

# Dotaz typu update v MarushkaDesignu



**GEOVAP**

## OBSAH

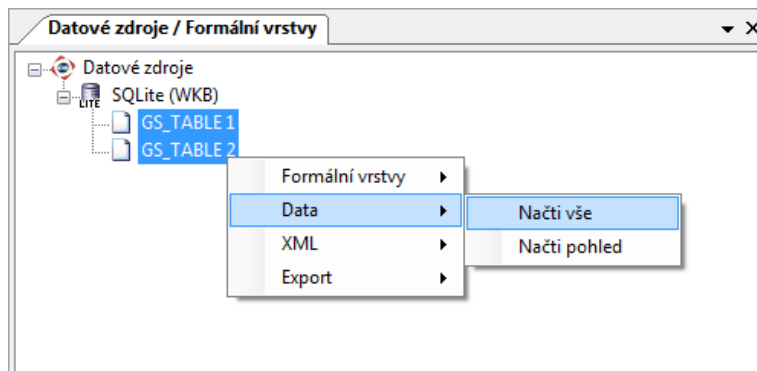
1	CÍL PŘÍKLADU .....	2
2	PRÁCE S PŘÍKLADEM .....	2
3	UKÁZKA DIALOGOVÉHO OKNA .....	3
4	STRUČNÝ POPIS PŘÍKLADU V MARUSHKADESIGN .....	4

## 1 Cíl příkladu

V tomto příkladu si ukážeme nastavení dotazu typu Update v MarushkaDesignu. Příklad byl vytvořen ve verzi 4.0.1.0 a nemusí být tedy kompatibilní se staršími verzemi.

## 2 Práce s příkladem

- Do složky **c:\MarushkaExamples\** rozbalíme obsah souboru **Update\_CZ.zip**. Cílovou složku je nutné respektovat kvůli provázanosti cest s projektem. V případě umístění souborů do jiné cesty by nebylo možné s příkladem pracovat.
- V prostředí MarushkaDesignu otevřeme projekt **Update\_CZ.xml**.
- Označíme obě formální vrstvy, v kontextovém menu vybereme Data – Načti vše:



- V mapovém okně zvolíme zobrazit „Vše“:



- Spustíme lokální webový server:



### 3 Ukázka dialogového okna

Obr 1: Ukázka příkladu v lokálním web serveru



Obr 2: Ukázka aktualizace dat pomocí dotazu typu Update - *Edituj záznam*



Obr 3: Výsledek informačního dotazu - *Informace*



## 4 Stručný popis příkladu v MarushkaDesign

Testovací příklad obsahuje databázi v SQLite, ve které je jedna publikační vrstva. V datovém zdroji jsou dvě formální vrstvy, odkazující se na fyzickou vrstvu (databázovou tabulku). Nositel dat je fyzická vrstva (databázová tabulka) „GS\_TABLE“.

Příklad obsahuje definici Update dotazu, sloužícího k aktualizaci popisných dat geometrických elementů s možností využít číselníky a jeden informační dotaz.

Dotaz **Edítuj záznam** obsahuje dynamický číselník „CODE“ (viz. pozn. 1) s našeptávačem (viz. pozn. 3), dvě položky pouze pro čtení „Note ReadOnly“ a „HTML obrazek“, položku datum „DATE“, textovou položku „NOTE2“, číselnou položku „NOTE3“ a statický číselník „DIVISION“ (viz. pozn. 2).

Dotaz pro každý vstupní parametr včetně číselníku (konfigurace shodná s konfigurací lokalizace s číselníky), musí mít nadefinovanou položku v kolekci parametrů dotazu „QueryParams“. Je velmi důležité těmto parametrům definovat jejich typ. HTML klient na základě této definice následně generuje vizuální komponenty, které kontrolují vstup, jenž uživatel zadal. (DateTime zobrazuje kalendář, pro hodnoty typu double nebo float řeší konverzi desetinné čárky na tečku atd.)

Konfigurace dotazu typu **UPDATE** spočívá v konfiguraci dvou SQL dotazů:

### A) Inicializační dotaz „InitSqlTemplate“

Tento SQL dotaz slouží k naplnění hodnot parametrů definovaných v položce *QueryParams*. Pokud budeme potřebovat vyplnit parametr s indexem 1, dotaz musí vrátet pojmenovaný sloupec konstantou „1“. Sloupce pojmenované indexy musí být v dotazu řazeny vzestupně. Dále musí dotaz inicializovat všechny parametry v položce *QueryParams*. Pokud budeme chtít zobrazit ve formuláři položku, která je jen pro čtení, tak sloupec pojmenujeme textovou konstantou, která musí být jiná od „N“. Pořadí položky pouze pro čtení určuje pořadí sloupce ve výsledku. Příklad dotazu:

```
SELECT CODE "1", NAME2 "Note ReadOnly",
'<imgsrc=./obrazy/PrintWidth.png/>' "HTML obrazek",
DATE "2", NOTE2 "3", DIVISION "4", NOTE3 "5" FROM GS_TABLE WHERE
id=~(long)ID~
```

Inicializujeme 5 parametrů a výsledný formulář obsahuje 5 parametrů + 2 pouze pro čtení.

### B) Update dotaz „UpdSqlTemplate“

Běžná update fráze, k hodnotám přistupujeme přes tildu a index parametru. Velmi důležité je použít přetypování parametrů, pokud nebudeme parametry přetypovávat, hrozí nebezpečí SQL injekce. Příklad dotazu:

```
UPDATE GS_TABLE set NOTE=~(string)1~, DATE=~(datetime)2~,
NOTE2=~(string)3~, DIVISION=~(string)4~, NOTE3=~(int)5~ where
id=~(long)ID~
```

Možné hodnoty při použití přetypování jsou:

*string* – textová hodnota - ~(string)1~

*int* – celočíselná hodnota - ~(int)1~

*long* – celočíselná hodnota - ~(long)1~

*decimal* – číselná hodnota - ~(decimal)1~

*double* – číselná hodnota - ~(double)1~

*datetime* – hodnota typu datum - ~(datetime)1~

*idlist* – používá se pouze, pokud potřebujeme podmínku typu ID in (id1,id2,id3), parametr v QueryParams definujeme jako string, a může nabývat hodnot znaku anglické abecedy a číslic - ... where id in ~(idlist)1~

Příklad obsahuje také informační dotaz nazvaný **Informace**. Jedná se o standardní informační dotaz zobrazující aliasy sloupců tabulky GS\_TABLE a hodnoty v nich obsažené.

Pozn. 1: *Dynamický číselník* – je pomocný dotaz pro vyhodnocení nabízených hodnot v parametru dotazu na základě dotazu. Dotaz vrací jediný sloupec, který bude seznamem nabízených hodnot.

Příklad dotazu: `select code from gs_table where code like ~(string)1~||'%' order by code asc`

Pozn. 2: *Statický číselník* – je pomocný dotaz pro vyhodnocení nabízených hodnot v parametru dotazu bez databázového dotazu. Seznam statických hodnot je napevno definován v položce *ListOfValues*, jednotlivé položky jsou zde odděleny klávesou ENTER.

Pozn. 3: *Našeptávač* – zapíná se v položce *DynamicCodeLists* u rodičovského dotazu a slouží k tomu, že při zadávání názvů položek je dynamicky nabízen seznam položek s odpovídajícím názvem.