

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

**Бази даних та інформаційні системи**

**ЛАБОРАТОРНА РОБОТА №13**

**Тема: «XML-схема XML-документа».**

Виконала:

*Ст. Пелещак Вероніка*

ПМІ-35с

**Тема:** «XML схема XML документа».

**Мета роботи:** Ознайомлення з конструкціями XML Schema XML документа. Отримати практичні навички її створення і використання.

**Завдання:** Використовуючи альтернативний метод створити XML Schema XML документа, який містить інформацію та структуру двох-трьох зв'язаних таблиць реляційної бази даних. Обов'язково використати обмеження (restriction або extension) та границі (facets). Використати on-line валідатор для перевірки валідності створеного XML документа разом з XML Schema.

### **Контрольні питання:**

#### **1. Які є два способи визначення структури та елементів в XML документі?**

Існує два основних способи визначення структури XML-документа:

- **DTD (Document Type Definition)** – описує структуру XML-документа, визначає дозволені елементи, їх вкладеність та атрибути. Не підтримує типи даних.
- **XML Schema (XSD — XML Schema Definition)** – сучасний спосіб визначення структури XML, який дозволяє задавати типи даних, обмеження, простори імен та наслідування типів.

#### **2. Як визначаються прості та складні типи елементів?**

- **Прості типи (simple types)** – елементи, що містять лише текстові дані і не мають вкладених елементів чи атрибутів.
- **Складні типи (complex types)** – елементи, які можуть містити інші елементи, атрибути або обидва одночасно.

### 3. Які особливості визначення атрибутів?

Атрибути визначаються всередині складних типів (complexType) за допомогою елемента `<xs:attribute>`. Основні особливості:

- Мають ім'я та тип даних.
- Можуть бути **обов'язковими** (use="required") або **необов'язковими** (use="optional").
- Не можуть містити вкладених елементів.

### 4. Які є обмеження (facets) для елемента XML або атрибута?

Обмеження (facets) використовуються для уточнення значень типів даних у XSD. Найпоширеніші види обмежень:

- **length, minLength, maxLength** — обмеження довжини рядка;
- **minInclusive, maxInclusive** — мінімальне та максимальне допустиме значення;
- **pattern** — визначає регулярний вираз, якому має відповідати значення;
- **enumeration** — перелік допустимих значень.

### Хід роботи:

1. Створила XML схему, використовуючи XML-документ з лабораторно роботи №11. (Так як XML документ достатньо великий, у звіті наведено тільки XML Schema).

```

3 / veronikapesak / Desktop / practice / data_base_2 / bug_tracking(xml schema).xsd
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <!-- Обмеження для priority -->
    <xs:simpleType name="PriorityType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Low"/>
            <xs:enumeration value="Medium"/>
            <xs:enumeration value="High"/>
            <xs:enumeration value="Critical"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- Обмеження для status -->
    <xs:simpleType name="StatusType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="New"/>
            <xs:enumeration value="In Progress"/>
            <xs:enumeration value="Fixed"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- Обмеження для role -->
    <xs:simpleType name="RoleType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Tester"/>
            <xs:enumeration value="Manager"/>
            <xs:enumeration value="Developer"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- Обмеження для email -->
    <xs:simpleType name="EmailType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}" />
        </xs:restriction>
    </xs:simpleType>

```

```

<!-- Тип користувача -->
<xs:complexType name="UserType">
    <xs:sequence>
        <xs:element name="first_name" type="xs:string"/>
        <xs:element name="last_name" type="xs:string"/>
        <xs:element name="email" type="EmailType"/>
        <xs:element name="role" type="RoleType"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="CorrectedByType">
    <xs:sequence>
        <xs:element name="first_name" type="xs:string"/>
        <xs:element name="last_name" type="xs:string"/>
        <xs:element name="email" type="EmailType"/>
    </xs:sequence>
</xs:complexType>

<!-- Тип для correction -->
<xs:complexType name="CorrectionType">
    <xs:sequence>
        <xs:element name="comment" type="xs:string"/>
        <xs:element name="corrected_by" type="CorrectedByType"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:integer" use="required"/>
    <xs:attribute name="date" type="xs:date" use="required"/>
</xs:complexType>

<!-- Історія змін -->
<xs:complexType name="HistoryCorrectionsType">
    <xs:sequence>
        <xs:element name="correction" type="CorrectionType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

```

```

<!-- Тип для bug -->
<xs:complexType name="BugType">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="priority" type="PriorityType"/>
    <xs:element name="status" type="StatusType"/>
    <xs:element name="module" type="xs:string"/>
    <xs:element name="description" type="xs:string"/>
    <xs:element name="created_by" type="UserType"/>
    <xs:element name="assigned_to" type="UserType"/>
    <xs:element name="history_corrections" type="HistoryCorrectionsType"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:integer" use="required"/>
  <xs:attribute name="date" type="xs:date" use="required"/>
</xs:complexType>

<!-- Тип для программного обеспечения -->
<xs:complexType name="SoftwareType">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="version" type="xs:string"/>
    <xs:element name="category" type="xs:string"/>
    <xs:element name="bug" type="BugType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:integer" use="required"/>
</xs:complexType>

<!-- Корневый элемент -->
<xs:element name="bug_tracking_system">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="software" type="SoftwareType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

## XML schema (XSD) data

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3
4   <!-- Омежения для priority -->
5   <xs:simpleType name="PriorityType">
6     <xs:restriction base="xs:string">
7       <xs:enumeration value="Low"/>
8       <xs:enumeration value="Medium"/>
9       <xs:enumeration value="High"/>
10      <xs:enumeration value="Critical"/>
11    </xs:restriction>
12  </xs:simpleType>
13

```

☒ **Data Security Warning** - Tick this box to confirm you understand all data is stored on our servers for processing and may be retained for service improvement.

Validate

If you have sensitive data, please [Download the Free Liquid Studio Community Edition](#) and use the tools directly from your desktop.

Document Valid

2. Визначено комплексні типи для користувачів (**UserType**), багів (**BugType**), програмного забезпечення (**SoftwareType**) та інших пов'язаних елементів. Було використано обмеження **restriction** і границі **facets**, а саме **enumeration** і **pattern**.
3. XML документ було перевірено на відповідність схемі XSD за допомогою онлайн-валідатора. XML документ пройшов валідацію без помилок.

**Висновок:** У ході роботи було створено XML-документ та відповідну XML Schema (XSD), яка описує його структуру та обмеження. Схема успішно пройшла перевірку на коректність, що підтвердило правильність побудови моделі даних.