Documentation for XTD: XML2DDL project to IPP 2014/2015

Vojtech Mašek (xmasek15)

1 Problem analysis

Description

Creates a set of SQL commands that will generate the appropriate structure of tables in the SQL database in which the data contained in input file can be stored. Script will analyze every element (except root element) and generate table named after the element. Table contains columns representing element attributes, foreign keys of other tables (sub-elements) and if is set any then also a element value. Every table contains primary key composited of prefix, element name and suffix. Tag (table) and attribute names are treated as case insensitive.

Input

As input file is considered an XML file that can be provided as standard input or passed via script parameters.

Output

Output of script can be divided to two possibilities, one is set of SQL commands generated from inputted XML and second one is generated output in XML format symbolizing all relations between tables in inputted XML.

Script options

If is set parameter “-a” there wont be generated any of the columns based on attributes of the element.

If is set parameter “-b” all same name sub-elements of the element will be considered as same sub-element (cannot be set with “--etc”).

If set “--etc”, than number after the etc symbolizes maximum count of the columns generated from same name sub-elements.

Also an header can be written to output if “--header” is set, sequence to write must be placed in apostrophes.

If user wants to valid an XML data he can type “--isvalid” to use extension, and algorithm will parse and validate the table structure and data.

For further information about script parameters try using “--help” parameter.

2 Implementation

XML input handling

Input in XML format is read and handled as UTF-8 string and parsed using SimpleXML library that is included in PHP standard extensions, it provides and easy to use tool set to convert XML into an object. Than it can be proceeded with selectors and array iterators to associative array which is representing a structure of tables, attributes, columns and values used by script to generate an SQL commands in DDL format or relationships written to output XML.

Parameters

Script parameters are handled by *getopt()* function, that will parse them and do the all work with optional parameters and values of switches for us. Than there is only a need to validate using regular expression, that there was no other parameter passed to script (which is considered as error) because *getopt()* is skipping not recognized options.

Cardinalities

For purposes of outputting relationships between tables there was implemented basic algorithm recognizing cardinalities of given tables in relations to another tables. it is described in detailed assignment of XTD project. It expects these types of relations: one to one, one to many, many to one, many to many and none. $Relation : Table \times Table \rightarrow \{1{:}1, 1{:}N, N{:}1, N{:}M, \varepsilon\}$

The none (ε) is set by default it means no relationship to any table. Than there are basic initialization to one to one (*1:1*) – if table *a* is same as table *b,* if they are not same tables, transitivity may be many to many (*N:M)* – if tables *a* and *b* are transitive in both ways, one to many (*1:N)* or many to one *(N:1)* – it is true if one of tables is transitive to other but backwards it is not true.

Data type recognizing

There are five types of generated data types in order from lowest to the highest: BIT (Boolean value), INT (whole number), FLOAT (real number), NVARCHAR (attribute text value), NTEXT (normal text value). Thees are recognized and assigned as data types of values (if is set any). If there is a conflict, always an highest data type is chosen.

Keyword conflicts

Implementation is case insensitive, it means that there is a need to check every keyword not only for key conflict as it is, but also lower it to all lower case characters so every keyword will be checked.

Validating extension

Basic algorithm to validate if XML data passed as file in parameter are valid to write to SQL structure generated from inputted XML file. It will iterate through array structure that needs to be validated and checks if there is entry in model table structure.

Implementation documentation

Whole script is documented using PHP Doc doxygen like documentation. For further information and detailed algorithm descriptions see source code or generate documentation from it.

3 Testing

Extended automatic testing was used to test script outputs, there were basic set and then tens of other special test to verify behavior in some of the more complex situations. I have edited basic tests to inspect outputs fully automatically. Some third party tools were used to verify DDL tables structures and XML output files with relations.