

Návrh replikujících aplikací

WinBase602 disponuje schopností replikovat informace z jednoho serveru na jiný.

V manuálu *WinBase602 – Příručka správce* v kapitole *Provoz replikací mezi servery* jsou vysvětleny pojmy a principy replikací a popsán provoz soustavy serverů sdílejících aplikací. Tam také hledejte informace o poštovních programech doručujících data mezi servery.

Na tomto místě se budeme zabývat otázkou, jak vyvinout aplikaci, která bude schopna replikovat svá data mezi servery.

Návrh sdílené aplikace

V této části popíšeme, jak se navrhuje databázová aplikace, která poběží souběžně na více serverech a mezi jejímiž instancemi se budou replikovat data.

Data, obsažená v tabulkách aplikace, mohou být buď soukromá, nebo sdílená. Pro sdílené tabulky je nutno zadat řadu doplňujících údajů. Návrh sdílené aplikace začíná proto analýzou toku dat mezi sdílenými tabulkami.

Při sdílení dat mohou různé servery hrát různé úlohy. V určité tabulce může server hrát specifickou roli pouze číst data, zatímco jiný server může data měnit a replikovat tyto změny na ostatní servery. Proto popis způsobu sdílení obsahu jednotlivých tabulek, není v aplikaci nastaven na jedinou možnou hodnotu, ale obsažen v tzv. *replikačních vztazích*, které servery mezi sebou podle svého postavení navazují.

Analýza toku informací mezi servery a rizik souběžné editace

Klíčovým bodem návrhu sdílené aplikace je otázka, jak se vypořádat s možností souběžné editace téhož záznamu na více serverech. Souběžná editace je zpravidla nežádoucí – pokud například dva lidé na dvou serverech doplní do stejného záznamu různé informace, pak se v procesu replikování mohou snadno jedny z těchto informací ztratit.

Tento problém nemá jediné univerzální řešení. Existuje řada cest, jak se s ním vypořádat. Nalezení optimální cesty vyžaduje důkladnou analýzu informačních toků mezi servery sdílejícími aplikaci.

Tok informací v některé části aplikace zpravidla odpovídá některému z těchto schémat (seřazených podle vzrůstající náročnosti na replikace):

- A) Záznamy vznikají a jsou editovány na pouze jednom serveru. Na ostatní servery se replikují změny.
- B) Záznamy vznikají na různých serverech, každý záznam může být editován z podstaty aplikace pouze na jednom serveru - například na tom serveru, na němž vzniknul.
- C) Tentýž záznam může být editován na různých místech a je nutno zajistit, že nebude editován souběžně.
- D) Tentýž záznam může být editován na různých místech souběžně. Pokud na dvou serverech vzniknou dvě verze téhož záznamu, je určeno, který server se prosadí se svou verzí.
- E) Tentýž záznam může být editován na různých místech souběžně. Pokud na dvou serverech vzniknou dvě kopie tohoto záznamu, není definováno, která se prosadí.

Při analýze požadavků na aplikaci je žádoucí co nejvíce informačních toků nespádajících do typu A) převést na schéma B). Typický obrat vypadá takto:

Pokud je třeba na některém serveru doplnit nebo pozměnit údaje v záznamu vzniklém na jiném serveru, pak se místo toho vloží do sdílené tabulky změnový záznam s relačním odkazem na původní záznam.

Využití tohoto obratu není povinné, ale zjednodušuje návrh a provoz aplikace. Dále popíšeme způsoby, jak zamezovat vzniku různých variant záznamu jinými způsoby.

Vlastnosti sdílených tabulek

Na základě analýzy informačních toků podle předchozího bodu se pro každou vytvářenou tabulku zodpoví tyto otázky:

1. Bude se obsah této tabulky vůbec replikovat? V typické sdílené aplikaci existují tabulky se sdíleným obsahem a zároveň i tabulky se soukromým obsahem týkajícím se pouze jednoho serveru.
2. Budou se nové záznamy vkládat do tabulky na jednom serveru nebo na více serverech? Pokud nastává druhý případ, pak je nutno tabulku vybavit *univerzálním klíčem pro replikaci*, jehož hodnoty budou unikátní na různých serverech.
3. Budou se změny přenášet mezi dvěma servery pouze jedním směrem nebo obousměrně? V druhém případě je žádoucí (ač není nutno) zajistit, aby změny vzniklé replikováním ze serveru A na server B nebyly zpětně přenášeny na server A. Tomuto nežádoucímu jevu říkáme *ozvěna*.
4. Může dojít k souběžnému editování téhož záznamu na více serverech? Pokud souběžná editace není vyloučena z podstaty aplikace, pak chceme bránit vzniku této situace nebo řešit ji, až vznikne? Má souběžné editaci bránit sama **WinBase602** předáváním editačního peška, anebo se o to postará aplikace jiným způsobem?

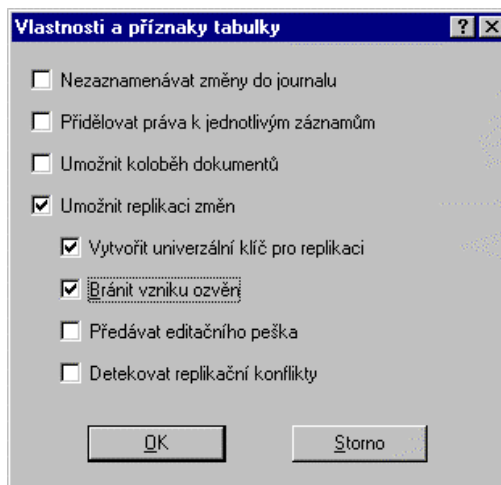
5. Povolíme-li souběžnou editaci téhož záznamu na více serverech, chceme tento stav detekovat, abychom mohli rozhodnout, která varianta má přednost?
6. Přejeme si, aby se replikovalo i zrušení záznamu, anebo zrušení záznamu na jednom serveru nemá ovlivnit tentýž záznam na jiných serverech?
7. Které sloupce a které záznamy z tabulky se mají replikovat?
8. Jak často se kontrolovat, zda je co replikovat, a případně vytvořit nový replikační paket?

Odpovědi na tyto otázky ovlivní vytváření tabulek a další fáze návrhu sdílené aplikace. Odpovědi na prvních šest otázek jsou v převážné míře společné pro celou aplikaci a zadávají se při návrhu tabulek. Odpověď na sedmou a osmou otázku se týká vztahu dvou serverů, a proto se specifikuje v jedné nebo více variantách v tzv. replikačních vztazích definovaných v aplikaci.

Návrh tabulky ve sdílené aplikaci

V menu **Vlastnosti** návrháře tabulek je příkaz **Příznaky**, který otevírá okno se speciálními příznaky tabulky. Kromě prvních tří (zaznamenávání změn do journalu, přidělování práv k jednotlivým záznamům, umožnění koloběhu dokumentů) se všechny příznaky týkají replikací.

- Příznak **Umožnit replikaci změn** musí být zapnut v každé replikované tabulce. Ne-li-li zapnut, není obsah tabulky přenášen mezi servery.
- Příznak **Vytvořit univerzální klíč pro replikaci** přidá do tabulky nový sloupec sloužící jako unikátní klíč. Jeho hodnoty se automaticky přidělují při vkládání záznamů a jsou unikátní i mezi servery. Ne-li-li příznak zapnut, používá se pro identifikaci záznamů během replikování primární klíč (ten musí existovat). Pokud by se do takové tabulky vkládaly záznamy na různých serverech, musí aplikace sama zajistit, že nenastane duplicita hodnot v primárním klíči.
- Příznak **Bránit vzniku ozvěn** přidá do tabulky dodatečnou informaci, která zamezí posílání téže změny mezi servery tam a zpět. Není důvod příznak zapínat, pokud se data budou replikovat pouze jedním směrem. Ne-li-li příznak zapnut při obousměrném replikování, vzroste objem komunikace mezi servery na dvojnásobek, ale nedojde z tohoto titulu k žádným chybám.
- Příznak **Předávat editačního peška** zajistí, že jeden záznam bude možno editovat v jednom okamžiku na nejvýše jednom serveru. Tím je zamezeno vzniku více verzí tohoto záznamu.
- Příznak **Detekovat replikační konflikty** zajistí, že se při replikaci rozpozná situace, kdy přichází změna do záznamu, který byl mezitím změněn. Detekování replikačních konfliktů funguje pouze tehdy, pokud v grafu replikací mezi servery neexistují cykly. Ne-li-li zapnut tento příznak ani příznak **Předávat editačního peška**, pak se do záznamu vždy zaznamená poslední došlá změna a některé změny se mohou ztratit.

Vlastnosti
tabulky

Využití univerzálního klíče

Při převádění starší aplikace na aplikaci schopnou replikování se v tabulkách, do nichž se budou vkládat záznamy na více serverech, nahrazuje dosavadní primární klíč (sloupec resp. sloupec s unikátní hodnotou identifikující záznam) *univerzálním klíčem*. Pouze u univerzálního klíče je vyloučen vznik duplicity hodnot při vkládání záznamů na různých serverech.

Sloupec obsahující univerzální klíč se v návrhu tabulky nedeklaruje, vznikne automaticky na základě nastavení příznaku **Vytvořit univerzální klíč pro replikaci**. Sloupec, který dříve plnil funkci primárního klíče, je vhodné z tabulky odstranit.

Relačně závislé tabulky si potřebují ve svých záznamech pamatovat hodnotu klíče z nadřazené tabulky. Sloupec, do něhož se v podřízených tabulkách zaznamená hodnota univerzálního klíče, musí být typu **Binary(16)**. Jméno univerzálního klíče v nadřazené tabulce je **_w5_UNIKEY**.

Návrh replikačních vztahů

Ty údaje o replikování, které mohou být na různých serverech nastaveny různě (v závislosti na roli, kterou server hraje), autor aplikace specifikuje v replikačních vztazích. Při navazování spojení mezi dvěma servery se pak vybere některý vztah a uplatní se informace v něm uvedené v nastavení konkrétních replikačních pravidel.

Vztahy jsou pojmenovanými objekty, které vytvoří autor aplikace. Uživatel ani správce provozu aplikace do návrhu vztahů nikdy nezasahuje.

Co obsahuje replikační vztah?

Vztah obsahuje pro aplikaci jako celek replikační periody pro každý směr komunikace. Pro každou tabulku schopnou sdílení a pro každý z obou serverů obsahuje:

- *seznam sloupců* replikovaných na protější server;
- *podmínku pro výběr záznamů* pro replikaci na protější server;
- Příznak **Replikovat rušení záznamů**, který způsobí, že zrušení záznamu na jednom serveru způsobí jeho zrušení i na protějším serveru. V takové tabulce se nesmí provádět operace uvolnění zrušených záznamů (přínejmenším před replikováním informace o zrušení).

Například aplikace pro distribuci ceníků a sběr objednávek může být instalována na serverech výrobce, regionálních distributorů a místních dealerů. Aplikace obsahuje dva vztahy: vztah „výrobce-distributor“ a vztah „distributor-dealer“.

Podmínka pro výběr záznamů se zapisuje v syntaxi předepsané pro podmínku v klauzuli WHERE příkazu SELECT jazyka SQL.

Dále se v replikačním vztahu specifikuje, jak se má na obou replikujících serverech nastavit způsob řešení případných replikačních konfliktů.

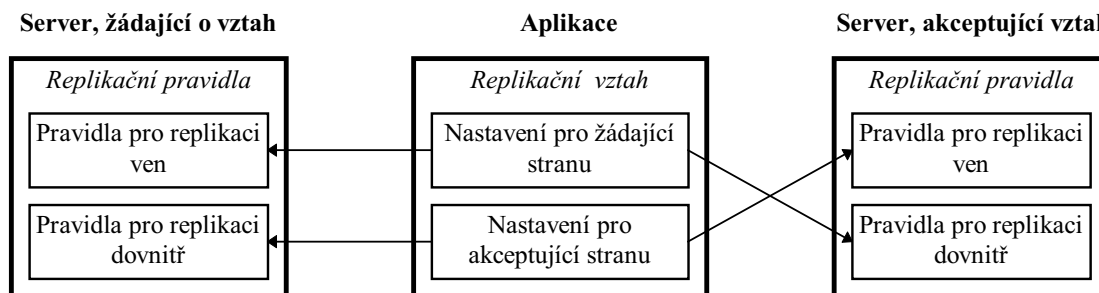
Dialog pro návrh replikačního vztahu

Replikační vztah se navrhuje v dialogovém okně rozděleném do tří záložek. Na první záložce se určí, jak se má postupovat při vzniku replikačního konfliktu. Další dvě záložky předepisují, které záznamy a které sloupce se replikují z jednotlivých replikovatelných tabulek.

Návrh replikačního vztahu

Uplatnění replikačního vztahu v replikačních pravidlech

Údaje specifikované v replikačním vztahu se uplatní tehdy, když se dva servery dohodou na navázání sdílení aplikace v tomto vztahu. Pak se informace z replikačního vztahu přepokopírují do iniciálního nastavení replikačních pravidel na obou stranách takto:



Vnitřní formát definice replikačního vztahu

Replikační vztahy definuje autor aplikace a využívá k tomu zpravidla interaktivní návrhář, který je součástí vývojového prostředí. V některých speciálních případech může být potřebné vytvořit replikační vztah programem.

Replikační vztah je objektem patřícím k aplikaci, je uložen v tabulce objektů OBJTAB a programy s ním mohou pracovat podle pravidel popsaných v manuálu.

Nový replikační vztah se z programu vytvoří voláním funkce `Insert_object` s parametrem `CATEG_REPLREL`. Do takto vytvořeného objektu (nebo do libovolného existujícího replikačního vztahu) se zapíše jeho definice v textové podobě vytvořená podle níže uvedených syntaktických pravidel:

- Definice začíná číslem ležícím v rozsahu 0 až 3 (včetně) udávajícím způsob řešení replikačních konfliktů. Jednotlivé hodnoty odpovídají po řadě přepínačům na první záložce dialogového okna pro editaci replikačního vztahu.
- Dále následují dvě části se shodnou syntaxí odpovídající po řadě nastavení parametrů v replikačních pravidlech na stranách žádající o replikační vztah a akceptující replikační vztah.
- Každá část začíná dvěma celými čísly udávajícími počet hodin a minut replikační periody.
- Za nimi následuje posloupnost záznamů odpovídajících jednotlivým tabulkám účastnícím se replikování.
- Záznam o tabulce se skládá z hvězdičky, jména tabulky, replikační podmínky uvedené jako řetězec znaků v apostrofech (smí být prázdná), čísla 0 nebo 1 udávajících, zda se replikuje zrušení záznamu (1: ano, 0: ne), a seznamu jmen replikovaných sloupců.

Příklad definice replikačního vztahu:

```
3
0 2
*`OBJEDNÁVKY` ` ` 1 `CÍSLO ZBOZÍ` `DEALER` `POCET KUSU`
`CÍSLO OBJEDNÁVKY` `DNE` `DATUM` `CÍSLO_ZBOZÍ`
`POCET_KUSU` `CÍSLO_OBJ`
0 2
*`CENÍK` ` ` 1 `EVID. CÍSLO` `NÁZEV ZBOZÍ` `CENA` `TRÍDA`
`TRÍDA ZBOZÍ` `POPIS ZBOZÍ` `PLATÍ DO` `EVID_CÍSLO`
`NÁZEV` `PLATNOST` `POPIS`
*`OBJEDNÁVKY` `Is_repl_destin(_W5_UNIKEY)` 0
`POTVRZENO KUSU` `POTVRZENO`
```

Definici replikačního vztahu vytvořeného interaktivním návrhářem si lze nejnázve prohlédnout tak, že vztah exportujete do souboru a ten otevřete textovým editorem.

Typické problémy při vývoji replikujících aplikací

Tato kapitola obsahuje shrnutí typických chyb a problémů, s nimiž se potýkají vývojáři replikujících aplikací.

Při ladění replikací se často vyskytnou tyto problémy:

1. Ladění probíhá mezi dvěma servery a na každém z nich je jiná verze sdílené aplikace.

Jelikož je aplikace ve vývoji, probíhají v ní změny. Pak se může snadno stát, že replikace nedopadne dobře, protože některá změna, která má vliv na replikování, byla provedena pouze na jednom serveru. Nejtypičtější jsou tyto chyby:

- Do jedné z replikovaných tabulek je přidán nový sloupec, který se na druhé straně nevyskytuje. Replikovaná data se pak zapisují (pokud vůbec) do nesprávných sloupců.
- Je provedena změna v definici replikačního vztahu na jednom serveru. Při navazování sdílení však každý server použije svou verzi popisu replikačního vztahu a vznikne vztah, v němž si servery nebudou rozumět.

2. V definici replikačního vztahu nebo v replikačním pravidle se předeíše replikace sledovacího atributu.

Sledovací atributy se nesmí replikovat. Počínaje verzí 5.1 server WinBase602 tuto chybu rozpozná a sledovací atribut uvedený v definici replikačního vztahu při generování replikačních pravidel ignoruje. K chybě však může dojít, pokud se replikace sledovacího atributu vyvolá explicitně zásahem do replikačních pravidel.