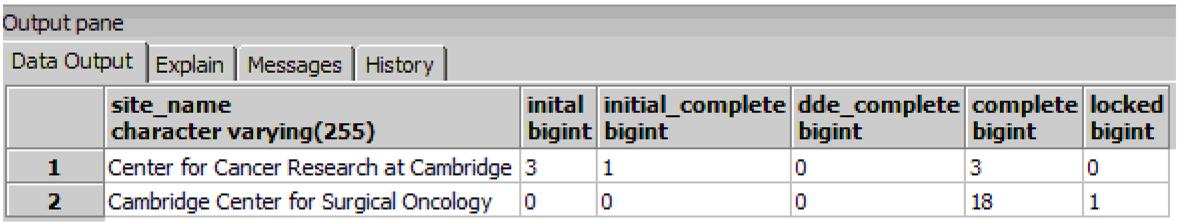
This page is not approved for publication.

5.6 Reporting Status of CRFs and Events

Each schema contains a table listing the status of each CRF and Event (crf_status_summary). Use SQL queries to create valuable reports from this data. For example, this SQL query will show the number of CRFs with each status, by Site:

```
SELECT
 site_name,
 sum (crf_status_initial_data_entry)as inital,
 sum (crf_status_initial_data_entry_complete) as initial_complete,
 sum (crf_status_double_data_entry) as dde_complete,
 sum (crf_status_data_entry_complete) as complete,
 sum (crf_status_locked) as locked
FROM
  r01_123456_1_docetaxel_534_items.crf_status_summary
GROUP BY site_name;
```

CRF Status by Site (pgAdmin Output Example):



	site_name character varying(255)	inital bigint	initial_complete bigint	dde_complete bigint	complete bigint	locked bigint
1	Center for Cancer Research at Cambridge	3	1	0	3	0
2	Cambridge Center for Surgical Oncology	0	0	0	18	1

This page is not approved for publication.