ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА
Факультет прикладної математики та інформатики

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №11
«Мова XML та її використання в СКБД PostgreSQL»

Виконав: Гуменюк Станіслав

Ст. Групи ПМІ-33с

Оцінка ___

Прийняв:

2024

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.

Завдання: Власноруч створити XML документ (схеми DTD чи XML створювати не потрібно), який містить інформацію з 3-х чи 4-х зв'язаних таблиць реляційної бази даних. Для створення документа не використовувати функцій PostgreSQL відображення таблиць table_to_xml чи всієї бази даних database_to_xml в XML. Натомість, використати SELECT з функціями створення XM-Lконтенту: xmlelement, xmlroot, xmlforest, xmlagg та ін.

Run on active connection | Select block

```sql
-- Запит для створення XML документу з даними про замовлення, клієнтів, продукти та категорії
SELECT xmlelement(name "supermarket_data",
    xmlattributes('1.0' AS version),
    (SELECT xmlagg(xmlelement(name "order",
        xmlattributes(co.id AS order_id),
        xmlelement(name "date", co.operation_time),
        (SELECT xmlelement(name "customer",
            xmlattributes(c.person_id AS customer_id),
            xmlelement(name "name", p.name),
            xmlelement(name "surname", p.surname),
            xmlelement(name "discount", c.discount),
            (SELECT xmlagg(xmlelement(name "contact",
                xmlattributes(ct.name AS type),
                pc.contact_value
            ))
            FROM person_contact pc
            JOIN contact_type ct ON pc.contact_type_id = ct.id
            WHERE pc.person_id = c.person_id)
        )
        FROM customer c
        JOIN person p ON c.person_id = p.id
        WHERE c.person_id = co.customer_id),
        xmlelement(name "order_details",
            (SELECT xmlagg(xmlelement(name "product",
                xmlattributes(od.product_id AS product_id),
                xmlelement(name "title", pt.title),
                xmlelement(name "category", pc.name),
                xmlelement(name "manufacturer", m.name),
                xmlelement(name "price", od.price),
                xmlelement(name "discount_price", od.price_with_discount),
                xmlelement(name "quantity", od.product_amount)
            ))
            FROM order_details od
            JOIN product p ON od.product_id = p.id
            JOIN product_title pt ON p.product_title_id = pt.id
            JOIN product_category pc ON pt.product_category_id = pc.id
            JOIN manufacturer m ON p.manufacturer_id = m.id
            WHERE od.customer_order_id = co.id)
        ),
        xmlelement(name "total_amount",
            (SELECT SUM(od.price_with_discount * od.product_amount)
             FROM order_details od
             WHERE od.customer_order_id = co.id)
        )
    ))
    FROM customer_order co
    WHERE co.operation_time >= CURRENT_DATE - INTERVAL '30 days'
    )
) AS supermarket_xml;
```

```xml
<supermarket_data version="1.0">
  <order order_id="1">
    <date>2024-09-17</date>
    <customer customer_id="1">
      <name>John</name>
      <surname>Doe</surname>
      <discount>0.05</discount>
      <contact type="Phone">555-1234</contact>
      <contact type="Email">john.doe@email.com</contact>
    </customer>
    <order_details>
      <product product_id="4">
        <title>Milk</title>
        <category>Dairy</category>
        <manufacturer>DairyBest</manufacturer>
        <price>2.5</price>
        <discount_price>2.375</discount_price>
        <quantity>1</quantity>
      </product>
      <product product_id="1">
        <title>Apple</title>
        <category>Fruits</category>
        <manufacturer>FreshFarms</manufacturer>
        <price>0.5</price>
        <discount_price>0.475</discount_price>
        <quantity>5</quantity>
      </product>
    </order_details>
    <total_amount>4.749999970197678</total_amount>
  </order>
  <order order_id="2">
    <date>2024-09-17</date>
    <customer customer_id="2">
      <name>Jane</name>
      <surname>Smith</surname>
      <discount>0.1</discount>
      <contact type="Phone">555-5678 </contact>
      <contact type="Email">jane.smith@email.com</contact>
    </customer>
    <order_details>
      <product product_id="5">
        <title>Bread</title>
        <category>Bakery</category>
        <manufacturer>BakerDelight</manufacturer>
        <price>1.5</price>
        <discount_price>1.35</discount_price>
        <quantity>2</quantity>
      </product>
      <product product_id="2">
        <title>Banana</title>
```

XML data to validate

```xml
      <title>Apple                                    </title>
      <category>Fruits                                   </category>
      <manufacturer>FreshFarms                                       </manufacturer>
      <price>0.5</price>
      <discount_price>0.475</discount_price>
      <quantity>5</quantity>
    </product>
  </order_details>
  <total_amount>7.124999970197678</total_amount>
 </order>
</supermarket_data>
```

Validate

Document Valid

SQL запит створює XML документ, який містить інформацію з таблиць бази даних супермаркету:

1. customer_order
2. customer
3. person
4. person_contact
5. contact_type
6. order_details
7. product
8. product_title
9. product_category
10. manufacturer

Структура створеного XML:

1. Кореневий елемент **<supermarket_data>** з атрибутом версії.
2. Для кожного замовлення (за останні 30 днів) створюється елемент **<order>**:
   - Атрибут **order_id**
   - Елемент **<date>** з датою замовлення
   - Елемент **<customer>** з інформацією про клієнта:
     - Атрибут customer_id
     - Ім'я та прізвище клієнта
     - Знижка клієнта
     - Контактна інформація клієнта (телефон, email тощо)
   - Елемент **<order_details>** з інформацією про кожен продукт у замовленні:
     - ID продукту
     - Назва продукту
     - Категорія продукту
     - Виробник
     - Ціна (звичайна та зі знижкою)
     - Кількість
   - Елемент **<total_amount>** із загальною сумою замовлення

Цей запит використовує наступні функції для створення XML:

- **xmlelement**: для створення XML елементів
- **xmlattributes**: для додавання атрибутів до XML елементів
- **xmlagg**: для агрегації кількох XML елементів в один список
- **xmlroot**: неявно використовується для створення кореневого елементу

Щоб отримати результат, ви можете виконати цей запит у вашому PostgreSQL клієнті. Результатом буде один рядок з колонкою `supermarket_xml`, яка містить згенерований XML документ.

Цей XML документ надає комплексний огляд замовлень у вашому супермаркеті, включаючи деталі про клієнтів та продукти. Ви можете модифікувати цей запит, щоб включити додаткову інформацію або змінити структуру XML відповідно до ваших потреб.