**Лабораторная работа №2\_3. Создание запросов на языке Transact-SQL**

Для выполнения работы необходимо подключить БД, которая была создана и заполнена данными в процессе выполнения практической работы Основной целью данной работы является изучение особенностей использования оператора SELECT-SQL при разработке запросов в среде SQL Server Management Studio, а также рассмотрение некоторых особенностей реализации оператора SELECTSQL в языке Transact-SQL (T-SQL).

Полный синтаксис инструкции SELECT сложен, однако основные предложения можно вкратце описать следующим образом:

[WITH { [XMLNAMESPACES ,] [ <common\_table\_expression> ] } ]

SELECT select\_list [ INTO new\_table ]

[FROM table\_source ]

[WHERE search\_condition ]

[GROUP BY group\_by\_expression ]

[HAVING search\_condition ]

[ORDER BY order\_expression [ ASC | DESC ] ]

Обработка элементов запроса SELECT выполняется в следующей последовательности:

1. FROM – определяет имена используемых таблиц;

2. WHERE – фильтрует строки таблицы в соответствии с заданными условиями;

3. GROUP BY – группирует строки, имеющие одинаковые значения в указанном столбце;

4. HAVING – фильтрует группы строк в соответствии с указанным условием;

5. SELECT – форматирует выходные данные;

6. ORDER BY – сортирует результаты выполнения запроса.

Порядок предложений в запросе SELECT не может быть изменен. Предложения SE-LECT и FROM являются обязательными, присутствие остальных зависит от контекста.

В предложении SELECT указывается список столбцов, которые должны быть возвращены запросом. Можно указать исходные элементы или вычисляемые поля во время выполнения запроса.

Конструкция DISTINCT | ALL исключает / разрешает вывод повторяющихся строк. Конструкция ALL используется по умолчанию.

\* означает вывод всех столбцов указанной таблицы. В случае, если выборка производится из нескольких таблиц, перед символом звездочки может указываться имя таблицы.

SQL-запрос может содержать вычисляемые столбцы, значения которых могут определятся на основе значений данных, хранящихся в БД конструкции. Вычисляемым столбцам следует давать название с помощью ключевого слова AS.

Вычисляемый столбец можно создать как: <Новое поле> = <выражение>

Если название столбца состоит из нескольких слов, разделенных пробелами, следует их записать в квадратных скобках: [].

Сортировка данных выполняется с помощью команды ORDER BY, которая добавляется в конец запроса, после чего перечисляется список столбцов. Для каждого столбца указывается тип сортировки ASC | DESC (ascending – по возрастанию | descending – по убыванию). ASC – по умолчанию, можно не указывать.

Конструкция TOP <N> позволяет выбрать определенное количество строк из таблицы. Дополнительный оператор PERCENT позволяет выбрать процентное количество строк из таблицы. Дополнительный оператор WITH TIES позволяет выбрать все строки с такими же свойствами.

Конструкция OFFSET <N> ROWS указывает число строк, которые необходимо пропустить, прежде чем будет начат возврат строк из выражения запроса.

Конструкция FETCH NEXT <N> ROWS ONLY указывает число строк, возвращаемых после обработки предложения OFFSET.

На языке T-SQL регистр не имеет значение (case insensitive).

Полезные ссылки: https://metanit.com/sql/mysql/3.2.php

Рассмотрим последовательность действий по созданию и выполнению запроса, позволяющего обрабатывать данные с помощью оператора SELECT- SQL с помощью утилиты ISQL на примере запроса 1.

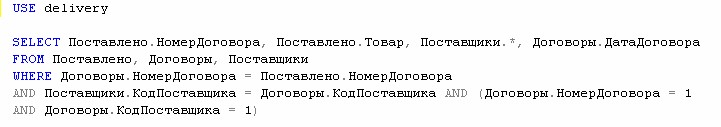
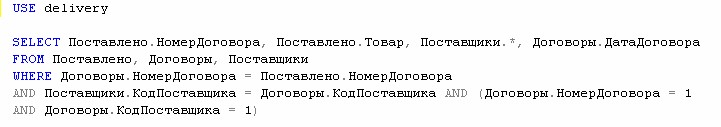
**Запрос 1**

# Условие

Вывести на экран список товаров, поставленных поставщиком 1 (Иванов И.И.) по договору 1.

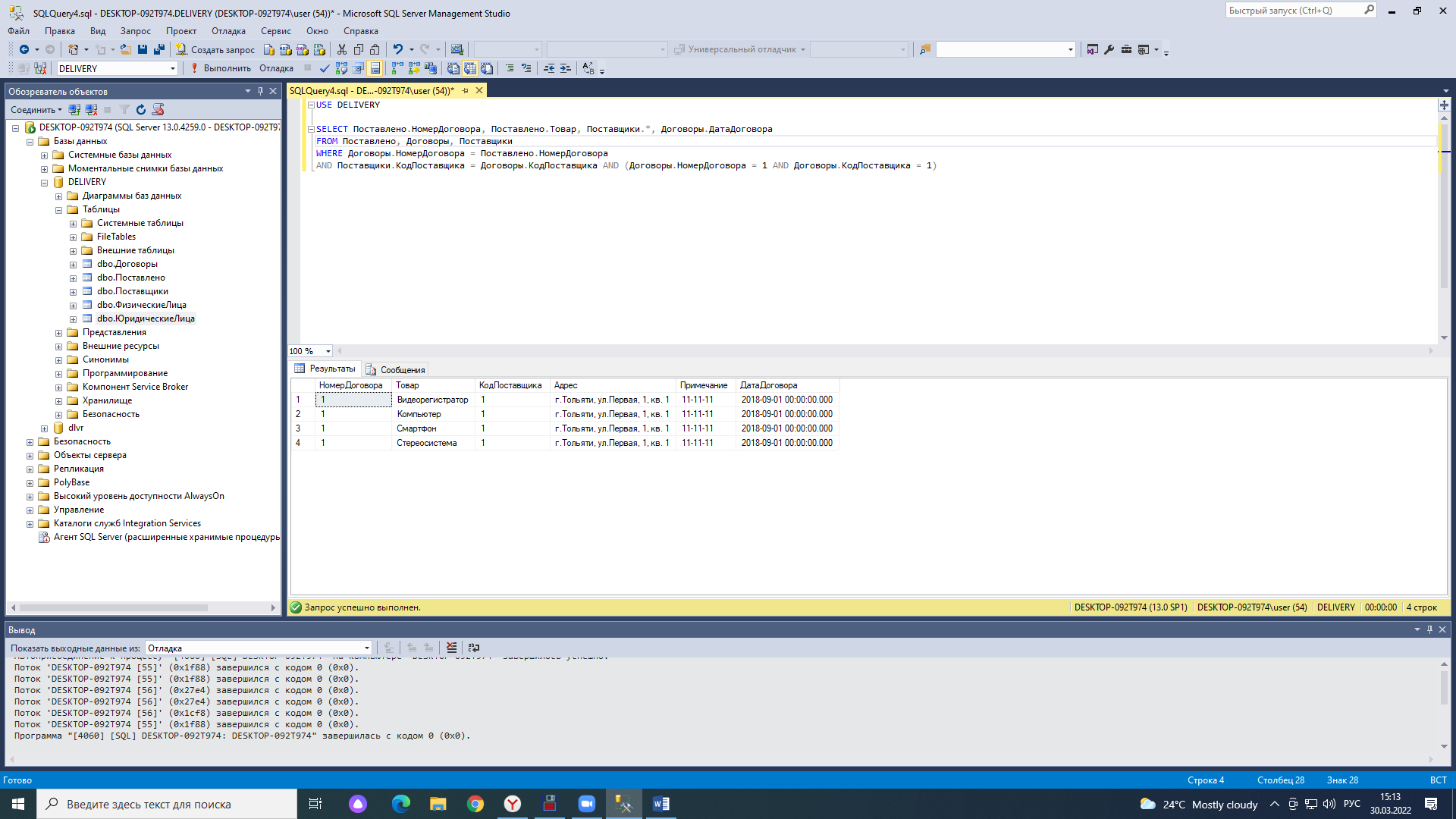
**Создание и выполнение запроса.**

1. На панели инструментов нажать кнопку Создать запрос (New Query).
2. Ввести текст запроса, приведенный на рис. 3.1.



## Рисунок 3.1

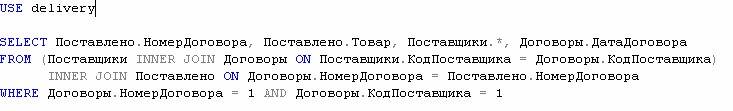
1. Нажать кнопку «!Выполнить». В том случае, если в тексте запроса нет ошибок, будет выведен результат запроса.

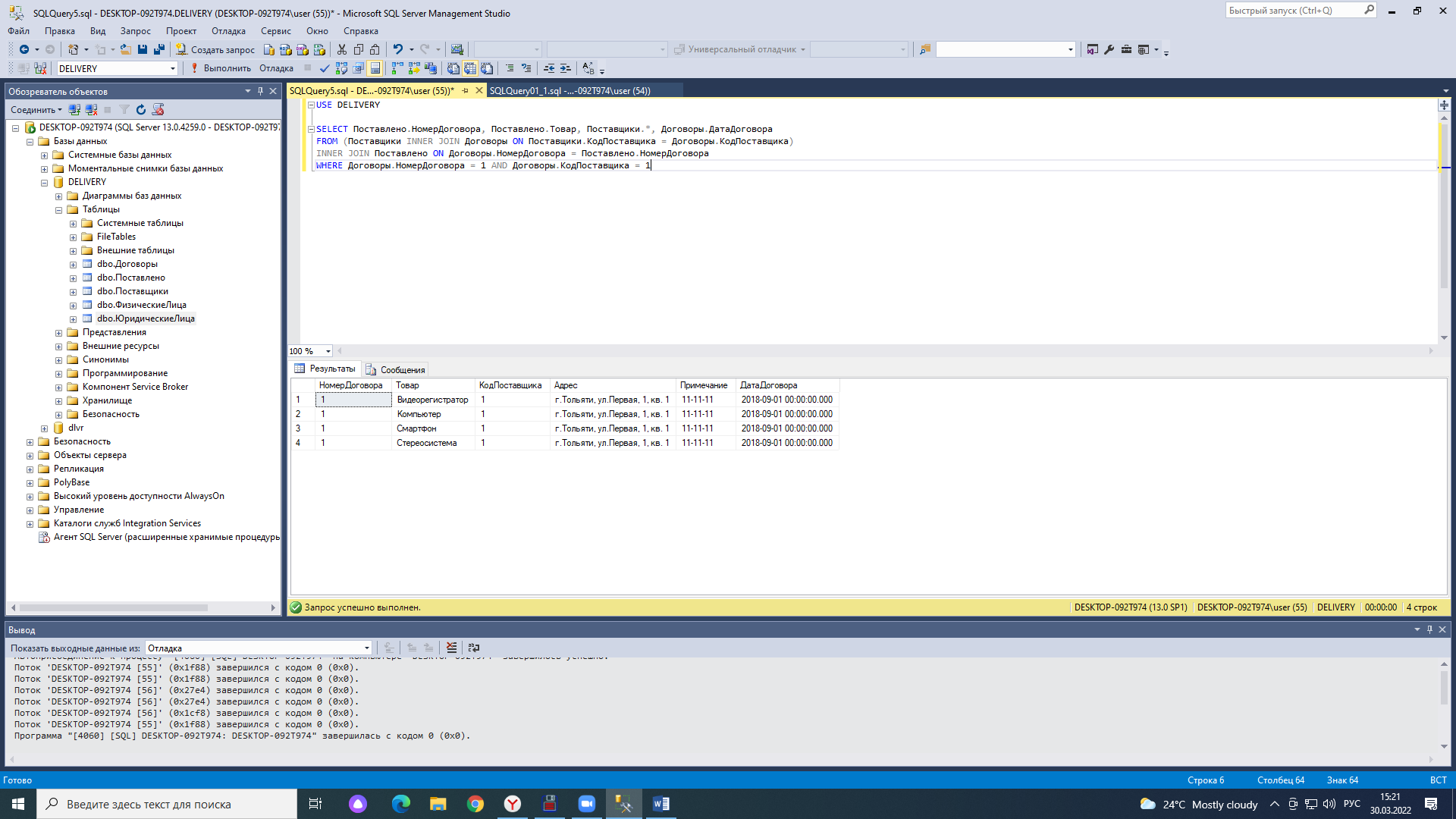


1. Текст запроса можно сохранить в виде файла (например, SQLQuery01\_1.sql). В том случае, если в дальнейшем этот запрос нужно будет выполнить повторно или изменить, можно открыть файл запроса. Для этого в главном меню нужно выбрать пункт Файл (File), а затем в вертикальном меню выбрать пункт Открыть (Open), подпункт Файл (File) и выбрать соответствующий файл.

Как видно из текста запроса, этот запрос является многотабличным, причем таблицы соединяются на основе использования естественного соединения. В случае использования открытого соединения этот запрос имел бы вид (рис. 3.2).

Этот запрос также необходимо создать и выполнить для проверки работоспособности, а затем сохранить в файле с именем SQLQuery01\_2.sql





Создание и выполнение остальных запросов выполняется аналогично. Поэтому далее будет приведено назначение каждого запроса и текст.

# Запрос 2

Вывести на экран список товаров, поставленных поставщиком 1 (ЧП Иванов И.И.) в период с 05/09/2018 по 12/09/2018.

Текст запроса приведен на рис. 3.3.

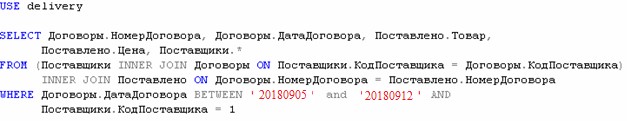
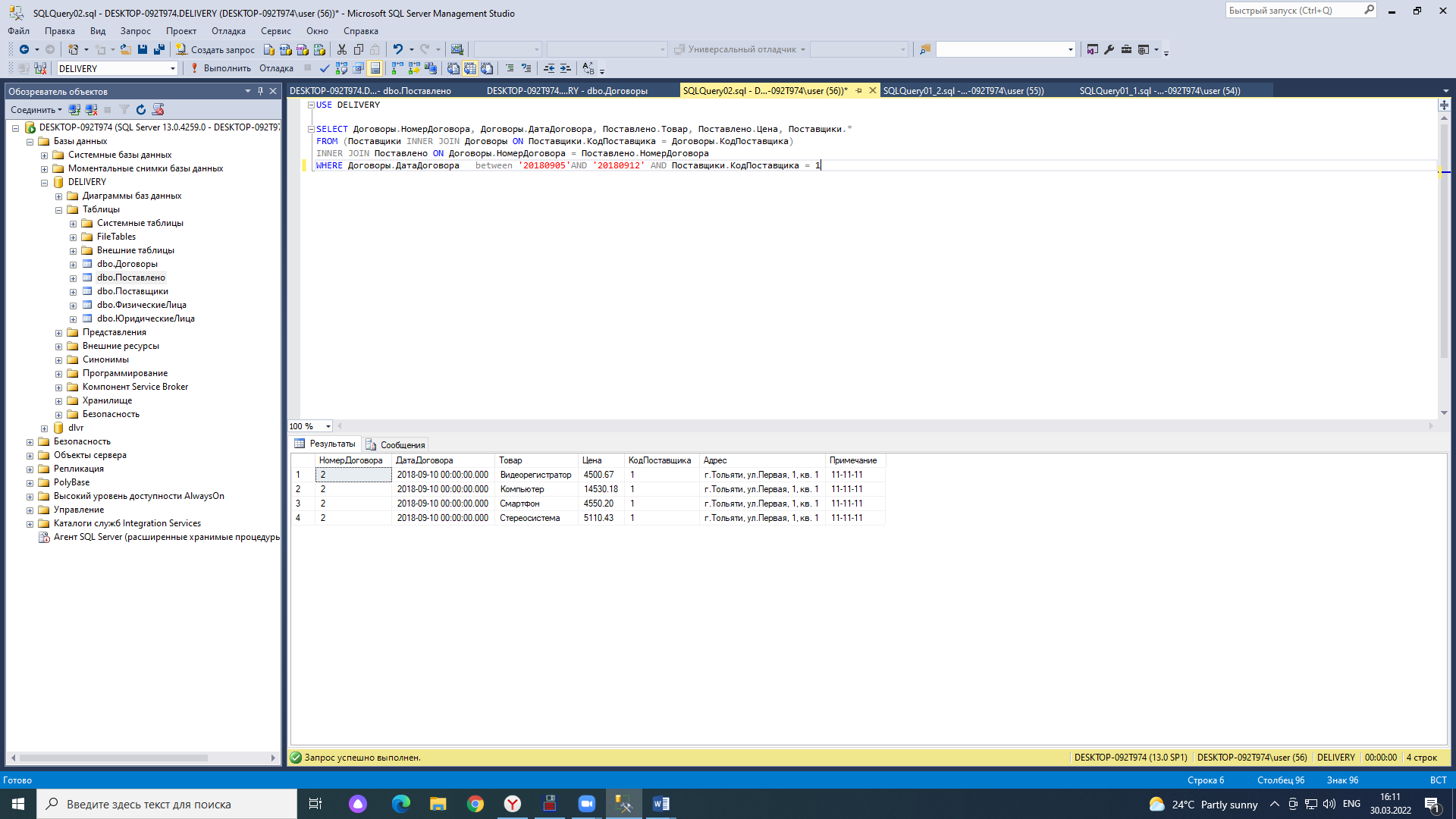


Рисунок 3.3



Запрос можно сохранить в файле с именем SQLQuery02.sql

# Запрос 3

Вывести на экран список товаров, поставленных в 9 месяце 2018 года с выводом наименования поставщика и даты поставки. Текст запроса приведен на рис. 3.4

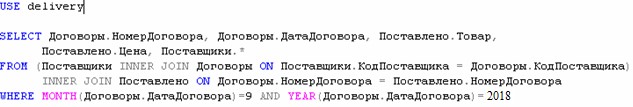


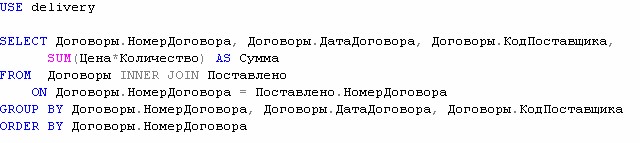
Рисунок 3.4

Запрос можно сохранить в файле с именем SQLQuery03.sql

# Запрос 4

Вывести на экран список договоров (номер, дата, название) и общую сумму по каждому договору (размер партии умножить на цену за штуку и просуммировать по договору). Список должен быть отсортирован в порядке возрастания номеров договоров.

Текст запроса приведен на рис. 3.5.



5

Запрос можно сохранить в файле с именем SQLQuery04.sql

# Запрос 5

Вывести на экран список договоров (номер, дата, название) и общую сумму по каждому договору (размер партии умножить на цену за штуку и просуммировать по договору). Список должен быть отсортирован в порядке возрастания общих сумм по каждому договору. После этого на список должно быть наложено условие фильтрации, состоящее в исключении из результата запроса записей, для которых номер договора больше 3. Текст запроса приведен на рис. 3.6.

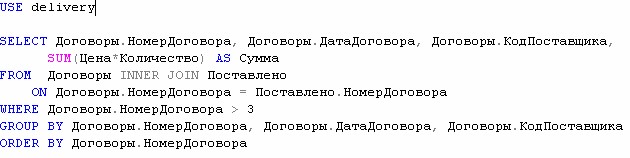


Рисунок 3.6

Запрос можно сохранить в файле с именем SQLQuery05.sql

# Запрос 6

Вывести на экран сведения о наибольшей по размеру партии товара во всех договорах с указанием поставщика, а также номера и даты договора. Текст запроса приведен на рис. 3.7.

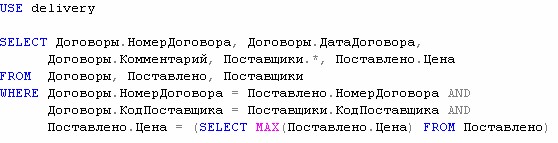
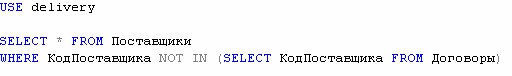


Рисунок 3.7

Запрос можно сохранить в файле с именем SQLQuery06.sql

# Запрос 7

Вывести на экран список поставщиков (наименование и код), с которыми не было заключено ни одного договора. Текст запроса приведен на рис. 3.8.



Запрос можно сохранить в файле с именем SQLQuery07.sql

# Запрос 8

Вывести на экран список наименований поставленных товаров с указанием средней цены поставки за единицу (вне зависимости от поставщика). Текст запроса приведен на рис. 3.9.

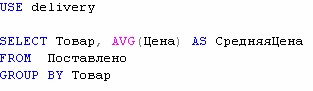


Рисунок 3.9

Запрос можно сохранить в файле с именем SQLQuery08.sql

# Запрос 9

Вывести на экран список товаров (наименование, количество и цена, поставщик), для которых цена за единицу больше средней. Текст запроса приведен на рис. 3.10.

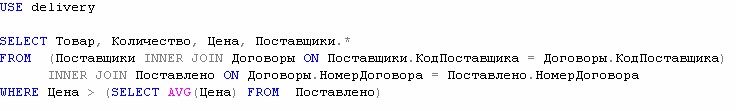


Рисунок 3.10

Запрос можно сохранить в файле с именем SQLQuery09.sql

# Запрос 10

Вывести на экран сведения о пяти самых дорогих товарах (наименование, цена за единицу, поставщик).

Текст запроса приведен на рис. 3.11

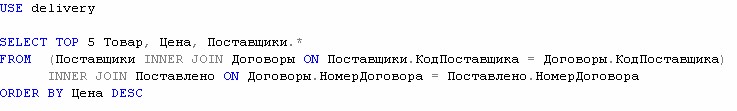


Рисунок 3.11

Запрос можно сохранить в файле с именем SQLQuery10.sql

# Запрос 11

Сформировать список поставщиков с указанием кода, адреса и данных поставщика. При формировании данных поставщика для поставщиков – физических лиц вывести фамилию и инициалы, для поставщиков – юридических лиц – наименование.

Текст запроса приведен на рис. 3.12.

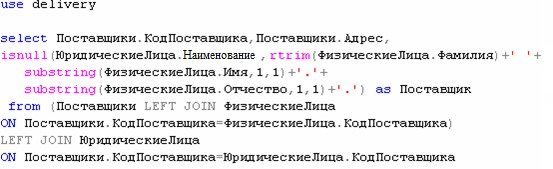


Рисунок 3.12

Запрос можно сохранить в файле с именем SQLQuery11.sql.

# Запрос 12

Сформировать список договоров (с указанием номера, даты поставки и данных о поставщике), общее количество поставленных товаров и общую сумму по каждому договору. Для поставщиков – физических лиц вывести фамилию и инициалы, для поставщиков – юридических лиц – ОГРН. В результат запроса должны быть включены только те договоры, на основании которых товары действительно поставлялись (т.е. в результат запроса не должны попасть так называемые «пустые» договоры)

Текст запроса приведен на рис. 3.13.

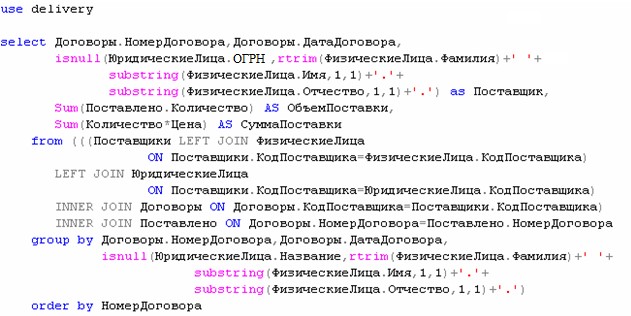


Рисунок 3.13

Запрос можно сохранить в файле с именем SQLQuery12.sql.

# Запрос 13

Сформировать список товаров (с указанием номера договора и даты поставки), поставленных поставщиками 1 (ЧП Петров П.П.) и 2 (ООО «Интерфрут»).

**Примечание.** Данный запрос иллюстрирует особенности использования операции объединения (UNION). Нетрудно заметить, что данный запрос может быть легко реализован без использования операции объединения. Текст запроса приведен на рис. 3.14.

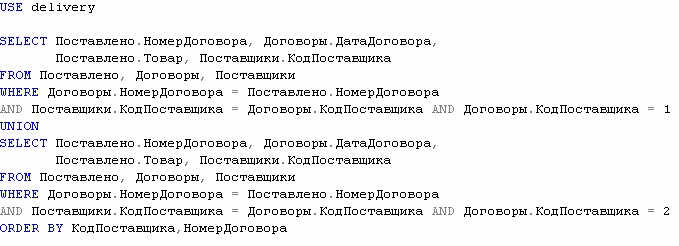


Рисунок 3.14

Запрос можно сохранить в файле с именем SQLQuery13.sql.

# Запрос 14

Сформировать номенклатуру товаров (т.е. список наименований товаров), которые поставлялись только поставщиком 1 (ЧП Петров П.П.), или только поставщиком 2 (ООО «Интерфрут»), или и поставщиком 1, и поставщиком 2. Текст запроса приведен на рис. 3.15.

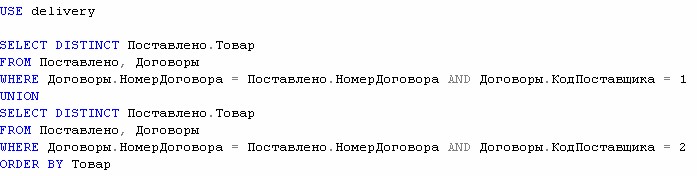


Рисунок 3.15

Запрос можно сохранить в файле с именем SQLQuery14.sql

# Запрос 15

Сформировать номенклатуру товаров (т.е. список наименований товаров), которые поставлялись и поставщиком 1 (ЧП Петров П.П.), и поставщиком 2 (ООО «Интерфрут»).

**Примечание.** Данный запрос иллюстрирует особенности использования операции пересечения (INTERSECT).

Текст запроса приведен на рис. 3.16.

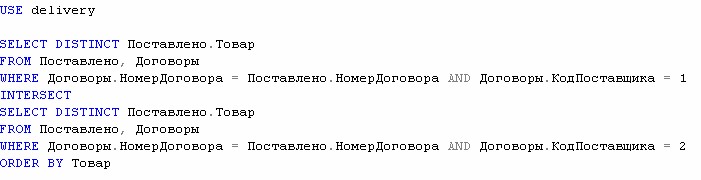


Рисунок 3.16

Запрос можно сохранить в файле с именем SQLQuery15.sql

# Запрос 16

Сформировать номенклатуру товаров (т.е. список наименований товаров), которые поставлялись поставщиком 1 (ЧП Петров П.П.), но поставлялись поставщиком 2 (ООО «Интерфрут»).

**Примечание.** Данный запрос иллюстрирует особенности использования операции разности (EXCEPT).

Текст запроса приведен на рис. 3.17.

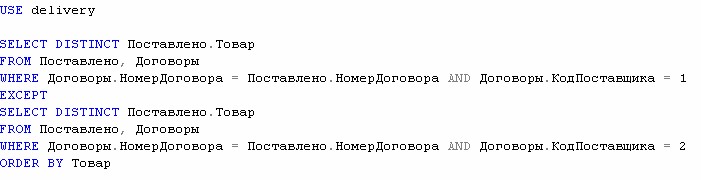


Рисунок 3.17

Запрос можно сохранить в файле с именем SQLQuery16.sql

# Запрос 17

Сформировать список товаров, который должен отражать частоту поставок товаров. В список включить только товары, которые поставлялись более одного раза. Список должен быть отсортирован в порядке убывания частоты поставок. Текст запроса приведен на рис. 3.18.

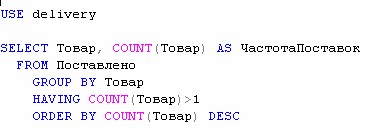


Рисунок 3.18

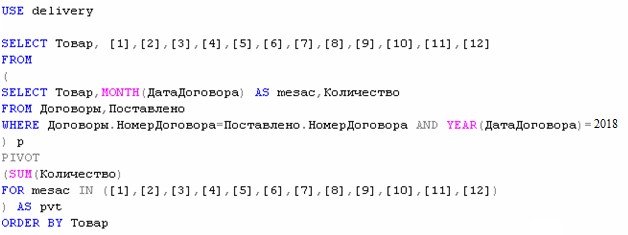
Запрос можно сохранить в файле с именем SQLQuery17.sql.

# Запрос 18

Сформировать данные о количественной динамике поставок товаров в течение 2018 года. Данные должны быть агрегированы помесячно и представлены в виде таблицы, строками которой являются названия товаров, а столбцами – номера месяцев 2018 года. На пересечении строки и столбца должно отображаться количество данного товара, поставленного в данном месяце.

**Примечание.** Данный запрос иллюстрирует особенности создания и использования перекрестного запроса средствами языка Transact-SQL.

Текст запроса приведен на рис. 3.19. Запрос можно сохранить в файле с именем SQLQuery18\_1.sql.



## Рисунок 3.19

Приведенный результат запроса может быть неудобным для восприятия (например, из-за наличия значений NULL). Этот недостаток может быть устранен, например, путем замены значений NULL на 0. Текст измененного запроса приведен на рис. 3.20. Запрос можно сохранить в файле с именем SQLQuery18\_2.sql.

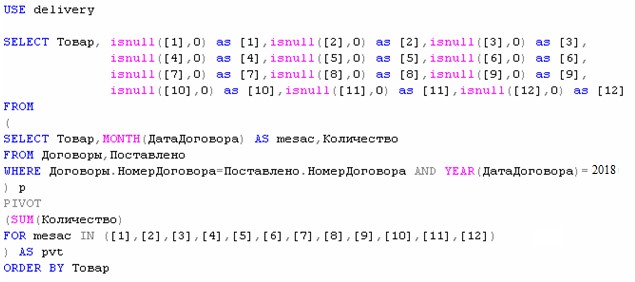
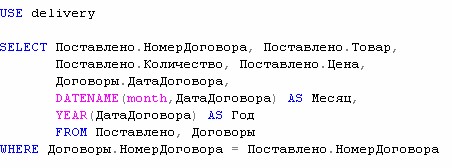


Рисунок 3.20

# Запрос 19

Сформировать список поставленных товаров. Для каждого товара в этом списке должны быть указаны следующие данные: номер договора, название товара, количество единиц, цена за единицу, дата поставки, название месяца и номер года.

Текст запроса приведен на рис. 3.21.



## Рисунок 3.21

Запрос можно сохранить в файле с именем SQLQuery19\_1.sql

Как видно из результата запроса, формально требование включения в результат запроса наименования месяца выполнено путем использования встроенной функции DATENAME(). Однако такие названия месяцев не всегда удобны для восприятия. Может возникнуть требование их замены на русскоязычные. Эту проблему можно решить путем разработки пользовательской функции, конвертирующей названия месяцев. Этот подход несколько более трудоемкий. Другим вариантом решения проблемы может быть использование в запросе функции CASE языка Transact-SQL. Текст такого запроса приведен на рис. 3.22.

Запрос можно сохранить в файле с именем SQLQuery19\_2.sql



## Рисунок 3.22

**Составить отчет по проделанной работе.**