TEMA XIS

Virlan Adrian, 341A3

1. Tema

* Nr. 29: Evidenta comenzilor catre furnizorii de marfa pentru un market.

1. Definirea domeniului ales

* Domeniul ales se concentreaza pe gestionarea comenzilor date de un

supermarket furinozrilor sai. Scopul aplicatiei va fi deci gestionarea,

monitorizarea si eficientizarea comenzilor.

1. Constrangeri

* Pot exista 0 sau mai multe comenzi.
* comanda are 1 sau mai multe produse.
* Toate campurile trebuie sa contina date relevante si complete
* Data si valabilitatea vor contine subcampuri de zi([0, 28/30/31]), luna[1, 12], an[2024, inf].
* Pretul comenzii este calculat ca suma preturilor produselor.
* Pretul produselor este calculat ca produsul dintre cantitate si pret unitar.
* Cantitatea produselor este int, > 0.
* Pretul unitar este float > 0.
* Data nu poate fi in viitor(as dori sa se ia automat data curenta).
* Valabilitatea nu poate fi in trecut sau prezent.
* Numar lot, id comanda/produs este un numar int > 0.
* Este\_fragil si este\_necesar\_refrigerare sunt de tip boolean.

1. Dcoumentul XML

Structura Generală:

* Declarația XML: <?xml version="1.0" encoding="UTF-8"?> indică versiunea și codificarea documentului.
* Referința la documentul XSL: <?xml-stylesheet type="text/xsl" href="supply\_stream.xsl"?> face legătura cu un fișier XSL pentru stilizare.
* Definiția Spațiului de Nume: xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" definește spațiul de nume XML Schema Instance.
* Schema de Validare: xsi:noNamespaceSchemaLocation="supply\_stream.xsd" specifică locația fișierului schema XML.
* Schema de validare strucutra: <!DOCTYPE supply\_stream SYSTEM "supply\_stream.dtd">, referinta catre documentul de validare DTD

Elemente și Atribute:

* Elementul Rădăcină: <**supply**\_**stream**>
  + Conține toate elementele **order**.
* Elementul **order**
  + Atribut **id**: Identificator unic pentru fiecare comandă.
  + Conține elemente pentru detalii de bază ale comenzii:
  + **date**: Data comenzii.
  + **supplier\_name**: Numele furnizorului.
  + **order\_price**: Prețul total al comenzii.
    - Atribut **currency**: Moneda utilizată pentru preț.
* Elementul **date**
  + Elemente Copil:
    - **day**: Ziua.
    - **month**: Luna.
    - **year**: Anul.
* Elementul **products**
  + Conține unul sau mai multe elemente **product**.
* Elementul **product**
  + Atribut **id**: Identificator unic pentru fiecare produs dintr-o comandă.
  + Conține elemente pentru detalii ale produsului:
    - **category**: Categoria produsului.
    - **product**\_**name**: Numele produsului.
    - **manufacturer**\_**name**: Numele producătorului.
    - **quantity**: Cantitatea comandată.
    - **unit**\_**price**: Prețul unitar.
* Elementul **additional\_details**
  + Conține detalii suplimentare despre produs:
    - **lot\_number**: Numărul lotului.
    - **expiry\_date**: Data de expirare.
    - Elemente Copil:
      * **day**: Ziua.
      * **month**: Luna.
      * **year**: Anul.
    - **is**\_**fragile**: Indică dacă produsul este fragil (true/false).
    - **requires**\_**refrigeration**: Indică dacă produsul necesită refrigerare (true/false).

1. Documentul XML Schema

* Tipul Simplu cu Restricție: currency

Definește un tip de date simplu care restricționează valorile posibile la trei monede: USD, EUR și RON.

Restricție (SimpleType currency): Acesta este un tip simplu restricționat care limitează valorile permise pentru moneda (currency) la "USD", "EUR" și "RON". Aceasta asigură că doar valorile definite sunt utilizate în documentul XML, prevenind erori de introducere a datelor.

* Tipul Complex cu Extensie: PriceWithCurrency

Definește un tip complex care extinde tipul xs:decimal pentru a include un atribut currency de tip currency.

Extensie (ComplexType PriceWithCurrency): Acesta este un tip complex extins din xs:decimal care include un atribut suplimentar currency de tip currency. Extinderea permite includerea de atribute suplimentare unui tip de bază, păstrând în același timp validarea și integritatea datelor.

* Au fost adaugate si restrictii pentru campurile zi, luna si an. In a acest fel, se respecta conditiile setate in 2.Constrangeri

Structura supply\_stream:

* Element supply\_stream
  + Descriere: Element rădăcină.
  + Tip de Date: xs:complexType (conține secvență de elemente order)
* Element order
  + Descriere: Reprezintă o comandă.
  + Tip de Date: xs:complexType (conține secvență de elemente)
  + Atribute:
    - * id de tip xs:integer
  + Elemente Copil:
    - * date, supplier\_name, order\_price, products
* Element date
  + Descriere: Reprezintă data comenzii.
  + Tip de Date: xs:complexType (conține secvență de elemente)
  + Elemente Copil:
    - * day de tip xs:integer
      * month de tip xs:integer
      * year de tip xs:integer
* Element supplier\_name
  + Descriere: Numele furnizorului.
  + Tip de Date: xs:string
* Element order\_price
  + Descriere: Prețul comenzii.
  + Tip de Date: PriceWithCurrency (extensie de xs:decimal)
* Element products
  + Descriere: Lista de produse.
  + Tip de Date: xs:complexType (conține secvență de elemente product)
* Element product
  + Descriere: Detalii despre produs.
  + Tip de Date: xs:complexType (conține secvență de elemente)
  + Atribute:
  + id de tip xs:integer
  + Elemente Copil:
    - category de tip xs:string
    - product\_name de tip xs:string
    - manufacturer\_name de tip xs:string
    - quantity de tip xs:integer
    - unit\_price de tip xs:decimal
    - additional\_details
* Element additional\_details
  + Descriere: Detalii suplimentare despre produs.
  + Tip de Date: xs:complexType (conține secvență de elemente)
  + Elemente Copil:
    - lot\_number de tip xs:integer
    - expiry\_date de tip xs:complexType (conține secvență de elemente)
    - is\_fragile de tip xs:boolean
    - requires\_refrigeration de tip xs:boolean
* Element expiry\_date
  + Descriere: Data de expirare a produsului.
  + Tip de Date: xs:complexType (conține secvență de elemente)
  + Elemente Copil:
    - day de tip xs:integer
    - month de tip xs:integer
    - year de tip xs:integer

1. Documentul DTD

Documentul DTD (Document Type Definition) supply\_stream.dtd definește structura și regulile pentru validarea unui document XML care gestionează informațiile legate de fluxul de aprovizionare. Acesta specifică elementele, atributele și relațiile dintre ele pentru a asigura conformitatea și consistența datelor.

Documentul DTD va asigura ca strucutra xml va avea urmatoarea forma arborescenta:

supply\_stream

|\_ order

| |\_ date

| | |\_ day

| | |\_ month

| | |\_ year

| |\_ supplier\_name

| |\_ order\_price

| |\_ products

| |\_ product

| |\_ category

| |\_ product\_name

| |\_ manufacturer\_name

| |\_ quantity

| |\_ unit\_price

| |\_ additional\_details

| |\_ lot\_number

| |\_ expiry\_date

| | |\_ day

| | |\_ month

| | |\_ year

| |\_ is\_fragile

| |\_ requires\_refrigeration

De asemenea, exista 0 sau mai multe noduri order, iar fiecare order are 1 sau mai multe produse.

1. Implementarea aplicatiei

* Funcția load\_file

Scop: Permite utilizatorului să selecteze și să încarce un fișier XML sau JSON.

Proces:

* + Deschide un dialog pentru selectarea fișierului dorit.
  + Citeste conținutul fișierului și determină tipul de fișier încărcat (XML sau JSON).
  + Direcționează conținutul fișierului către funcția de parsare corespunzătoare (parse\_xml sau parse\_json).
* Funcția parse\_xml

Scop: Parsarea documentului XML

Proces:

-Se foloseste metoda fromstring(din biblioteca xml.etree.ElementTree) pentru a se gasi root-ul structurii XML

-Avand acest root, se itereaza print totate elementele order. Folsind metoda .find(*tag*) se selecteaza tagurile copil, iar cu metoda .get(*atribut*) se selecteaza atributele nodului curent,

-Se procedeaza similar si pentru nodurile produse(care contin mai multe subnoduri)(se itereaza prin fiecare produs).

-Toate valorile se salveaza in variabile, pentru a fi ulterior afisate pe ecran

* Funcția parse\_json

Scop: Parsarea documentului JSON.

Proces:

-Se utilizează funcția loads din biblioteca json pentru a încărca și parsa conținutul fișierului JSON într-o structură de date Python (de obicei un dicționar sau listă).

-Se navighează în structura dicționarului pentru a accesa lista de comenzi sub cheia order.

-Pentru fiecare element order, se extrag informațiile dorite

-Se accesează direct valorile cheilor corespunzătoare, cum ar fi id, date, supplier\_name, order\_price.

* Functia search\_order & search\_order\_json

Aceste 2 functii au rolul de a returna comenzile care au un produs intridus de utilizator de la tastatura. Principiul de parcurgere este acelasi cu cel descris mai sus. Diferenta este ca exista mereu un if la parcurgerea produselor. In momentul in care numele produsului este acelasi cu numele intridus de utilizator, se salveaza date, se comuta un flag in true si se face break la bucla for. Se va afisa pe ecran doar doatele comenzii si produslui selectate.