The Panoptes data structure

1 Overview

1.1 Terminology

Dataset

A complete set of data that can be loaded and visualized in a single Panoptes session. A dataset can consist of:

- Several **data tables**.
- A **reference genome**.
- Several **summary values**.

Workspace

Each **dataset** has one or more workspaces associated. The user always opens a specific workspace, and can add custom information to the **dataset** that is only visible in the context of this workspace.

Data table

A data table is a table of records that can be queries and visualized in Panoptes, corresponding to a type of information. Typical examples are samples and genomic variants.
A record in a data table is called a **data item**. Each data table has a number of columns called **properties**.

Data item

A data item is a record in a **data table**.

Property

A property is a column in a data table. As such, it defines a property of a **data item**. There are two types of properties:

- Standard property: provided in the dataset.
- Custom property: added by a user in the context of a **workspace**.

Reference genome

A **dataset** can have information that related to a reference genome, such a genomic variants.

Summary value

A summary value is a filterbanked property defined over a **reference genome**. There are three types:

- Standard summary value: provided by the **dataset**.
- Custom summary value: added by a user in the context of a **workspace**.
- Data table-related summary value: a type of summary value that has an instance for each **data item** in a **data table**. Example: coverage info for a table of samples with sequenced genomes.

2 Database datasetindex

This database contains an overview of all the datasets served by the Panoptes instance, and some tables for general usage.

2.1 Table "*datasetindex*"

Each record describes a dataset served by the Panoptes instance that reads from this database.

Table columns

- *id*, varchar(20). Unique identifier for the data set.
- *name*, varchar(50). Display name of the data set.

2.2 Table "*calculations*"

Each record corresponds to a calculation that was done or is being done on the server. The state of the calculation (in progress, completed, failed) is also stored in the record.

Table columns

- *id*, varchar(50). Unique identifier of the calculation.
- *user*, varchar(50). User id of the person who started the calculation.
- *timestamp*, varchar(50). Time stamp of the start of the calculation.
- *name*, varchar(300). Descriptive name.
- *status*, varchar(300). Text describing the current calculation activity.
- *progress*, float. Progress indication as a fraction.
- *completed*, int. Completed: 1, else: 0.
- *failed*, int. Failed: 1, else: 0.
- *scope*, varchar(100).

2.3 Table "*storedviews*"

Each record contains a stored snapshot view of a Panoptes instance, that can be accessed by a permanent url.

Table columns

- *dataset*, varchar(100), FK: *datasetindex.id*. Specifies the dataset the instance is based on.
- *workspace*, varchar(100), FK: *{datasetid}.workspaces.id*. Specifies the workspace in the dataset the instance is based on.
- *id*, varchar(100). Unique identifier of the stored view.
- *settings*, text. Serialised settings string, specifying the instance state.

2.4 Table "*storage*"

Standard utility table used by the DQX server storage mechanism.

Table columns

- *id*, varchar(50)
- *content*, text

3 Database {datasetid}

Each dataset served by the Panoptes instance maps to a corresponding database. The name of the database should correspond to the content of the field *datasetindex.datasetindex.id*.

3.1 Table "*tablecatalog*"

A Panoptes dataset consists in a set of data tables being served. Each record in *tablecatalog* describes an individual data table in the dataset.

Table columns

- *id*, varchar(20). Unique identifier.
- *name*, varchar(50). Display name of the data table.
- *primkey*, varchar(20). Name of the column that contains the primary key of the data table.
- *IsPositionOnGenome*, int. Indicates if this data table contains records that refer to positions on the genome (1 if true, 0 if not). Note that, if true, the corresponding table should have the columns "*chrom*" and "*pos*".
- *settings*, varchar(2000). JSON map object containing further settings.

Settings tokens

- "*QuickFindFields*": A subset of the data table properties that are used in the "Find record" wizard. Formatted as a comma separated string.
- "*GenomeMaxViewportSizeX*". (Only applicable to data tables that contain genomic positions). Maximum viewport size (number, in bp) for which individual records are displayed in the genome browser.
- "*FetchRecordCount*". If true, the record count of the result set is fetched for each query in the table view.

3.2 Table "*workspaces*"

Contains a list of all workspaces defined for the dataset.

Table columns

- *id*, varchar(50). Unique identifier.
- *name*, varchar(50). Display name of the workspace.

3.3 Table "*{tableid}*"

Each data table that is part of a Panoptes dataset maps to a table in the {datasetid} database. The name of that table should correspond to the content of the field *tablecatalog.id*. The records in the table are the data items.

Table columns

The structure of the data table table is not defined by Panoptes, and can be freely chosen in function of the type of data served in this table. The following constraints apply:

- There should be a primary key column with the name referred to in *tablecatalog.primkey*. Panoptes uses the content of this column as a unique identifier for the records of this table.

- Each column that is served by Panoptes (called a “property”) has to be defined in the table *propertycatalog* (see 3.6). The column name should link to *propertycatalog.propid*.
- If, *tablecatalog.IsPositionOnGenome* is set for this data table, the columns *chrom* (string), and *pos* (int) have to be present. These columns contain the chromosome id and position for each record. The column *chrom* should map to the content of *chromosomes.id*.

3.4 Table “*{tableid}INFO_{workspaceid}*”

This table is present for each combination data table×workspace. Its purpose is to hold the custom properties that are added to a data table in the context of a specific workspace. Each custom property corresponds to a column in this table. In addition, there should be an indexed column containing the primary key as defined in 3.3.

3.5 View “*{tableid}CMB_{workspaceid}*”

This view is present for each combination data table×workspace. It joins the information from the tables *{tableid}* and *{tableid}INFO_{workspaceid}*.

3.6 Table “*propertycatalog*”

Each record in this table describes a single property of a data table. This includes both standard properties, and custom properties added in the context of a workspace.

Table columns

- *workspaceid*, varchar(50). For custom properties: the workspace id this property belongs to. FK to *workspaces.id*.
- *source*, varchar(50). Possible states:
  - “fixed”: a standard property of the data table.
  - “custom”: a custom property, uploaded in the context of a workspace.
- *datatype*, varchar(20). Possible states:
  - “Text”
  - “Value”
- *propid*, varchar(50). Identifier of the property (unique in the context of the data table). Links to column names in the table *{tableid}*.
- *tableid*, varchar(20). Identifier of the data table this property belongs to. FK to *tablecatalog.id*, and corresponds to the name of the table *{tableid}*.
- *name*, varchar(50). Display name of the property.
- *ordr*, int. Used to sort the properties.
- *settings*, varchar(1000). JSON map object containing further settings.

Settings tokens

- "*isCategorical*": if true, the content of this field is considered to be categorical, and displayed as such in the Panoptes UI. The states are automatically determined from the corresponding *{tableid}* column content.
- "*decimDigits*": for value properties, the number of decimal digits to be displayed.

- "*minval*": for value properties, the minimum value this property can have.
- "*maxval*": for value properties, the maximum value this property can have.
- "*showInTable*": if true, the content of this field is displayed as a column in the Panoptes query table.
- "*showInBrowser*": if true, the content of this field is displayed as a track on the genome browser (only applicable if *IsPositionOnGenome* is set for the data table in *tablecatalog*).
- "*channelName*": if *showInBrowser* is set, the optional channel name this track will be displayed in. This can be used to group several tracks in the same channel.
- "*channelColor*": if *showInBrowser* is set, the color of the track on the genome browser. Formatted as as string "rgb(r,g,b)".
- "*connectLines*": for value properties, and if *showInBrowser* is set, determines whether or not the points will be connected with lines.

3.7 Table "*chromosomes*"

Contains the chromosomes of the reference genome served in this dataset.

Table columns

- *id*, varchar(20). Identifier of the chromosome.
- *len*, float. Length of the chromosome (in megabases).

3.8 Table "*annotation*"

Contains the genome feature annotation information of the reference genome served in this dataset.

Table columns

- *chromid*, varchar(20). Chromosome the feature is located on. Maps to *chromosomes.id*.
- *fstart*, int. Feature start position (in bp).
- *fstop*, int. Feature end position (in bp).
- *fid*, varchar(20). Unique identifier.
- *fparentid*, varchar(20). If applicable, identifier of the parent feature this feature is a component from. Maps to *annotation.fid*.
- *ftype*, varchar(20). Feature type. Currently supported:
    - "gene": a gene.
    - "CDS": a coding region in a gene. *fparentid* should point the the gene feature it belongs to.
- *fname*, varchar(100). Primary display name.
- *fnames*, varchar(200). Comma-separated list of all equivalent feature names.
- *descr*, varchar(100). Extra description string of the feature.

3.9 Table "*summaryvalues*"

Each record in this table defines a summary value track that can be displayed in the genome browser. Section 4.2 describes how the actual filterbank data for this kind of tracks is stored.

NOTE: if the *tableid* and *propid* values of a summary value track corresponds to a record in the table *propertycatalog*, Panoptes will automatically show both tracks in a single channel on the genome browser. In this way, it is possible to configure two aspects of the same data set:

1. Individual values as a property of a data table (when sufficiently zoomed in, with *tablecatalog.settings[GenomeMaxViewportSizeX]* determining the zoom factor threshold)
2. Summarised filterbank values of the same property (when zoomed out).

Table columns

- *workspaceid*, varchar(50). For custom summary values: the workspace id this property belong to. FK to *workspaces.id*.
- *source*, varchar(50). Possible states:
  - "fixed": a standard summary value.
  - "custom": a custom summary value, uploaded in the context of a workspace.
- *propid*, varchar(20). Unique identifier of the summary value.
- *tableid*, varchar(20).
- *name*, varchar(50). Display name.
- *ordr*, int. Used to sort the summary values.
- *settings*, varchar(2000). Further settings formatted as JSON map.
- *minval*, float. Minimum value.
- *maxval*, float. Maximum value.
- *minblocksize*, int. Size of the smallest block used in the filterbank processing.

Settings tokens

- "*channelColor*": the color of the track on the genome browser. Formatted as string "rgb(r,g,b)".

3.10 Table *"tablebasedsummaryvalues"*

In addition to the summary tracks defined in 3.9, Panoptes can define genome browser summary tracks for each data item in a data table. A common usage includes a data table containing sequenced samples, and summary tracks containing coverage or mapping quality for each item in that data table.
Each record in this table defines a class of data table-based summary tracks. Section 4.3 describes how the actual filterbank data for this kind of tracks is stored.

Table columns

- *tableid*, varchar(50). FK to *tablecatalog.id*, identifying the data table the track is based on.
- *trackid*, varchar(50). Unique identifier of the type of data table based genomic track.
- *trackname*, varchar(50). Display name used for this type of track.
- *settings*, varchar(5000). Further settings formatted as JSON map.
- *minval*, float. Minimum value.
- *maxval*, float. Maximum value.

- *minblocksize*, int. Size of the smallest block used in the filterbank processing.

Settings tokens

- "*channelColor*": the color of the track on the genome browser. Formatted as string "rgb(r,g,b)".

3.11 Table "*settings*"

Extra settings defining the behavior of the dataset.

Table columns

- *id*, varchar(20)
- *content*, varchar(2000)

Possible content

- *AnnotMaxViewportSize*: maximum size of the genome browser viewport (in megabases) for which the reference genome annotation is still displayed.
- *RefSequenceSumm*: "Yes" indicates that the genome browser will contain a track displaying summary information for the reference genome.

3.12 Table "*externallinks*"

Contains a definition of all external web links that are applicable to entities in the Panoptes database. These will show up as action buttons in the relevant popup boxes.

Table columns

- *linktype*, varchar(20). Entity type this link applies to. Currently supported:
    - "annotation_gene": a gene in the reference genome annotation. The entity identifier maps to *annotation.fid*.
- *linkname*, varchar(50). Display name of this web link.
- *linkurl*, varchar(200). Url of this web link. This string can contain the token {id}, which will expand to the entity identifier.

3.13 Table "storedqueries"

Contains the named datatable queries stored by the user.

Table columns

- *id*, varchar(50). Unique identifier.
- *name*, varchar(50). Display name.
- *tableid*, varchar(50). Identifier of the datatable this query relates to. FK to *tablecatalog.id*.
- *workspaceid*, varchar(50). Workspace this query is stored in. FK to *workspaces.id*.
- *content*, text. Base64 encoded query definition.

4 File structure

Panoptes relies on a file structure to store and read extra data that is not contained in the database. The root of this file structure is referred to as {BASEDIR}, as defined in config.py.

4.1 Helper directories

{BASEDIR}/Uploads
Used to store client uploads to the server.

4.2 Global genome summary tracks

The data for the global genome summary tracks, as defined in 4.2, is stored in the directory:
{BASEDIR}/SummaryTracks/{datasetid}/{summaryid},
where

- {datasetid} is the identifier of the dataset, corresponding to *datasetindex.datasetindex.id*.
- {summaryid} is the identifier of the summary track, corresponding to *{datasetid}.summaryvalues.propid*.

4.3 Data table-related genome summary tracks

The data for the data table-related genome summary tracks, as defined in 4.3, is stored in the directory:
{BASEDIR}/SummaryTracks/{datasetid}/TableTracks/{tableid}/{trackid}/{item id},
where:

- {datasetid} is the identifier of the dataset, corresponding to *datasetindex.datasetindex.id*.
- {tableid} is the identifier of the data table, corresponding to *{datasetid}.tablecatalog.id*.
- {trackid} is the identifier of the summary track type, corresponding to *{datasetid}.tablebasedsummaryvalues.trackid.*
- *{itemid}* is the identifier of a data item record in the data table *{tableid}* (i.e. content of the column as defined in *{datasetid}.tablecatalog.primkey*).