

DB_READ

Akcia DB_READ a DBS_READ

Funkcia

íťanie jedného alebo viacerých riadkov z tabuky.

Deklarácia

```
DB_READ handleIdent_Int, rowIdent, retCodeIdent_Int [WHERE  
strExpression_Str [BINDIN varIdent1, varIdent2, ... ]] [ORAHINT  
hintIdent_Str]
```

```
DB_READ handleIdent_Int, rowIdent, retCodeIdent_Int [WHERE  
strExpression_Str [BINDIN structRowIdent ]] [ORAHINT hintIdent_Str]
```

```
DBS_READ dbObjIdent, rowIdent, retCodeIdent_Int [WHERE strExpression_Str  
[BINDIN varIdent1, varIdent2, ... ]] [TRANS transHandle_Int] [ORAHINT  
hintIdent_Str]
```

```
DBS_READ dbObjIdent, rowIdent, retCodeIdent_Int [WHERE strExpression_Str  
[BINDIN structRowIdent ]] [TRANS transHandle_Int] [ORAHINT hintIdent_Str]
```

alebo

```
DB_READ handleIdent_Int, localStructIdent, retCodeIdent_Int, maxRows_Int  
[WHERE strExpression_Str [BINDIN varIdent1, varIdent2, ... ]] [ORAHINT  
hintIdent_Str]
```

```
DB_READ handleIdent_Int, localStructIdent, retCodeIdent_Int, maxRows_Int  
[WHERE strExpression_Str [BINDIN structRowIdent ]] [ORAHINT hintIdent_Str]
```

```
DBS_READ dbObjIdent, localStructIdent, retCodeIdent_Int, maxRows_Int [WHERE  
strExpression_Str [BINDIN varIdent1, varIdent2, ... ]] [TRANS  
transHandle_Int] [ORAHINT hintIdent_Str]
```

```
DBS_READ dbObjIdent, localStructIdent, retCodeIdent_Int, maxRows_Int [WHERE  
strExpression_Str [BINDIN structRowIdent ]] [TRANS transHandle_Int]  
[ORAHINT hintIdent_Str]
```

Parametre

handleIden	in	Identifikátor typu <i>Int</i> - (handle) spojenia s tabukou (DB_CONNECT).
dbObjIden	in	Odkaz na objekt typu Tabuka .
rowIdentInt	in / out	Identifikátor jedného riadku štruktúry (výsledok ítania).
retCodeIden	out	Identifikátor návratovej hodnoty - úspešnos akcie.
localStructIdent	out	Identifikátor celej lokálnej štruktúry.
maxRowwsInt	in	Identifikátor - maximálny počet ítaných riadkov.
strExprsionStr	in	Výraz, ktorý definuje ítaný riadok tabuky (riadky). Výraz môže obsahova aj podmienku triedenia (napr. " <i>ID<100 ORDER BY Name</i> "). Pokia je výraz parametrizovaný , musí by uvedené aj kúové slovo BINDIN nasledované hodnotami parametrov (<i>structRowIdent1, varIdent2, ...</i>).
varliden1, varliden2, ...	in	Zoznam objektov, konštánt alebo lokálnych premenných , ktoré sa použijú na mieste parametrov parametrizovaného výrazu <i>strExpression_Str</i> .
structRowIdent	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 výrazu <i>strExpression_Str</i> .
transHandleInt	in	Identifikátor Pripojenia na databázu.
hintIdentStr	in	Výraz typu <i>String</i> , ktorý definuje Oracle SQL hint. Ten je použitý ako inštrukcia pre optimalizátor výkonu SQL príkazu. Zapisuje hodnotu bez úvodných a koncových znakov /*+ <orahint> */. Príklad použitia je uvedený tu .

Návratový kód

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

Popis

Klauzula **WHERE** pre SQL príkaz **SELECT**, ktorý vykoná výber z tabuky, je definovaná bu hodnotou výrazu *strExpression_Str*, alebo, ak tátó hodnota nie je zadaná, tak hodnotami definovaných (platných) kúových položiek (položky oznaené ako **kúové** pri definícii objektu typu [Tabuka](#)) v prvom riadku štruktúry (*rowIdent* alebo *localStructIdent*). Nastavenie týchto hodnôt musí by realizované pred akciou **DB_READ** (**DBS_READ**).

Pozn: ak nie je zadaná ani podmienka, ani tabuка nemá kúové stpce, naítajú sa všetky dátá z tabuky (ak ich nie je viac ako špecifikované maximum).

Pozn: ak sú ítané dátá do jedného riadku štruktúry (verzie s parametrom *rowIdent*) s použitím kúových položiek, tak hodnoty všetkých kúových položiek v *rowIdent* musia by platné, inak sa ítanie neuskutoční, skoní s chybou a [%GetLastErrorCode\(\)](#) vráti chybu 667.

Z toho vyplýva, že klaузula **WHERE** nie je súčasou SQL príkazu **SELECT**, ak nie je definovaná hodnota výrazu *strExpression_Str*

- neexistujú kúové položky (položky oznaené ako **kúové** pri definícii objektu typu [Tabuka](#)),
- alebo nie je definovaná (platná) ani jedna z hodnôt kúových položiek,

- alebo štruktúra má nulovú dimenziu (v prípade *localStructIdent*).

Prvá forma zápisu akcie umožňuje preíta jeden riadok z tabuky do jedného riadku štruktúry.

Ak pri výbere vyhovuje podmienke pre výber viac ako jeden riadok tabuky, akcia **DB_READ(DBS_READ)** naša prvy riadok. Ak nevyhovuje ani jeden riadok, akcia vráti chybu prostredníctvom *retCodeIdent_Int*. Typ štruktúry (*rowIdent*) musí by zodpovedajúci typom tabuky.

Druhá forma zápisu umožňuje preíta jeden, alebo viac riadkov z tabuky. Maximálny počet riadkov uruje hodnota parametra *maxRows*. Ak hodnota *maxRows* je nastavená na -1, systém sa snaží preíta všetky riadky zodpovedajúce zadanej podmienke. Maximálny počet riadkov je obmedzený konfiguráciou parametrom databázy **Maximum vrátených riadkov**. Pri prekročení tohto počtu riadkov bude do lokálnej štruktúry *localStructIdent* náitaných prvých **Maximum vrátených riadkov** a v *retCodeIdent_Int* sa vráti chyba **_ERR_DATABASE_ROWS_LIMIT**.

Ak je hodnota *maxRows* väčšia ako **Maximum vrátených riadkov**, naša sa najviac **Maximum vrátených riadkov** riadkov. Pokiaľ je k dispozícii viac riadkov, v *retCodeIdent_Int* sa vráti chyba **_ERR_DATABASE_ROWS_LIMIT**.

Ak je hodnota *maxRows* nastavená na -2, systém najskôr zistí počet riadkov zodpovedajúcich danej podmienke. Ak ich je 0 alebo viac ako **Maximum vrátených riadkov**, nenaša žiadne riadky, v opanom prípade naša zistený počet riadkov. Tento spôsob môže byť optimalizaný dôvodov rýchlejší ako pre *maxRow=-1*. Počet riadkov sa zisuje pomocou výberu "SELECT COUNT(*) FROM table_name [WHERE where_condition]."

Ak je zadaná podmienka *StrExpression_Str*, pred zistením počtu riadkov sa ešte zisuje, i podmienka obsahuje klauzulu **ORDER BY**. Ak klauzulu obsahuje, táto sa aj s nasledujúcim textom odstráni. Dôvodom je, že pri zisovaní počtu riadkov napr. Sybase neakceptuje "SELECT COUNT(*) FROM aaaa WHERE bbbb = cccc ORDER BY dddd" a vracia chybu *Function or column reference to 'ddd' in the select list must also appear in a GROUP BY*.

Parameter *localStructIdent* musí predstavova meno lokálnej štruktúry zodpovedajúceho typu (t.j. rovnakého, aký je použitý v konfigurácii objektu *dbObjIdent* typu **Tabuка**). Pokiaľ chcete ma istotu, že výstup nebude riadený klauzulou **WHERE** vyskladanou na základe kľúčových položiek, je vhodné pred volaním akcie **DB_READ(DBS_READ)** redimenovať lokálnu štruktúru na 0.

Po výbere riadkov z tabuky sú tieto prenesené do premennej *localStructIdent*. V prípade potreby je jej zmenený rozmer (**REDIM**). Ak výberovej podmienke nevyhovuje ani jeden riadok, akcia sa ukončí úspešne (*retCodeIdent_Int = _ERR_NO_ERROR*) a rozmer štruktúry *localStructIdent* sa zmení na 0.

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

Pre D2000 verziu 5: Nevyhoda akcie **DBS_READ** je v rýchlosťi. Každé volanie akcie **DBS_READ** má za následok, v rámci jej obsluhy v procese **D2000 DBManager**, nutnosť databázu otvoriť a následne zatvoriť, čo môže byť asovo nároná innoš a predstavuje z databázového hadiska 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_READ** sa vykoná rovako rýchlo ako **DB_READ** a navyše sa ušetrí as potrebný na spracovanie explicitného príkazu na otvorenie databázy **DB_CONNECT**.

Príklad

Súvisiace odkazy

[Príklad práce s tabukou \(akcie DB_ ...\)](#)

[DB_CONNECT](#)
[DB_DELETE](#)
[DB_DISCONNECT](#)
[DB_INSERT](#)
[DB_INSUPD](#)
[DB_READ_BLOB](#)
[DB_UPDATE](#)
[DB_UPDATE_BLOB](#)

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

[Všetky databázové akcie](#)



Súvisiace stránky:

[Akcie v skriptoch](#)