

- Область пошуку є дискретною множиною певних наборів конструктивних та теплофізичних параметрів МЕР.
- Складність алгоритму повного перебору визначається об'ємом дискретної області пошуку і може бути зменшена заданням напрямку пошуку.
- Напрямок пошуку задається базою типових рішень конструкції, які сформовані як висновки про роль елементів конструкції в процесі тепловідведення.

3. Зважаючи на недоліки побудованого алгоритму конструктивно-параметричного пошуку (ресурсомісткість, присутність людського чинника), запропоновано для розв'язування задачі забезпечення заданого температурного режиму застосування нейромережових технологій.

4. Для розв'язування задачі забезпечення заданого теплового режиму в конструкціях МЕР можна використати властивість нейронної мережі навчатися на розв'язках прямих задач теплового проектування. Це дає ряд переваг: здатність нейромережі до абстрагування дає змогу створювати образи, з якими вона ніколи не зустрічалася; зменшення людського фактору в процесі теплового проектування; високу точність вихідних даних, оскільки навчання здійснюється на реальних даних, і точність контролюється вибором моделі і алгоритмів нейромережі.

1. Федасюк Д.В. *Методи та засоби теплового проектування мікроелектронних пристроїв*. – Львів, 1999. 2. Батищев Д. И. *Методы оптимального проектирования*. – М., 1984. 3. Федасюк Д., Левус Є. *Комп'ютерне моделювання та забезпечення температурних режимів мікроелектронних пристроїв з кристалами встановленими на жорсткі виводи // Матеріали Міжнародної конференції з індуктивного моделювання (МКІМ2002)*. – Львів. – 2002. – Т. 3. – С.335–341. 4. Федасюк Д.В. *Аналіз параметричної чутливості температурних полів у конструкціях мікроелектронних пристроїв // Вісн. Державного університету “Львівська політехніка”*. – №313. – 1996. – С.3–8. 5. Осовский С. *Нейронные сети для обработки информации / Пер. с польского И.Д. Рудинского*. – М., 2004. 6. Bernhard Pichler and Herbert A. Mang. *Parameter Identification for Sophisticated Material Models by Means of Iterative Back Analyses Based on Soft Computing / European Conference on Computational Mechanics*. – Cracow (Poland). – June 26–29, 2001. –P.1–20. 7. Fedasyuk D., Petrov D., Levus E. *Web-based thermal simulator WEBTAFС /Комп'ютерні технології друкарства – № 6. – 2001. – С. 146–152*.

УДК 004.413

П.І. Жежнич

Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж

## ОРГАНІЗАЦІЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ ЗА ДОПОМОГОЮ XML

© Жежнич П.І., 2005

**Визначено правила описання XML-документів, які містять реляційні дані. Описано правила визначення DTD (Document Type Definition) та структури XML-документа, якщо DTD не використовується.**

**This paper considers rules of XML-documents description that contain relational data. Rules of DTD definition and XML-document structure are described.**

**Постановка проблеми.** Аналіз сучасних інформаційних технологій показує, що за останні десять років найстрімкіше розвинулося глобальне середовище Internet та відповідні програмні, апаратні та технологічні засоби. Практично всі підприємства чи установи, які здійснюють інформатизацію своєї діяльності, намагаються належно подати себе у World Wide Web. Це призводить до того, що інформація, яка зберігається в межах корпоративних інформаційних систем,

у певний спосіб повинна відобразитися у відповідному Web-сайті. Існує декілька способів такого відображення:

- Заповнення інформації на статичному Web-сайті на основі даних корпоративної інформаційної системи. Це вимагає ручного втручання у Web-сторінки. Якщо Web-сайт є невеликим (наприклад, десятків сторінок), то такий спосіб може бути допустимим. Якщо ж сторінок є багато (наприклад, десятки чи сотні), то такий спосіб виявиться занадто трудомістким і неприйнятним.

- Заповнення інформації на динамічному Web-сайті на основі даних корпоративної інформаційної системи. У такому разі корпоративні дані автоматизовано або напівавтоматизовано переводять на Web-сайт. Це переведення відбувається незалежно від функціонування корпоративної інформаційної системи та Web-сайту.

- Заповнення інформації на динамічному Web-сайті шляхом його інтеграції з корпоративною інформаційною системою. Цей спосіб найлегший щодо публікації корпоративних даних у World Wide Web, але він є найнебезпечніший з погляду захисту та безпеки даних.

Здебільшого найприйнятнішим є другий спосіб публікації корпоративних даних, оскільки він є безпечним з погляду захисту корпоративної інформаційної системи (вона безпосередньо не з'єднана з Web-сайтом), він не є трудомістким як перший варіант, і цей спосіб є надійним, оскільки робота Web-сайту не залежить від функціонування корпоративної інформаційної системи, а сам процес публікації є повністю автономним.

З іншого боку, розвиваються не тільки технології публікації чи відображення інформації у World Wide Web, а також і програмно-апаратні засоби перегляду цієї інформації. Усе частіше доступ до інформації у мережі Internet здійснюється не тільки за допомогою комп'ютерів, але й за допомогою, наприклад, мобільних телефонів чи кишенькових комп'ютерів.

Отже, організація публікації даних у World Wide Web повинна передбачати не тільки якісне перенесення даних у Web-середовище, але й належне їхнє відображення на апаратних засобах кінцевих користувачів. Особливо це стосується табличних даних, які часто містяться на динамічних сайтах (наприклад: прайс-листи, каталоги тощо).

Звичайно, використання комп'ютерів з достатньо великими моніторами не приводить до особливих труднощів при відображенні табличних даних, на відміну від мобільних телефонів чи кишенькових комп'ютерів з малим екраном.

Розглянемо деякі підходи до передавання табличних даних (у Web-середовищі) за допомогою стандарту XML.

**Зв'язок висвітленої проблеми із науковими завданнями.** Метою статті є застосування стандарту XML для передавання реляційних даних.

**Наукова новизна** статті полягає в побудові XML-орієнтованої мови для організації реляційних баз даних.

**Практична цінність** статті визначається можливістю передавання реляційних (табличних) даних у Web-середовищі незалежно від комунікаційних пристроїв.

**Аналіз останніх досліджень.** XML-орієнтованою називається база даних (БД), яка як модель даних використовує XML-модель даних (тобто дані зберігаються в XML-документах). Необхідно відрізнити XML-орієнтовані бази даних (див. рис.1) від реляційних баз даних, що підтримують обмін даними мовою XML (див. рис. 2).

Необхідно відзначити, що в основі баз даних, які підтримують обмін даними на XML, лежить реляційна модель. Сьогодні відомі дві XML-орієнтовані БД – Tamino (Software AG) і Cache (InterSystems). Прикладом реляційних БД, які підтримують обмін даними на XML, є Oracle, MS SQL-Server та ін [4].

У цій статті ми будемо вважати, що дані зберігаються в реляційних БД, а на їхній основі будують XML-документи, які є засобом перенесення, тиражування даних.

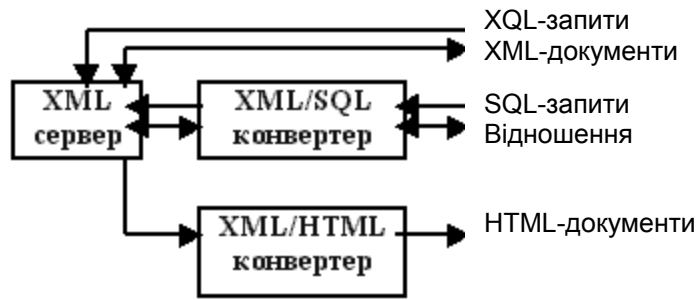


Рис. 1. XML-орієнтовані БД

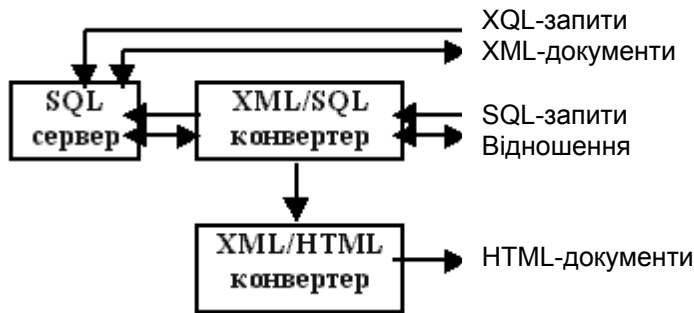


Рис. 2. Реляційні БД, які підтримують обмін даними на XML

**Виділення невирішених частин проблеми.** Переведення даних з реляційного вигляду в XML синтаксично не є однозначним [1]. Для ілюстрації розглянемо простий приклад, що складається з 3 відношень, 5 атрибутів і 5 кортежів (див. рис.3).

Tab1		Tab2	
A	C	B	D
a1	c1	b1	d1
		b2	d2

Tab3		
A	B	E
a1	b1	e1
a1	b2	e2

Рис. 3. Приклад реляційних даних

В найпростішому і найкомпактнішому варіанті можна побудувати таку конструкцію (варіант 1):

```
<DataBase>
<Tab1 A="a1 C="c1"/>
<Tab2 B="b1 D="d1"/>
<Tab2 B="b2 D="d2"/>
<Tab3 A="a1" B="b1 E="e1"/>
<Tab3 A="a1" B="b2 E="e2"/>
</DataBase>
```

Її недоліком є неоднорідність представлення кортежів і значень, що, наприклад, ускладнює відображення розширених реляційних моделей. Якщо значення оформляти також у вигляді тегів, то отримаємо таке (варіант 2):

```
<DataBase>
<Tab1><A>a1</A><C>c1</C></Tab1>
```

```

<Tab2><B>b1</B><D>d1</D></Tab2>
<Tab2><B>b2</B><D>d2</D></Tab2>
<Tab3><A>a1</A><B>b1</B><E>e1</E></Tab3>
<Tab3><A>a1</A><B>b2</B><E>e2</E></Tab3>
</DataBase>

```

Обидва варіанти використовують дворівневу вкладеність XML-вузлів, за допомогою якої встановлюються направлені зв'язки кортеж–значення.

Направлені зв'язки від записів Tab1 до записів Tab3 і від записів Tab2 до записів Tab3 вказуються однаковими значеннями ключових атрибутів A і B. Мовою XML зв'язки звичайно вказуються явно вкладенням тегів один в одного і застосуванням посилань. Це дає змогу в нашому прикладі прибрати зовнішні ключі в Tab3 і встановити посилання на одного батька вкладенням тега Tab3 в Tab2 і на другого батька (Tab1) за допомогою атрибутів Id і Ref (варіант 3):

```

<DataBase>
<Tab1 Id="#1"><A>a1</A><C>c1</C></Tab1>
<Tab2><B>b1</B><D>d1</D><Tab3 Ref1="#1"><E>e1</E></Tab3></Tab2>
<Tab2><B>b2</B><D>d2</D><Tab3 Ref1="#1"><E>e2</E></Tab3></Tab2>
</DataBase>

```

Для того, щоб виконати зворотну операцію – привести довільні XML-дані до реляційних – їх необхідно перетворити до одного з описаних вище варіантів.

**Постановка задачі.** Основним нашим завданням є визначення таких правил переведення реляційних даних в XML-формат, що дало б змогу однозначно інтерпретувати XML-дані при переведенні їх у реляційний вигляд. Для реалізації цієї задачі необхідно визначити правила описання відповідних XML-документів, а саме:

- Визначити правильне оформлення реляційних даних за допомогою DTD (Document Type Definition).
- Визначити структуру XML-документа, якщо DTD не використовується.

Другий крок є важливим, оскільки XML-документ може існувати і без DTD, тобто його використання не повинно повністю залежати від DTD.

**Основний матеріал.** Основна ідея цієї статті полягає в тому, що реляційні дані повинні оформлятися в XML-документі разом з їхньою структурою. Тобто логічно XML-документ повинен складатися з двох частин:

- описання структури даних.
- дані.

З погляду реляційної моделі даних це логічно, оскільки кожна таблиця (реляція) характеризується своєю схемою (структурою) та вмістом (множиною кортежів) [2]. Відповідно DTD повинно передбачати такий поділ XML-документа. Для прикладу розглянемо таблицю Клієнт, в якій зберігається інформація про клієнтів деякого санаторію.

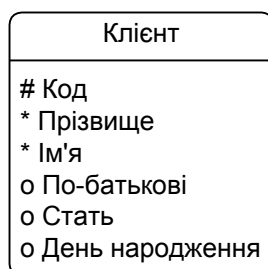


Рис. 4. Схема таблиці Клієнт

Тоді DTD матиме такий вигляд (файл client.dtd):

```
<?xml encoding="win-1251"?>
<!ELEMENT database(table)+>
  <!ELEMENT table(name,scheme,data)>
    <!ELEMENT scheme(field,constraint)+>
      <!ELEMENT field(name,type,length,pk,uk,notnull,fk,default)>
        <!ELEMENT name(#PCDATA)>
        <!ELEMENT type(#PCDATA)>
        <!ELEMENT length(#PCDATA)>
        <!ELEMENT pk(#PCDATA)>
        <!ELEMENT uk(#PCDATA)>
        <!ELEMENT notnull(#PCDATA)>
        <!ELEMENT fk(#PCDATA)>
        <!ELEMENT default(#PCDATA)>
      <!ELEMENT constraint(name,fieldlist,pk,uk,notnull,fk)>
        <!ELEMENT name(#PCDATA)>
        <!ELEMENT fieldlist(#PCDATA)>
        <!ELEMENT pk(#PCDATA)>
        <!ELEMENT uk(#PCDATA)>
        <!ELEMENT notnull(#PCDATA)>
        <!ELEMENT fk(#PCDATA)>
    <!ELEMENT data(row)+>
      <!ELEMENT row(id,surname,name,fathername,gender,birthday)>
        <!ELEMENT id(#PCDATA)>
        <!ELEMENT surname(#PCDATA)>
        <!ELEMENT name(#PCDATA)>
        <!ELEMENT fathername(#PCDATA)>
        <!ELEMENT gender(#PCDATA)>
        <!ELEMENT birthday(#PCDATA)>
```

Відповідно до цього визначення кожна таблиця XML-документа матиме таку структуру.

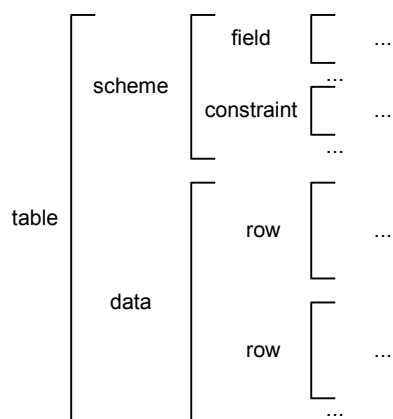


Рис. 5. Структура таблиці в XML-документі

А таблиця Клієнт у XML-документі матиме такий вигляд (файл client.xml).

```
<?xml encoding="win-1251"?>
<!--DOCTYPE client SYSTEM "client.dtd"-->
<database>
  <table>
    <name>client</name>
    <scheme>
      <field>
        <name>id</name><type>VARCHAR</type><length>50</length>
```

```

    <pk>TRUE</pk>
</field>
<field>
    <name>surname</name><type>VARCHAR</type><length>200</length>
    <nn>TRUE</nn>
</field>
<field>
    <name>name</name><type>VARCHAR</type><length>200</length>
    <nn>TRUE</nn>
</field>
<field>
    <name>fathername</name><type>VARCHAR</type>
    <length>200</length>
</field>
<field>
    <name>gender</name><type>VARCHAR</type><length>50</length>
    <nn>TRUE</nn><default>M</default>
</field>
<field>
    <name>birthday</name><type>DATE</type>
</field>
</scheme>
<data>
    <row>...</row>
    <row>...</row>
    ...
</data>
</table>
</database>

```

Очевидно, в цьому XML-документі тег <scheme> відповідає за описання структури таблиці client, а тег <data> - за вміст цієї таблиці.

Такий XML-документ є самодостатній (добто для коректної його інтерпретації не обов'язково мати DTD). Дані з цього документа достатньо очевидно переводяться в таблицю бази даних. Крім того, такий документ може використовуватися у побудові відповідних XML-запитів [3].

**Висновки.** Організація публікації даних у World Wide Web повинна передбачати не тільки якісне перенесення даних у Web-середовище, але й належне їхнє відображення на апаратних засобах кінцевих користувачів. Враховуючи різноманіття пристроїв, за допомогою яких кінцеві користувачі переглядають інформацію у WWW, XML виступає єдиним засобом, який дає змогу оптимізувати такий перегляд. Особливо це стосується табличних даних, які часто містяться на динамічних сайтах. У такому разі XML документ відіграє роль носія даних, а спосіб відображення залежить від програмно-апаратного засобу перегляду. Отже, задача передачі табличних даних у Web-середовищі за допомогою стандарту XML є актуальною.

У цій статті визначено правила описання XML-документів, які містять реляційні дані. Для цього описано правила визначення DTD (Document Type Definition) та структури XML-документа, якщо DTD не використовується.

1. Веселов В.В. Долженков А.Н. XML и технологии БАЗ ДАННЫХ //Сайт информационных технологий, 2001. <http://www.semenov.webservis.ru/links/internet/xml/ar1.html>. 2. Мейер Д. Теория реляционных баз данных. – М., 1987. 3. Cohen S., Kanza Y., Sagiv Y. SQL4X: A Flexible Query Language for XML and Relational Databases. //The Hebrew University of Jerusalem, Institute of Computer Science, 2001. <http://citeseer.ist.psu.edu/635551.html>. 4. Fernandez M., Morishima A., Suci D., Tan W.-C. Publishing Relational Data in XML: the SilkRoute Approach. //IEEE Data Engineering Bulletin, 2001. <http://citeseer.ist.psu.edu/fernandez01publishing.html>.