

SQLGate – popis protokolu

SQLGate je program vyvinutý pre interné potreby Tangram a.s. Jeho použitie na iné účely je bez akejkoľvek záruky funkčnosti a budúcich zmien. Úlohou SQLGate je vytvoriť pripojenie do databázy ORACLE pre programy a zariadenia, ktoré sa nedokážu pripojiť do ORACLE štandardným spôsobom.

SQLGate otvorí TCP socket server na porte 2400 (možno zmeniť v konfigurácii). Klient sa musí pripojiť na TCP socket. Komunikáciu začína vždy klient spôsobom dotaz – odpoveď. Každý paket je textový reťazec, žiadne binárne znaky. Na jeden SQLGate sa môže pripojiť ľubovoľný počet klientov, ale z jednej IP adresy vždy len jeden.

Obmedzenia:

- datová časť paketov je kodovaná vo Windows 1250, takže niektoré národné znaky nie sú podporované
- oddelovaciu položku v zázname je znak | (ASCII 182), a tento znak nesmie obsahovať žiadnu položku
- Sqlgate klient z jednej IP je schopný vytvoriť len jedno pripojenie na SqlGate server

Priказы a odpovede sú odosielané v paketoch. Štruktúra paketu je nasledovná:

xxxx:prikaz data

xxxx - je hexa číslo, vždy na 4 cifry, ktoré určujú počet znakov v pakete za znakom „:“

: - znak ':'

prikaz - hlavička paketu, pre odosielané pakety to je prikaz, pre prijaté pakety to je typ odpovede

_ - znak ' ' (medzera, ASCII<32>)

data - parametre prikazu alebo odpovede

| Príkaz + parametre | Funkcia, očakávaná odpoveď |
|---------------------------|--|
| CONNECT User Password | Vytvorí nové pripojenie k databáze (User a Password na prístup do Oracle). Ak už bolo nejaké pripojenie vytvorené, tak bude korektné zrušené a všetky resultsety budú uvoľnené. OK session_handle (6 ASCII znakov) |
| RECONNECT Session_handle | Pokus o obnovenie už vytvoreného pripojenia príkazom CONNECT. Všetky resultsety ostávajú funkčné. Tento príkaz je užitočný, ak dôjde k prerušeniu komunikácie medzi klientom a SQLGate, ale spojenie medzi SQLGate a ORACLE je funkčné. OK session_handle |
| DISCONNECT Session_handle | Zrušenie pripojenia k databáze, všetky resultsety sa uvoľnia. OK session_handle |
| QUERYX SELECT ... | Vykoná SELECT a pripraví zoznam riadkov (resultset) typu Forward_only OK row_handle, vráti hexa číslo určujúce handle pre prechádzanie resultsetu |
| QUERYXN SELECT ... | To isté ako QUERYX, ale resultset sa dá prechádzať v ľub. poradí (Scrollable). Nepoužívajte na rozsiahle resultsety s tisícami riadkov, vyžaduje pamäť. OK row_handle počet_riadkov, okrem handle vráti aj počet riadkov (hexa) |
| FETCH row_handle [nnn] | Načítanie ďalšieho riadka z resultsetu. Pre Scrollable resultset (QUERYXN) sa musí zadať aj číslo riadka nnn (začínajúce od 1). Pre QUERYX musí byť nnn=0. ROW rowdata (stĺpce oddelené znakom - ASCII 182) OK EOF, ak už nie je ďalší riadok k dispozícii |
| CLOSE row_handle | Zatvorenie resultsetu, vytvoreného QUERYX alebo QUERYXN, uvoľní sa pamäť OK row_handle |
| EXEC SQL command | Vykonanie ľubovoľného SQL príkazu v databáze OK EOF, ak sql príkaz vrátil prázdny resultset OK nnnn, ak sql príkaz bol update, nnnnnnnn je hexa číslo určujúce počet updatovaných záznamov ROW rowdata, ak sql príkaz vrátil nejaký resultset, rowdata obsahuje údaje prvého záznamu z resultsetu. Jednotlivé položky záznamu sú oddelené znakom ASCII<182> |

Pre každý príkaz môže byť vrátená odpoveď ERROR text, kde text je podrobný popis chyby. Môžu to byť chyby vzniknuté na SQLGate serveri, alebo chyby vrátené od ORACLE.

Príklad

```
CONNECT user password
QUERYX SELECT Xx,Nazov FROM TOVARY ORDER BY Xx
FETCH hhh 0
FETCH hhh 0
. . . atď, pokiaľ SQLGate vracia ROW
FETCH hhh 0
CLOSE hhh
. . . tu môžu nasledovať ďalšie QUERYx alebo EXEC
EXEC INSERT INTO nejaka_tabulka (stlpce) VALUES (hodnoty)
. . .
DISCONNECT session_z_connectu
```

Nezabudnite, že pred každým vyslaným aj prijatým paketom je ešte dĺžka paketu v hexa-tvare „nnnn:“ !