МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

(КНИТУ-КАИ)

# Институт компьютерных технологий и защиты информации

(наименование института (факультета), филиала)

# Кафедра Прикладной математики и информатики

(наименование кафедры)

# (01.03.02) Прикладная математика и информатика

(шифр и наименование направления подготовки (специальности))

# Лабораторная работа №1

по дисциплине «Программирование на Transact SQL» Тема: «Формирование XML при использовании FOR XML»

# Выполнил: студент группы 4317

Мохамед М.О

Сафин Андрей

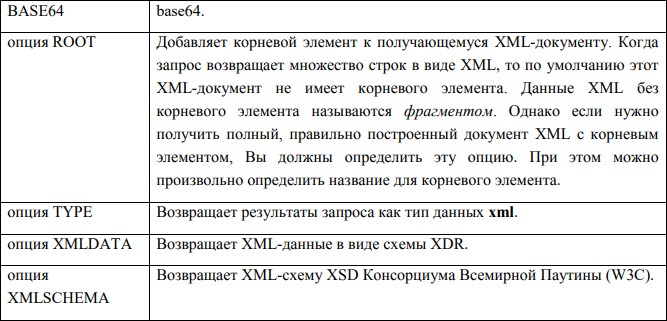
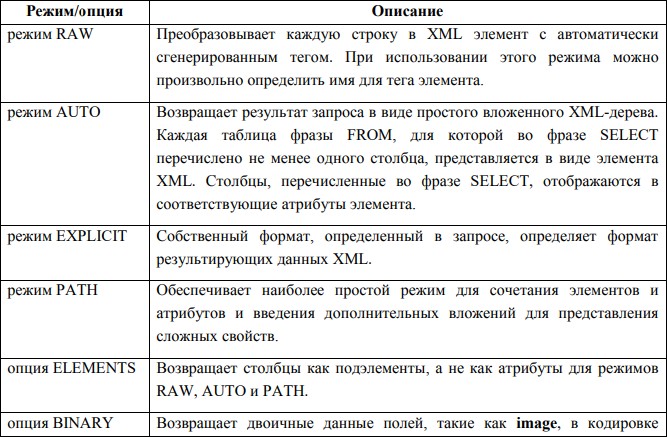
# Казань, 2023

## КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**For XML**

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 четырёх режимов: RAW, AUTO, EXPLICIT или PATH. Кроме того, Вы можете определить различные опции для управления выводом.

Обычно используемые режимы и опции фразы FOR XML описываются в следующей таблице:



Существует множество ситуаций, в которых необходимо получить данные в формате XML вместо набора строк. Например, как описано в следующих сценариях доступа к данным:

* Получение данных для публикации на Web-сайте. Получая данные в формате XML, Вы можете применить стилевые таблицы XSLT для передачи данных в HTML формате. Вы можете также применить различные стилевые таблицы к тем же самым данным XML, чтобы сгенерировать альтернативные форматы представления, чтобы поддержать различные устройства клиента, не переписывая логики доступа к данным.
* Получение данных для обмена с торговым партнёром. XML - естественный формат данных, которые Вы хотите послать торговому партнёру. Получая коммерческую информацию в XML формате, Вы можете легко объединить свои системы с внешними организациями, и не имеет значения, какие технологии данных они используют внутренне.

## Режим RAW

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.

Например, Получение данных в виде элементов В следующем примере показывается, как можно получить те же данные в виде элементов вместо атрибутов, определяя опцию ELEMENTS.

SELECT Books.book\_name, Books.year\_of\_publication,Books\_in\_use.date\_of\_issue FROM Books LEFT JOIN Books\_in\_use

ON Books.ID\_book = Books\_in\_use.ID\_book

ORDER BY date\_of\_issue

FOR XML RAW

## Режим AUTO

Запросы в режиме AUTO производят XML представления данных о сущностях. Опция AUTO имеет следующие особенности:

* Каждая строка, возвращаемая в соответствии с запросом, представляется элементом XML с тем же самым именем как у таблицы, из которой она была получена (или с псевдонимом, используемым в запросе).
* Каждое соединение JOIN в запросе преобразуется во вложенный элемент XML, уменьшая дублирование данных в получающемся фрагменте XML. Порядок операторов JOIN влияет на порядок вложенных элементов.
* Чтобы гарантировать, что дочерние элементы сопоставлены правильно с их родителем, используйте фразу ORDER BY, чтобы возвратить данные в правильном иерархическом порядке.
* Каждый столбец в результирующем наборе представлен атрибутом, если не определена опция ELEMENTS, когда каждый столбец представлен дочерним элементом.
* Агрегированные столбцы и фраза GROUP BY не поддерживаются в запросах в режиме AUTO (хотя можно использовать запросы в режиме AUTO, чтобы получить агрегированные данные из представления, которое использует фразу GROUP BY).

## Режим EXPLICIT

Иногда деловые документы, которые Вы должны обменяться с торговыми партнёрами, требуют формат XML, который не может быть получен при использовании режимов RAW или AUTO. Когда данные таблицы соответствуют элементу XML, таблица таблицы могут быть представлены как значение элемента, атрибут, дочерний элемент.

Универсальная таблица - табличное представление XML документа. Каждая строка в универсальной таблице представляет данные, которые будут представлены в виде элемента в получающемся документе XML.

Tag (Тег). Числовое значение, которое уникально идентифицирует тег для элемента, который содержит данные в этой строке.

Parent (Родитель). Числовое значение, которое идентифицирует непосредственный родительский тег для этого элемента.

Создание запроса для построения универсальной таблицы:

SELECT 1 AS Tag,

NULL AS Parent,

ID\_customers AS [User!1!ID\_customers], Customers.first\_name AS [User!1!Name!Element] FROM Customers

FOR XML EXPLICIT, ROOT

## Режим PATH

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.

Получение данных при использовании режима PATH

SELECT Customers.ID\_customers "@ID\_customers", Customers.first\_name "First\_Name",

Customers.phone\_number "Phone"

FROM Customers

FOR XML PATH, ROOT

## 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

## ЛИСТИНГ И РЕЗУЛЬТАТ РАБОТЫ

SELECT Books.book\_name, Books.year\_of\_publication,Books\_in\_use.date\_of\_issue FROM Books LEFT JOIN Books\_in\_use

ON Books.ID\_book = Books\_in\_use.ID\_book

ORDER BY date\_of\_issue

FOR XML



SELECT Books.book\_name, Books.year\_of\_publication,Books\_in\_use.date\_of\_issue FROM Books LEFT JOIN Books\_in\_use

ON Books.ID\_book = Books\_in\_use.ID\_book

ORDER BY date\_of\_issue

FOR XML AUTO, ROOT

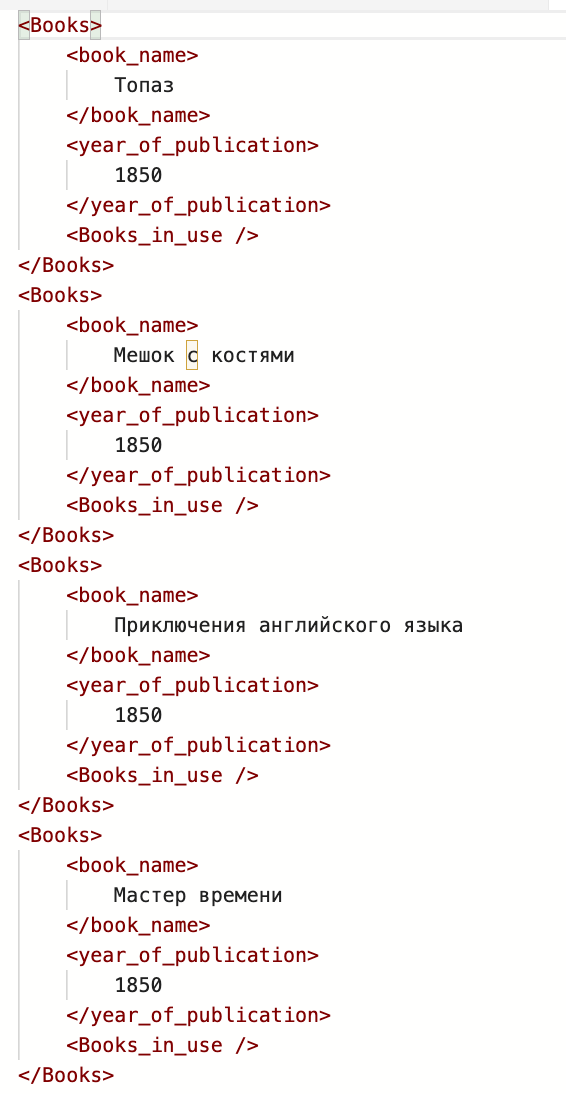


SELECT Books.book\_name, Books.year\_of\_publication,Books\_in\_use.date\_of\_issue FROM Books LEFT JOIN Books\_in\_use

ON Books.ID\_book = Books\_in\_use.ID\_book

ORDER BY date\_of\_issue

FOR XML AUTO, ELEMENTS

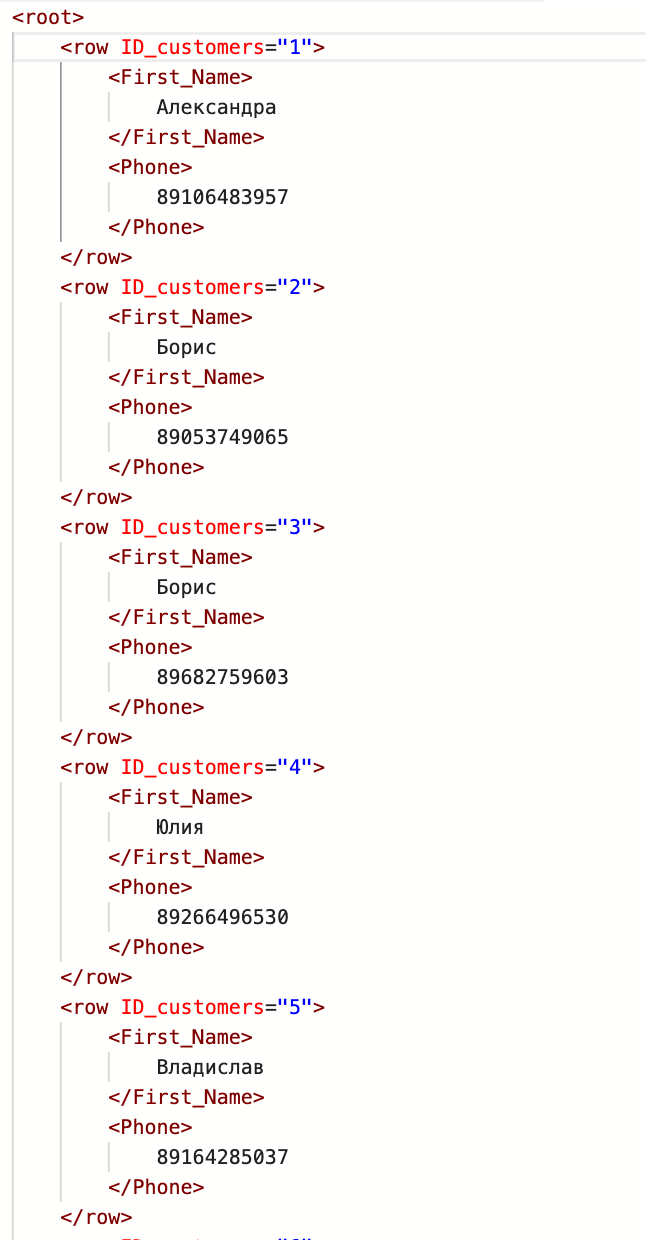


SELECT Customers.ID\_customers "@ID\_customers", Customers.first\_name "First\_Name",

Customers.phone\_number "Phone"

FROM Customers

FOR XML PATH, ROOT

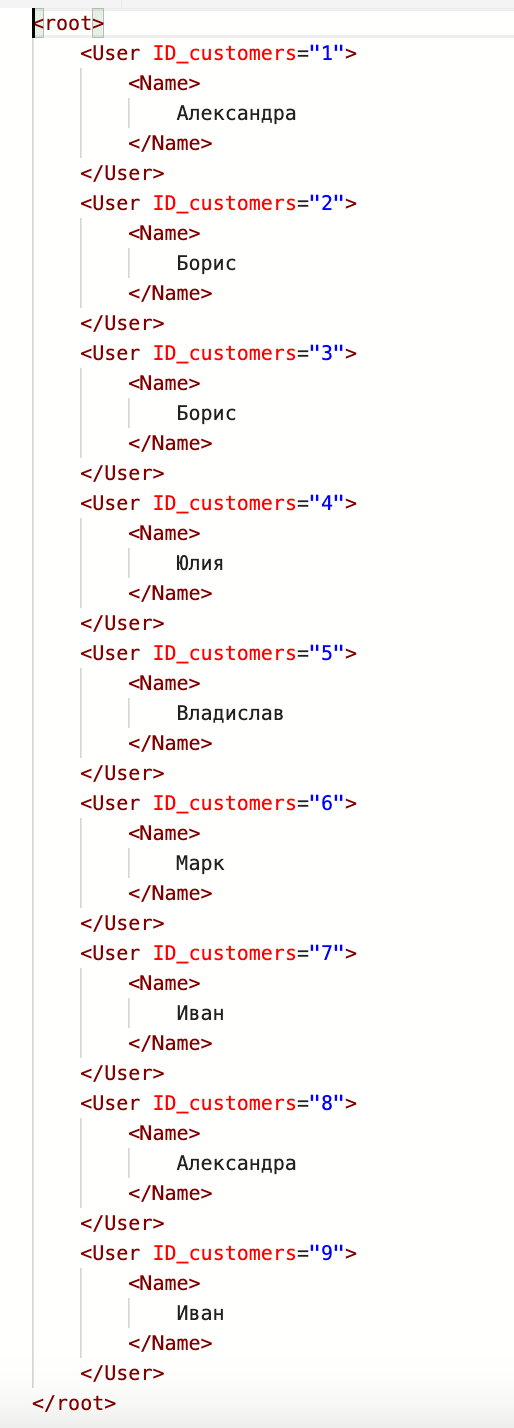


SELECT 1 AS Tag,

NULL AS Parent,

ID\_customers AS [User!1!ID\_customers], Customers.first\_name AS [User!1!Name!Element] FROM Customers

FOR XML EXPLICIT, ROOT



**Вывод:**

# В данной лабораторной работе я научился описывать назначение фразы FOR XML, описывать режим запросов RAW и объяснять их синтаксис, описывать режим запросов AUTO и их синтаксис, описывать режим запросов EXPLICIT и их синтаксис, описывать режим запросов PATH и их синтаксис, описывать синтаксис для получения вложенных XML-документов.