УДК 004.65

Рудаченко Д.Г., магістрант, Попівщий В. І., доцент, науковий керівник

РОЗРОБКА ВБУДОВАНОЇ СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ НА ОСНОВІ XML

Запорізька державна інженерна академія, кафедра ПЗАС

На сьогоднішній день досить багато Desktop-застосунків мають потребу зберігати дані на комп'ютері користувача. Це необхідно у випадках якщо не має стабільного зв'язку з Інтернетом та, як наслідок, з сервером баз даних, якщо швидкість доступу до даних є критичною або за причинами безпеки.

У цих випадках використання «важких» систем керування базами даних (СКБД), таких як Microsoft SQL Server, Oracle SQL, PostgreSQL і т.і. не є виправданим з точки зору продуктивності використання ресурсів ПК та породження зайвих програмних залежностей.

Як наслідок, окремий застосунок може використовувати вбудовану СКБД, кожна з яких має свої недоліки, або зберігати свої дані в у власному форматі та власних файлах тим самим створюючи проблеми при доступі поза межами цього застосунку.

Проблема уніфікації може бути вирішена за допомогою формату XML який використовується в тому числі для серіалізації та збереження складних структур даних [1]. XML також вирішує проблему доступу до даних поза межами за стосунку. .XML- файли відкриваються у більшості редакторах, а формат читається легко. Більшість сучасних мов програмування мають компоненти які дозволяють працювати з XML. Єдиною проблемою яку не вирішує зберігання даних у файлах формату XML є швидкість доступу до даних.

Метою даної роботи є дослідження можливості створення вбудованої СКБД зі зберіганням даних у файлах на основі формату XML та методів забезпечення швидкого доступу до цих даних, а також порівняльний аналіз з готовими рішеннями на ринку вбудованих СКБД.

Для зберігання об'єктів з програми у БД (функції ORM) треба виконувати приведення цих даних (серіалізацію) до типу даних у якому вони будуть зберігатися у БД

Можливі **методи серіалізації**:

1) Внутрішня серіалізація даних C# до XML – має велику *перевагу* у вигляді можливості перевести будь-який об'єкт до XML. *Недоліки* виражається в тому що процедура серіалізації займає багато часу по міркам БД, також цей формат може бути прочитаним тільки з .NET-застосунку та має структуру погано сумісну з БД. При його використанні може буди прискорений тільки пошук самого об'єкта (блоку даних), побудувати зв'язки БД буде досить проблематично.
2) Серіалізація в масив байтів – *переваги* в тому що також може перевести будь який об'єкт до масиву байтів та проводить цю процедуру досить швидко. *Недоліки* – має нечитабельний вигляд, може буду використаним тільки у .NET-застосунках та робить неможливим встановлення зв'язків БД на основі властивостей полів об'єкта [2].
3) Серіалізація на основі полів об'єкта з приведенням типів цих полів до базових типів даних БД, зі зберіганням інформації про цей тип даних, якщо тип поля об'єкта не є простим типом даних, – створення таблиці та серіалізація цього об'єкта окремо, а потім додавання зв'язку між цими об'єктами (таблицями) у вигляді зовнішнього ключа. *Перевагами* цього методу є:незалежність від мови програмування, тому що на основі даних про тип, які зберігаються у XML-файлі, провайдер для кожної мови програмування може приводити цей тип до внутрішнього типу даних цієї мови. Також цей метод дозволяє використовувати не тільки індексацію всього об'єкта, а й окремих полів, що дозволяє створювати

зв'язки між таблицями. Метод забезпечує швидкий доступ до даних поза межами ORM (звичайними запитами до БД). *Недоліками* є необхідність реалізації конвертора типів з мови програмування до типів БД, повільна конвертація з об'єктів, але досить швидка у порівнянні з внутрішньою сериалізаціею C#.

По сукупності переваг було обрано третій метод задля реалізації структури БД та організації роботи функцій ORM, так як він забезпечує найбільш близький до реляційної моделі вигляд даних [3].

З метою прискорення доступу до даних були використані наступні методи [4]:

1. **Хешування** - використано як вказівник на фрагменти файлу БД, задля розрахунку позиції таблиць згідно початку файлу БД та позицій записів згідно початку таблиці.
2. **Індексація** та **Б-дерева –** задля прискорення пошуку по полях таблиці; як гілки дерева вказано зміщення в файлі.

Інформація про індекси та хешування зберігаються як частина XML- файлу БД, але самі індекси та хеші будуть знаходитися у службових файлах задля прискорення доступу до них. Таким чином, навіть у випадку втрати цих файлів, індекси \ хеші можуть буди перебудовані на основі інформації з файлу XML.

У якості тестового застосунку був розроблений провайдер даних у вигляді компонента .NET та тестова програма на мові C#, яка виконує запити різного типу до СКБД та відображує результати продуктивності роботи [5]. Провайдер використовує файли XML для зберігання даних та службові файли (індекси, зв'язки, хешування и т. і.) для забезпечення швидкості доступу до даних. Реалізовано базове забезпечення цілісності даних, ключі та індекси. Також реалізована базова ORM, яка є дуже популярною в сучасних провайдерах даних, та дозволяє зберігати (сериалізувати) та завантажувати з БД C# об'єкти. У випадку, якщо застосунок буде втрачено, БД може бути відновлена лише за рахунок файлу XML. Коли цей файл буде завантажений до провайдера, індекси, зв'язки і таблиці хешування буде відтворено на основі даних цього файлу.

Порівнюючи результати тестування з іншими вбудованими СКБД, можна зробити висновки, що даний підхід до реалізації є дієздатним для використання у Desktop-застосунках, тому що він забезпечує уніфікацію формату, гнучкі можливості сериалізації та досить швидкий доступ до даних.

Висновки:

- вивчені існуючі підходи рішення проблем швидкого доступу до даних;
- оглянуто технології блокування та доступу до даних з різних потоків \ з'єднань;
- досліджено принципи роботи сучасних СКБД;
- оглянуто існуючі реалізації вбудованих СКБД;
- розроблено вбудовану СКБД на основі XML у вигляді .NET-компоненту;
- оглянуто та використано методи сериалізації структур мови C# в формат XML;
- виконано порівняння з іншими вбудованими СКБД, які були розроблені за різними технологіями, виявлено недоліки та переваги.

Література

1. *Марк Грейвс. – Проектирование баз данных на основе XML / Вильямс 2002.*
2. *Рихтер Дж. – CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Питер, 3-е издание 2012.*
3. *Дж. Д. Ульман, Дженнифер Уидом. – Основы реляционных баз даннях / Лори 2006.*
4. *Роберт Вийера. – Программирование баз данных Microsoft SQL Server 2005 для профессионалов / WROX 2008.*
5. *Роберт К. Мартин – Чистый код. Создание, анализ и ре факторинг / Питер 2010.*