SQL_PREPARE

Akcia SQL_PREPARE

Funkcia

Akcia pripraví vykonávanie SQL príkazu **SELECT**.

Deklarácia

```
 SQL_PREPARE handleIdent_Int, retCodeIdent_Int, selectStringExpr BIND
_locVar1, _locVar2, ...


 SQL_PREPARE handleIdent_Int, retCodeIdent_Int, selectStringExpr BIND
_locVarRowIdent


 SQL_PREPARE handleIdent_Int, retCodeIdent_Int, selectStringExpr BIND
_locVarRecordIdent


 SQL_PREPARE handleIdent_Int, retCodeIdent_Int, selectStringExpr BINDOUT
_locVar1, _locVar2, ...


 SQL_PREPARE handleIdent_Int, retCodeIdent_Int, selectStringExpr BINDOUT
_locVarRowIdent


 SQL_PREPARE handleIdent_Int, retCodeIdent_Int, selectStringExpr BINDOUT
_locVarRecordIdent
```

Parametre

handleIdent_Int	in	Identifikátor typy *Int* - jednoznané íslo (handle) spojenia s databázou.
retCodeIdent_Int	out	Identifikátor typu *Int* - návratový kód.
selectStringExpr	in	Výraz typu *Text.*
_locVar1, _locVar2, ...	in	Zoznam lokálnych premenných.
_locVarRowIdent	in	Odkaz na riadok lokálnej premennej typu *Record.*
_locVarRecordIdent	in	Identifikátor lokálnej premennej typu *Record.*

Návratový kód

Hodnota parametra *retCodeIdent_Int* - pozri tabuku chybových stavov. Je možné získa rozšírenú informáciu o chybe.

Popis

ítanie databázy SQL príkazom **SELECT** je implementované v dvoch alebo troch fázach.
Prvú (prípravnú) fázu vykoná akcia **SQL_PREPARE**. Nad databázou (je daná hodnotou identifikátora *handleIdent_Int*) pripraví (a pokia nie je použité kúové slovo **BINDOUT**, aj vykoná) príkaz **SELECT** definovaný hodnotou výrazu *selectStringExpr*. Úspešnos akcie je indikovaná hodnotou *retCodeIdent_Int*.
Druhá fáza je potrebná vtedy, ke bolo použité kúové slovo **BINDOUT**. Znamená to, že vo výraze *selectStringExpr* bola použitá parametrizácia a je nutné volaním SQL_BINDIN nastavi hodnoty vstupných parametrov výrazu *selectStringExpr* a následne výraz vykona.
Posledná fáza je postupné ítanie riadkov, ktoré pripravil príkaz **SELECT** akciou SQL_FETCH.

ítanie prebieha zásadne do lokálnych premenných, ktoré sú uvedené za kúovým slovom **BIND** alebo **BINDOUT** akcie **SQL_PREPARE**.
Tu sú tri možné **varianty**:

1. **Zoznam neštruktúrovaných lokálnych premenných.**
   ítanie prebieha po jednom riadku do lokálnych premenných, ktoré sú vymenované za kúovým slovom **BIND** alebo **BINDOUT**.

2. **Odkaz na jeden riadok lokálnej premennej typu *Record.***
   ítanie prebieha po jednom riadku do jedného riadku lokálnej štruktúrovanej premennej. Štruktúra ítaných dát musí by zhodná so štruktúrou lokálnej premennej.

3. **Odkaz na lokálnu premennú typu *Record*.**
   ítanie prebieha po jednom alebo viacerých riadkov lokálnej premennej. Tejto je poda potreby pred naplnením výsledku ítania zmenený rozmer. Štruktúra ítaných dát musí by zhodná zo štruktúrou lokálnej premennej.

Pre jedno handle (získané akciou SQL_CONNECT) môže by súasne aktívne len jedno ítanie. Akcia **SQL_PREPARE** zruší platnos predchádzajúcej akcie. Explicitne ukoni ítanie je možné pomocou akcie SQL_FREE.

Poznámka: použitím parametrizácie je možné uahi prácu SQL databáze, pretože príprava (kompilácia) parametrizovaného SQL výrazu sa vykoná iba raz (v rámci akcie **SQL_PREPARE**). Následne je nutné nastavi parametre akciou SQL_BINDIN (ktorá SQL príkaz aj vykoná) a jeden alebo viackrát vola SQL_FETCH na získanie výsledkov. Potom je možné nastavi nové hodnoty parametrov a znovu vykona SQL príkaz opätovným volaním SQL_BINDIN a získa nové výsledky volaním SQL_FETCH.
Vhodným nastavením parametrov databázy (napr. Oracle: *session_cached_cursors*) je možné zabezpei recyklovanie kurzorov (skompilovaných príkazov) medzi volaniami **SQL_PREPARE**.

Príklad

Príklad práce s databázou (akcie SQL_ ...).

```
 BOOL _useBinding = @TRUE ; use parametrized SQL command
 INT  _handle    ; handle to database
 INT  _retCode   ; return code
 TEXT _name      ; product name
 TEXT _type      ; product type

                 ; parametrized SQL command
 TEXT _sqlPar =  "SELECT Name, Type FROM Products WHERE ID>= #PAR# AND
ID<= #PAR#"

                 ; non-parametrized SQL command
 TEXT _sqlNpar =  "SELECT Name, Type FROM Products WHERE ID>= 1 AND ID<=
100"

 SQL_CONNECT MyDatabase, _handle, _retCode

 IF _useBinding THEN ; parametrized alternative
   SQL_PREPARE _handle, _retCode, _sqlPar BINDOUT _name, _type
   SQL_BINDIN  _handle, _retCode, 1, 100 ; read all products between 1 and
100
 ELSE ; non-parametrized alternative
   SQL_PREPARE _handle, _retCode, _sqlNpar BIND _name, _type
 ENDIF

 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_BINDIN
SQL_FETCH
SQL_FREE

SQL_SELECT

Všetky databázové akcie

Súvisiace stránky:

Akcie v skriptoch