

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Žilvinas Arlauskas

**DUOMENŲ APSIKEITIMO TARP DBVS IR
MOBILIŲJŲ ĮRENGINIŲ TYRIMAS**

Magistro darbas

Vadovas

Prof. R. Butleris

KAUNAS, 2009

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Žilvinas Arlauskas

**DUOMENŲ APSIKEITIMO TARP DBVS IR
MOBILIŲJŲ ĮRENGINIŲ TYRIMAS**

Magistro darbas

**Vadovas
Prof. R. Butleris**

**Konsultantas
Lekt. T. Danikauskas**

**Recenzentas
Dr.doc. A. Lenkevičius**

**Atliko
IFM 3/1 gr. stud.
Ž.Arlauskas**

KAUNAS, 2009

Duomenų apsikaitimo tarp DBVS ir mobilaus įrenginio tyrimas

Santrauka

Tradiciškai informacinėse sistemose (IS) vienas pagrindinių objektų yra duomenų bazės, kurios paprastai realizuotos reliacinių duomenų bazių valdymo sistemų (DBVS) pagrindu. Duomenų apdorojimui ar jų saugojimui mobiliuosiuose įrenginiuose taip pat tenka naudoti DBVS. Tai gali būti tiek mobilios, naudojamos lokaliai, tiek nutolusios DBVS. Norint pasiekti informacinės sistemos mobilios darbo vietos funkcionalumą ir našumą ne prastesnį nei stacionarių darbo vietų reikia pasirinkti tinkamą duomenų apsikaitimo metodą.

Šiame darbe tiriami duomenų apsikaitimo tarp DBVS ir mobilaus įrenginio metodai ir jų pasirinkimą lemiantys faktoriai. Darbe sprendžiami šie uždaviniai: išanalizuoti duomenų apsikaitimo tarp DBVS ir mobilaus įrenginio metodus ir suformuluoti kriterijus, lemiančius sprendimo priėmimą, pasirenkant duomenų apsikaitimo metodą; sudaryti duomenų apsikaitimo tarp DBVS ir mobilaus įrenginio pasirinkimo procesą.

Pagal atliktą literatūros analizę darbe suformuluoti 5 kriterijai, turintys įtakos pasirenkant duomenų apsikaitimo metodus tarp nutolusios DBVS ir mobiliojo įrenginio. Pagal kriterijus sudarytas duomenų apsikaitimo metodo pasirinkimo procesas. IS kūrėjai pasinaudodami pasiūlytu duomenų apsikaitimo tarp DBVS ir mobilaus įrenginio metodo pasirinkimo procesu gali parinkti tinkamą sprendimą kiekvienos kuriamos IS atveju. Procesas pateikiamas grafinėje notacijoje su ją lydinčių detaliu aprašu.

Papildomai darbe atliktas nutolusių duomenų prieigos ir tiesioginis nutolusios duomenų bazės naudojimo duomenų apsikaitimo metodų eksperimentinis našumo tyrimas. Eksperimentui atlikti buvo suprojektuota ir sukurta programinė įranga. Taikomoji programa leidžia atlikti operacijas su duomenimis naudojant nutolusios duomenų prieigos paslaugą ir tiesioginę nutolusios duomenų bazės prieigą. Tyrimui buvo naudojamas duomenų bazės epizodas iš 5 lentelių, o duomenų operacijos apdorojavo nuo 100 iki 100 000 įrašų vienos transakcijos metu. Atlikas eksperimentas padėjo nustatyti dar du kriterijus, kurie lemia duomenų apsikaitimo tarp DBVS ir mobilaus įrenginio metodo pasirinkimą. Šiais kriterijais buvo papildytas pasiūlytas duomenų apsikaitimo procesas. Pasiūlytas procesas esant reikalui gali būti plečiamas IS kūrėjų.

Raktiniai žodžiai: DBVS, nutolusi duomenų prieigos paslauga, suliejimo replikavimas, nutolusi duomenų bazė, procesas.

Data transfer between DBMS and mobile device research Summary

Database is one the main object in the informatics systems, which is usually realized relational database management systems (DBMS) basis. DBMS are necessarily used for data manipulation and storing in mobile devices. This can be mobile, used locally as well as remote DBMS. In order to achieve similar functionality and performance of desktop systems for informational mobile system we have to choose a suitable data synchronization method.

This work investigates the exchange of data between the DBMS and the mobile device and methods of determining factors for their choice. In this task are pending tasks: analyze the data exchange between the DBMS and the mobile device and methods of formulating the criteria that determine the decision-making, the choice of data-exchange method; allow the exchange of data between the DBMS and the mobile device choice process.

According to the literary analysis 5 criteria were formulated, which affect the choice of methods for data exchange between the remote DBMS and the mobile device. According to the criteria were made data exchange methods for the choice process. IS developers using the process for choosing a method for data exchange between DBMS and mobile device are able to select the appropriate solution for each IS that is being developed. The process provides the graphical notation with the accompanying detailed description.

Additionally, the work carried out in remote data access and direct remote database using data exchange methods for the experimental performance research. For experiments were designed and developed software. Application allows doing operations on data using remote data access service and direct access to remote databases. The research was conducted using a set of 5 database tables and data operations which processed from 100 to 100 000 entries in one transaction. Experiment helped to establish two further criteria, which leads to the exchange of data between the DBMS and the mobile device selection method. These criteria were added to the proposed data exchange process. The proposed process, if necessary, can be extended IS developers.

Key words: mobile database, replication, remote data access, DBMS, direct access to remote databases.

Turinys

1. ĮVADAS.....	6
2. DUOMENŲ APSIKEITIMO TARP DBVS IR MOBILAUS ĮRENGINIO METODŲ IR PRIEMONIŲ ANALIZĖ	8
2.1 Nutolusių duomenų prieigos paslaugos analizė	8
2.2 Duomenų bazių replikavimo analizė	13
2.3 Tiesioginės nutolusios duomenų bazės prieigos naudojimo analizė.....	24
2.4 Mobilias duomenų bazes realizuojančių įrankių analizė.....	25
2.5 Duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodų ir priemonių analizės išvados.....	33
3. DUOMENŲ APSIKEITIMO TARP DBVS IR MOBILAUS ĮRENGINIO METODO PASIRINKIMO PROCESAS	34
4. DUOMENŲ APSIKEITIMO TARP DBVS IR MOBILAUS ĮRENGINIO METODŲ EKSPERIMENTINIS TYRIMAS	38
4.1 Eksperimentiniam tyrimui sukurtos programinės įrangos ir naudojamų duomenų specifikacija	38
4.2 Duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodų eksperimentinio tyrimo rezultatai	48
5. IŠVADOS	55
6. LITERATŪROS SĄRAŠAS	56

1. Įvadas

Šiandienos informacinių technologijų rinka sparčiai plečiasi į mobiliuosius įrenginius, kurie pasižymi kompaktiškumu, greitumu, lengvu naudojimu, lankstumu, integruotumu. Dėl išvardintų mobiliųjų įrenginių savybių informacinių sistemų srityje daugėja ir sprendimų, kurie skirti mobiliams įrenginiams. Tradiciškai informacinėse sistemose (IS) vienas pagrindinių objektų yra duomenų bazės, kurios paprastai realizuotos reliacinių duomenų bazių valdymo sistemų (DBVS) pagrindu. Duomenų apdorojimui ar jų saugojimui mobiliuosiuose įrenginiuose taip pat tenka naudoti DBVS. Tai gali būti tiek mobilios, naudojamos lokaliai, tiek nutolusios DBVS. Norint pasiekti informacinės sistemos mobilios darbo vietos funkcionalumą ir našumą ne prastesnį nei stacionarių darbo vietų tenka susidurti su problemomis, kurios būdingos kuriant IS mobilias darbo vietas.

Pirma, mobilus įrenginys turi palaikyti nuolatinį interneto ryšį, jei reikia, nuolat keistis duomenimis su nutolusia (centrine) DBVS. Nesant nuolatinio ir stabilaus interneto ryšio, vartotojas negalės tiesiogiai komunikuoti su nutolusia duomenų baze. Todėl nebus galima užtikrinti nuoseklaus darbo su informacine sistema. Taip pat nukenčia ir duomenų apdorojimo našumas. Duomenų apskaitos metu nutrūkus interneto ryšiui, reikės pakartoti duomenų apskaitą, taip bus sugaištamas papildomas laikas.

Antra, duomenų apskaitos su nutolusiomis duomenų bazėmis gali būti nepatogus, jei dirbama su dideliais duomenų kiekiais, nes tam bus reikalingi dideli laiko kaštai. Problemos gali kilti, jei vartotojui reikia atlikti ir didesnius kiekius duomenų apdorojimo operacijų vienu metu.

Trečia, nors mobilieji įrenginiai yra smarkiai patobulėję, bet vis dar negali saugoti ir apdoroti neribotų duomenų kiekių, tai taip pat turi būti įvertinta kuriant sistemą.

Šio darbo sritis yra mobilaus įrenginio ir nutolusios duomenų bazės duomenų apskaitos metodai. Tyrimo objektas – duomenų apskaitos metodai ir procesą realizuojantys technologiniai sprendimai, programinė įranga.

Šio darbo tikslas – iširti duomenų apskaitos tarp DBVS ir mobilaus įrenginio metodus ir pateikti procesą, kuris padėtų informacinės sistemos kūrėjui (projektuotojui, analitikui, programuotojui) pasirinkti teisingausia duomenų apskaitos metodą kuriant mobilias darbo vietas.

Uždaviniai:

- išanalizuoti duomenų apskaitos tarp DBVS ir mobilaus įrenginio metodus ir suformuluoti kriterijus, lemiančius sprendimo priėmimą, pasirenkant duomenų apskaitos metodą;
- sudaryti duomenų apskaitos tarp DBVS ir mobilaus įrenginio pasirinkimo procesą, su kuriuo kūrėjui parankiausia atlikti duomenų apskaitą;

- atlikti pateiktos koncepcijos eksperimentinį tyrimą siekiant nustatyti sudaryto proceso pilnumą.

Siekiant įgyvendinti tikslą buvo suprojektuota ir sukurta taikomoji programa, kuri padėtų nustatyti, koks duomenų apskaitos metodas patogiausias kūrėjui atsižvelgiant į konkrečius kriterijus. Taikomoji programa geba atlikti nutolusių duomenų prieigos paslaugą ir tiesioginį nutolusių duomenų bazės naudojimą. Naudojant sukurta sistemą, duomenys buvo apdorojami mobiliajame įrenginyje ir tiesiogiai nutolusioje duomenų bazėje. Tyrimui buvo naudojamos 5 lentelės ir nuo 100 iki 100 000 įrašų apdorojančios operacijos. Tuo siekiama nustatyti operacijos atlikimo laiko kaštų priklausomybę nuo apdorojamų įrašų kiekio. Tiriama tiesioginės duomenų bazės naudojimą, buvo tiriamos pagrindinės operacijos su duomenimis ir galiausiai palyginamos su nutolusių duomenų prieigos paslauga.

Darbo struktūra:

Pirmame skyriuje atlikta duomenų apskaitos tarp DBVS ir mobilios įrenginio analizė. Pateikiama duomenų apskaitos su mobiliomis duomenų bazėmis pagrindinės sąvokos, su kuriomis dažniausiai kūrėjai susiduria, apskaitos metu naudojami modeliai ir jų komponentai. Apibūdinamas mobilios DBVS naudojimas, architektūra, palyginami jos perdavimo metodai. Pateikiama nutolusių duomenų bazės naudojimo metodai ir palyginimas. Atlikus pirmo skyriaus medžiagos analizę, pateikiama trumpa išvada.

Antrame skyriuje suformuluoti kriterijai, turintys įtakos pasirenkant duomenų apskaitos metodus tarp nutolusių DBVS ir mobilios įrenginio. Pagal kriterijus pateiktas duomenų apskaitos metodo pasirinkimo procesas. IS kūrėjai pasinaudodami pasiūlytu duomenų apskaitos tarp DBVS ir mobilios įrenginio metodo pasirinkimo procesu galės parinkti tinkamą sprendimą kiekvienos kuriamos IS atveju. Procesas pateikiamas grafinėje notacijoje su ją lydinčiu detaliu aprašu.

Trečiame skyriuje pateikiama eksperimento aprašymas: kaip jis buvo daromas ir kokios naudojamos priemonės, reikalingos sukurti taikomąją programą. Pateikiami atlikto tyrimo rezultatai ir išvada.

Ketvirtajame skyriuje pateikiamos baigiamosios darbo išvados.

2. Duomenų apskaitimo tarp DBVS ir mobilios įrenginio metodų ir priemonių analizė

Egzistuoja trys metodai, skirti duomenims perduoti tarp mobilios ir nutolusios duomenų bazės ir tiesiogiai duomenis apdoroti nutolusioje duomenų bazėje. Pirmas metodas yra nutolusių duomenų prieigos paslauga. Paslauga suteikia galimybę duomenis apdoroti mobilioje duomenų bazėje. Naudojantis pagrindiniais paslaugos metodais, duomenys yra persiunčiami į mobilią duomenų bazę, o po to grąžinami į nutolusią duomenų bazę. Mobilioje duomenų bazėje duomenys yra apdorojami SQL sakiniiais ir procedūromis. Antras metodas yra pagrįstas SQL Server suliejimo replikavimu. Tai yra duomenų bazės technologija, kuri sujungia pakeitimus, padarytus tarp dviejų ar daugiau mazgų. Šis vienintelis replikavimo metodas skirtas mobiliesiems įrenginiams. Ši technologija automatiškai atnaujina pakeitimus į nutolusią duomenų bazę, atsiradus arba esant sujungimo ryšiui su ja. Trečias metodas, kai taikomoji programa naudoja SQL sakinius, kurie vykdomi tiesiogiai į nutolusią duomenų bazę.

2.1 Nutolusių duomenų prieigos paslaugos analizė

Nutolusių duomenų prieigos paslauga, kuri naudoja mobilią duomenų bazę, leidžia taikomajai programai prieiti prie nutolusios duomenų bazės duomenų ir saugoti juos mobilios duomenų bazės lentelėje. Kai duomenys yra įtraukti mobilioje duomenų bazėje, tada taikomoji programa gali skaityti ir atnaujinti mobilias lenteles. Taikomoji programa vėliau siunčia atnaujintus įrašus iš mobilios duomenų bazės lentelės į nutolusios duomenų bazės lentelę.

Nutolusių duomenų prieigos paslaugos paskirtis (tipinis naudojimas)

Perkelti duomenys. Taikomoji programa gali traukti duomenis iš nutolusios į mobilią duomenų bazę.

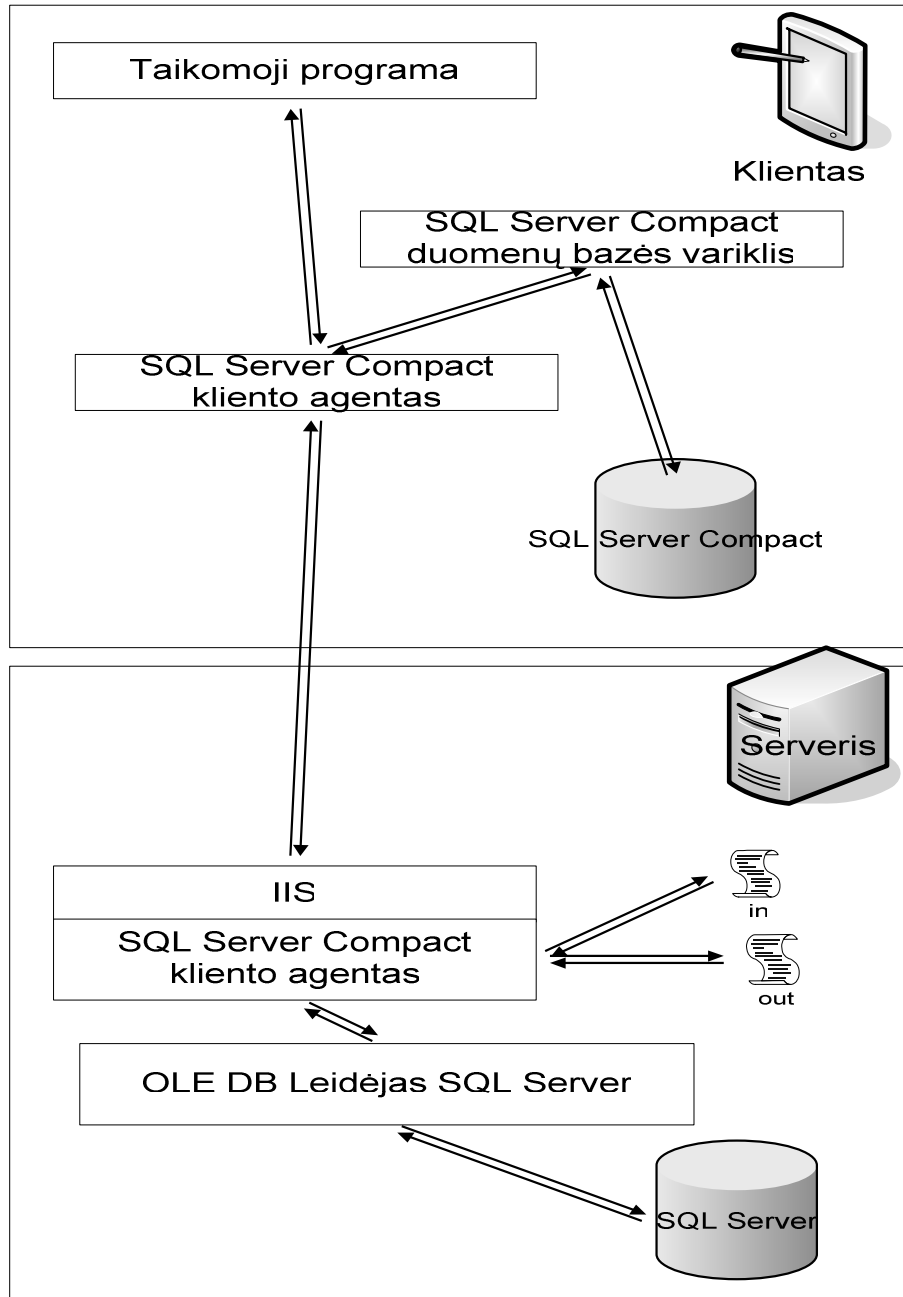
Užfiksuojami ir išsiunčiami duomenys. Taikomoji programa gali užfiksuoti duomenis iš vartotojo įvedamos formos. Programa gali saugoti užfiksuotus duomenis mobilioje duomenų bazėje, kuri yra įrenginyje. Po to periodiškai taikomoji programa gali stumti užfiksuotus duomenis iš mobilios duomenų bazės į nutolusią duomenų bazę.

Perkeliami, atnaujinami ir išsiunčiami duomenys. Taikomoji programa gali traukti duomenis iš nutolusios duomenų bazės į mobilią duomenų bazę. Kai duomenys yra mobilioje duomenų bazėje, tada ji gali atnaujinti mobilią duomenų bazę ir vėl periodiškai siųsti pakeitimus į nutolusią duomenų bazę.

Pateikiami (Patvirtinami) SQL sakiniai. Taikomoji programa gali pateikti SQL sakinius vykdymui nutolusioje duomenų bazėje. Tai yra naudinga, kai įrenginys nuolat palaiko interneto ryšį.

Nutolusių duomenų prieigos paslaugos architektūra

Ši paslauga naudoja tris mobilios DBVS komponentus: mobilios duomenų bazės variklį, mobilios DBVS kliento agentą ir mobilios DBVS serverio agentą, kurie pavaizduoti 1 paveiksle. Paslauga naudoja Interneto Informacinę Paslaugą (IIS) kaip komunikacijos mechanizmą tarp nutolusios duomenų bazės ir mobilios duomenų bazės. Visi šie trys komponentai veikia kartu, kad maksimaliai išnaudotų nutolusios duomenų prieigos paslaugą.[12]



1 pav. Nutolusių duomenų prieigos paslaugos klientas/serveris architektūra

Mobilios DBVS variklis

Jis valdo nutolusios duomenų bazės duomenų saugojimą mobiliajame įrenginyje. Duomenų bazės variklis pastebi visus įrašus, kurie yra įterpti, atnaujinti ar ištrinti. Jei egzistuoja indeksai SQL serverio duomenyse, nutolusių duomenų prieigos paslauga taip pat palaiko sukurtus indeksus mobiliajame įrenginyje.

Mobilios DBVS kliento agentas

Jis nustatomas įrenginyje, kuriame naudojama paslauga. Kliento agentas vykdo nutolusių duomenų prieigos paslaugos objekto sąsają ir yra pagrindinis mobilios duomenų bazės komponentas. Taikomosios programos kviečia šią sąsają, kuri yra parašyta programiniu kodu.

1 lentelė. Pagrindinių nutolusių duomenų prieigos paslaugos metodų atsakymas į mobilios DBVS kliento agentą

Metodas	Veiksmas
Pull	Persiunčiami reikalavimai į mobilios DBVS serverio agentą per informacijos perdavimo internete protokolą (HTTP). Kai mobilios DBVS kliento agentas gauna įrašų aibę iš nutolusios DBVS, jie saugojami mobilioje DBVS.
Push	Ištraukiami visi įterpimai, atnaujinimai, ištrinami iš mobilios DBVS ir siunčiami į mobilios DBVS serverio agentą per HTTP.
SubmitSQL	Persiunčiami specialiai apibrėžti SQL reikalavimai į mobilios DBVS serverio agentą per HTTP.

Mobilios DBVS serverio agentas

Jis nustatomas kompiuteryje, kuriame yra paleistas IIS, reguliuojami (prižiūrimi) HTTP reikalavimai, kurie daromi mobilios DBVS kliento agento. Tam yra naudojami laikini žinučių failai (*.in ir *.out), kurie valdo duomenų pasikeitimus tarp nutolusios duomenų bazės ir mobilios DBVS.

2 lentelė Pagrindinių nutolusių duomenų prieigos paslaugos metodų atsakymas į mobilios DBVS serverio agentą

Metodas	Veiksmas
Pull	Gauna reikalavimus iš mobilios DBVS kliento agento, susijungia su nutolusia duomenų baze per OLE DB tiekėją, skirtą nutolusiai duomenų bazei, ir kreipiasi į klientų SQL sakinius. Mobilios DBVS serverio agentas gražina apskaičiuotas įrašo aibes į mobilios DBVS kliento agentą per HTTP.
Push	Gauna visus įterptus, atnaujintus ir ištrintus įrašus iš mobilios DBVS kliento agento, susijungia su nutolusia duomenų baze per OLE DB ir įterpia, atnaujina ar ištrina įrašus nutolusioje duomenų bazėje. Jei įvyksta klaida, mobilios DBVS serverio agentas praneša klaidas į mobilios DBVS kliento agentą per HTTP.
SubmitSQL	Gauna specialiai apibrėžtus SQL reikalavimus iš mobilios DBVS kliento agento per HTTP, susijungia su nutolusia duomenų baze per OLE DB ir kreipiasi į kliento SQL sakinius. Jei įvyksta klaida, mobilios DBVS serverio agentas praneša klaidas į mobilios DBVS kliento agentą per HTTP.

Nutolusių duomenų prieigos paslaugos veikimas

Jis pasirūpina paprastu būdu taikomajai programai prieiti prie duomenų, išdėstytų nutolusioje duomenų bazėje. Duomenų platinimas iš pradžių yra inicijuojamas kliento. Duomenys yra traukiami iš serverio lentelės į kliento pusės lentelę. Tada klientas atlieka pakeitimus mobiliojo įrenginio duomenų bazės lentelėje. Galiausiai klientas stumia duomenis atgal į serverį.

3 lentelė. Pagrindiniai nutolusių duomenų prieigos paslaugos metodai

Terminas	Apibrėžimas
Pull	Ištraukiami duomenys iš nutolusios duomenų bazės ir perkeltami į mobilią DBVS.
Push	Siunčiami pakeitimai iš mobilios DBVS lentelės į egzistuojančią nutolusią duomenų bazę.

Nutolusių duomenų prieigos paslauga nereikalauja serverio konfigūracijos. Tai yra geras sprendimas, kai verslo reikalavimai sukliudo pokyčiams į duomenų schemą, esančią serveryje. Klientai privalo modifikuoti šią paslaugą per taikomąją programą, kuri veikia įrenginiuose. Šitas suliejimo sprendimas yra geriausias, kad neaptarnautų kelių klientų ar tais atvejais, kai reikalingas nedidelis kodo pakeitimas, kad būtų galima prisijungti prie duomenų per taikomąją programą.

Paslauga nepalaiko konfliktų sprendimo, kad reguliuotų eilutes, kai jos nėra pritaikomos serveryje, jei yra klaidų. Taikomoji programa, esanti įrenginyje, privalo turėti kodą, kad valdytų (reguliuotų) klaidas. Paslauga neaptinka tipinių konfliktų, tokių kaip dviejų vartotojų duomenų pakeitimas. Yra pritaikomas paskutinis atnaujinimas. Paslaugos naudojimas mobilios DBVS turi būti apribojamas tuose laukuose:

Karkaso tikslumas. Mobilios DBVS palaiko nustatymus karkaso tikslumui palyginant duomenų bazės lygį. Objekto pavadinimai, tokie kaip lentelės pavadinimas, kalba, funkcijos ir peržiūros, yra netraktuojami kaip karkaso tikslumas mobilios DBVS duomenų bazėje.

Trigeriai. Mobilios DBVS nepalaiko trigerių.

Lentelės ir stulpeliai. Stebimi lentelių stulpelių numeriai yra prieinami iki 1017, nes 7 sistemos stulpeliai yra apsaugoti ir naudojami stebėti nutolusių duomenų prieigos paslaugos lentelėms.

Apdoroti stulpeliai. Pull metodas yra nepasiekiamas, jei įrašų aibė turi apdorojamus stulpelius.

IDENTITY stulpeliai. Paslauga dinamiškai nepalaiko IDENTITY stulpelių. Reikalingas nustatymas rankiniu būdu, kad mobilios DBVS lentelėje būtų nustatomas IDENTITY.

Schemas apribojimas

Mobilios DBVS gali stebėti pokyčius savo lentelėse. Mobilios duomenų bazė stebi visus įterpimus, atnaujinimus ir ištrynimus, kurie padaromi mobilioje lentelėje. Taikomoji programa gali iškviešti Push metodą, kad dalintų tuos pakeitimus atgal į originalią nutolusią lentelę. Pull ir Push

metodai naudoja optimistinį vykdymo sutapimą. Nutolusi duomenų bazė nepalaiko traukiamų įrašų blokavimo. Kai taikomoji programa iškviečia Push, mobilioje DBVS daromi pakeitimai, kurie yra besąlygiškai pritaikomi nutolusioje duomenų bazėje. Tai gali būti pagrindas prarasti duomenis, padarytus įvestus kito vartotojo, naudojant Pull ir Push metodus, kai įrašai yra atnaujinami, logiškai suskirstomi ir konfliktai yra skirtingi.

Nutolusių duomenų prieigos paslaugos konfliktų nustatymas ir pranešimas

Nors ir nutolusių duomenų prieigos paslauga nepalaiko tikslaus apsisprendimo būdo, kad nereplikuotų, mobilios DBVS pasirūpina klaidų lentele, į kurią įrašomi visi konfliktai. Klaidų lentelę galima apibrėžti kaip Pull metodo dalį. Bet kokios klaidos, kurios įvyksta stūmimo procese, įrašomos į klaidų lentelę. Naudojant klaidų lentelę galima valdyti konfliktų nustatymus ir pranešimus.

Pokyčiai yra daromi mobilioje DBVS, kur stūmimas pritaikomas serveryje, iš kurio yra gaunamos visos klaidos. Nutolusios duomenų bazės lentelė yra atnaujinama, kad atspindėtų pokyčius, kuriuos padarė paskutinis vartotojas.

Konfliktas egzistuoja, kai eilutė negali būti išstumta iš mobilios DBVS į nutolusių duomenų bazę. Nutolusių duomenų prieigos paslauga palaiko tik eilutės lygio stebėjimą.

Kai įvyksta nutolusių duomenų prieigos paslaugos konfliktai

Nutolusių duomenų prieigos paslauga stebi įterpimo, atnaujinimo ir ištrynimo operacijas, ypač kiekvienoje eilutėje. Jei duomenys yra netiksliai ir neteisingai padalinti kiekvienam vartotojui, vienas vartotojas gali įrašą ištrinti, kitas gali tuo metu jį atnaujinti.

Eilutės siuntimas į serverį nutrūksta, jei ankstesniu veiksmu buvo įvykdytas nutraukimas.

Kelių vartotojų prieiga

Traukimo operacija reikalauja, kad mobili lentelė nedalyvautų traukimo metu, o ji tiesiog yra sukuriama proceso metu, kai duomenys gaunami iš serverio. Kai vyksta stūmimas, jei duomenys pakeičiami stūmimo metu, tai visi pokyčiai bus nusiųsti į serverį kitu stūmimo metu. Vartotojas, vykdamas duomenų atnaujinimą serveryje, turi pirma užbaigti traukimo arba stūmimo operacijas, kitu atveju bus prarasti duomenys. Leidžiant keliems sujungimams į prieigą, mobilios DBVS siūlo kūrėjams paprasta plėtojimo modelį.[12]

2.2 Duomenų bazių replikavimo analizė

Paskirstyta sistema

Paskirstyta sistema yra nepriklausomų kompiuterių (įrenginių) rinkinys, kaip viena sujungta sistema. Šitas apibrėžimas turi keletą aspektų. Pirma, paskirstyti sistemos komponentai, kuriuos turi pati sistema, yra autonominiai. Kitas aspektas yra, kad naudojami kompiuteriai priskiriami vienai sistemai. Tai reiškia, kad vienu ar kitu būdu autonominiai komponentai turi bendradarbiauti.[1]

Paskirstyta duomenų bazė

Paskirstyta duomenų bazė yra iš dalies nepriklausomos duomenų bazės, kad padalintų bendrą schemą, koordinuotų transakcijų vykdymą ir naudotų nutolusius duomenis.[1]

Transakcija

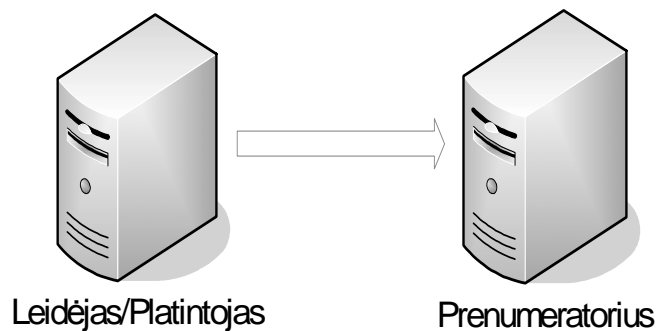
Transakcija yra vykdomos programos vienetas, kuris gauna informaciją ir atnaujina įvairius duomenų elementus. Paprastai transakcija yra pridedama vartotojo programos parašyta aukšto lygio duomenų manipuliavimo kalba arba programavimo kalba, kuri yra apibrėžiama nuo begin transaction ir end transaction sakiniais. Transakcijos visi operatoriai vykdomi tarp tų sakinių.[2][3]

Replikavimas

Replikavimas yra „technologijų aibė“, kuri gali perduoti duomenis ir duomenų bazės objektus iš vienos duomenų bazės į kitą, tam pritaikydamas skirtingas platformas ir geografinį išdėstymą. Tai leidžia vartotojui dirbti su duomenų bazės kopija ir bet kokius pakeitimus siųsti į vieną ar kelis nutolusius serverius arba mobiliems vartotojams perduoti duomenis per tinklą. Replikuojamos duomenų bazės privalumai yra fizinis duomenų bazės atskyrimas ir greitas duomenų apsikeitimo laikas.[6]

Leidėjas/Platintojas - Prenumeratorius modelis

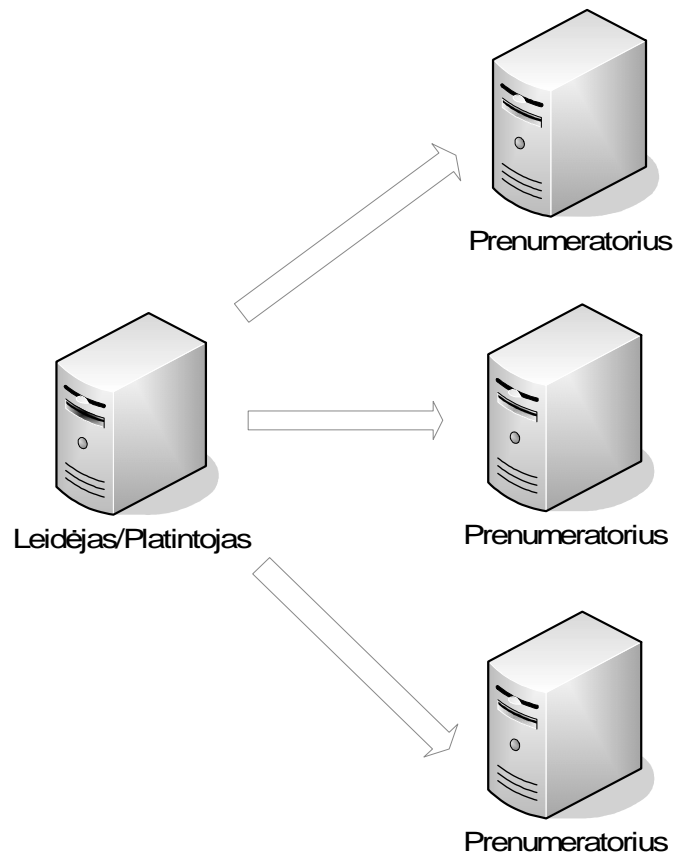
Platintojo serveris kartu su Leidėjo serveriu sudaro viena fizinį serverį, o kitas atskiras Prenumeratoriaus fizinis serveris. Tai paprasčiausias iš visų modelių, kuris pavaizduotas 2 paveiksle. Duomenys yra replikuojami iš Leidėjo serverio į Prenumeratoriaus serverį.



2 pav. Leidėjas/Platintojas - Prenumeratorius modelis

Centrinis Leidėjas – Keli Prenumeratoriai modelis

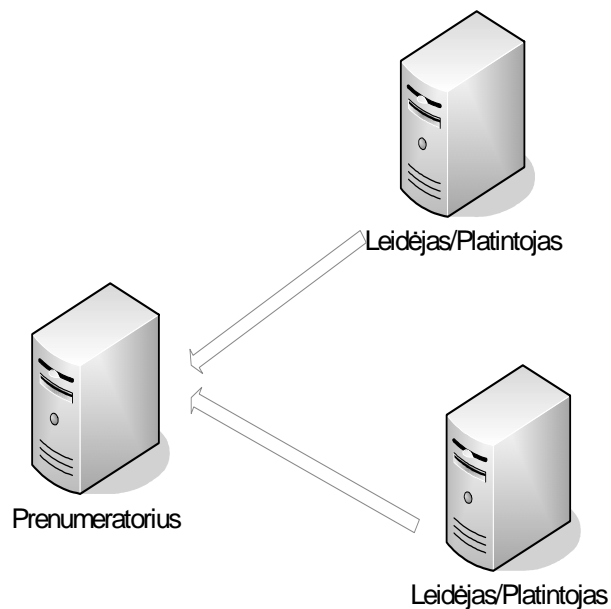
Centrinis Leidėjas – Keli Prenumeratoriai modelis yra iš tikrųjų ankstesnio modelio praplėtimas, kuris pavaizduotas 3 paveiksle. Turimas vienas Leidėjo serveris, kuris paviešina duomenis. Jie yra siunčiami keliems Prenumeratorių serveriams. Leidėjo serveris yra centrinis platintojas, kuris paskirsto duomenis keliems Prenumeratorių serveriams. Platintojo serveris gali būti naudojamas ant kai kurio fizinio serverio, kaip Leidėjo serveris, arba ant skirtingo, atskiro. Tipiškai šitas modelis naudojamas, kai norima, kad prenumeratoriai turėtų tik skaitymo teises. Platintojo serveris vykdo pakeitimus duomenyse iš Leidėjo serverio ir juos siunčia į Prenumeratorių serverius. Kai norima Prenumeratoriaus serverį padėti į kita fizinį serverį, tai reikia visus replikavimo darbus perkrauti į kitą. Tai gali duoti aukšto greičio tinklo sujungimo privalumą ar optimizuoti replikavimo našumą. Leidėjo serveris perskirsto ir išsiuntinėja duomenis juos užsisakiusiems prenumeratoriams. Paskirstymo serveris gauna visus duomenys iš Leidėjo serverio.



3 pav. Centrinis Leidėjas – Keli Prenumeratoriai modelis

Centrinis prenumeratorius – kelių Leidėjų modelis

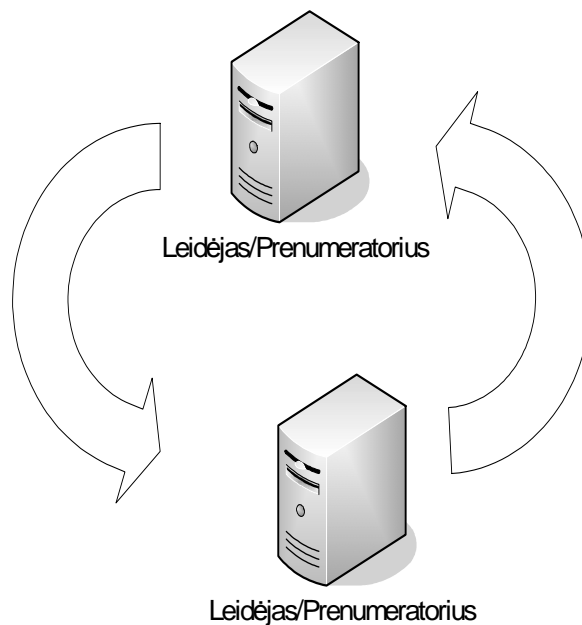
Centrinis prenumeratorius – kelių Leidėjų modelis yra bendrai naudojamas situacijose, kuriose duomenis iš kelių vietų reikia sutraukti į centrinį DBVS. Modelis pavaizduotas 4 paveiksle. Keli Leidėjo serveriai daro tikslią duomenų kopiją į konkrečias prenumeratas.



4 pav. Centrinis Prenumerorius – Keli Leidėjai modelis

Keli Leidėjai – kelių Prenumeratorių modelis

Keli Leidėjai – kelių Prenumeratorių modelis leidžia Leidėjo serveriui veikti kaip Prenumeratoriui. Kai naudojamas toks modelis, kuris pavaizduotas 5 paveiksle, reikalinga įvertinti duomenų apdorojimo nuoseklumą ir išskylančius konfliktus duomenų atnaujinimo metu. Šis modelis suteikia galimybę įgyvendinti kelių Leidėjų – Kelių Prenumeratorių sprendimą, naudojant transakcijų ir suliejimo replikavimus. Bet kuriuo laiku duomenys lentelėje yra atnaujinami, kai daroma kopija atliekant transakcinį replikavimą.[6]



5 pav. Keli Leidėjai – keli Prenumeratoriai modelis

Agentas

SQL Server Agentas yra Microsoft Windows paslauga, kuri vykdo suplanuotas valdymo užduotis. SQL Server Agentas naudoja SQL Server, kad saugotų darbo informaciją. SQL Server Agentas gali darbą vykdyti pagal grafiką, reaguodamas į specialius įvykius arba pagal reikalavimus. [6]

Momentinio vaizdo agentas

Šitas agentas dažniausiai būna Platintojo serveryje. Momentinio vaizdo agentas yra naudojamas visose replikacijose, ypač pradinio sinchronizavimo metu. Jis sukuria schemas kopiją ir lentelių duomenis, kurie yra paviešinti, tada saugoma momentinio vaizdo faile ir įrašų informacija apie sinchronizaciją paskirstymo duomenų bazėje.

Paskirstymo agentas

Ir momentinis vaizdas, ir transakcinis replikavimas naudoja šį agentą. Paskirstymo agentas yra atsakingas už judėjimą momentinio vaizdo ir transakcijos, kurie laikomi paskirstymo duomenų bazėje, patalpintoje prenumeruojamoje duomenų bazėje. Kai atliekamas prenumeratų stūmimas iš platinamos duomenų bazės, tai paskirstymo agentas būna Prenumeratoriaus serveryje.

Suliejimo agentas

Šis agentas yra naudojamas su suliejimo replikavimu. Suliejimo agentas taikosi į pradinį momentinį vaizdą, kuris yra Prenumeratorių serveriuose. Padidėjimas keičia sub seką į pradinę sinchronizaciją, kuri kontroliuoja ir sujungia į Prenumeratoriaus serverį, naudojant suliejimo agentą. Kiekviena duomenų bazės paimta duomenų dalis procese turi viena suliejimo agentą. Panašiai veikia paskirstymo agentas. Suliejimo agentas saugojamas Platintojo serveryje, kai reikia stumti prenumeratas, ir Prenumeratoriaus serveryje, kai traukiamos prenumeratas.

Protokolo(registro) skaitytojo agentas

Šis agentas yra naudojamas transakciniam replikavimui. Protokolo(registro) skaitytojo agentas stebi visų duomenų bazių transakcijų protokolus(registrus), kurie yra įtraukiami transakcijų replikavime. Agento kopijų bet kokie duomenų pokyčiai, kurie yra pažymėti replikavimo, publikuojami duomenų bazės transakciniame protokole(registre) ir tada jie siunčiami į Platintojo serverį, kur jie yra saugojami paskirstytoje duomenų bazėje. Transakcijos laikomos ligi tol, kol jos yra siunčiamos į Prenumeratoriaus serverį.[6]

Platintojas

Platintojo serveris yra replikavimo procesą valdantis objektas, jis užtikrina visų replikavime dalyvaujančių serverių darnų komunikavimą ir priskirtų rolių vykdymą. Jis turi platinamą duomenų bazę, kuri yra tarpininkė grandis tarp Leidėjo serverių ir Prenumeratoriaus serverių. Jei Platintojo serveris yra išdėstomas toje pačioje mašinoje kaip Leidėjo serveris, tai yra žinomas kaip vietinis Platintojo serveris, bet jei tai yra skirtingoje mašinoje negu Leidėjo serveris, tai yra vadinamas

nutolusiu Platintojo serveriu. Kai atliekamas didelis duomenų replikavimas, tai geriau duomenis talpinti Platintojo serveryje.

Platintojo serverio rolė keičiasi priklausomai nuo replikavimo tipo:

Naudojant momentinio vaizdo ir transakcinį replikavimą, platinama duomenų bazė, esanti Platintojo serveryje, saugo laikinai replikuojamus meta duomenis ir darbo eigos istoriją. Replikavimo agentai yra irgi saugojami Platintojo serveryje, išskyrus atvejus, kai agentai yra konfigūruojami nutolusiu būdu arba prenumeratoriai traukia duomenis.

Suliejimo replikavimas nepanašus į momentinio vaizdo ir transakcinį replikavimą, paskirstytos duomenų bazės, esančios Platintojo serveryje, saugoja meta duomenis ir duomenų apsikeitimo istoriją. Jis taip pat turi savyje momentinio vaizdo agentą ir suliejimo agentą, skirtą prenumeratoriams, kurie stumia duomenis.

Leidėjas

Kol Platintojo serveris valdo duomenų srautus, Leidėjo serveris užtikrina, kad duomenys yra prieinami replikavimui į kitus serverius. Leidėjas yra serveris, kuris turi duomenis, skirtus replikavimui. Jis taip pat gali identifikuoti ir patvirtinti duomenų pakeitimus.

Prenumeratorius

Prenumeratoriaus serveryje saugomos tikslios kopijos ir gaunami atnaujinimai iš Leidėjo serverio. Periodiniai atnaujinimai daromi Prenumeratoriaus serveryje, kuris gali gražinti duomenis į Leidėjo serverį. Tai gali būti reikalinga Prenumeratoriaus serveriui, kad veiktų kaip Leidėjo serveris, ir iš naujo skelbti duomenis į kitus prenumeruojamus serverius.

Publikacija

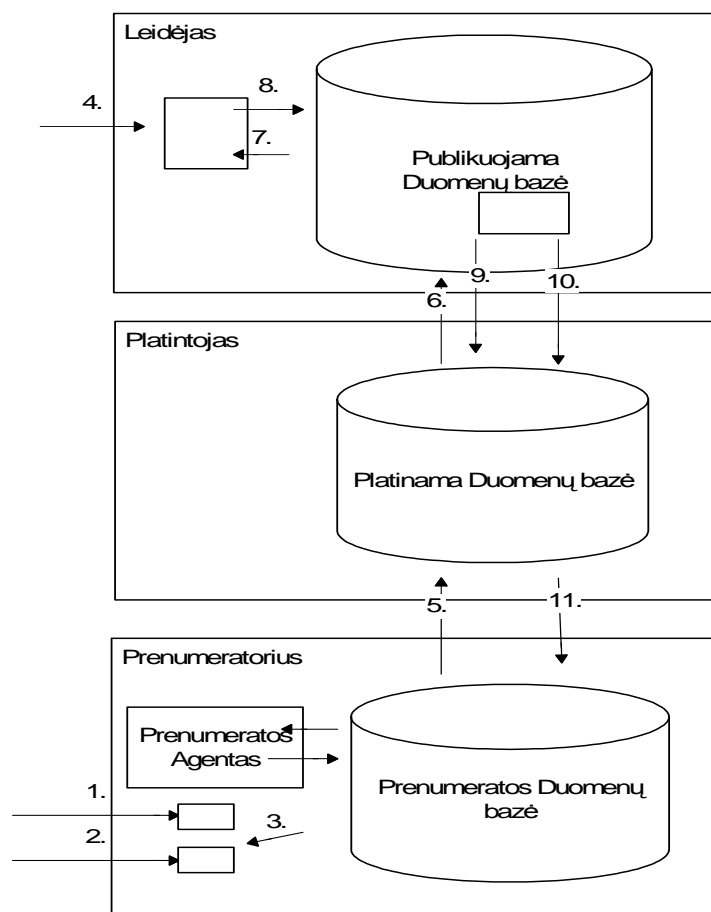
Leidėjo serveris turi straipsnių kolekciją, esančia publikacijos duomenų bazėje. Ši duomenų bazė pasako Leidėjo serveriui, kurie duomenys reikalingi siųsti į kitus serverius ar į prenumeruojančius serverius. Publikavimo duomenų bazė veikia kaip duomenų kodas, skirtas replikavimui. Bet kuri duomenų bazė yra naudojama kaip replikavimo kodas, kad būtų galima naudoti kaip Leidėjo serverį. Duomenų bazė, kuri yra publikuojama, turi turėti vieną ar kelias publikacijas. Publikacija yra vienetas (blokas), kuris turi vieną ar kelis straipsnius, kurie yra siunčiami į prenumeravimo serverius.

Straipsnis

Straipsnis yra bet koks duomenų grupavimas, kuris turi būti replikuojamas, jis yra publikacijos komponentas. Straipsnis gali turėti lentelių aibę ar lentelių poaibį. Straipsnis taip pat gali turėti eilučių aibę, stulpelių aibę, saugojamų procedūrų, peržiūras, indeksuotas peržiūras ar vartotojo apibrėžtas funkcijas.

Prenumeratos

Prenumeratoriaus serveriai turi apibrėžti jų prenumeratas, skirtas tam tikroms publikacijų aibės užsakymams, kad gautų momentinį vaizdą iš Leidėjo serverio. Visiems trims replikavimo tipams momentinis vaizdo failas yra daromas iš schemos ir pradinių duomenų publikavimo failų ir yra saugomas momentinio vaizdo aplanke. Vėlesni duomenų ar schemos pakeitimai yra perduodami iš Leidėjo serverio į Prenumeratoriaus serverį. Šitas procesas yra žinomas kaip duomenų apsikeitimas. Prenumeratos atvaizduoja skirtingus straipsnius į atitinkamas lenteles, esančias Prenumeratoriaus serveryje. Jie taip pat tiksliai apibrėžia, kai Prenumeratorių serveriai turi gražinti publikacijas iš publikuojančių serverių. Anoniminė prenumerata ir pavadinta prenumerata yra metodai, kurie atlieka duomenų pakeitimus publikacijoje. Anoniminėje prenumeratoje nėra informacijos apie prenumeruojamą serverį ar prenumeratą, kuri saugojama Leidėjo serveryje. Tai yra prenumeruojamų serverių visapusiška atsakomybė išlaikyti duomenų ir prenumeratų kelio (judėjimo) istoriją. Ši prenumerata dar yra žinoma kaip traukimo (Pull) prenumerata, kuri pavaizduota 6 paveiksle. Visa informacija apie operacijas saugojama Platintojo agente tol, kol įvyksta kitas duomenų apsikeitimas.[6]



6 pav. Duomenų leidimas su traukimo prenumerata

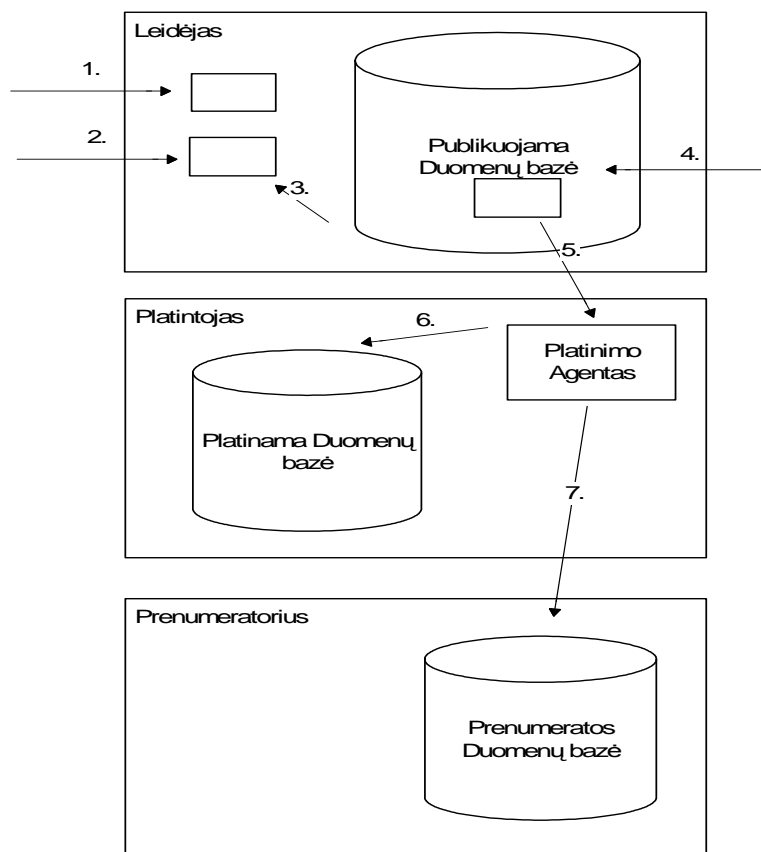
Paveikslo detalus paaiškinimas pateikiamas toliau:

1. Prenumeratorių sukūrimas Prenumeratoriaus serveryje;
2. Prenumeratų sukūrimas Prenumeratoriaus serveryje;
3. Leidėjas suteikia teisę (galimybę) prenumeratoriui ir prenumeratams;
4. Leidėjas gauna informaciją iš prenumeratorių ir prenumeratų;
5. Prenumeratorius reikalauja duomenų;
6. Platintojas siunčia reikalavimus Leidėjui;
7. Leidėjas peržiūri informaciją;
8. Leidėjas gauna atsakymą, kur informacija yra prieinama, o kur ne;
9. Jei informacija yra prieinama, Leidėjas prenumeratas siunčia Platintojui;
10. Leidėjas siunčia klaidos pranešimą ir duomenų istoriją Platintojui, prenumeratos duomenų bazė saugoja transakcijas ir registruoja atnaujinimus;
11. Platintojas siunčia transakcijas prenumeratoriui.

Pavadinta prenumerata yra ta, iš kurios Prenumeratoriaus serveris gali prieiti prie Leidėjo serveryje esančių duomenų. Ši prenumerata dar yra žinoma kaip stūmimo (Push) prenumerata, kurios veikimas pavaizduotas 7 paveiksle. Stūmimo prenumeratos yra sukuriamos Leidėjo serveryje. Leidėjo serveris išlaiko atmintyje prenumeratų valdymą ir gali dalinti pakeitimus, pareikalavus arba tolygiai, arba suplanuotais intervalais. Duomenų apsikeitimas stūmimo prenumeratoje yra tipiskai perduodamas tolygiai, kai tik pokyčiai įvyksta publikacijoje, nelaukiant Platintojo serverio padarytos užklauskos. Čia nereikia administruoti individualiai prenumeruojamų serverių – prenumeratos ar suliejimo agentas, kuris būna Prenumeratoriaus serveryje, vykdo planavimus.[6]

Paveikslo detalus paaiškinimas pateikiamas toliau:

1. Prenumeratorių sukūrimas Leidėjo serveryje;
2. Prenumeratų sukūrimas Leidėjo serveryje;
3. Leidėjas suteikia teisę (galimybę) prenumeratoriui ir prenumeratams;
4. Leidėjas perduoda prenumeratų pakeitimus į Platintojo serverį;
5. Platintojo agentas gauna replikuotus duomenys;
6. Platintojo agentas saugoja darbo istoriją ir registracija platinamojoje bazėje;
7. Po to Platintojo agentas vykdo transakcinį ir momentinį vaizdo replikavimą į tinkamą prenumeratorių.



7 pav. Duomenų leidimas su stūmimo prenumerata

Suliejimo replikavimas nutolusioje duomenų bazėje

Suliejimo replikavimas, panašus į transakcinį replikavimą, prasideda kartu su publikacijos momentinių vaizdinių duomenų bazės objektais ir duomenimis. Vėlesni duomenų pokyčiai ir schemos modifikavimas, kuriuos atlieka Leidėjas ir Premuneratoriai, yra pasiekiami su triggeriais. Premuneratoriaus duomenų apsikeitimas su Leidėju įvyksta, kai sujungia į tinklą ir yra pakeičiamos visos eilutės, kurios turi pakeitimų tarp Leidėjo ir Premuneratoriaus nuo paskutinio apsikeitimo įvykio. Suliejimo replikavimas yra naudojamas iš serverio į vartotojo aplinką. Suliejimo replikavimas yra tinkamas naudoti tokiose situacijose:

Keli Premuneratoriai gali atnaujinti duomenis įvairių laiku ir dalinti šituos pokyčius Leidėjui ir kitiems Premuneratoriams.

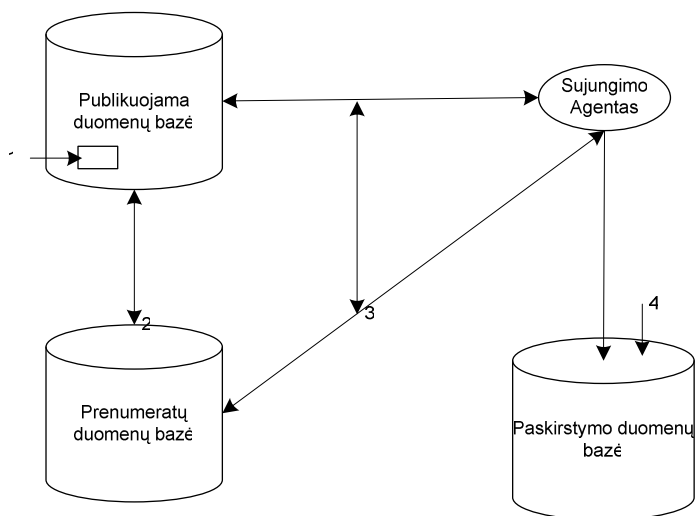
Premuneratoriams reikia gauti duomenis, kad padarytų pakeitimus neprisijungus ir vėliau perduoti pakeitimus su Leidėju ir kitais Premuneratoriais.

Kiekvienas Premuneratorius reikalauja skirtingų duomenų dalių.

Taikomoji programa reikalauja tinklo duomenų pokyčių, kad prisijungtų prie tarpinių duomenų būsenų.

Suliejimo replikavimas leidžia įvairiose vietose dirbti autonomiškai ir vėliau sujunti atnaujinimus į vieną, kad būtų gautas vienodas rezultatas. Kadangi atnaujinimai yra padaromi

daugiau nei viename mazge, kai kurie duomenys turi būti atnaujinami Leidėjo ir daugiau nei vieno Premuneratorių.[6] Visas suliejimo replikavimo procesas atvaizduotas 8 paveiksle.



8 pav. Suliejimo replikavimo veikimo principas

Paveikslo detalus paaiškinimas pateikiamas toliau:

1. Publikuojama duomenų bazė turi straipsnius Leidėjo serveryje.
2. Konflikto lentelės, pokyčių sekimo lentelės ir trigeriai yra pateikti publikacijos ir prenumeratos duomenų bazėje.
3. Suliejimo agentas perduoda duomenis.
4. Suliejimo Agentas rašo istoriją MSmerge_history lentelėje, kuri yra prenumeratos duomenų bazėje.

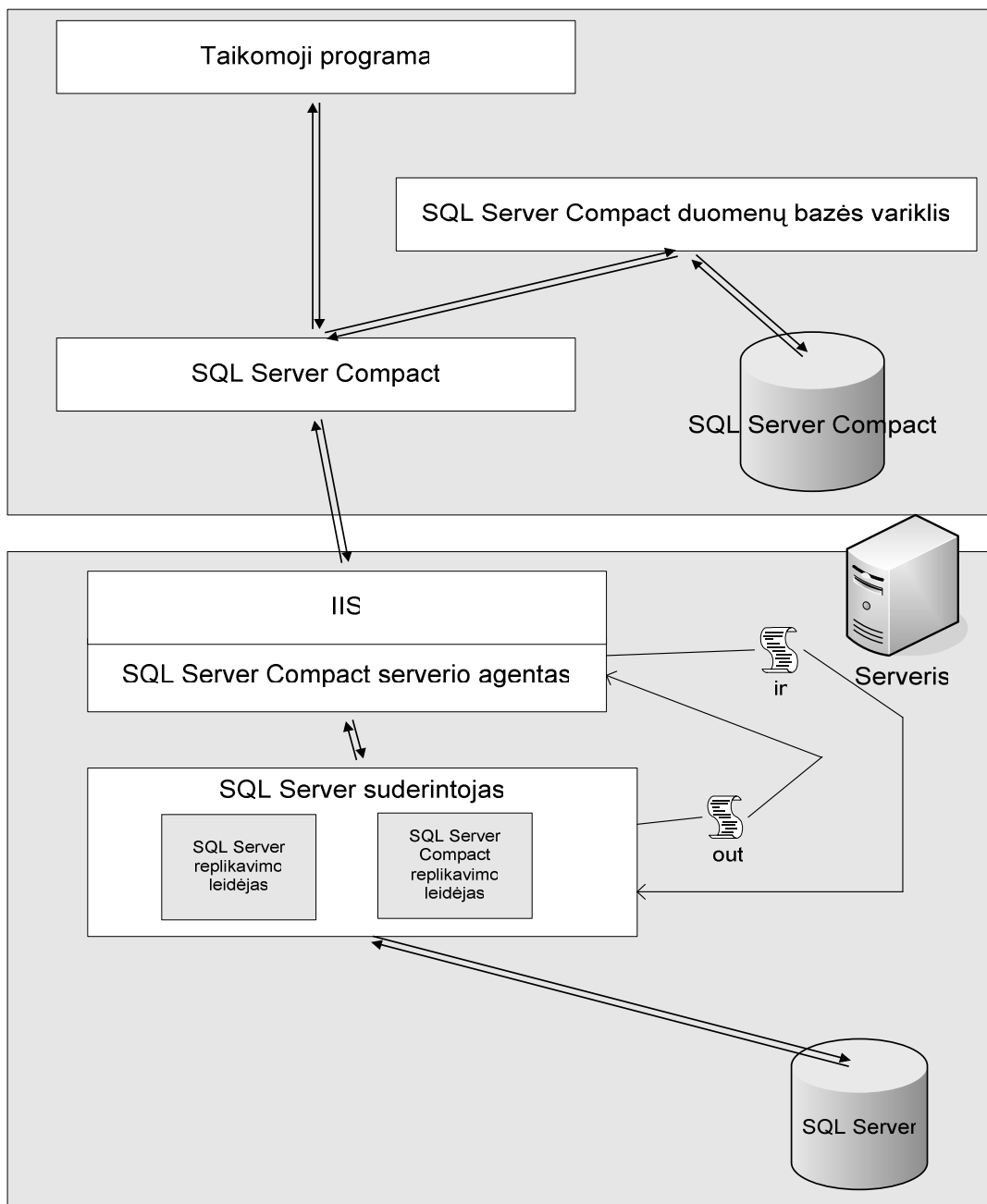
Suliejimo replikavimas mobilioje duomenų bazėje

Mobilios duomenų bazės replikavimas yra pagrįstas nutolusios duomenų bazės suliejimo replikavimu. Jo architektūros modelis pateikiamas 9 paveiksle. Šį procesą galima vykdyti naudojant nutolusią duomenų bazę.

Suliejimo replikavimas yra skirtas naudoti nešiojamiems įrenginiams, kadangi jie leidžia duomenis atnaujinti automatiškai ir nepriklauso nuo nešiojamo įrenginio ir serverio. Įrenginyje esantys duomenys su serveriu gali vykdyti duomenų apsikeitimą, kai yra susijungimas tarp jų. Tada siunčiami pakeitimai iš kliento į serverį ir atvirkščiai.

Suliejimo replikavimas reikalauja didesnių nustatymų ir palaikymų serveryje negu nutolusių duomenų prieigos paslauga, bet tai yra būdinga taikomosioms programoms, kurios yra nešiojamuose įrenginiuose. Šis replikavimas pasirūpina įdiegtomis ir įprastomis konfliktų sprendimo galimybėmis, leidžiančiomis duomenų replikavimą iš kelių lentelių vienu metu. Palaiko duomenų replikavimo nustatymus, tokius kaip straipsnių tipai ir filtravimas, kuris įrodo našumą, priklausantį nuo duomenų poreikių ir identiškios srities valdymo. Suliejimo replikavimas yra

sprendimas, kuris sprendžia konfliktus, kai duomenys platinami iš serverio į įrenginį ir atvirkščiai.[12]



9 pav. Suliejimo replikavimo architektūra

4 lentelė. Mobilios duomenų bazės perdavimo metodų palyginimas

	Nutolusių duomenų prieigos paslauga	Suliejimo replikavimas
Sinchronizavimas naudojant paslaugas	-	-
Heterogeninių duomenų bazių palaikymas	-	-
Laipsniško pokyčio sekimas	-	+
Konflikto suradimas ir sprendimas	-	+
Lengvai sukuriamų duomenų peržiūra kliente	-	+
Automatiškai inicijuojama schema ir duomenys	+	+
Didelių duomenų aibių palaikymas	+	+
Užklausų procesorius yra vietoje prieinamas	+	+
Automatiškai platinami schemos pokyčiai	-	+
Automatiškai paskirstomi duomenys	-	+
Naudojama įrenginyje	+	+

Lentelėje parodoma mobilios duomenų bazės perdavimo metodų skirtumai ir panašumai. Jie padeda pasirinkti duomenų perdavimo metodą, parankiausią kūrėjui.

5 lentelė. Mobilios duomenų bazės perdavimo metodų savybių palyginimas

Savybės	Suliejimo replikavimas	Nutolusių duomenų prieigos paslauga
Stebimų duomenų pakeitimų pokyčiai	Replikavimo pasikeitimai į ir iš serverio ir kliento Stulpelio lygio stebėjimas- mažinamas persiųstų duomenų kiekis Eilutės lygio stebėjimas – visos eilutės yra persiųstos	Duomenų išstūmimo pakeitimai iš kliento tik į serverį. Visiško kliento duomenų atnaujinimo poreikiai, kad gautume serverio pakeitimus. Eilutės lygio stebėjimas – visos eilutės yra persiųstos
Lentelių numeriai, iš kurių duomenys gali būti platinami	Sudėtinės lentelės – kaip keletas apibrėžtų publikacijų.	Viena lentelė per nutolusių duomenų prieigos paslaugos metodą
Lentelių tipai, kurie gali būti platinami	Skirtingi lentelių tipai, kad valdytų duomenų srautus. Lentelių pridėjimas ar atėmimas publikacijoje yra replikuojamas automatiškai per įdiegiamą į kliento prenumeratą	Lentelių pridėjimas ar atėmimas serveryje nėra automatiškai replikuojami kliento
Apribojimas ir indeksavimas	Integruoti apribojimai ir indeksai yra replikuojami iš serverio automatiškai	Integruoti apribojimai nėra replikuojami, indeksai nebūtinai visada replikuojami. Pridedamos schemos apibrėžimas turi apibrėžti kliento apribojimus arba kriterijus
Konfliktai	Konfliktuose esantys duomenys pakeičiami skirtingų vartotojų ir eilutės, kurios negali būti pavartotos. Konflikto sprendimas ir valdymas	Nutolusių duomenų prieigos paslauga neaptinka, jei duomenys buvo pakeisti skirtingų vartotojų. Konfliktai nėra valdomi, bet yra būtinai pranešami klaidų lentelės

	serveryje. Standartinis ir pasirinkto konflikto apsisprendimų palaikymas	klientui Nekonfliktiniai apsisprendimai
Schemos pokyčiai	Schemos pokyčiai gali replikuoti	Schemos pakeitimai nėra leidžiami.
Tapatumo stulpeliai	Rankinis ir automatinis identifikavimo diapazono nustatymas	Rankinis identifikavimo diapazono nustatymas.
Duomenų/schemos apibrėžimas	Įvykiai serveryje, kai publikavimas yra keičiamas ir automatiškai apibrėžtas kliento, kai prenumerata yra sukurta	Duomenys ir schemos apibrėžimai įvyksta kliente, kai duomenys perduodami iš serverio klientui.
Įrankiai	Vykdomieji įrankiai yra sukurti ir valdomi prenumeratų ir stebėjimo įrankiai valdo kelias prenumeratas, stebi našumą ir duomenų apsikeitimo laikus	Nepalaikomi įrankiai
Replikuojamų duomenų bazių modifikavimas	Lentelės ir stulpeliai pridedami į replikuojamą duomenų bazę, kad valdytų replikavimą.	Neatliekami pakeitimai replikuojamoje duomenų bazėje.

Lentelė vaizduoja duomenų apsikeitimo metodų savybių skirtumus ir panašumus. Kokiomis savybėmis pasižymi kiekvienas perdavimo metodas.

2.3 Tiesioginės nutolusios duomenų bazės prieigos naudojimo analizė

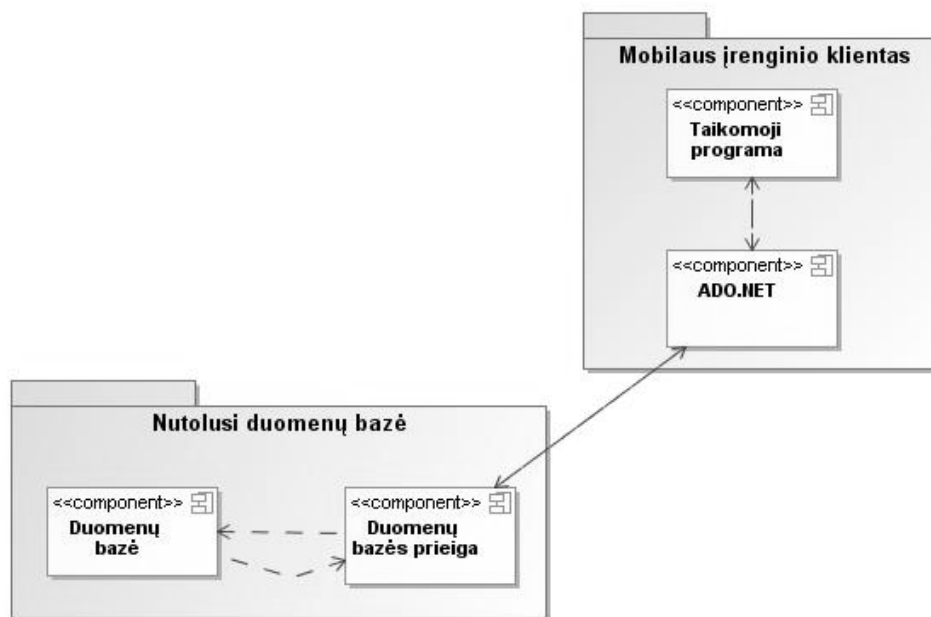
Šis duomenų apsikeitimo metodas yra paprasčiausias ir reikalauja mažiausiai programinio kodo. Juo vartotojas tiesiogiai kreipiasi į nutolusią duomenų bazę ir atlieka reikiamus veiksmus su duomenimis. Kriterijus, kuris reikalingas šitam duomenų perdavimui, yra nuolatinis interneto ryšys su nutolusia duomenų baze. Duomenys yra perduodami tik esant tarpusavio ryšiui.

Tiesioginės nutolusios duomenų bazės naudojimo architektūrą sudaro trys komponentai: klientas, prieiga prie nutolusios duomenų bazės ir nutolusi duomenų bazė, kuri pateikta 10 paveiksle.

Kliento komponentas skirtas valdyti duomenis, esančius nutolusioje duomenų bazėje. Sukurta sistema naudodama ADO.NET komponentą, skirtą duomenims siųsti tiesiai į nutolusią duomenų bazę.

Prieiga prie nutolusios duomenų bazės turi ADO.NET komponentą, skirtą prisijungti prie duomenų resursų, tokių kaip Microsoft SQL Server. Tai yra biblioteka, kuri naudojama kartu su Microsoft. NET Framework. Jis leidžia naudoti interneto paslaugą, per kurią sukurta sistema prisijungia prie duomenų bazės, vykdo komandas ir gauna rezultatus.

Duomenų bazė, kuri saugoma nutolusiame duomenų bazės serveryje, yra naudojamų atlikti eksperimentui duomenų aibė.



10 pav. Tiesioginės nutolusios duomenų bazės naudojimo architektūra

2.4 Mobilias duomenų bazes realizuojančių įrankių analizė

Microsoft SQL Server 2005 Compact Edition (SQL Server Compact Edition) yra kompaktiška duomenų bazė, kuri plėtojama staliniuose, intelektualiuose kompiuteriuose.

Microsoft SQL Server 2005 Compact Edition yra kompaktiška reliacinė duomenų bazė, kuri reikalauja mažesnės nei 2MB disko vietos ir tik apie 5 MB atminties. Be to, apdoroja mobilioje duomenų bazėje esančius duomenis, kai vyksta duomenų apsikeitimas su nutolusia duomenų baze. SQL Server Compact Edition pasirūpina savybėmis, kurios leidžia kūrėjams kurti taikomąsias programas, palaikančias „atsitiktinio suliejimo“ aplinką. Mobilus duomenų bazė dalinasi bendru programavimo modeliu su kitais SQL Server leidiniais. Tai suteikia galimybę kūrėjams perduoti įgūdžius (sugebėjimus) ir žinias greitai ir lengvai.

SQL Server Compact Edition savybės:

Kompaktiškas duomenų bazės variklis ir tvirtas užklausos optimizatorius;

Palaikomas suliejimo replikavimas ir nutolusių duomenų prieiga;

Integravimas su Microsoft SQL Server 2005;

Suliejimo replikavimas ir nutolusių duomenų prieiga, skirta apsikeisti duomenimis;

Microsoft .NET Framework ir .NET Compact Framework duomenų tiekėjas, skirtas SQL Server Compact Edition (System.Data.SqlServerCe);

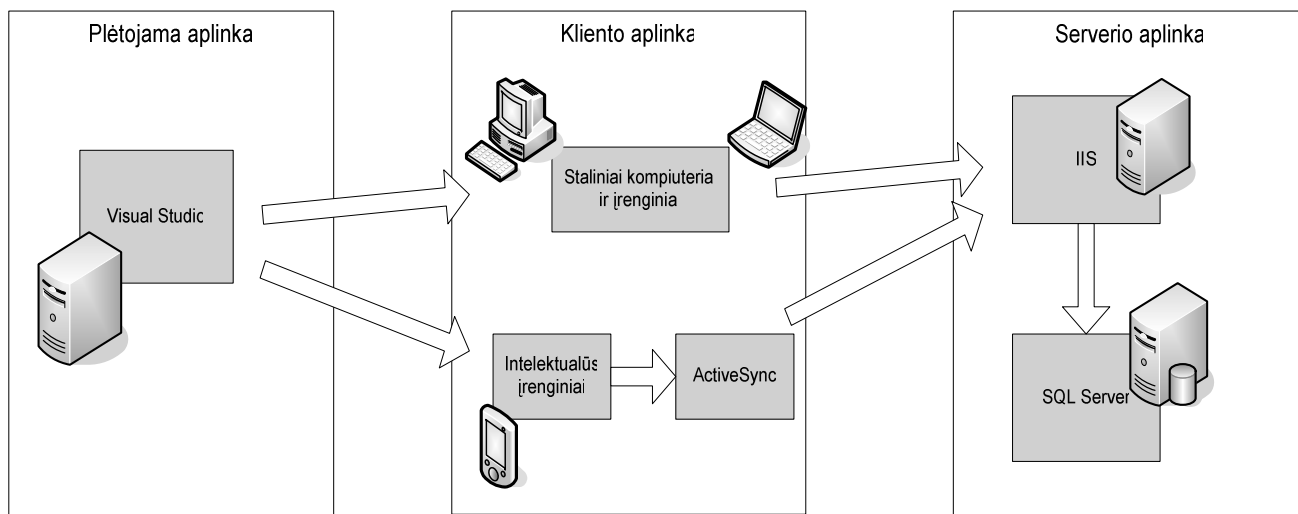
Palaikomas Microsoft ADO.NET ir OLE DB tiekėjas, skirtas SQL Server Compact Edition;

SQL sintaksės poaibis;

Plėtojama kaip įdiegta duomenų bazė staliniuose kompiuteriuose ir mobiliuose įrenginiuose.

SQL Server Compact Edition architektūra

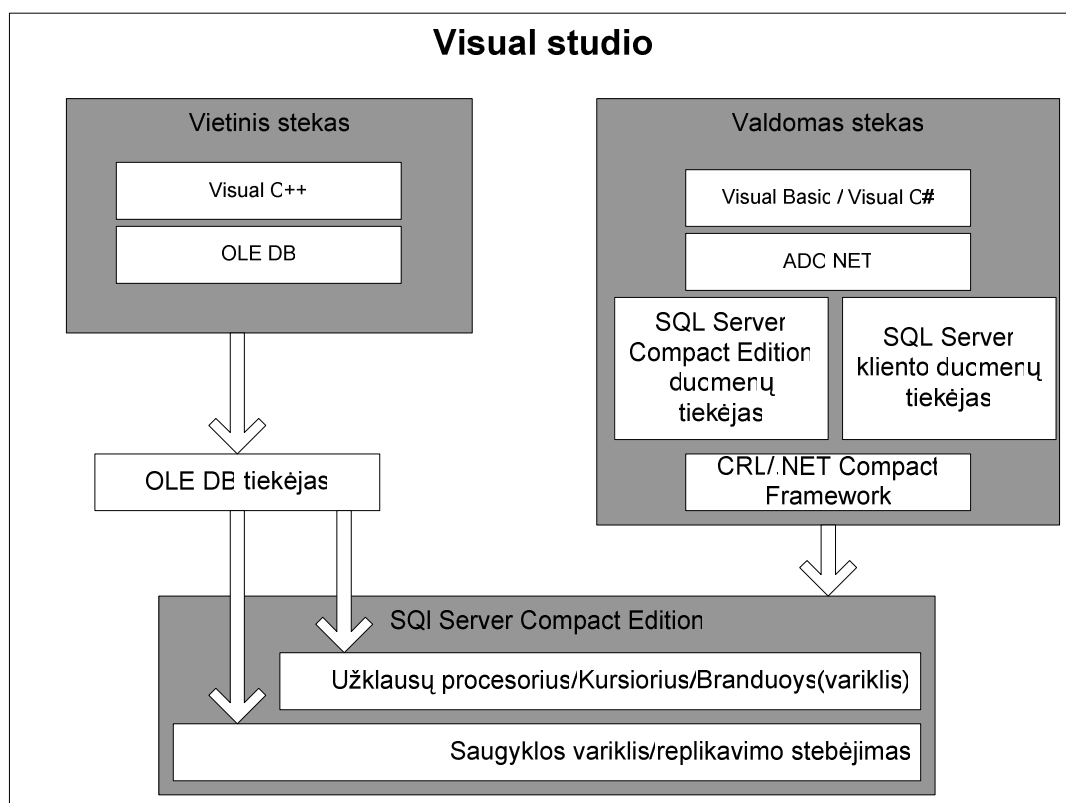
SQL Server Compact Edition architektūra sudaryta iš plėtojamos aplinkos ir kliento/serverio aplinkos, kuri pateikta 11 paveiksle.



11 pav. SQL Server Compact Edition architektūra

Plėtojimo aplinka

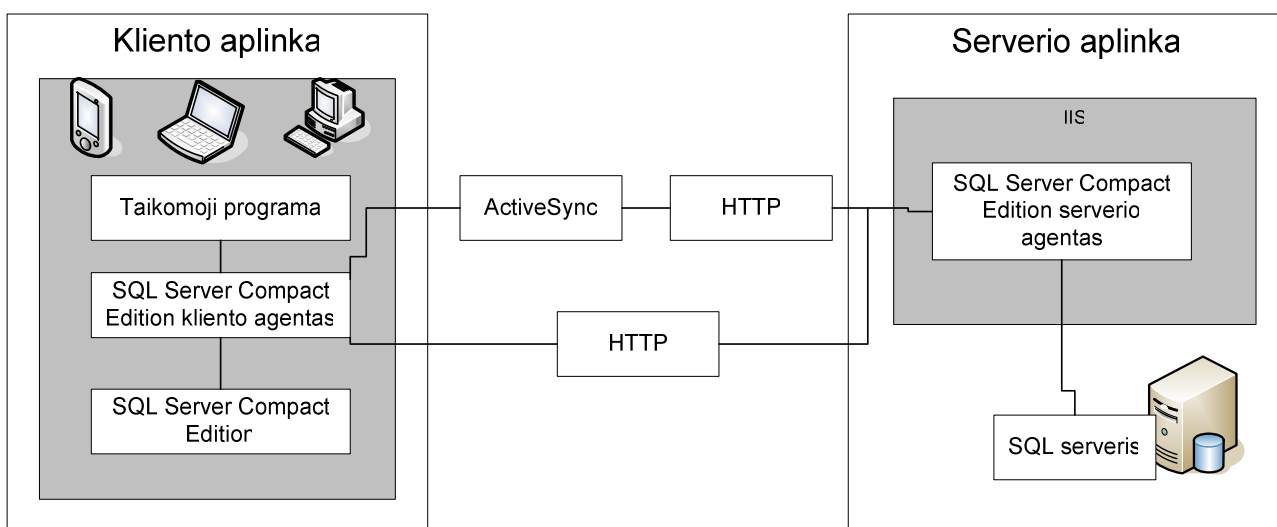
Ją turi kompiuteris, kuriame yra kuriama ir plėtojama taikomoji programa. Šis kompiuteris turi turėti Microsoft Visual Studio 2005 kartu su .NET Compact Framework, kuris leidžia kurti taikomas programas, skirtas SQL Server Compact Edition. Galima kurti valdomąsias taikomas programas, naudojant Microsoft Visual Basic arba C#, jo architektūra pavaizduota 12 paveiksle.



12 pav. Plėtojimo aplinkos architektūra

Kliento/serverio aplinka

Kliento aplinka yra sukurta taip, kad būtų palaikoma iš vieno arba iš kelių įrenginių, kuriuose yra plėtojamos taikomosios programos su SQL Server Compact Edition, jo architektūra pateikta 13 paveiksle. Kai įrenginys negali susijungti į tinklą, tada yra naudojamas Microsoft ActiveSync, kuris sujungia SQL Server Compact Edition į serverio aplinką. Serverio aplinka yra sukurama viename ar keliuose kompiuteriuose, kur veikia (IIS) ir Microsoft SQL Server reikalavimas arba duomenų dalinimo heterogeniniais duomenimis šaltinis. Galima naudoti IIS ir SQL Server konkrečiame kompiuteriujė arba nustatyti juos per kelis kompiuterius. IIS yra reikalingas sujunti duomenų apsijungimui tarp serverių ir klientų.



13 pav. Kliento/serverio aplinkos architektūra

Taikomoji programa

Taikomoji programa yra sukurta naudojant NET Compact Framework ir Microsoft Visual Studio 2005 kalbą, tokią kaip Microsoft Visual Basic arba Microsoft Visual C#, arba skirta Microsoft Visual C++ įrenginiams.

SQL Server Compact Edition

SQL Server Compact Edition turi įrankius, skirtus įdiegti, konfigūruoti, sujungti ir duomenų prieigai, ir modifikavimui, užklauso procesorių, duomenų bazės saugyklos variklį ir programavimo sąsają, kuri naudojama kurti taikomąsias programas, kad būtų galimapieiti prie SQL Server Compact Edition duomenų.

SQL Server Compact Edition duomenų bazės variklis

Jis valdo SQL Server Compact Edition duomenų saugojimą palaikančiuose įrenginiuose. Duomenų bazės variklis gali stebėti visus duomenų bazės įrašus, kurie yra įterpti, atnaujinti arba ištrinti. Stebėjimo funkcionalumas yra prieinamas, kai naudojamas vienas iš dviejų sujungimo sprendimų: suliejimo replikavimas ir nutolusių duomenų prieigos paslauga.

SQL Server Compact Edition kliento agentas

Jis yra pagrindinis komponentas, skirtas sujungti duomenis, esančius mobiliuose ir nutolusiose duomenų bazėse. SQL Server Compact Edition objektai:

Replication objektas;

RemoteDataAccess objektas;

Engine objektas.

Naudojant visus tris šiuos objektus, galima programiniu kodu valdyti sujungimus su SQL serveriu.

SQL Server Compact Edition serverio agentas

Jis rankiniu būdu nustato HTTP reikalavimus, skirtus SQL Server Compact Edition kliento agentui. SQL Server Compact Edition serverio agentas sujungia į SQL serverį ir grąžina duomenis ir schemas informaciją į SQL Server Compact Edition kliento agentą per HTTP. SQL serverio sujungimo komponentai yra įtraukti į šitą procesą. Jie yra nustatomi kompiuteryje, kuriame veikia IIS.

Interneto Informacinė paslauga (IIS)

IIS pasirūpina integruotomis tinklalapio serverio galimybėmis. Visokio dydžių organizacijos naudoja IIS, kad priimti ir valdyti tinklalapio puslapius internetu arba intranetu, priimti ir valdyti FTP vietas (saitus) ir siųsti naujienas arba elektroninius laiškus naudojant tinklo žinučių perdavimo protokolą (NNTP) ir paprasto pašto perdavimo protokolą (SMTP). Kai naudojamas su SQL Server Compact Edition, IIS pasirūpina protokolu, su kuriuo įrenginiai gali sujungti perdavimui ir duomenų keitimuisi naudojant nutolusių duomenų prieigos paslauga arba replikavimo technologijas.

Microsoft SQL Server

Jis yra komponentų aibė, kad būtų galima dirbti kartu su tinkamomis duomenų saugyklomis ir analizuoti didžiausius Web saitų poreikius ir duomenų vykdymo sistemas. Tai reliacinis duomenų valdymas, duomenų sandėliavimas ir verslo sumanūs komponentai.

Savybės, kurios padidina mobilios duomenų bazės našumą ir patikimumą

SQL Server Compact Edition turi keletą naujų ir atnaujintų savybių, kurios padidina duomenų bazės našumą ir patikimumą. Šituos patobulinimus galima suskirstyti į pagrindines savybių kategorijas: duomenų apsikeitimas, saugyklos (saugojimo) variklis ir užklausų procesorius.[12]

Duomenų apsikeitimas

Kelių prenumeratų palaikymas

IS kūrėjas turi sukurti tokias prenumeratas, skirtas duomenų apdorojimui, kurios yra vykdomos su kiekviena publikacija. Taip pat tos sukurtos publikacijos turi būti gautos ar sudarytos

iš kelių publikacijų. Pasinaudojus taikomąja programa, jos suvykdomos. Dirbant su SQL Server Compact Edition, kelios prenumeratos gali būti laikomos vienoje duomenų bazėje, tokiu būdu yra sumažinami programavimo poreikiai.

Kelių prenumeratų palaikymo privalumai:

Galima sumažinti straipsnių skaičių kiekvienoje publikacijoje.

Galima valdyti, kai duomenys yra sinchronizuojami iš kiekvienos publikacijos.

Užklauso vykdytas keliose lentelėse kiekvienoje publikacijoje.

Kelių vartotojų palaikymas

Palaikomas kelių taikomųjų programų priėjimas prie tos pačios duomenų bazės konkrečiu laiku. SQL Server Compact Edition pasirūpina kelių vartotojų palaikymu. Kelių vartotojų palaikymas suteikia galimybę keliems duomenų bazės vartotojams apsikeisti duomenimis vienu metu. Tada kyla problemos su duomenimis, kurie gaunami iš Leidėjo.

6 lentelė. Kelių vartotojų priėjimo problemos

Savybė	Problema
Užrakinimas	Apsikeitimo metu pokyčiai iš Leidėjo gali būti nepritaikomi į Prenumeratoriaus duomenų bazę, kadangi duomenys užrakinami. Jei duomenų pokyčiai yra padaromi Prenumeratoriuje ir čia yra ilgai trunkantys, užrakinami, duomenų apsikeitimas gali būti nutrauktas.
Patvirtinimas	Jei eilučių skaičius SQL Server Compact Edition duomenų bazėje yra pakeistas duomenų apsikeitimo metu, patvirtinimas bus nutrauktas.
Pakartotinis paleidimas	Prenumeratos pakartotiniu paleidimo metu replikavimo lygmuo ištrinamas ir perkuriami visi replikuojami vartotojai ir sistemos lentelės. Su SQL Server prenumeratose bet kokie pakeitimai daromi po duomenų apsikeitimo pradžios ir prarandami, kai įvyksta pakartotinis paleidimas.
Schemas pokyčiai	Visos duomenų apibrėžimo kalbos (DDL) operacijos turi turėti ypatingus (privilegiuotus) priėjimus prie lentelės. Schemas pakeitimas bus nutraukiamas, jei lentelė yra naudojama kitame procese.
Pokyčiai duomenų apsikeitimo metu	Jei duomenų pokyčiai įvyksta duomenų apsikeitimo metu, tai pakeitimas bus siunčiamas kito duomenų apsikeitimo metu. Jei duomenų apsikeitimo sesiją sukelia vietinis konfliktas, eilutė yra skaidoma į eilutės lygį, netgi jei straipsnis yra stulpelio lygio.

Lentelėje aprašytos bendros SQL Server Compact Edition, kurios naudojamos, kai keli vartotojai nori vienu metu naudotis ta pačia duomenų bazė. Jei įvyksta bent viena iš šių savybių, kitas vartotojas negali atlikti duomenų apsikeitimo.

Saugyklos variklis

SQL Server Compact Edition naudojama savybė, kuri padidina mobilios taikomosios programos patikimumą ir našumą. Saugyklos variklio savybę apima:

Patikslintas(ištaisytas) saugyklos variklis

SQL Server Compact Edition duomenų bazės variklis pasirūpina kontroliuojama prieiga ir greitu transakcijų vykdymu, kad užtikrintų svarbiausias operacijas su duomenimis taikomosios programos aplinkoje. Jo atliekami veiksmai:

Valdyti failą, kuriame saugoma duomenų bazė ir naudoti vietą faile;

Įdiegti ir skaityti fizinius puslapius, kurie yra skirti duomenims saugoti;

Valdyti duomenų buferius ir visus I/O į fizinius failus;

Valdyti transakcijas ir naudoti užrakinimus, kad kontroliuotų vienu metu vartotojų prieiga į eilutes ir schemą, esančią duomenų bazėje;

Užtikinti transakcijų nedalumą, nuoseklumą, izoliavimą ir patvarumą (ACID);

Kurti ir palaikyti indekso struktūrą;

Palaikyti duomenų bazės užšifravimą ir slaptažodžio saugumą;

Kelių vartotojų palaikymas

Kad kelios taikomosios programos palaikytų priėjimą prie tam tikros duomenų bazės konkrečiu laiku, SQL Server Compact pasirūpina eilutės lygio duomenų užrakinimu, kuris padeda garantuoti duomenų integralumą.

Kai keli vartotojai arba taikomosios programos prieina prie tam tikrų duomenų konkrečiu laiku, užrakinimas sutrukdo viena laikius pokyčius į apdorojamus duomenis. Užraktas yra valdomas naudojant SQL Server Compact Edition duomenų bazės variklį. Užraktas yra automatiškai sukombinuotas ir paleidžiamas vartotojo pagrįstais veiksmais. Jei užrakinimas nėra naudojamas ir keli vartotojai atnaujina tam tikrus duomenis konkrečiu laiku, tai duomenys, esantys duomenų bazėje, tampa logiškai neteisingi. Jei tai įvyksta, užklausos vykdymas gali pateikti netikėtus rezultatus. Taikomoji programa niekada nevykdo tiesioginio reikalavimo, skirto užraktui. Tuo metu, kai transakcijos pradeda darbą ir komandos vykdo užklausų kalbą, duomenų manipuliavimo kalbą arba duomenų apibrėžimo kalbą transakcijos viduje, SQL Server Compact Edition užraktai reikalauja padėti apsaugoti resursus izoliavimo lygiu.

SQL Server Compact Edition automatiškai nulemia, kai užraktai sutinkami ir paleidžiami visiems sprendimams. Eilutės lygio užrakinimas yra naudojamas duomenų puslapiams, o puslapio lygio užrakinimas yra naudojamas indeksų puslapiams. Kad būtų išsaugoti sistemos resursai,

užrakto valdytojas automatiškai vykdo užrakto išplėtimą, kai konfigūracijos pradžia eilučių skaičių užraktai yra viršyti.

SQL Server Compact Edition užrakto resursai:

Eilutės;

4 KB duomenų ir indeksų puslapiai;

Lentelės schemas;

Lentelės;

Duomenų bazės.

Tuščių puslapių automatinis pakartotinis naudojimas

SQL Server Compact Edition palaiko automatinio sumažėjimo savybę. Tai automatiškai gražina nenaudojamus duomenų puslapius ir saugoja įrenginio saugyklos vietoje. Kad būtų suspausta duomenų bazė, reikia sukurti nauja duomenų bazę ir kopijuoti visus objektus iš pirminės duomenų bazės į naudojamą duomenų bazę. Suspaudimas nėra inicijuotas automatiškai. Automatiškai nustatomi duomenų bazės failo dydžiai, kurie vadinami AutoShrink. Šis metodas nenaudoja procesoriaus laiko ir atminties, jo sudarymas ypač tinkamas įrenginiams ir mobilių duomenų bazės produktams. AutoShrink technika perduoda puslapius failo viduje taip, kad viskas, kas tuščia arba neleistina puslapiuose, yra ribojama išdėstymu failo pabaigoje. Tušti puslapiai yra nutraukiami. Nutraukiami puslapiai yra prieinami duomenų bazės failo sistemos, kai yra naudojami. Gražinant nutraukiamus puslapius į duomenų bazės failo sistemą, padidinama jo sistemos vieta.

Užklausų procesorius

Kaštais pagrįsta optimizacija

Užklausos procesorius duoda statistinius privalumus, kurie palaikomi saugyklos variklyje, kad sukurtų prasmingai geresnius kaštais grįstus užklausos planus. SQL Server Compact Edition užklausos procesorius gramatiškai nagrinėja, kompiliuoja, optimizuoja ir vykdo SQL išraiškas, užklausas ir komandas. SQL gramatika, kuri naudojama su SQL Server Compact Edition, yra Transact-SQL gramatikos poaibį palaikanti Microsoft SQL Server 2005. Kai vyksta gramatinis nagrinėjimas, užklausos sintaksė yra tvirtinama ir duomenų struktūros simbolizuoja, kad užklausos gramatinis nagrinėjimas yra įdiegtas. Tada užklausos procesorius kompiliuoja ir optimizuoja užklausa. Kai užklausa optimizuojama, procesorius kuria vykdymo planą, skirtą užklausiai.

SQL Server Compact Edition užklausos optimizatorius laiko (svarsto) visus prieinamus indeksus, kai vykdoma užklausa, kuri turi:

Duomenų manipuliavimo kalbos sakinius;

Sujungti ir tvirtinti su ORDER BY;

GROUP BY ir DISTINCT sakinius.

SQL Server Compact Edition užklausos procesorius automatiškai kuria statistinę informaciją apie reikšmes, platinamas indeksuose. Jis naudoja informaciją, kuri padeda nustatyti

optimalią strategiją įvertinant užklausas. Užklausos procesorius naudoja statistikas, kad padėtų atrinkti labiausiai būdingus ar tinkamus indeksus veiksmingam vykdymui. Kūrėjui, naudojant SQL Server Compact Edition, galima peržiūrėti planą ir tobulinti užklausas, kad būtų pasiektas geriausias rezultatas.[12]

7 lentelė. SQL Server Compact 3.5 ir SQL Server 2005 skirtumai

Transact-SQL komanda, savybė, duomenų tipai	SQL Server Compact 3.5	SQL Server 2005
SET DEFAULT nustatymas naudojant ALTER TABLE ALTER COLUMN.	Palaikoma	Nepalaikoma
DISTINCT sujungimas	Nepalaikoma	Palaikoma
HAVING Clause (SQL Server Compact) išlyga (punktas) gali turėti sudėtinę išraišką, kai naudojama sugrupuoti arba sujunti užklausas	Tik paprasto stulpelio informacija	Palaikoma
WITH TIES ir PERCENT sakiniai, naudojami su TOP	Nepalaikoma	Palaikoma
Stulpeliai kaip argumentai TOP sakiniai	Nepalaikoma	Palaikoma
TOP (SQL Server Compact)(N) limitas.	Pritaikyti [0, MAX_INT] skirtas TOP	Pritaikyti [0, MAX_BIGINT] skirtas TOP
ORDER BY	Nepalaikoma	Palaikoma
UPDATE STATISTICS (SQL Server Compact) ON <TABLE NAME>	Palaikomas	Palaikoma
Duomenų bazės vardų priskyrimas su „=“	Nepalaikoma	Palaikoma
Duomenų bazės vardų priskyrimas su AS	Palaikoma	Palaikoma
IDENTITY savybės keitimas	Palaikoma	Nepalaikoma
Realūs, piniginiai ir skaitmeniniai duomenų tipai	Nepalaikoma	Palaikomas
Bendri reikšmių sakiniai, turintys ntext arba paveikslo duomenų tipus	Nepalaikoma	Palaikomas
Dideli objektai	Nepalaikoma	Palaikoma
Ntext ir paveikslo tipai negali būti naudojami su WHERE, HAVING, GROUP BY, ON, arba IN sakiniiais	Nepalaikoma	Palaikoma
Procedūros ir trigeriai	Nepalaikoma	Palaikoma
Peržiūros	Nepalaikoma	Palaikoma
Vartotojo nustatytos funkcijos	Nepalaikoma	Palaikoma
Paslaugos agentas	Nepalaikoma	Palaikoma
Biblioteka, skirta paleisti SQL	Nepalaikoma	Palaikoma
Lentelės vertinimo parametrai	Nepalaikoma	Palaikoma
Filtruojami indeksai	Nepalaikoma	Palaikoma
Erdviniai indeksai	Nepalaikoma	Palaikoma
Numatyta mažo prieinamumo indekso reikšmė	Palaikoma	Nepalaikoma
Duomenų suspaudimas	Nepalaikomas	Palaikomas

Lentelėje parodyti svarbiausiai skirtumai tarp mobilios ir nutolusios DBVS. Išskirti svarbiausi kriterijų skirtumai.

8 lentelė. Microsoft SQL Server 2005 Compact Edition charakteristika

Saugykla	Stulpelio pavadinimas	128 simboliai
	Stulpeliai lentelėje	1024
	Eilutės dydis	8060 baitai
	Duomenų bazės slaptažodis	40 simbolių
	Duomenų bazės dydis	4 GB
	Duomenų bazės dydžio padidinimas	1 – lapas ar 16 lapų pokytis, priklausantis nuo lentelės dydžio
	Puslapio dydis	4 KB
	Sesijos	256
	BLOB stulpelio dydis	2 GB
	Lentelės dydis	128 simboliai
	Lentelės pavadinimas	Nelimituojama
Užklausa	Simbolių eilutė SQL sakinyje	Nelimituojama
	Stulpeliai žymeklyje, kursoriuje	1024
	Stulpeliai ORDER BY, GROUP BY ar DISTINCT sakinyje	1024
	Įdėtas lygių sub-užklausų	Nelimituojama
	Pavadinti parametrai	Palaikoma
	Operandai užklausoje	Nelimituojama
	Sujungtos lentelės	Nelimituojama
Indeksai	BLOB stulpeliai	Negali būti indeksuojama
	Baitai indeksuotame rakte	512
	Stulpeliai indeksuose	16
	Indeksai per lentelę	249
Apribojimai	PRIMARY KEY, UNIQUE, FOREIGN KEY	Palaikoma
	Apribojimai per lentelę	249

Lentelėje parodoma naudojamos mobilios DBVS charakteristika. Svarbiausi kriterijai ir apribojimai joje.

2.5 Duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodų ir priemonių analizės išvados

1. Atlikus literatūros analizę, buvo nustatyti trys metodai (nutolusių duomenų paslauga, tiesioginės nutolusios duomenų bazės naudojimas ir suliejimo replikavimas), kurie leidžia apsikeisti duomenimis tarp DBVS ir mobilaus įrenginio.

2. Atlikus duomenų apsikeitimo tarp DBVS ir mobilaus įrenginių metodų analizę, paaiškėjo, kad suliejimo replikavimas vykdo automatišką ir nuolatinį duomenų apsikeitimą tarp mobiliojo įrenginio ir nutolusios DBVS. Pagal savo veikimo pobūdį ir našumą šis metodas labai artimas nutolusių duomenų prieigos metodei. Todėl eksperimentiškai suliejimo replikavimas papildomai tiriamas nebus.

3. Atlikus esamų mobilių duomenų bazių valdymo sistemų analizę tyrimui pasirinkta Microsoft kompanijos produktų šeima, kadangi nemokamai galima pasinaudoti visais reikiama produktais.

3. Duomenų apsisikeitimo tarp DBVS ir mobilaus įrenginio metodo pasirinkimo procesas

Atlikus literatūros analizę, buvo suformuluoti svarbiausi kriterijai, pagal kuriuos IS kūrėjas gali nuspręsti, kuris duomenų apsisikeitimo metodas tinkamiausias.

Žemiau pateikiami nustatyti kriterijai:

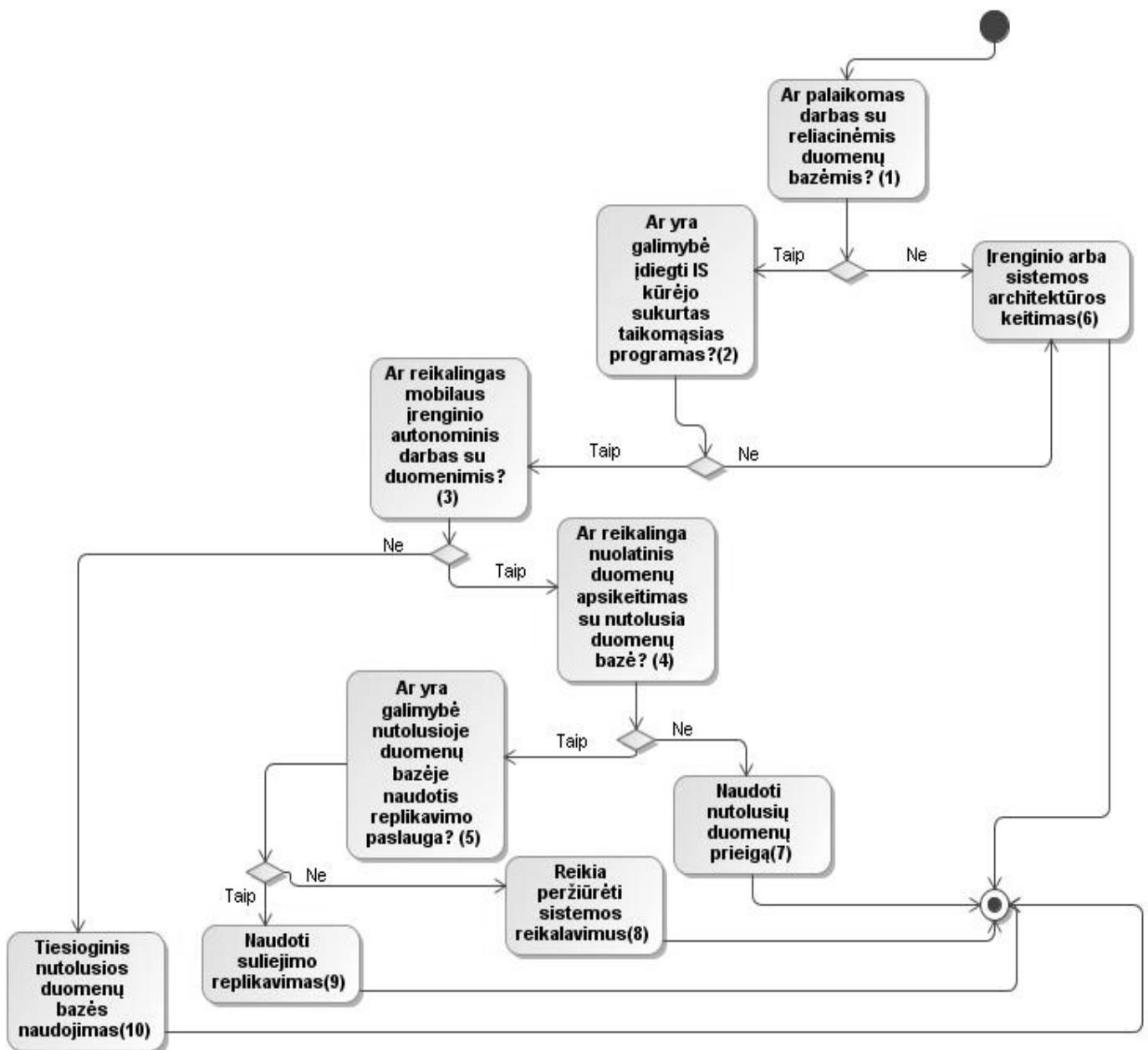
- Reliacinių duomenų bazių palaikymas, jas turi palaikyti mobilieji įrenginiai;
- Mobilios informacinės sistemos darbo vieta, pritaikyta konkrečiam duomenų apsisikeitimo metodui;
- Duomenų bazės duomenys galima apdoroti mobilioje duomenų bazėje ir tiesiogiai kreiptis į nutolusią duomenų bazę;
- Nuolatinė internetinė prieiga. Kai kuriems apsisikeitimo būdams reikalinga nuolatinė prieiga, kad nuosekliai vyktų procesas ir būtų galima tiksliai apskaičiuoti laiko kaštus;
- Palaikoma duomenų replikavimo paslauga, tam, kad būtų atliekama ši paslauga, reikia atlikti papildomus nustatymus nutolusioje duomenų bazėje.

Atsižvelgiant į šiuos kriterijus, buvo sukurtas duomenų apsisikeitimo tarp DBVS ir mobilaus įrenginio pasirinkimo procesas, kuris pateikiamas 14 paveiksle. Juo pasinaudojęs kūrėjas ar projektuotas galės pasirinkti vieną iš variantų duomenų perdavimui tarp mobiliame įrenginyje esančios IS mobilios darbo vietos ir nutolusios DBVS.

Duomenų apsisikeitimo tarp DBVS ir mobilaus pasirinkimo procesas padeda kūrėjui pasirinkti norimą naudoti perdavimo metodą. Pasinaudojus proceso kriterijais, galima lengviau nuspręsti, kuris metodas parankiausias. Toliau pateikiamas kiekvieno žingsnio naudojimo aprašas:

1) žingsnio metu nustatoma, ar palaikomas darbas su reliacinėmis duomenų bazėmis. Mobilus įrenginys turi naudoti reliacines duomenų bazes, skirtas duomenų apdorojimui, kad galėtų būti vykdomas duomenų apsisikeitimo procesas tarp DBVS ir mobilaus įrenginio. Jei žingsnis tenkinamas, yra pasirenkamas kitas žingsnis. Priešingu atveju IS kūrėjui siūloma keisti mobilių įrenginį ar duomenų apsisikeitimo proceso architektūrą

2) žingsnio metu nustatoma, ar yra galimybė įdiegti IS kūrėjo sukurtas taikomąsias programas. Duomenų apsisikeitimo metodai yra vykdomi pasinaudojant sukurtomis taikomosiomis programomis. Jei IS kūrėjas negali sukurti pritaiktos programos, tai siūloma keisti mobilių įrenginį arba duomenų apsisikeitimo proceso architektūrą. Jei žingsnis yra tenkinamas einama prie kito.



14 pav. Duomenų apsikeitimo tarp DBVS ir mobilus įrenginio pasirinkimo procesas

3) žingsnio metu nustatoma, ar palaikomas darbas su reliacinėmis duomenų bazėmis. Mobilus įrenginys turi naudoti reliacines duomenų bazės, skirtas duomenų apdorojimui, kad galėtų būti vykdomas duomenų apsikeitimo procesas tarp DBVS ir mobilus įrenginio. Jei žingsnis tenkinamas, yra pasirenkamas kitas žingsnis. Priešingu atveju IS kūrėjui siūloma keisti mobilių įrenginį ar duomenų apsikeitimo proceso architektūrą

4) žingsnio metu nustatoma, ar yra galimybė įdiegti IS kūrėjo sukurtas taikomas programas. Duomenų apsikeitimo metodai yra vykdomi pasinaudojant sukurtomis taikomosiomis programomis. Jei IS kūrėjas negali sukurti pritaikytos programos, tai siūloma keisti mobilių įrenginį arba duomenų apsikeitimo proceso architektūrą. Jei žingsnis yra tenkinamas, einama prie kito.

5) žingsnio metu nustatoma, ar reikalingas mobilaus įrenginio autonominis darbas su duomenimis. Yra galimybė duomenis apdoroti mobilioje duomenų bazėje, jei duomenų apskaitimo metodas palaiko prieigą prie jų. Jei duomenų saugojimas mobiliajame įrenginyje nereikalingas, tai kūrėjas gali naudoti tiesioginę nutolusią duomenų bazę. Jei žingsnis yra tenkinamas, einama prie 4 žingsnio, kuris sumažina pasirinkimą tarp perdavimo metodų.

6) žingsnio metu nustatomas, ar reikalingas nuolatinis duomenų apskaitimas su nutolusia duomenų baze. Kai kurie duomenų apskaitimo metodai naudoja nuolatinį interneto ryšį, kad būtų pasiekiamas rezultatas. Jei duomenų apskaitimui nereikalingas nuolatinis interneto ryšys, tai pasirenkamas nutolusios duomenų prieigos naudojimas, priešingu atveju keliami sąlyga, ar bus naudojamas replikavimas, esant pastoviam interneto ryšiui.

7) žingsnio metu nustatoma, ar yra galimybė nutolusioje duomenų bazėje naudoti replikavimo paslaugą. Ši paslauga reikalauja papildomų replikavimui būdingų nustatymų nutolusioje duomenų bazėje, jie palengvina duomenų apskaitimo tarp DBVS ir mobilaus įrenginio metodą:

- Sukuriama platinama duomenų bazė, iš kurios duomenys bus viešinami;
- Pasirenkamas replikavimo metodas ir DBVS, su kuria bus atliekamas duomenų perdavimas;
- Nusifiltruojami duomenys, kurie bus perduodami į naują duomenų bazę;
- Sukuriamas momentinio vaizdo agentas, kuriuo remiantis bus atliekamas duomenų replikavimas;
- Sukuriama publikacija, kurią vykdys suliejimo replikavimas;
- Sukuriami vartotojo prisijungimo duomenys prie duomenų bazės;
- Kuriami prenumerata;
- Pasirenkama sukurta publikacija;
- Pasirenkama suliejimo agento saugojimo vieta (Platintojo ar Prenumeratoriaus pusėje bus);
- Sukuriama duomenų bazė, kuria bus perduodami duomenys;
- Sukuriami vartotojo prisijungimo duomenys prie duomenų bazės;
- Kiekvieno suliejimo agento proceso ir sujungimo nustatymas;
- Nustatomi prenumeratos tipai (serveris, klientas) ir konflikto sprendimo prioritetai;
- Sukuriama prenumerata, kuri vykdys suliejimo replikavimas;

Tie nustatymai reikalingi tam, kad vyktų duomenų perdavimas tarp mobilios ir nutolusios duomenų bazės. Jei sąlyga netenkinama, reikia žiūrėti sistemos reikalavimų, priešingu atveju yra pasirenkamas suliejimo replikavimas.

8) žingsnio metu nustatoma, kad reikalingas mobilus įrenginys arba architektūros keitimas. Kadangi yra netenkinamos sąlygos, tai reikia ieškoti kitų sprendimų, vykdant duomenų apsikeitimo tarp DBVS ir mobilus įrenginio pasirinkimo procesą.

9) paslauga naudojama, kai duomenys apdorojami mobiliajame įrenginyje ir nereikalingas nuolatinis interneto ryšys.

10) žingsnio metu nustatoma, kad reikia žiūrėti sistemos reikalavimų. Galimas atvejis, kad bus naudojami kiti reikalavimai ir kitas duomenų apsikeitimo metodas.

11) replikavimas, naudoja automatišką duomenų apsikeitimą, bet prieš tai reikia atlikti papildomus nustatymus pačioje nutolusioje duomenų bazėje.

12) metodas, kuris tiesiogiai duomenis apdoroja nutolusioje duomenų bazėje.

4. Duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodų eksperimentinis tyrimas

Duomenų perdavimo tarp DBVS ir mobilaus įrenginio pasirinkimo proceso pagalba kūrėjas gali lengviau pasirinkti duomenų perdavimo metodą, kuris skirtas duomenų apdorojimui, atsižvelgdamas į tam tikrus apribojimus. Kūrėjas, pasirinkęs apribojimus, gali sukurti taikomąją programą, kuri skirta su nutolusia duomenų prieigos paslauga arba duomenis apdoroti tiesioginėje nutolusioje duomenų bazėje. Kitas variantas yra suliejimo replikavimas, kuris reikalauja daugiau nustatymų DBVS. Šis replikavimo metodas pagal našumą nenusileidžia nutolusiai duomenų prieigos paslaugai, bet duomenų apsikeitimas vyksta nuolatos. Yra sunku atlikti eksperimentą, kuris nustatytų laiko našumą, naudojant skirtingus duomenų kiekius. Taip pat šitam replikavimo metodui reikalingas nuolatinis interneto ryšys, kuris užtikrintų sklandų ir tikslų eksperimento tyrimą. Atliekant eksperimentą buvo siekiama iširti duomenų apsikeitimo galimybes tarp mobilaus įrenginio ir duomenų bazės. Tam buvo pasinaudota dviem būdais. Pirma, kai įrenginys naudoja savąją duomenų bazę, skirtą apdoroti duomenims. Antras variantas, kai iš mobiliojo įrenginio duomenys siunčiami į nutolusią duomenų bazę. Įgyvendinant abu variantus reikalinga taikomoji programa, kuri galėtų palyginti nustatytus abiejų metodų našumus.

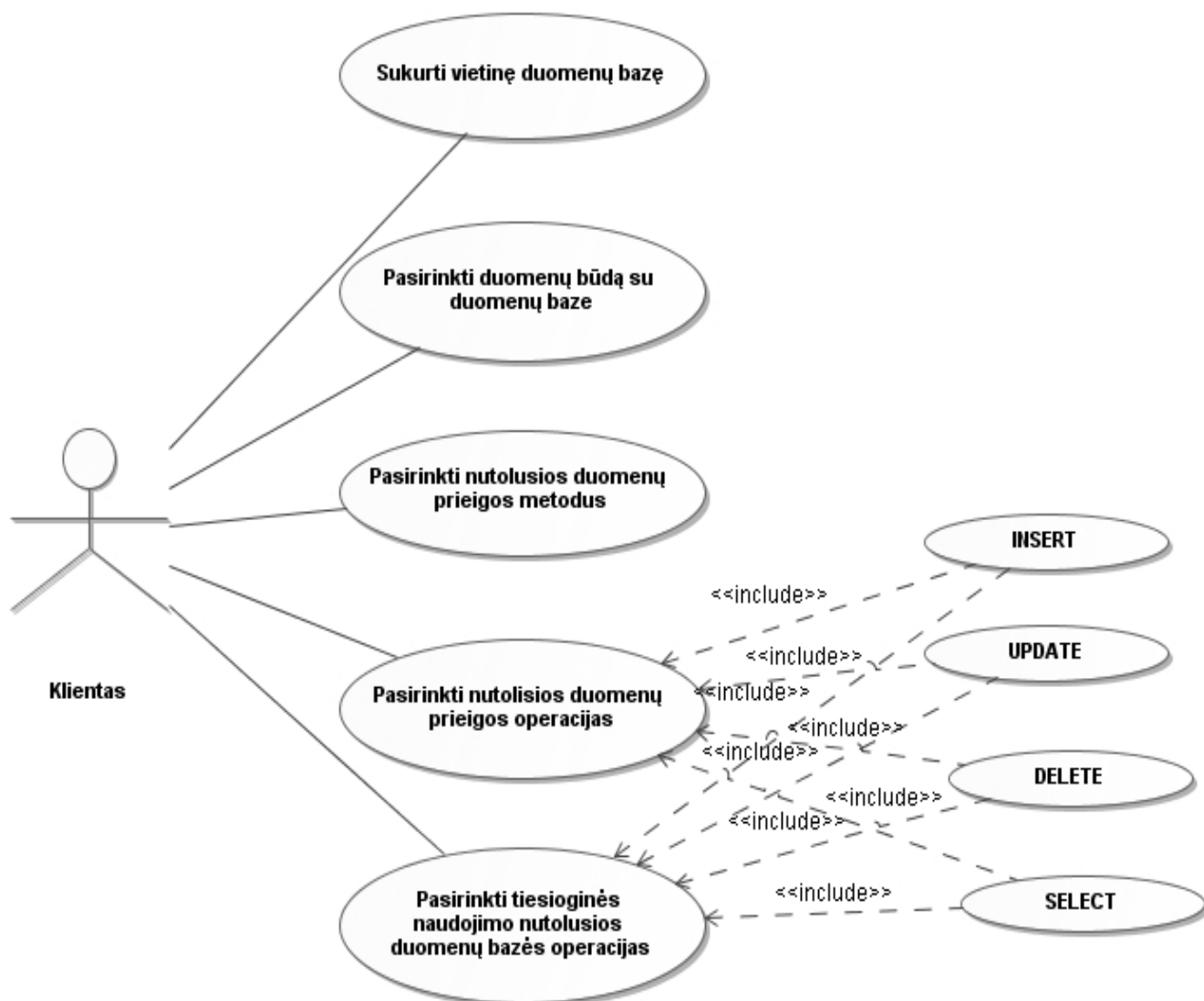
4.1 Eksperimentiniam tyrimui sukurtos programinės įrangos ir naudojamų duomenų specifikacija

Taikomoji programa buvo sukurta pasinaudojant MS Visual Studio 2008 paketu, Microsoft .NET Compact Framework 3.5 biblioteka ir C sharp programavimo kalba. Microsoft .NET Compact Framework 3.5 yra integruotas komponentas Windows Mobile ir Windows Embedded CE įrenginiuose, kuriuo įdiegiamos ir valdomos taikomosios programos, naudojančios Web paslaugą. Tai pagrindinė dalis Microsoft tikslams realizuoti, jis pasirūpina vartotoju bet kurio metu ir bet kurioje vietoje. Jis valdo C sharp kodą, kuris skirtas apsaugoti plėtojimą ir vykdymą, naudoti taikomąją programą mobiliuose įrenginiuose. Jos kuriamos panaudos atvejis pavaizduotas 15 paveiksle, kad vartotojas galėtų sužinoti, kokias komandas jis galės naudoti ir vykdyti

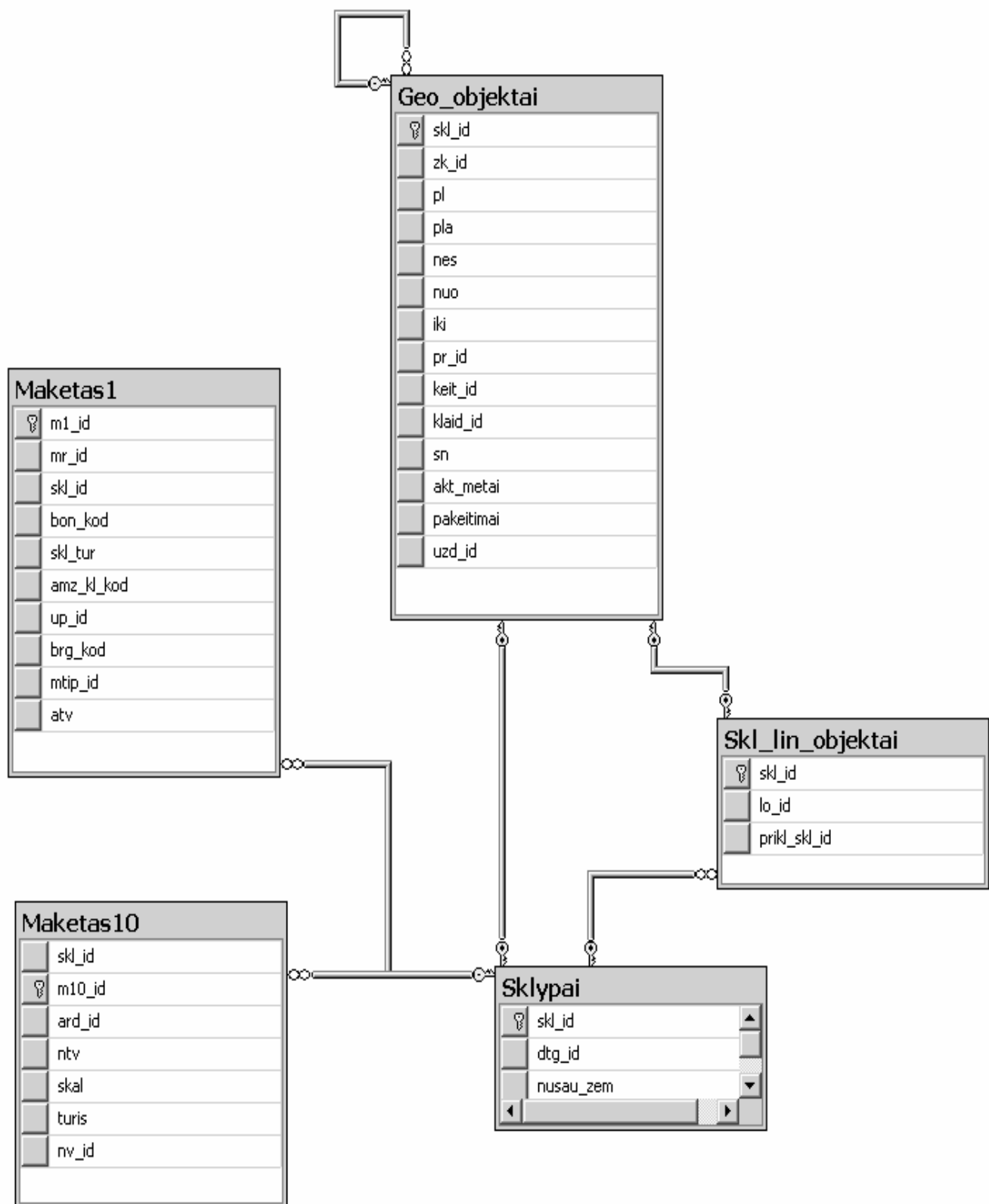
Atliekant eksperimentą buvo pasinaudota Miškų kadastro integruotos informacinės sistemos duomenų baze. Duomenų bazė užima apie 4 GB diskinės atminties. Tyrimui buvo panaudotas 5 lentelių epizodas (geo_objektai, sklypai, skl_lin_objektai, maketas1 ir maketas10) iš 243 lentelių. Šios 5 lentelės pasirinktos dėl didžiausio saugomų duomenų kiekio. Detalesnė informacija apie pasirinktas lenteles ir jose saugomą duomenų kiekį pateikiama 9 lentelėje. Duomenų bazės struktūros schema pateikta paveikslėlyje 16.

9 lentelė. Eksperimentui panaudoto duomenų bazės epizodo lentelių informacija

Geo_objektai	Sklypai	Skil_objektai	Maketas10	Maketas1
int	int	Int	int	int
int	int	Int	int	int
decimal(12, 6)	varchar(1)	Int	int	int
decimal(6, 2),			decimal(3, 0)	varchar(2)
decimal(12, 6)			decimal(3, 2)	decimal(2, 0)
datetime			decimal(7, 2),	int
datetime			int	varchar(1)
int				int
int				varchar(1)
int				
varchar(1)				
decimal(4, 0),				
bigint				
int				
Viso įrašų sudaro: 1663031	Viso įrašų sudaro: 1466565	Viso įrašų sudaro: 196466	Viso įrašų sudaro: 1858894	Viso įrašų sudaro: 1323561



15 pav. Kuriamos programinės įrangos panaudos atvejais



16 pav. Miškų kadastro integruotos informacinės sistemos duomenų bazės epizodas

Sekų diagrama pavaizduoja, kaip sukurtoje programinėje įrangoje esantys objektai komunikuoja ir kokia eilės tvarka yra vykdomos visos operacijos. 17 paveiksle pateikta darbų sekų diagrama parodo visą sistemos komunikavimo eigą su nutolusių duomenų prieiga. Šią seką sudaro keturi objektai: klientas, sistema, mobili duomenų bazė, nutolusi duomenų bazė. Kiekvienas objektas atlieka konkrečias funkcijas:

1. Klientas

- Įvesti duomenų bazės pavadinimą(1);
- Pasirinkti Pull metodą(4);
- Pasirinkti operaciją(10);
- Pasirinkti Push metodą(14);

2. Sistema

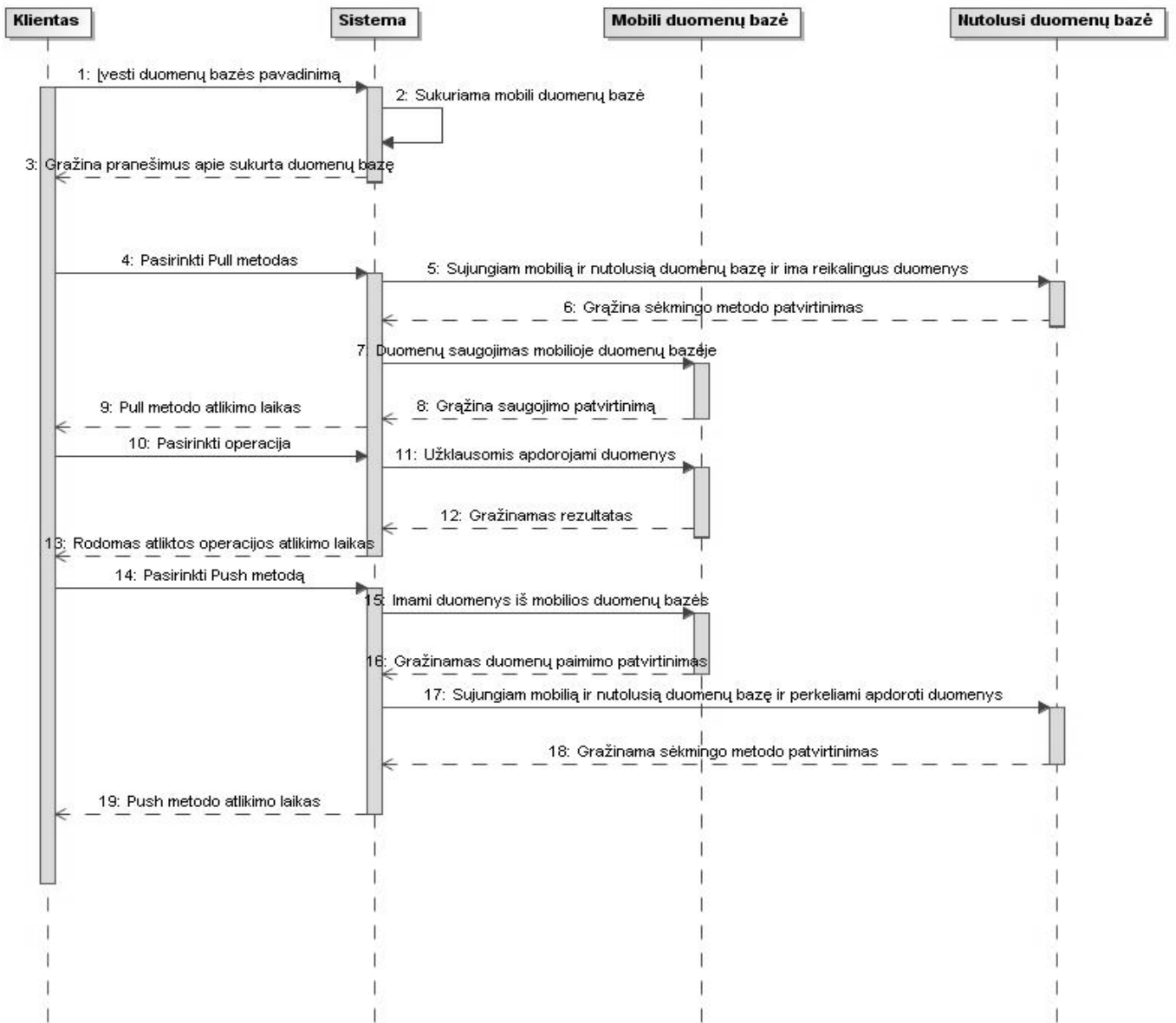
- Sukuriama mobili duomenų bazė(2);
- Gražina pranešimus apie sukurtą duomenų bazę(3);
- Per nutolusių duomenų prieigos paslaugą jungiamasi prie nutolusios duomenų bazės ir imami reikalingi duomenys(5);
- Duomenų saugojimas mobiloje duomenų bazėje(7);
- Pull metodo atlikimo laikas(9);
- Užklausomis apdorojami duomenys(11);
- Rodomas atliktos operacijos atlikimo laikas(13);
- Imami duomenys iš mobilios duomenų bazės(15);
- Sujungiama mobili ir nutolusi duomenų bazės ir perduodami apdoroti duomenys(17);
- Push metodo atlikimo laikas(19);

3. Mobilus duomenų bazė

- Gražina saugojimo patvirtinimą(8);
- Gražina užklausomis apdorotą rezultatą(12);
- Gražina paimtų duomenų patvirtinimą(16);

4. Nutolusi duomenų bazė

- Gražina sėkmingo Pull metodo patvirtinimą(6);
- Gražina sėkmingo Push metodo patvirtinimą(18).

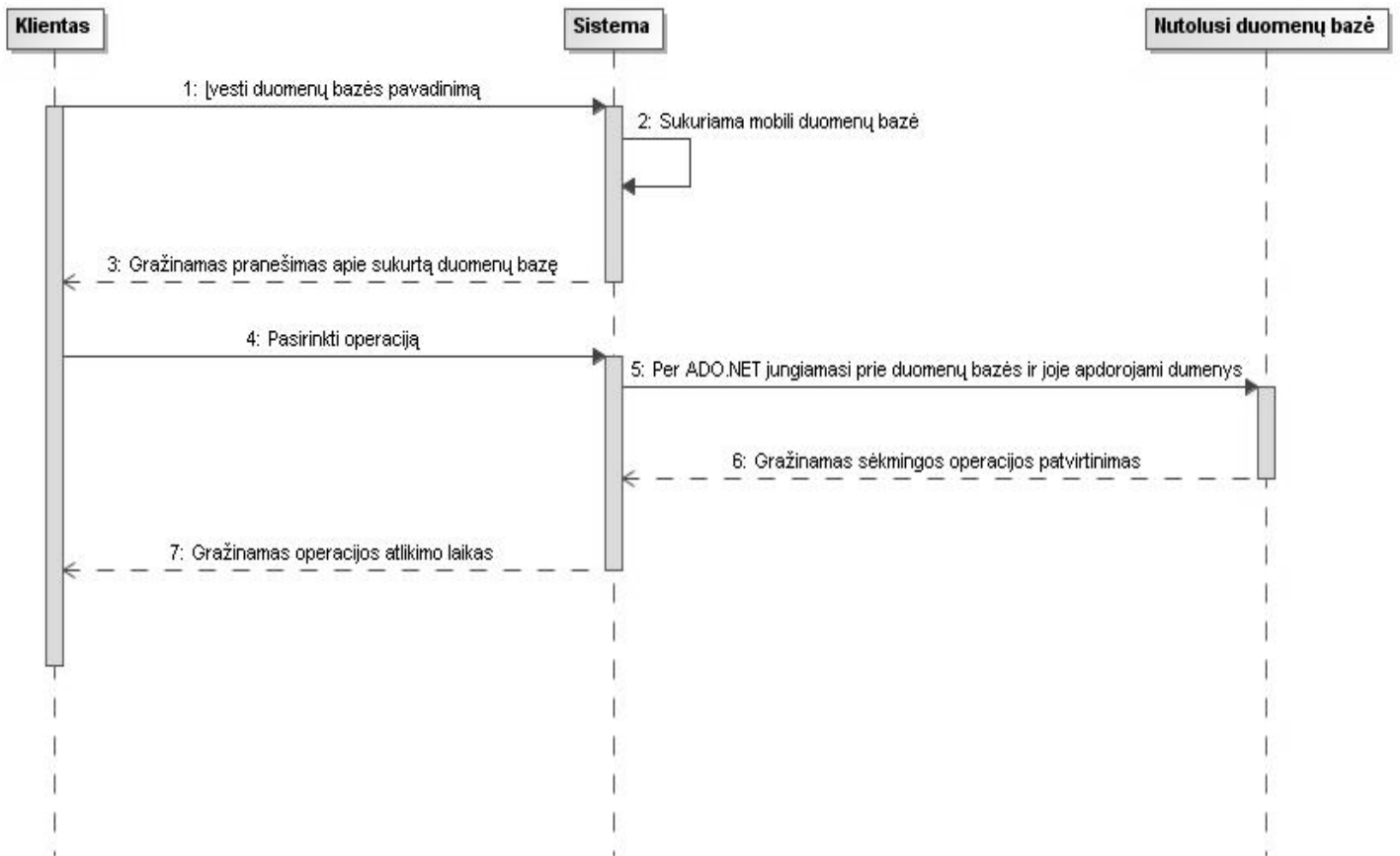


17 pav. Sekų diagrama nutolusių duomenų prieigos paslaugai

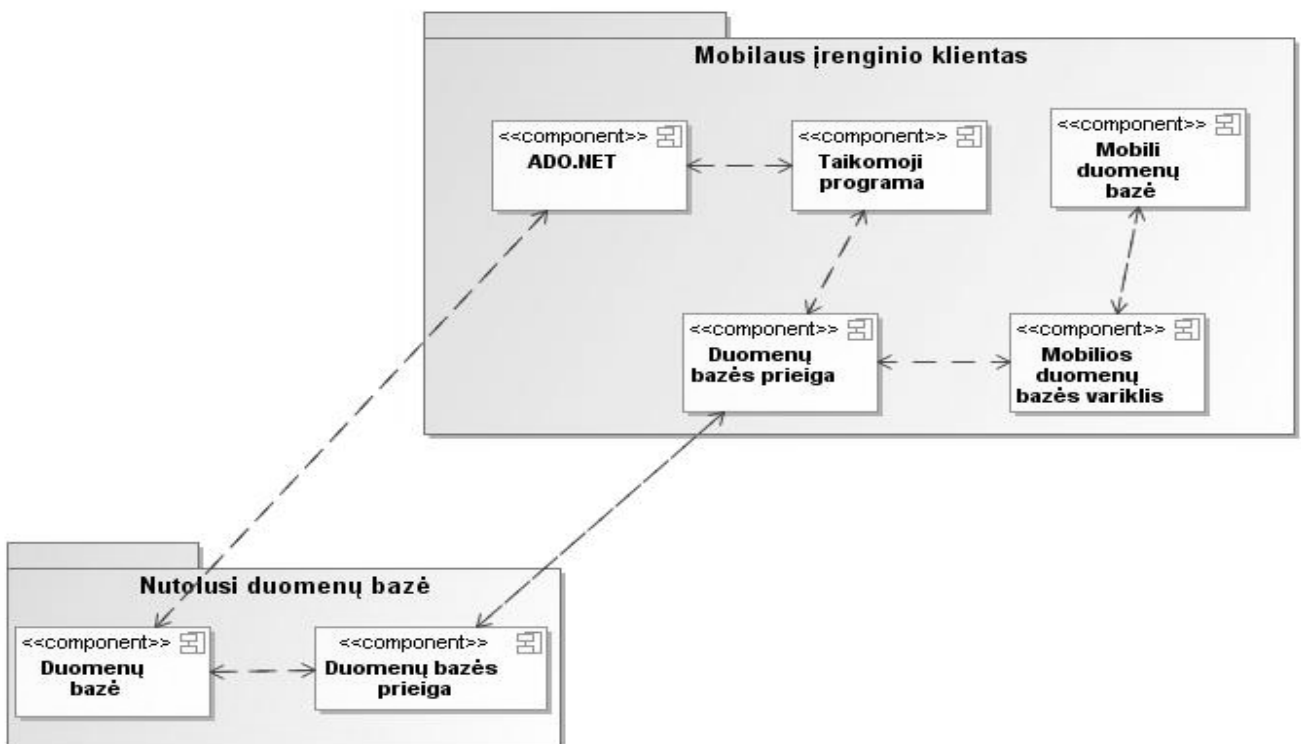
Kitoje sekų diagramoje pateikiama tiesioginės nutolusios duomenų bazės naudojimo atvejis, ji pavaizduota 18 paveiksle. Ją sudaro trys objektai: klientas, sistema ir nutolusi duomenų bazė. Kiekvienas objektas atlieka konkrečias funkcijas:

1. Klientas
 - Įvesti duomenų bazės pavadinimą(1);
 - Pasirinkti operaciją(4);
2. Sistema
 - Sukuriama mobili duomenų bazė(2);
 - Gražina pranešimus apie sukurtą duomenų bazę(3);
 - Gražinamas atliktos operacijos laikas(7);
3. Nutolusi duomenų bazė

- Per ADO.NET jungiamasi prie nutolusios duomenų bazės ir joje apdorojami duomenys(5);
- Gražinamas sėkmingos operacijos patvirtinimas(6);

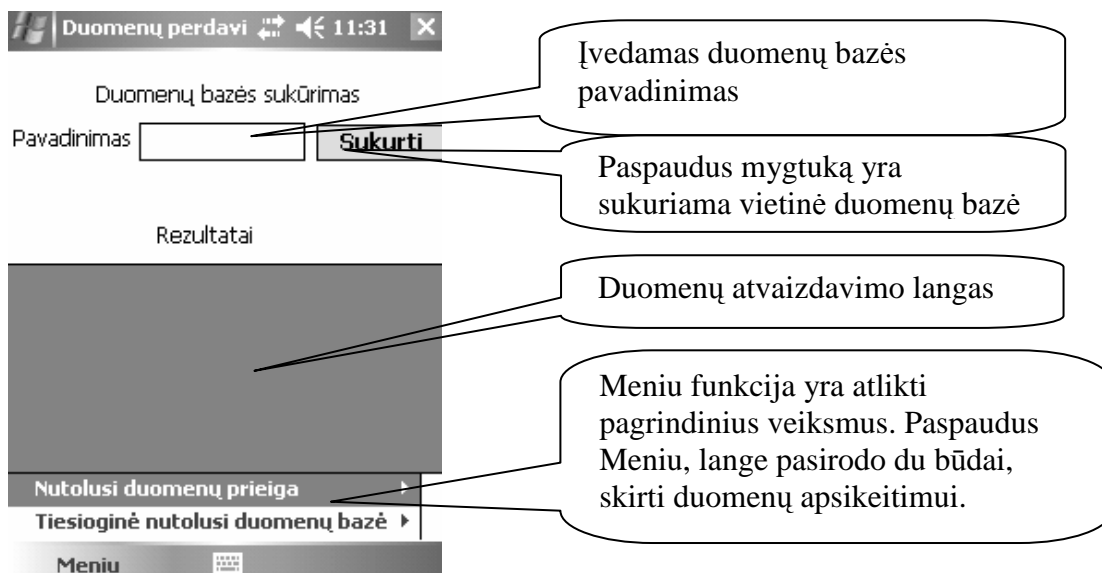


18 pav. Sekų diagrama tiesioginės nutolusios duomenų bazės naudojimui



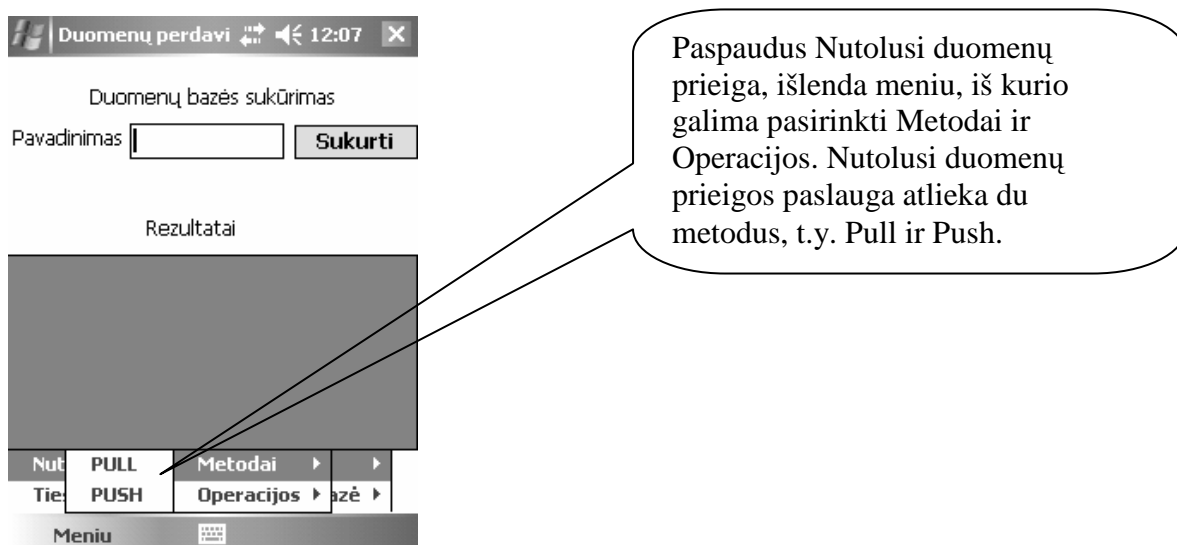
19 pav. Bendra nutolusių duomenų prieigos ir tiesioginis nutolusios duomenų bazės architektūra

Paveiksle pateikiama bendra architektūros schema, pagal kurią buvo atliekamas eksperimentas. Joje matomi pagrindiniai architektūros skirtumai ir panašumai tarp komponentų. Eksperimentas atliekamas dviem būdais naudojant nutolusių duomenų prieigos paslaugą, naudojant mobilią duomenų bazę ir tiesioginis nutolusios duomenų bazės naudojimas, kai iš kart yra naudojama nutolusi duomenų bazė duomenims apdoroti.



20 pav. Pagrindinis sukurtos taikomosios programos langas

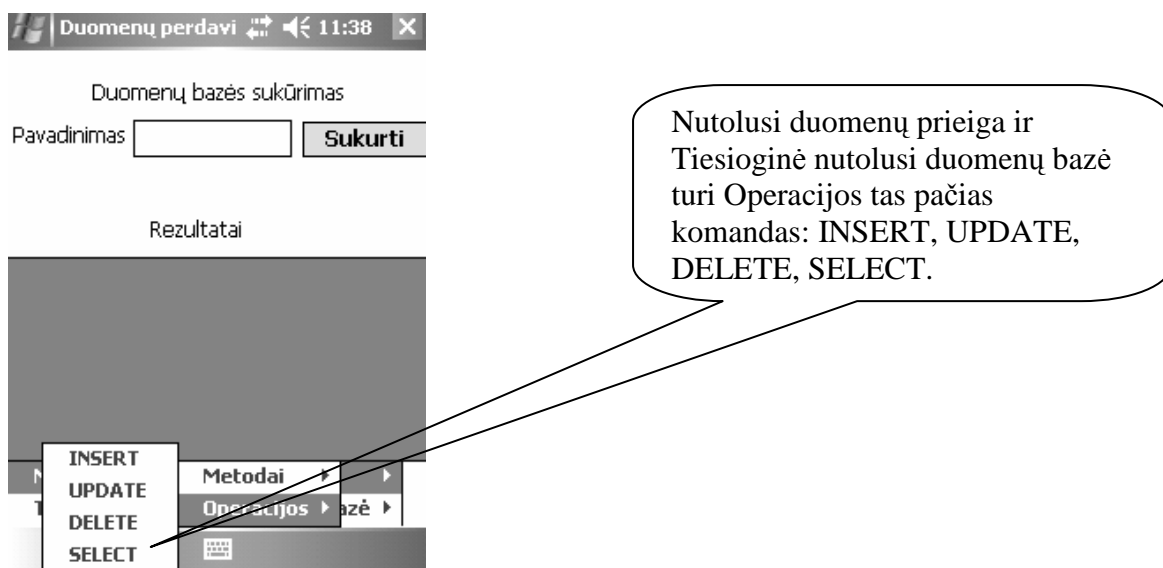
20 paveiksle pavaizduotas pagrindinis taikomosios programos langas, skirtas sukurti mobilios duomenų bazės failą su plėtiniu .sdf, kuriame saugomi visi duomenų bazės duomenys. Taip pat skirtas pasirinkti nutolusių duomenų prieigos paslaugos arba Tiesioginės duomenų bazės naudojimo perdavimo metodą.



21 pav. Pagrindiniai nutolusių duomenų prieigos paslaugos metodai

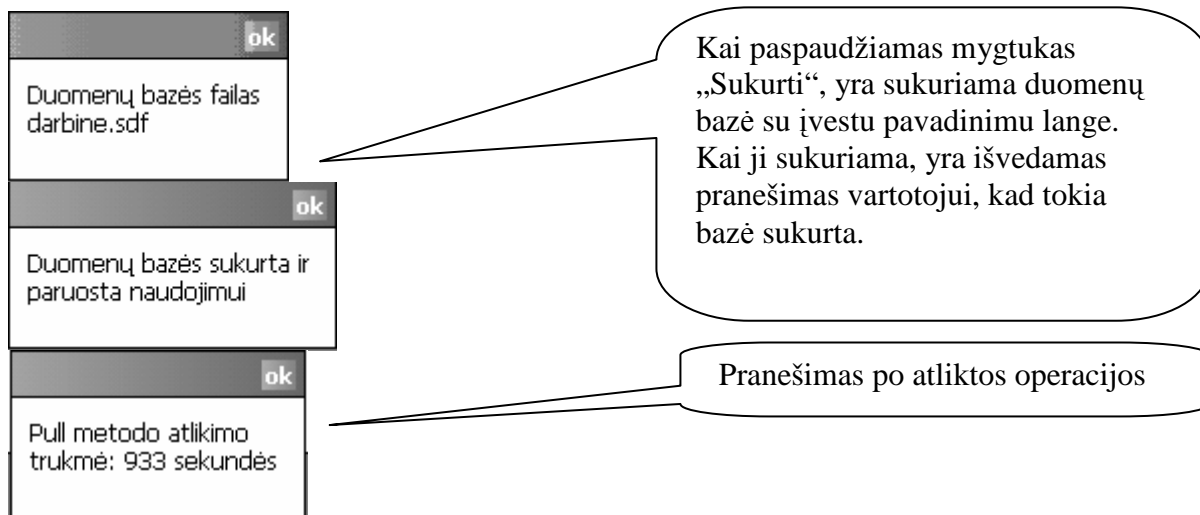
21 paveikslas vaizduoja paslaugos naudojamus du pagrindinius metodus: traukimas ir

stūmimas. Pirmasis skirtas duomenis pertraukti į mobilią duomenų bazę. Antrasis – duomenis stumti į nutolusią duomenų bazę.



22 pav. Pagrindinės operacijos su duomenimis, naudojant nutolusią duomenų prieigą ir tiesioginę nutolusią duomenų bazę

22 paveiksle vaizduojamos taikomosios programos, naudojančios nutolusios duomenų prieigos paslaugas ir Tiesioginio naudojimo duomenų bases, operacijos. Eksperimentas buvo atliekamas su INSERT, UPDATE, DELETE ir SELECT operacijomis, su kuriomis duomenys apdorojami mobilojoje duomenų bazėje ir tiesioginėje nutolusioje duomenų bazėje.



23 pav. Pranešimai

23 paveiksle vaizduojami pranešimai, gaunami po kiekvieno atlikto veiksmo su metodais ar operacijomis.

11 lentelė. Taikomosios programos kodas, kuriuo naudojami pagrindiniai nutolusių duomenų prieigos metodai

Pull metodo naudojimas
<pre>rda.Pull("geo_objektai", "SELECT * FROM geo_objektai where (skl_id > 5452959)", "Provider=SQLOLEDB.1;Password=magas87;Persist Security Info=true;User ID=zilvinas;Initial Catalog=zilvinui;Data Source=193.219.66.87", RdaTrackOption.TrackingOnWithIndexes, "klaidu_lentele");</pre>
<pre>rda.Pull("Sklypai", "SELECT skl_id, dtg_id, nusau_zem FROM Sklypai where (skl_id > 5452959)", "Provider=SQLOLEDB.1;Password=magas87;Persist Security Info=true;User ID=zilvinas;Initial Catalog=zilvinui;Data Source=193.219.66.87", RdaTrackOption.TrackingOnWithIndexes, "klaidu_lentele1");</pre>
<pre>rda.Pull("skl_lin_objektai", "SELECT * FROM skl_lin_objektai where (skl_id > 5452959)", "Provider=SQLOLEDB.1;Password=magas87;Persist Security Info=true;User ID=zilvinas;Initial Catalog=zilvinui;Data Source=193.219.66.87", RdaTrackOption.TrackingOnWithIndexes, "klaidu_lentele2");</pre>
<pre>rda.Pull("maketas10", "SELECT * FROM maketas10 where (m10_id > 5633015)", "Provider=SQLOLEDB.1;Password=magas87;Persist Security Info=true;User ID=zilvinas;Initial Catalog=zilvinui;Data Source=193.219.66.87", RdaTrackOption.TrackingOnWithIndexes, "klaidu_lentele3");</pre>
<pre>rda.Pull("maketas1", "SELECT * FROM maketas1 where (m1_id > 4639795)", "Provider=SQLOLEDB.1;Password=magas87;Persist Security Info=true;User ID=zilvinas;Initial Catalog=zilvinui;Data Source=193.219.66.87", RdaTrackOption.TrackingOnWithIndexes, "klaidu_lentele4");</pre>
Push metodo naudojimas
<pre>rda.Push("maketas1", rdaOleDbConnectionString, RdaBatchOption.BatchingOn); rda.Push("maketas10", rdaOleDbConnectionString, RdaBatchOption.BatchingOn); rda.Push("geo_objektai", rdaOleDbConnectionString, RdaBatchOption.BatchingOn); rda.Push("Sklypai", rdaOleDbConnectionString, RdaBatchOption.BatchingOn); rda.Push("skl_lin_objektai", rdaOleDbConnectionString, RdaBatchOption.BatchingOn);</pre>

12 lentelė. Pull metodo aprašas

Pull metodo atributas	Apibrėžimas
Lentelė	Duomenų bazės lentelė, į kurią bus traukiami duomenys iš nutolusios duomenų bazės.
SELECT	SELECT sakiniai arba procedūros, su kuriomis atrenkami duomenys į vietinę duomenų bazę.
OLEDB prisijungimas	OLEDB prisijungimas prie nutolusios duomenų bazės
RDA_TRACKOPTION	Opcija, kuri stebi, ar bus atliekamas duomenų stūmimas į nutolusią duomenų bazę.
Klaidų lentelė	Lentelėje yra saugomos klaidos.

13 lentelė. Push metodo aprašas

Push metodo atributas	Apibrėžimas
Lentelė	Duomenų bazės lentelė, kuri bus stumiamą į nutolusią duomenų bazę. Mobilios ir nutolusios duomenų bazės lentelės pavadinimai turi sutapti.
OLEDB prisijungimas	OLEDB prisijungimas prie nutolusios duomenų bazės
RDA_BATCHOPTION	Opcija, kuri sujungia visas lentelės eilutes ir stumia per vien transakciją į nutolusią duomenų bazę
Klaidų lentelė	Lentelėje yra saugomos klaidos.

14 lentelė. Taikomosios programos kodas, kuriuo naudojamos operacijos duomenims apdoroti

INSERT operacija
<pre>while (i < 100000) { comanda.CommandText = "insert into geo_objektai (zk_id, nuo, iki, pr_id, keit_id, sn, pakeitimai, uzd_id) values (" + zk_id_tarpinis + ", '2007.01.01 00:00:00', '2008.01.01 00:00:00', " + pr_id_tarpinis + ", " + keit_id_tarpinis + ", '" + sn_tarpinis + "', " + pakeitimai_tarpinis + "," + uzd_id_tarpinis + ")"; comanda.ExecuteNonQuery(); comanda.CommandText = "select max(skl_id) from geo_objektai"; ind = comanda.ExecuteScalar().ToString(); comanda.CommandText = "insert into sklypai (skl_id, dtg_id, nusau_zem) values ('" + ind + "', " + dtg_id_tarpinis + ", '" + nusau_zeme_tarpinis + "')"; comanda.ExecuteNonQuery(); comanda.CommandText = "insert into skl_lin_objektai (skl_id, lo_id, prikl_skl_id) values ('" + ind + "', " + lo_id_tarpinis + ", '" + ind + "')"; comanda.ExecuteNonQuery(); comanda.CommandText = "insert into Maketas10 (skl_id, ard_id, nv_id) values ('" + ind + "', " + ard_id_tarpinis + ", " + nv_id_tarpinis + ")"; comanda.ExecuteNonQuery(); comanda.CommandText = "insert into Maketas1 (mr_id, skl_id, up_id, mtip_id, atv) values (36, '" + ind + "', 4, 35, 'd')"; comanda.ExecuteNonQuery(); pakeitimai_tarpinis = pakeitimai_tarpinis + 500; i = i + 1; }</pre>
UPDATE operacija
<pre>comanda.CommandText = "update maketas1 set mr_id = 35, up_id = 4, mtip_id = 35 where m1_id >" + id1; comanda.ExecuteNonQuery(); comanda.CommandText = "update maketas10 set ard_id = 5, nv_id = 3 where m10_id >" + id2; comanda.ExecuteNonQuery(); comanda.CommandText = "update sklypai set nusau_zem = 'y', dtg_id = 61 where skl_id >" + id3; comanda.ExecuteNonQuery(); comanda.CommandText = "update skl_lin_objektai set lo_id = 307990 where skl_id</pre>

```

>" + id3;
comanda.ExecuteNonQuery();
comanda.CommandText = "update geo_objektai set sn = 'm', pakeitimai = 280045
where skl_id >" + id3;
comanda.ExecuteNonQuery();

```

DELETE operacija

```

comanda.CommandText = "delete from maketas1 where m1_id >" + id1;
comanda.ExecuteNonQuery();
comanda.CommandText = "delete from maketas10 where m10_id >" + id2;
comanda.ExecuteNonQuery();
comanda.CommandText = "delete from skl_lin_objektai where skl_id >" + id3;
comanda.ExecuteNonQuery();
comanda.CommandText = "delete from sklypai where skl_id >" + id3;
comanda.ExecuteNonQuery();
comanda.CommandText = "delete from geo_objektai where skl_id >" + id3;
comanda.ExecuteNonQuery();

```

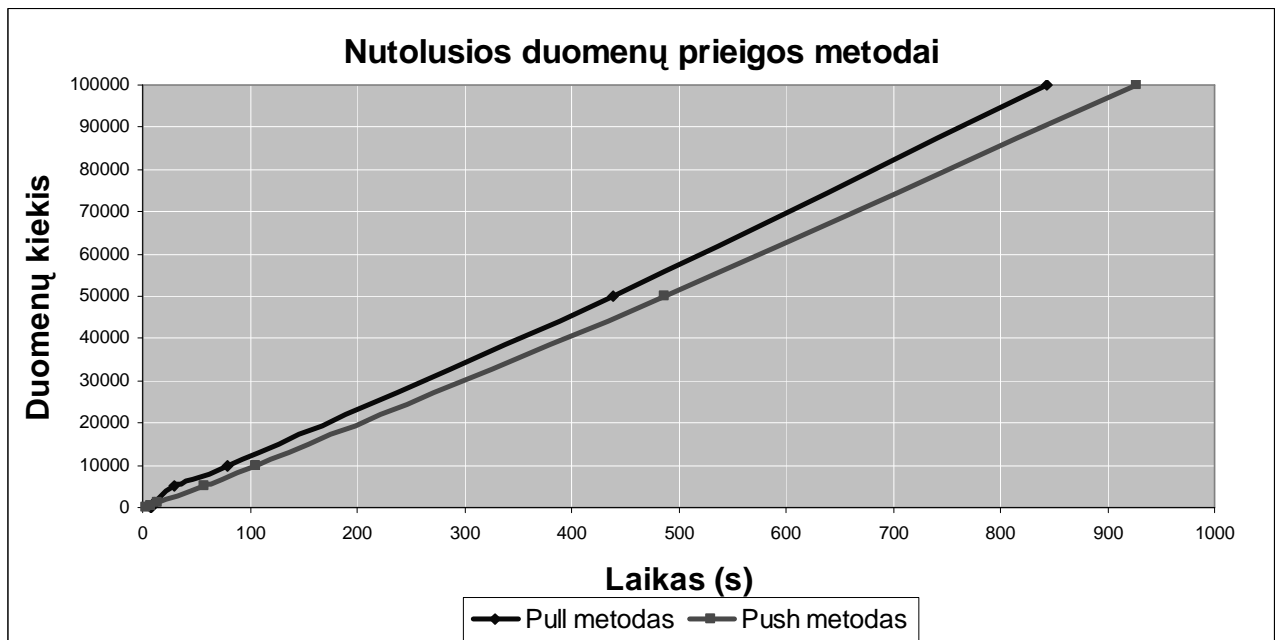
SELECT operacija

```

command.CommandText = "select * from maketas1 where m1_id >" + id1;
command.ExecuteNonQuery();
command.CommandText = "select * from maketas10 where m10_id >" + id2;
command.ExecuteNonQuery();
command.CommandText = "select * from skl_lin_objektai where skl_id >" + id3;
command.ExecuteNonQuery();
command.CommandText = "select * from sklypai where skl_id >" + id3;
command.ExecuteNonQuery();
command.CommandText = "select * from geo_objektai where skl_id >" + id3;
command.ExecuteNonQuery();

```

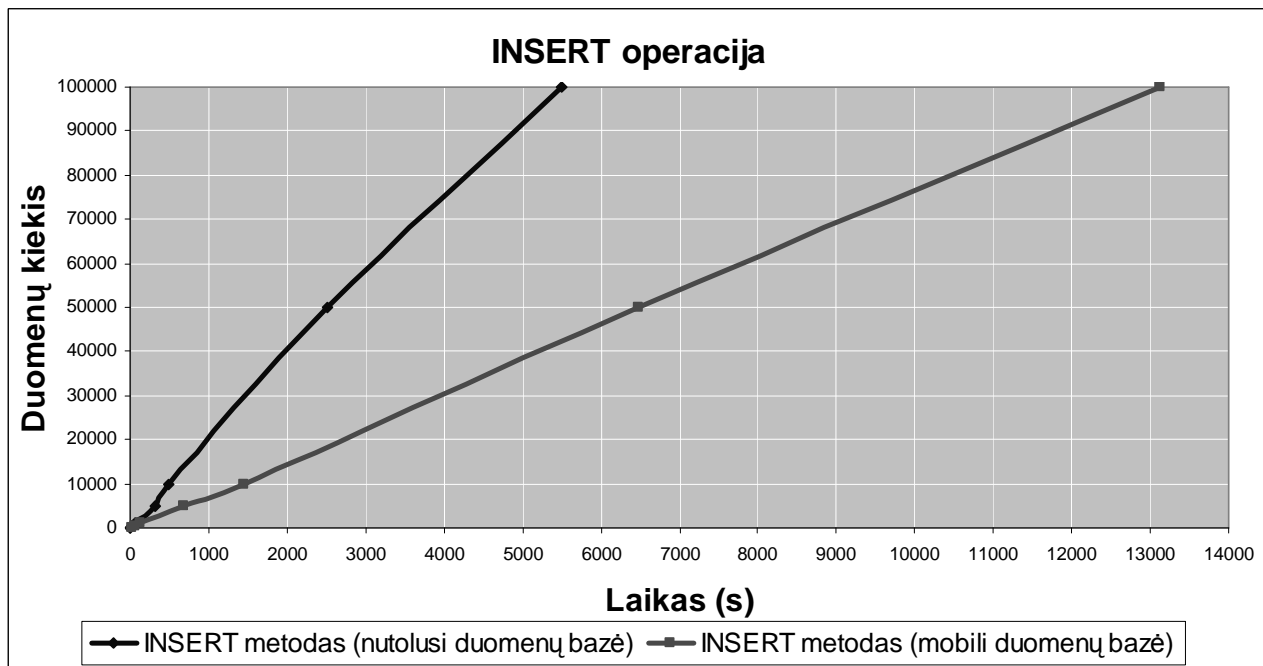
4.2 Duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodų eksperimentinio tyrimo rezultatai



24 pav. Nutolusių duomenų prieigos paslaugos metodų palyginimas

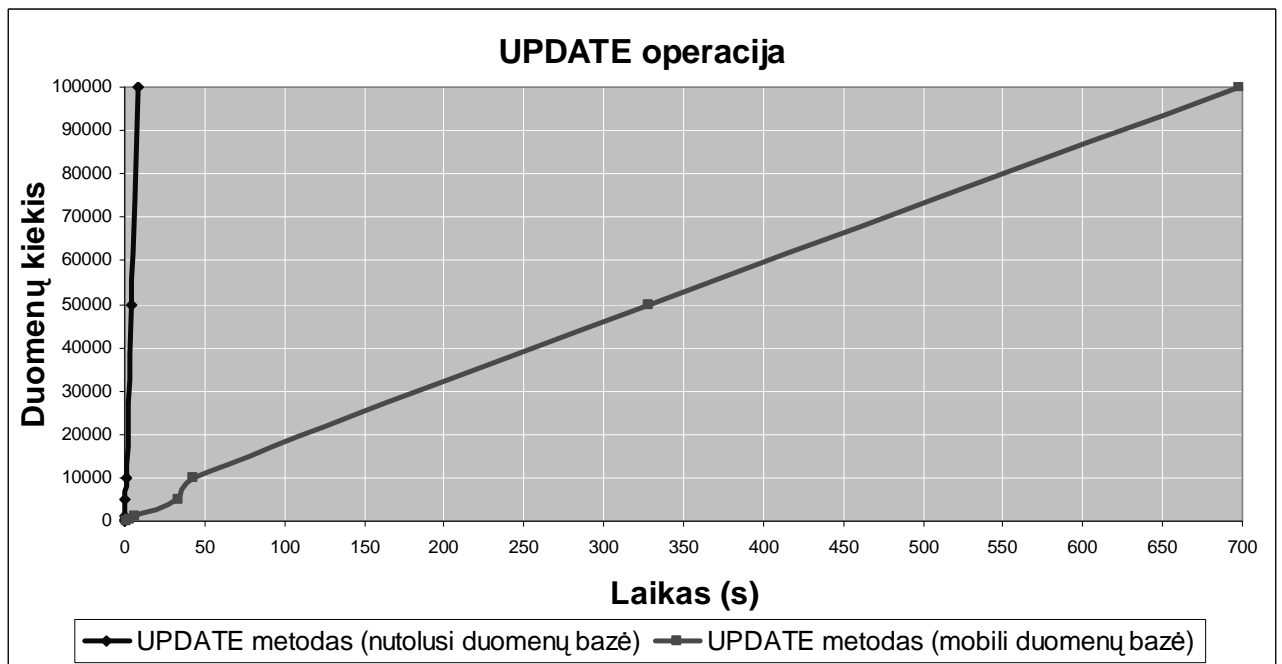
Pritaikant taikomąją programą buvo eksperimentuojama traukimo ir stūmimo metodais. Naudojant 5 lentelės ir nuo 100 iki 100 000 lentelės įrašus, bandymas buvo kartojamas po 10 kartų ir išvedamas aritmetinis vidurkis. Šių dviejų metodų palyginimas pavaizduotas 24 paveiksle, iš

kurio galima daryti prielaidą, kad turėsime tiesinę laiko priklausomybę nuo duomenų kiekio. Iš grafiko matosi, kad abiejų nutolusių duomenų prieigos paslaugos metodų sparta panaši. Jų priklausomybė priklauso nuo apdorojamų duomenų kiekių. Kuo duomenų kiekis didesnis, tuo atlikimo laikas yra ilgesnis.



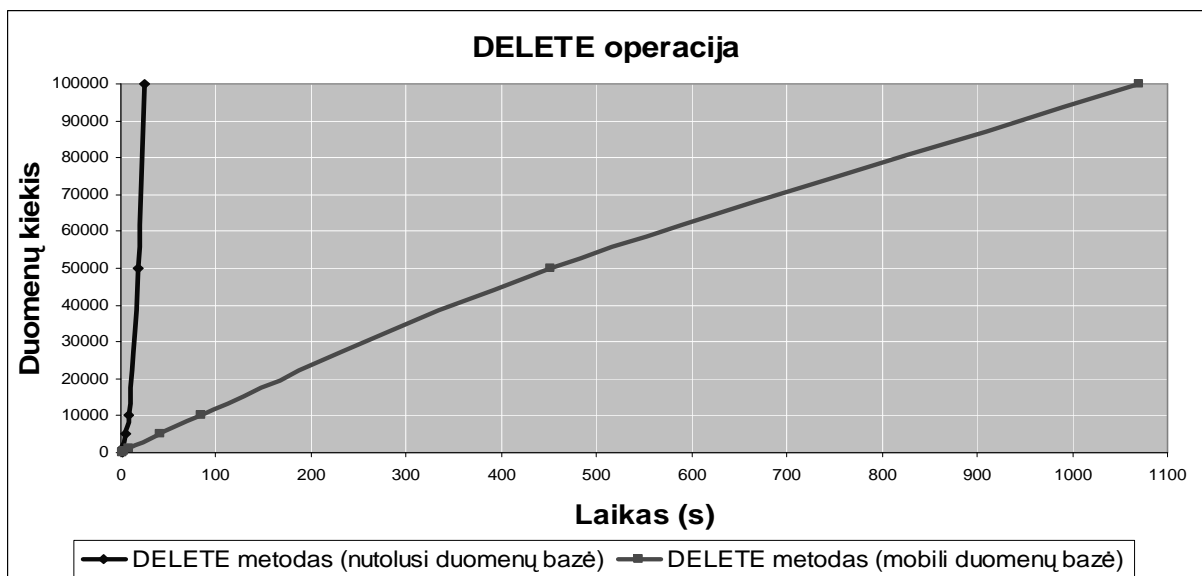
25 pav. Nutolusių duomenų prieigos paslaugos ir tiesioginės nutolusios duomenų bazės naudojimo INSERT operacijos palyginimas

Pritaikant taikomąją programą buvo eksperimentuojamos nutolusių duomenų prieigos ir tiesioginės nutolusios duomenų bazės naudojimo INSERT operacijos. Naudojant 5 lentelės ir nuo 100 iki 100 000 lentelės įrašų, bandymas buvo kartojamas po 10 kartų ir išvedamas aritmetinis vidurkis. Šios operacijos palyginimas pavaizduotas 25 paveiksle. Iš grafiko galima matyti, kad naudojant tiesioginę nutolusią duomenų bazę operacija atliekama greičiau. Tai lėmė interneto pralaidumas. Naudojant lėtesnio interneto ryšį, grafiko duomenys skirsis. Galimas atvejis, kad geriau naudoti nutolusią duomenų prieigą. Lyginat su kitomis atlikimo operacijomis, INSERT operacijos atlikimo laikas didelis.



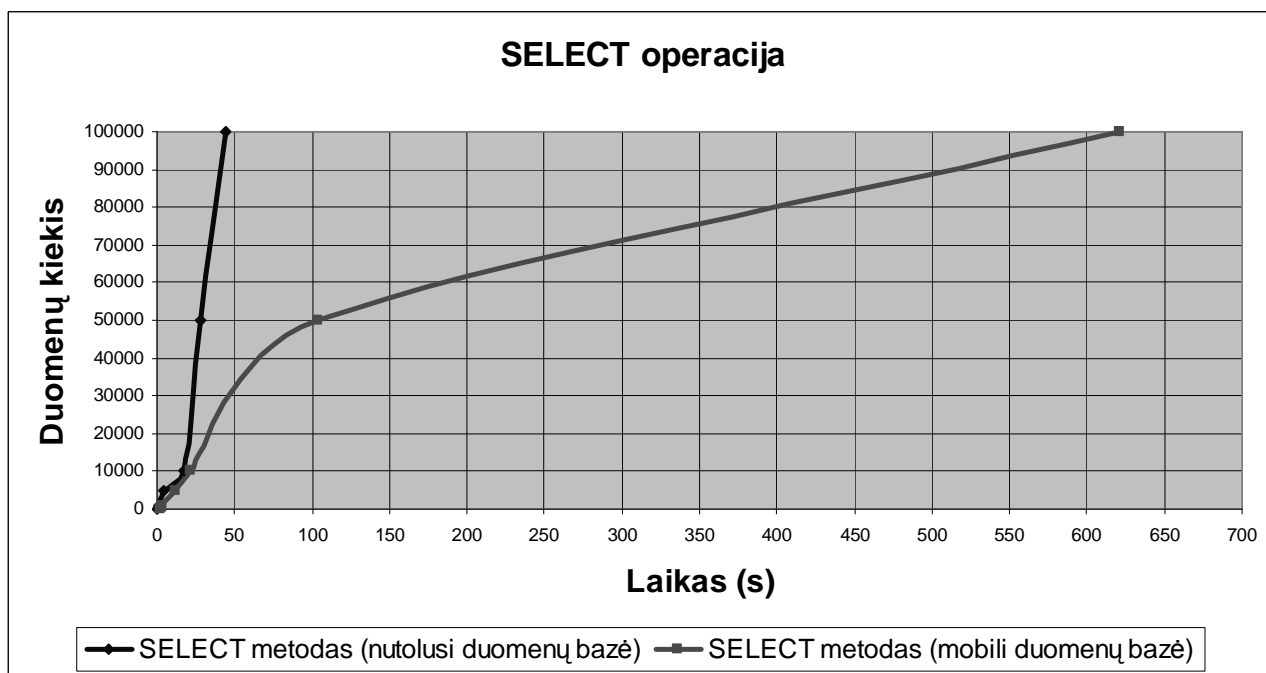
26 pav. Nutolusių duomenų prieigos paslaugos ir tiesioginės nutolusios duomenų bazės naudojimo UPDATE operacijos palyginimas

Pritaikant taikomąją programą buvo eksperimentuojamas nutolusių duomenų prieigos ir tiesioginės nutolusios duomenų bazės naudojimo UPDATE operacijos. Naudojant 5 lentelės ir nuo 100 iki 100 000 lentelės įrašų, bandymas buvo kartojamas po 10 kartų ir išvedamas aritmetinis vidurkis. Šios operacijos palyginimas pavaizduotas 26 paveiksle. Iš grafiko galima matyti, kad naudojant tiesioginę nutolusią duomenų bazę operacija atliekama greičiau. Tai lėmė interneto pralaidumas ir apdorojamų duomenų kiekis vienos transakcijos metu. Naudojant lėtesnio interneto ryšį, grafiko duomenys skirsis. Galimas atvejis, kad geriau naudoti nutolusią duomenų prieigą. Galima teigti, kad nutolusių duomenų prieigos paslaugos tiesės trukdžiai atsiranda dėl interneto apkrautumo.



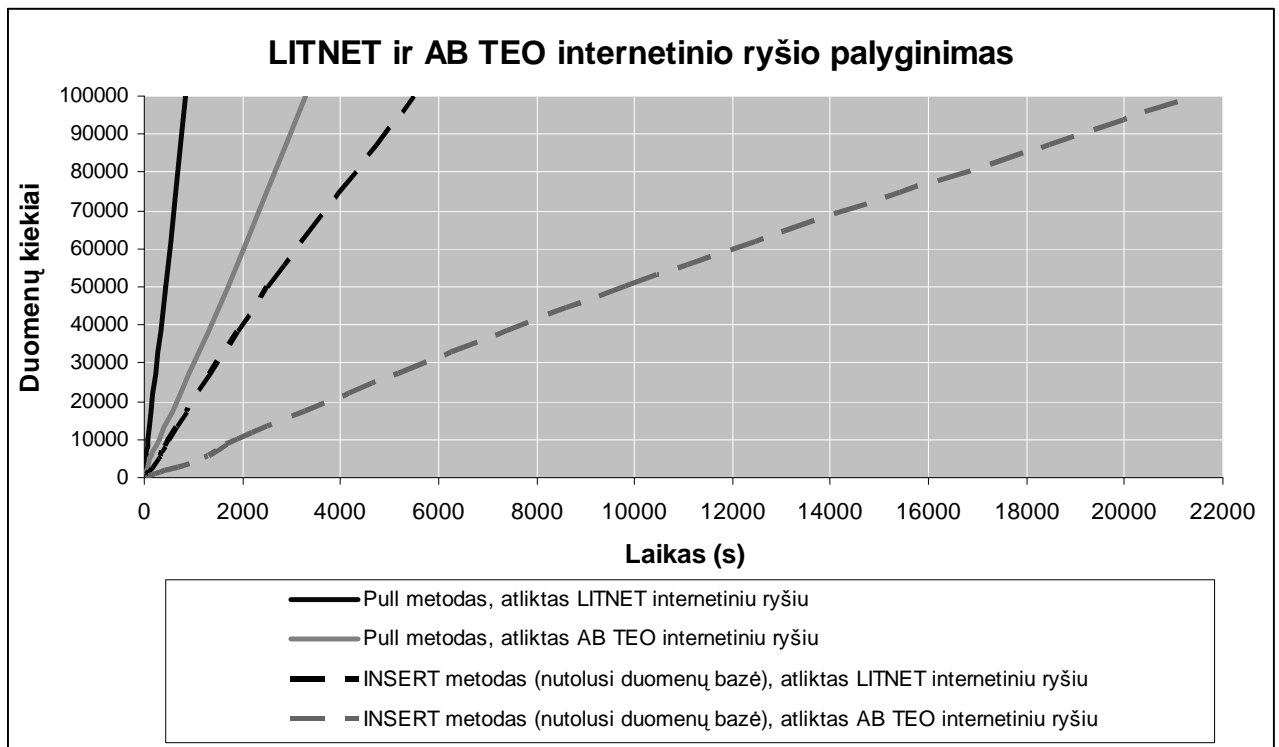
27 pav. Nutolusių duomenų prieigos paslaugos ir tiesioginės nutolusios duomenų bazės naudojimo DELETE operacijos palyginimas

Pritaikant taikomąją programą buvo eksperimentuojamas nutolusių duomenų prieigos ir tiesioginės nutolusios duomenų bazės naudojimo DELETE operacijos. Naudojant 5 lentelės ir nuo 100 iki 100 000 lentelės įrašų, bandymas buvo kartojamas po 10 kartų ir išvedamas aritmetinis vidurkis. Šios operacijos palyginimas pavaizduotas 27 paveiksle. Iš grafiko galima matyti, kad naudojant tiesioginę nutolusią duomenų bazę operacija atliekama greičiau. Tai lėmė interneto pralaidumas ir apdorojamų duomenų kiekis vienos transakcijos metu. Naudojant lėtesnio interneto ryšį, grafiko duomenys skirsis. Galimas atvejis, kad geriau naudoti nutolusią duomenų prieigą.



28 pav. Nutolusių duomenų prieigos paslaugos ir tiesioginės nutolusios duomenų bazės naudojimo SELECT operacijos palyginimas

Pritaikant taikomąją programą buvo eksperimentuojamas nutolusių duomenų prieigos ir tiesioginės nutolusios duomenų naudojimo SELECT operacijos. Naudojant 5 lentelės ir nuo 100 iki 100 000 lentelės įrašų, bandymas buvo kartojamas po 10 kartų ir išvedamas aritmetinis vidurkis. Šios operacijos palyginimas pavaizduotas 28 paveiksle. Iš grafiko galima matyti, kad naudojant tiesioginę nutolusią duomenų bazę operacija atliekama greičiau. . Tai lėmė interneto pralaidumas ir apdorojamų duomenų kiekis vienos transakcijos metu. Naudojant lėtesnio interneto ryšį, grafiko duomenys skirsis. Galimas atvejis, kad geriau naudoti nutolusią duomenų prieigą. Galima teigti, kad nutolusių duomenų prieigos paslaugos ir tiesioginės nutolusios duomenų bazės tiesių trukdžiai atsiranda dėl interneto apkrautumo.



29 pav. LITNET ir AB TEO internetinio ryšio palyginimas, naudojant nutolusių duomenų prieigos Pull metodą ir tiesioginės nutolusios duomenų bazės naudojimo INSERT operacija

Pritaikant taikomąją programą buvo eksperimentuojamas LITNET ir AB TEO teikiamas internetinis ryšys, naudojant nutolusių duomenų prieigos Pull metodą ir tiesioginės nutolusios duomenų bazės naudojimo INSERT operaciją. Eksperimentui buvo naudojama 5 lentelės ir nuo 100 iki 100 000 lentelės įrašų, bandymas buvo kartojamas po 10 kartų ir išvedamas aritmetinis vidurkis. LITNET ir AB TEO internetinio ryšio palyginimas pavaizduotas 29 paveiksle. Spėjama, kad naudojantis AB TEO internetiniu ryšiu nustatinėjant nutolusių duomenų prieigos paslaugos Push metodo ir kitų tiesioginės nutolusios duomenų bazės naudojimo operacijų našumus, eksperimentas duos panašų rezultatą, kuris skirsis apie keturis kartus nuo eksperimente naudojamo LITNET internetinio ryšio.

Atlikus eksperimentą, nustatyti nauji kriterijai, turintys įtakos duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodo pasirinkimui.

Nauji kriterijai, atlikus eksperimentą:

- Transakcijos metu yra apdorojamas didelis duomenų kiekis;

Žiūrint į grafinius rezultatus, matomi dideli skirtumai tarp duomenų apsikeitimo metodų. Todėl buvo iškeltas šis kriterijus, kad susiaurintų duomenų apsikeitimo metodo pasirinkimą. Vartotojas, apdorodamas didelius kiekius, gali naudoti tiesioginę nutolusią duomenų bazę. Ją naudodamas vartotojas nepraras laiko sąnaudų, apdorodamas didelius duomenų kiekius vienos transakcijos metu.

- Internetinio ryšio sparta, kuri turi įtakos apdorojamiems duomenims.

Eksperimentas buvo atliekamas naudojantis LITNET interneto ryšiu, kurio našumas tris kartus didesnis nei naudojantis TEO paslaugomis. Taigi galima kelti kriterijų, kad, neprarandant duomenų apsikeitimo našumo, reikia naudoti spartų interneto ryšį prie nutolusios duomenų bazės.

Atsižvelgiant į naujus kriterijus, buvo papildytas duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio pasirinkimo procesas, kuris pateikiamas 30 paveiksle.

Pateikiamas papildytas duomenų apsikeitimo tarp DBVS ir mobilaus pasirinkimo proceso žingsnių aprašymas:

- 1) žingsnio metu nustatoma, ar palaikomas darbas su reliacinėmis bazėmis;
- 2) žingsnio metu nustatoma, ar yra galimybė įdiegti IS kūrėjo sukurtas taikomas programas;
- 3) žingsnio metu nustatoma, ar transakcijos metu apdorojamas didelis duomenų kiekis. Jos atlikimo metu apdorojami dideli duomenų kiekiai, vykdant kelias SQL užklausas. Naudojant mobilųjį įrenginį, dideliu kiekiu yra įvardijama, kai transakcijos metu apdorojama 1000 ir daugiau įrašų. Jei sąlyga tenkinama, pasirenkamas 5) kriterijus, kuris atsižvelgia į interneto spartą;
- 4) žingsnio metu nustatoma, ar reikalingas mobilaus įrenginio autonominis darbas su duomenimis;
- 5) žingsnio metu nustatoma, ar palaikomas darbas su reliacinėmis bazėmis;
- 6) žingsnio metu nustatoma, ar yra galimybė įdiegti IS kūrėjo sukurtas taikomas programas;
- 7) žingsnio metu nustatoma, ar transakcijos metu apdorojamas didelis duomenų kiekis. Jos atlikimo metu apdorojami dideli duomenų kiekiai, vykdant kelias SQL užklausas. Naudojant mobilųjį įrenginį, dideliu kiekiu yra įvardijama, kai transakcijos metu apdorojama 1000 ir daugiau įrašų. Jei sąlyga tenkinama, pasirenkamas 5) kriterijus, kuris atsižvelgia į interneto spartą;
- 8) žingsnio metu nustatoma naudoti mobilų DBVS. Duomenų apdorojimui pasirenkamas mobilusis įrenginys su savo mobilia duomenų baze;
- 9) žingsnio metu nustatoma, ar reikia peržiūrėti sistemos reikalavimus;
- 10) įrenginio arba sistemos architektūros keitimas;
- 11) tiesioginis nutolusios duomenų bazės naudojimas;
- 12) naudoti nutolusių duomenų prieigą;
- 13) naudoti suliejimo replikavimą.

5. Išvados

1. Atlikus literatūros analizę, buvo nustatyti trys metodai (nutolusių duomenų prieigos paslauga, tiesioginis nutolusios duomenų bazės naudojimas ir suliejimo replikavimas), kurie leidžia apsieisti duomenimis tarp DBVS ir mobilaus įrenginio.
2. Atlikus literatūros šaltinių analizę, buvo suformuoti kriterijai, kurie turi įtakos duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodo pasirinkimui kuriant IS darbo vietas mobiliesiems įrenginiams.
3. Pagal suformuotų kriterijų aibę pasiūlytas duomenų apsikeitimui tarp DBVS ir mobilaus įrenginio pasirinkimo procesas, kuriuo naudojantis IS kūrėjai nuosekliai ir tikslingai gali pasirinkti optimalų duomenų apsikeitimo metodą kiekvienos kuriamos IS atveju.
4. Siekiant patikrinti, ar pasiūlytame duomenų apsikeitimo metodų pasirinkimo procese naudojamų kriterijų skaičius yra pakankamas, buvo atliktas nutolusių duomenų prieigos ir tiesioginio nutolusios duomenų bazės naudojimo metodų eksperimentinis našumo tyrimas.
5. Eksperimentiniam nutolusių duomenų prieigos ir tiesioginio nutolusios duomenų bazės naudojimo metodų tyrimui buvo suprojektuota ir realizuota programinė įranga, palaikanti abu šiuos metodus.
6. Atlikus eksperimentą su nutolusia duomenų prieigos paslauga ir tiesiogine prieiga prie nutolusios DBVS, pastebėta, kad vykdant vieną transakciją, kurios metu apdorojami dideli duomenų kiekiai, žymiai greičiau duomenys apdorojami naudojant tiesioginę prieigą prie nutolusios duomenų bazės.
7. Eksperimento metu naudojant LITNET teikiamą internetinį ryšį (tinklo pralaidumas eksperimento metu buvo ne mažiau 7 Mbps) ir AB TEO internetinį ryšį (tinklo pralaidumas eksperimento metu buvo ne daugiau 512 Kbps) pastebėtas ženklus duomenų apsikeitimo tarp mobilaus įrenginio ir nutolusios DBVS našumo kritimas.
8. Pagal 6 ir 7 išvadas suformuluoti dar du kriterijai (internetinio ryšio pralaidumas, didelio duomenų kiekio apdorojimas vienos transakcijos metu), kurie lemia duomenų apsikeitimo tarp DBVS ir mobilaus įrenginio metodo pasirinkimą. Šiais kriterijais buvo papildytas pasiūlytas duomenų apsikeitimo procesas.
9. Procesą naudojantys IS kūrėjai gali papildyti procesą savo kriterijais, kurie gali būti suformuluoti dėl naudojamų technologijų ar esant kuriamos informacinės sistemos dalykinės srities tam tikrai specifikai.

6. Literatūros sąrašas

1. A. Silberschatz, H. F. Korth, S.Sudarshan. Database system concepts 4nd Edition, United States, 2001.- 916 p.
2. Andrew S. Tanenbaum, Maarten Van Steen. Distributed Systems: Principles and Paradigms (2nd Edition). Prentice Hall, 2006.-704 p.
3. D. Dove. A Technical Comparison of Replication and Remote Data Access Features in SQL Server 2005 Mobile Edition 3.0. 2005. [žiūrėta 2008-12-20], Prieiga per internetą: <http://msdn.microsoft.com/en-us/library/ms345126.aspx>
4. Yuri Breitbart, Henry F. Korth: Replication and Consistency in a Distributed Environment. J. Comput. Syst. Sci. 59(1): 29-69 (1999)
5. J. Gray and A. Reuter. Transaction Processing: Concepts and Techniques. Morgan Kaufmann, 1993.
6. J. Gray, P. Helland, P.O'Neil, The Dangers of Replication and a Solution
7. Microsoft SQL Server Compact. Comparing Remote Data Access and Merge Replication. [žiūrėta 2009-01-17], Prieiga per internetą [http://msdn.microsoft.com/en-us/library/ms172916\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms172916(SQL.90).aspx)
8. Microsoft SQL Server Compact. Comparing Remote Data Access and Merge Replication. [žiūrėta 2009-03-08], Prieiga per internetą [http://msdn.microsoft.com/en-us/library/ms172916\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms172916(SQL.90).aspx)
9. Microsoft SQL Server Compact. Merge replication. [žiūrėta 2008-10-06], Prieiga per internetą <http://msdn.microsoft.com/en-us/library/ms172449.aspx>
10. Nori, Anil. K. Mobile and Embedded Databases. 2007.
11. P.Dhingra, T.Swanson. Microsoft SQL Server 2005 Compact Edition., USA., 2007.-624 p.
12. Sujoy Paul. Pro SQL Server 2005 Replication. USA. 2005. – 995 p.
13. T. Özsu, P. Valduriez: Principles of Distributed Database Systems. 2nd Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1999. - 666 p.
14. Todd A. Anderson, Yuri Breitbart, Henry F. Korth, Avishai Wool: Replication, Consistency, and Practicality: Are These Mutually Exclusive? SIGMOD Conference 1998: 484-495
15. V. Kumar., Mobile Database Systems., Kansas City – USA., 2006.- 320 p.

16. Wiesmann, M.; Pedone, F.; Schiper, A.; Kemme, B.; Alonso, G. Database replication techniques: a three parameter classification. Proceedings The 19th IEEE Symposium on Volume , Issue , 2000.-206 – 215 p.