Documentation of XTD: XML2DDL in PHP 5 for IPP 2016/2017
Name and surname: Jakub Kulich
Login: xkulic03

Program options

Program options are parsed by PHP's `getopt()` function. Function parses options and creates associative array of these options. `getopt()` doesn't handles options that are not defined so error handling of such options must be done manually.

Input and output

Input and output are selected by the program options `--input` and `--output`. If no file is selected then `STDIN` and `STDOUT` are defaultly used as input stream and output stream.

Parsing input file

Loaded file is parsed using `simplexml_load_string()` function that is part of PHP. Function parses XML string and returns XML as object. This XML object is then converted to associative array so operations above XML is simplier than with operations above XML object.

Overall implementation

Implementation is composed of two important parts:

- One part finds out what contains each table
- Second part finds out relations between tables

I've chosen approach of recursive functions - after som investigation I found out that this solution won't be perfect but it'd be sufficient and implementation speed is highest.

Naming

XML's are not case-sensitive, so if XML contains 'car', 'Car or 'CAR', it will always be one entity. All entity's name were lowercased to simplify solution.

Relations

There are four kind of relations between tables:

- None
- 1:1
- 1:N
- N:1

Defaultly is selected relation 'None' and later is promoted to higher relation.

Data types

Data types are similiar to relations - data type is selected on first detection. If higher datatype is found later then it's set to higher data type. Data types are these from lowest to highest:
BIT (boolean), INT (number without floating point), FLOAT (number with floating point), NVARCHAR (text in attribute) and NTEXT (text).

Extensions

Variant XTD has one possible extension - validation that XML data in passed file are compatible with data structure of input file. Extension works in way that it creates tables from passed file and compares them to the tables created from input file. If tables from validated file contains more attributes or more relations with other tables that don't exist in validation file then file is not valid. Same applies if data types in validated files are higher than in validation file.