

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №3

Мова XML та її використання в СКБД PostgreSQL

Виконав:

Ст. Прізвище Ім'я

Група

Тема: Вивчення мови XML та її використання в СКБД PostgreSQL .

Мета роботи: Ознайомлення з конструкціями мови XML та її використанням в СКБД PostgreSQL, зокрема, зі створенням XML даних та XML документів, перетворенням таблиць реляційної бази в XML-документ.

Теоретичний матеріал:

XML або *Розширювана мова розмітки* - це мова розмітки, яка визначає набір правил для кодування документів у форматі, який зручний для читання і створення документів як людиною, так і машиною. Мова називається розширюваною, бо нею не фіксується розмітка (теги), яка використовується в документах: розробник може створювати розмітку відповідно до потреб конкретної предметної області, аби лише вона відповідала синтаксичними правилами мови.

XML це текстовий формат даних із потужною підтримкою через Unicode для різних людських мов. Хоча дизайн XML зосереджений на документах, мова широко використовується для подання довільних структур даних, таких як ті, що використовуються у веб-сервісах.

Вона була створена W3C¹ для подолання обмежень HTML (яка, по суті, спочатку була просто набором тегів) - мови розмітки гіпертексту, яка є основою для всіх веб-сторінок. Проблема з HTML полягає в тому, що його теги говорять браузеру, як *відобразити* цю інформацію, вони не вказують браузеру, *що це за інформація*

Як і HTML, XML заснована на SGML - стандартній узагальненій мові розмітки. HTML до версії 4 включно був додатком SGML, а XML - це підмножина SGML, розроблена для спрощення процесу машинного розбору документа. Хоча SGML використовується у видавничій галузі впродовж десятиліть (ще з кінця 60-х рр.), її складність сприйняття злякала багатьох, що хотіли використати її (SGML також розшифровується як "Sounds great, maybe later" - Звучить чудово, можливо, пізніше ☺). І, що важливо, XML був розроблений з урахуванням Інтернету.

Отже, XML - це програмно- та апаратно- незалежний інструмент для зберігання та транспортування даних.

- XML розшифровується як розширювана мова розмітки
- XML - мова розмітки, схожа на HTML
- XML був розроблений для зберігання та транспортування даних
- XML був розроблений так, щоб він був самоописовим
- XML - це рекомендація W3C

На XML можна створювати текст і розмічати його за допомогою тегів, перетворюючи кожне слово, речення або фрагмент в ідентифіковану відсортовану інформацію. Створювані вами файли або екземпляри документа, складаються з тегів і тексту, причому тегам в XML ви можете присвоїти деяке значення, що допоможе правильно розуміти документ при читанні на папері або навіть обробляти його в електронному вигляді. Наприклад, ви можете легко витягти

¹ (World Wide Web Consortium - це міжнародна спільнота, де організації-члени, штатний персонал та громадськість працюють разом над розробкою веб-стандартів під керівництвом веб-винахідника та директора Тіма Бернерса-Лі)

поштовий індекс з документа, просто розмістивши вміст, оточений `<postal-code>` та `</postal-code>` тегами, технічно відомим як `<postal-code>` елемент .

Чим більше описових тегів, тим більше частин документа можна ідентифікувати. З перших днів існування розмітки одне з її переваг полягає в тому, що в разі втрати комп'ютерної системи, роздруковані дані все одно залишаються читабельними завдяки тегами.

У XML можна створювати свої власні теги, що дозволяє точно представляти фрагменти даних. Документи можна не просто розділяти на абзаци і заголовки, а й виділяти будь-які фрагменти всередині документа. Щоб це було ефективно, потрібно визначити кінцевий перелік своїх елементів і дотримуватися його. Елементи можна визначати в Описі типу документа (Document Type Definition - DTD) або в схемі.

Існують три поширені терміни, що використовуються для опису частин документа XML: *теги* , *елементи* та *атрибути* . Ось зразок документа, який ілюструє терміни:

```
<address>
  <name>
    <title>Mrs.</title>
    <first-name>
      Mary
    </first-name>
    <last-name>
      McGoon
    </last-name>
  </name>
  <street>
    1401 Main Street
  </street>
  <city state="NC">Anytown</city>
  <postal-code>
    34829
  </postal-code>
</address>
```

- Тег - це текст між лівою кутовою дужкою (`<`) та правою кутовою дужкою (`>`). Існують початкові теги (наприклад `<name>`) та кінцеві теги (наприклад `</name>`)
- Елементом є початковий тег, закінчуючий тег і все, що знаходиться між ними. У наведеному вище прикладі, `<name>` елемент містить три дочірніх елемента: `<title>` , `<first-name>` і `<last-name>` .
- Атрибут - пара ім'я="значення" всередині *початкового* тегу елемента. У цьому прикладі `state` є атрибут `<city>` елемента.

Основними поняттями XML є *коректність* (*well formed*) та *валідність* (*valid*).

- **Коректний документ** відповідає всім синтаксичним правилам XML. Документ, що не є коректним, не може називатись XML-документом. Сумісний синтаксичний аналізатор ² (conforming parser) не повинен обробляти такі документи.

Коректний XML документ повинен мати:

- лише один кореневий елемент;
 - непорожні елементи розмічено початковим та кінцевим тегам. Порожні елементи можуть помічатись «закритим» тегом, наприклад <IAmEmpty/>. Така пара еквівалентна <IAmEmpty></IAmEmpty>;
 - Один елемент не може мати декілька атрибутів з однаковим іменем. Значення атрибутів знаходяться або в одинарних ('), або у подвійних (") лапках;
 - Теги можуть бути вкладені, але не можуть перекриватись. Кожен некореневий елемент мусить повністю знаходитись в іншому елементі;
 - Документ має складатися тільки з правильно закодованих дозволених символів множини Юнікоду. Єдиними кодуваннями, які обов'язково має розуміти XML-процесор, є UTF-16 та UTF-8. Фактичне та задеклароване кодування (character encoding) документа мають збігатись. Кодування може бути задекларовано ззовні, як у заголовку «Content-Type» при передачі по протоколу HTTP, або в самому документі використанням явної розмітки на самому початку документа. У разі відсутності інформації про кодування, документ має бути в кодуванні UTF-8 (або його підмножині ASCII).
- Документ називається **валідним**, якщо він є коректним, містить посилання на граматичні правила та повністю відповідає обмеженням, вказаним у цих правилах (DTD або XML Schema).

Переваги застосування XML в Інтернеті:

- *XML спрощує обмін даними.* Оскільки різні організації (або навіть різні частини однієї організації) рідко стандартизуються на одному наборі інструментів, це може створити значний обсяг роботи для обміну даними між програмами. Використовуючи XML, кожна група створює єдину утиліту, яка перетворює їх внутрішні формати даних у XML і навпаки. Можливо, що їхні постачальники програмного забезпечення вже надають інструменти для перетворення своїх записів баз даних (або каталогів LDAP, або замовлень на купівлю тощо) у та з XML.
- *XML дозволяє створювати смарт-код (розумний код).* Оскільки документи XML можуть бути структуровані так, щоб ідентифікувати кожен важливу інформацію (а також взаємозв'язки між частинами), можна написати код, який може обробляти ці XML-документи без втручання людини. Той факт, що виробники програмного забезпечення витратили величезну кількість

² Синтаксичним аналізатором називається програма або компонент, що читає XML-документ, проводить синтаксичний аналіз, та відтворює його структуру. Якщо синтаксичний аналізатор перевіряє документ на валідність, то такий аналізатор називають валідатором.

часу та грошей на створення інструментів розробки XML, означає написання цього коду - досить простий процес.

- XML дозволяє інтелектуальний пошук. Незважаючи на те, що пошукові системи постійно вдосконалювалися протягом багатьох років, але все ж досить часто трапляються помилкові результати пошуку. Якщо ви шукаєте HTML-сторінки на когось з назвою "Chip", ви також можете знайти сторінки з шоколадними чіпсами, комп'ютерними чіпами, деревними трісками та безліччю інших марних збігів. Пошук XML-документів за <first-name> елементами, які містять текст Chip, дасть вам набагато кращий набір результатів.

Посилання для ознайомлення з теоретичним матеріалом теми «Мова XML»:

- Doug Tidwell, Introduction to XML:
<https://www.ibm.com/developerworks/xml/tutorials/xmlintro/xmlintro.html>
- Beginning XML, 5-edition © 2012 Joe Fawcett, Liam R.E. Quin, Danny Ayers
- XML Tutorial W3C: <https://www.w3schools.com/xml/default.asp>
- документація PostgreSQL: <https://www.postgresql.org/docs/>
- Тип XML в PostgreSQL: <https://postgrespro.ru/docs/postgresql/11/datatype-xml>
- XML-функції PostgreSQL: <https://postgrespro.ru/docs/postgresql/11/functions-xml>

Перелік розділів та понять, з якими необхідно ознайомитись для виконання завдання лабораторної роботи (за документацією до PostgreSQL v.11):

Вступ до XML (Doug Tidwell)

Тип XML в PostgreSQL

Створення XML-значень

Обробка кодування

Звернення до XML-значенням

XML-функції PostgreSQL

Створення XML-контенту

Умови з XML

Обробка XML

Відображення таблиць в XML

Синтаксичний валідатор

Хід роботи

1. Опрацювати теоретичний матеріал.
2. Власноруч створити XML документ (схеми DTD чи XML створювати не потрібно), який містить інформацію та структуру не менше трьох зв'язаних таблиць реляційної бази даних (можна використовувати Select з функціями створення XML-контенту: xmlelement, xmlroot, xmlforest та інші).
3. Використати on-line синтаксичний валідатор для перевірки коректності створеного XML документа.

4. Оформити звіт про виконання лабораторної роботи, який має містити:
 - титульну сторінку;
 - тему, мету та завдання лабораторної роботи;
 - короткий перелік та опис створення XML документа;
 - навести скріни екрану з кодом XML документа та результатом перевірки його синтаксичним валідатором.
5. Завантажити на Google Диск в папку “Група’ Бази даних” звіт у вигляді файлу з іменем: Прізвище_ЛР2.pdf .