

УДК 004.657

## ВЕБ-ІНТЕРФЕЙСИ ДОСТУПУ ДО БАЗ ДАНИХ

О. Столярчук, Р. Шувар, А. Продивус

*Львівський національний університет імені Івана Франка  
вул. Драгоманова, 50, 79005, Львів, Україна  
stolyarchuk@electronics.wups.lviv.ua*

Розглянуто особливості та методику розробки уніфікованих користувацьких веб-інтерфейсів доступу до баз даних з мережі Інтернет. Проаналізовано запити, які користувач може створювати, раціональний вибір та оптимальність елементів керування та форм під час створення користувацького інтерфейсу для роботи з базами даних через протокол НТТР. Проаналізовано також обмеження, які виникають під час проектування веб-інтерфейсу, вимоги до цілісності та несуперечливості даних.

*Ключові слова:* Інтернет, база даних, веб-інтерфейс.

Сьогодні практично в усіх галузях використовують спеціалізовані електронні бази даних. Зазвичай, функціонує кілька баз, незалежних одна від одної. Вони можуть бути як з використанням dbf-орієнтованих засобів розробки, так і на основі оригінальних розробок. Для передавання інформації між ними використовують різноманітні звіти або конвертори. З розвитком локальних мереж, приєднанням їх до мережі Інтернет, створенням внутрішніх корпоративних мереж стало можливим з будь-якого робочого місця отримати доступ до інформаційних ресурсів. У цьому разі виникає потреба в уніфікованому інтерфейсі до різних системи керування базами даних, використання якого не потребувало б наявності різного програмного забезпечення. Однак у випадку спроби використати наявні системи керування базами даних виникають проблеми, пов'язані з сумісністю робочих місць. Кожна з них потребує наявності свого програмного забезпечення на клієнтській машині. Тобто до кожної з цих систем потрібно налаштувати окремий модуль. Це, відповідно, зумовлює сильний трафік у мережі, бо доступ іде напряму до файлів бази даних, завантаження файлового сервера і, як наслідок, часто неможливість віддаленої роботи. Вирішити ці проблеми може використання якогось уніфікованого інтерфейсу для доступу до віддалених баз даних.

Найліпше для цього використовувати протокол НТТР, що зумовлено його універсальністю й підтримкою на різноманітних платформах та операційних системах, можливістю використання в локальних і глобальних мережах. Для використання НТТР не потрібно жодного додаткового програмного забезпечення, крім браузера.

Створення веб-інтерфейсів до баз даних, перетворення даних у гіпертекстові документи дає змогу вирішити проблему сумісності між різними системами керування

базами даних та забезпечити доступ до даних у локальних і глобальних мережах з будь-якого робочого місця.

Вибір конкретних рішень під час розробки веб-інтерфейсів залежить від особливостей самої інформації та потреб роботи з нею. Наприклад, коли інформація оновлюється нечасто, то достатньо згенерувати набір HTML-документів, які б охоплювали всю сукупність даних, і забезпечити між ними взаємне індексування (рис. 1). В такому випадку можна навіть не використовувати ніяких спеціальних систем керування базами даних.

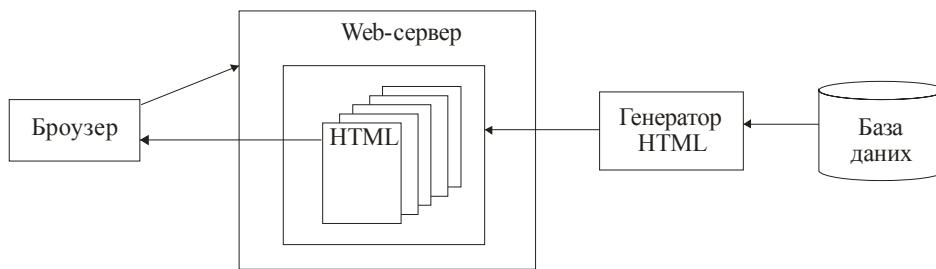


Рис. 1.

Часто буває, що база даних має просту структуру і невеликий розмір. Як приклади, можна назвати список користувачів невеликого вузла, список комп'ютерів у локальній мережі, власну адресну книгу. Все це можна реалізувати за допомогою простих форм баз даних, для збереження яких повністю підійдуть звичайні текстові файли. Такі бази даних можна без проблем використовувати практично на будь-якій комп'ютерній платформі, редагувати за допомогою звичайного текстового редактора, а також роздруковувати на папері без залучення яких-небудь спеціальних ресурсів. Їх легко створювати, а також вносити в них початкові дані та імпортувати в програми електронних таблиць, текстових процесорів або систем керування базами даних (СКБД). Однак у разі розробки веб-інтерфейсів до таких баз потрібно забезпечити коректний процес запису, редагування й видалення даних. У випадку додавання нових записів файл потрібно повністю або частково оновлювати. Тому вставка даних у початок або середину файлу автоматично викликає перезапис наступних даних. І в разі збою в момент перезапису файлу може відбутися часткова або повна втрата даних. Подібні проблеми виникають і під час видалення записів з початку та середини файлу, бо тоді необхідно перезаписати частину файлу на нове місце. Під час пошуку інформації в текстовому файлі доводиться послідовно переглядати файл від початку до кінця. А час виконання цього процесу залежить від розміру текстової бази даних.

Усі наведені вище особливості роботи з текстовими файлами треба враховувати під час розробки веб-інтерфейсів до них. Однак, незважаючи на це, текстові файли можна використовувати, коли бази даних мають невеликий розмір. Виконання операцій додавання, знищення і пошуку записів відбувається швидко і без особливих проблем, якщо розглядати текстовий файл як одновимірний масив.

У випадку складної структури і великих обсягів бази даних найліпшим варіантом є використання СКБД. Під час вибору необхідної СКБД потрібно керуватися потребами задач, для яких їх використовуватимуть. У випадку використання веб-інтерфейсів

доступу найліпшим варіантом є використання СКБД MySQL, оскільки це безкоштовний продукт, високошвидкісний, стійкий і легкий у використанні. Вона є ідеальним рішенням для малих і середніх прикладних програм. Вихідні тексти сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються на серверах із Unix та Unix-подібними операційними системами, де є підтримка багатопотоковості, що дає значний приріст продуктивності.

У разі розробки веб-інтерфесів для перетворення даних у гіпертекстовий вигляд найліпше використовувати мову програмування PHP. Вона дає змогу веб-розробникам швидко створювати динамічні веб-сторінки. Її перевагою перед таким мовами, як Perl і C, є можливість створення HTML документів зі вставками команд PHP. PHP-скрипти виконуються на боці сервера. Ця мова доступна для більшості операційних систем. PHP-включення підтримує більшість сучасних веб-серверів. Використовуючи її, ми отримаємо універсальний продукт, сумісний з різними операційними системами та веб-серверами. Під час розробки можна використовувати як процедурне, так і об'єктно-орієнтоване програмування або їхнє поєднання. PHP, крім того, дає змогу виводити будь-які текстові дані, у тому числі XML-файли. Однією з головних переваг PHP є підтримка нею практично всіх СКБД, у тому числі MySQL, а також її безкоштовність.

Отже, зв'язка PHP-MySQL є найоптимальнішою для написання веб-інтерфесів (рис. 2).

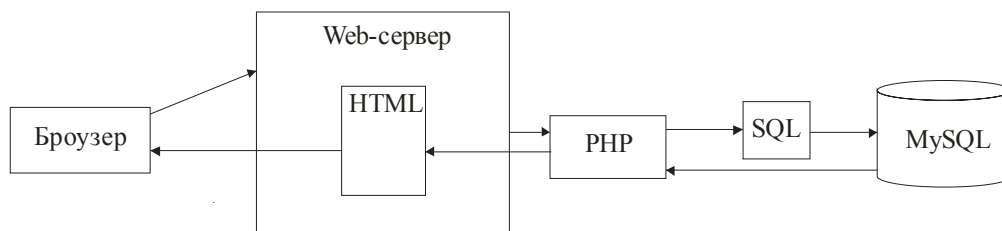


Рис. 2.

Під час розробки користувацького веб-інтерфейсу необхідно звернути увагу на його продуманість, функціональність та ергономічність. Якщо веб-інтерфейс незручний, то ним користуватися не будуть, а намагатимуться знайти інший, можливо, навіть менш досконалий, але зручніший.

Якщо веб-інтерфейс програми є продуманим, функціональним та ергономічним одночасно, то користувач зазвичай цього не помічає, та якщо інтерфейс програми незручний, то це помічають відразу.

Розробляючи веб-інтерфейси доступу до баз даних, треба також врахувати необхідність забезпечити цілісність та несуперечливість даних. Це можна частково реалізувати, використовуючи можливості користувацького інтерфейсу. Для забезпечення цілісності даних інтерфейс перед внесенням будь яких змін у базу даних (командами INSERT, UPDATE, DELETE) повинен виконувати перевірку на допустимість проведення таких змін.

Є два механізми оновлення даних у зв'язаних між собою таблицях. У разі каскадного оновлення даних зі зміною запису в материнській таблиці аналогічно змінюються і записи в дочірніх таблицях, які мають зв'язок із записом з материнської

таблиці. У цьому випадку в разі видалення запису з головної таблиці відбувається видалення всіх зв'язаних записів з дочірніх таблиць. У випадку спроби змінити дані ключового поля в дочірній таблиці зміна допускається лише за умови, що в материнській таблиці існує запис, який відповідає змінюваному в дочірній.

У другому підході до забезпечення оновлення заборонено змінювати дані в головній таблиці, якщо в дочірніх таблицях існують записи, пов'язані з цими даними. Аналогічно не можна витерти запис з материнської таблиці. В обох випадках не можна вставити запис у дочірню таблицю, якщо він не має аналога в материнській.

Цілісність інформації в базі даних можна забезпечити кількома шляхами. Можна створити інтерфейс, який не давав би змоги користувачу проводити запити, які можуть порушити цілісність даних. Наприклад, під час додавання даних в дочірню таблицю треба заборонити пряме введення ключового поля і надати користувачеві змогу вибрати його значення тільки з допустимих величин. Це можна реалізувати, розмістивши на формі об'єкти вибору (SELECT, CHECKBOX, RADIOGROUP), елементами яких будуть значення ключового поля материнської таблиці.

У випадку, коли описаний вище метод використати неможливо, необхідно перед внесенням змін у дочірню таблицю перевірити наявність відповідного ключового значення в материнській і або вивести повідомлення про неможливість внесення змін, або спробувати автоматично створити запис у головній таблиці з потрібним ключовим значенням, у разі потреби запитавши в користувача необхідні додаткові дані.

Під час розробки інтерфейсів до баз даних необхідно також зважати на особливості самої інформації, яка зберігається в базі даних, щоб у ній не містилися взаємовиключні записи.

Веб-інтерфейс повинен забезпечити цілісність і несуперечливість інформації, що зберігається в базі даних. Якщо не забезпечити дотримання цих умов, то в базі даних можливе накопичення непотрібних і недостовірних даних, збої в роботі, і рано чи пізно це може призвести до втрати інформації.

Зазначимо також, що у веб-програміста набагато менший вибір елементів керування, які можна використати для розробки форм, ніж у програміста, який розробляє свої продукти під конкретну операційну систему, йому необхідно враховувати, що користувачі можуть використовувати різні браузері, працювати з різними операційними системами. Особливо це стосується графічного оформлення. Це все викликає значні труднощі та накладає певні обмеження під час розробки зовнішнього інтерфейсу програми.

Створення користувацького веб-інтерфейсу програм, призначених для роботи з базами даних, можна розбити на декілька етапів:

- аналіз запитів, які користувач може створювати, використовуючи цю форму;
- вибір елементів керування форми;
- створення самої форми і розміщення на ній елементів керування.

Форми веб-інтерфейсу доцільно будувати так, щоб за їхньою допомогою можна було виконувати переважну більшість завдань.

Для оптимізації роботи з віддаленими базами даних необхідно максимально забезпечити правильність даних, уведених користувачем. Це потрібно, щоб не пересилати по мережі наперед некоректні дані (наприклад, коли користувач не ввів даних у всі обов'язкові поля). Тому є сенс за допомогою JavaScript організувати перевірки на допустимість уведених даних перед відсиленням їх на сервер, де

відбувається вже детальніший аналіз на коректність уведених даних. Це зменшить завантаженість каналу зв'язку і підвищить швидкість роботи.

Як зазначено вище, для забезпечення цілісності даних під час їхнього введення в дочірні таблиці краще користуватися елементами, які дають змогу вибрати потрібні значення, аніж дозволяти користувачу самому вводити ці значення, тим більше, що такий варіант заповнення форми є зручнішим для самого користувача, йому не потрібно нічого згадувати, боятися помилок набору тощо.

Одним із найвідповідальніших етапів створення веб-інтерфейсу є вибір правильних розмірів форм, їхніх кольорів, зручне розміщення елементів на них. Не варто створювати велику кількість елементів керування на одній формі. Заповнення великої кількості елементів керування є незручним і спричинюватиме втому користувача. У цьому разі також збільшується ймовірність того, що якісь параметри можуть бути не встановлені, а оскільки процес введення у випадку однієї форми однокроковий, то проконтролювати введені дані користувач не має змоги, і неправильна інформація відразу записується в базу даних. Враховуючи сказане, краще розбивати процес введення на кілька кроків.

Запропоновану методику застосовано під час розробки різноманітних систем доступу до баз даних для навчального процесу на факультеті електроніки Львівського національного університету імені Івана Франка, зокрема, роботу системи «Розклад» можна переглянути в мережі Інтернет за адресою <http://www.electronics.wups.lviv.ua/rozk/>.

1. *Ульман. Л.* MySQL; [пер. с англ. А. Слинкин]. – М.: ДМК Пресс, СПб: Питер, 2004. – 352 с.
2. *Райордан Р.* Основы реляционных баз данных; [пер. з англ.]. – М.: Издательско-торговый дом «Русская Редакция», 2001.
3. Методи та алгоритми моделювання Web-систем / А.М. Пелещин // Вісн. Держ. ун-ту «Львівська політехніка». – 2000, № 406. – С.199-211.
4. *Кузнецов С.* От баз данных к пространствам данных: новая абстракция управления информацией. – 2006, [http://www.citforum.ru/database/articles/from\\_db\\_to\\_ds](http://www.citforum.ru/database/articles/from_db_to_ds)
5. *Дрюэк К.* (Katherine Drewek). Хранилища данных: сходство и различия подходов Билла Инмона и Ральфа Кимболла. – 2005, <http://www.b-eye-network.com/view/743>.
6. *Пасічник В.В., Шаховська Н.Б.* Сховища даних. Навч. посібник. – Львів, Магнолія, 2006, 2007.

**WEB-INTERFACES FOR ACCESS TO DATABASES****O. Stolyarchuk, R. Shuwar, A. Prodyvus**

*Ivan Franko National University of L'viv,  
Dragomanova Str. 50, UA-79005, Lviv, Ukraine.  
stolyarchuk@electronics.wups.lviv.ua*

Features and development method of unified user access interfaces to the databases from the Internet are considered. Queries, which an user can create, rational choice and optimum of managing elements and forms at creation of user interface for working with the databases using HTTP protocol are analysed. Limitations, which appear at planning of web-interface, requirements for integrity and uncontradictions of the data are also analysed.

*Key words:* Internet, database, web-interface.

**ВЕБ-ИНТЕРФЕЙСЫ ДОСТУПА К БАЗАМ ДАННЫХ****О. Столярчук, Р. Шувар, А. Прудивус**

*Львовский национальный университет имени Ивана Франко,  
ул. Драгоманова, 50, 79005, Львов, Украина.  
stolyarchuk@electronics.wups.lviv.ua*

Рассмотрены особенности и методика разработки унифицированных пользовательских интерфейсов доступа к базам данных из сети Интернет. Проанализированы запросы, которые пользователь может создавать, рациональный выбор и оптимальность управляющих элементов и форм при создании пользовательского интерфейса для работы с базами данных по протоколу HTTP. Проанализированы также ограничения, которые появляются при проектировании веб-интерфейса, требования к целостности и непротиворечивость данных.

*Ключевые слова:* Интернет, база данных, веб-интерфейс.

Стаття надійшла до редколегії 16.05.2011

Стаття прийнята до друку 31.05.2011