

*Кириллов Д.С.,  
Студент магистратуры,  
Казанский национальный исследовательский  
технологический университет  
Россия, г. Казань*

*Молостов Д.Д.,  
Студент магистратуры,  
Казанский национальный исследовательский  
технологический университет  
Россия, г. Казань*

*Мертинс Г.Р.,  
Студент магистратуры,  
Казанский национальный исследовательский  
технологический университет  
Россия, г. Казань*

*Научный руководитель: Старыгина С.Д.,  
кандидат педагогических наук, доцент  
Казанский национальный исследовательский  
технологический университет  
Россия, г. Казань*

## **КАК ПИСАТЬ ЧИТАЕМЫЕ T-SQL ЗАПРОСЫ**

*Аннотация: В статье рассматриваются ключевые положения в области правильной организации структуры T-SQL запросов, включая общепринятые сложившиеся рекомендации в данной области. В ходе анализа подробно описывается важность грамотно структурированных запросов и*

*описательных комментариев к ним. Помимо этого, в статье затронуты процессы оптимизации работы с базой данных.*

**Ключевые слова:** *SQL, SQL Server, T-SQL, T-SQL запросы, структура кода, удобочитаемость кода, структура запросов, структура кода, описательные комментарии, SSMS.*

**Annotation:** *The article discusses the key provisions in the field of the correct organization of the structure of T-SQL queries, including generally accepted established recommendations in this area. The analysis describes in detail the importance of well-structured queries and descriptive comments on them. In addition, the article touches on the optimization processes of working with the database.*

**Key words:** *SQL, SQL Server, T-SQL, T-SQL queries, code structure, code readability, query structure, code structure, descriptive comments, SSMS.*

Основное ожидание от кода состоит конечно в том, чтобы он работал правильно и выдавал ожидаемые результаты. Однако эти выдаваемые результаты не являются достаточным критерием, чтобы сказать, что этот код хороший и грамотно написан. Одним из таких критериев является читабельность написанного кода. Современные приложения сегодня в большинстве случаев разрабатываются командами, включающими в себя множество разработчиков. Следовательно, над одним и тем же кодом в разные периоды времени или даже одновременно может работать несколько разработчиков, поэтому часто возникает потребность чтения, проверки, редактирования или обслуживания другими разработчиками. На этом этапе понимание кода очень сильно зависит от его удобочитаемости.

Как упоминалось выше, запрос T-SQL может быть перечитан еще много раз и многими другими разработчиками баз данных, когда потребуется его редактирование. Однако, когда запрос не читается, разработчик базы данных потратит гораздо больше времени на чтение запроса, чем на его изменение. В

худшем случае разработчики даже могут неправильно понять код и изменить его так, как не предполагалось изначально. Следующие правила помогают повысить удобочитаемость запроса T-SQL и избежать множества таких ситуаций:

- Правильное формирование запроса;
- Использование комментариев;
- Применение принципа неповторности.

Правильное форматирование запросов можно определить как преобразование кодов запросов в хорошо организованный структурированный и легко читаемый формат. Правильно отформатированный структурированный запрос легко читается и обрабатывается разработчиками, что облегчает следующие дальнейшие работы:

- Ведение кодов приложений;
- Процесс отладки и исправления ошибок.

Чтобы сделать код более читабельным и правильно структурированным, можно использовать различные онлайн-инструменты для форматирования SQL или сторонние надстройки в формате SQL для SQL Server Management Studio (SSMS).

«Код должен быть самодокументируемым» — это очень известный лозунг среди разработчиков, он означает, что вместо использования комментариев код должен быть сам по себе достаточно разборчивым, чтобы уменьшить потребность в комментариях. То есть, такой подход заставляет разработчиков писать более читаемый код, но у каждого разработчика может быть разное поведение при кодировании, поэтому добавление комментариев всегда повышает читаемость кода.

Добавление некоторых описательных комментариев в начале или после T-SQL запроса также улучшит читаемость кода и запросов. Этот комментарий может вкратце пояснить, какие записи запросов выборки, бизнес-логики, описания таблиц и схем используются в конкретных процедурах. Основная

характеристика комментария заключается в том, что он может кратко и ясно объяснить детали конкретного запроса.

После разработки схемы базы данных еще одним важным моментом является ее документирование. Иногда этот шаг можно пропустить, так как разработчики и администраторы баз данных считают его трудоемким и ненужным. Тем не менее, документация схемы является хорошим руководством для следующих разработчиков, когда им будет необходимо решить некоторые возникшие проблемы или процессы обновления, связанные с проектируемой базой данных. По крайней мере, хотя бы добавление описаний столбцов в базы данных поможет понять цель использования этого столбца в таблице, к которой он принадлежит.

Еще одним преимуществом описаний столбцов является то, что мы можем добавлять их в качестве комментариев к запросам для идентификации имен столбцов. Описания столбцов очень легко создавать с помощью SSMS.

Мы можем назначать временные имена таблицам или столбцам во время выполнения запроса, и это тоже улучшает читаемость запроса. Псевдоним будет очень полезен, когда истинное имя таблицы очень длинное и непонятное, поэтому можно преобразовать его в более разборчивое и понятное для восприятия. Так, например, если мы работаем с базой данных SAP, исходные имена таблиц будет не так легко понять.

Основная цель принципа неповторности — уменьшить повторение написания одних и тех же фрагментов кода. Мы можем использовать определяемые пользователем функции для реализации этого принципа в T-SQL запросе, поскольку определяемые пользователем функции позволяют нам инкапсулировать запросы и вызывать их в дальнейшем откуда угодно. В основном, существует 3 различных типа определяемых пользователем функций:

- Функция со скалярным значением – она возвращает одно значение.

- Функция Multi-Statement Table-Valued - возвращает таблицу в качестве выходных данных, и эта структура выходной таблицы может быть определена пользователем.

- Функция встроенного табличного значения — это определяемая пользователем функция, которая возвращает табличный тип данных, а также может принимать параметры.

Оператор «SELECT \*» возвращает все столбцы запрашиваемой таблицы (таблиц), использовать его конечно очень легко. Однако там, где это возможно – всегда необходимо явно указывать имена столбцов в наших запросах, а не запрашивать все имеющиеся. Это правило используется для того, чтобы уточнить, какие столбцы необходимы для запроса, чтобы также повысить удобочитаемость запроса и кода. Конечно, есть и другие преимущества в явном запросе необходимых столбцов, а не использовании оператора «SELECT \*»:

- Предотвращение лишних ненужных операций ввода-вывода.
- Уменьшение сетевого трафика.
- Уменьшение времени ответа на запрос.

Таким образом, в этой статье мы познакомились с основными полезными методологиями, которые помогают увеличить и улучшить читаемость T-SQL запросов.

#### **Использованные источники:**

1. Пржиялковский, В. В. Введение в Oracle SQL / В.В. Пржиялковский. - М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2013. - 320 с.

2. Молинаро, Э. SQL. Сборник рецептов / Э. Молинаро. - М.: Символ-плюс, 2013. - 820 с.

3. Мартишин, С. А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench. Учебное пособие / С.А.

Мартишин, В.Л. Симонов, М.В. Храпченко. - М.: Форум, Инфра-М, 2015. - 160 с.

4. Маркин, А.В. Построение запросов и программирование на SQL. Учебное пособие / А.В. Маркин. - М.: Диалог-Мифи, 2014. - 384 с.

5. Дунаев, В.В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2016. - 288 с.