

Príklad práce s databázou (akcie SQL_ ...)

V příklade sa predpokladá existencia objektu [SD.StrojVyrabal](#) a je definované DSN na databázu *Database* potrebnej štruktúry. alej musí by spustený proces [D2000 DBManager](#) s menom *SELF.DBM*.

Štruktúra databázy:

Tabuky:

| Meno tabuky | Štruktúra tabuky | | |
|--------------|------------------|----------|----------------------------------|
| Stroje | Meno stpca | Typ | Popis |
| | ID | Integer | Jednoznany identifikátor stroja |
| | Meno | Char(50) | Meno stroja |
| Vyrobyky | Meno stpca | Typ | Popis |
| | ID | Integer | Jednoznany identifikátor výrobku |
| | Meno | Char(50) | Meno výrobku |
| StrojVyrabal | Meno stpca | Typ | Popis |
| | ID_STROJ | Integer | Jednoznany identifikátor stroja |
| | ID_VYROBOK | Integer | Jednoznany identifikátor výrobku |

V tabuke **Stroje** je evidencia strojov, priom stroj je reprezentovaný jednoznanným identifikátorom *Stroj.ID* a ubovolným menom *Stroj.Meno*. V tabuke **Vyrobyky** je evidencia druhov výrobkov, priom výrobok je reprezentovaný jednoznanným identifikátorom *Vyrobok.ID* a ubovolným menom *Vyrobok.Meno*. V tabuke **StrojVyrabal** je evidencia o tom, že stroj *StrojVyrabal.ID_STROJ* vyrábala výrobok *StrojVyrabal.ID_VYROBOK*.

Príklad najprv naplní tabuky **Stroj** a **Vyrobok**. Potom náhodne generuje záznamy o vyrábání výrobkov na strojoch.

Následne zisuje 3-mi rôznymi spôsobmi, aké výrobky a množstvá sa vyrobili na stroji 1.

```
INT _handle      ; handle na databázu
INT _retCode     ; návratový kód

; Procedúra vykoná SQL príkaz. Neúspech zaznamená do Monitorovacej databázy
PROCEDURE ExecSql(IN TEXT _sql)
    TEXT _errorMsg
    SQL_EXEC_DIRECT _handle, _retCode, _sql
    IF _retCode # _ERR_NO_ERROR THEN
        _errorMsg := &"Error SQL_EXEC_DIRECT "; + _sql
        LOG _errorMsg PRIORITY _LOG_PRTY_ERROR
    ENDIF
END ExecSql

; Vloží do tabuky definíciu stroja
PROCEDURE Vloz_Stroj(IN INT _id, IN TEXT _meno)
    TEXT _sql
    _sql := "INSERT INTO Stroje VALUES (" + %IToStr(_id) + ", '" + _meno + "'"
    CALL ExecSql(_sql)
END Vloz_Stroj

; Vloží do tabuky definíciu výrobku
PROCEDURE Vloz_Vyrobok(IN INT _id, IN TEXT _meno)
    TEXT _sql
    _sql := "INSERT INTO Vyrobyky VALUES (" + %IToStr(_id) + ", '" + _meno + "'"
    CALL ExecSql(_sql)
END Vloz_Vyrobok

; Naplním tabuku Strojov
PROCEDURE Napln_Stroje
    INT _id
    TEXT _meno
    TEXT _sql

    _sql := "DELETE FROM STROJE"
    CALL ExecSql(_sql)
```

```

_id := 1
_meno := "Stroj 1"
CALL Vloz_Stroj(_id, _meno)

_id := 2
_meno := "Stroj 2"
CALL Vloz_Stroj(_id, _meno)

_id := 3
_meno := "Stroj 3"
CALL Vloz_Stroj(_id, _meno)

_id := 4
_meno := "Stroj 4"
CALL Vloz_Stroj(_id, _meno)

_id := 5
_meno := "Stroj 5"
CALL Vloz_Stroj(_id, _meno)
END Napln_Stroje

; Naplním tabuku Výrobkov
PROCEDURE Napln_Vyrobky
  INT _id
  TEXT _meno
  TEXT _sql

  _sql := "DELETE FROM VYROBKY"
  CALL ExecSql(_sql)

  _id := 1
  _meno := "Vyrobo 1"
  CALL Vloz_Vyrobo(_id, _meno)

  _id := 2
  _meno := "Vyrobo 2"
  CALL Vloz_Vyrobo(_id, _meno)

  _id := 3
  _meno := "Vyrobo 3"
  CALL Vloz_Vyrobo(_id, _meno)

  _id := 4
  _meno := "Vyrobo 4"
  CALL Vloz_Vyrobo(_id, _meno)

  _id := 5
  _meno := "Vyrobo 5"
  CALL Vloz_Vyrobo(_id, _meno)
END Napln_Vyrobky

; Vloží do tabuky záznam o tom, že stroj vyráběl výrobek
PROCEDURE StrojVyrabal(IN INT _idStroj, IN INT _idVyrobo)
  TEXT _sql
  _sql := "INSERT INTO StrojVyrabal VALUES (" + %IToStr(_idStroj) + ", " + %IToStr(_idVyrobo) + ")"
  CALL ExecSql(_sql)
END StrojVyrabal

; Procedúra generuje náhodné záznamy o tom,
; že stroj vyráběl výrobek
PROCEDURE Napln_StrojVyrabal
  INT _idStroj
  INT _idVyrobo
  INT _idx
  TEXT _sql

  _sql := "DELETE FROM StrojVyrabal"
  CALL ExecSql(_sql)

```

```

_idx := 1
DO_LOOP
    EXIT_LOOP _idx = 100
    _idStroj := %Rnd() * 4.0 + 1.0
    _idVyrobo := %Rnd() * 4.0 + 1.0
    CALL StrojVyrabal(_idStroj, _idVyrobo)
    _idx := _idx + 1
END_LOOP
END Napln_StrojVyrabal

; Zoznam výrobkov a ich počet pre stroj s menom "Stroj 1"
; prostredníctvom SQL_PREPARE - verzia 1 alebo 4
PROCEDURE Stroj_1_LocVarList
    BOOL _useBinding = @TRUE ; pouzit SQL_BINDIN?
    TEXT _sql
    TEXT _errorMsg
    TEXT _vyrobo
    INT _pocet

    IF _useBinding THEN ; alternativa s SQL_BINDIN (verzia 4)

        _sql := "SELECT Vyroby.Meno, Count(*) FROM Vyroby, Stroje, StrojVyrabal WHERE Stroje.Meno = #PAR# "
        _sql := _sql + "AND Stroje.ID=StrojVyrabal.ID_STROJ AND Vyroby.ID=StrojVyrabal.ID_VYROBO GROUP BY Vyroby.
Meno"
        SQL_PREPARE _handle, _retCode, _sql BINDOUT _vyrobo, _pocet
        SQL_BINDIN _handle, _retCode, "Stroj 1" ; nastav hodnotu vstupneho parametra

    ELSE ; alternativa bez SQL_BINDIN (verzia 1)

        _sql := "SELECT Vyroby.Meno, Count(*) FROM Vyroby, Stroje, StrojVyrabal WHERE Stroje.Meno = 'Stroj 1' "
        _sql := _sql + "AND Stroje.ID=StrojVyrabal.ID_STROJ AND Vyroby.ID=StrojVyrabal.ID_VYROBO GROUP BY Vyroby.
Meno"
        SQL_PREPARE _handle, _retCode, _sql BIND _vyrobo, _pocet

    ENDIF

    IF _retCode # _ERR_NO_ERROR THEN
        _errorMsg := "Error SQL_PREPARE " + _sql
        LOG _errorMsg PRIORITY _LOG_PRTY_ERROR
        RETURN
    ENDIF

    DO_LOOP
        SQL_FETCH _handle, _retCode ; načítanie 1 riadku výsledku SQL príkazu SELECT do premenných _vyrobo a _pocet
        EXIT_LOOP _retCode # _ERR_NO_ERROR
    END_LOOP

END Stroj_1_LocVarList

; Zoznam výrobkov a ich počet pre stroj s menom "Stroj 1"
; prostredníctvom SQL_PREPARE - verzia 2 alebo 5
PROCEDURE Stroj_1_LocRowIdent
    BOOL _useBinding = @TRUE ; pouzit SQL_BINDIN?
    TEXT _par = "Stroj 1" ; hodnota parametra
    TEXT _sql
    TEXT _errorMsg
    RECORD NOALIAS (SD.StrojVyrabal) _vyrabal

    IF _useBinding THEN ; alternativa s SQL_BINDIN (verzia 5)

        _sql := "SELECT Vyroby.Meno, Count(*) FROM Vyroby, Stroje, StrojVyrabal WHERE Stroje.Meno = #PAR# "
        _sql := _sql + "AND Stroje.ID=StrojVyrabal.ID_STROJ AND Vyroby.ID=StrojVyrabal.ID_VYROBO GROUP BY Vyroby.
Meno"
        SQL_PREPARE _handle, _retCode, _sql BINDOUT _vyrabal[1]
        SQL_BINDIN _handle, _retCode, _par ; nastav hodnotu vstupneho parametra

    ELSE ; alternativa bez SQL_BINDIN (verzia 2)

        _sql := "SELECT Vyroby.Meno, Count(*) FROM Vyroby, Stroje, StrojVyrabal WHERE Stroje.Meno = 'Stroj 1' "

```

```

    _sql := _sql + "AND Stroje.ID=StrojVyrabal.ID_STROJ AND Vyroby.ID=StrojVyrabal.ID_VYROBOK GROUP BY Vyroby.
Meno"
    SQL_PREPARE _handle, _retCode, _sql BIND _vyrabal[1]

ENDIF

IF _retCode # _ERR_NO_ERROR THEN
    _errorMsg := "Error SQL_PREPARE " + _sql
    LOG _errorMsg PRIORITY _LOG_PRTY_ERROR
    RETURN
ENDIF

DO_LOOP
    SQL_FETCH _handle, _retCode ; načítanie 1 riadku výsledku SQL príkazu SELECT do riadku .1 premennej
_vyrabal
    EXIT_LOOP _retCode # _ERR_NO_ERROR
END_LOOP
END Stroj_1_LocRowIdent

; Zoznam výrobkov a ich počet pre stroj s menom "Stroj 1"
; prostredníctvom SQL_PREPARE - verzia 3 a 6
PROCEDURE Stroj_1_LocRecIdent
    BOOL _useBinding = @TRUE ; použiť SQL_BINDIN?
    TEXT _par = "Stroj 1" ; hodnota parametra
    TEXT _sql
    TEXT _errorMsg
    INT _maxRows
    RECORD NOALIAS (SD.StrojVyrabal) _vyrabal

    IF _useBinding THEN ; alternatíva s SQL_BINDIN (verzia 6)

        _sql := "SELECT Vyroby.Meno, Count(*) FROM Vyroby, Stroje, StrojVyrabal WHERE Stroje.Meno = #PAR# "
        _sql := _sql + "AND Stroje.ID=StrojVyrabal.ID_STROJ AND Vyroby.ID=StrojVyrabal.ID_VYROBOK GROUP BY Vyroby.
Meno"
        SQL_PREPARE _handle, _retCode, _sql BINDOUT _vyrabal
        SQL_BINDIN _handle, _retCode, _par ; nastav hodnotu vstupného parametra

    ELSE ; alternatíva bez SQL_BINDIN (verzia 3)

        _sql := "SELECT Vyroby.Meno, Count(*) FROM Vyroby, Stroje, StrojVyrabal WHERE Stroje.Meno = 'Stroj 1' "
        _sql := _sql + "AND Stroje.ID=StrojVyrabal.ID_STROJ AND Vyroby.ID=StrojVyrabal.ID_VYROBOK GROUP BY Vyroby.
Meno"
        SQL_PREPARE _handle, _retCode, _sql BIND _vyrabal

    ENDIF

    IF _retCode # _ERR_NO_ERROR THEN
        _errorMsg := "Error SQL_PREPARE " + _sql
        LOG _errorMsg PRIORITY _LOG_PRTY_ERROR
        RETURN
    ENDIF

    _maxRows := 10
    DO_LOOP
        SQL_FETCH _handle, _retCode, _maxRows ; načítanie maximálne 10 riadkov výsledku SQL príkazu SELECT do
premennej _vyrabal
        EXIT_LOOP _retCode # _ERR_NO_ERROR
    END_LOOP
END Stroj_1_LocRecIdent

BEGIN
    ; Napojím sa na databázu
    SQL_CONNECT "UID=dba;PWD=sql;DSN=Database", _handle, _retCode ON SELF.DBM
    IF _retCode # _ERR_NO_ERROR THEN
        LOG "Chyba pri konektovaní databázy" PRIORITY _LOG_PRTY_ERROR
    END
    ENDIF

    ; Naplním Tabuku Stroje
    CALL Napln_Stroje

```

```
; Naplním Tabuku Vyroby
CALL Napln_Vyroby

; Naplním Tabuku StrojVyrabal
CALL Napln_StrojVyrabal

; Zistím, aké výrobky vyrábal "Stroj 1"
CALL Stroj_1_LocVarList

; Zistím, aké výrobky vyrábal "Stroj 1"
CALL Stroj_1_LocRowIdent

; Zistím, aké výrobky vyrábal "Stroj 1"
CALL Stroj_1_LocRecIdent

END
```

Poznámka

Pri ukonení skriptu akciou END (alebo iným ubovoným spôsobom), sú automaticky zatvorené všetky spojenia s databázou.

**Súvisiace stránky:**

[Akcie v skriptoch](#)