Task documentation XTD: XML2DDL in Python3 for IPP 2012/2013
Name and surname: Marek Marušic
Login: xmarus05

1 Purpose of script

This script, written in Python3, converts valid xml files to set of SQL commands, which creates corresponding database of tables with appropriate table structures and relations. Furthermore, it is able to output table relations from input file in xml format.

2 Processing of arguments

Firstly we have to process commandline arguments. For this, we used library argparse from python standard library. Moreover, we had to check whether the given combination of arguments meets required conditions. Error is printed to standard error stream when there are multiple occurrences of specified argument found or argument –etc is specified along with -b, or –help is specified along with any arguments.

3 XML parsing

We used xml.etree.ElementTree library to parse xml file. Next, we used recursive function parse_children() to create dictionary of tables, columns and relations from xml element tree.

3.1 I/O methods

When –input is specified, scripts expects specification of file name, otherwise script is reading from standard input. Likewise, when no –output parameter is used, script prints output to standard output. When there is error during reading, parsing or writing to files, program exits with error 90.

3.2 Data type conflicts

During parsing input, we can encounter few difficulties with different data types of processed elements and attributes. We used integer values as representation of specific data types. Thereafter, we could easily compare types the elements or attributes, and determine which type we should use for the data. Next we used the integer values as index to dictionary, which contains string representations of the data types.

3.3 Argument –etc

During data prasing from xml, we used dictionaries for every element. The dictionaries contains data about how many times was element listed in its table. From this counts, we was able to process and edit data by function process_etc(). This function iterates over all tables and columns, and either it creates that much columns from the one column, as we have its count in dictionary or it deletes column from the table and adds record to table with name column. This depends on number occurrences of the column, and whether it is smaller or greater than specified etc argument.

3.4 Argument –g

When this parameter is specified, we have to generate xml output, with table relations. To store the xml data we used again xml.etree.ElementTree, and function create_xml (). This function has two recursive sub functions get_rel() and get_brother_recursive(). The first function generates relations from parent tables. The second function generates all other relations (N:M), which we didn't create during the first function. To output created xml, we used function prettify() with help of xml.dom.minidom library.

4 Extensions

This script has extension –isvalid, which is controlled by function validate(), if the file is able to be inserted to generated data, script continues, otherwise it exits with 91 exit code.