**Лабораторная работа № 15. Использование XML**

1. Разработать сценарий создания XML-документа в режиме PATH из таблицы **TEACHER** для преподавателей кафедры ИСиТ.

use UNIVER;

set nocount on;

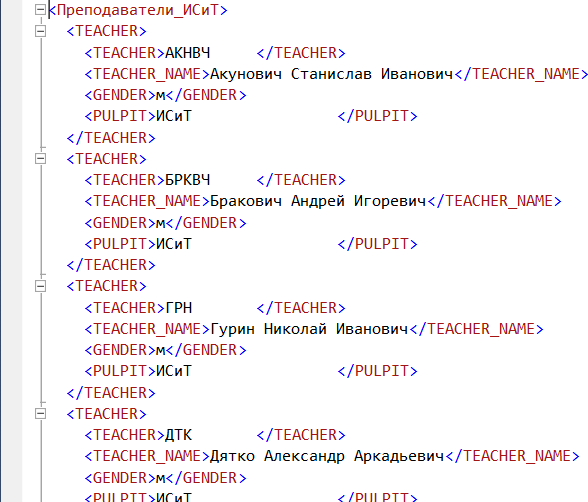
go

SELECT \* from TEACHER

where PULPIT = 'ИСиТ'

FOR xml PATH('TEACHER'), root('Преподаватели\_ИСиТ'), elements;





1. Разработать сценарий создания XML-документа в режиме AUTO на основе SELECT-запроса к таблицам **AUDITORIUM** и **AUDITORIUM\_TYPE,** которыйсодержит следующие столбцы: наименование аудитории, наименование типа аудитории и вместимость. Найти только лекционные аудитории.

use UNIVER;

set nocount on;

go

SELECT

AUDITORIUM.AUDITORIUM\_NAME,

AUDITORIUM.AUDITORIUM\_TYPE,

AUDITORIUM.AUDITORIUM\_CAPACITY

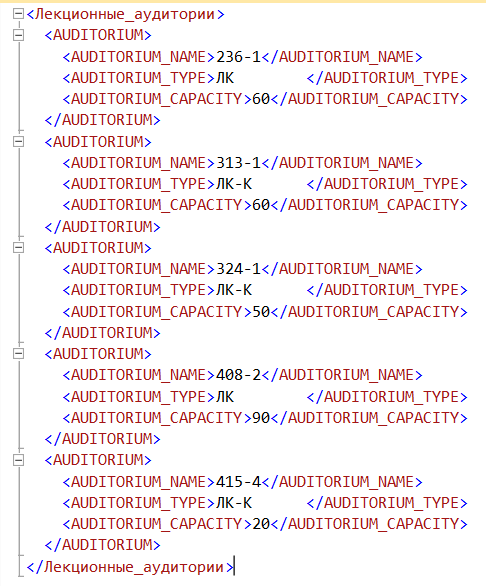
FROM AUDITORIUM\_TYPE

JOIN AUDITORIUM ON AUDITORIUM\_TYPE.AUDITORIUM\_TYPE = AUDITORIUM.AUDITORIUM\_TYPE

WHERE AUDITORIUM.AUDITORIUM\_TYPE like 'ЛК%'

for xml AUTO, root('Лекционные\_аудитории'), elements;





1. Разработать XML-документ, содержащий данные о трех новых учебных дисциплинах, которые следует добавить в таблицу **SUBJECT**. Разработать сценарий, извлекающий данные о дисциплинах из XML-документа и добавляющий их в таблицу **SUBJECT**. При этом применить системную функцию **OPENXML** и конструкцию INSERT… SELECT.

use UNIVER;

set nocount on;

go

declare @h int = 0,

@x varchar(2000) =

'<?xml version="1.0" encoding="windows-1251" ?>

<SUBJECTS>

<SUBJECT\_T SUBJECT\_T="new\_subj1" SUBJECT\_NAME="new\_subj1" PULPIT="ИСиТ" />

<SUBJECT\_T SUBJECT\_T="new\_subj2" SUBJECT\_NAME="new\_subj2" PULPIT="ИСиТ" />

<SUBJECT\_T SUBJECT\_T="new\_subj3" SUBJECT\_NAME="new\_subj3" PULPIT="ИСиТ" />

</SUBJECTS>';

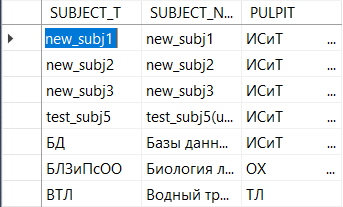
exec sp\_xml\_preparedocument @h output, @x;

insert SUBJECT\_T select [SUBJECT\_T], [SUBJECT\_NAME], [PULPIT]

from openxml(@h, '/SUBJECTS/SUBJECT\_T', 0)

with([SUBJECT\_T] nvarchar(10), [SUBJECT\_NAME] nvarchar(100), [PULPIT] nvarchar(20))

exec sp\_xml\_removedocument @h;



1. Используя таблицу **STUDENT** разработать XML-структуру, содержащую паспортные данные студента: серию и номер паспорта, личный номер, дата выдачи и адрес прописки. Разработать сценарий, в который включен оператор INSERT, добавляющий строку с XML-столбцом. Включить в этот же сценарий оператор UPDATE, изменяющий столбец **INFO** у одной строки таблицы **STUDENT** и оператор SELECT, формирующий результирующий набор, аналогичный представленному на рисунке. В SELECT-запросе использовать методы QUERY и VALUEXML-типа.

use UNIVER

INSERT INTO STUDENT(IDGROUP, NAME, BDAY, INFO) values

(2, 'Добриян Александр Витальевич', '1970-01-01',

'<паспорт>

<серия>1</серия>

<номер>123456</номер>

<дата\_выдачи>1980-01-01</дата\_выдачи>

<адрес>Беларусь, г. Минск, Свердлова 13а</адрес>

</паспорт>');

select \* from STUDENT where NAME = 'Добриян Александр Витальевич';

update STUDENT set INFO =

'<паспорт>

<серия>2</серия>

<номер>654321</номер>

<дата\_выдачи>1981-01-01</дата\_выдачи>

<адрес>Беларусь, г. Минск, Свердлова 13а</адрес>

</паспорт>'

where STUDENT.INFO.value('(/паспорт/серия)[1]','int') = 1;

select \* from STUDENT where NAME = 'Добриян Александр Витальевич';

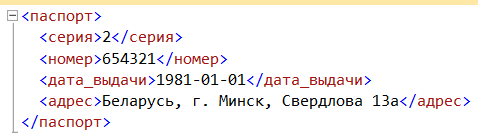
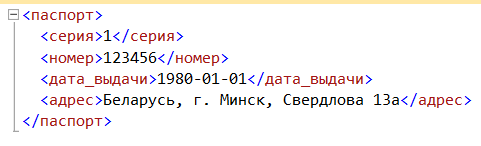
select NAME,

INFO.value('(/паспорт/дата\_выдачи)[1]', 'date') [дата\_выдачи],

INFO.value('(/паспорт/адрес)[1]', 'varchar(100)') [адрес],

INFO.query('/паспорт') [паспорт]

from STUDENT where NAME = 'Добриян Александр Витальевич';



1. Изменить (ALTER TABLE) таблицу **STUDENT** в базе данных **UNIVER** таким образом, чтобы значения *типизированного* столбца с именем **INFO** контролировались коллекцией XML-схем (XML SCHEMACOLLECTION), представленной в правой части. Разработать сценарии, демонстрирующие ввод и корректировку данных (операторы INSERT и UPDATE) в столбец **INFO** таблицы **STUDENT**, как содержащие ошибки, так и правильные. Разработать другую XML-схему и добавить ее в коллекцию XML-схем в БД UNIVER**.**

use UNIVER;

create xml schema collection Student\_info as

N'<?xml version="1.0" encoding="utf-16" ?>

<xs:schema attributeFormDefault="unqualified"

elementFormDefault="qualified"

xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="студент">

<xs:complexType><xs:sequence>

<xs:element name="паспорт" maxOccurs="1" minOccurs="1">

<xs:complexType>

<xs:attribute name="серия" type="xs:string" use="required" />

<xs:attribute name="номер" type="xs:unsignedInt" use="required"/>

<xs:attribute name="дата" use="required" >

<xs:simpleType> <xs:restriction base ="xs:string">

<xs:pattern value="[0-9]{2}.[0-9]{2}.[0-9]{4}"/>

</xs:restriction> </xs:simpleType>

</xs:attribute> </xs:complexType>

</xs:element>

<xs:element maxOccurs="3" name="телефон" type="xs:unsignedInt"/>

<xs:element name="адрес"> <xs:complexType><xs:sequence>

<xs:element name="страна" type="xs:string" />

<xs:element name="город" type="xs:string" />

<xs:element name="улица" type="xs:string" />

<xs:element name="дом" type="xs:string" />

<xs:element name="квартира" type="xs:string" />

</xs:sequence></xs:complexType> </xs:element>

</xs:sequence></xs:complexType>

</xs:element>

</xs:schema>';

alter table STUDENT alter column INFO xml(Student\_info);

drop XML SCHEMA COLLECTION Student\_info;

1. Разработать сценарии, демонстрирующие использование XML для базы данных **X\_MyBASE**.

use Dobriyan\_MyBase;

set nocount on;

go

SELECT \* from Courses

where Courses.Lessons\_Type = 'аудиторно'

FOR xml PATH('Courses'), root('Курсы\_в\_аудиториях'), elements;

--- #

use Dobriyan\_MyBase;

set nocount on;

go

SELECT

Groups.Number,

Groups.Students\_Count,

Specialities.Speciality

FROM Groups

JOIN Specialities ON Groups.Speciality = Specialities.id

WHERE Specialities.Speciality = 'ДЭиВИ'

ORDER BY Groups.Number

for xml AUTO, root('ДЭиВИ\_группы'), elements;

--- #

use Dobriyan\_MyBase;

set nocount on;

go

declare @h int = 0,

@x varchar(2000) =

'<?xml version="1.0" encoding="windows-1251" ?>

<specs>

<Specialities id="22" Speciality="new\_spec1" />

<Specialities id="23" Speciality="new\_spec2" />

</specs>';

exec sp\_xml\_preparedocument @h output, @x;

insert Specialities select id, Speciality

from openxml(@h, '/specs/Specialities', 0)

with(id int, Speciality nvarchar(20))

exec sp\_xml\_removedocument @h;