

Parametrizácia SQL príkazov a výrazov

Parametrizácia SQL príkazov a výrazov znamená symbolické zadanie hodnôt v SQL príkazoch a výrazoch a neskoršie naplnenie parametrov konkrétnymi hodnotami.

Príklad parametrizovaného výrazu: " *Id* = #PAR# AND NAME LIKE #PAR# "

Príklad parametrizovaného príkazu: "SELECT Name, Surname FROM Persons WHERE *Id* = #PAR# AND Name LIKE #PAR# "

Existujú 3 spôsoby špecifikácie parametrov použité v množine akcií procesu [D2000 DBManager](#):

- **Znakom "?"**
Konvencia použitá v ODBC rozhraní (funguje iba pre ODBC verziu procesu DBManager). Každý parameter je špecifikovaný jedným otáznikom.
Príklad: " *Id* > ? AND Name = ? "
Poznámka: Takto parametrizovaný SQL výraz nebude fungovať pre OCI verziu procesu DBManager.
- **Výrazom ":name"**
Konvencia použitá v OCI rozhraní (funguje iba pre OCI verziu procesu DBManager). Každý parameter je špecifikovaný dvojčipkou a menom.
Príklad: " *Id* > :par1 AND Name = :par2 "
Poznámka 1: Takto parametrizovaný SQL výraz nebude fungovať pre ODBC verziu procesu DBManager.
Poznámka 2: Tento spôsob parametrizácie umožňuje viacsobné použitie toho istého parametra (napr. :par1) v SQL príkaze alebo výraze:
"SELECT Name, Surname FROM Person WHERE Name= :par1 OR Surname = :par1 OR BirthDate > :par2 "
• **Výrazom "#PAR#"**
Vnútroňný zápis pre proces DbManager. ODBC verzia procesu DBManager pred použitím nahradí všetky výrazy #PAR# znakom ? (prevedie ich do ODBC konvencie), OCI verzia procesu DbManager pred použitím nahradí všetky výrazy #PAR# výrazmi :01, :02, at. (prevedie ich do OCI konvencie).
Tento zápis je univerzálny a takto parametrizované SQL príkazy, ktoré ho používajú, nie je potrebné pri prechode medzi ODBC a OCI verziami procesu DBManager meniť.

Parametrizáciu môže použiť v SQL výraze vybraná množina akcií pracujúcich s databázou. Jedná sa o akcie:

- [DB\(S\)_DELETE](#)
- [DB\(S\)_READ](#)
- [DB\(S\)_UPDATE](#)

Pokiaľ je SQL výraz parametrizovaný, akcia musí použiť kúové slovo **BINDIN** a za ním jedno z nasledujúcich:

- zoznam objektov, konštánt alebo [lokálnych premenných](#),
- [odkaz na riadok lokálnej premennej](#) typu *Record* alebo na riadok [štruktúrovanej premennej](#).

Hodnoty parametrov sa nastavujú podľa hodnôt uvedených za kúovým slovom **BINDIN**.

Parametrizáciu podporuje aj akcia [SQL_PREPARE](#), ale v tomto prípade nie je zoznam hodnôt parametrov súčasťou akcie. Uvedením kúového slova **BINDOUT** (namiesto **BINDIN**) sa špecifikuje, že SQL príkaz je parametrizovaný a následne sa volaním akcie [SQL_BINDIN](#) nastavujú hodnoty parametrov. Výhodou je, že pre získanie výsledkov s inými hodnotami parametrov nie je potrebné znovu volať [SQL_PREPARE](#), ale stačí zopakovať [SQL_BINDIN](#) s novými hodnotami a následne je možné volať [SQL_FETCH](#) na získanie výsledkov.

Zmyslom parametrizácie je zrýchliť a ušetriť prácu SQL databázy. Pri vyhodnocovaní SQL príkazu dochádza k jeho parsovaniu a kompilácii. Pokiaľ sa použije viackrát ten istý SQL príkaz, SQL databáza môže recyklovať už parsovaný a skompilovaný príkaz. Parametrizáciou možno napríklad príkazy:

"SELECT Name FROM Person WHERE *Id*=1"

"SELECT Name FROM Person WHERE *Id*=5"

"SELECT Name FROM Person WHERE *Id*=100"

nahradí jediným príkazom

"SELECT Name FROM Person WHERE *Id*= #PAR#"

a teda miesto troch SQL príkazov bude databáza parsovať a kompilovať iba jeden (a pri druhom a ďalších vykonávaníach ho môže recyklovať).

Pokiaľ sa použije parametrizácia pri volaní [SQL_PREPARE](#) a nastavenie parametrov viacsobnou akciou [SQL_BINDIN](#), je zarúbené, že parsovanie a kompilácia SQL príkazu prebehne iba raz (pri volaní [SQL_PREPARE](#)).

Pokiaľ sa použije parametrizácia pri volaní akcií [DB\(S\)_DELETE](#), [DB\(S\)_READ](#) alebo [DB\(S\)_UPDATE](#), je vhodné upraviť parametre databázy (napríklad Oracle: *session_cached_cursors*), aby táto uchovávala (cachovala) dostatočné množstvo skompilovaných príkazov (kurzorov), aby medzi volaniami [DB\(S\)_*](#) príkazov zostali v kurzorovej cache.

Súvisiace odkazy:

[DB_DELETE](#)

[DB_READ](#)

[DB_UPDATE](#)

[SQL_PREPARE](#)

[SQL_BINDIN](#)

[SQL_FETCH](#)

[SQL_FREE](#)

[Všetky databázové akcie](#)