SQL_SELECT

Akcia SQL_SELECT

Funkcia

Akcia vykoná SQL príkaz SELECT.

Deklarácia

```
SQL_SELECT dbObjIdent, retCodeIdent_Int, [maxRowsIdent_Int],
selectIdent_String [TRANS transHandle_Int]  BIND ...
```

alebo

```
SQL_SELECT connectString, retCodeIdent_Int, [maxRowsIdent_Int],
selectIdent_String  ON dbManIdent  BIND ...
```

priom za kúovým slovom BIND sa musí nachádza jedna z možností:
BIND _locVar1, _locVar2, ...
BIND _locVarRowIdent
BIND _locVarRecordIdent

Parametre

dbObjIdent	in	Odkaz na objekt typu Databáza alebo Tabuka.
connectString	in	Identifikátor typu Text, alebo textová konštanta obsahujúca konektovací string na databázu.
retCodeIdent_Int	out	Identifikátor typu *Int*- návratový kód.
maxRowsIdent_Int	in	Identifikátor typu *Int* - poet naraz naítaných riadkov (ak nie je uvedený, náhradná hodnota je 1).
selectIdent_String	in	Identifikátor typu *String*.
_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

Akcia SQL_SELECT vznikla za úelom o najjednoduchšieho vykonania SQL príkazu SELECT. Vo svojej funknosti združuje akcie SQL_CONNECT, SQL_PREPARE, SQL_FETCH a SQL_DISCONNECT.

Databázu, v rámci ktorej sa SQL príkaz SELECT vykoná je možné uri:

1. Parametrom *dbObjIdent* - tento predstavuje objekt typu *Databáza* alebo *Tabuka*. V tomto prípade je možné zada (nepovinný) parameter *transHandle_Int*, ktorý definuje databázové spojenie (hodnota parametra sa získala predchádzajúcim volaním akcie DB_TRANS_OPEN), v rámci ktorého sa SQL príkaz vykoná.
   Ak parameter *transHandle_Int* nie je definovaný, SQL príkaz bude vykonaný na niektorom z preddefinovaných databázových pripojení.
2. Parametrom *connectString* - ktorý je bližšie popísaný v akcii SQL_CONNECT. V tomto prípade je potrebné špecifikova proces DB Manager, ktorý príslušný SQL príkaz vykoná.

Vlastný príkaz SELECT je definovaný hodnotou parametra *selectIdent_String*. Poda potu stpcov príkazu SELECT je potrebné vybra vhodnú variantu zverejnenia jeho výsledkov prostredníctvom povinného kúového slova BIND. Význam a interpretácia jednotlivých variant je popísaný v rámci akcie SQL_PREPARE.

Nepovinný identifikátor *maxRowsIdent_Int* definuje poet riadkov výberu, ktorý bude zverejnený. Ak nie je uvedený, jeho hodnota je 1. Poet riadkov, ktorý je možné preíta, závisí aj od použitého variantu BIND. Príslušný popis obsahuje akcia SQL_FETCH.

Maximálny poet vrátených riadkov je obmedzený konfiguraným parametrom databázy Maximum vrátených riadkov. Pri prekroení tohoto potu riadkov bude do lokálnej štruktúry *_locVarRecordIdent* naítaných prvých Maximum vrátených riadkov a v *retCodeIdent_Int* sa vráti chyba _ERR_DATABASE_ROWS_LIMIT.

Poznámka

Hodnoty, ktoré vracia SQL príkaz, sú vložené do lokálnych premenných poda poradia (1. hodnota do 1. bindovanej premennej, 2. do druhej at). Toto platí aj pri vkladaní do lokálnej štruktúrovanej premennej (t. j. nepozerá sa na názvy stpcov). Potenciálne nebezpené sú SELECT-y typu "SELECT *" - pokia sa poradie stpcov v databáze líši od poradia stpcov v štruktúrovanej premennej, hodnoty budú vložené do nesprávnych stpcov. Preto odporúame explicitné vymenovanie stpcov v SQL dotaze (SELECT COL1, COL2, COL3 ...).

Ak sú naítavané dáta z jedinej tabuky alebo pohadu, namiesto SQL_SELECT použite akciu DBS_READ.

Príklad

Príklad predpokladá existenciu objektu typu *Databáza* s menom **gnat_test**. V rámci databázy existuje tabuka OBJLIST, v ktorej je stpec id íselného typu.
Procedúra Demo1 predstavuje rôzne (nie všetky) varianty akcie SQL_SELECT.
Procedúry Demo2 a Demo3 plnia tú istú funkciu. Demo2 je implementovaná prostredníctvom akcie SQL_SELECT a Demo3 je implementovaná akciami SQL_CONNECT, SQL_PREPARE, SQL_FETCH a SQL_DISCONNECT.

```
 PROCEDURE Demo1

  INT  _retCod
  INT _maxId

   ; výber prostredníctvom objektu typu databáza
  SQL_SELECT gnat_test, _retCode,,  "select max(id) from OBJLIST" BIND
_maxId
  IF _retCode = _ERR_NO_ERROR THEN
  ; premenná _maxId je naplnená
   ELSE
  ; chyba
  ENDIF

  ; výber prostredníctvom objektu typu tabuka
  SQL_SELECT DB.OBJLIST, _retCode,,  "select max(id) from OBJLIST" BIND
_maxId
  IF _retCode = _ERR_NO_ERROR  THEN
  ; premenná _maxId je naplnená
   ELSE
  ; chyba
  ENDIF

  INT _transHandle
   ; vytvorenie transakcie prostredníctvom objektu typu databáza
  DB_TRANS_OPEN gnat_test, _transHandle, _retCode
  IF _retCode = _ERR_NO_ERROR  THEN
    ; transakcia otvorená
    ; výber v rámci transakcie
    SQL_SELECT DB.OBJLIST, _retCode,, "select max(id) from OBJLIST"  TRANS
_transHandle BIND _maxId
    IF _retCode = _ERR_NO_ERROR THEN
    ; premenná _maxId je naplnená
    ELSE
    ; chyba
    ENDIF

    DB_TRANS_CLOSE _transHandle
  ELSE
    ; chyba
   ENDIF

  ; výber s pripojením
  SQL_SELECT "connection String", _retCode,, "select max(id) from OBJLIST"
ON SELF.DBM BIND _maxId
  IF _retCode = _ERR_NO_ERROR THEN
  ; premenná _maxId je naplnená
  ELSE
  ; chyba
  ENDIF

 END Demo1


 PROCEDURE Demo2(INT _maxId)
```

```
 INT _retCode

  _maxId := %SetInvalid(0)
  ; výber prostredníctvom objektu typu databáza
  SQL_SELECT gnat_test, _retCode,, "select max(id) from OBJLIST" BIND
_maxId
  IF _retCode = _ERR_NO_ERROR THEN
  ; premenná _maxId je naplnená
  ELSE
  ; chyba
  ENDIF

  END Demo2


 PROCEDURE Demo3(INT _maxId)
  INT _retCode
  INT _handle
  _maxId := %SetInvalid(0)
  ;
  SQL_CONNECT gnat_test, _handle, _retCode
  IF _retCode <> _ERR_NO_ERROR THEN
   RETURN ; chyba
  ENDIF

  SQL_PREPARE _handle, _retCode, "select max(id) from OBJLIST" BIND _maxId
  IF _retCode <> _ERR_NO_ERROR THEN
   RETURN ; chyba
  ENDIF

  SQL_FETCH _handle, _retCode
  IF _retCode <> _ERR_NO_ERROR THEN
   RETURN ; chyba
  ENDIF

  SQL_DISCONNECT _handle

 END Demo3


 BEGIN

  CALL Demo1

  INT _max
  CALL Demo2(_max)

  CALL Demo3(_max)
 END
```

Súvisiace odkazy

DB_TRANS_OPEN
DB_TRANS_COMMIT
DB_TRANS_ROLLBACK
DB_TRANS_CLOSE

SQL_CONNECT
SQL_EXEC_DIRECT
SQL_EXEC_PROC

SQL_PREPARE
SQL_BINDIN
SQL_FETCH
SQL_FREE

Všetky databázové akcie

Súvisiace stránky:

Akcie v skriptoch