

1 Zadání

Úkolem skriptu je vyhodnotit správnost zadaného dotazu, který je podobný příkazu `SELECT` jazyka `SQL`, a následně provést analýzu vstupu ve formátu `XML`. Výstupem je `XML` soubor obsahující elementy splňující požadavky dané dotazem. Skript je napsán v jazyce `PYTHON 3.6`, přičemž je možné jej rozčlenit do několika funkčních celků, které budou blíže popsány v následujících částech dokumentace.

2 Zpracování vstupních argumentů

Prvním úkolem skriptu je správně analyzovat vstupní argumenty, jejichž zpracování je implementováno pomocí vstavené funkce `getopts()`, která kontroluje shodu s předem specifikovanou množinou argumentu. Pokud se nenajde žádná shoda, skript vyhodnotí argument jako neplatný a skript je ukončen se zprávou vypsanou na chybový výstup, kde je blíže popsána konkrétní chyba. Taktéž je skript ukončen, pokud nasūtane nesprávná kombinace a nebo se argumenty vyskytují duplicitně. V případě bezproblémového načtení jsou nastaveny příznaky, které určují následující chování skriptu.

3 Ověření správnosti zadaného dotazu

V další fázi se pomocí funkce `queryParse()` ověří syntaktická a sémantická správnost dotazu a současně se naplní atributy třídy `queryClass`, které reprezentují informace ze zadaného dotazu. Funkce `queryParse()` funguje na principu ořezávání řetězce dotazu pomocí regulárního výrazu po logických celcích a vyhodnocování pomocí vhodně zvolených podmínek. Regulární výraz je navrhnutý tak, aby ořezával všechny znaky až po nejbližší posloupnost bílých znaků. Z toho vyplývá, že algoritmus očekává za každým tokenem alespoň jeden bílý znak, naproti tomu se ale algoritmus dokáže vypořádat i s vynecháním bílých znaků před a za relačním operátorem.

Dotaz se dělí na čtyři části, a to `SELECT`, `FROM`, `WHERE`, `LIMIT`, přičemž části `WHERE` a `LIMIT` jsou nepovinné. Nejrozsáhlejší částí dotazu bylo ověření syntaktické a sémantické správnosti podmínky `WHERE`, kde se může vyskytovat libovolný počet terminálů `NOT`. Implementace tohoto pravidla je pomocí `while` cyklu, kde se při každém výskytu terminálu `NOT` neguje hodnota výrazu uložená jako `qWhereNot` v atributu třídy `queryClass`. Za terminálem `NOT` následuje `element/atribut/element.atribut`, přičemž skript vyhodnocuje lexikální správnost pro každou variantu z těchto možností pomocí funkce `isElement()`, založené na regulárním výrazu, který je nastavený podle specifikace identifikátorů `elementů/atributů` jazyka `XML`. Dalším terminálem může být jedna ze čtyř podporovaných operací, za kterou bude následovat číslo nebo řetězec, přičemž operand `CONTAINS` pracuje jen s řetězcem. V případě jakékoliv syntaktické či sémantické chyby, skript vypíše chybové hlášení a ukončí program s návratovou hodnotou 80.

4 Načtení vstupního XML dokumentu

Načtení vstupního dokumentu `XML` je zabezpečené pomocí funkce `parseString()` z knihovny `xml.dom.minidom`. Tato funkce vrátí objekt, nad kterým jsou většinou volané metody `getElementsByTagName()` a `hasAttribute()`.

5 Výběr výstupních elementů

Při výběru výstupních elementů se v prvním řadě najde první `element/atribut`, který se nachází za terminálem `FROM`, a v něm se podle `SELECT element` hledají všechny elementy. V případě upřesnění terminálem `WHERE`, se ještě vyhodnocuje, zda `element` splňuje i tuto podmínku. Úspěšně nalezené elementy se ukládají do řetězce spolu se všemi atributy, hodnotami a podelementy. V případě definování maximálního počtu elementů pomocí `LIMIT` je množina elementů ořezaná právě na tento počet.

Při zápisu do výsledného souboru anebo na standardní výstup, se podle přepínače `-n` doplní resp. nedoplní `XML` hlavička, a taktéž se podle přepínače `-root=element` doplní párový `element`, který bude zaobalovat všechny výsledky dotazu.