

SQL_BINDIN

Akcia SQL_BINDIN

Funkcia

Akcia špecifikuje hodnoty parametrov a vykoná SQL príkaz **SELECT** pripravený akciou [SQL_PREPARE](#), pokiaľ v tejto akcii bola použitá [parametrizácia](#) a kúové slovo **BINDOUT**.

Deklarácia

```
SQL_BINDIN handleIdent_Int, retCodeIdent_Int, _Par1, _Par2, ...  
  
SQL_BINDIN handleIdent_Int, retCodeIdent_Int, _VarRowIdent
```

Parametre

handleIdent_Int	in	Identifikátor typu <i>Int</i> - jednoznačné číslo (handle) spojenia s databázou.
retCodeIdent_Int	out	Identifikátor typu <i>Int</i> - návratový kód.
_Par1, _Par2, ...	in	Zoznam objektov, konštánt alebo lokálnych premenných , ktoré sa použijú na mieste parametrov parametrizovaného SQL príkazu SELECT .
_VarRowIdent	in	Odkaz na riadok lokálnej premennej typu <i>Record</i> alebo na riadok štruktúrovanej premennej . Hodnoty z tohto riadku sa použijú na mieste parametrov parametrizovaného SQL príkazu SELECT .

Návratový kód

Hodnota parametra *retCodeIdent_Int* - pozri tabuku [chybových stavov](#). Je možné získať [rozšírenú informáciu o chybe](#).

Popis

Ďalšie údaje SQL príkazom **SELECT** je implementované v dvoch alebo troch fázach. Prvú (prípravnú) fázu vykoná akcia [SQL_PREPARE](#). Nad databázou pripraví (a pokiaľ nie je použité kúové slovo **BINDOUT**, aj vykoná) príkaz **SELECT**. Ak bolo použité kúové slovo **BINDOUT** znamená, že SQL príkaz bol [parametrizovaný](#), je potrebná druhá fáza - volaním akcie **SQL_BINDIN** je nutné nastaviť hodnoty vstupných parametrov a následne výraz vykonať. Posledná fáza je postupné čítanie riadkov, ktoré pripravil príkaz **SELECT** akciou [SQL_FETCH](#).

Poznámka: Použitím [parametrizácie](#) je možné ušetriť prácu SQL databáze, pretože príprava (kompilácia) parametrizovaného SQL výrazu sa vykoná iba raz (v rámci akcie [SQL_PREPARE](#)). Následne je nutné nastaviť parametre akciou **SQL_BINDIN** (ktorá SQL príkaz aj vykoná) a jeden alebo viackrát volať [SQL_FETCH](#) na získanie výsledkov. Potom je možné nastaviť nové hodnoty parametrov a znovu vykonať SQL príkaz opätovným volaním [SQL_BINDIN](#) a získať nové výsledky volaním [SQL_FETCH](#). Vhodným nastavením parametrov databázy (napr. Oracle: *session_cached_cursors*) je možné zabezpečiť recyklovanie kurzorov (skompilovaných príkazov) medzi volaniami **SQL_PREPARE**.

Príklad

[Príklad práce s databázou \(akcie SQL_ ...\)](#).

```
INT _handle      ; handle to database
INT _retCode     ; return code
TEXT _name       ; product name
TEXT _type       ; product type
                ; parametrized SQL command
TEXT _sql =      "SELECT Name, Type FROM Products WHERE ID>= #PAR# AND ID<=
#PAR#"

SQL_CONNECT MyDatabase, _handle, _retCode
SQL_PREPARE _handle, _retCode, _sql BINDOUT _name, _type
SQL_BINDIN  _handle, _retCode, 1, 100 ; read all products between 1 and
100

DO_LOOP
    SQL_FETCH _handle, _retCode
    EXIT_LOOP _retCode # _ERR_NO_ERROR
    ; data processing goes here
END_LOOP

SQL_FREE _handle
SQL_DISCONNECT _handle
```

Súvisiace odkazy

[DB_TRANS_OPEN](#)
[DB_TRANS_COMMIT](#)
[DB_TRANS_ROLLBACK](#)
[DB_TRANS_CLOSE](#)

[SQL_CONNECT](#)
[SQL_DISCONNECT](#)
[SQL_EXEC_DIRECT](#)
[SQL_EXEC_PROC](#)

[SQL_PREPARE](#)
[SQL_FETCH](#)
[SQL_FREE](#)

[SQL_SELECT](#)

[Všetky databázové akcie](#)



Súvisiace stránky:

[Akcie v skriptoch](#)