**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Институт компьютерных технологий и защиты информации**

**Кафедра прикладной математики и информатики**

Лабораторная работа №1

По дисциплине: Программирование на Transact SQL

 Выполнил: студент группы 4311

                    Федотов А.Д., Червов В.А.

                                                                                      Проверил: Зайдулин С.С.

                  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

Казань, 2024 год

**Краткие теоретические сведения**

Вы можете использовать фразу FOR XML в операторе SELECT языка Transact-SQL, чтобы получить данные в формате XML вместо строк и столбцов. Вы можете управлять форматом XML, определяя один из четырех режимов: RAW, AUTO, EXPLICIT или PATH.

Вы используете запросы в режиме RAW, чтобы получить XML представление набора строк. Приложения могут обработать XML в естественном формате или применить стилевые таблицы XSLT, чтобы преобразовать XML в необходимый деловой формат документа или представление в интерфейсе пользователя.

Запросы в режиме AUTO производят XML представления данных о сущностях.

Запросы в режиме EXPLICIT.

Ключом к пониманию формирования настраиваемых XML документов является понятие универсальной таблицы. Универсальная таблица - табличное представление XML документа. Каждая строка в универсальной таблице представляет данные, которые будут представлены в виде элемента в получающемся документе XML.

Первые два столбца универсальной таблицы определяют в результирующем XML документе иерархическое положение элемента, который содержит данные строки.

Остальные столбцы универсальной таблицы содержат данные, которые будут представлены в документе.

После того, как Вы определили универсальную таблицу, необходимую для получения желаемого XML-документа, Вы можете построить запрос Transact-SQL, необходимый, чтобы сгенерировать таблицу с использованием псевдонимов для именования столбцов.

Режим PATH позволяет получить настроенный XML-документ при использовании синтаксиса XPath, чтобы значения соответствовали атрибутам, элементам, подэлементам, текстовым узлам и значениям данных. Эта способность соотнести столбцы таблиц с узлами XML-документа позволяют получить сложный XML без сложности запроса в режиме EXPLICIT.

Синтаксис для получения вложенных XML-документов.

Вложение XML позволяет Вам представить отношение родитель/потомок как иерархию XML – например, клиенты и их заказы, или заказы и их позиции. Есть несколько способов вложения элементов XML при использовании фразы FOR XML, включая:

- Определение соединений в запросах режима AUTO

- Определение опции TYPE в подзапросах, чтобы получить значения типа xml

- Объединение множества универсальных таблиц при использовании UNION ALL в

запросах в режиме EXPLICIT

**Задание на лабораторную работу**

1. Изучить материалы урока 1 из файла "Теоретические материалы - Работа с XML".
2. Ответить на контрольные вопросы 1-6.
3. Выполнить все примеры, приведенные в уроке 1 файла "Теоретические материалы - Работа с XML". Для выполнения примеров необходима БД AdventureWorks, входящая в комплект поставки MS SQL Server (устанавливается вместе с СУБД, либо скачивается отдельно и присоединяется к серверу).
4. Создать базу данных с 3-5 таблицами по индивидуальному заданию. Можно использовать БД, созданную для курсовой работы по дисциплине "Базы данных".
5. Для каждого режима (RAW, AUTO, PATH, EXPLICIT) создать по одному-два запроса, переводящему реляционные данные в формат XML. При этом должны быть запросы, формирующие вложенные XML-документы, содержащие данные 2-3 таблиц.

**Ход работы**

**Примеры**

-- Использование базы данных AdventureWorks2012

USE AdventureWorks2012

GO

-- Получение данных о клиентах и заказах с формированием XML в формате RAW

SELECT Cust.CustomerID CustID, TaxAmt, SalesOrderID

FROM Sales.Customer Cust JOIN Sales.SalesOrderHeader [Order]

ON Cust.CustomerID = [Order].CustomerID

ORDER BY Cust.CustomerID

FOR XML RAW

-- Получение данных о клиентах и заказах с формированием XML в формате RAW и включением элементов

SELECT Cust.CustomerID CustID, TaxAmt, SalesOrderID

FROM Sales.Customer Cust JOIN Sales.SalesOrderHeader [Order]

ON Cust.CustomerID = [Order].CustomerID

ORDER BY Cust.CustomerID

FOR XML RAW, ELEMENTS

-- Получение данных о клиентах и заказах с формированием XML в формате RAW, с пользовательским названием корневого элемента и корневого тега

SELECT Cust.CustomerID CustID, TaxAmt, SalesOrderID

FROM Sales.Customer Cust JOIN Sales.SalesOrderHeader [Order]

ON Cust.CustomerID = [Order].CustomerID

ORDER BY Cust.CustomerID

FOR XML RAW('Order'), ROOT('Orders')

-- Получение данных о клиентах и заказах с формированием XML в формате AUTO

SELECT Cust.CustomerID CustID, TaxAmt, SalesOrderID

FROM Sales.Customer Cust JOIN Sales.SalesOrderHeader [Order]

ON Cust.CustomerID = [Order].CustomerID

ORDER BY Cust.CustomerID

FOR XML AUTO

-- Получение данных о клиентах и заказах с формированием XML в формате AUTO и включением элементов

SELECT Cust.CustomerID CustID, TaxAmt, SalesOrderID

FROM Sales.Customer Cust JOIN Sales.SalesOrderHeader [Order]

ON Cust.CustomerID = [Order].CustomerID

ORDER BY Cust.CustomerID

FOR XML AUTO, ELEMENTS

-- Получение данных о клиентах и заказах с формированием XML в формате AUTO, включением элементов и добавлением корневого тега

SELECT Cust.CustomerID CustID, TaxAmt, SalesOrderID

FROM Sales.Customer Cust JOIN Sales.SalesOrderHeader [Order]

ON Cust.CustomerID = [Order].CustomerID

ORDER BY Cust.CustomerID

FOR XML AUTO, ELEMENTS, ROOT('Orders')

-- Получение данных о заказах с использованием формата EXPLICIT для более детальной структуры XML

SELECT 1 AS Tag, NULL AS Parent, SalesOrderID AS [Invoice!1!InvoiceNo], OrderDate AS [Invoice!1!Date!Element]

FROM Sales.SalesOrderHeader

FOR XML EXPLICIT

-- Получение данных о сотрудниках с использованием формата PATH для построения XML

SELECT Person.BusinessEntity.BusinessEntityID "@EmpID",

FirstName "EmpName/First",

LastName "EmpName/Last"

FROM Person.Person INNER JOIN

Person.BusinessEntity ON Person.Person.BusinessEntityID = Person.BusinessEntity.BusinessEntityID

FOR XML PATH

-- Получение данных о сотрудниках с использованием формата PATH и указанием пользовательского названия элемента

SELECT Person.BusinessEntity.BusinessEntityID "@EmpID",

FirstName "EmpName/First",

LastName "EmpName/Last"

FROM Person.Person INNER JOIN

Person.BusinessEntity ON Person.Person.BusinessEntityID = Person.BusinessEntity.BusinessEntityID

FOR XML PATH('Employee')

-- Повторное получение данных о клиентах и заказах с формированием XML в формате AUTO и включением элементов

SELECT Cust.CustomerID CustID, TaxAmt, SalesOrderID

FROM Sales.Customer Cust JOIN Sales.SalesOrderHeader [Order]

ON Cust.CustomerID = [Order].CustomerID

ORDER BY Cust.CustomerID

FOR XML AUTO, ELEMENTS

-- Получение категорий продуктов с вложенным подзапросом для получения подкатегорий в формате XML

SELECT Name CategoryName,

(SELECT Name SubCategoryName

FROM Production.ProductSubCategory SubCategory

WHERE SubCategory.ProductCategoryID=Category.ProductCategoryID

FOR XML AUTO, TYPE, ELEMENTS)

FROM Production.ProductCategory Category

FOR XML AUTO

-- Получение данных о заказах и деталях заказа с использованием формата EXPLICIT для сложной структуры XML

SELECT 1 AS Tag,

NULL AS Parent,

SalesOrderID AS [Invoice!1!InvoiceNo],

OrderDate AS [Invoice!1!Date!Element],

NULL AS [LineItem!2!ProductID],

NULL AS [LineItem!2]

FROM Sales.SalesOrderHeader

UNION ALL

SELECT 2 AS Tag,

1 AS Parent,

OrderDetail.SalesOrderID,

NULL,

OrderDetail.ProductID,

Product.Name

FROM Sales.SalesOrderDetail OrderDetail JOIN

Sales.SalesOrderHeader OrderHeader

ON OrderDetail.SalesOrderID= OrderHeader.SalesOrderID

JOIN Production.Product Product

ON OrderDetail.ProductID = Product.ProductID

ORDER BY [Invoice!1!InvoiceNo], [LineItem!2!ProductID]

FOR XML EXPLICIT

**Запросы к своей бд**

-- Переход к базе данных master

USE master

GO

-- Создание базы данных "shop"

CREATE DATABASE shop

GO

-- Использование базы данных "shop"

USE shop

GO

-- Создание таблицы "Users" для хранения информации о пользователях

CREATE TABLE Users (

UserID INT IDENTITY PRIMARY KEY,

UserName VARCHAR(255),

Email VARCHAR(255),

PasswordHash VARCHAR(255)

);

-- Создание таблицы "Categories" для хранения информации о категориях продуктов

CREATE TABLE Categories (

CategoryID INT IDENTITY PRIMARY KEY,

CategoryName VARCHAR(255)

);

-- Создание таблицы "Products" для хранения информации о продуктах

CREATE TABLE Products (

ProductID INT IDENTITY PRIMARY KEY,

ProductName VARCHAR(255),

Price DECIMAL(10, 2),

CategoryID INT,

FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)

);

-- Создание таблицы "Orders" для хранения информации о заказах

CREATE TABLE Orders (

OrderID INT IDENTITY PRIMARY KEY,

UserID INT,

OrderDate DATETIME,

TotalPrice DECIMAL(10, 2),

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

-- Создание таблицы "OrderDetails" для хранения деталей заказов

CREATE TABLE OrderDetails (

OrderDetailID INT IDENTITY PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT,

Price DECIMAL(10, 2),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

-- Вставка данных в таблицу "Users"

INSERT INTO Users (UserName, Email, PasswordHash) VALUES

('IvanIvanov', 'ivan@example.com', 'hash1'),

('PetrPetrov', 'petr@example.com', 'hash2');

-- Вставка данных в таблицу "Categories"

INSERT INTO Categories (CategoryName) VALUES

('Электроника'),

('Книги');

-- Вставка данных в таблицу "Products"

INSERT INTO Products (ProductName, Price, CategoryID) VALUES

('Телефон', 50000, 1),

('Ноутбук', 80000, 1),

('Война и мир', 1500, 2);

-- Вставка данных в таблицу "Orders"

INSERT INTO Orders (UserID, OrderDate, TotalPrice) VALUES

(1, '2024-02-01 12:00:00', 65000),

(2, '2024-02-02 15:30:00', 1500);

-- Вставка данных в таблицу "OrderDetails"

INSERT INTO OrderDetails (OrderID, ProductID, Quantity, Price) VALUES

(1, 1, 1, 50000),

(1, 2, 1, 80000),

(2, 3, 1, 1500);

-- Использование базы данных "shop"

USE shop;

GO

-- Получение данных о продуктах по категориям в формате XML

SELECT cat.CategoryID, cat.CategoryName, prod.ProductID, prod.ProductName, prod.Price

FROM Categories cat

JOIN Products prod ON cat.CategoryID = prod.CategoryID

FOR XML RAW('category'), ROOT('products\_by\_category');

-- Повторный запрос для получения данных о продуктах по категориям в формате XML

SELECT cat.CategoryID, cat.CategoryName, prod.ProductID, prod.ProductName, prod.Price

FROM Categories cat

JOIN Products prod ON cat.CategoryID = prod.CategoryID

FOR XML RAW('category'), ROOT('products\_by\_category');

-- Получение данных о пользователях и их заказах с деталями в формате XML AUTO

SELECT [user].UserName, [order].OrderID, [order].OrderDate, item.ProductID,

product.ProductName, item.Quantity, item.Price

FROM Users [user]

JOIN Orders [order] ON [user].UserID = [order].UserID

JOIN OrderDetails item ON [order].OrderID = item.OrderID

JOIN Products product ON item.ProductID = product.ProductID

FOR XML AUTO, ROOT('users');

-- Получение данных о пользователях и их заказах с деталями в формате XML PATH с указанием пользовательских тегов

SELECT [user].UserID '@customusername', [order].OrderID, [order].OrderDate, item.ProductID,

product.ProductName, item.Quantity, item.Price

FROM Users [user]

JOIN Orders [order] ON [user].UserID = [order].UserID

JOIN OrderDetails item ON [order].OrderID = item.OrderID

JOIN Products product ON item.ProductID = product.ProductID

ORDER BY [user].userid, [order].orderid

FOR XML PATH('user'), ROOT('users');

-- Получение данных о заказах и их деталях в формате XML EXPLICIT

SELECT 1 AS Tag,

NULL AS Parent,

OrderID AS [Order!1!OrderID],

OrderDate AS [Order!1!OrderDate],

TotalPrice AS [Order!1!TotalPrice],

NULL AS [Item!2!OrderDetailID],

NULL AS [Item!2!ProductID],

NULL AS [Item!2!Quantity],

NULL AS [Item!2!Price]

FROM Orders

UNION ALL

SELECT 2 AS Tag,

1 AS Parent,

OrderDetails.OrderID,

NULL,

NULL,

OrderDetailID,

ProductID,

Quantity,

Price

FROM OrderDetails

JOIN Orders ON OrderDetails.OrderID = Orders.OrderID

ORDER BY [Order!1!OrderID], [Item!2!OrderDetailID]

FOR XML EXPLICIT, ROOT('orders')

**Вывод**

В результате выполнения лабораторной работы были успешно созданы и протестированы различные объекты базы данных, включая таблицы, а также были выполнены сложные запросы для формирования данных в формате XML. Это позволило получить практические навыки работы с базой данных, создания и управления структурой данных, а также форматирования и извлечения данных в виде XML. Полученные знания можно применить для разработки сложных систем, требующих гибкой работы с данными и их представлением в различных форматах.