

Винахід стосується способу і системи оновлення бази даних, зокрема, стосується способу і системи синхронізації віддалених баз даних.

У багатьох підприємствах база даних серверу ("центральна") використовується для зберігання даних, що стосуються багатьох працівників або віддалених користувачів підприємства. Базу даних серверу звичайно роблять доступною для віддалених комп'ютерних систем, щоб зробити інформацію більш доступною для віддалених користувачів. Застосування бази даних серверу, доступної для віддалених комп'ютерних систем, полегшує розповсюдження інформації у компанії. Дистанційний доступ стає особливо важливим у випадках, коли відділки продажу або численні працівники компанії працюють за межами офісу. Наприклад, робота віддалених працівників може залежати від інформації, що зберігається у базі даних і стосується інвентаризаційних змін, цінових даних і подій у компанії. Замість запровадження постійного зв'язку невизначеної тривалості з базою даних, пов'язаного з сплатою за зв'язок або орендою телефонних ліній, використовують уривчастий зв'язок комп'ютерів віддалених користувачів з сервером для одержання доступу до його бази даних. За таких обставин віддалені комп'ютерні системи звичайно зберігають у себе частину бази даних серверу, щоб віддалений відділок міг функціонувати навіть тоді, коли клієнт не має з сервером зв'язку. Тоді, використовуючи уривчастий зв'язок, до серверу надсилають лише зміни, що були зроблені клієнтом, а від серверу до клієнта — лише ті зміни, що стосуються клієнта. Такий тип організації віддаленої комп'ютерної системи називають Базою Даних з Уривчастими Зв'язками (БДУЗ). БДУЗ знайшли широке застосування у автоматизації відділків продажу, обробці страхових випадків і взагалі при виконанні мобільних функцій.

Важливим фактором у таких комп'ютерних системах є своєчасний і ефективний обмін інформацією між клієнтом і базою даних серверу. Термін "синхронізація бази даних" часто використовують для опису процесу підтримання цілісності і відповідності даних у базах даних серверів і клієнтів. Існують багато варіантів такої синхронізації. Згідно з деякими відомими схемами, при обміні даними з ними пов'язують часову позначку, яка вказує на дату останнього оновлювального обміну інформацією. База даних серверу і база даних клієнта використовують цю часову позначку для визначення, які саме записи були оновлені. У інших схемах використовують бітову позначку для записів, що зазнали зміни. У інших варіантах передбачено зберігання попередніх даних для стеження за змінами. Звичайно надсилають лише ті записи, які зазнали змін після останньої дати обміну.

У системах БДУЗ для забезпечення постійного доступу необхідно, щоб кожний клієнт підтримував локальну копію спільних даних. Крім того, уривчастий зв'язок робить неможливим прямий обмін даними між клієнтами і, отже, для здійснення зв'язку між клієнтами і забезпечення спільного доступу до даних виникає потреба у певному посереднику. Звичайно функції такого посередника виконує база даних серверу, дані і структура якої є об'єднанням даних і структур клієнтів. Такий сервер є завжди доступним для клієнтів, що мають з ним уривчастий зв'язок. Клієнт надсилає оновлені дані до серверу. Сервер визначає тих клієнтів, для яких ці дані є спільними і які потребують копії цього оновлення. Отже, такий "клієнтоцентричний" підхід вимагає від серверу формувати окремі набори даних для кожного клієнта. На жаль, таке рішення призводить до ускладнення з появою кожного додаткового клієнта і породжує обмеження щодо максимальної кількості клієнтів, що мають обслуговуватись, оскільки для кожного додаткового клієнта сервер має повторювати усі операції, які він виконав для інших клієнтів.

Існує багато способів надійної синхронізації. Один з способів синхронізації віддалених баз даних описаний у патенті США 5 649 195. У цій системі кожний з віддалених комп'ютерів містить дублікат головної бази даних, а центральний комп'ютер зберігає у файлі усі зміни, зроблені у головній базі даних протягом зумовленої одиниці часу. Кожний запис зміни, що відповідає одній й тій же одиниці часу, пов'язують з певним номером сповіщення і порядковим номером. Послідовний номер визначає порядкове місце запису зміни у файлі. Останній за порядком з цих записів є останнім записом, пов'язаним з певним номером сповіщення. Для наступної одиниці часу номер сповіщення інкрементується і знову встановлюється початковий порядковий номер. Після зумовленої кількості часових одиниць файл записів змін надсилається до віддалених копій бази даних. Віддалені комп'ютери, використовуючі номери сповіщення і порядкові номери, перевіряють порядок записів змін, щоб визначити, чи всі записи змін у сповіщенні були прийняті і належним чином використані для оновлення віддаленого дублікату бази даних.

Іншу систему оновлення інформації у базі даних описано у патенті США 5 491 820. Тут для системи клієнт/сервер у сервері використано об'єктно-орієнтовану базу даних, яка забезпечує даними уривчасто приєднаних клієнтів. У цій системі база даних містить об'єкти даних, які можуть бути передані до клієнта, з'єднаного з сервером, і пізніше оновлені після відновлення зв'язку з сервером. Приймаючи цю інформацію, клієнт пов'язує час прийому з об'єктом даних. Надсилаючи вимогу оновлення об'єкту, клієнт передає список об'єктів, що підлягають оновленню, разом з часом прийому для кожного об'єкту. Сервер зберігає час останньої модифікації для кожного об'єкту і порівнює час прийому для кожного об'єкту даних з списку об'єктів, одержаного від клієнта, з часом останньої модифікації. Сервер формує нову версію об'єктів, які були модифіковані з часу останнього прийому.

У типовій системі синхронізації індивідуальний файл, що містить зміни і має бути надісланий, створюється у сервері для кожного клієнта системи. Це перевантажує сервер, збільшуючи у ньому об'єм обробки, необхідної для створення цих файлів, причому цей об'єм прямо пропорційно залежить від кількості клієнтів у системі. Виникає потреба у системі синхронізації бази даних, яка дозволяє масштабувати базу даних згідно з зростанням кількості клієнтів без погіршення ефективності роботи.

Суть винаходу

Взагалі винахід стосується способу, який зменшує зберігання і обробку у системі БДУЗ. Зменшення об'єму обробки робить сервер здатним обслуговувати більше клієнтів. Зокрема, винахід включає спосіб синхронізації комп'ютерної системи клієнта з базою даних серверу шляхом розділення даних, що підлягають розподіленню сервером, на групи і призначення однієї або більше груп кожній з комп'ютерних систем клієнтів. Зміни даних у групах зберігаються для подальшої передачі до клієнтів, які є абонентами

груп.

Коли комп'ютерна система вводить номер або "з'єднується" з сервером, вона надсилає до нього зміни, внесені у її локальну базу даних. Далі сервер може оновити свою базу даних згідно з змінами, надісланими клієнтом. Після оновлення його бази даних сервер визначає, для яких груп ці зміни є спільними, і створює для цих груп файли модифікації. Після цього клієнти можуть завантажити файли модифікації для кожної з груп, що включають клієнта як абонента. Оскільки файли модифікації можуть містити надлишкові дані, клієнт оцінює їх, щоб визначити, чи треба прийняти, або злити, або ігнорувати певні зміни.

Кожний клієнт має список груп, абонентом яких він є. Коли клієнт з'єднується з сервером, він запитує файли модифікації для груп цього списку, з'єднує завантажені файли модифікації, відфільтровує зайві дані і оновлює власну базу даних.

Отже, об'єктом винаходу є спосіб синхронізації бази даних і системи, що надає базі даних серверу здатність обслуговувати збільшену кількість комп'ютерних систем клієнтів.

Задачею винаходу є досягти цього групуванням даних для полегшення масштабування системи бази даних.

Ці і інші задачі винаходу наведено для фахівців у подальшому описі з посиланнями на креслення, у яких:

фіг. 1 - схема БДУЗ,

фіг. 2 - детальна блок-схема компонентів клієнта і компонентів серверу, що використовуються у процесі оновлення клієнта сервером,

фіг. 3a, 3b, 3c - приклади групування даних у базі даних серверу і утворення зв'язку клієнта з цими групами згідно з винаходом,

фіг. 4 - схема формату файла модифікації згідно з винаходом,

фіг. 5a, 5b, 5c - схема алгоритму способу винаходу.

Детальний опис винаходу

Фіг. 1 ілюструє основні компоненти системи 10 БДУЗ згідно з кращим втіленням винаходу. Система 10 включає комп'ютерні системи 16a, 16b, 16c клієнтів. Кожний клієнт має прикладні програми і локальну базу даних 15a, 15b, 15c. Комп'ютер-сервер 18 містить прикладні програми і базу даних 15d серверу, до яких комп'ютерні системи 16 клієнтів одержують доступ через проміжні зв'язки 26. У сервері 18 працює програма-адміністратор комп'ютерної мережі, яка керує доступом до мережі або до її частин. Комп'ютерні системи 16 клієнтів містять дані, спільні з базою даних 18 серверу і можуть одержувати доступ до серверу 18 через Інтернет, локальну мережу, глобальну мережу 26 або через телефонну лінію і модем. Сервер 18 може мати зв'язок з локальною мережею організації.

Структура і функції системи 10 БДУЗ дозволяють її серверу 18 і базі даних 15d серверу обслуговувати клієнтів більш ефективно ніж існуючі системи. Зокрема, винахід передбачає організацію вмісту бази даних серверу у вигляді певних груп і стеження за змінами даних у цих групах, а не у окремих клієнтів. Періодично для кожної групи створюється файл модифікації ("дельта" або "оновлення"), який містить усі відповідні зміни, що мали місце після створення останнього файлу модифікації. Комп'ютерні системи 16 клієнтів асоціюють з певними групами і, коли клієнт встановлює зв'язок з сервером 18, файли модифікації, що відповідають групі, передаються до клієнта або клієнтів, пов'язаних з цією групою і використовуються для оновлення індивідуальних баз даних клієнтів.

Комп'ютерні системи 16a, 16b, 16c можуть бути віддаленими. У цьому випадку, якщо користувач однієї з цих віддалених систем 16 бажає оновлення даних поточною інформацією від бази даних 18 серверу, комп'ютерна система 16 встановлює з сервером зв'язок через глобальну мережу або телефонну лінію. Перевагою системи і способу оновлення комп'ютерних систем клієнтів, передбачених винаходом, є те, що комп'ютерні системи клієнтів можна додавати до системи БДУЗ без створення у цій системі окремих файлів модифікації для кожного клієнта, доданого до системи. У такій системі дані у сервері можуть бути організовані у вигляді груп на основі вмісту і семантики. Кожному з клієнтів залежно від його вимог призначаються одна або більше таких груп. Періодично сервер визначає для кожної з груп, які дані зазнали змін з часу останньої оцінки, і записує ці зміни у файл модифікації. Коли клієнт зв'язується з сервером, він одержує файли модифікації групи, абонентом якої він є, об'єднує одержані файли модифікації, відфільтровує зайві дані і оновлює локальну базу даних.

Отже, винахід передбачає "інформаційно-центричний" підхід до розподілу даних. Згідно з таким підходом, складності зберігання і обробки в базі даних серверу не залежать від кількості клієнтів, і це полегшує масштабування серверу. Замість уваги до даних, потрібних індивідуальному клієнту, спосіб згідно з винаходом передбачає стеження за змінами у підмножинах даних, що стосуються груп. Отже, сервер має лише стежити за змінами у цих підмножинах даних замість стеження за змінами, що стосуються окремих клієнтів. Клієнти одержують від серверу ці підмножини, які містять релевантні клієнтам дані, і відфільтровують зайві дані, які їх не стосуються. Оскільки сервер стежить за обмеженою сукупністю підмножин даних, релевантних групам клієнтів, а не усім існуючим клієнтам, загальна масштабованість системи підвищується.

Синхронізація бази даних

Фіг. 2 містить блок-схему основних компонентів, що використовуються у процесі оновлення баз даних системи 10 БДУЗ. На фіг. 2 не показані відомі фахівцям компоненти стандартних комп'ютерних систем, які не є необхідними для розгляду винаходу, наприклад, адресні буфери, буфери пам'яті та ін. Комп'ютерні програми і файли даних (фіг. 2) використовуються на різних стадіях способу згідно з винаходом. Процедури винаходу можуть бути реалізовані на ПК, робочих станціях і серверах.

Як уже відзначалось, система винаходу підвищує здатність системи баз даних обробляти дані і обслуговувати більше клієнтів, ніж відомі системи. Групуванням даних у базі даних 15d у групи 80 даних і зв'язуванням індивідуальних файлів 86 модифікації ("дельта") з кожною групою 80 загальний об'єм операцій у системі 10 БДУЗ зменшується і, отже, масштабованість системи баз даних поліпшується. Наприклад, у

системі 15d бази даних визначено чотири групи даних 80a, 80b, 80c, 80d. Компоненти даних у кожній з груп не є обов'язково ексклюзивними, тобто угруповання даних можуть містити посилання на одні й ті ж компоненти. Для кожної з груп 80 існує ряд пов'язаних з нею файлів модифікації. Адміністратор системи періодично створює для кожної групи новий файл модифікації, який містить усі оновлення даних, які відповідають групі і відбулись з часу створення останнього файлу модифікації. Файли 86a, 86b, 86c, 86d модифікації відповідають групам даних 80a, 80b, 80c, 80d. Файли 86aa, 86dd оновлень відповідають файлам 86a, 86d модифікації. Коли будь-який компонент даних у одній з груп 80a - 80d зазнає змін, ці зміни записуються у базу даних. Хоча з системою 10 БДУЗ можуть бути з'єднані сотні клієнтів, для ідентифікації змін у групах 80a - 80d у цьому прикладі використовується відносно невелика кількість файлів 86a - 86dd. У відомих системах БДУЗ довелося би мати файли модифікації для кожного клієнта і ідентифікувати у них усі змінені компоненти даних, до яких клієнт має доступ. Отже, у цих системах система керування базою даних має зберігати і підтримувати набір файлів клієнта для кожного з клієнтів системи. Використання системи, передбаченої винаходом, у якій компоненти даних групуються у групи 80 даних і пов'язуються з файлами модифікації для кожної з цих груп, дозволяє суттєво знизити загальний об'єм операцій у системі БДУЗ.

Коли комп'ютерна система 16 клієнта встановлює зв'язок з базою даних 15d серверу, система 100 керування базою даних виконує функції інтерфейсу між базою даних 15d серверу і комп'ютерною системою 16с клієнта. Слід відзначити, що комп'ютерна система 16с репрезентує комп'ютерні системи 16a, 16b клієнта. Система 15с також має систему 106 керування базою даних, яка є інтерфейсом між локальною базою даних 15с і іншими комп'ютерними системами. У системі 10 комп'ютерна система 16 клієнта користується частиною даних, що містяться у базі даних і можуть розглядатись як підмножина даних бази даних серверу, яка зберігається у локальній базі даних 15с. Локальна база даних 15с містить інформацію, що використовується у роботі комп'ютерної системи 16с клієнта. Система 16с одержує дані від пов'язаної з нею групи 80 даних і зберігає її у локальній базі даних 15с. Подібним чином, коли зазнають змін компоненти даних у локальній базі даних 15с, ці модифікації передаються від комп'ютерної системи 16с клієнта до бази даних 15d серверу і використовуються для оновлення останньої. Під час передачі інформації від локальної бази даних 15с або від бази даних 15d серверу дані, що передаються, синхронізуються з відповідною базою даних.

Комп'ютерна система 16с клієнта, одержуючи доступ до бази даних. 15d серверу, має список або таблицю 88с сукупності груп, які їй потрібні. Комп'ютерна система клієнта входить у контакт з системою серверу і запитує поточні файли модифікації для груп, пов'язаних з комп'ютерною системою клієнта. Відповідні операції у комп'ютерній системі 16с клієнта виконує керуючий програмний модуль 107. Протягом сеансу зв'язку комп'ютерна система 16с клієнта може передавати до бази даних 15d серверу ті транзакції і зміни даних, які впливають на дані у базі даних 15d. Зміни даних, які мали місце у комп'ютерній системі 16с клієнта і впливають на дані у базі даних 15d серверу, вказуються у файлах 105 модифікації. Після цього база даних 15d серверу оновлюється згідно з змінами, що надійшли від комп'ютерної системи клієнта і відповідні зміни вносяться у файли модифікації, пов'язані з групами, що містять дані, модифіковані клієнтом.

Система 100 керування базою даних може отримати доступ до файлів модифікації, що відповідають групам даних, до яких клієнт має доступ. Ці файли містять послідовність транзакцій і послідовність операцій над відповідними групами даних. Файли 86 модифікації, що відповідають групам даних, до яких клієнт має доступ, передаються після цього до комп'ютерної системи 16с клієнта. У існуючих системах до системи керування базою даних клієнта надсилається єдиний файл клієнта. Згідно з винаходом, на відміну від таких систем, до комп'ютерної системи клієнта передаються кілька файлів модифікації, пов'язаних з обраними групами даних. Комп'ютерна система 16с клієнта оцінює кожний з цих файлів модифікації, щоб визначити, які оновлення мають бути стерті, реорганізовані або об'єднані для належної синхронізації локальної бази даних 15с з базою даних 15d серверу. Для збереження цілісності бази даних комп'ютерна система 16с клієнта може профільтувати і/або об'єднати файли 80 модифікації. Система 106 керування базою даних використовує обробляючий модуль 108 для фільтрування і об'єднання багатьох файлів модифікації, прийнятих від системи 100 керування базою даних.

З одною групою можуть бути асоційовані кілька файлів модифікації (фіг. 2). У кожному наборі або послідовності файлів модифікації, пов'язаних з одною групою, наприклад, 86a, 86aa, кожному з цих файлів модифікації призначають порядкові номери. Коли комп'ютерна система 16с клієнта одержує доступ до серверу 18, ця комп'ютерна система клієнта передає до серверу порядковий номер останнього файлу оновлення, прийнятого нею від серверу. Сервер тоді передає до комп'ютерної системи 16с клієнта лише ті файли модифікації, які не були передані раніше, базуючись на порядковому номері, прийнятому від комп'ютерної системи 16с. Сервер надсилає лише файли модифікації з порядковими номерами, більшими за порядковий номер, прийнятий від комп'ютерної системи клієнта. Якщо кількість транзакцій, що містяться у файлі, перевищує певне значення, може бути створений окремий файл 86 модифікації після завершення зумовленого періоду часу або після того, як певна група приєднаних комп'ютерних систем клієнтів завершить внесення змін у дані певних груп даних. Винахід включає також інші очевидні для фахівців схеми визначення умов створення багатьох файлів модифікації для групи.

Фіг. 3a, 3b, 3c ілюструють основну схему групування клієнтів. Приклади 3a, 3b, 3c розглядаються стосовно реляційної бази даних і відповідних таким базам процедур програмування. Щоб згрупувати дані, можна використати різні способи. Спосіб групування у конкретній системі залежить від багатьох факторів, включаючи швидкість обробки у сервері, розмір диску серверу і ширину діапазону мережі. Групування може бути статичним і динамічним.

При статичному групуванні (фіг. 3a, 3b, 3c) групи даних є відомими а priori і їх визначає адміністратор. Комп'ютерні системи клієнтів призначаються як абоненти або прив'язуються до обраних груп. Статичне групування знижує функціональне навантаження серверу, обслуговуючи розповсюдження файлів модифікації незалежно від приєднаних клієнтів. Як відомо, горизонтальні або вертикальні сегменти зв'язків

даних називають фрагментами і визначають а ріорі схемою сегментування. Фрагменти містяться у одній або кількох групах.

Можна також використовувати динамічне групування (не ілюстроване), при якому кількість і побудова груп залежать від параметрів, що постійно змінюються, наприклад, від поточного набору транзакцій і/або поточного набору клієнтів. Один з типів динамічного групування передбачає окрему групу для кожної транзакції і комп'ютерні системи клієнтів стають абонентами груп залежно від типів транзакцій. Такий тип групування називають групуванням за типами транзакцій. Іншим типом динамічного групування є групування за потужністю. Таке групування передбачає формування мінімальної кількості груп, щоб клієнти ставали абонентами мінімальної кількості груп і тому одержували мінімум надлишкових даних.

Динамічне групування допомагає знизити для клієнта ширину діапазону і кількість

операцій фільтрування, оскільки передбачає надсилання лише тих даних, що стосуються поточного набору приєднаних клієнтів. Однак, таке групування більше завантажує сервер операціями, оскільки сервер має координувати угруповання для роботи з комбінаціями клієнтів і даних. При групуванні за потужністю сервер координує оновлення груп для передачі даних кожному набору приєднаних клієнтів. При групуванні за типами транзакцій сервер має зв'язок з кожним клієнтом окремо для виконання операцій оновлення для кожного релевантного клієнта.

На фіг. 33а показано варіант організації даних у сервері. Запис 112 студента містить ідентифікатор 114 студента, його ім'я 116, його номер 118 телефону і змінну 120 класу, яка може визначати студентів 122 останнього курсу або студентів-випускників 124. Загальна схема серверу включає також запис 134 реєстрацій, який містить ідентифікатор 136 студента, ідентифікатор 138 курсу і ідентифікатор 140 студента-випускника. Згідно з схемою даних серверу передбачено також запис 150 курсу, який включає ідентифікатор 152 курсу, назву 154 курсу, і змінний запис 156 класу, який містить характеристику 158 студента останнього курсу і характеристику 160 студента-випускника.

Фіг. 3b містить таблицю ідентифікаторів груп. Групам даних у базі даних призначені ідентифікатори 202, 206, 210, 214 і 218. Група "студенти останнього курсу" (ідентифікатор 202) включає інформацію 112 про студента, де клас визначено як студенти 122 останнього курсу мінус номер телефону 118. Група "студенти-випускники" (ідентифікатор 206) включає інформацію 112 про студента, де клас визначено як студенти-випускники 124 без номера телефону 118, Група "студентські телефони" (ідентифікатор 210) включає запис 112 студента лише з ідентифікатором 114 студента і телефонним номером 118. Група "всі курси" (ідентифікатор 214) включає всі дані про курси з запису 150. Клас "всі реєстрації" (ідентифікатор 218) включає всі дані запису 134 реєстрацій.

Фіг. 3с ілюструє призначення клієнтів до груп. Спочатку цю таблицю використовує система керування базою даних серверу для визначення, які файли 86 модифікації є необхідними для клієнта. Клієнти останнього курсу призначені до групи 202 "студенти останнього курсу", до групи 214 "усі курси" і до групи 218 "всі реєстрації". Клієнти, які є студентами-випускниками, призначені до групи 206 "студенти-випускники", до групи 214 "усі курси" і до групи 218 "всі реєстрації". Клієнти-професори призначені до групи 206 "студенти-випускники", до групи 202 "студенти останнього курсу", до групи 210 "студентські телефони", до групи 214 "усі курси" і до групи 218 "всі реєстрації". Клієнти реєстрації призначені до групи 218 "всі реєстрації". Дані цих груп і асоціації використовуються для оновлення і синхронізації комп'ютерів системи 10 БДУЗ.

Сервер 18 обробляє згруповані оновлення, прийняті від клієнтів 16, а оновлення, виконані сервером 18, утворюють для кожної групи дельта-файл на основі знання даних, спільних з кожним клієнтом. Фіг. 4 ілюструє файл 86 модифікації (дельта-файл). Файл 86 модифікації для клієнта містить усі операції над набором даних, що відповідають змінам даних групи, до якої був призначений клієнт. Файл 86 модифікації включає послідовність транзакцій, наприклад, транзакцію А, транзакцію С, транзакцію R і транзакцію J, кожна з яких ідентифікується унікальним глобальним порядковим номером (наприклад, 1, 2, 3, ...). Ці номери визначають порядок виконання транзакцій у сервері 18. Кожна транзакція репрезентує послідовність операцій 410, кожна з яких має унікальний порядковий номер, наприклад, OP1, OP2, OP3, OP4 і OP5 і ці номери вказують на порядок виконання операцій у транзакції. Кожна операція виконується лише над одним фрагментом. Кожна транзакція ідентифікує комп'ютерну систему 16 клієнта, у якій вона була вперше виконана як частина запису транзакції. Транзакції відповідає ідентифікаційний номер 416 клієнта.

Використовуючи порядкові номери 404 транзакцій, комп'ютерна система 16 може реконструювати порядок виконання операцій у сервері 18 незважаючи на фрагментацію операцій у файлах 400 модифікації. Оскільки кожна операція виконується з одним фрагментом, між операціями не існує інтерреляційної залежності, за винятком залежностей від сторонніх ключів. Інтерреляційні операції на відношеннях без залежностей від сторонніх ключів можуть виконуватись у будь-якому порядку. Такі операції зберігають послідовний порядок у сервері. Якщо сторонній ключ породжений відношенням, яке називають "батьківським відношенням", а первинний ключ - відношенням, яке називають "нащадковим відношенням", то інтерреляційні операції на відношеннях з залежностями сторонніх ключів упорядковуються таким чином, що введення у батьківське відношення передує відповідному введенню у нащадкове відношення, а стирання у нащадковому відношенні передує стиранню у батьківському відношенні.

Фільтрування

Як уже відзначалось, групи даних можуть містити інформацію, яку клієнт не повинен приймати, або таку, яку він уже має. Наприклад, якщо клієнт зробив зміну у даних бази даних, він вже має цю зміну або запис зміни, зробленої у базі даних. Отже, якщо файли модифікації, які містять усі виконані операції, включають операції оновлення, виконані певним клієнтом, то ці операції не вимагають обробки або використання, коли комп'ютерна система 16 клієнта приймає файли 86 модифікації для відповідних груп даних. Таким чином, у процесі виконання оновлень комп'ютерна система 16 здійснює операції фільтрування, відображення і об'єднання для забезпечення цілісності локальної бази даних. Типи обробки,

що виконується у комп'ютерній системі 16 клієнта включають рефлексивне фільтрування, об'єднання операцій і референційне цілісне упорядкування. Фільтрування розглядається на прикладах фіг. 3а, 3б, 3с.

Рефлексивним фільтруванням називають перевірку комп'ютерною системою клієнта файлів 86 модифікації на наявність переданих цій системі транзакцій, які походять від цієї системи. Схемним відображенням є процес виконання клієнтом синонімічних перетворень над іменами груп, які у локальній базі даних були найменовані інакше, ніж у базі даних 15d серверу. Цим під час оновлення забезпечується належна відповідність груп і цілісність даних груп у базах даних. Фільтрування даних згідно з відношенням, що визначається даними, внутрішніми відносно нього, називають інтерреляційним фільтруванням. Таке фільтрування видаляє зайві колонки і рядки даних, які могли б бути надіслані до клієнта. Наприклад, "студенти останнього курсу" одержують дані на базі відношень реєстрацій, але клієнти останнього курсу не мають права знайомитись з інформацією для випускників. Отже, інформація для випускників ігнорується або стирається з групи записів реєстрацій, коли вона передається до студента останнього курсу. Подібно до цього, клієнти-випускники одержують усі дані з запису курсів, але мають право одержувати інформацію лише про випускні курси. Фільтрування групи курсів (фіг. 3а, 3б, 3с) для студентів останнього курсу означає, що інформація для випускників стирається або ігнорується.

"Фільтрування даних" у групі, що є спільною, яке здійснюється на підставі даних таблиці з іншої бази даних, називають інтерреляційним фільтруванням. Таке фільтрування є необхідним тому, що не усі зайві дані відфільтровуються даними у групі. Наприклад, клієнти останнього курсу мають одержувати лише дані 218 групи реєстрацій для студентів останнього курсу, але для визначення, чи група 134 реєстрацій є групою для студентів останнього курсу, ідентифікатор курсу необхідно визначати, використовуючи групу 134 курсів. У прикладі групування (фіг. 3б) студент останнього курсу приймає усі дані реєстрацій у групі 218 "усі реєстрації". Після цього клієнт останнього курсу, використовуючи групу 214 курсів, визначає, які дані реєстрацій мають бути відфільтровані. Видалення дублікатів є процесом виявлення і стирання/відкидання дубльованих транзакцій. Такі транзакції трапляються внаслідок наявності однакових даних у кількох групах. У прикладі (фіг. 3а, 3б, 3с) клієнт-професор об'єднує групу 202 студентів останнього курсу, групу 206 студентів-випускників і групу 210 студентських телефонів. Якщо видалити студента, транзакція буде надіслана до клієнта-професора двічі, оскільки група 210 студентських телефонів і група 202 студентів останнього курсу визначені через запис 112 студентів. У кожній з них є ідентифікатор 112 студента. Отже, клієнт-професор має виявити дубльовану транзакцію і виконати одне стирання для локальної бази даних 15d.

"Об'єднання операцій" включає об'єднання операцій, які були подрібнені при занесенні у групи. Операції транзакції можуть зазнати подрібнення при занесенні у групи. Клієнти мають забезпечити об'єднання цих операцій для збереження цілісності. Розглянемо послідовність операцій у сервері:

ВВЕСТИ (1000, "Джон Сміт", 555-5555, СТУД. ОСТ. КУРСУ) У "Студенти";

СТЕРТИ З "Студенти" ДЕ Ідент. студента = 10000.

Ця послідовність операцій для груп "Студенти останнього курсу" і "Студентські телефони" буде розділена на такі послідовності:

ВВЕСТИ (1000, "Джон Сміт", СТУД. ОСТ. КУРСУ) У "Студенти";

СТЕРТИ З "Студенти" ДЕ Ідент. студента = 1000;

і

ВВЕСТИ (1000, 555-5555) У "Студенти";

СТЕРТИЗ "Студенти" ДЕ Ідент. студента- 1000.

Клієнт-професор має об'єднати транзакції з "Студентів останнього курсу" і "Студентських телефонів", таким чином, щоб (1) введення були об'єднані в одну операцію введення, (2) дубльовані стирання були усунені і (3) об'єднане введення було виконане перед стиранням.

"Референційне цілісне упорядкування" включає таку обробку файлів модифікації, яка забезпечує цілісність при об'єднанні транзакцій з посиланнями на інші групи. У процесі об'єднання транзакцій клієнт має брати до уваги обмеження референційної цілісності. Виконання транзакцій у неналежному порядку клієнтом може призвести до відхилення у клієнта операції, яка була припустимою у сервері, і до порушення цілісності даних. Розглянемо два введення у сервері:

ВВЕСТИ (6555, "Теорія черг", ВИПУСКНИКИ) У "Курси";

ВВЕСТИ (1001, 6555) У "Реєстрації".

Ці два введення будуть прийняті клієнтом-студентом випускником від груп "Усі курси" і "Усі реєстрації" відповідно. Відзначимо, що "Курси. Ідентифікатор курсу" є стороннім ключем "Реєстрація Ідентифікатор курсу" і тому необхідно зберегти порядок введення, зокрема, кортеж для таблиці "Курси" має бутиведений до введення кортежу "Реєстрація" Взагалі, треба зберігати деякий порядок серед операцій над різними відношеннями, пов'язаними обмеженнями референційної цілісності

Процедури системи БДУЗ

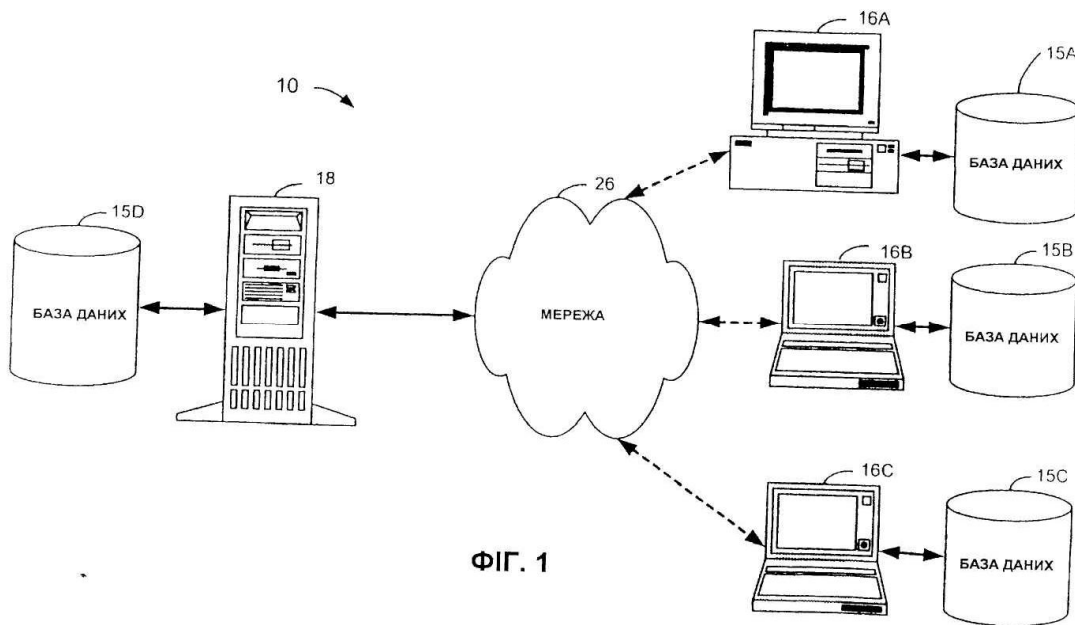
Фіг 5а, 5б, 5с ілюструють процедури, які, згідно з винаходом, використовуються для створення зв'язків комп'ютерної системи 16 клієнта з групами даних бази даних 15d серверу у процесі з'єднання останньої з комп'ютерною системою 16 клієнта Зокрема, описано процес синхронізації баз даних системи 10 БДУЗ

Операцією 502 комп'ютерна система 16 клієнта встановлює зв'язок з сервером 18, який приймає вимогу оновлення від комп'ютерної системи клієнта (опер 504) Далі операцією 512 файли 86 модифікації передаються до комп'ютерної системи 16 клієнта, яка надіслала вимогу Ця комп'ютерна система клієнта починає аналізувати файли 400 модифікації (опер 516), щоб визначити, які дані файлу 86 модифікації мають бути фільтровані, об'єднані, стерті або модифіковані

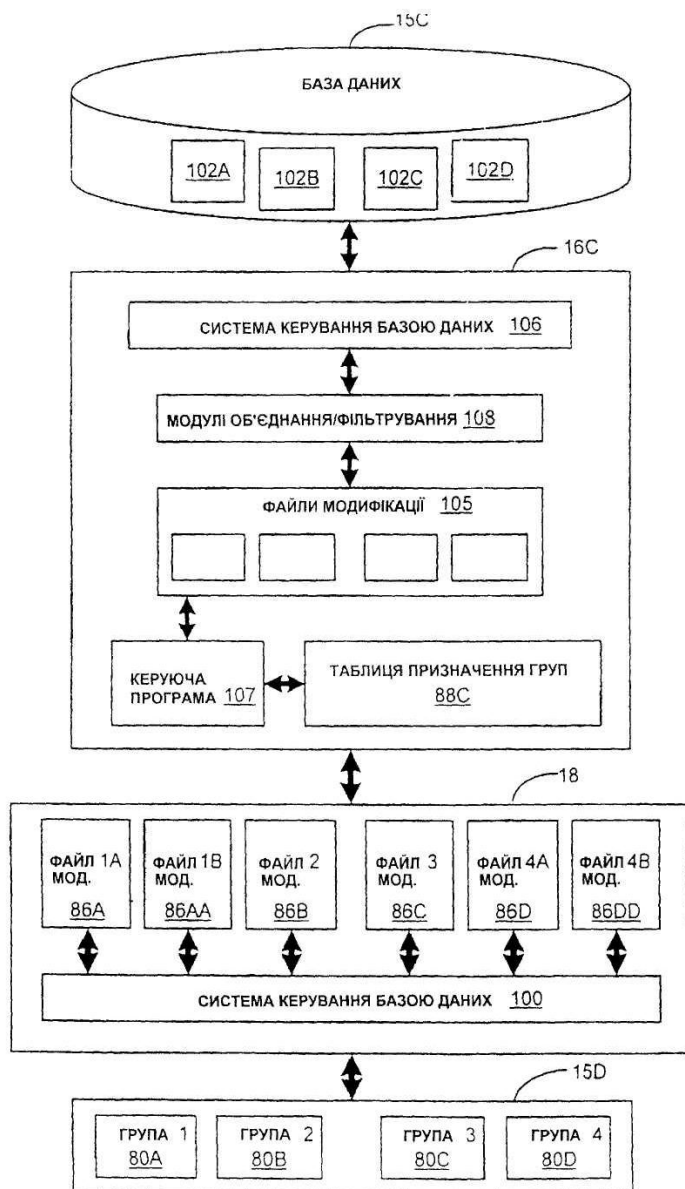
Операцією 518 здійснюється оцінювання ідентифікатора клієнта і файлу модифікації, а потім (опер 520) - визначення, чи здійснила комп'ютерна система 16 клієнта які-небудь транзакції у файлі модифікації Якщо так, операцією 522 відфільтровуються виконані клієнтом транзакції і здійснюється перехід до операції 524 Якщо ні (опер 520), також здійснюється перехід до операції 524, якою комп'ютерна система клієнта, якщо

потрібно, здійснює синонімічне перетворення імен, згадане вище Далі операція 530 визначає, чи має клієнт прийняти усі дані групи, а операцією 532 здійснюється відфільтровування дубльованих даних, після чого здійснюється перехід до операції 540 Якщо клієнт має прийняти усі дані групи (опер 530), також здійснюється перехід до операції 540, якою визначається, чи потребує комп'ютерна система клієнта інформації з інших груп для надійного оновлення локальної бази даних 15с Якщо так, здійснюється перехід до операції 542 одержання необхідної для оновлення інформації від інших груп з переходом до операції 544, яка здійснює фільтрування інтерреляційних даних для виявлення необхідної інформації, як це було описано вище Далі операцією 546 перевіряється наявність дубльованих транзакцій Якщо таких транзакцій нема, здійснюється перехід до операції 556, а якщо вони є, то такі транзакції стираються (опер 554) Операція 556 визначає необхідність об'єднання операцій, виявлених у файлах модифікації і, якщо необхідно, об'єднує файли. Далі операцією 562 оцінюється референційна цілісність файлів модифікації. Процедура завершується операцією 566.

Наведений опис є лише прикладом і не обмежує винаходу, але дає змогу фахівцям, керуючись цим описом і викладеними концепціями винаходу, зробити необхідні модифікації, зміни і поширення у межах об'єму винаходу, визначеного формулою.



ФІГ. 1



ФІГ. 2

¹¹² ¹¹⁴ ¹¹⁶ ¹¹⁸ ¹²⁰ ¹²² ¹²⁴
 СТУДЕНТИ (ІД.СТУДЕНТА, ІМ'Я, ТЕЛ., КЛАС {СТУД.ОСТ.КУРСУ, ВИПУСКНИКИ})
¹³⁴ ¹³⁶ ¹³⁸ ¹⁴⁰
 РЕЄСТРАЦІЯ (ІД.СТУДЕНТА, ІД.КУРСУ, ВИПУСКНИКИ)
¹⁵⁰ ¹⁵² ¹⁵⁴ ¹⁵⁶ ¹⁵⁸ ¹⁶⁰
 КУРС (ІД.КУРСУ, НАЗВА КУРСУ, КЛАС {СТУД.ОСТ.КУРСУ, ВИПУСКНИКИ})

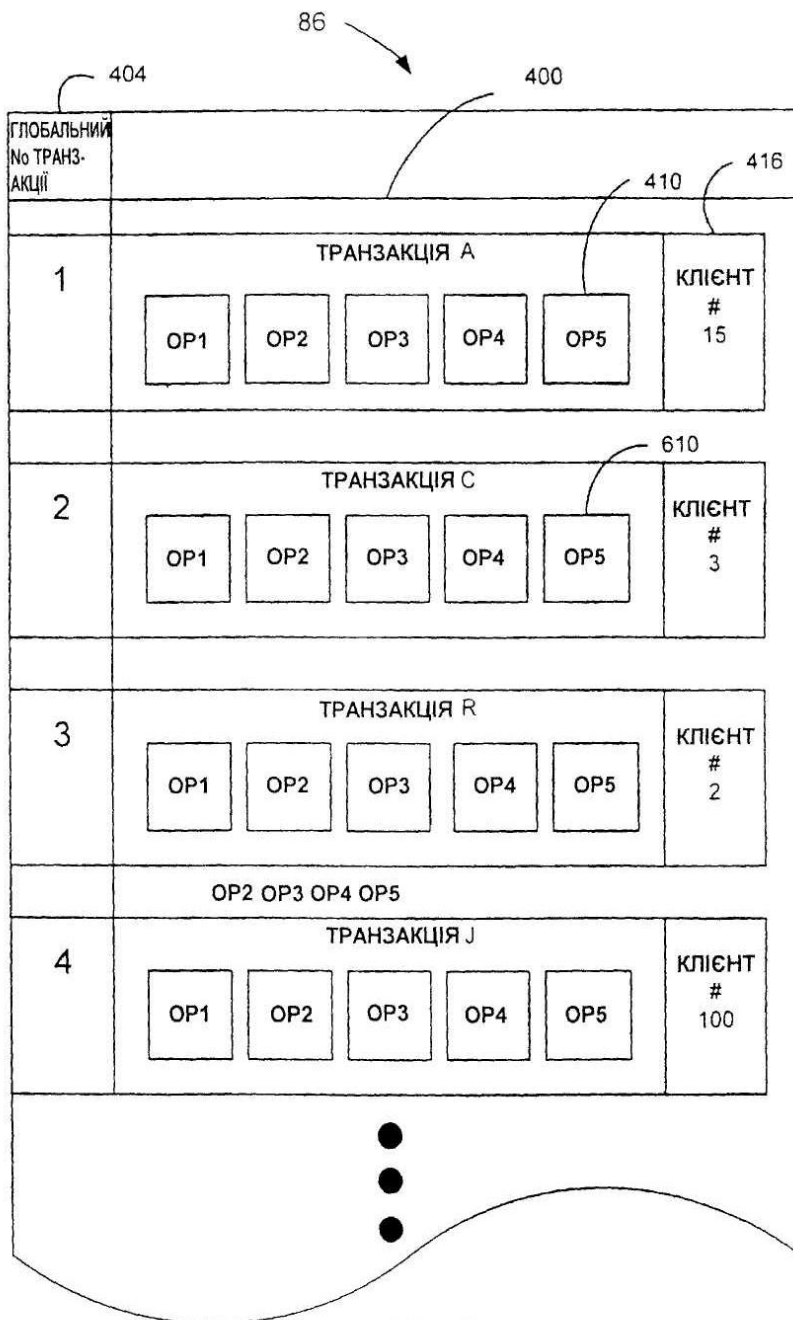
ФІГ. 3А

<u>ГРУПА</u>	<u>ДАНІ ГРУПИ</u>
СТУД.ОСТ.КУРСУ	СТУДЕНТИ (ІД.СТУДЕНТА, ІМ'Я, {СТУД.ОСТ.КУРСУ})
СТУД.ВИПУСКН. ²⁰²	СТУДЕНТИ (ІД.СТУДЕНТА, ІМ'Я, {ВИПУСКНИКИ})
СТУД.ТЕЛ. ²⁰⁶	СТУДЕНТИ (ІД.СТУДЕНТА, ТЕЛ.)
УСІ КУРСИ ²¹⁰	КУРС (УСІ ДАНІ)
УСІ РЕЄСТРАЦІЇ ²¹⁴	РЕЄСТРАЦІЇ (УСІ ДАНІ)
²¹⁸	

ФІГ. 3В

<u>ТИП КЛІЄНТА</u>	<u>ПРИЗНАЧЕНІ ГРУПИ</u>
СТУД.ОСТ.КУРСУ	СТУД.ОСТ.КУРСУ ²⁰²
	УСІ КУРСИ ²¹⁴
	УСІ РЕЄСТРАЦІЇ ²¹⁸
ВИПУСКНИКИ	ВИПУСКНИКИ ²⁰⁶
	УСІ КУРСИ ²¹⁴
	УСІ РЕЄСТРАЦІЇ ²¹⁸
ПРОФЕСОРИ	ВИПУСКНИКИ ²⁰⁶
	СТУД.ОСТ.КУРСУ ²⁰²
	СТУД.ТЕЛ. ²¹⁰
	УСІ КУРСИ ²¹⁴
	УСІ РЕЄСТРАЦІЇ ²¹⁸
РЕЄСТРАЦІЇ	УСІ РЕЄСТРАЦІЇ ²¹⁸

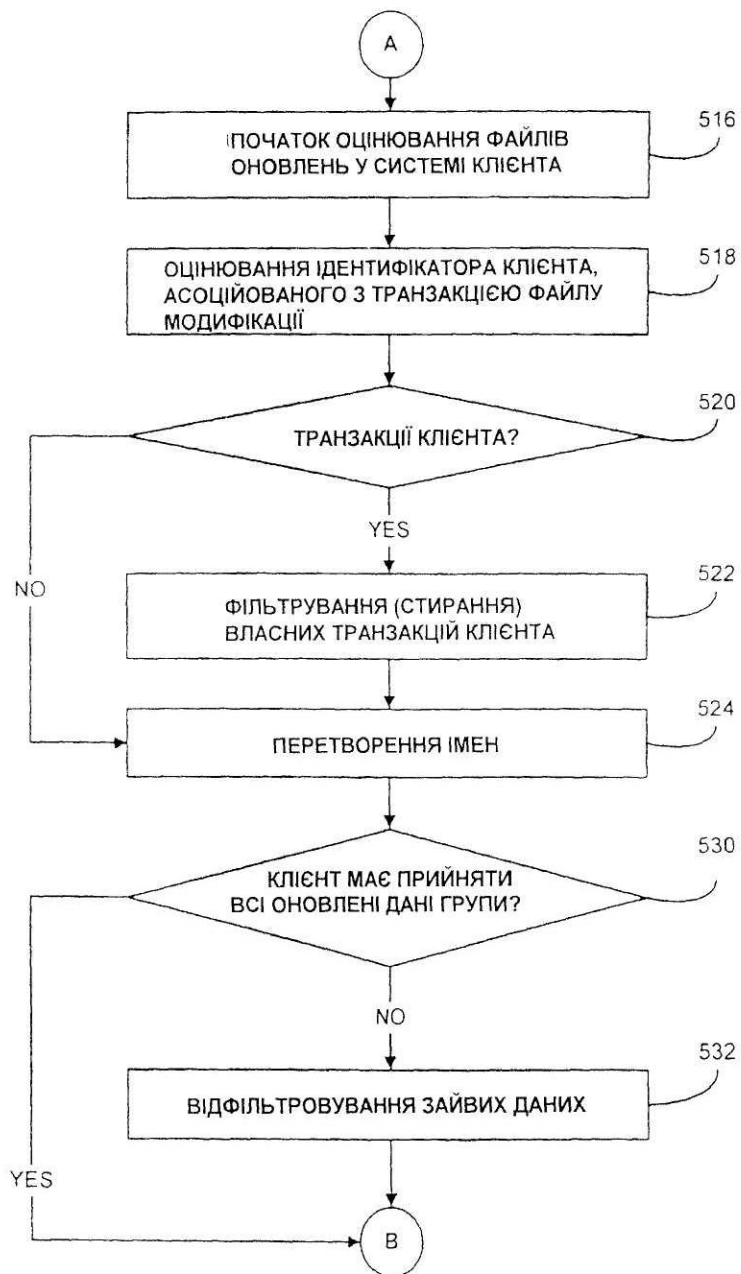
ФІГ. 3С



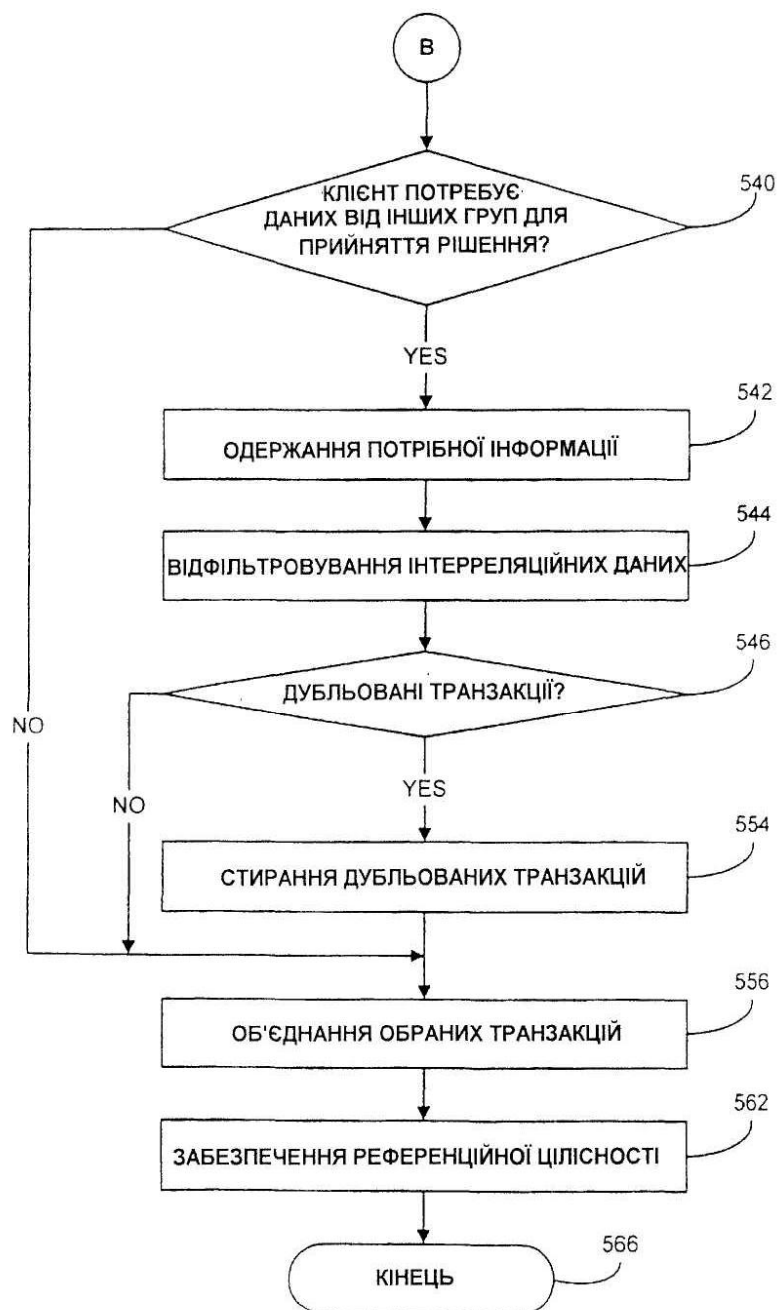
ФІГ. 4



ФІГ. 5а



ФІГ. 5b



ФІГ. 5с