# Plot, Plot\_history and forest\_monitoring tables

The most of the functionality in FRMS QGIS plugin is related to the three tables in FRMS data model: Plot, forest\_monitoring and plot\_history. The data in these three tables are interrelated to each other primary via global primary keys of the tables:

**plot\_uuid:** Primary key of the plot table. plot\_uuid is used to link rows in plot\_history and forest\_monitoring to rows in plot\_table.

**plot\_history\_uuid:** Primary key of plot\_history table. Plot\_history\_uuid is used to link rows in forest\_monitoring a table to plot\_history table

**forest\_monitoring\_uuid:** Primary key of forest\_monitoring table.

A business key which is a combination of commune\_code, compt\_code, sub\_compt\_code and plot\_code is also available but it does not define unambiguously the row in plot\_history table. The uniqueness of the business key among local databases (among FRMS installations) is managed manually in organizational level.

# Plot

Plot table contains the current data of the plot. When plot is created or when plot data is changes the content of the plot is copied to plot\_history table.

When user deletes the plot, the plot is deleted from plot -table but also the rows related to the plot in Forest monitoring are deleted. Also some of the rows related to plot in plot\_history table are deleted, but which rows, is dependent on the content of the plot\_history table:

1. The oldest plot\_history row (based on datemodify), so called plot baseline, is always left to plot\_history table.
2. If plot\_history rows does not contain any forest\_monitoring activities only the plot\_baseline is left
3. If all plot\_history rows does contain only forest\_monitoring activities only the plot\_baseline is left
4. If plot\_history contains both monitoring activity and non monitoring activity rows, only the baseline row and non-monitoring activity rows are left

Note that as the plot\_history rows are left when the plot is removed, it is possible for user to create a new plot with a plot\_code which aready exists in plot\_history table but not in plot\_table. In that case the existing plot\_history rows will be associated to the new plot.

When user changes plot\_code, the application verifies that the new commune\_code, compt\_code, sub\_compt\_code and plot\_code combination is not in use. If is it already in use is plot table,user is given an error message. If it is not in use in plot\_table then the plot\_code is changed in plot, plot\_history and forest\_monitoring tables.

# Forest\_monitoring

When the forest monitoring activity is added to the plot, the data related to that activity is saved into forest\_monitoring -table. Adding forest monitoring activity to the plot also changes data in plot\_table and creates a new row into plot\_history table and links the rows created into forest\_monitoring and plot\_history tables with plot\_history\_uuid column.

Each row in forest\_monitoring table records the change of the following attributes by keeping the previous and current values of the attributes:

|  |  |
| --- | --- |
| **Previous value** | **Current value** |
| forest\_type\_code | forest\_type\_code\_after |
| forest\_func\_sub\_code | forest\_func\_sub\_code\_after |
| forest\_org\_code | forest\_org\_code\_after |
| forest\_use\_sit\_code | forest\_use\_sit\_code\_after |
| actor\_type\_code | actor\_type\_code\_after |
| actor\_id | actor\_id\_after |
| site\_cond\_code | site\_cond\_code\_after |
| p\_forest\_org\_code | p\_forest\_org\_code\_after |
| tree\_spec\_code | tree\_spec\_code\_after |
| nar\_for\_org\_code | nar\_for\_org\_code\_after |
| land\_use\_cert\_code | land\_use\_cert\_code\_after |
| conflict\_sit\_code | conflict\_sit\_code\_after |
| plant\_state\_code | plant\_state\_code\_after |

The attribute – attribute\_after pairs are used so that when attribute is changed the old value is copied to attribute column and the new value is saved to \_after column. If attribute is not changed then the both columns have the same value.

For example the following scenario of updating forest\_type\_code attribute:

1. users adds a forest\_monitoring a monitoring activity which changes forest\_type\_code from Null to 50
2. User adds second forest monitoring activity to the same plot but the forest\_type\_code is not changed
3. User adds third forest monitoring activity to the same plot but the forest\_type\_code is not changed
4. users adds fourth monitoring activity which changes forest\_type\_code from 50 to 60
5. users adds fifths a monitoring activity which changes forest\_type\_code from 60 to 70
6. User adds sixth forest monitoring activity to the same plot but the forest\_type\_code is not changed

Would be recorded into forest\_monitoring table as following

|  |  |  |  |
| --- | --- | --- | --- |
|  | **forest\_type\_code** | **forest\_type\_code\_after** | **action** |
| **1** | NULL | 50 | NULL -> 50 |
| **2** | 50 | 50 | no change |
| **3** | 50 | 50 | no change |
| **4** | 50 | 60 | 50 -> 60 |
| **5** | 60 | 70 | 60 -> 70 |
| **6** | 70 | 70 | no change |

# Editing forest\_monitoring

User can edit existing forest monitoring activities from Monitoring History window of FRMS QGIS plugin. User can edit all attributes in forest\_monitoring table, which are related to the monitoring type under editing. User cannot change existing forest monitoring type to another, but can change the subtype. Also user can change the implementation date of the monitoring activity (date\_time attribute). All the changes user does to forest\_monitoring entry (row) are also done to the row in plot\_history table.

1. If the attribute value in forest\_monitoring table is changed -> the attribute value is also changed in the row in plot\_history table, which has the same plot\_history\_uuid as in the edited row in forest\_monitoring table.
2. If user changes forest\_monitoring.date\_time the plot\_history. dateapproval and plot\_history. year\_approval is changed accordingly. Plot\_history table does not have date\_time column but dateapproval is used for that. Year approval column contains the year part of dateapproval value.
3. If forest\_monitoring.change\_type\_id is changed, the change\_type\_id and change\_type\_id\_last values are changed in plot\_history accordingly.
4. if forest\_monitoring record (row) is deleted, the row in plot\_history table, which has the same plot\_history\_uuid as the deleted row in forest\_monitoring table is deleted too.

Editing existing forest\_monitoring entries in monitoring history does not change anything in the plot -table.

# Plot history

Every time when data in plot table is changed or monitoring activity is created, a new row into plot\_history table is created. The change\_type\_id field is updated in plot\_history only when the forest monitoring activity is added to the plot. Also the datemodify and dateapproval fields are different depending on whether the plot\_history row is forest monitoring activity row or change in plot row. When forest monitoing activity is added into the plot, the datemodify contains the time stamp of the moment when the row is created and dateapproval contains the date when the activity was executed in the field (same as forest\_monitoring.date\_time). If the plot is changed but no forest monitoring activity is added to the plot, then both datemodify and dateapproval contain timestamp of the moment when the row is created.

Plot\_history table is used when thematic maps by year are created and when plot data need to be restored to older version. When forest monitoring activity is created, the plot\_history row does not contain information of change in the forest\_monitoring attributes but the latest value is always saved to the actual attribute e.q forest\_monitoring. forest\_type\_code\_after is saved to plot\_history.forest\_type\_code. The only exception to this is forest\_use\_sit\_code of which the change information is saved in the same way as in forest\_monitoring table i.e. forest\_monitoring.forest\_use\_sit\_code is saved to plot\_history.forest\_use\_sit\_code and forest\_monitoring.forest\_use\_sit\_code\_after is saved to plot\_history.forest\_use\_sit\_code\_after. The reason for this exception is that the change of forest\_use\_sit\_code is needed in thematic maps.

User can restore plot to older version in History plot window of FRMS. When plot is restored then:

1. The data is copied from selected plot\_history row to plot table
2. The plot history rows which datemodify > datemodify of selected row are deleted
3. The forest\_monitoring rows which have the same plot\_history\_uuid as the rows which are deleted from plot\_history table, are deleted.

# Forest\_monitoring attributes in plot and plot\_history tables

The following table shows how different plot state attributes of forest\_monitorig table are saved to plot\_history and plot tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Forest\_monitoring** | **Plot\_history** | **Plot** |  |
| forest\_type\_code\_after | forest\_type\_code | forest\_type\_code |  |
| forest\_func\_sub\_code\_after | forest\_func\_sub\_code\_after | forest\_func\_sub\_code |  |
| forest\_func\_sub\_code | forest\_func\_sub\_code | N/A |  |
| forest\_org\_code\_after | forest\_org\_code | forest\_org\_code |  |
| forest\_use\_sit\_code\_after | forest\_use\_sit\_code\_after | forest\_use\_sit\_code |  |
| forest\_use\_sit\_code | forest\_use\_sit\_code | N/A |  |
| actor\_type\_code\_after | actor\_type\_code | actor\_type\_code |  |
| actor\_id\_after | actor\_id | actor\_id |  |
| site\_cond\_code\_after | site\_cond\_code | site\_cond\_code |  |
| p\_forest\_org\_code\_after | p\_forest\_org\_code | p\_forest\_org\_code |  |
| tree\_spec\_code\_after | tree\_spec\_code | tree\_spec\_code |  |
| nar\_for\_org\_code\_after | nar\_for\_org\_code | nar\_for\_org\_code |  |
| land\_use\_cert\_code\_after | land\_use\_cert\_code | land\_use\_cert\_code |  |
| conflict\_sit\_code\_after | conflict\_sit\_code | conflict\_sit\_code |  |
| plant\_state\_code\_after | plant\_state\_code | plant\_state\_code |  |

# Thematic maps

FRMS has two thematic map layers. Plot layer shows the thematic maps based on the data in the plot table (Show thematic maps selection in FRMS menu) and plot history layer which shows the thematic maps by year based on the data in plot\_history table.

Even the thematic maps by year are related to forest monitoring activities, The forest\_monitoring table is not used in thematic maps at all. It was decided to use plot\_history instead because the most of the required attributes were already saved also to plot\_history and it had also the full plot data including geometry in it. Also the use of a database view for the thematic maps by year were considered but it was not feasible due to performance issues.

FRMS application keeps forest\_monitoring and plot\_history tables synchronized for thematic maps i.e. when something is changed in forest\_monitoring table the change is also done in plot\_history table. The synchronization algorithm uses plot\_history\_uuid to link the data between forest\_monitoring and plot\_history table.