

Plantalytics Database Layout

The Plantalytics database design is based on the information needed in the final deployment of the Plantalytics system. This design includes several tables: an environmental data table, a hub/node metadata table, a vineyard metadata table, and a user authentication table.

Environmental Data

The `environmental_data` table is designed to collect the data stream sent from each vineyard's hub. The individual sensor nodes transmit data, which is received by the hub. The hub then batches the collected data and passes it to the database. Data is generated at the following levels:

Node:

- Identification number (ID)
- Environmental conditions
 - Temperature
 - Leaf Wetness
 - Humidity
- Timestamp (when data is sent to the hub)

Hub:

- Identification number (ID)
- Timestamp (when data is sent to the database)
- Vineyard ID

The timestamp from the hub acts as a “batch identifier” for the data.

Collected environmental data table: `environmental_data`

- Compound primary key is (`vineid`, `batchsent`).

Value	Datatype	Purpose
<code>nodeid</code>	<code>int</code>	Node ID number
<code>batchsent</code>	<code>timestamp</code>	Time data sent by hub
<code>datasent</code>	<code>timestamp</code>	Time data sent by node
<code>hubid</code>	<code>int</code>	Hub ID number

Last updated: 08/29/2016

humidity	float	Collected humidity reading
leafwetness	float	Collected leaf wetness reading
Temperature	float	Collected temperature reading
vineid	int	Vineyard ID number

Hardware Metadata

Metadata for the hardware, such as the latitude / longitude locations for each node is manually entered into the database and stored in the hardware_meta table. This data is static and is known before installation of any hardware. Each node in the table has an associated hub which is associated with a vineyard. The data pertaining to that hub and vineyard is replicated in the row for each node. Because any vineyard may have more than one hub, a vineyard ID is included in each batch of data the hub sends. Subscription start and end dates have also been included as any node (with it unique, static ID) may be reused in a different vineyard at some point.

Hardware metadata table: hardware_meta

- Compound primary key is (vineid, hubid, nodeid).

Value	Datatype	Purpose
vineid	int	Vineyard ID number
hubid	int	Hub ID number
nodeid	int	Node ID number
enddate	timestamp	Node end date
lasthubbatchsent	timestamp	Last time the hub sent data
node location	frozen<tuple<float, float>>	Node location (lat / lon)
startdate	timestamp	Node start date

Vineyard Metadata

Metadata for the vineyard, such as the latitude/longitude locations for the center and boundary edge points is manually entered into the database for each vineyard and stored in the vineyard_meta table. This data is static and is known before installation of any hardware. Each vineyard has only one entry in the table, and can be marked as active or inactive.

Vineyard metadata table: vineyard_meta

- Primary key is (vineid)

Value	Datatype	Purpose
vineid	int	Vineyard ID number
boundaries	list<frozen<tuple<float, float>>>	All vineyard boundary edge points (lat/lon), in clockwise order
center	frozen<tuple<float, float>>	Vineyard center
enable	boolean	Flags whether vineyard is active
ownerlist	list<text>	Vineyard owner names
vinename	text	Vineyard name

User Authentication

The user authentication table includes information about each user account, and can be marked as active or inactive.

User authentication table: user_auth

- Compound primary key is (username, password)

Value	Datatype	Purpose
username	text	Login username
password	text	Login password

admin	boolean	Flags whether user is an administrator
email	text	User email address
enable	boolean	Flags whether user is active
securitytoken	text	User unique security token
subenddate	text	Date when user was marked as inactive
userid	int	User ID number
vineyards	list<int>	User vineyard ID list

Administrative Details

Links pertinent to database maintenance:

- Maintenance: https://wiki.bitnami.com/Applications/Bitnami_Cassandra
 - This also includes a “how to” guide for setting up a true cluster. Unfortunately, we were not able to make this work using the guide.
- CQL documentation: <http://docs.datastax.com/en/cql/3.3/>