

Meno a priezvisko: Ladislav Šulák

Login: xsulak04

1 Zadanie

Úlohou bolo vytvoriť skript v jazyku PHP 5, ktorý podľa vstupného XML súboru vytvorí sadu SQL príkazov pre vkladanie tabuliek do databázy. Program využíva základné knižnice, a tiež triedu SimpleXML.

2 Riešenie

Nebolo využívané objektovo orientované programovanie, objekty a triedy sa využívali iba v rámci práce s triedou SimpleXML. Využívajú sa funkcie, globálne premenné, a rekurzia. Program je členený do viacerých častí ktoré budú popísané nižšie.

A. Spracovanie parametrov

Spracovanie parametrov vykonáva funkcia `params()`, pri ošetrovaní a hľadani parametrov sa využívajú funkcie pre prácu s reťazcami, ako napríklad `substr()`, `strcmp()`. Ak dôjde k chybovému stavu, program končí s návratovou hodnotou 1.

B. Spracovanie vstupného XML súboru

Spracovanie vstupného XML súboru vykonáva funkcia `input()` ktorá obsahuje funkciu `children_processing()`. V nej sa využíva trieda SimpleXML, a jej metódy: `attributes()`, `children()`, `getName()`. Táto funkcia rekurzívne vyhľadáva všetky elementy, podelementy, atribúty, textové podelementy, a dátové typy posledných dvoch spomenutých, a priebežne ich ukladá do globálneho viacrozmerného poľa `element_array`. Parametre ovplyvňujú spracovanie vstupného súboru. Parameter `-a` ignoruje všetky atribúty. Ak dôjde k chybovému stavu pri otvorení vstupného súboru, program končí s návratovou hodnotou 2. Pri nezadaní parametru `--input=` sa načíta zo štandardného vstupu. Ak je vstupný XML súbor invalidný, program končí s návratovou hodnotou 4.

C. Konflikty

Počas spracovania vstupného XML súboru môžeme naraziť na niekoľko konfliktov.

Prvý z nich je dátový. Generujeme typy textových podelementov, a typ atribútov. Pokiaľ je viac atribútov rovnakého názvu ale rozdielneho typu, vygeneruje sa iba 1, s najvyšším dátovým typom podľa poradia priority BIT, INT, FLOAT, NVARCHAR, NTEXT (najväčšia). To isté platí aj pre textové podelementy. NVARCHAR sa vzťahuje iba na atribúty, a NTEXT na textové podelementy.

Druhý typ konfliktu je konflikt názvov. Pokiaľ je názov stĺpcu vznikajúci z atribútov alebo textových podelementov zhodný so stĺpcom ktorý reprezentuje odkaz do tabuľky program končí s návratovou hodnotou 90.

D. Prepínače `--etc=n`, `-b`

Prepínač `--etc=n` určuje maximálny počet vygenerovaných podelementov. Ak má daný element viac podelementov než je číslo `n`, nevygeneruje sa žiadny, ale vygeneruje sa cudzí kľúč. Pokiaľ je zadaný súčasne aj prepínač `-b`, dôjde k chybovému stavu, program skončí s návratovou hodnotou 1.

Prepínač `-b` ovplyvňuje počet vygenerovaných podelementov rovnakého názvu. Tento počet je 1.

Implementačne to bolo riešené vytvorením nového globálneho poľa `N_pole`, ktoré vzniklo duplikovaním hlavného poľa, a podľa neho sa pridávajú cudzie kľúče do hlavného poľa. Túto činnosť vykonáva funkcia `N_arr_modify()`.

E. Prepínač -g

Prepínač `-g` určuje aké majú tabuľky vzťahy vzájomne medzi sebou. Zaisťujú to funkcie `get_relation()`, `transitive_iteration()`, `transitive_iteration()`, ešte predtým sa vytvorí pomocné pole `children_array`, ktoré vzniklo z hlavného poľa, a `relation_array`, do ktorého sa budú ukladať typy vzťahov. Relácie sú symetrické, reflexívne a tranzitívne. To v praxi znamená, že ak vzťah medzi tabuľkou A a B je 1:N, tak vzťah medzi tabuľkou B a A je N:1 – symetrickosť.

Reflexívna preto, že je každá tabuľka v relácii sama so sebou vo vzťahu 1:1. Okrem týchto troch vzťahov existuje ešte vzťah N:M.

Tranzitívnosť v tomto prípade znamená to, že ak majú nejaký vzťah tabuľky A a B, aj B a C, existuje vzťah aj medzi tabuľkami A a C.

F. Výstup

O generovanie výsledného výstupu sa stará funkcia `print_elements()`. Pokiaľ nie je zadaný parameter `--output=file` výsledný výstup bude presmerovaný na štandardný výstup. V opačnom prípade sa generuje do súboru daného reťazcom file. V prípade, že je zadaný parameter `-header=head`, prvý riadok vo výslednom výstupe bude mať podobu `--head` nasledovaný dvoma prázdnyimi riadkami. Pokiaľ sa nepodari otvoriť súbor pre výpis, program sa ukončí s návratovou hodnotou 2.

Pokiaľ sa program skončí s inou hodnotou než je 0, tak sa na štandardný chybový výstup vypíše vhodná chybová hláška. Funkcia ktorá zaisťuje koniec programu a obsahuje chybové hlášky sa nazýva `err_msg()`.

3 Záver

Skript bol vyvíjaný a testovaný na školskom serveru Merlin s operačným systémom CentOS 6.5. Pri testovaní boli použité poskytnuté testy, a tiež vlastné. Pri vývoji sa osvedčilo prostredie PhpStorm 7.1.2.