



Matlab (9)

Database Toolbox

Databázu môžeme chápať ako nástroj na zbieranie dát a manipuláciu s nimi. Najväčšie zastúpenie majú databázy v informatike a našli si svoje uplatnenie aj v priemysle. Typickým príkladom je archivácia procesných dát (napríklad z PLC) a ich neskoršie využívanie na analýzu, prípadne optimalizáciu výrobného procesu. Jednou z mnohých zaujímavých nastavení Matlabu je aj Database Toolbox [1], [2], ktorý slúži práve na spoluprácu Matlabu a databáz.

Database Toolbox teda dovoľuje vymieňať dáta medzi Matlab workspace a relačnými databázami. Môžeme napríklad načítať dáta z databázy do Matlabu, využiť nástroje Matlabu na analýzu alebo spracovanie dát a neskôr zapísať upravené dáta do tej istej, prípadne inej databázy. Toolbox však nemá za úlohu nahradiť administratívnu časť databázy (tvorba tabuliek a pod.). Na prácu s Database Toolbox-om treba mať samostatný Matlab a databázu, ktorú podporuje toolbox (z najpoužívanejších Microsoft Access, Microsoft SQL Server, MySQL, Oracle a iné). Obsahuje Visual Query Builder (VQB), ktorý je grafickým prostredím na prácu s databázami a umožňuje vytváranie požiadaviek (query) na databázu. Znalosť jazyka SQL je potrebná iba na prácu so zložitými požiadavkami na databázu. Opis SQL presahuje rámec tohto článku, stručne a jednoducho je vysvetlený napríklad na stránke W3schools [3].

Pred samostatným pripojením potrebujeme nastaviť zdroj dát – samostatné dáta, ovládač, adresár, server a podobne. Zdroj dát komunikuje s ODBC (Microsoft Windows rozhranie) alebo JDBC (rozhranie pre aplikácie založené na Java) ovládačmi. Database Toolbox je postavený na Java, preto používa na pripojenie k ODBC ovládaču JDBC/ODBC most. Obr. 1 ilustruje, ako prebieha komunikácia pomocou ovládačov a Database Toolboxu.

Skôr, ako začneme komunikovať, treba si vytvoriť zdroj dát. V tomto článku využijeme webový server WAMP [4], ktorý obsahuje databázu MySQL. Pomocou phpMyAdmin sme si vytvorili používateľa na prístup do databázy a databázu s tabuľkou na archiváciu dát. Aby zdroj dát komunikoval cez JDBC ovládač, treba urobiť niekoľko krokov. Ako prvé musíme pridať cestu k JDBC ovládaču v súbore

`matlabroot/toolbox/local/classpath.txt`

V našom prípade sme si stiahli oficiálny JDBC ovládač pre databázu MySQL [5] Connector/J 5.1 a v spomínanom súbore sme nastavili jeho

umiestnenie. Pri zmene ovládača alebo pri jeho umiestnení v inom adresári treba súbor s cestou prepísať a reštartovať Matlab. JDBC ovládač nastavíme vo Visual Query Builder. Builder spustíme v príkazovom riadku pomocou príkazu

`querybuilder`

a v menu Query otvoríme položku define JDBC Data Source. Vytvoríme si nový zdroj dát JDBC v podobe `.mat` súboru a nastavíme mu parametre, ako sú na obr. 2. Parameter Name určuje meno, ktoré je priradené zdroju dát, ale väčšinou ide o meno databázy. Parameter Driver určuje meno JDBC ovládača a je to v podstate trieda, ktorá implementuje Java SQL ovládač. URL špecifikuje JDBC URL objekt vo forme

`jdbc:subprotocol://hostname:port/databasename`

V našom prípade je meno databázy obsiahnuté v parametri Name, preto ho nepíšeme. Korektné údaje nájdete v manuáli výrobcu konkrétneho ovládača. Spojenie môžeme vyskúšať tlačidlom Test a zadaním prihlasovacieho mena a hesla do databázy. Tlačidlom Add/Update pridáme náš zdroj dát do zoznamu a môžeme s ním pracovať po potvrdení tlačidlom OK.

Práca s Database Toolboxom sa dá rozdeliť na dve časti. Importovanie dát z databázy



Obr.2 Nastavenie JDBC zdroja dát



Obr.3 Vloženie dát do databázy

do Matlab workspace a exportovanie dát z Matlab workspace do databázy. Export dát z databázy spravíme v nasledujúcich krokoch:

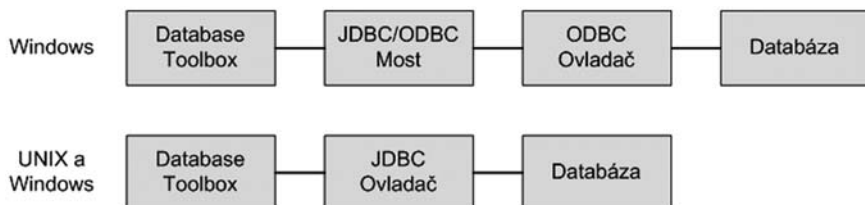
- konštrukcia query na export do zdroja dát,
- špecifikácia premennej, ktorá obsahuje dáta na exportovanie,
- spustenie query.

V našom prípade chceme vložiť do databázy dáta zo simulácie, ako je žiadaná hodnota, regulačná odchýlka, akčný zásah, výstup z procesu a čas (samozrejme v databáze už máme tabuľku, do ktorej môžeme tieto dáta uložiť). Vo Visual Query Builder označíme ako operáciu s dátami (Data operation) vloženie dát (Insert). Ako zdroj dát (Data source) zvolíme našu databázu, ako tabuľku vyberieme tabuľku s našimi dátami a označíme položky, ktoré chceme vložiť. Vyberieme premennú z Matlab workspace, ktorú chceme exportovať, v našom prípade premennú `export`. Táto premenná obsahuje vo svojich stĺpcoch už spomínané veličiny, je typu pole buniek (cellarray). Po označení všetkých položiek Visual Query Builder vytvorí príkaz na exportovanie dát, ako je to na obr. 3.

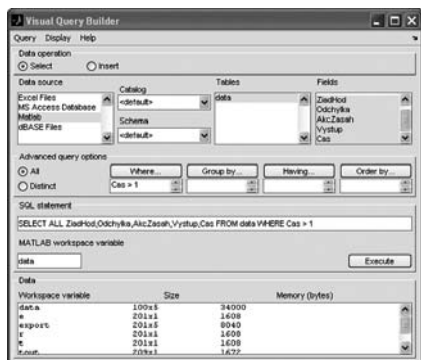
Podobne, ako vkladanie dát z databázy, možno pomocou Database Toolboxu aj čítať. Čítanie dát vykonáme v týchto krokoch:

- konštrukcia query na import zo zdroja dát,
- definovanie premennej, do ktorej sa dáta uložia,
- spustenie query,
- úprava a manipulácia s dátami.

V našom prípade prečítame časť dát, pričom použijeme jednoduchú podmienku. Vo Visual Query Builder označíme ako operáciu s dátami (Data operation) čítanie dát (Select). Ako zdroj dát (Data source) zvolíme našu databázu



Obr.1 Komunikácia medzi databázou a Database Toolboxom



Obr.4 Čítanie dát z databázy

zu, ako tabuľku vyberieme tabuľku s našimi dátami a označíme položky, ktoré chceme čítať. Do políčka MATLAB workspace variable napíšeme premennú, do ktorej chceme uložiť dáta z databázy. Z databázy chceme získať iba časť dát (napríklad dáta po prvej sekunde simulácie), preto využijeme pokročilejšie nastavenie query. Vyberieme položku Where, v ktorej zadáme podmienku, aby bol Cas väčší ako jedna sekunda. Samozrejme Visual Query Builder dovoľuje vytvárať aj zložitejšie požiadavky na databázu. Po označení všetkých položiek Visual Query Builder vytvorí príkaz na čítanie dát, ako je to na obr. 4.

Na urýchlenie práce obsahuje Visual Query Builder v menu položku, pomocou ktorej môžeme vytvorené query uložiť (Query – Save) alebo načítať (Query – Load). Visual Query Builder obsahuje ďalej nástroj na zobrazovanie výsledkov query, ako grafy alebo reporty. Na väčšinu práce s Database Toolbox stačí Vi-

sual Query Builder. Neobsahuje však funkcie na úpravu dát v databáze z workspace a ukladanie binárnych dát. Všetky funkcie, ktoré podporuje Visual Query Builder, sa dajú, samozrejme, vykonávať aj v príkazovom riadku Matlabu.

Príkazový riadok podporuje ďalšie funkcie, ako už spomínanú úpravu dát, prípadne dynamické vykonávanie SQL požiadaviek na databázu. Prácu pomocou príkazového riadku si môžeme ukázať na nasledujúcom príklade. Príkazom database sa pripojíme na našu databázu, pričom ďalšie parametre sú prihlasovacie meno, heslo a objekt spojenia je definovaný premennou conn. Ďalším príkazom si môžeme overiť úspešnosť spojenia s databázou. Príkaz exec vykoná požiadavku na databázu, vráti objekt kurzora výsledku a setdbprefs nastaví konvertovanie návratových hodnôt, ktoré sú typu string na formát, ktorý string podporuje, v tomto prípade na cellarray. Fetch vyberie maximálny počet položiek výsledku na zobrazenie. Posledné príkazy ukončia kurzor a spojenie s databázou.

```
conn = database('Matlab', 'meno', 'heslo')
ping(conn)
curs = exec(conn, 'select AkcZasah from data')
setdbprefs('DataReturnFormat','cellarray')
curs = fetch(curs, 10)
close(curs)
close(conn)
```

Databázy sú dnes nástrojom na efektívnu prácu s dátami. Pod prácou sa rozumie archívacia a manipulácia s dátami. Neobsahujú však

nástroje na analýzu dát. Matlab Database Toolbox môžeme chápať ako rozhranie medzi databázou a Matlabom. Toolbox umožňuje načítavanie, prípadne zápis dát do databázy a Matlab poskytuje množstvo príkazov na ich analýzu. Database Toolbox obsahuje Visual Query Builder (jednoduché grafické prostredie), ktorý ponúka dostatočné množstvo funkcií na tvorbu jednoduchých, ale aj komplikovaných požiadaviek na databázu.

Prípadné otázky k tomuto článku, alebo k programovému systému Matlab, môžete kľásť na diskusnom fóre www.matlab.sk.

Článok vznikol s podporou APVV-99-045805.

Literatúra

- [1] Database Toolbox, Getting Started Guide, The MathWorks
- [2] Database Toolbox, User's Guide, The MathWorks
- [3] www.w3schools.com
- [4] www.wampserver.com
- [5] <http://dev.mysql.com/downloads/connector/j/5.1.html>

Ing. Michal Blaho
Ing. Martin Foltin, PhD.

29

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3, 812 19 Bratislava
e-mail: michal.blaho@syprin.sk
martin.foltin@syprin.sk