

Dokumentace úlohy XTD: XML2DDL v PHP do IPP 2016/2017

Jméno a příjmení: **Petr Buchal**

Login: **xbucha02**

Základní popis

Skript slouží k získání příkazů pro vytvoření tabulek databáze, popřípadě pro zjištění vztahů mezi jednotlivými tabulkami. Skript je psaný v jazyce PHP.

Vstup: Na vstupu skript očekává validní XML soubor, tento soubor je specifikován parametrem --input. Pokud je skript spuštěn bez tohoto parametru, čte skript ze standardního vstupu.

Výstup: Na výstup skript tiskne implicitně SQL příkazy na vytvoření tabulek databáze, pokud je zadán parametr skriptu -b, tiskne skript XML strukturu obsahující vztahy mezi jednotlivými tabulkami. Tisk probíhá implicitně na STDOUT, při zadání parametru --output tiskne do souboru jehož název a umístění je součástí parametru.

Parametry skriptu: Skript může být spuštěn celkem s osmi různými parametry. Parametr --help vypíše nápovědu. Parametry --input a --output jsou využívány jako ukazatele z jakého souboru do jakého má skript data zpracovat. Parametr -g slouží pro výběr varianty skriptu, která tiskne vztahy mezi tabulkami. Pokud je zadán parametr -a netisknou se atributy elementů z XML. Parametr --header slouží pro výpis hlavičky na výstup. Parametry --etc a -b jsou určeny k manipulaci se strukturou tabulek, jejich akce jsou detailněji vysvětleny v části „Upravení databáze“.

Implementace

Třídy: Skript obsahuje dvě třídy (database a table). Database má dvě property, property arrayoftables je pole (obsahuje instance třídy table), které slouží k ukládání tabulek a dále má property arguments, což je rovněž pole, které ale v tomto případě slouží k uložení parametrů skriptu pro rychlý přístup k nim. Třída table má tři property, první je name (jedná se o jméno tabulky), druhé je pole primarykeys (slouží k ukládání cizích klíčů, respektive pro ukládání jmen podelementů daného elementu) a třetí je attributes (slouží k ukládání atributů daného elementu). Do polí primarykeys a attributes se ke každé hodnotě ukládá její datový typ.

Hlavní tělo programu: Program nejdříve pomocí několika řádků kódu zpracuje argumenty, pokud nastává chyba je ukončen s chybovou návratovou hodnotou 1. Pokud jsou parametry skriptu validní vytvoří se třída database. Následně se zkontroluje vstupní soubor (funkce fileload, emptyfile), při chybě je vrácena návratová hodnota 2. Pokud nebyl soubor zadán je čten obsah ze STDIN. Funkce fileload ze souboru následně udělá SimpleXMLElement (pomocí funkcí z knihovny SimpleXML). V dalším kroku jsou do databáze nahrány parametry skriptu. Následně probíhá rekursivní zpracovávání SimpleXMLElementu a nahrávání žádoucích složek do připravené databáze. Daná databáze je poté editována pomocí funkcí, které ji upravují podle zadaných argumentů. Skript se následně rozhodne jakým způsobem bude databázi tisknout (podle parametru -g) a podle toho se dává dohromady string, který je následně vytištěn na místo, které si přeje uživatel. Jednotlivé části skriptu jsou popsány v následujících řádcích podrobněji.

Zpracovávání SimpleXMLElementu: Zavolá se funkce recursivegold, ta pomocí funkce children_names zjistí názvy podelementů SimpleXMLElementu a pokud ještě nejsou uloženy

v databázi (jako tabulky), tak je tam uloží. Tato funkce se volá rekurzivně pro každý podelement. Ve funkci recursivegold jsou rovněž pomocí funkce uelements prozkoumány podlementy každého elementu pro každou tabulku v aktuální úrovni XML. Funkce uelements využívá pomocné funkce arraytodb, prkuniq a prkedit k úpravě pole primarykeys. Často totiž dochází k situaci, že se v poli nachází položka item_id, v souboru se nachází položek item více a její název je nutné změnit na item1_id. K atributům a klíčům se rovněž ukládá jejich typ, ten zjišťuje funkce control. Pokud již položka existuje, předchozí datový typ s tím současným porovnává funkce typeenum a následně vrátí ten větší. Pokud element obsahuje text, ukládá se do primarykeys položka value s daným datovým typem.

Upravení databáze: Nejdříve se volá funkce etc_b_correction, ta upravuje databázi podle zadaných parametrů --etc a -b. Při zadání parametru -b se sloupce stejných názvů s odlišným číslem (např. item1_id, item2_id), sloučí do jednoho a to bez čísla pořadí (item_id). Při zadání parametru --etc se počítá počet stejných sloupců s rozdílným číslem, pokud překročí hodnotu danou parametrem --etc, prvky se z dané tabulky smažou a do tabulky na kterou tyto klíče odkazovali se vloží klíč z tabulky ve které se prvky mažou. Následně se volá funkce attcheck, která projde soubor a zjistí, zdali nedochází ke kolizi mezi atributy a cizími klíči, pokud nastává, skript je ukončen s návratovou hodnotou 90.

Tisk SQL příkazů: Při spuštění skriptu bez parametru -g se volá funkce print_database. V této funkci se ověřuje existence parametru --header, pokud existuje vypíše se do stringu \$final zadaná hlavička, následně se pro každou tabulku z database volají tři funkce (add1, add2, add3). Tyto funkce modifikují string \$final (přidávají do něj primární klíč, cizí klíče a atributy) podle zadaných parametrů a podle určitých vazeb v databázi. String je nakonec vytisknut na STDOUT nebo do souboru pomocí funkce output.

Tisk XML souboru: Při spuštění skriptu s parametrem -g se volá funkce xml_print. Ta využívá svých pomocných funkcí velmi podobně jako výše zmíněná funkce print_database a na konci rovněž volá funkci output.

Závěr: Skript byl otestován jak poskytnutými, tak přídavnými testy, k porovnání XML souborů jsem využil nástroj JExamXML a pro porovnání SQL příkazů na tvorbu databázových tabulek software apgdiff. Testování proběhlo na školním serveru Merlin.