



Ekonomická
fakulta
Faculty
of Economics

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Ekonomická fakulta
Katedra aplikované matematiky a informatiky

Bakalářská práce

Problematika replikace mezi databázemi

Vypracoval: Pavel Karlíček
Vedoucí práce: doc. Ing. Ladislav Beránek, CSc.

České Budějovice 2022

Prohlašuji, že svou [typ] práci jsem vypracoval/a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury. Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své [typ] práce, a to - v nezkrácené podobě/v úpravě vzniklé vypuštěním vyznačených částí archivovaných Ekonomickou fakultou - elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Datum

Podpis studenta

Abstrakt

Cílem této práce je zmapovat současný stav postupů při replikaci dat v relačních databázích a analyzovat možnost inovace replikace ve stávajícím procesu podnikové sféry. V uceleném přehledu metod popisuje formy replikace, co replikace dat umožňuje a nabízí, ve kterých různých scénářích ji lze použít a k jakým různým účelům se jí využívá.

V praktické části se práce zaměřuje právě na způsob použití replikace v praxi. Z pohledu technika servisní organizace spravující replikace databáze podnikového informačního systému Helios Green analyzuje možnosti inovace stávajících postupů. V závěru pak práce zhodnocuje možnosti posunu v praxi a hledá rovnováhu mezi nadšením pro nové technologie a nutnou správcovskou skepsí.

Abstract

The focus of this work is to map the current state of procedures for data replication in relational databases and to analyze the possibility of replication innovation in the existing business process. In a comprehensive overview of methods, it describes the forms of replication, what data replication enables and offers. It also shows in which different scenarios it can be used and for what different purposes it is used.

In the practical part, the work focuses on how to use replication in practice. It represents the point of view of a service organization technician managing the replication of the enterprise information system database Helios Green. The work analyzes the possibilities of innovating the existing procedures. In the conclusion, the thesis evaluates the possibilities of a progress in practice and seeks a balance between enthusiasm for new technologies and the necessary administrative skepticism.

Key words: database, replication, distributor, publisher, subscriber, agent

Obsah

Úvod	4
1. Pojmy	5
2. Využití replikace	7
2.1 Databázové aplikace	7
2.2 Dostupnost a spolehlivost	7
2.3 Databázový clustering	7
2.4 Zálohování a replikace	8
3. Složitost replikace	8
4. Techniky (typy) replikace	9
4.1 Rozdíl mezi asynchronní a synchronní replikací	9
4.2 Synchronní replikace	10
4.3 Asynchronní replikace	11
4.4 Částečně synchronní systém	11
4.5 Podle architektury	11
5. Strategie správy replikací	12
5.1 Horlivé šíření aktualizací	12
5.2 Líné šíření aktualizací	13
5.3 Techniky centralizovaného šíření aktualizací	14
5.4 Distribuované techniky	15
6. Formy replikačních strategií	15
6.1 Snímková replikace	15
6.2 Transakční replikace	16
6.3 Slučovací replikace	16
6.4 Replikace založená na SQL příkazech	16

7.	Nástroje pro replikace.....	16
7.1	Slony-I.....	17
7.2	Postgres-R.....	17
7.3	Pgpool-II.....	18
7.4	Replikace na MS SQL Serveru.....	18
7.5	IBM Informix Replication.....	19
7.6	Sybase Replication Server.....	19
7.7	Replikace v systému Helios.....	20
8.	Problémy při replikaci	20
9.	Fragmentace replikace	21
9.1	Horizontální fragmentace	21
9.2	Vertikální fragmentace	21
9.3	Hybridní fragmentace.....	21
	Praktická část	22
10.	Cíle práce	22
11.	Prezentační vrstva.....	22
11.1	Nastavení prostředí na PLDB.....	22
11.1.1	Uživatel pro replikace	23
11.1.2	Site ID a URL aplikace	23
11.2	Nastavení prostředí na zdrojové databázi	24
11.2.1	Místa replikace.....	24
11.2.2	Vzdálená přihlášení.....	25
11.2.3	Cíle publikace	25
11.3	Nastavení rozsahu replikace.....	26
11.3.1	Subjekt, pořadač a třída.....	26

11.3.2	Sestavy replikace.....	26
11.3.3	Položky sestavy replikace.....	29
11.3.4	Konkrétní nastavení sestavy replikace	30
12.	Databázová vrstva.....	30
12.1	Globální identifikátor záznamu	32
12.2	Tabulky nastavení propojení	33
12.3	Tabulky replikace	35
12.4	Speciální tabulky pro replikace	36
13.	Aplikační vrstva.....	37
13.1	SOAP protokol	37
13.2	XML.....	37
	Závěr.....	39
	Použité zdroje.....	39
	Seznam zkratk	40
	Seznam obrázků	41

Úvod

Téma replikací jsem pro svou práci zvolil proto, že jsem se v rámci své praxe několikrát setkal s nutností řešení replikací mezi databázemi a chtěl jsem své zkušenosti z praxe porovnat s ostatními teoretickými poznatky a ověřit, zda nejsou pro praxi k dispozici i jiné metody a postupy, jak databáze replikovat. V současné době je jednou ze součástí mé práce nastavování replikací mezi databázemi velmi rozšířeného podnikového systému Helios Green. Pracuji jako specialista technické podpory ve společnosti, která implementuje ERP (Enterprise Resource Planning) systémy pro několik velkých společností zejména ze branže utilit. Jako administrátor tohoto systému jsem dostal v několika projektech za úkol mimo jiné i nastavit replikace mezi databázemi mateřské a dceřiných společností. V tu dobu jsem měl o replikacích jen velmi malou představu a způsob jejich správného nastavení jsem musel nastudovat z různých zdrojů. Až posléze jsem zjistil, že replikace jsou využívány i pro jiné účely a v jiných souvislostech, a tak jsem rád zvolil toto téma i jako svou bakalářskou práci.

1. Pojmy

Nejprve jsou uvedeny definice některých základních pojmů, které budou užity dále v bakalářské práci (BP).

Databáze je strukturovaná a trvalá sada informací o některých aspektech reálného světa, organizovaná a uložena způsobem, který usnadňuje efektivní vyhledávání a úpravy. Struktura databáze je určena abstraktním datovým modelem. Primárně je to tato struktura, která odlišuje databázi od datového souboru. Nejoblíbenější datový model je relační, který představuje data jako sadu tabulek. Databáze obsahuje instance entit a vztahy mezi nimi. Například databáze pro školu může obsahovat instance studentů, profesorů, kurzů, kteří studenti absolvují, které kurzy, kteří profesori vyučují které kurzy atd. [4]

Replikace databáze je proces, který reprodukuje (také se dá říct duplikuje či kopíruje) data ze zdrojové databáze (též „master“ databáze nebo primární databáze) do cílové databáze (také „slave“ databáze nebo sekundární databáze). Do „master“ databáze jsou zaznamenávány změny (vkládání, aktualizace či mazání záznamů), které jsou replikovány do jedné nebo více „slave“ databází. Každá „slave“ databáze tyto změny přijme a odešle potvrzení o úspěšné aktualizaci, čímž umožní odeslání následných aktualizací. Toto je zjednodušený popis procesu, který může probíhat na různých platformách, různými způsoby a za různým účelem.

Replikou z tohoto pohledu nazýváme kopii dat, která může existovat ve stejné databázi jako zdrojová data, v jiné databázi ve stejné instanci nebo i v jiné samostatné instanci nebo na jiném serveru.

Jak uvádí encyklopedie databázových systémů: Při použití replikace dat má každá logická datová položka databáze několik fyzických kopií, z nichž každá je umístěna na jiném počítači, také nazývaném místo nebo uzel. V závislosti na kontextu a typu replikační architektury může termín replika odkazovat na jednu z fyzických kopií konkrétní datové položky nebo na celý web se všemi jeho datovými kopiemi. Replikace dat může sloužit různým účelům. Za prvé, může být použit pro zvýšení dostupnosti a zajištění odolnosti proti chybám, protože data mohou být v zásadě přístupná, pokud je k dispozici jedna replika. Za druhé, může poskytovat dobrý výkon. Ukládáním replik v blízkosti uživatelů, kteří chtějí mít přístup k datům, umožňuje replikace rychlý místní přístup. Za třetí, žádosti

o přístup lze distribuovat napříč replikami. Přidáním více replik do systému lze zvládnout vyšší příchozí pracovní zátěž, a tudíž lze dosáhnout vyšší propustnosti. [4]

Při replikaci databáze má každá datová položka několik fyzických kopií, nazývaných také repliky, které jsou distribuovány přes různé uzly (místa). V případě úplné replikace má každá datová položka kopii na každém místě. V tomto případě může termín replika také odkazovat na uzel hostující kopii celé databáze. Řízení replik má na starosti převod operací čtení a zápisu, které klienti zadávají na položkách logických dat, do operací na fyzických kopiích dat. Cílem je udržet konzistentní stav mezi všemi replikami a poskytnout klientovi konzistentní pohled na data. Řízení replik rozšiřuje kontrolu souběžnosti za účelem koordinace provádění souběžných transakcí v různých replikách.[8]

Distribuované databázové systémy (DDBS) jsou systémy využívající distribuované databáze. Distribuovaná databáze jsou datové kolekce, které přesto, že patří jednomu systému, jsou fyzicky umístěny na různých uzlech počítačové sítě. DDBS využívají ke svému fungování replikace databáze.

Simple Object Access Protocol (SOAP) je standardizovaný protokol používaný na aplikační vrstvě k přenosu zpráv v distribuovaných systémech. Standard byl definován a stále je udržován pracovní skupinou XML Protocol Working Group World Wide Web Consortium. SOAP se běžně používá v kontextu webových služeb. Zprávy SOAP jsou kódovány pomocí XML a jsou určeny k přenosu dat aplikací kódovaných XML.[4]

Extensible Markup Language (XML) je nejrozšířenější částečně strukturovaný formát pro data, který zavedla W3C v roce 1998. Soubory XML obsahují pouze značky a text podobný HTML. Zatímco však HTML definuje, jak se prvky zobrazují a tisknou, XML přiřazuje význam prvkům. HTML používá předdefinované značky, ale XML vyžaduje, aby vývojář obsahu definoval většinu značek. Takže jeho pomocí lze zadat prakticky jakékoli datové položky. Tím, že poskytuje společnou metodu identifikace dat, podporuje XML transakce mezi podniky a stal se „formátem“ pro elektronickou výměnu dat a webové služby.

2. Využití replikace

2.1 Databázové aplikace

Samotná povaha mnoha aplikací způsobuje, že tyto musí být rozděleny do několika uzlů. Je to velmi často z důvodu geografického rozmístění velkých korporací, které potřebují sdílet určitou část dat „globálních“ dat mezi všemi lokalitami, zatímco individuální data jednotlivých lokálních uživatelů zůstávají uložena pouze v místních uzlech.

Tento typ využití replikací bude zkoumán i v praktické části této bakalářské práce (dále jen BP), kde bude zkoumáno, jak replikace fungují v systému Helios Green.

2.2 Dostupnost a spolehlivost

Tyto vlastnosti jsou určitě největšími výhodami DDBS. Pravděpodobnost, že je systém stále korektní a aktuální nazýváme spolehlivostí. Pravděpodobnost, že je systém stále dostupný nazýváme dostupností. U distribuovaných systémů může nastat situace, kdy některý konkrétní uzel selže. Přesto jsou ostatní uzly dostupné a systém může s určitým omezením dále pracovat. Tím se zlepšuje jeho dostupnost a spolehlivost zejména v porovnání s centralizovanými databázovými systémy. A je to právě replikace, která zabezpečuje, že lze dosáhnout vyšší spolehlivosti a dostupnosti jak dat, tak i softwaru.

2.3 Databázový clustering

Replikace jsou užívány pro stavbu databázových clusterů. Clustering databází je proces kombinování více než jednoho serveru nebo více než jedné instance propojujících jednu databázi. Někdy jeden server nemusí stačit ke správě množství dat nebo počtu požadavků, to je situace, kdy je potřeba datový cluster.

Hlavními důvody pro využití databázového clusteru jsou:

- Redundance dat (data redundancy)
Všechny databáze v clusteru jsou synchronizovány (za pomoci replikací), takže každý uzel obsahuje přesně stejná data jako všechny ostatní. Pokud tedy dojde k selhání jednoho uzlu, stále máme data k dispozici na ostatních uzlech.

V extrémním případě může dojít k selhání $n-1$ uzlů, kde n je celkový počet uzlů v databázovém clusteru.

- **Vyvažování zátěže (load balancing)**
Škálovatelnost (jak se též nazývá vyvažování zátěže) rozkládá zátěž mezi různé uzly, které jsou součástí clusteru. To zajišťuje, že systém dokáže obsluhovat dotazy i při velmi vysokém vytížení.
- **Vysoká dostupnost (high availability)**
High availability (HA) souvisí s vyvažováním zátěže. Právě díky rozložení zátěže dokáže být databáze v clusteru extrémně dlouho dostupná, a to i v případě selhání či vypnutí některého serveru.

2.4 Zálohování a replikace

Replikace může být též vhodným doplňkem zálohování. Velkou výhodou replikace je, že může vytvářet kopie (téměř) v reálném čase, což se nám při vytváření Disaster recovery plánů (DRP) může velmi hodit, jelikož by se mohlo zdát, že při eventuálním selhání nepřicházíme o žádná data. Výrazná nevýhoda replikace z pohledu obnovy dat je ta, že replikace vytváří i kopie poškozených či jinak nehodnocených dat (např. chybným přepisem). Proto replikací nelze nahradit zálohy, ale je velmi vhodné, když se zálohování a replikace doplňují.

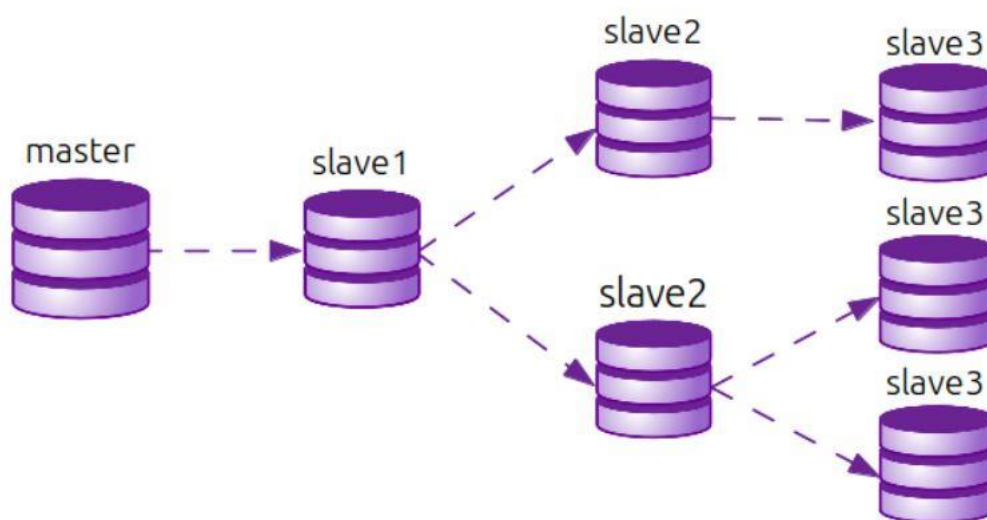
I když se proces replikace dat používá k vytváření instancí stejných dat nebo částí stejných dat, nesmíme si to plést s procesem zálohování, protože repliky jsou často aktualizovány a rychle ztrácejí jakýkoli historický stav. Zálohování na druhou stranu ukládá kopii dat nezměněnou po dlouhou dobu.

3. Složitost replikace

Složitostí replikace lze posuzovat ze dvou hledisek.

Prvním z nich je geografická vzdálenost mezi jednotlivými uzly a s ní spojená závislost na spolehlivosti internetu.

Druhým hlediskem je počet vrstev infrastruktury replikace. Pokud každý cílový uzel je přímo napojen na master, pak jde o jednovrstvý protokol. Vícevrstvý protokol pak dokáže využít některé cílové uzly k předávání dat do dalších cílových uzlů – označuje se též jako kaskádová replikace.



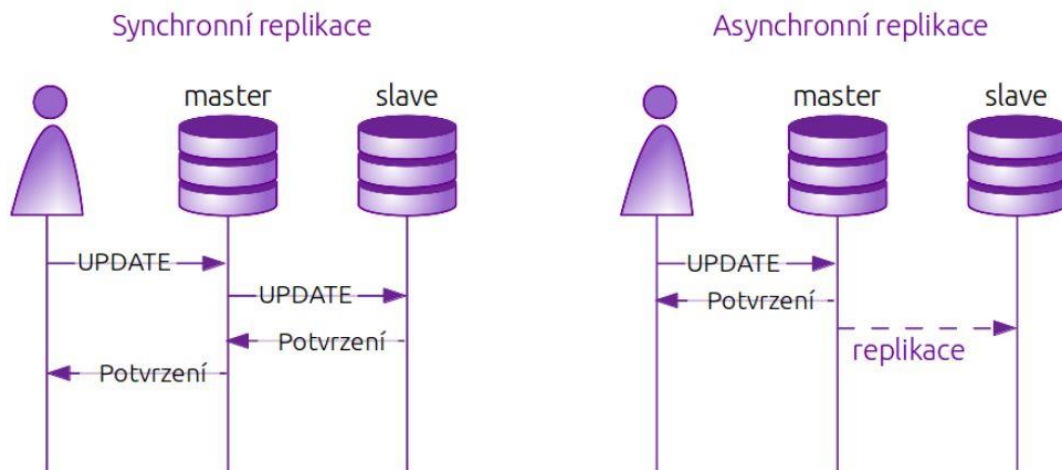
Obrázek 1: Složitost replikace [9]

4. Techniky (typy) replikace

Databázové replikace mohou být a jsou prováděny různými způsoby a technikami. Pro každé využití replikace se může jevit jako vhodná jiná technika replikace. To však nelze příliš zobecňovat a je potřeba vždy provést analýzu příslušného systému a na teprve základě jejího výsledku doporučit a zvolit techniku, kterou budeme v příslušném systému data replikovat.

4.1 Rozdíl mezi asynchronní a synchronní replikací

Jde o dva extrémní případy, jak replikace nakládá s potvrzením uložení dat. Zatímco u synchronní replikace dojde k potvrzení zápisu do databáze až v ten moment, kdy má master server potvrzení zápis i od všech slave serverů, u asynchronní replikace vrátí master server potvrzení o provedení zápisu okamžitě a až poté zabezpečuje samotnou replikaci dat.



Obrázek 2: Rozdíl mezi synchronní a asynchronní replikací [9]

Z toho jsou patrné i výhody i nevýhody obou přístupů.

- Synchronní replikace – pomalá, ale přesná
- Asynchronní replikace – rychlá, ale s rizikem nekonzistence

Synchronní systém umožňuje přesnou detekci poruch, kdežto v asynchronním systému neexistuje žádná vazba na zpoždění přenosu zpráv a žádná vazba na relativní rychlost procesů. To obvykle modeluje systém s nepředvídatelným zatížením sítě a CPU. Důsledkem je, že v asynchronním systému není nikdy možné zjistit, zda proces zhavaroval nebo ne.

4.2 Synchronní replikace

Synchronní replikace (master-slave replikace) funguje na principu dvoufázového potvrzovacího protokolu. Ve dvoufázovém potvrzovacím protokolu, když je požadována aktualizace hlavní databáze, master site se připojí ke všem ostatním systémům (podřízené databáze), uzamkne tyto databáze na úrovni záznamu a poté je aktualizuje současně. Pokud jeden z podřízených uzlů není dostupný, data nemusí být aktualizována. Konzistence dat je zachována, ale vyžaduje však dostupnost všech uzlů v době šíření aktualizací. [6]

4.3 Asynchronní replikace

Existují dvě varianty asynchronní replikace periodická a neperiodická. V periodické replikaci se aktualizace datových položek provádějí v určitých intervalech, zatímco v neperiodické replikaci se aktualizace šíří pouze v případě potřeby (obvykle na základě triggeru). Čas, kdy jsou kopie nekonzistentní, je nastavitelný parametr, který je závislý na aplikaci.[6]

Dále můžeme metody asynchronní replikace rozdělit podle toho, zda může být šíření aktualizace iniciované kterýmkoli z uzlů nebo je to pouze master site, kdo může vynutit aktualizaci a všechny ostatní uzly jsou podřízené (sekundární) kopie a jsou aktualizovány s ohledem na změny na master site.

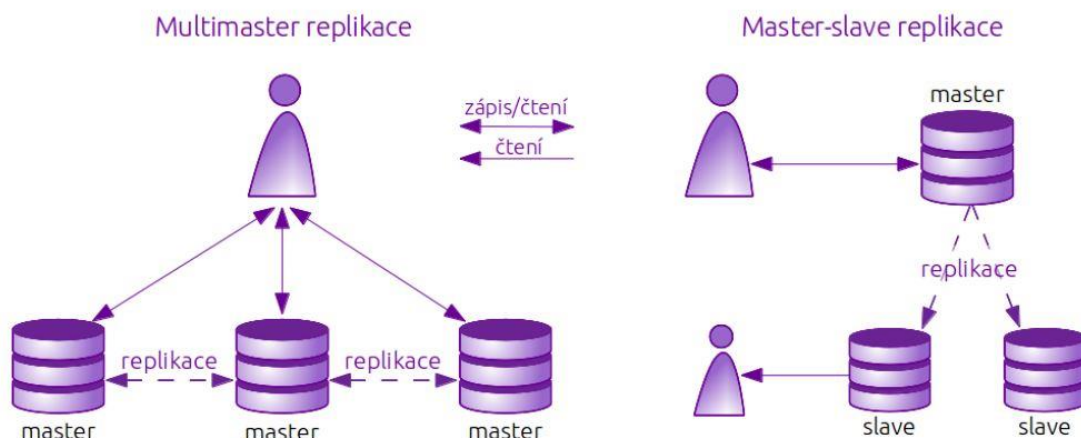
4.4 Částečně synchronní systém

V praxi se často používá systém, který je zpočátku asynchronní, a nakonec se stane synchronním. Čas, kdy se systém stává synchronním, nazýváme Global Stabilization Time (GST). Jde o to, že první potvrzení zápisu jde okamžitě po zápisu do primární databáze. Tím je záznam uvolněn pro další práci. Ovšem master databáze stále očekává od všech replik potvrzení a až poté co dorazí potvrzení z poslední repliky je považována synchronizace záznamu za kompletní.

4.5 Podle architektury

Je-li určena jedna hlavní replika pro zpracování všech požadavků, pak ji říkáme primární a mluvíme o master-slave schématu nebo též nazýváme single leader replikaci. Ten využíváme nejčastěji v clusterech s nutností vysoké dostupnosti.

Na druhé straně, pokud jakákoli replika zpracuje požadavek a poté distribuuje nový stav, pak se jedná o multiprimární schématu (multi-master nebo multi-leader).



Obrázek 3: Srovnání multimaster a master-slave replikace [9]

5. Strategie správy replikací

Replikační protokoly lze klasifikovat podle toho, kdy jsou aktualizace šířeny do replik. To lze dosáhnout synchronními (horlivými) a asynchronními (línými) metodami. A podle toho, kde je aktualizace povolena (centralizované versus distribuované). Tato dvě rozhodnutí se obecně označují jako strategie správy aktualizací.

5.1 Horlivé šíření aktualizací

Anglicky „eager update propagation“. Tato strategie aplikují změny na všechny repliky v kontextu transakce aktualizace. Díky tomu mají všechny replikované záznamy při potvrzení transakce stejnou hodnotu. Horlivé šíření aktualizací používá často synchronní šíření (aplikuje aktualizace na všechny repliky ve stejnou dobu), nebo odložené šíření (aktualizace jsou aplikovány na jednu repliku, když jsou vydány, ale jejich aplikace na ostatní repliky je prováděna dávkově a odložena na konec transakce).

Vzhledem k tomu, že všechny repliky jsou na konci aktualizací transakce vzájemně konzistentní, pro následné čtení může být využita jakákoliv jedna replika. Zápis však musí být proveden na všechny repliky. Proto jsou protokoly, které jsou užívány při horlivém šíření aktualizací známy jako read-one/write-all (ROWA).

Mezi výhody horlivého šíření aktualizací patří:

- Data jednotlivých replik jsou navzájem konzistentní.
- Transakce může číst místní kopii datového záznamu a mít jistotu, že je načtena aktuální hodnota. Není tedy potřeba provádět vzdálené čtení.
- Změny replik jsou provedeny atomicky, takže obnova po selhání může být řízena protokoly.

Hlavní nevýhody horlivého šíření aktualizací plynou z toho, že transakce musí před ukončením aktualizovat všechny kopie. Důsledky jsou:

- Delší doba odezvy, protože rychlost aktualizace je závislá na nejpomalejším uzlu.
- V případě nedostupnosti některé repliky nemůže být transakce ukončena. [3]

5.2 Líné šíření aktualizací

Anglicky „lazy update propagation“. Při líném šíření aktualizací nejsou všechny aktualizace replik prováděny v kontextu transakce aktualizace. To znamená, že transakce před potvrzením nečeká, dokud nebudou její aktualizace aplikovány na všechny repliky a je potvrzena hned, jakmile je aktualizována jedna replika. Šíření do dalších replik se provádí asynchronně pomocí obnovovacích transakcí, které jsou odesílány do zbylých replik se zpožděním po potvrzení transakce aktualizace. Obnovovací transakce nese sekvenci aktualizací odpovídající původní aktualizací transakci.

Líné šíření se užívá v těch aplikacích, pro které není konzistence nejvyšší prioritou. Tyto aplikace mohou být ochotny tolerovat určitou míru nekonzistence mezi replikami výměnou za lepší výkon.

Typickými příklady použití této strategie je služba DNS (Domain Name Service), dále databáze působící na geograficky široce distribuovaných webech (např. Netflix) nebo většina mobilních databází.

Hlavní výhodou strategie líného šíření aktualizací je:

- Kratší doba odezvy pro aktualizací transakce. Aktualizační transakce se může potvrdit, ihned jakmile aktualizuje jednu repliku.

Nevýhody strategie líného šíření aktualizací jsou:

- Repliky nemusí být vzájemně konzistentní.
- Některé repliky mohou být zastaralé, v důsledku čehož nemusí místní čtení obdržet aktuální data. [3]

5.3 Techniky centralizovaného šíření aktualizací

Tyto techniky vyžadují, aby byly aktualizace nejprve aplikovány na hlavní repliku a teprve poté rozšířeny do dalších (podřízených) replik. Uzel, který je hostitelem hlavní repliky, se podobně nazývá master site, zatímco uzly, hostící podřízené repliky, se nazývají slave sites.

Pokud existuje pouze jeden master site pro všechna replikovaná data mluvíme o single master technice. Ovšem v některých protokolech může být master site pro každý datový záznam jiný. Pak mluvíme o primary copy technice.

Výhody centralizovaných technik jsou:

- Aplikace aktualizací je snadná, protože se odehrávají pouze na master site a nevyžadují synchronizace mezi více replikami.
- Vždy alespoň jeden uzel (master site) má aktuální hodnoty pro datové záznamy.
- Tyto protokoly jsou obecně vhodné v datových skladech a dalších aplikacích, kde je zpracování dat centralizováno na jednom nebo několika hlavních místech.

Hlavní nevýhodou centralizovaných technik je:

- Existuje jeden centrální uzel, který hostí všechny master sites.
- Ten může být přetížen a stát se úzkým místem.

Rozdělení odpovědnosti master site za každý datový záznam (jak tomu je u primary copy technik) je jedním ze způsobů, jak tuto režii snížit, ale vyvolává zase problémy s konzistencí dat, zejména s ohledem na zachování globální serializace v technikách líné replikace, protože obnovovací transakce musí být prováděny na replikách ve stejném pořadí serializace. [3]

5.4 Distribuované techniky

Distribuované techniky aplikují aktualizaci nejprve na místní repliku uzlu, odkud transakce aktualizace pochází, a poté se aktualizace rozšíří na další replikované uzly. Tyto techniky se nazývají distribuované proto, že různé transakce mohou aktualizovat různé repliky stejného datového záznamu umístěného na různých uzlech. Jsou vhodné pro aplikace spolupracující s distribučními rozhodovacími nebo operačními centry. Mohou rovnoměrněji rozložit zátěž, díky čemuž mohou nabídnout maximální dostupnost systému, zvláště jsou-li propojeny s technikami líného šíření.

Závažnou komplikací, která v těchto systémech vzniká je, že různé repliky datového záznamu mohou být aktualizovány na různých místech (masterech) současně. Pokud jsou distribuované techniky propojeny horlivými metodami šíření, pak může nastat problém souběžných aktualizací. Pokud se použijí metody líného šíření, transakce mohou být prováděny na různých místech v různém pořadí, což může způsobit nesoulad v synchronizaci replik. Ke zvládnutí těchto problémů se používá metoda, která zahrnuje vrácení a opakování transakcí takovým způsobem, že pořadí provedení transakce je na každém místě stejné. To ovšem není snadný úkol, a jeho řešení bývá obecně závislé na aplikaci. [3]

6. Formy replikačních strategií

6.1 Snímková replikace

Při replikaci snímků je snímek (snapshot) nebo kopie dat pořízena z jednoho serveru a přesunuta na jiný server nebo do jiné databáze na stejném serveru. Po počáteční synchronizaci může replikace snímků pravidelně obnovovat data v publikovaných tabulkách. Ačkoli je replikace snímků nejjednodušší formou replikace, vyžaduje zkopírování všech datových položek při každé aktualizaci tabulky.

6.2 Transakční replikace

Při transakční replikaci sleduje agent replikace změny v databázi na serveru a přenáší tyto změny na ostatní servery. Tento přenos může probíhat okamžitě nebo opakovaně ve stanovených intervalech. Transakční replikace se používá pro scénáře server-server.

6.3 Slučovací replikace

Slučovací replikace umožňuje replikám pracovat nezávisle. Obě entity mohou pracovat off-line. Když jsou připojeni, agent slučovací replikace zkontroluje změny v obou sadách dat a podle toho upraví každou databázi. Pokud dojde ke konfliktu transakcí, použije předdefinovaný algoritmus řešení konfliktů k dosažení konzistence. Slučovací replikace se používá většinou v bezdrátových prostředích.

6.4 Replikace založená na SQL příkazech

Replikace založená na příkazech zachytí každý SQL příkaz (insert, update, delete) a odešle jej do různých replik. Každá replika funguje samostatně. K vyřešení konfliktů se příkaz pro zápis zasílají na všechny servery, zatímco lze dotazy pouze pro čtení odesílat pouze na jeden server. To umožňuje rozložení zátěže čtení. Replikace založená na příkazech je použitelná pro optimistické přístupy, kde každá mezipaměť udržuje stejnou repliku.

7. Nástroje pro replikace

Existuje mnoho nástrojů pro replikaci databází, ať už se jedná o komerční nástroje či Open Source produkty. Uvedeny jsou zde nejpoužívanější z nich a některé jsou detailněji rozebrány dále.

Komerční nástroje:

- DB2 Data Propagator
- Sybase Replication Server
- IBM Informix Replication
- Oracle's Stream

- Q-Replication Tools
- Helios Green – Sestavy relikace

Open Source produkty

- Slony
- Postgres-R
- ESCADA Replication Server
- Pgpool-II
- DB Replicator

7.1 Slony-I

Slony-I je replikační systém podporující kaskádou propagaci replik z jednoho master serveru. Slony-I je systém schopný replikovat velké databáze na omezený počet podřízených uzlů. Pokud počet serverů překročí určitý limit, mohou se zvýšit náklady na komunikaci. Účelem Slony-I je udržovat datová centra a úložiště zálohovacích systémů s běžným zabezpečením a stálou dostupností. Tam, kde jsou uzly systému náchylné k častým poruchám, není Slony-I tím lepším řešením pro replikaci.

Systém Slony-I replikuje objekty (tabulky, pořadová čísla atd.) logicky seskupeny do sad. Každá sada je nezávislá na jiné, která pochází ze stejného master site. Tabulky, jejich vztahy a sekvence používané ke generování sériových čísel jsou seskupeny do sad.

7.2 Postgres-R

Postgres-R je rozšířením databáze PostgreSQL, která poskytuje efektivní replikaci databáze pro clustery při zachování konzistence. Je navržen tak, aby byl pro klienta maximálně transparentní, stabilní a bezpečný. Postgres-R se primárně používá pro vyrovnávání zátěže a vysokou dostupnost databázových systémů. Pružná architektura Postgres-R umožňuje škálovat proces replikace. Dále ve srovnání s centralizovaným systémem je Postgres-R spolehlivější a flexibilnější.

Postgres-R dokáže replikovat velké objekty a používat datová úložiště. Použití „Group communication System“ od Postgres-R pomáhá při přidávání nebo odebírání uzlů, a to

i během provozu systému. Havarované uzly jsou detekovány a odstraněny automaticky bez ovlivnění provozu databázového systému.

Provádění transakcí v Postgres-R využívá ROWA přístup a je rozděleno do čtyř fází. V místní fázi se transakce provádí v místní replice (místní databázi). Ve fázi odesílání se aktualizace dat šíří do replik. Ve fázi synchronizace je zachována řaditelnost transakcí. Ve fázi zápisu vzdálené repliky se provádí zápisy a příslušné repliky zpracovávají transakce.

7.3 Pgpool-II

Pgpool-II je middleware, který funguje jako rozhraní mezi klientem databáze PostgreSQL a jeho servery. Protože Pgpool-II je middleware, databázová aplikace (klient) má pocit, že Pgpool-II je ve skutečnosti Postgres server. Naopak server vizualizuje Pgpool-II jako jednoho ze svých klientů.

Pgpool-II spravuje více serverů. Záloha se provádí na dva nebo více disků. Replikační funkce Pgpool-II pomáhají k dostupnosti systému tím, že poskytují účinný mechanismus odolnosti proti chybám.

Když klient požaduje velké množství operací čtení, Pgpool-II efektivně implementuje rozložení zátěže distribucí dotazů na více serverů. To pomáhá při provádění více dotazů současně. Protože požadovaná operace je operace čtení, každý server vrátí stejný výsledek. Existuje však limit pro maximální počet klientských připojení ve stejnou dobu.

7.4 Replikace na MS SQL Serveru

Replikace je nejstarší z technologií, které na SQL Serveru zajišťují vysokou dostupnost. Replikace představená v SQL Server 6 synchronizuje data mezi instancemi SQL Serveru, obvykle kopíruje data z vydavatele (Publisher), který je zdrojem dat, do odběratele (Subscriber), který je hostitelem zkopírovaných dat. Pokud vydavatel selže, může odběratel převzít datové operace.

Na rozdíl od jiných technologií vysoké dostupnosti poskytuje replikace správci databázi plnou kontrolu nad tím, kolik dat by se mělo replikovat. DBA si může vybrat, které tabulky replikovat z vydavatele na jednoho nebo více odběratelů, a dokonce rozhodnout, které sloupce v těchto tabulkách replikovat.[2]

7.5 IBM Informix Replication

IBM Informix Replikace je nástroj, která replikuje data z pomoci asynchronního protokolu mezi databázovými servery Informix Dynamic. Transakce zachycené na zdrojovém serveru přečtením logu, jsou přeneseny na cílový server. Na cílovém serveru replikace přijímá a aplikuje všechna transakční data do příslušných databázových tabulek a zaznamenává aktuální stav transakce.

Tato replikace využívá líné šíření dat, aktualizuje repliku ve chvíli, kdy master databáze potvrdí změny. Zpoždění aktualizace dat na jednotlivých sekundárních uzlech se může lišit, nakonec jsou však synchronizovány všechny uzly na konzistentní hodnoty. Výhodou tohoto přístupu je, že v případě selhání proces replikace pokračuje a všechny transakce v replikačním systému budou potvrzeny.

7.6 Sybase Replication Server

Také Sybase má své řešení, které umožňuje sdílení a replikaci dat mezi databázemi. Sybase Replication server řídí operace distribuce dat, což pomáhá snižovat zátěž správců databází a čímž připravuje cestu k plnění obchodních cílů.

Sybase používá tři komponenty k dosažení řešení problémů s distribucí dat,

- databáze SQL Anywhere
- agent replikace pro SQL Anywhere
- SQL remote distribution solution

Replikační server Sybase je sada nástrojů, které přesouvají a synchronizují data z jednoho serveru na druhý. Zajišťuje nejen dostupnost dat, ale také rychlou obnovu dat v případě selhání. Integruje a synchronizuje operace prováděné na více místech.

7.7 Replikace v systému Helios

Poněkud odlišně zachází s replikacemi systém Helios. Replikace jsou zde přímo integrovány do samotného systému. V prezentační vrstvě lze nastavit rozsah a směr replikací. Zjistit, jak se toto nastavení promítá do databázových tabulek a aplikační vrstvy a jakým způsobem by bylo možno zakomponovat získané teoretické znalosti ke zlepšení funkčnosti replikací bude cílem této práce.

8. Problémy při replikaci

Při využití replikací může dojít k několika typům problémů, které se liší podle použitých způsobů replikace. Např. pokud některý cílový uzel není dostupný v čase replikace, pak při použití synchronní nepůjde ukončit transakci replikace, zatímco u asynchronní replikace bude standardně transakce replikace ukončena, ale vznikne nekonzistence mezi jednotlivými replikami.

V systémech s distribuovanými databázemi mohou (kromě problémů vznikajících v systémech s centrálními databázemi) vznikat ještě další problémy. Jsou to zejména:

- Manipulace s vícenásobnými kopiemi datových položek – existují metody, které jsou zodpovědné za řízení přístupu k jednotlivým záznamům. Musí v nich být jednoznačně definováno, jak se systém zachová v případě, kdy je manipulováno s daty ve více uzlech současně. Cílem bývá udržení konzistence dat.
- Selhání jednotlivých uzlů – v případě selhání některého uzlu je možné pokračovat v provádění operací, ale před znovuzapojením dotčeného uzlu zpět do replikační infrastruktury je potřeba provést opětovnou synchronizaci.
- Selhání komunikace (sítě) – v extrémním případě je řešení stejné, jako u selhání uzlu, ale v některých konkrétních případech je možné, aby se s tímto problémem systém vypořádal sám díky jinému nastavení komunikace.
- Selhání potvrzení – může se stát, že přestože transakce proběhla, selhalo její potvrzení. Tento problém bývá řešen takzvaným dvoufázovým potvrzovacím protokolem (1. fáze – záznam zapsán, 2. fáze – záznam zapsán na všech replikách)

- Distribuovaný deadlock – nepříjemností replikace je, deadlock se může v některých případech vyskytnout i mezi více uzly a je tak potřeba tomu přizpůsobit techniky, které deadlocku zabráňují.

9. Fragmentace replikace

V distribuovaných systémech je řešeno i to, jak ideálně rozdělit data na jednotlivá místa. Existují dvě základní strategie fragmentace – horizontální a vertikální. Jejich kombinací pak vzniká třetí možnost – vnořování fragmentů hybridním způsobem.

9.1 Horizontální fragmentace

Horizontální fragmentace rozděluje data do n-tic podle konkrétní vybrané hodnoty některého z atributů nebo kombinací hodnot několika atributů. Existují dvě verze horizontálního rozdělení: primární a odvozené. Primární horizontální fragmentace relace se provádí pomocí predikátů, které jsou na dané relaci definovány. Odvozená horizontální fragmentace je na druhé straně rozdělení vztahu, které vyplývá z toho, že predikáty jsou definovány v jiném vztahu.[10]

9.2 Vertikální fragmentace

Vertikální fragment obsahuje pouze vybrané atributy (sloupce tabulky). Takže používáme-li vertikální fragmentaci při replikaci, pak kopírujeme do cílových replik jen vybrané atributy záznamů tabulek z master site.

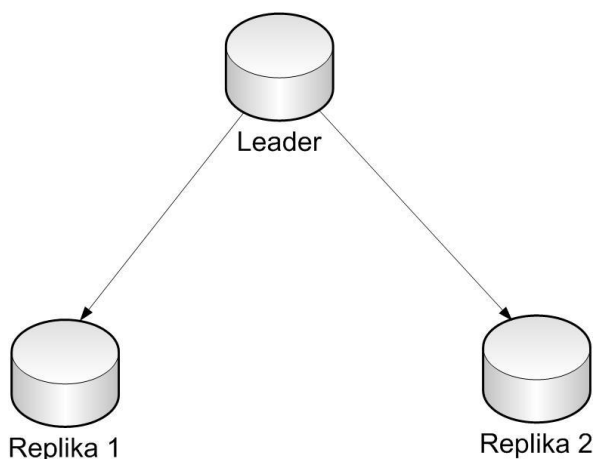
9.3 Hybridní fragmentace

Je kombinací horizontální a vertikální fragmentace. To znamená, že jejím výsledkem část dat filtrovaná podle určitých atributů, o které jsou přenášeny pouze vybrané informace.

Praktická část

10. Cíle práce

Cílem práce je dopodrobna kompletně nastavit a otestovat replikace v systému Helios Green. Provéřit, jak se toto nastavení promítá do databázových tabulek a aplikační vrstvy. K tomu bude vytvořeno modelové replikační prostředí se třemi databázemi. Po otestování replikace bude nejen určeno, jakou techniku, formu a strategii replikace v systému Helios Green dokážou využít, ale též rozmyslet, zda je možno s úspěchem aplikovat některé replikační nástroje popsané v teoretické části.



Obrázek 4: Schéma propojení databází

Pro otestování funkce replikace byla vytvořena jedna databáze s názvem „Leader“ a její kopii pak další dvě databáze s názvy „Replika 1“ a „Replika 2“.

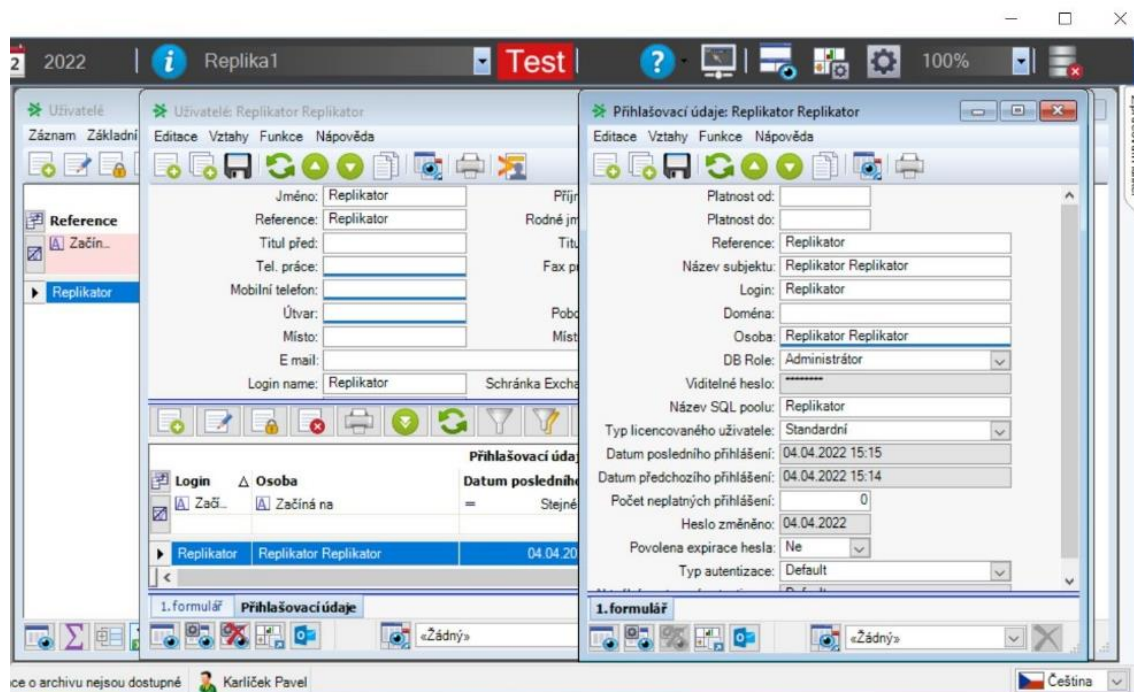
11. Prezentační vrstva

11.1 Nastavení prostředí na PLDB

Vzhledem k tomu, že budeme používat master slave architekturu nastavíme větší část potřebných údajů na zdrojové databázi Leader. Nejprve však musíme na databázích replik zjistit či vytvořit některá nastavení.

11.1.1 Uživatel pro replikace

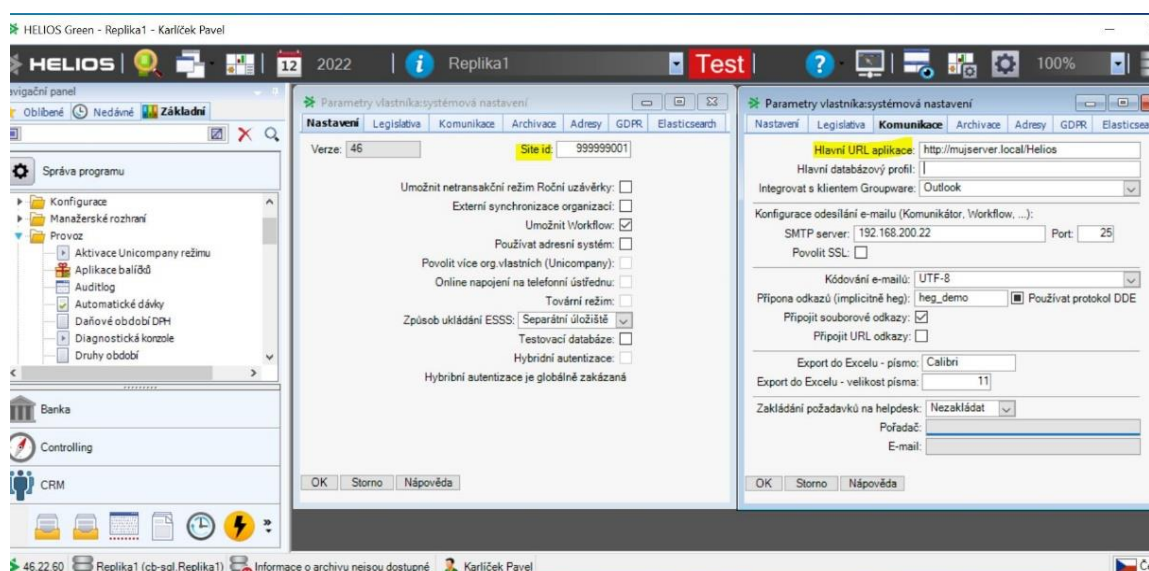
Je potřeba vytvořit technické uživatele, kterými se replikace budou přihlašovat podřízeným lokálním databázím (PLDB). Na obou PLDB (Replika 1 i Replika 2) tedy vytvoříme uživatele s loginem „replikator“ a příslušnými přihlašovacími údaji. Tento uživatel musí mít práva insert, update a delete programem na cílových databázích.



Obrázek 5: Uživatel a přihlašovací údaje pro replikaci

11.1.2 Site ID a URL aplikace

Také je potřeba zjistit Site ID což je unikátní číslo databáze používané systémem Helios Green, které slouží (nejen) k jednoznačné identifikaci databáze při replikacích. To je zjištěno v modulu Správa programu, složce Provoz, v konfiguraci Parametry vlastníka: systémová nastavení, na záložce Nastavení.

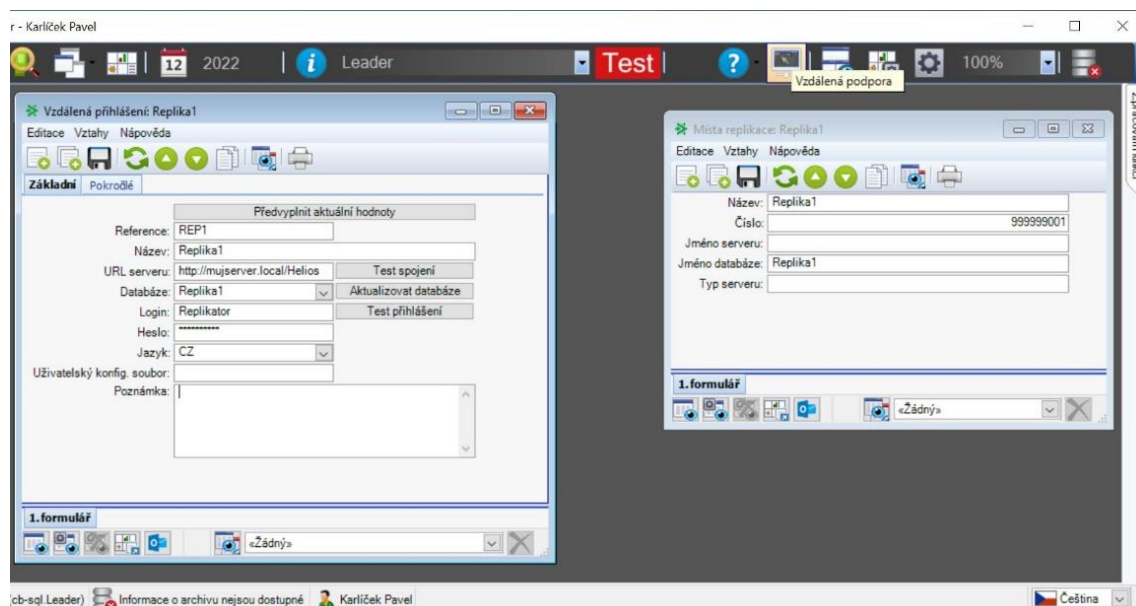


Obrázek 6: Parametry vlastnicka: systémová nastavení

Tamtéž jen na záložce komunikace najdeme další potřebný atribut „URL aplikace“.

11.2 Nastavení prostředí na zdrojové databázi

11.2.1 Místa replikace



Obrázek 7: Místa replikace a Vzdálená přihlášení

V pořadači místa replikace je potřeba založit záznamy pro všechny uzly, mezi kterými bude replikace probíhat. To znamená jak zdrojové (MDB), tak i cílové databáze (PLDB).

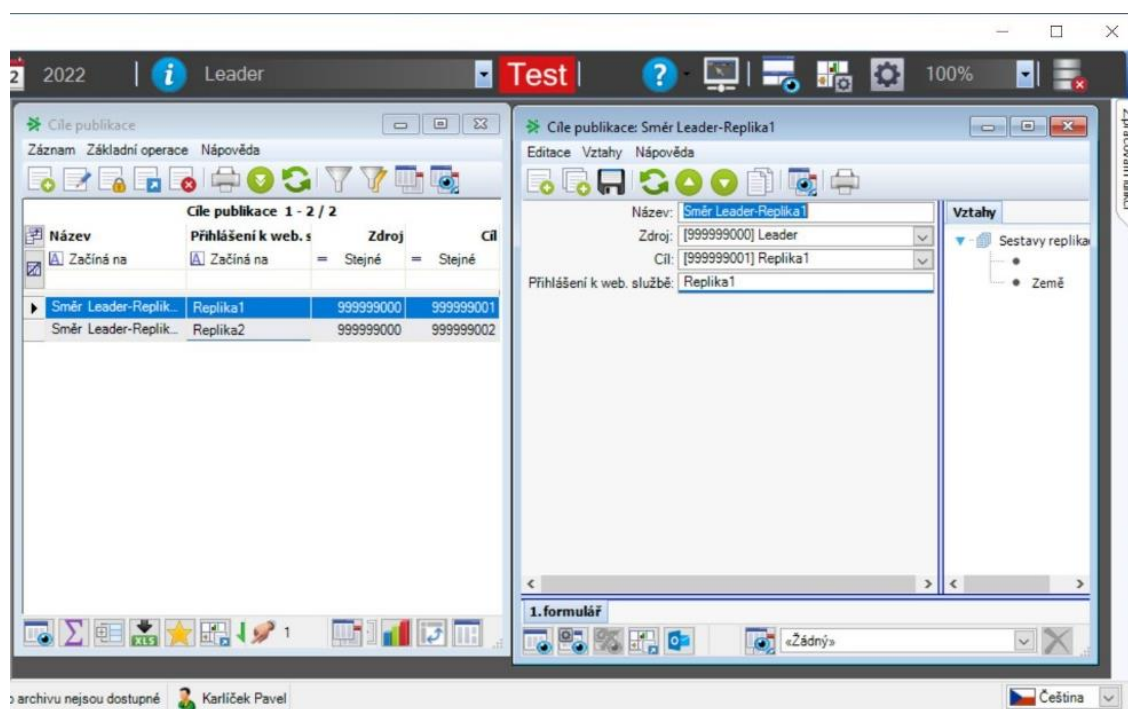
Důležitým atributem záznamů v pořadači „Místa replikace“ je atribut „číslo“, ve kterém je uvedeno Site ID příslušného místa replikace.

11.2.2 Vzdálená přihlášení

Replikaci v systému Helios lze provádět dvěma způsoby. Buď lze využít webovou službu nebo lze zvolit kopírování. Pokud zvolíme webovou službu (což je i náš případ) pak je potřeba mít vyplněny záznamy v pořadači vzdálená přihlášení pro všechny cílové databáze (PLDB). Každý takový záznam musí obsahovat kromě názvu a reference i URL serveru, login a heslo (mělo by být hashované) pro přihlášení do PLDB.

11.2.3 Cíle publikace

V pořadači s tímto názvem se nastavují jednotlivé směry replikace. Atribut záznamu s názvy „Zdroj“ a „Cíl“ jsou vztahy do pořadače Místa replikace, zatímco pole „Přihlášení k webové službě“ je vztahem do pořadače vzdálená přihlášení, kde musí být zadáno konkrétní vzdálené přihlášení k cílové databázi. V dynamickém vztahu je možno vidět, ve kterých sestavách replikace je záznam použit.



Obrázek 8: Cíle publikace a vzorový záznam

11.3 Nastavení rozsahu replikace

Zatímco doposud jsme nastavovali prostředí pro replikace tak, aby tyto mohly fungovat, nyní se zaměříme na určení rozsahu replikovaných dat.

11.3.1 Subjekt, pořadač a třída

K tomu je zapotřebí si nejprve definovat pojmy pořadač a třída a specifikovat jaký je mezi nimi rozdíl. Další důležitý termín pro pochopení architektury Heliosu je subjekt.

- Třída – je obraz databázové tabulky v prezentační vrstvě Heliosu. Obsahuje vždy alespoň jeden, často však více pořadačů. Ve třídě jsou vždy umístěny všechny záznamy databázové tabulky. Třídy dělíme na subjektové (jejich záznamy mají svůj obraz v tabulce subjektů) a nonsubjektové (záznamy těchto tříd nemají obraz v tabulce subjektů; obsahují právě jeden pořadač).
- Pořadač – je horizontálně fragmentovaná část třídy. Atribut, podle kterého dochází k fragmentaci je číslo pořadače a je umístěn v tabulce subjektů (u nonsubjektových tříd je počet záznamů v pořadači shodný s počtem záznamů ve třídě). Každý záznam v databázi Helios je přiřazen právě jednomu pořadači.
- Subjekty – název největší (co do počtu řádků) tabulky v Heliosu, která obsahuje omezené informace o všech záznamech všech subjektových tříd Heliosu. Atributy v tabulce subjekty jsou: číslo subjektu, název, reference, číslo pořadače, datum vzniku, autor, datum poslední aktualizace, pachatel poslední změny, období platnosti. Díky tomu, že číslo subjektu je unikátní a je přeneseno do záznamů tabulek konkrétních tříd, využívá se s výhodou pro propojování jednotlivých tabulek.

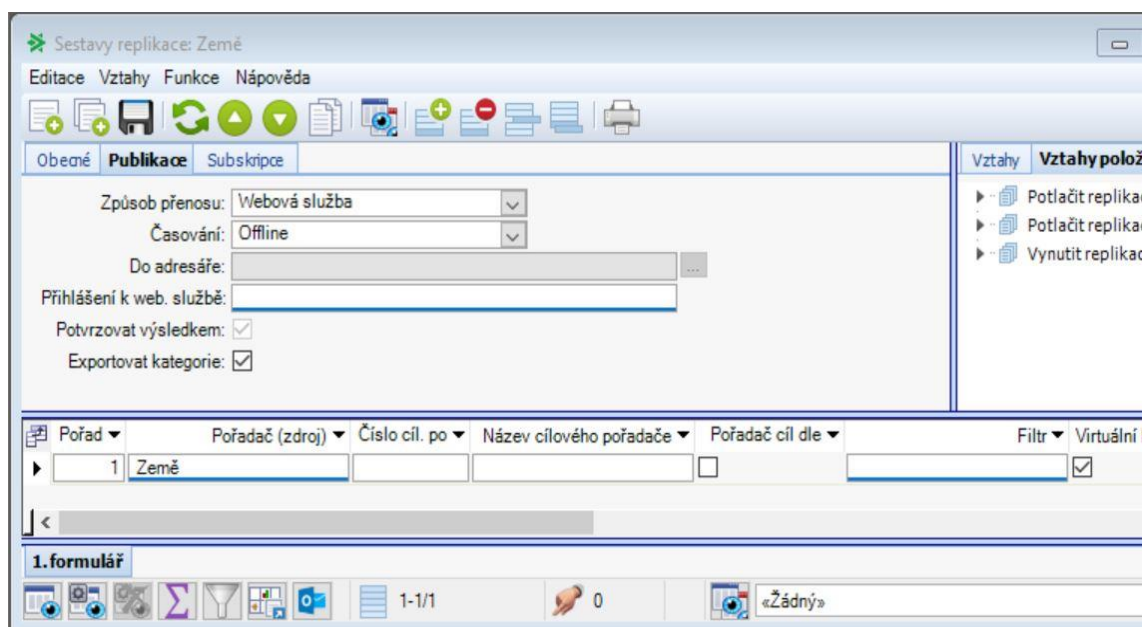
11.3.2 Sestavy replikace

Každý záznam sestavy replikace obsahuje atributy, které určují chování příslušné replikace. Hlavičkové atributy se týkají sestavy replikace jako celku, zatímco položkové se vztahují k replikaci konkrétního pořadače.

Na záložce *Publikace* jsou tyto atributy:

Typ přenosu – určuje, jak probíhá přenos replikovaných dat. Možnosti jsou:

- Webová služba – využity cíle publikace, resp. vzdálená přihlášení (preferovaný – automatický typ přenosu)
- Kopírování – využít atribut „Do adresáře“, kde je uvedena cesta k vygenerovanému XML souboru (lze využít pro otestování, jak bude XML vypadat)



Obrázek 9: Sestava replikace – záložka publikace

Časování – určuje kdy k replikaci dojde

- Online – k replikaci dochází okamžitě po uložení záznamu
- Offline – k replikaci dochází při spuštění funkce vytvoření repliky (ta může být spuštěna uživatelem, častěji je však spouštěna pravidelným jobem běžícím pod Agentem v MS SQL Managemet studiu)

Do adresáře – souvisí s atributem Typ přenosu – pokud je zvoleno Kopírování (jinak je editace atributu potlačena)

Přihlášení k webové službě – je vztahem do pořadače vzdálená přihlášení (je zvoleno vzdálené přihlášení zdrojové databáze, využívá se jen v případě typu přenosu – webová služba)

Potvrzovat výsledkem – tento přepínač je užíván při přenosu kopírováním a určuje, zda potvrzení úspěšného replikování záznamu závisí na zpětném logu.

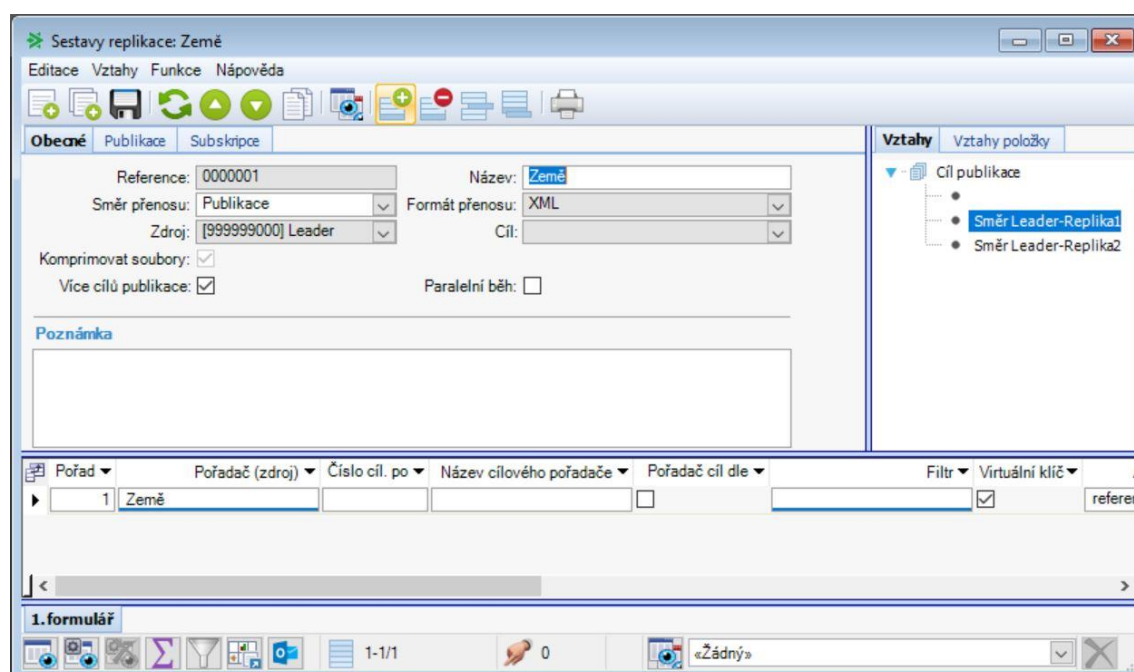
Exportovat kategorie – přepínač umožňuje vypnout zápis informací o kategoriích (má smysl, pokud se struktura kategorií ve zdrojové a cílové databázi liší)

Na záložce obecné jsou tyto atributy:

Reference, název – slouží k identifikaci sestavy replikace

Směr přenosu – při způsobu přenosu „kopírování“ je potřeba nastavit tutéž sestavu replikace na obou uzlech. Na zdrojovém bude směr „Publikace“ a v cílovém „Subskripce“. Při použití webové služby je směr přenosu vždy subskripce.

Formát přenosu – jediným podporovaným formátem je v současné době XML, jak vypadá a funguje bude předvedeno dále.



Obrázek 10: Sestava replikace – záložka obecné

Zdroj a cíl – jsou vztahy do pořadače „Místa replikace“ a určují odkud a kam se budou záznamy definované v položkách replikace přenášet.

Komprimovat soubory – přepínač umožní kompresi souborů při tomto typu přenosu.

Více cílů publikace – při zaškrtnutí tohoto atributu je možné zadat v jedné sestavě replikace více cílů publikace pomocí dynamického vztahu „Cíle publikace“.

Paralelní běh – přepínač umožní současné odeslání záznamů definovaných sestavou replikace do všech cílů publikace.

Poznámka – atribut pro volný text upřesňující informace o sestavě replikace.

11.3.3 Položky sestavy replikace

V položkové části formuláře sestavy replikace (jde o spodní část) jsou definovány rozsahy replikovaných dat. V každé sestavě replikace musí být definována alespoň jedna položka (pořadač), aby měla sestava nějaký smysl. Atributy položek jsou:

Pořadí – určuje v jakém pořadí sestava replikace přistupuje k přenášným datům

Pořadač (zdroj) – určuje pořadače, jejichž data jsou replikovány. Tím, že jde o pořadač a nikoliv třídu, může již zde docházet k horizontální fragmentaci.

Číslo cílového pořadače, Název cílového pořadače – slouží k identifikaci pořadače v cílové databázi. Pokud není žádný z těchto atributů vyplněn, jsou data replikována do pořadače se stejným číslem, jako má zdrojový pořadač.

Pořadač cíle dle syncID – také užíváno k identifikaci pořadače, a to v případě, že konfigurace pořadače byla přenesena za pomoci přenosu konfigurace a díky tomu má vyplněnu shodnou hodnotu atributu syncID.

Filtr – zde lze nastavit horizontální fragmentaci replikace za pomoci definovaného filtru.

Virtuální klíč – přepínač se užívá, pokud je potřeba navázat záznamy cílové databáze podle konkrétního unikátního klíče. Není-li použit virtuální klíč, používá se GUNID (ten je generován při první replikaci každého záznamu a uložen do tabulky repl_proxy).

Atributy klíče – je-li zaškrtnut předcházející atribut, pak je zde uveden atribut či atributy pro párování záznamů mezi zdrojovou a cílovou databází.

Virtuální klíč položek – přepínač se užívá, pokud je potřeba navázat položky záznamu cílové databáze podle konkrétního unikátního klíče.

Atributy klíče položek – je-li zaškrtnut předcházející atribut, pak je zde uveden atribut či atributy pro párování položek záznamů mezi zdrojovou a cílovou databází.

Zpracování – tímto atributem jsou definovány možnosti zpracování položek při replikaci položkových tříd.

- Nahrazování – položka se smaže a znovu založí (defaultní nastavení)
- Synchronizace – položka se synchronizuje pomocí virtuálního klíče položky

Transformace – vztah byl navržen z důvodu použití XSL šablon pro transformace dat při integraci Heliosu s jinými IS.

Akce před a po replikaci – vazba na funkci, která se spustí nad replikovaným záznamem před nebo po provedení replikace.

Prefix názvu výstupního souboru – pokud není zadán (výchozí stav) bude jméno generováno systémem. V prefixu lze použít: ":reference_subjektu", ":nazev_subjektu", ":main_document".

11.3.4 Konkrétní nastavení sestavy replikace

Bylo zde popsáno spoustu atributů pro nastavení replikace. V zásadě je potřeba nastavit jen některé a větší část lze nechat ponechánou nevyplněnou či v defaultním nastavení. Konkrétně ve většině případů postačí mít vyplněno těchto několik atributů.

- Název
- Směr
- Více cílů publikace – zatrženo a navázáno v dynamickém vztahu
- Způsob přenosu – webová služba
- Časování
- Pořadač (zdroj)

Zvláště při prvotní replikaci bývá využit virtuální klíč.

Pokud potřebujeme horizontální fragmentaci, pak „Filtr“ a pokud vertikální, pak lze využít dynamické vztahy položek „potlačit replikaci vztahu“, „potlačit replikaci atributu“ či „vynutit replikaci vztahu“. Pokud vertikální fragmentace není nastavena, replikují se všechny atributy a ze vztahů jen ty, které už byly replikovány předtím, tj. mají stejný GUNID na obou databázích.

12. Databázová vrstva

Pro jednodušší vysvětlení principu replikace na databázové vrstvě byla nastavena jediná sestava replikace (viz. obrázek 7) s jediným replikovaným pořadačem „Země“ (číslo pořadače 138).

SQLQuery1.sql - CB...TEMS\pavelk (122)*

```
select * from lcs.zeme
```

	cislo_subjektu	plny_nazev	implicitni_jazyk	celni_kod
1	25	Česká republika	21696	NULL
2	26	Austrálie	NULL	NULL
3	27	Rakousko	NULL	NULL
4	28	Belgie	NULL	NULL
5	29	Německo	NULL	NULL
6	30	Bulharsko	NULL	NULL
7	32	Dánsko	NULL	NULL
8	33	Finsko	NULL	NULL
9	34	Francie	NULL	NULL
10	35	Velká Británie	NULL	NULL
11	36	Řecko	NULL	NULL
12	37	Maďarsko	NULL	NULL
13	38	Irsko	NULL	NULL
14	39	Itálie	NULL	NULL

Obrázek 11: Databázová tabulka "zeme"

Tabulka Země obsahuje 16 atributů, z níž první a zároveň primární klíč je číslo subjektu, a v našem případě 247 záznamů. Podle toho, co bylo zmíněno v kapitole 10.3.1. by měly být tytéž záznamy uvedeny i v tabulce *subjekty*. To lze ověřit dotazem do tabulky *subjekty* s podmínkou na konkrétní číslo pořadače.

SQLQuery1.sql - CB...TEMS\pavelk (122)*

```
select * from lcs.subjekty
where cislo_poradace = 138
order by cislo_subjektu
```

	cislo_subjektu	reference_subjektu	nazev_subjektu	cislo_poradace
1	25	420	CZ	138
2	26	61	AU	138
3	27	43	AT	138
4	28	32	BE	138
5	29	49	DE	138
6	30	359	BG	138
7	32	45	DK	138
8	33	358	FI	138
9	34	33	FR	138
10	35	44	GB	138
11	36	30	GR	138

Obrázek 12: Záznamy zemí v tabulce subjektů

Čísla subjektu u subjektových tříd jsou opravdu unikátním primárním klíčem a zároveň vazbou mezi libovolnou subjektovou třídou a tabulkou subjektů.

12.1 Globální identifikátor záznamu

Globální identifikátor záznamu – GUNID – je unikátním identifikátorem záznamu přes více databází a používá se pro identifikaci záznamů, a to nejen pro účely replikace, ale i pro offline export a import. Je v něm zakódováno číslo třídy, číslo záznamu a číslo zdrojové databáze (*site_id*). Zatímco číslo subjektu je unikátním v každé konkrétní databázi, GUNID je unikátní napříč databázemi. Resp. pokud ve dvou databázích existuje záznam se stejným GUNID, pak se jedná o týž záznam a byl do jedné databáze replikován. Tento identifikátor je uveden v tabulce *rpl_proxy*.

Pokud tedy s replikací mezi konkrétními databázemi začínáme, je důležité si tyto tabulky zkontrolovat a případně pročistit od záznamů vzniklých historicky např. offline exportem nebo importem. Dotazy pro zjištění replikovaných záznamů:

```
SELECT *
FROM lcs.rpl_state AS s
JOIN lcs.rpl_proxy AS p ON p.gunid = s.gunid
    AND s.site_id = /* site id */
WHERE p.cislo_tridy = /* číslo třídy */

SELECT *
FROM lcs.rpl_proxy WHERE cislo_tridy = /* číslo třídy */

SELECT *
FROM lcs.rpl_state WHERE folder_number = /* číslo pořadače */
```

Výraz pro vymazání informací o replikaci:

Na zdrojové databázi:

```
DELETE
FROM lcs.rpl_state
WHERE folder_number = /* číslo pořadače */
```

Tím se automaticky smažou i záznamy v *rpl_proxy* (Trigger).

Na cílové databázi:

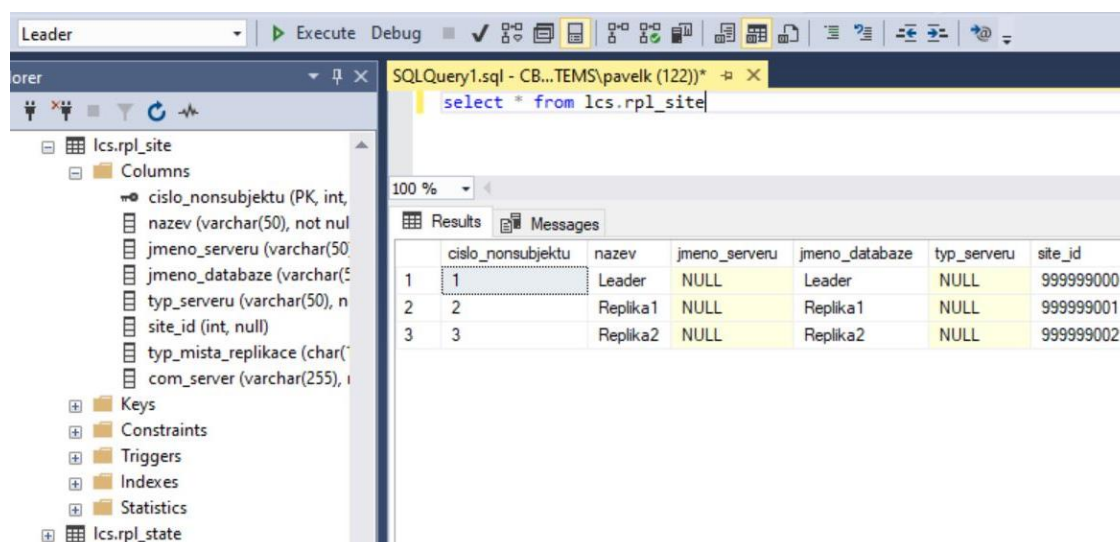
```
DELETE  
FROM lcs.rpl_proxy  
WHERE cislo_tridy = /* číslo třídy */
```

V *rpl_state* na cílové databázi záznamy nejsou.

Replikace (a operace Offline export/import) GUNID na cílové straně zakládá/ aktualizuje v rámci zpracování záznamu, který je vkládán/aktualizován prostřednictvím zprávy INSERTUPDATE. V případě, že je záznam dohledáván použitím klíčových atributů a element RECORD obsahuje GUNID, který je rozdílný než existující GUNID v tabulce *rpl_proxy* pro daný replikovaný záznam, je tento existující GUNID přepsán GUNIDem ze zdrojové databáze. Společně s tímto přepisem GUNIDu je také aktualizován GUNID u příslušných záznamů v tabulce *rpl_state*.

12.2 Tabulky nastavení propojení

V tabulkách s nastavením propojení na databázové vrstvě lze zkontrolovat správnost nastavení. První z nich *lcs.rpl_sites* spravuje záznamy vytvořené v pořadači místa replikace.



The screenshot shows a SQL client interface. On the left, the 'Columns' tab for the *lcs.rpl_sites* table is expanded, showing fields: *cislo_nonsubjektu* (PK, int), *nazev* (varchar(50), not null), *jmeno_serveru* (varchar(50)), *jmeno_databaze* (varchar(50)), *typ_serveru* (varchar(50), not null), *site_id* (int, null), *typ_mista_replikace* (char(1)), and *com_server* (varchar(255)). The 'Results' tab shows the following data:

	cislo_nonsubjektu	nazev	jmeno_serveru	jmeno_databaze	typ_serveru	site_id
1	1	Leader	NULL	Leader	NULL	999999000
2	2	Replika1	NULL	Replika1	NULL	999999001
3	3	Replika2	NULL	Replika2	NULL	999999002

Obrázek 13: Tabulka *rpl_sites* se záznamy míst replikace

Je zde vidět všechny tři vytvořené záznamy pořadače míst replikace, tedy jak zdroj, tak i obě repliky. Jeden z těchto záznamů je možno vidět na Obrázek 7 stejně jako záznam z další tabulky *remote_login*.

	cislo_subjektu	server_url	db_profile	login	password	language	note	user_config	application_service_location	redirection_service_location
1	132158	http://cb-app.poprnsystems.local/Helios	Replika1	Replikator	Replika1	CZ				
2	132159	http://cb-app.poprnsystems.local/Helios	Replika2	Replikator	Replika2	CZ				

Obrázek 14: Tabulka *remote_login* se záznamy vzdálených přihlášení

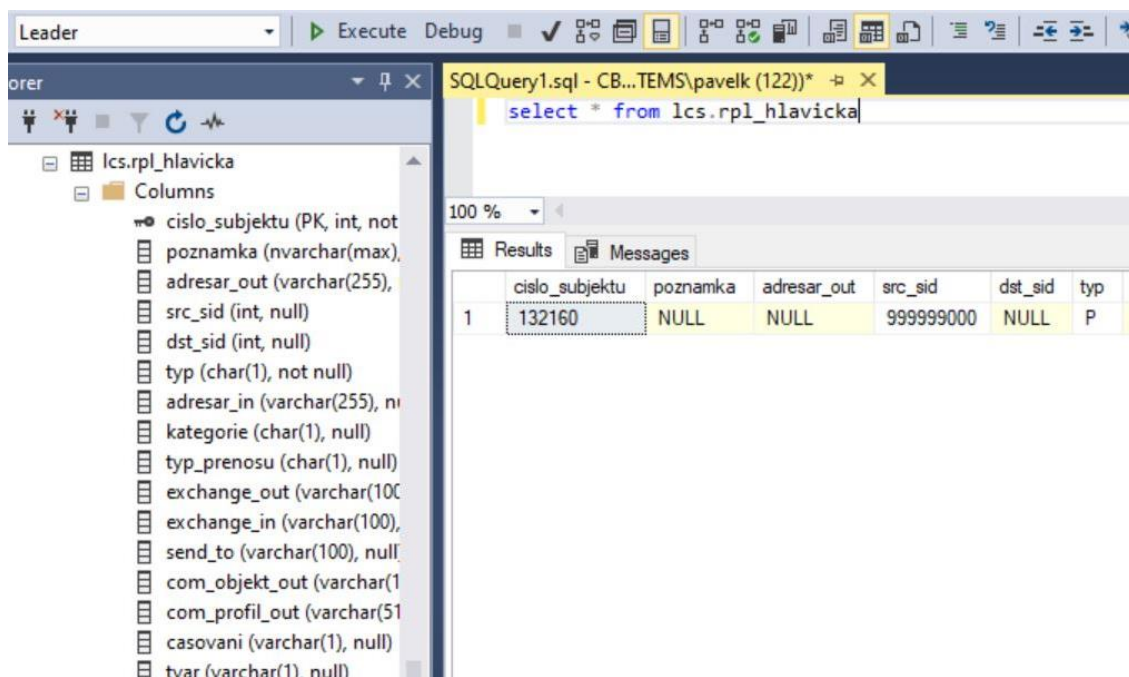
Třetí tabulkou zabezpečující propojení mezi zdrojovou a cílovými databázemi, je tabulka *publication_target*. Ta obsahuje záznamy uvedené na obrázku 6. Má ještě jedno specifikum, a to, že se jedná o tabulku nonsubjektovou (její záznamy nemají obraz v tabulce subjektů, nemají tedy číslo subjektu, nýbrž číslo nonsubjektu – to začíná jedničkou).

	cislo_nonsjektu	cislo_poradace	src_sid	dst_sid	remote_login	nazev
1	1	7204	999999000	999999001	132158	Směr Leader-Replika1
2	2	7204	999999000	999999002	132159	Směr Leader-Replika2

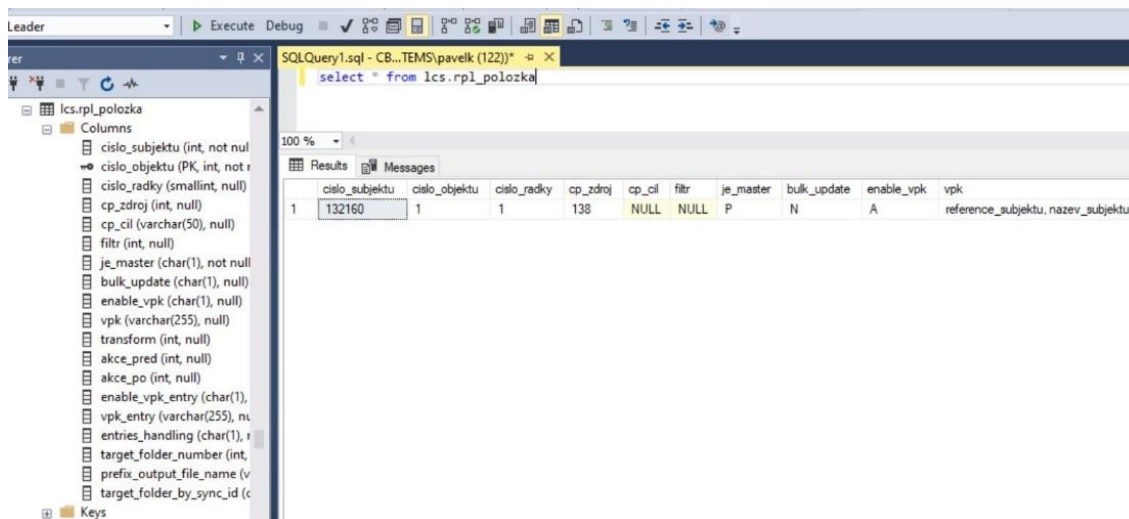
Obrázek 15: Tabulka *publication_target* se záznamy cílů publikace

12.3 Tabulky replikace

Tabulky samotných sestav replikace jsou *rpl_hlavicka* (11.3.2) a *rpl_položky* (11.3.3). Na obrázcích je vidět jediná vytvořená sestava replikace. Také si lze povšimnout množství atributů těchto tabulek.



Obrázek 16: Tabulka hlaviček sestavy replikace



Obrázek 17: Tabulka položek sestavy replikace

U obrázku tabulky položky sestavy replikace atribut *cp_zdroj* uvádí číslo replikovaného pořadače (138 - Země).

12.4 Speciální tabulky pro replikace

Jako speciální tabulky replikace jsou označovány ty tabulky, které nejsou dostupné z prezentační vrstvy systému Helios. Jsou to právě ty tabulky, které byly v předchozím kroku pročištěny. První z nich je tabulka *rpl_proxy*.

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' displays the structure of the *lcs.rpl_proxy* table, including columns *cislo* (int, not null), *cislo_tridy* (int, not null), and *gunid* (char(18), not null). On the right, the 'SQLQuery1.sql' window shows the query: `select * from lcs.rpl_proxy order by gunid`. The 'Results' pane displays the following data:

	cislo	cislo_tridy	gunid
1	20713	26	00000000000Q000FZD
2	20714	26	00000000000Q000FZE
3	20715	26	00000000000Q000FZF
4	20716	26	00000000000Q000FZG
5	20717	26	00000000000Q000FZH

Obrázek 18: Tabulka *rpl_proxy* se záznamy

V záznamech tabulky *rpl_proxy* jsou obsaženy tyto atributy:

- Číslo – zde je uvedeno číslo subjektu (popř. nonsubjektu) uvedeného záznamu
- Číslo třídy – je číslem třídy replikovaného záznamu
- GUNID – je již dříve zmíněný globální unikátní identifikátor

Další speciální tabulkou je tabulka stavů replikace *rpl_state*

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' displays the structure of the *lcs.rpl_state* table, including columns *gunid* (char(18), not null), *site_id* (int, not null), *smazano* (char(1), not null), *last_rpl* (timestamp, not null), *slouceno* (varchar(18), null), *state* (smallint, null), *last_rpl_date* (datetime, null), *last_rpl_attempt* (datetime, null), *rpl_attempt_count* (int, null), and *folder_number* (int, null). On the right, the 'SQLQuery1.sql' window shows the query: `select * from lcs.rpl_state order by gunid`. The 'Results' pane displays the following data:

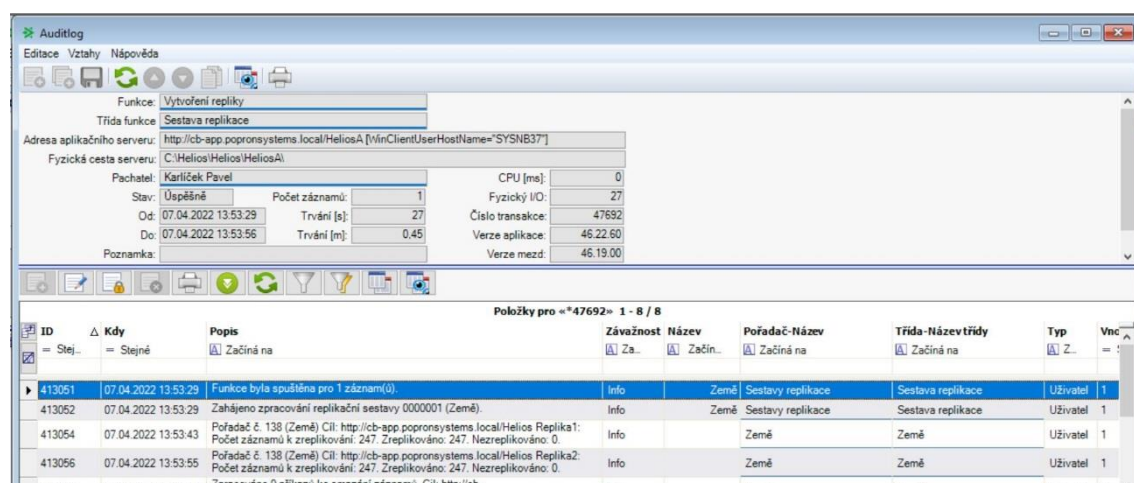
	gunid	site_id	smazano	last_rpl	slouceno	state
1	1U5HSP00001U000MBL	999999001	N	0x000000000006DFCF	NULL	1
2	1U5HSP00001U000MBL	999999002	N	0x000000000006E0C7	NULL	1
3	1U5HSP00001U000MC3	999999002	N	0x000000000006E0C8	NULL	1
4	1U5HSP00001U000MC3	999999001	N	0x000000000006DFD0	NULL	1
5	1U5HSP00001U000MC5	999999001	N	0x000000000006DFD1	NULL	1
6	1U5HSP00001U000MC5	999999002	N	0x000000000006E0C9	NULL	1
7	1U5HSP00001U000MCP	999999002	N	0x000000000006E0CA	NULL	1
8	1U5HSP00001U000MCP	999999001	N	0x000000000006DFD2	NULL	1
9	1U5HSP00001U000MD5	999999001	N	0x000000000006DFD3	NULL	1
10	1U5HSP00001U000MD5	999999002	N	0x000000000006E0CB	NULL	1
11	1U5HSP00001U000MDG	999999002	N	0x000000000006E0CC	NULL	1

Obrázek 19: Tabulka *rpl_state* se záznamy

V této tabulce jsou k jednotlivým záznamům v tabulce *rpl_proxy* aktualizovány stavy replikace. K tomuto slouží atribut *state*. Na Obrázek 19 je vidět, že s každým GUNIDem existují v tabulce dva záznamy. Rozdíl v těchto záznamech je v atributu *site_id*. Je to proto, že záznamy jsou replikovány do dvou různých cílových databází (replika 1, replika 2). Opět je na záznamech vidět atribut *folder_number*, který udává číslo pořadače replikovaného záznamu. Snímky všech zmíněných tabulek jsou pořízeny z vytvořeného testovacího prostředí, takže se vztahují k replikovaným záznamům.

13. Aplikační vrstva

Celý proces replikace je auditován v logovacím pořadači jménem *auditlog*. V něm jsou zaznamenány veškeré činnosti funkce „vytvoření repliky“ chronologicky v pořadí, jak k nim došlo.



Obrázek 20: Auditlog funkce vytvoření repliky

13.1 SOAP protokol

Protokol, kterým je zabezpečována replikace v Heliosu je SOAP (Sample Object Access Protocol). Je založen na výměně souborů ve formátu XML mezi webovými službami. Ta se děje v Heliosu Green za pomoci HTTP protokolu.

13.2 XML

Formát XML souboru pro replikace je možno zjistit nejjednodušeji nastavením způsobu replikace kopírování. Je-li pak provedena změna replikovaného záznamu a spuštěna

funkce vytvoření repliky. V adresáři určeném atributem sestavy replikace „Do adresáře“ jsou vytvořeny XML soubory a komprimovány do jediného souboru pojmenovaného podle aktuálního data a času. Každý replikovaný záznam má svůj XML soubor tomto formátu.

```
<?xml version="1.0" encoding="UTF-8"?>
<INSERTUPDATE SourceDatabaseNumber="999999000" action="insertUpdate"
folderActionOnUpdate="setFolder">
<RECORD
  SUBJECTID="60708"
  CLASSID="66"
  GUNID="1U5HSP00001U000MDQ"
  FOLDERID="138"
  REFERENCE="555"
  NAME="CC"
  CLASSNAME="Země"
  FOLDERNAME="Země"
  TIMESTAMP="000000000006E1A5"
  datum_vzniku="2011-04-15 13:34:36"
  last_update="2022-04-09
17:39:28" keyAttributes="reference_subjektu,nazev_subjektu" plny_nazev="Kokosové
ostrovy"
  celní_kod=""
  celní_nazev=""
  celní_preferovaná_zeme=""
  iso_kod_zeme="CC"
  exists_area_code="N"
  clenska_zeme_eu="N"
  iso_kod_zeme_numeric3="166"
  iso_kod_zeme_a3=""
  plny_nazev_en=""
  nazev_en=""
  sazba_srazkove_dane=""
  nazev=""
  platnost_do="" />
</INSERTUPDATE>
```

Každý záznam při replikaci přenáší všechny vybrané atributy a vztahy. Identifikovaný bude na cílové databázi podle klíčového atributu (není-li určen, pak podle GUNIDu).

Závěr

Cílem práce bylo dopodrobna kompletně nastavit a otestovat replikace v systému Helios Green. K tomu bylo vytvořeno modelové replikační prostředí se třemi databázemi. Bylo prověřeno, jak se toto nastavení promítá do databázových tabulek a aplikační vrstvy.

Replikace v Helios Green se chovají jako částečně synchronní systém, jelikož potvrzení zápisu jde okamžitě po zápisu do primární databáze, ale zdrojová databáze stále očekává od všech replik potvrzení a teprve poté co dorazí potvrzení z poslední repliky je považována synchronizace záznamu za kompletní. Systém tedy využívá strategii horlivého šíření, a to jak v případě online časování, tak i v případě offline časování replikace. Využití replikace pro Helios Green je typickým využitím replikace po databázové aplikaci a co se týká formy, jde o transakční replikaci.

Vývojáři z české firmy Asseco Solutions při vytváření vlastního nástroje pro replikace odvedli velmi dobrou práci. Funkcionalita replikací je přímo zakomponována do systému Helios Green a má propracované možnosti jak horizontální, tak i vertikální fragmentace. Doporučil bych pokračovat v rozvoji funkcionality replikace přímo v ERP systému.

Použité zdroje

- [1] Charron-Bost, B., Pedone, F., & Schiper, A. (2010). *Replication: Theory and practice*. Springer Science & Business Media.
- [2] Meine, S. (2013). *Fundamentals of SQL server 2012 replication*. Red Gate Books.
- [3] Özsu, M. T., & Valduriez, P. (2016). *Principles of distributed database systems*. Springer.
- [4] Liu, L., & Özsu, M. T. (2016). *Encyclopedia of database systems*. Springer.

- [5] Wiesmann, M., Pedone, F., Schiper, A., Kemme, B., & Alonso, G. (n.d.). Understanding replication in databases and distributed systems. *Proceeding's 20th IEEE International Conference on Distributed Computing Systems*.
<https://doi.org/10.1109/icdcs.2000.840959>
- [6] Abdul Moiz, S., P., S., G., V., & N. Pal, S. (2011). Database replication: A survey of open source and commercial tools. *International Journal of Computer Applications*, 13(6), 1-8. <https://doi.org/10.5120/1788-2469>
- [7] Kemme B. (2009) Data Replication. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_110
- [8] Jimenez-Peris R., Patiño-Martinez M. (2009) Replica Control. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA.
https://doi.org/10.1007/978-0-387-39940-9_310
- [9] Solanská, Markéta. *Synchronizace a replikace geodat v prostředí Esri platformy*. Olomouc, 2014. Diplomová práce. Katedra geoinformatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci.
- [10] Langer, Miroslav. Distribuované databázové systémy. *Databázové systémy II* [online]. 2020, 6-7 Dostupné z: <http://langer.zam.slu.cz/teaching/dbaii/>

Seznam zkratek

Zkratka	Její význam
BP	Bakalářská práce.
DBA	Databázový administrátor
DBMS	Database management system
DDBS	Distribuované databázové systémy.
DNS	Domain name service
DRP	Disaster recovery plan

DSŘBD	Distribuovaný systém řízení báze dat
DDBMS	Distributed database management system
GST	Global Stabilization Time
GUNID	Globální identifikátor záznamu
HA	High availability (vysoká dostupnost)
MDB	Master databáze
PLDB	Podřízená lokální databáze
SOAP	Simple Object Access Protocol
SŘBD	Systém řízení báze dat
ROWA	Read-one/write-all – protokol pro šíření aktualizací
XML	Extensible Markup Language

Seznam obrázků

Obrázek	Strana
Obrázek 1: Složitost replikace [9].....	9
Obrázek 2: Rozdíl mezi synchronní a asynchronní replikací [9].....	10
Obrázek 3: Srovnání multimaster a master-slave replikace [9].....	12
Obrázek 4: Schéma propojení databází	22
Obrázek 5: Uživatel a přihlašovací údaje pro replikaci.....	23
Obrázek 6: Parametry vlastníka: systémová nastavení	24
Obrázek 7: Místa replikace a Vzdálená přihlášení.....	24
Obrázek 8: Cíle publikace a vzorový záznam.....	25
Obrázek 9: Sestava replikace – záložka publikace	27
Obrázek 10: Sestava replikace – záložka obecné.....	28
Obrázek 11: Databázová tabulka "zeme"	31
Obrázek 12: Záznamy zemí v tabulce subjektů	31
Obrázek 13: Tabulka rpl_sites se záznamy míst replikace.....	33

Obrázek 14: Tabulka remote_login se záznamy vzdálených přihlášení	34
Obrázek 15: Tabulka publication_target se záznamy cílů publikace.....	34
Obrázek 16: Tabulka hlaviček sestavy replikace	35
Obrázek 17: Tabulka položek sestavy replikace.....	35
Obrázek 18: Tabulka rpl_proxy se záznamy	36
Obrázek 19: Tabulka rpl_state se záznamy	36
Obrázek 20: Auditlog funkce vytvoření repliky	37