Dokumentace úlohy XTD: XML2DDL v Python 3 do IPP 2014/2015
Jméno a příjmení: Ján Jusko
Login: xjusko00

1 Analýza zadání

V druhé úloze do předmětu IPP bylo za úkol vypracovat skript v jazyce Python 3, který analyzuje vstupní soubor ve formátě XML a podle něho vytvoří sadu příkazů v jazyce SQL které vygenerují databázové tabulky.

2 Implementace

Celý skript je obsažený v jediném souboru a je modulován do několik podprogramů které jsou popsány níže v pořadí, jakém se vykonávají. Skript využíva knihovny *sys*, *argparse*, *xml.dom.minidom*, *os*, *os.path*.

2.1 Zpracování argumentů

Argumenty z příkazové řádky spracováva hned na začátku funkce *spracovanie_argumentov* za pomoci importované knihovny *argparse*. Funkce také ověřuje správnost zadaných argumentov a v případě chybových stavů ukončuje skript s příslušnými návratovými hodnotami.

2.2 Zpracování XML souborů

Načtení souboru, který se má analyzovat se řeší přímo ve funkci *main*. Při absenci parametrů input a output se podle zadáni použijou „soubory“ *stdin* nebo *stdout*. Pak se ze vstupního souboru za pomoci funkice *parse()* vytvoří objekt Document reprezentujíci obsah souboru. Daný objekt se pak pošle do funkce *parse_me_gently()* která prochází vstup po jednotlivých uzlech a vytváří strukturu elementů, podelementů a atributů kterou ukládá do slovníka *tab_elements*, který obsahuje vnořené slovníky s atributama a podelementama včetně informací o jejích datových typech.

2.3 Kontrola parametru -etc

Hned po vytvoření struktury *tab_elements* dochází při zadaném parametru *etc=[n]* k volání funkce *etc()* která má za úkol kontrolovat, počet podelementů se stejným jménem. Je-li číslo n menší než počet podelementů se stejným jménem, musí dojít k záměně cizího klíče mezi tabulkami. Parametr *–etc* se nesmí kombinovat s parametrem *-b* protože ten zajistí, že z více podelementů stejného názvu generuje pouze jediný sloupec tabulky.

2.4 Kontrola konfliktů v názvech

Ve struktuře je potřeba taky kontrolovat konflikty v názvech mezi primárními klíčemi, cizími klíčemi a atributami. O to se stará funkce *konflikt()* která sekvenčne porovnáva všechny názvy a v případě konfliktu končí skript s chybovým kódem *90*.

2.5 Parametr -g

Bohužel není v tomto skriptu implementován.

2.6 Výpis tabulek

Ak je zadán parametr *–header*, zapíše se do výstupního souboru nejprve hlavička a pak ve funkci *create_tables()* se vypisují ze struktury *tab_elements()* příslušné SQL tabulky. Tato funkce je docela přímočará a nemůže zde dojít k žádnemu chybovému stavu.

2.7 Rozšíření VAL

Implementace obsahuje taky dobrovolné rozšíření *VAL*, kde se jedná o validaci, zda je možné vložit do vytvořené struktury tabulek ze vstupního souboru vložit i data se souboru zadaného v parametru *–isvalid=filename*. Funkce *validate()* dostáva jako parametry dvě struktury – původní tabulky *tab_elements* a tabulky z které chceme vkládat - *val_tab_elements*. Tyto dvě struktury porovná, zda by bylo možné přidávat elementy a atributy do *tab_elements*. Nejsou-li struktury kompatibilní, skript končí s kódem *91*, jinak se standarně vypisují tabulky.

3 Závěr

Skript xtd.py byl vytvářen na operačním systému *Ubuntu* a řádně otestován sadou oficiálních testů jak na vytvářeném počítači, tak i na školním serveru Merlin s operačním systémem *CentOS*. K porovnávaní výstupů byl použit nástroj *apgdiff 2.4*. Většina testů dopadla úspešne.

3.1 Metriky kódu

1 soubor
363 *LOC*