SeasonWatch Database documentation

The intent of this document is to explain the different tables, fields and their linkages within the SeasonWatch DB for the purpose of programming for SeasonWatch.

Overview of the data

The basic data consists of Trees which belong to Species, Users some of whom belong to User Groups , User Trees which is created when a user adopts a tree and Observations on trees submitted by users.

To capture all Species details there are tables for species alternate names and species images.

Trees have in addition a Tree Measurement table to store height, girth etc of the tree.

All User details are in the table Users and in addition some users may belong to User Groups which are listed in the User Groups table.

When a User adopts a tree a User Tree Table row is created as a relation.

When a User submits an observation on a User Tree, it is stored in the User Tree Observations table.

1) Species details:

a) Species_master
b) Species_alternate_name
c) Language_master
d) Species_images

Species_master			
Description	Stores all the species details like primary name, scientific name, family etc.		
Attribute	Description	Type	Examples of values
species_id	Id of a species	Int(4 digit no)	1000
species_primary_common_name	Common name of the species	String	White Babool
species_scientific_name	Scientific name of the species	string	Acacia leucophloea
species_search_names	Search name of the species(common name	string	White Babool, Acacia leucophloea

	,scientific name)		
family	Family which it belong to	String	Fabaceae
focal			
vegetation_type	Habitat of the species	String	
status_in_india		String	
habitat_type		String	
distribution_in_india		String	
leaf_shape_category		Number	
size_description		String	
flower_description		String	
bark_description		String	
fruit_description		String	
leaf_type		String	
spine_thorn_description		String	
flowering_time		String	
fruiting_time		String	
time_of_leaf_flush		String	
special_notes_on_phenology		String	
similar_species		String	
known_pollinators		String	
known_seed_dispersers		String	
uses_by_humans		String	
list_of_references		String	
special_notes_on_the_species		String	

Primary Key	Species_id

species_alternate_name			
Description	Stores other/alternate names of each species. The alternate names may be of English, regional languages or scientific names. The language_id field tells us which language is the name in a record in.		
Attribute	Description	Type	Examples of values
alternate_name_id	Alternate name Id of a species	int(10)	1
language_id	Language id	int(10)	3
alternative_name	name of the species in different language.	varchar(60)	Acacia leucophloea
species_id	Species_id from the species_master table	int(10)	1000
Primary Key	alternate_name_id		
Foreign Keys	Species_id (p_species_master_species_alternate_name_fk) Language id (language_master_species_alternate_name_fk)		

language_master			
Description	Table which lists all languages and gives an id language_id.		
Attribute	Description	Type	Examples of values
language_id	Language id	int(10)	1
Language_name	Language name	varchar(20)	1-scientific,2-English,3-Hindi,4-Kannada,5-Tamil,6-Telgu,7-Marathi,8-Malayalam

Primary Key	language_id		

species_images			
Description	Stores all the species image,image type,path,file name gives species_image_id link to connect.		
Attribute	Description	Type	Examples of values
species_image_id	Id of a species	Integer(4 digit number)	1000
tree_image_desc	Common name of the species	String	White Babool
image_type	Scientific name of the species	string	Acacia leucophloea
file_name	Search name of the species(common name ,scientific name)	string	White Babool, Acacia leucophloea
path_name	Family which it belong to	String	Fabaceae
species_id		int(10)	
Primary Key	species_image_id		
Foreign Key	Species_id		

2) Tree details:-

The following three tables are used to store the tree information added by the user.

a) User_tree_table.
b) Location_master
c) tree_measurement .
d) seswatch_states

user_tree_table			
Description	Once a user adopts a tree we add a row here to link the user_id with a tree_id and get a unique reference to this relation i.e. user_tree_id. This is important because later another user may adopt the same tree and then we will have a different user_tree_id for the same tree_id. This table also stores a tree_nickname to uniquely identify the tree in common language without having to use a numeric id etc		
Attribute	Description	Type	Examples of values
tree_Id	Id of a tree	int(10)	1000
tree_desc	Description about tree	varchar(1000)	
Is_fertilized		Tinyint(1)	
Is_watered		Tinyint(1)	
species_id		int(10)	
tree_location_id		int(10)	
location_type		varchar(60)	
aspect		varchar(60)	
distance_from_water		int(10)	
date_of_addition		timestamp	
Tree_code_sms		Varchar(100)	
deleted		int(11)	
added_by_user_id		int(11)	
Primary Key	tree_Id		
Foreign Keys	species_id (p_species_master_trees_fk), tree_location_id location_master_trees_fk		

location_master	

Description	Master table which stores all address, latitude/longitude and gives a location_id to them for reference in other tables.		
Attribute	Description	Type	Examples of values
tree_location_id	Tree_location_id	int(10)	
state_id		int(10)	
city		varchar(40)	
longititude		Decimal(9,7)	
latitude		decimal(9,7)	
location_name		varchar(100)	
Zoom_factor		Int(10)	
Primary Key	tree_location_id		
Foreign Keys	state_id(p_states_fk)		

tree_measurement			
Description	Master table which stores all measurement of the tree, tree_height,tree_girth,tree_damage information,date of addition.		
Attribute	Description	Type	Examples of values
measurement_id	Id of a tree	int(10)	
tree_Id		int(10)	
User_id		Tinyint(10)	
Date_of_measurement		date	
tree_girth		decimal(7,2)	
tree_height		decimal(7,2)	
tree_damage		tinyint(1)	
date_of_addition		timestamp	

Primary Key	measurement_id
Foreign Keys	tree_Id (p_trees_tree_measurement__fk) tree_location_id user_tree_measurement__fk

tree_photos			
Description	Stores all the tree photo with file name,caption.		
Attribute	Description	Type	Examples of values
photo_id	Id of a tree	int(10)	
photo_filename		varchar(1000)	
photo_caption		tinyint(1)	
Tree_id		tinyint(1)	
Primary Key	photo_Id		
Foreign-Key	Tree-id		

seswatch_states			
Description	Stores all the states name with state_id.		
Attribute	Description	Type	Examples of values
state_id	Id of a tree	bigint(20)	
state		varchar(20)	
Primary Key	state_id		
Foreign-Key			

3) User details:-

All seasonwatch user details will stored in these following tables.

1) Users.
2) User_groups.

users			
Description	Stores all SeasonWatch user details like username, full name, pwd (to be encrypted as MD5), address etc. Further it also stores user_category which maybe individual (user is just an individual participant), schools (user is part of a school group) or school-seed (user is part of a school group within Kerala's SEED program) etc. group_id field is relevant if the user is a non-individual user. If so this field tells us which group the user belongs to		
Attribute	Description	Type	Examples of values
user_id		bigint(20)	
md5_id		varchar(200)	
full_name		tinytext	
user_name		varchar(200)	
user_email		varchar(200)	
user_role		varchar(200)	
pwd		Varchar(200)	
address		text	
address1		text	
address1		text	
city		text	
educational_district		varchar(50)	
district		text	
state_id		int(10)	
landline_stdcode		varchar(6)	
landline_num		bigint(15)	

mobile		bigint(15)	
zip		varchar(6)	
fax		varchar(200)	
website		text	
date		Date	
users_ip		varchar(200)	
approved		int(1)	
activation_code		int(10)	
banned		int(1)	
user_category		varchar(30)	
group_id		int(10)	
group_role		varchar(50)	
group_role		varchar(10)	
country		varchar(100)	
registered_on		varchar(100)	
hashkey		varchar(300)	
date_of_addition		timestamp	
Primary Key	user_id, user_email		
Foreign Keys	state_id		

user_groups			
Description	Master table to store all group details like group_name etc and give them a group_id for referring within user table.		
Attribute	Description	Type	Examples of values

group_id	Id of a tree	int(10)	
coord_id		varchar(1000)	
group_name		Tinyint(1)	
school_code_sms		Tinyint(1)	
Primary Key	Group_id		
Foreign Keys	coord_id (p_species_master_trees_fk),		

4) After a User adopts a Tree:

i) User_tree_table
ii) User_tree_observations

user_tree_table			
Description	Once a user adopts a tree we add a row here to link the user_id with a tree_id and get a unique reference to this relation i.e. user_tree_id. This is important because later another user may adopt the same tree and then we will have a different user_tree_id for the same tree_id. This table also stores a tree_nickname to uniquely identify the tree in common language without having to use a numeric id etc.		
Attribute	Description	Type	Examples of values
user_tree_id		int(10)	
tree_nickname		varchar(1000)	
tree_id		Int(10)	
user_id		bigint(20)	
Last_observation_date		date	
Members_assigned		varchar(200)	
tree_code_sms		varchar(5)	

Date_of_addition		timestamp	
Primary Key	user_tree_id		
Foreign Keys	tree_id (trees_user_tree_table_fk), user_id(p_user_user_tree_table_fk)		

User_tree_observations			
Description	Once a user starts doing his weekly observations the observation details are entered on this table. It stores the observation date, user_tree_id to identify the tree, the user_id of the user and then all the observation parameters about the lead, flower and the fruit.		
Attribute	Description	Type	Examples of values
observation_id		int(10)	
date		date	
observation_time		time	
is_leaf_mature		tinyint(1)	
is_leaf_fresh		tinyint(1)	
is_flower_bud		tinyint(1)	
is_fruit_ripe		tinyint(1)	
is_fruit_unripe		tinyint(1)	
is_flower_open		tinyint(1)	
freshleaf_count		varchar(10)	
matureleaf_count		varchar(10)	
bud_count		varchar(10)	
fruit_ripe_count		varchar(10)	
fruit_unripe_count		varchar(10)	
open_flower_count		varchar(10)	

leaf_caterpillar		tinyint(1)	
flower_butterfly		tinyint(1)	
flower_bee		tinyint(1)	
fruit_bird		tinyint(1)	
animal_desc		varchar(1000)	
birds_desc		varchar(1000)	
insect_desc		varchar(1000)	
other_desc		varchar(1000)	
temperature_max		int(4)	
temperature_min		int(4)	
rainfall_mm		int(4)	
humidity_mm		decimal(3,2)	
user_tree_id		nt(10)	
user_id		bigint(20)	
deleted		int(11)	
presence		int(11)	
date_of_addition		timestamp	
Primary Key	observation_id		
Foreign Keys	User_tree_id		

What should a programmer do in the following situations?

1. To add users
    - Insert a row in users with all details. Ensure that user_name, pwd (to be encrypted as MD5), full_name and user_category are surely there. (For SEED, currently for group_role='member' and user_category='school-seed' these are SEED member users and they don't necessarily need a username and pwd.

- If the user belongs to a group, then insert a row in the user_groups with details like group_name.
- Get the new group_id and update this in the newly created row in users. Further update the group_role as ‘coord’ (if user is a group coordinator) or ‘member’.

2. To add trees and assign it to users
    - Trees are always added by users.
    - Select a tree with particular species.
    - So first insert a row in trees with all the details.
    - Insert a row in tree_measurement with height etc.
    - Insert a row in location_master to store new location details.
    - Update the new location_id in the newly created row in trees.
    - Then insert a row in user_tree_table with the user_id and tree_id just created.
3. To add observations
    - Insert a row in user_tree_observations using the appropriate user_tree_id and user_id. Ensure that the date and other observation details are correctly entered.