Лабораториска вежба - 223260

**ЗАДАЧА 1**

Предложете решение како би ги споиле дадените датотеки во една датотека System.xml, така што би овозможиле брз и едноставен пристап до информациите за Клиентите и Артистите.

<?xml version="1.0" encoding="UTF-8"?>  
<SYSTEM>  
 <ARTISTS>  
 <ARTIST ID="1" type="band">  
 <NAME>Metallica</NAME>  
 <COUNTRY>USA</COUNTRY>  
 <GENRE>Metal</GENRE>  
 </ARTIST>  
 <ARTIST ID="2" type="band">  
 <NAME>Disturbed</NAME>  
 <COUNTRY>USA</COUNTRY>  
 <GENRE>Metal</GENRE>  
 </ARTIST>  
 <ARTIST ID="3" type="pop\_group">  
 <NAME>The Backstreet Boys</NAME>  
 <COUNTRY>USA</COUNTRY>  
 <GENRE>Pop</GENRE>  
 </ARTIST>  
 <ARTIST ID="16" type="dj">  
 <YEAR\_STARTED\_PERFORMING>2009</YEAR\_STARTED\_PERFORMING>  
 <NET\_WORTH>25,000,000</NET\_WORTH>  
 <YEAR\_OF\_BIRTH>1991</YEAR\_OF\_BIRTH>  
 </ARTIST>  
 <ARTIST ID="17" type="dj">  
 <YEAR\_STARTED\_PERFORMING>1980</YEAR\_STARTED\_PERFORMING>  
 <NET\_WORTH>75,000,000</NET\_WORTH>  
 <YEAR\_OF\_BIRTH>1967</YEAR\_OF\_BIRTH>  
 </ARTIST>  
 <ARTIST ID="18" type="dj">  
 <YEAR\_STARTED\_PERFORMING>2013</YEAR\_STARTED\_PERFORMING>  
 <YEAR\_OF\_BIRTH>1987</YEAR\_OF\_BIRTH>  
 </ARTIST>  
 <ARTIST ID="13" type="singer">  
 <YEAR\_STARTED\_PERFORMING>1988</YEAR\_STARTED\_PERFORMING>  
 <YEAR\_OF\_BIRTH>1972</YEAR\_OF\_BIRTH>  
 </ARTIST>  
 <ARTIST ID="14" type="singer">  
 <YEAR\_STARTED\_PERFORMING>1987</YEAR\_STARTED\_PERFORMING>  
 <YEAR\_OF\_BIRTH>1971</YEAR\_OF\_BIRTH>  
 </ARTIST>  
 <ARTIST ID="15" type="singer">  
 <YEAR\_STARTED\_PERFORMING>1988</YEAR\_STARTED\_PERFORMING>  
 </ARTIST>  
 </ARTISTS>  
 <GROUPS>  
 <GROUP ID="1">  
 <YEAR\_FORMED>1981</YEAR\_FORMED>  
 <NUMBER\_OF\_MEMBERS>9</NUMBER\_OF\_MEMBERS>  
 </GROUP>  
 <GROUP ID="2">  
 <YEAR\_FORMED>1994</YEAR\_FORMED>  
 <NUMBER\_OF\_MEMBERS>6</NUMBER\_OF\_MEMBERS>  
 </GROUP>  
 <GROUP ID="3">  
 <YEAR\_FORMED>1993</YEAR\_FORMED>  
 <NUMBER\_OF\_MEMBERS>10</NUMBER\_OF\_MEMBERS>  
 </GROUP>  
 </GROUPS>  
 <ALBUMS>  
 <ALBUM ID="1" ARTIST\_ID="1">  
 <NAME>Master Of Pupets</NAME>  
 <RELEASE\_YEAR>1986</RELEASE\_YEAR>  
 <PRICE>$10.00</PRICE>  
 </ALBUM>  
 <ALBUM ID="2" ARTIST\_ID="1">  
 <NAME>Metallica</NAME>  
 <RELEASE\_YEAR>1991</RELEASE\_YEAR>  
 <PRICE>$12.99</PRICE>  
 </ALBUM>  
 <ALBUM ID="3" ARTIST\_ID="2">  
 <NAME>Believe</NAME>  
 <RELEASE\_YEAR>2002</RELEASE\_YEAR>  
 <PRICE>$8.00</PRICE>  
 </ALBUM>  
 <ALBUM ID="4" ARTIST\_ID="2">  
 <NAME>Evolution</NAME>  
 <RELEASE\_YEAR>2018</RELEASE\_YEAR>  
 <PRICE>$15.00</PRICE>  
 </ALBUM>  
 <ALBUM ID="5" ARTIST\_ID="2">  
 <NAME>Immortalized</NAME>  
 <RELEASE\_YEAR>2015</RELEASE\_YEAR>  
 <PRICE>$16.00</PRICE>  
 </ALBUM>  
 <ALBUM ID="6" ARTIST\_ID="2">  
 <NAME>Indestructible</NAME>  
 <RELEASE\_YEAR>2008</RELEASE\_YEAR>  
 <PRICE>$14.00</PRICE>  
 </ALBUM>  
 </ALBUMS>  
 <CATALOG>  
 <CD ID="1" ALBUM\_ID="1">  
 <STATE>functional</STATE>  
 <OCCUPIED>0</OCCUPIED>  
 </CD>  
 <CD ID="2" ALBUM\_ID="1">  
 <STATE>0</STATE>  
 <OCCUPIED>1</OCCUPIED>  
 </CD>  
 <CD ID="3" ALBUM\_ID="1">  
 <STATE>2</STATE>  
 <OCCUPIED>free</OCCUPIED>  
 </CD>  
 <CD ID="4" ALBUM\_ID="2">  
 <STATE>0</STATE>  
 <OCCUPIED>0</OCCUPIED>  
 </CD>  
 <CD ID="5" ALBUM\_ID="2">  
 <STATE>0</STATE>  
 <OCCUPIED>0</OCCUPIED>  
 </CD>  
 </CATALOG>  
 <CLIENTS>  
 <CLIENT ID="1">  
 <NAME>Stephen</NAME>  
 <SURNAME>Sims</SURNAME>  
 <ADDRESS>Hollywood St.75</ADDRESS>  
 <EMAIL>stephen.sims@example.com</EMAIL>  
 <PHONE\_NUMBER>754-1234</PHONE\_NUMBER>  
 </CLIENT>  
 <CLIENT ID="2">  
 <BASIC\_INFO>  
 <NAME>Barney</NAME>  
 <SURNAME>Stinson</SURNAME>  
 <ADDRESS>Some St.75</ADDRESS>  
 </BASIC\_INFO>  
 <EMAIL>barney.stinson@example.com</EMAIL>  
 <PHONE\_NUMBER>755-1111</PHONE\_NUMBER>  
 </CLIENT>  
 <CLIENT ID="3">  
 <BASIC\_INFO>  
 <NAME>Walter</NAME>  
 <SURNAME>White</SURNAME>  
 <ADDRESS>Other St.55</ADDRESS>  
 </BASIC\_INFO>  
 <EMAIL>walter.white@example.com</EMAIL>  
 </CLIENT>  
 <CLIENT ID="4">  
 <NAME>Ned</NAME>  
 <SURNAME>Stark</SURNAME>  
 <ADDRESS>Second St.3</ADDRESS>  
 <EMAIL>ned.stark@starks.com</EMAIL>  
 </CLIENT>  
 <CLIENT ID="5">  
 <NAME>John</NAME>  
 <SURNAME>Snow</SURNAME>  
 <ADDRESS>Third St.2</ADDRESS>  
 <EMAIL>john.snow@almoststark.com</EMAIL>  
 <PHONE\_NUMBER>754-1112</PHONE\_NUMBER>  
 </CLIENT>  
 </CLIENTS>  
 <RENTS>  
 <RENT ID="1" CLIENT\_ID="24" CD\_ID="1">  
 <FROM\_DATE>2020-10-01</FROM\_DATE>  
 <RETURN\_STATE>functional</RETURN\_STATE>  
 <RETURN\_DATE>2020-10-20</RETURN\_DATE>  
 </RENT>  
 <RENT ID="2" CLIENT\_ID="1" CD\_ID="2">  
 <FROM\_DATE>2020-01-10</FROM\_DATE>  
 <RETURN\_STATE>0</RETURN\_STATE>  
 <RETURN\_DATE>2020-01-21</RETURN\_DATE>  
 </RENT>  
 <RENT ID="3" CLIENT\_ID="16" CD\_ID="3">  
 <FROM\_DATE>2020-01-10</FROM\_DATE>  
 <RETURN\_STATE>0</RETURN\_STATE>  
 <RETURN\_DATE>2020-01-25</RETURN\_DATE>  
 </RENT>  
 </RENTS>  
</SYSTEM>

1. Напишете ја XML Schema датотеката за System.xml, и додадете ги следните ограничувања:

▪ атрибутот occupied во ентитетот CD и атрибутот return state во релацијата Rent може да бидат 0, 1 и 2 или functional, slightly damaged и damaged, соодветно.

▪ проверка на валидноста на форматот на email атрибутот кај Client ентитетот.

▪ телефонските броеви треба да се во формат 75x-yyyy каде x е цифра од 0 до 9 а y е цифра од 1 до 9

▪ адресата може и да не постои, но ако постои може да биде составена или од два поделементи (улица и број) или пак да биде во формат улица (составена само од мали и големи букви) по што следува St. па една или две цифри за бројот.

▪ сите други ограничувања кои би ги извеле од податоците дадени во XML документите.

<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
 elementFormDefault="qualified" attributeFormDefault="unqualified">  
 <xs:element name="SYSTEM">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="ARTISTS">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="ARTIST" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:choice minOccurs="0" maxOccurs="1">  
 <xs:sequence>  
 <xs:element name="NAME" type="xs:string"/>  
 <xs:element name="COUNTRY" type="xs:string"/>  
 <xs:element name="GENRE" type="xs:string"/>  
 </xs:sequence>  
 </xs:choice>  
 <xs:element name="YEAR\_STARTED\_PERFORMING" type="xs:gYear" minOccurs="0"/>  
 <xs:element name="NET\_WORTH" type="xs:string" minOccurs="0"/>  
 <xs:element name="YEAR\_OF\_BIRTH" type="xs:gYear" minOccurs="0"/>  
 </xs:sequence>  
 <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>  
 <xs:attribute name="type" use="required">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="band"/>  
 <xs:enumeration value="pop\_group"/>  
 <xs:enumeration value="dj"/>  
 <xs:enumeration value="singer"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:attribute>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
   
 <xs:element name="GROUPS">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="GROUP" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="YEAR\_FORMED" type="xs:gYear"/>  
 <xs:element name="NUMBER\_OF\_MEMBERS" type="xs:positiveInteger"/>  
 </xs:sequence>  
 <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
   
 <xs:element name="ALBUMS">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="ALBUM" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="NAME" type="xs:string"/>  
 <xs:element name="RELEASE\_YEAR" type="xs:gYear"/>  
 <xs:element name="PRICE" type="xs:string"/>  
 </xs:sequence>  
 <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>  
 <xs:attribute name="ARTIST\_ID" type="xs:positiveInteger" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
   
 <xs:element name="CATALOG">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="CD" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="STATE" type="CDStateType"/>  
 <xs:element name="OCCUPIED" type="CDStateType"/>  
 </xs:sequence>  
 <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>  
 <xs:attribute name="ALBUM\_ID" type="xs:positiveInteger" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
   
 <xs:element name="CLIENTS">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="CLIENT" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:choice>  
 <xs:sequence>  
 <xs:element name="NAME" type="xs:string"/>  
 <xs:element name="SURNAME" type="xs:string"/>  
 <xs:element name="ADDRESS" type="AddressType" minOccurs="0"/>  
 </xs:sequence>  
 <xs:element name="BASIC\_INFO">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="NAME" type="xs:string"/>  
 <xs:element name="SURNAME" type="xs:string"/>  
 <xs:element name="ADDRESS" type="AddressType" minOccurs="0"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:choice>  
 <xs:sequence>  
 <xs:element name="EMAIL" type="EmailType"/>  
 <xs:element name="PHONE\_NUMBER" type="PhoneNumberType" minOccurs="0"/>  
 </xs:sequence>  
 <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
   
 <xs:element name="RENTS">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="RENT" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="FROM\_DATE" type="xs:date"/>  
 <xs:element name="RETURN\_STATE" type="CDStateType"/>  
 <xs:element name="RETURN\_DATE" type="xs:date"/>  
 </xs:sequence>  
 <xs:attribute name="ID" type="xs:positiveInteger" use="required"/>  
 <xs:attribute name="CLIENT\_ID" type="xs:positiveInteger" use="required"/>  
 <xs:attribute name="CD\_ID" type="xs:positiveInteger" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
   
 <xs:simpleType name="CDStateType">  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="0"/>  
 <xs:enumeration value="1"/>  
 <xs:enumeration value="2"/>  
 <xs:enumeration value="functional"/>  
 <xs:enumeration value="slightly damaged"/>  
 <xs:enumeration value="damaged"/>  
 <xs:enumeration value="free"/>  
 </xs:restriction>  
 </xs:simpleType>  
   
 <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>  
   
 <xs:simpleType name="PhoneNumberType">  
 <xs:restriction base="xs:string">  
 <xs:pattern value="75[0-9]-[1-9]{4}"/>  
 </xs:restriction>  
 </xs:simpleType>  
   
 <xs:simpleType name="AddressType">  
 <xs:union memberTypes="FullAddressType SplitAddressType"/>  
 </xs:simpleType>  
   
 <xs:simpleType name="FullAddressType">  
 <xs:restriction base="xs:string">  
 <xs:pattern value="[a-zA-Z ]+ St\.[0-9]{1,2}"/>  
 </xs:restriction>  
 </xs:simpleType>  
  
 <xs:simpleType name="SplitAddressType">  
 <xs:restriction base="xs:string">  
 <xs:pattern value=".\*"/>  
 </xs:restriction>  
 </xs:simpleType>  
</xs:schema>

1. Напишете XQuery израз со кој користејќи ги оригиналните датотеки ќе ја изгенерирате датотеката System.xml. Забелешка: Резултатот од XQuery прашањето треба да ја претставува содржината на System.xml со точната структура што ја дефиниравте во одговорот на барањето под а).

**let** **$clients** := *doc*("Clients.xml")//**CLIENT**  
**let** **$artists** := *doc*("Artists.xml")//**ARTIST**  
**let** **$albums** := *doc*("Albums.xml")//**ALBUM**  
**let** **$groups** := *doc*("Groups.xml")//**GROUP**  
**let** **$rents** := *doc*("Rent.xml")//**RENT**  
**let** **$catalog** := *doc*("CatalogCD.xml")//**CD**  
  
**return**  
<SYSTEM>  
 <CLIENTS>  
 {  
 **for** **$client** **in** **$clients**  
 **return**  
 <CLIENT **ID**=**"**{**$client**/***@ID***}**"**>  
 <NAME>{**$client**/**NAME**/*text*()}</NAME>  
 <SURNAME>{**$client**/**SURNAME**/*text*()}</SURNAME>  
 <ADDRESS>  
 {  
 **if** (**$client**/**ADDRESS**/**STREET** and **$client**/**ADDRESS**/**NUMBER**) **then** (  
 <STREET>{**$client**/**ADDRESS**/**STREET**/*text*()}</STREET>,  
 <NUMBER>{**$client**/**ADDRESS**/**NUMBER**/*text*()}</NUMBER>  
 )  
 **else**  
 <FULL\_ADDRESS>{**$client**/**ADDRESS**/**FULL\_ADDRESS**/*text*()}</FULL\_ADDRESS>  
 }  
 </ADDRESS>  
 <EMAIL>{**$client**/**EMAIL**/*text*()}</EMAIL>  
 <PHONE\_NUMBER>{**$client**/**PHONE\_NUMBER**/*text*()}</PHONE\_NUMBER>  
 </CLIENT>  
 }  
 </CLIENTS>  
   
 <ARTISTS>  
 {  
 **for** **$artist** **in** **$artists**  
 **return**  
 <ARTIST **ID**=**"**{**$artist**/***@ID***}**"** **type**=**"**{**$artist**/***@type***}**"**>  
 <NAME>{**$artist**/**NAME**/*text*()}</NAME>  
 <COUNTRY>{**$artist**/**COUNTRY**/*text*()}</COUNTRY>  
 <GENRE>{**$artist**/**GENRE**/*text*()}</GENRE>  
 {  
 **if** (**$artist**/***@type*** = 'dj') **then** (  
 <YEAR\_STARTED\_PERFORMING>{**$artist**/**YEAR\_STARTED\_PERFORMING**/*text*()}</YEAR\_STARTED\_PERFORMING>,  
 <NET\_WORTH>{**$artist**/**NET\_WORTH**/*text*()}</NET\_WORTH>,  
 <YEAR\_OF\_BIRTH>{**$artist**/**YEAR\_OF\_BIRTH**/*text*()}</YEAR\_OF\_BIRTH>  
 )  
 **else** ()  
 }  
 </ARTIST>  
 }  
 </ARTISTS>  
  
 <ALBUMS>  
 {  
 **for** **$album** **in** **$albums**  
 **return**  
 <ALBUM **ID**=**"**{**$album**/***@ID***}**"** **ARTIST\_ID**=**"**{**$album**/***@ARTIST\_ID***}**"**>  
 <NAME>{**$album**/**NAME**/*text*()}</NAME>  
 <RELEASE\_YEAR>{**$album**/**RELEASE\_YEAR**/*text*()}</RELEASE\_YEAR>  
 <PRICE>{**$album**/**PRICE**/*text*()}</PRICE>  
 </ALBUM>  
 }  
 </ALBUMS>  
   
 <GROUPS>  
 {  
 **for** **$group** **in** **$groups**  
 **return**  
 <GROUP **ID**=**"**{**$group**/***@ID***}**"**>  
 <YEAR\_FORMED>{**$group**/**YEAR\_FORMED**/*text*()}</YEAR\_FORMED>  
 <NUMBER\_OF\_MEMBERS>{**$group**/**NUMBER\_OF\_MEMBERS**/*text*()}</NUMBER\_OF\_MEMBERS>  
 </GROUP>  
 }  
 </GROUPS>  
   
 <RENTS>  
 {  
 **for** **$rent** **in** **$rents**  
 **return**  
 <RENT **ID**=**"**{**$rent**/***@ID***}**"** **CLIENT\_ID**=**"**{**$rent**/***@CLIENT\_ID***}**"** **CD\_ID**=**"**{**$rent**/***@CD\_ID***}**"**>  
 <FROM\_DATE>{**$rent**/**FROM\_DATE**/*text*()}</FROM\_DATE>  
 <RETURN\_STATE>{**$rent**/**RETURN\_STATE**/*text*()}</RETURN\_STATE>  
 <RETURN\_DATE>{**$rent**/**RETURN\_DATE**/*text*()}</RETURN\_DATE>  
 </RENT>  
 }  
 </RENTS>  
 <CATALOG>  
 {  
 **for** **$cd** **in** **$catalog**  
 **return**  
 <CD **ID**=**"**{**$cd**/***@ID***}**"** **ALBUM\_ID**=**"**{**$cd**/***@ALBUM\_ID***}**"**>  
 <STATE>{**$cd**/**STATE**/*text*()}</STATE>  
 <OCCUPIED>{**$cd**/**OCCUPIED**/*text*()}</OCCUPIED>  
 </CD>  
 }  
 </CATALOG>  
</SYSTEM>

1. Кои се проблемите кои ги има претходното решение, доколку ги има? Што би смениле за да овозможите брз пристап до сите албуми кои ги има во системот, но и сите албуми кои ги има даден артист? Образложете зошто сте го избрале ова решение!

Претходното XQuery решение **технички е точно** – ги зема податоците од оригиналните XML-документи и ја генерира посакуваната структура System.xml. Сепак, од аспект на **организација на податоците, ефикасност и брз пристап**, има неколку **важни слабости**.

1. Раздвоени податоци (структури кои не се вгнездени)

* Албумите (<ALBUM>) се ставени во посебен <ALBUMS> елемент, додека артистите (<ARTIST>) се во друг.
* За да ги најдеш **сите албуми на еден артист**, мора да ги филтрираш сите албуми според @ARTIST\_ID.
* Ова води до **неоптимални пребарувања**, особено ако се користи оваа XML структура за преглед, анализа или веб-апликација.

1. Нема би-дирекционална навигација

* Не можеш од <ALBUM> лесно да „скокнеш“ до податоци за артистот (освен преку ID), ниту пак обратно.
* Секој ID мора експлицитно да се обработува преку where или let, што значи **потребни се дополнителни XQuery изрази** секогаш кога се прави поврзување.

Решение:

Секој <ARTIST> да ги вклучува своите албуми како вложен <ALBUMS> елемент.

<ARTIST ID="a1" type="solo">

<NAME>John Smith</NAME>

<COUNTRY>USA</COUNTRY>

<GENRE>Rock</GENRE>

<ALBUMS>

<ALBUM ID="alb1">

<NAME>First Album</NAME>

<RELEASE\_YEAR>2018</RELEASE\_YEAR>

<PRICE>12.99</PRICE>

</ALBUM>

<ALBUM ID="alb2">

...

</ALBUM>

</ALBUMS>

</ARTIST>

**ЗАДАЧА 2**

Да се напишат XQuery изразите со кои ќе се одговори на следните прашања:

a. Прикажи ги сите артисти (сите информации) заедно со насловите и датумите на издавање на сите албуми кои ги имаат снимено.

**for** **$artist** **in** *doc*("Artist.xml")/**ARTISTS**/**ARTIST**  
**let** **$albums** := *doc*("Albums.xml")/**ALBUMS**/**ALBUM**[***@ARTIST\_ID*** = **$artist**/***@ID***]  
**return**  
<ArtistInfo>  
 {**$artist**}  
 <Albums>  
 {  
 **for** **$album** **in** **$albums**  
 **return**  
 <Album>  
 <Name>{**$album**/**NAME**/*text*()}</Name>  
 <ReleaseYear>  
 {**if** (**$album**/**RELEASE\_YEAR**) **then** **$album**/**RELEASE\_YEAR**/*text*() **else** "N/A"}  
   
 </ReleaseYear>  
 </Album>  
 }  
 </Albums>  
</ArtistInfo>

A screenshot of a computer

AI-generated content may be incorrect.

b. Најди ги првите три члена со најголем број изнајмувања.

**let** **$clients** := *distinct-values*(*doc*("Rent.xml")//**RENT**/***@CLIENT\_ID***)  
**let** **$result** :=  
 **for** **$client\_id** **in** **$clients**  
 **let** **$rental\_count** := *count*(*doc*("Rent.xml")//**RENT**[***@CLIENT\_ID*** = **$client\_id**])  
 **order by** **$rental\_count** **descending**  
 **return**  
 <CLIENT>  
 <CLIENT\_ID>{*data*(**$client\_id**)}</CLIENT\_ID>  
 <RENTAL\_COUNT>{**$rental\_count**}</RENTAL\_COUNT>  
 </CLIENT>  
**return** *subsequence*(**$result**, 1, 3)

A screenshot of a computer

AI-generated content may be incorrect.

c. Врати ги сите албуми кои биле изнајмени најмалку три пати во периодот од јануари до март 2020.

**let** **$janToMarchRentals** := *doc*("Rent.xml")//**RENT**[  
 **let** **$date\_parts** := *tokenize*(**FROM\_DATE**, '/')  
 **let** **$month** := *number*(**$date\_parts**[1])  
 **let** **$year** := *number*(**$date\_parts**[3])  
 **return** **$year**=2020 and **$month**>=1 and **$month**<=3  
]  
  
**let** **$rentedAlbums** := **for** **$album** **in** *doc*("Albums.xml")//**ALBUM**  
**let** **$album\_id** := **$album**/***@ID***  
**let** **$rental\_count** := *count*(  
 **for** **$rent** **in** **$janToMarchRentals**  
 **let** **$cd\_id** := **$rent**/***@CD\_ID***  
 **where** *doc*("CatalogCD.xml")//**CD**[***@ID***=**$cd\_id** and ***@ALBUM\_ID*** = **$album\_id**]  
 **return** **$rent**  
 )  
 **where** **$rental\_count**>=3  
 **return**  
 <ALBUM>  
 <ID>{**$album\_id**}</ID>  
 <NAME>{**$album**/**NAME**/*text*()}</NAME>  
 <RENTAL\_COUNT>{**$rental\_count**}</RENTAL\_COUNT>  
 </ALBUM>  
**return** **$rentedAlbums**

d. Прикажи го клиентот кој најмногу изнајмувал од најизнајмениот албум заедно со информациите за албумот

**let** **$all\_rentals** := *doc*("Rent.xml")//**RENT**  
  
**let** **$album\_rentals\_counts** := **for** **$album** **in** *doc*("Albums.xml")//**ALBUM**  
 **let** **$album\_id** := **$album**/***@ID***  
 **let** **$rental\_count**:=*count*(  
 **for** **$rent** **in** **$all\_rentals**  
 **let** **$cd\_id**:=**$rent**/***@CD\_ID***  
 **where** *doc*("CatalogCD.xml")//**CD**[***@ID***=**$cd\_id** and ***@ALBUM\_ID*** = **$album\_id**]  
 **return** **$rent**  
 )  
 **order by** **$rental\_count** **descending**  
 **return**  
 <ALBUM\_COUNT>  
 <ALBUM\_ID>{**$album\_id**}</ALBUM\_ID>  
 <COUNT>{**$rental\_count**}</COUNT>  
 </ALBUM\_COUNT>  
   
**let** **$most\_rented\_album\_id** := **$album\_rentals\_counts**[1]/**ALBUM\_ID**/*text*()  
**let** **$most\_rented\_album** := *doc*("Albums.xml")//**ALBUM**[***@ID***=**$most\_rented\_album\_id**]  
  
**let** **$album\_cd\_id** := **for** **$cd** **in** *doc*("CatalogCD.xml")//**CD**[***@ALBUM\_ID*** = **$most\_rented\_album\_id**]  
 **return** **$cd**/***@ID***  
   
**let** **$album\_rentals** := **for** **$rent** **in** **$all\_rentals**  
 **where** **$rent**/***@CD\_ID*** = **$album\_cd\_id**  
 **return** **$rent**   
   
**let** **$client\_rental\_counts** :=   
 **for** **$client\_id** **in** *distinct-values*(**$album\_rentals**/***@CLIENT\_ID***)  
 **let** **$count** := *count*(**$album\_rentals**[***@CLIENT\_ID*** = **$client\_id**])  
 **order by** **$count** **descending**  
 **return**   
 <CLIENT\_COUNT>  
 <CLIENT\_ID>{**$client\_id**}</CLIENT\_ID>  
 <COUNT>{**$count**}</COUNT>  
 </CLIENT\_COUNT>  
   
**let** **$top\_client\_id** := **$client\_rental\_counts**[1]/**CLIENT\_ID**/*text*()  
**let** **$top\_client** := *doc*("Clients.xml")//**CLIENT**[***@ID***=**top\_client\_id**]  
  
**return**  
 <RESULT>  
 <TOP\_CLIENT>  
 {**$top\_client**}  
 </TOP\_CLIENT>  
 <MOST\_RENTED\_ALBUM>  
 {**$most\_rented\_album**}  
 </MOST\_RENTED\_ALBUM>  
 </RESULT>

A screenshot of a computer

AI-generated content may be incorrect.

e. Прикажи го просечниот број на изнајмувања на секој албум посебно.

**let** **$albums** := *doc*("Albums.xml")//**ALBUM**  
**let** **$catalog** := *doc*("CatalogCD.xml")//**CD**  
**let** **$rents** := *doc*("Rent.xml")//**RENT**  
  
**return**  
 **for** **$album** **in** **$albums**  
 **let** **$album\_id** := **$album**/***@ID***  
 **let** **$cds** := **$catalog**[***@ALBUM\_ID*** = **$album\_id**]  
 **let** **$cd\_ids** := **$cds**/***@ID***  
 **let** **$rent\_count** := *count*(  
 **for** **$r** **in** **$rents**  
 **where** **some** **$id** **in** **$cd\_ids** **satisfies** **$r**/***@CD\_ID*** = **$id**  
 **return** **$r**  
 )  
 **let** **$cd\_count** := *count*(**$cd\_ids**)  
 **let** **$average** :=   
 **if** (**$cd\_count** > 0) **then**  
 *round*(**$rent\_count** div **$cd\_count** \* 100) div 100  
 **else**  
 0  
 **return**  
 <AlbumAverage>  
 <AlbumID>{ **$album\_id** }</AlbumID>  
 <Name>{ **$album**/**NAME**/*text*() }</Name>  
 <CD\_Count>{ **$cd\_count** }</CD\_Count>  
 <TotalRentals>{ **$rent\_count** }</TotalRentals>  
 <AverageRentalsPerCD>{ **$average** }</AverageRentalsPerCD>  
 </AlbumAverage>

f. Врати го вкупниот профит кој е направен за секој албум посебно.

**let** **$albums** := *doc*("Albums.xml")//**ALBUM**  
**let** **$catalog** := *doc*("CatalogCD.xml")//**CD**  
**let** **$rents** := *doc*("Rent.xml")//**RENT**  
  
**return**  
 **for** **$album** **in** **$albums**  
 **let** **$album\_id** := **$album**/***@ID***  
 **let** **$album\_price** := *xs:decimal*(*replace*(**$album**/**PRICE**, '[$]', ''))  
 **let** **$cds** := **$catalog**[***@ALBUM\_ID*** = **$album\_id**]  
 **let** **$cd\_ids** := **$cds**/***@ID***  
 **let** **$rent\_count** := *count*(  
 **for** **$r** **in** **$rents**  
 **where** **some** **$id** **in** **$cd\_ids** **satisfies** **$r**/***@CD\_ID***=**$id**  
 **return** **$r**  
 )  
 **let** **$total\_profit** := **$rent\_count** \* **$album\_price**  
 **return**  
 <AlbumProfit>  
 <AlbumID>{ **$album\_id** }</AlbumID>  
 <Name>{ **$album**/**NAME**/*text*() }</Name>  
 <Price>{ **$album\_price** }</Price>  
 <TotalRentals>{ **$rent\_count** }</TotalRentals>  
 <TotalProfit>{ *format-number*(**$total\_profit**, '#.00') }</TotalProfit>  
 </AlbumProfit>

g. Најди ја групата чиј албум е најмногу пати изнајмен, но постои барем едно CD кое во моментот не е изнајмено.

**let** **$all\_rentals** := *doc*("Rent.xml")//**RENT**  
  
**let** **$album\_rental\_counts** :=  
 **for** **$album** **in** *doc*("Albums.xml")//**ALBUM**  
 **let** **$album\_id** := **$album**/***@ID***  
 **let** **$artist\_id** := **$album**/***@ARTIST\_ID***  
 **let** **$rental\_count** := *count*(  
 **for** **$rent** **in** **$all\_rentals**  
 **let** **$cd\_id** := **$rent**/***@CD\_ID***  
 **where** *doc*("CatalogCD.xml")//**CD**[***@ID*** = **$cd\_id** and ***@ALBUM\_ID*** = **$album\_id**]  
 **return** **$rent**  
 )  
 **let** **$has\_available\_cd** :=   
 **some** **$cd** **in** *doc*("CatalogCD.xml")//**CD**[***@ALBUM\_ID*** = **$album\_id**]   
 **satisfies** (**$cd**/**OCCUPIED** = "0" or *string*(**$cd**/**OCCUPIED**) = "free" or *string*(**$cd**/**OCCUPIED**) = "0")  
 **where** **$has\_available\_cd**  
 **order by** **$rental\_count** **descending**  
 **return**  
 <ALBUM\_GROUP>  
 <ALBUM\_ID>{*data*(**$album\_id**)}</ALBUM\_ID>  
 <ARTIST\_ID>{*data*(**$artist\_id**)}</ARTIST\_ID>  
 <RENTAL\_COUNT>{**$rental\_count**}</RENTAL\_COUNT>  
 </ALBUM\_GROUP>  
  
**let** **$top\_album** := **$album\_rental\_counts**[1]  
**let** **$group\_id** := **$top\_album**/**ARTIST\_ID**/*text*()  
**let** **$group** := *doc*("Groups.xml")//**GROUP**[***@ID*** = **$group\_id**]  
  
**return**  
 <RESULT>  
 <GROUP\_ID>{*data*(**$group**/***@ID***)}</GROUP\_ID>  
 <YEAR\_FORMED>{**$group**/**YEAR\_FORMED**/*text*()}</YEAR\_FORMED>  
 <NUMBER\_OF\_MEMBERS>{**$group**/**NUMBER\_OF\_MEMBERS**/*text*()}</NUMBER\_OF\_MEMBERS>  
 <TOP\_ALBUM\_ID>{**$top\_album**/**ALBUM\_ID**/*text*()}</TOP\_ALBUM\_ID>  
 <RENTAL\_COUNT>{**$top\_album**/**RENTAL\_COUNT**/*text*()}</RENTAL\_COUNT>  
 </RESULT>

A screenshot of a computer

AI-generated content may be incorrect.

h. Прикажи ги сите клиенти кои барем еднаш изнајмиле CD во времетраење пократко од 10 дена.

**let** **$short\_term\_renters** :=  
 **for** **$client** **in** *doc*("Clients.xml")//**CLIENT**  
 **let** **$client\_id** := **$client**/***@ID***  
 **where** **some** **$rent** **in** *doc*("Rent.xml")//**RENT**[***@CLIENT\_ID***=**$client\_id**]  
 **satisfies**  
 **let** **$from\_date\_parts** := *tokenize*(**$rent**/**FROM\_DATE**,'/')  
 **let** **$return\_date\_parts** := *tokenize*(**$rent**/**RETURN\_DATE**, '/')  
 **let** **$from\_date** :=   
 **if** (*count*(**$from\_date\_parts**) = 3 and   
 *string-length*(**$from\_date\_parts**[3]) >= 4 and   
 *string-length*(**$from\_date\_parts**[1]) > 0 and   
 *string-length*(**$from\_date\_parts**[2]) > 0)   
 **then** *xs:date*(*concat*(**$from\_date\_parts**[3], '-', **$from\_date\_parts**[1], '-', **$from\_date\_parts**[2]))  
 **else** ()  
 **let** **$return\_date** :=   
 **if** (*count*(**$return\_date\_parts**) = 3 and   
 *string-length*(**$return\_date\_parts**[3]) >= 4 and   
 *string-length*(**$return\_date\_parts**[1]) > 0 and   
 *string-length*(**$return\_date\_parts**[2]) > 0)   
 **then** *xs:date*(*concat*(**$return\_date\_parts**[3], '-', **$return\_date\_parts**[1], '-', **$return\_date\_parts**[2]))  
 **else** ()  
 **return**   
 **if** (*exists*(**$from\_date**) and *exists*(**$return\_date**)) **then**  
 *days-from-duration*(**$return\_date** - **$from\_date**) < 10  
 **else**  
 *false*()  
 **return** **$client**  
   
**return**  
 <SHORT\_TERM\_RENTERS>  
 {  
 **for** **$client** **in** **$short\_term\_renters**  
 **return**  
 <CLIENT>  
 <ID>{**$client**/***@ID***}</ID>  
 {  
 **if** (**$client**/**NAME**) **then**  
 <NAME>{**$client**/**NAME**/*text*()}</NAME>  
 **else**  
 <NAME>{**$client**/**BASIC\_INFO**/**NAME**/*text*()}</NAME>  
 }  
 {  
 **if**(**$client**/**SURNAME**) **then**  
 <SURNAME>{**$client**/**SURNAME**/*text*()}</SURNAME>  
 **else**  
 <SURNAME>{**$client**/**BASIC\_INFO**/**SURNAME**/*text*()}</SURNAME>  
 }  
 {  
 **if** (**$client**/**EMAIL**) **then**  
 <EMAIL>{**$client**/**EMAIL**/*text*()}</EMAIL>  
 **else** ()  
 }  
 </CLIENT>  
 }  
 </SHORT\_TERM\_RENTERS>

A screenshot of a computer code

AI-generated content may be incorrect.

i. Најди го омилениот артист на секој клиент посебно (сите информации за артистот од кој клиентот има изнајмено најмногу албуми, ако има повеќе вратете го првиот).

**for** **$client** **in** *doc*("Clients.xml")//**CLIENT**  
**let** **$rentals** := *doc*("Rent.xml")//**RENT**[**CLIENT\_ID**=**$client**/***@ID***]  
**let** **$artistRentals** :=  
 **for** **$artist** **in** *doc*("Artists.xml")//**ARTIST**  
 **let** **$artistAlbums** := *doc*("Albums.xml")//**ALBUM**[***@ARTIST\_ID***=**$artist**/**ID**]  
 **let** **$rentCount** := *count*(**$rentals**[**CD\_ID**=**$artistAlbums**/***@ID***])  
 **where** **$rentCount** > 0  
 **order by** **$rentCount** **descending**  
 **return**  
 <ARTIST\_RENTAL>  
 <ARTIST\_ID>{**$artist**/***@ID***}</ARTIST\_ID>  
 <ARTIST\_NAME>{**$artist**/**NAME**/*text*()}</ARTIST\_NAME>  
 <RENT\_COUNT>{**$rentCount**}</RENT\_COUNT>  
 <COUNTRY>{**$artist**/**COUNTRY**/*text*()}</COUNTRY>  
 <GENRES>  
 {**for** **$genre** **in** **$artist**/**GENRE**  
 **return** <GENRE>{**$genre**/*text*()}</GENRE>}  
 </GENRES>  
 </ARTIST\_RENTAL>  
   
**let** **$favoriteArtist** := **$artistRentals**[1]  
**where** *count*(**$artistRentals**) > 0  
**return**  
 <CLIENT\_FAVORITE>  
 <CLIENT\_ID>{**$client**/***@ID***}</CLIENT\_ID>  
 <CLIENT\_NAME>  
 {**if** (**$client**/**NAME**) **then** **$client**/**NAME**/*text*()  
 **else** **$client**/**BASIC\_INFO**/**NAME**/*text*()}  
 </CLIENT\_NAME>  
 <CLIENT\_SURNAME>  
 {**if** (**$client**/**SURNAME**) **then** **$client**/**SURNAME**/*text*()  
 **else** **$client**/**BASIC\_INFO**/**SURNAME**/*text*()}  
 </CLIENT\_SURNAME>  
 <FAVORITE\_ARTIST>  
 {**$favoriteArtist**}  
 </FAVORITE\_ARTIST>  
   
 </CLIENT\_FAVORITE>

j. Најди го најомилениот артист (артистот кој е омилен на најголемиот број клиенти).

☹

k. Напиши кориснички дефинирана функција за генерирање месечен извештај. Месецот и годината се задаваат како влезни параметри, а во извештајот за тековниот месец ќе се вратат профитот направен од сите албуми и бројот на продажби по албум за секој артист посебно.

☹