

DB_UPDATE

Akcia DB_UPDATE a DBS_UPDATE

Funkcia

Deklarácia

Modifikácia existujúceho riadku (riadkov) v databáze.

```
DB_UPDATE handleIdent_Int, rowIdent, retCodeIdent_Int [WHERE  
strExpression_Str [BINDIN varIdent1, varIdent2, ... ]] [ORAHINT  
hintIdent_Str]  
  
DB_UPDATE handleIdent_Int, rowIdent, retCodeIdent_Int [WHERE  
strExpression_Str [BINDIN structRowIdent]] [ORAHINT hintIdent_Str]  
  
DBS_UPDATE dbObjIdent, rowIdent, retCodeIdent_Int [WHERE  
strExpression_Str [BINDIN varIdent1, varIdent2, ... ]] [TRANS  
transHandle_Int] [ORAHINT hintIdent_Str]  
  
DBS_UPDATE dbObjIdent, rowIdent, retCodeIdent_Int [WHERE  
strExpression_Str [BINDIN structRowIdent]] [TRANS transHandle_Int]  
[ORAHINT hintIdent_Str]
```

alebo

```
DB_UPDATE handleIdent_Int, structIdent, retCodeIdent_Int [ORAHINT  
hintIdent_Str]  
  
DBS_UPDATE dbObjIdent, structIdent, retCodeIdent_Int [TRANS  
transHandle_Int] [ORAHINT hintIdent_Str]
```

Parametre

handleIdent_Int	in	Identifikátor typu <i>Int</i> - (<i>handle</i>) spojenia s databázou (DB_CONNECT).
dbObjIdent	in	Odkaz na objekt typu <i>Tabuka</i> .
rowIdent	in	Identifikátor jedného riadku štruktúry (vkladaný riadok).
structIdent	in	Identifikátor celej štruktúry (vkladané riadky).
retCodeIdent_Int	out	Návratová hodnota typu <i>Int</i> - úspešnosť akcie.
strExpression_Str	in	Výraz typu <i>String</i> , ktorý identifikuje modifikované riadky. Ak je výraz parametrizovaný, tak kúové slovo BINDIN a hodnoty parametrov (<i>structRowIdent</i> alebo <i>varIdent1</i> , <i>varIdent2</i> , ...) sú povinné.
varIdent1, varIdent2, ...	in	zoznam objektov, konštánt alebo lokálnych premenných , ktoré špecifikujú hodnoty parametrov parametrizovaného SQL výrazu <i>strExpression_Str</i> .
structRowIdent	in	Odkaz na riadok lokálnej premennej typu <i>Record</i> alebo na riadok štruktúrovanej premennej . Hodnoty riadkov špecifikujú hodnoty parametrov parametrizovaného SQL výrazu <i>strExpression_Str</i> .
transHandle_Int	in	Identifikátor Pripojenia na databázu.
hintIdent_Str	in	Výraz typu <i>String</i> , ktorý definuje Oracle SQL hint. Ten je použitý ako inštrukcia pre optimalizátor výkonu SQL príkazu. Zapisuje hodnotu bez obaúčujúcich znakov <i>/*+ <orahint>*</i> . Príklad použitia je uvedený tu .

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

Popis

Databáza musí by otvorená s prístupom `_DB_MODIFY`. Podobne ako pri akcii `DB_READ` (`DBS_READ`), môžu by modifikované riadky určené dvoma spôsobmi:

1. Nastavením kúovej položky v riadku štruktúry (`rowIdent`), resp. v celej štruktúre (`structIdent`). Nastavenie musí by realizované pred akciou **DB_UPDATE**. Položka musí by oznaená ako kú pri definícii objektu typu [Databáza](#). Potom sa v databáze hadá riadok, ktorý má zhodnú hodnotu pre kúovú položku a tento je modifikovaný. Vkladaný riadok alebo štruktúra musia ma korektný typ štruktúry. Všetky položky každého vkladaneho riadku musia ma platné hodnoty.
2. Hodnotou výrazu `strExpression_Str`, ktorý predstavuje klauzulu **WHERE** pre SQL príkaz **UPDATE**, ktorý vykonáva zmenu databázy. V tomto prípade je hodnota kúovej položky v štruktúre nepoužitá.

Výhodou akcie **DBS_UPDATE** pri práci s tabukou je možnosť vynechať jej otvorenie a zatvorenie (jednoduchší zápis).

Pre D2000 verziu 5: Nevýhoda akcie **DBS_UPDATE** je v rýchlosti. Každé volanie akcie **DBS_UPDATE** má za následok, v rámci jej obsluhy v procese [D2000 DBManager](#), nutnosť databázu otvoriť a následne zatvoriť, čo môže by asovo náročné a predstavuje z databázového hľadiska pomerne neštandardný postup. Nutnosť otvárania a zatvárania databázy je možné potlačiť v rámci transakčného spracovania tak, že za príkaz uvedieme parameter **TRANS**.

Pre D2000 verziu 6 a vyššie: [Optimalizácia](#) procesu [D2000 DBManager](#) (recyklovanie spojení, predpripravené spojenia) spôsobila, že akcia **DBS_UPDATE** sa vykoná rovnako rýchlo ako akcia **DB_UPDATE** a navyše sa ušetrí čas potrebný na spracovanie explicitného príkazu na otvorenie databázy **DB_CONNECT**.

Potreba otvorenia a zatvorenia databázy sa môže eliminovať v rámci transakčného spracovania tak, že za príkazom nasleduje parameter **TRANS**.

Príklad

Príklad práce s databázou (akcie `DB_...`).

Súvisiace odkazy

[DB_CONNECT](#)
[DB_DELETE](#)
[DB_DISCONNECT](#)
[DB_INSERT](#)
[DB_INSERT](#)
[DB_READ](#)
[DB_READ_BLOB](#)
[DB_UPDATE_BLOB](#)

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

Všetky databázové akcie



Súvisiace stránky:

[Akcie v skriptoch](#)