

Manipulation Queries in Marushka Design



GEOVAP

CONTENTS

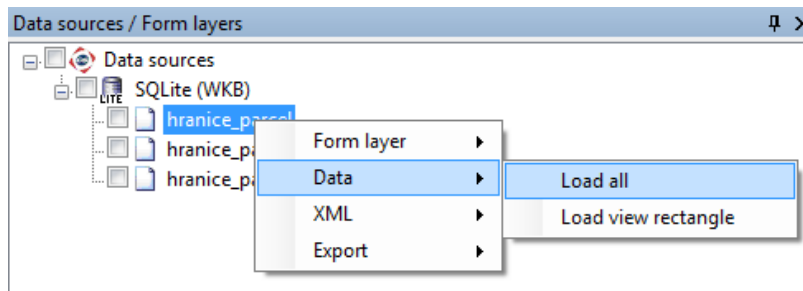
- 1 AIM OF THE EXAMPLE.....2
- 2 WORKING WITH EXAMPLE2
- 3 DIALOG BOX SAMPLE.....3
- 4 A BRIEF DESCRIPTION OF THE EXAMPLE IN MARUSHKA DESIGN4

1 Aim of the Example

In this example we will demonstrate manipulation queries in Marushka Design. This example was created in version 4.1.0.4, so it does not have to be compatible with older versions.

2 Working with Example

- Unzip the **ModifySelection_EN.zip** into **c:\MarushkaExamples** folder. The target folder must be respected due to interconnection of paths with the project. In case of placing the files in the different folder, it would not be possible to work with an example.
- Open the project **ModifySelection_EN.xml** in MarushkaDesign environment.
- Select form layer **hranice_parcel**, in the context menu choose Data – Load all:



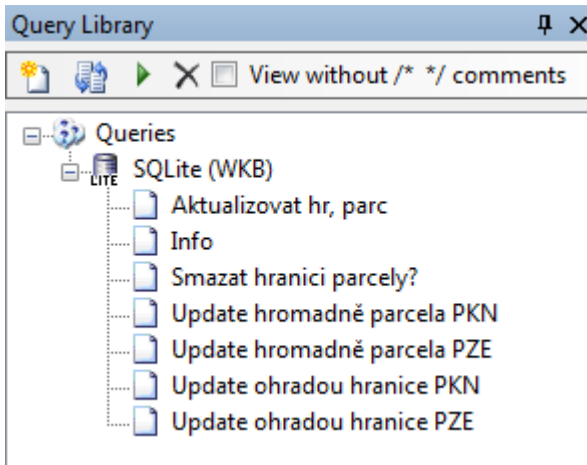
- In map window choose „Fit All“:



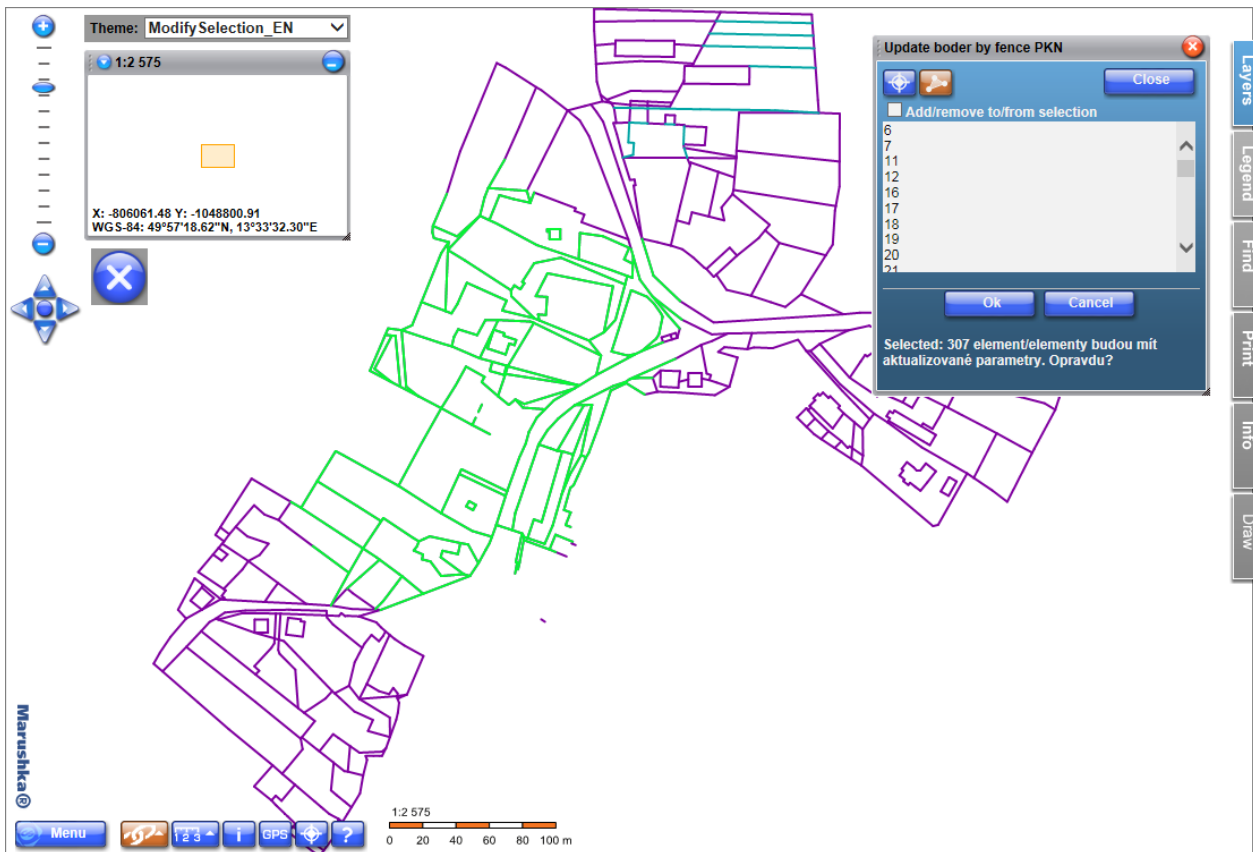
- Launch the local web server:



3 Dialog Box Sample



Img 1: Query library from this example



Img 2: Update border by fence PKN query result, by green are highlighted parcels with modified attribute, their list is in top right in the dialog window


4 A brief description of the Example in Marushka Design

This example demonstrates queries, manipulating with elements, including multiple attribute modifications and deleting elements.

This example contains a database in SQLite with three form layers. The first layer *hranice_parcel* contains line parcel boundaries in the area of desire. Layers *hranice_parcel highlight* and *hranice_parcel highlight2* are identical layers with *hranice_parcel*, but each has set different *LineColor* in symbology, because they are used for highlight in MultiSelect. The last layer is called *hranice_parcel select* and is used in multiselect query for selection.

The main content of this example is stored in the query library, where is a total of 7 queries.

Query *Info* is an information query to check the values modified by query *Update border by fence PKN*, respectively *Update border by fence PZE*. This query as a result displays all the columns from the table *hranice_parcel*, including column *PAR_TYPE* that will change after execution of the given queries. It is not possible to start this query separately in client.

Query *Delete parcel border?* is a simple delete query, which deletes a selected line element from the database. It is started in client with activated button *Preselect*  and clicking on selected line element from the table *hranice_parcel*.

Queries *Update border fence PKN* and *Update border fence PZE* are queries type MultiSelect, which will after the end of multiselect induce their subordinate queries, concretely *Update border multiselect parcel PKN*, respectively *Update border multiselect parcel PZE*. Query *Update border fence PKN* is used for multi selection of parcels and subsequently to change their attributes, specifically in the table *hranice_parcel* will be set the value of the column *PAR_TYPE* to PKN. Similarly, the query *Update border fence PZE* changes the value of column *SET_TYPE* to PZE. These queries are quite variable and it is possible e. g. instead of update query can be added delete query, which would after the execution of multi select delete selected elements or it is possible to modify more attributes at the same time.

These queries are available in client after clicking on button  the list of queries is offered.

Queries *Update border by fence PKN* and *Update border by fence PZE* are auxiliary queries to the queries *Update border fence PKN* and *Update border fence PZE* and these do perform aforementioned updates. The specific form of both queries is here:

```
UPDATE hranice_parcel SET PAR_TYPE= 'PKN' WHERE ID in (~(idlist)ID~)
UPDATE hranice_parcel SET PAR_TYPE= 'PZE' WHERE ID in (~(idlist)ID~)
```

It is not possible to run these auxiliary queries run separately, it is possible only through its superior queries.