Lab 2
Using the SELECT statement - SQL to process data

To perform the work, you need to connect a database that was created and filled with data in the course of laboratory work 1. The main purpose of this work is to study the peculiarities of using the SELECT-SQL statement when developing queries in the SQL Server Management Studio environment, as well as to consider some of the features of the statement implementation SELECT-SQL in Transact-SQL (T-SQL).

Let's consider the sequence of actions for creating and executing a query that allows you to process data using the SELECT-SQL statement using query 1 as an example.

Request 1

Condition

Display a list of goods supplied by supplier 1 (private entrepreneur Ivanov I.I.) under contract 1.

Creation and execution of a request.

1. On the toolbar, click the New Query button
2. Enter the request text shown in Figure 2.1

```
USE delivery

SELECT Поставлено.НомерДоговора, Поставлено.Товар, Поставщики.*, Договоры.ДатаДоговора
FROM Поставлено, Договоры, Поставщики
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора
AND Поставщики.КодПоставщика = Договоры.КодПоставщика AND (Договоры.НомерДоговора = 1
AND Договоры.КодПоставщика = 1)
```

Figure 2.1

3. Press the "Execute" button. In the event that the request text does not contain errors, the result of the request will be displayed. This result may look like (Figure 2.2).

Results | Messages

	НомерДоговора	Товар	КодПоставщика	Адрес	Примечание	ДатаДоговора
1	1	Видеомагнитофон	1	г.Харьков, пр. Ленина, 55, к.108	тел. 32-18-44	1999-09-01 00:00:00.000
2	1	Компьютер	1	г.Харьков, пр. Ленина, 55, к.108	тел. 32-18-44	1999-09-01 00:00:00.000
3	1	Магнитофон	1	г.Харьков, пр. Ленина, 55, к.108	тел. 32-18-44	1999-09-01 00:00:00.000
4	1	Стереосистема	1	г.Харьков, пр. Ленина, 55, к.108	тел. 32-18-44	1999-09-01 00:00:00.000
5	1	Телевизор	1	г.Харьков, пр. Ленина, 55, к.108	тел. 32-18-44	1999-09-01 00:00:00.000

Figure 2.2

4. The request text can be saved as a file (for example, SQLQuery01_1.sql). In the event that in the future this request needs to be repeated or changed, you can open the request file. To do this, select the File item in the main menu, and then select the Open item in the vertical menu, the File sub-item and select the appropriate file.

As you can see from the text of the query, this query is multi-table, and the tables are joined using a natural join. In the case of using an open connection, this request would have the form (Figure 2.3). This query also needs to be created and executed to validate it, and then saved to a file named SQLQuery01_2.sql

```
USE delivery

SELECT Поставлено.НомерДоговора, Поставлено.Товар, Поставщики.*, Договоры.ДатаДоговора
FROM (Поставщики INNER JOIN Договоры ON Поставщики.КодПоставщика = Договоры.КодПоставщика)
      INNER JOIN Поставлено ON Договоры.НомерДоговора = Поставлено.НомерДоговора
WHERE Договоры.НомерДоговора = 1 AND Договоры.КодПоставщика = 1
```

Figure 2.3

The creation and execution of other queries is carried out in the same way. Therefore, only the condition of each request and its text will be given below.

Attention! Everything **considered requests must to be successful (i.e., as a result of the query execution, one or several records should be displayed). The absence of a query result is a sign of errors in the construction of the query, a mismatch of the query with the available data, etc. Such a request needs to be analyzed and verified.**

Request 2

Display a list of goods delivered by supplier 1 (Ivanov I.I.'s private enterprise) in the period from 05/09/1999 to 12/09/1999.

The request text is shown in Figure 2.4.

```
USE delivery

SELECT Договоры.НомерДоговора, Договоры.ДатаДоговора, Поставлено.Товар,
       Поставлено.Цена, Поставщики.*
FROM (Поставщики INNER JOIN Договоры ON Поставщики.КодПоставщика = Договоры.КодПоставщика)
       INNER JOIN Поставлено ON Договоры.НомерДоговора = Поставлено.НомерДоговора
WHERE Договоры.ДатаДоговора BETWEEN '19990905' and '19990912' AND
       Поставщики.КодПоставщика = 1
```

Figure 2.4

The query can be saved in a file named SQLQuery02.sql

Request 3

Display the list of goods delivered in the 9th month of 1999, displaying the name of the supplier and the delivery date.

The request text is shown in Figure 2.5.

```
USE delivery

SELECT Договоры.НомерДоговора, Договоры.ДатаДоговора, Поставлено.Товар,
       Поставлено.Цена, Поставщики.*
FROM (Поставщики INNER JOIN Договоры ON Поставщики.КодПоставщика = Договоры.КодПоставщика)
       INNER JOIN Поставлено ON Договоры.НомерДоговора = Поставлено.НомерДоговора
WHERE MONTH(Договоры.ДатаДоговора)=9 AND YEAR(Договоры.ДатаДоговора)=1999
```

Figure 2.5

The query can be saved in a file named SQLQuery03.sql

Request 4

Display a list of contracts (number, date, name) and the total amount for each contract (multiply the lot size by the price per piece and add up according to the contract). The list must be sorted in ascending order of contract numbers.

The request text is shown in Figure 2.6.

```
USE delivery

SELECT Договоры.НомерДоговора, Договоры.ДатаДоговора, Договоры.КодПоставщика,
       SUM(Цена*Количество) AS Сумма
FROM  Договоры INNER JOIN Поставлено
    ON Договоры.НомерДоговора = Поставлено.НомерДоговора
GROUP BY Договоры.НомерДоговора, Договоры.ДатаДоговора, Договоры.КодПоставщика
ORDER BY Договоры.НомерДоговора
```

Figure 2.6

The query can be saved in a file named SQLQuery04.sql

Request 5

Display the list of contracts (number, date, name) and the total amount for each contract (multiply the lot size by the price per piece and add up according to the contract). The list should be sorted in ascending order of the total amounts for each contract. After that, a filtering condition should be imposed on the list, which consists in excluding records for which the contract number is less than 4 from the query result.

The request text is shown in Figure 2.7.

```
USE delivery

SELECT Договоры.НомерДоговора, Договоры.ДатаДоговора, Договоры.КодПоставщика,
      SUM(Цена*Количество) AS Сумма
FROM  Договоры INNER JOIN Поставлено
    ON Договоры.НомерДоговора = Поставлено.НомерДоговора
WHERE Договоры.НомерДоговора > 3
GROUP BY Договоры.НомерДоговора, Договоры.ДатаДоговора, Договоры.КодПоставщика
ORDER BY Договоры.НомерДоговора
```

Figure 2.7

The query can be saved in a file named SQLQuery05.sql

Request 6

Display information about the largest batch of goods in all contracts, indicating the supplier, as well as the number and date of the contract.

The request text is shown in Figure 2.8.

```
USE delivery

SELECT Договоры.НомерДоговора, Договоры.ДатаДоговора,
      Договоры.Комментарий, Поставщики.*, Поставлено.Цена
FROM  Договоры, Поставлено, Поставщики
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора AND
      Договоры.КодПоставщика = Поставщики.КодПоставщика AND
      Поставлено.Цена = (SELECT MAX(Поставлено.Цена) FROM Поставлено)
```

Figure 2.8

The query can be saved in a file named SQLQuery06.sql

Request 7

Display a list of suppliers (name and code) with which no contracts have been concluded.

The request text is shown in Figure 2.9

```
USE delivery

SELECT * FROM Поставщики
WHERE КодПоставщика NOT IN (SELECT КодПоставщика FROM Договоры)
```

Figure 2.9

The query can be saved in a file named SQLQuery07.sql

Request 8

Display on the screen a list of the names of the goods supplied with the indication of the average delivery price per unit (regardless of the supplier).

The request text is shown in Figure 2.10.

```
USE delivery

SELECT Товар, AVG(Цена) AS СредняяЦена
FROM  Поставлено
GROUP BY Товар
```

Figure 2.10

The query can be saved in a file named SQLQuery08.sql

Request 9

Display a list of goods (name, quantity and price, supplier) for which the unit price is higher than the average.

The request text is shown in Figure 2.11.

```
USE delivery

SELECT Товар, Количество, Цена, Поставщики.*
FROM  (Поставщики INNER JOIN Договоры ON Поставщики.КодПоставщика = Договоры.КодПоставщика)
      INNER JOIN Поставлено ON Договоры.НомерДоговора = Поставлено.НомерДоговора
WHERE Цена > (SELECT AVG(Цена) FROM  Поставлено)
```

Figure 2.11

The query can be saved in a file named SQLQuery09.sql

Request 10

Display information about the five most expensive items (name, unit price, supplier).

The request text is shown in Figure 2.12.

```
USE delivery

SELECT TOP 5 Товар, Цена, Поставщики.*
FROM  (Поставщики INNER JOIN Договоры ON Поставщики.КодПоставщика = Договоры.КодПоставщика)
      INNER JOIN Поставлено ON Договоры.НомерДоговора = Поставлено.НомерДоговора
ORDER BY Цена DESC
```

Figure 2.12

The query can be saved in a file named SQLQuery10.sql

Request 11

Generate a list of suppliers indicating the code, address and supplier data. When generating supplier data for suppliers - individuals, output the surname and initials, for suppliers - legal entities - the name.

The request text is shown in Figure 2.13.

```
use delivery

select Поставщики.КодПоставщика,Поставщики.Адрес,
isnull(ЮридическиеЛица.Название,rtrim(ФизическиеЛица.Фамилия)+' '+
   substring(ФизическиеЛица.Имя,1,1)+'.'+
   substring(ФизическиеЛица.Отчество,1,1)+'.') as Поставщик
 from (Поставщики LEFT JOIN ФизическиеЛица
ON Поставщики.КодПоставщика=ФизическиеЛица.КодПоставщика)
LEFT JOIN ЮридическиеЛица
ON Поставщики.КодПоставщика=ЮридическиеЛица.КодПоставщика
```

Figure 2.13

The query can be saved in a file named SQLQuery11.sql. The query result may look like the one shown in Figure 2.14.

Results | Messages

	КодПоставщика	Адрес	Поставщик
1	1	г.Харьков, пр. Ленина, 55, к.108	Петров П.П.
2	2	г. Киев, пр. Победы, 154, к. 3	ООО "Интерфруг"
3	3	г. Харьков, ул. Пушкинская, 77	Иванов И.И.
4	4	г. Одесса, ул. Дерибасовская, 75	ЗАО "Транссервис"
5	5	г. Полтава, ул. Ленина, 15, кв. 43	Сидоров С.С.

Figure 2.14

Request 12

Generate a list of contracts (indicating the number, delivery date and information about the supplier), the total number of delivered goods and the total amount for each contract. For suppliers - individuals, display the surname and initials, for suppliers - legal entities - the number of the certificate of the VAT payer. The query result should include only those contracts on the basis of which the goods were actually delivered (ie so-called "empty" contracts should not be included in the query result)

The request text is shown in Figure 2.15.

The query can be saved in a file named SQLQuery12.sql.

```
use delivery

select Договоры.НомерДоговора,Договоры.ДатаДоговора,
       isnull(ЮридическиеЛица.Название,rtrim(ФизическиеЛица.Фамилия)+' '+
              substring(ФизическиеЛица.Имя,1,1)+'.'+
              substring(ФизическиеЛица.Отчество,1,1)+'.') as Поставщик,
       Sum(Поставлено.Количество) AS ОбъемПоставки,
       Sum(Количество*Цена) AS СуммаПоставки
    from (((Поставщики LEFT JOIN ФизическиеЛица
                  ON Поставщики.КодПоставщика=ФизическиеЛица.КодПоставщика)
       LEFT JOIN ЮридическиеЛица
                  ON Поставщики.КодПоставщика=ЮридическиеЛица.КодПоставщика)
       INNER JOIN Договоры ON Договоры.КодПоставщика=Поставщики.КодПоставщика)
       INNER JOIN Поставлено ON Договоры.НомерДоговора=Поставлено.НомерДоговора
    group by Договоры.НомерДоговора,Договоры.ДатаДоговора,
             isnull(ЮридическиеЛица.Название,rtrim(ФизическиеЛица.Фамилия)+' '+
                        substring(ФизическиеЛица.Имя,1,1)+'.'+
                        substring(ФизическиеЛица.Отчество,1,1)+'.')
    order by НомерДоговора
```

Figure 2.15

Request 13

Generate a list of goods (indicating the contract number and delivery date) supplied by suppliers 1 (PE Petrov P.P.) and 2 (Interfruit LLC).

Note. This request illustrates the features of using union operations (UNION). It is easy to see that this query can be easily implemented without using the union operation.

The request text is shown in Figure 2.16.

```
USE delivery

SELECT Поставлено.НомерДоговора, Договоры.ДатаДоговора,
       Поставлено.Товар, Поставщики.КодПоставщика
FROM Поставлено, Договоры, Поставщики
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора
AND Поставщики.КодПоставщика = Договоры.КодПоставщика AND Договоры.КодПоставщика = 1
UNION
SELECT Поставлено.НомерДоговора, Договоры.ДатаДоговора,
       Поставлено.Товар, Поставщики.КодПоставщика
FROM Поставлено, Договоры, Поставщики
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора
AND Поставщики.КодПоставщика = Договоры.КодПоставщика AND Договоры.КодПоставщика = 2
ORDER BY КодПоставщика,НомерДоговора
```

Figure 2.16

The query can be saved in a file named SQLQuery13.sql

Request 14

Generate a nomenclature of goods (ie a list of names of goods) that were supplied only by supplier 1 (PE Petrov P.P.), or only by supplier 2 (LLC Interfruit), or by both supplier 1 and supplier2.

The request text is shown in Figure 2.17.

```
USE delivery

SELECT DISTINCT Поставлено.Товар
FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора AND Договоры.КодПоставщика = 1
UNION
SELECT DISTINCT Поставлено.Товар
FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора AND Договоры.КодПоставщика = 2
ORDER BY Товар
```

Figure 2.17

The query can be saved in a file named SQLQuery14.sql

Request 15

Form a nomenclature of goods (ie a list of names of goods) that were supplied by both supplier 1 (PE Petrov P.P.) and supplier 2 (LLC Interfruit).

Note. This request illustrates the features of using intersection operations (INTERSECT).

The request text is shown in Figure 2.18.

```
USE delivery

SELECT DISTINCT Поставлено.Товар
FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора AND Договоры.КодПоставщика = 1
INTERSECT
SELECT DISTINCT Поставлено.Товар
FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора AND Договоры.КодПоставщика = 2
ORDER BY Товар
```

Figure 2.18

The query can be saved in a file named SQLQuery15.sql

Request 16

Generate a nomenclature of goods (ie a list of names of goods) that were supplied by supplier 1 (PE Petrov P.P.), but were not supplied by supplier 2 (LLC Interfruit).

Note. This request illustrates the features of using difference operations (EXCEPT).

The request text is shown in Figure 2.19.

```
USE delivery

SELECT DISTINCT Поставлено.Товар
FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора AND Договоры.КодПоставщика = 1
EXCEPT
SELECT DISTINCT Поставлено.Товар
FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора AND Договоры.КодПоставщика = 2
ORDER BY Товар
```

Figure 2.19

The query can be saved in a file named SQLQuery16.sql

Request 17

Create a list of goods, which should reflect the frequency of deliveries of goods. Only include items that have been shipped more than once. The list should be sorted in descending order of frequency of deliveries.

The request text is shown in Figure 2.20

```
USE delivery

SELECT Товар, COUNT(Товар) AS ЧастотаПоставок
  FROM Поставлено
    GROUP BY Товар
    HAVING COUNT(Товар)>1
    ORDER BY COUNT(Товар) DESC
```

Figure 2.20

The query can be saved in a file named SQLQuery17.sql

Request 18

Generate data on the quantitative dynamics of the supply of goods during 1999. The data should be aggregated on a monthly basis and presented in the form of a table, the rows of which are the names of the goods, and the columns are the numbers of the months of 1999. At the intersection of the row and column, the quantity of the given item, delivered in the given month, should be displayed.

Note. This request illustrates the features of creating and using a cross query using Transact-SQL.

The request text is shown in Figure 2.21.

The query result can look like the one shown in Figure 2.22.

The query can be saved in a file named SQLQuery18_1.sql

```
USE delivery

SELECT Товар, [1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12]
FROM
(
SELECT Товар,MONTH(ДатаДоговора) AS mesac,Количество
FROM Договоры,Поставлено
WHERE Договоры.НомерДоговора=Поставлено.НомерДоговора AND YEAR(ДатаДоговора)=1999
) p
PIVOT
(SUM(Количество)
FOR mesac IN ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12])
) AS pvt
ORDER BY Товар
```

Figure 2.21

Results | Messages

	Товар	1	2	3	4	5	6	7	8	9	10	11	12
1	Видеомагнитофон	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	37	NULL	NULL	NULL
2	Компьютер	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	67	47	NULL	NULL
3	Магнитофон	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	96	NULL	NULL	NULL
4	Монитор	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	129	73	NULL	NULL
5	Принтер	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	41	NULL	NULL	NULL
6	Стереосистема	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	50	NULL	NULL	NULL
7	Телевизор	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	132	96	NULL	NULL

Figure 2.22

The given query result may be inconvenient (for example, due to the presence of NULL values). This drawback can be eliminated, for example, by replacing the NULL values with 0. The text of the modified query is shown in Figure 3.23. The query can be saved in a file named SQLQuery18_2.sql

```
USE delivery

SELECT Товар, isnull([1],0) as [1],isnull([2],0) as [2],isnull([3],0) as [3],
              isnull([4],0) as [4],isnull([5],0) as [5],isnull([6],0) as [6],
              isnull([7],0) as [7],isnull([8],0) as [8],isnull([9],0) as [9],
              isnull([10],0) as [10],isnull([11],0) as [11],isnull([12],0) as [12]
FROM
(
SELECT Товар,MONTH(ДатаДоговора) AS mesac,Количество
FROM Договоры,Поставлено
WHERE Договоры.НомерДоговора=Поставлено.НомерДоговора AND YEAR(ДатаДоговора)=1999
) p
PIVOT
(SUM(Количество)
FOR mesac IN ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12])
) AS pvt
ORDER BY Товар
```

Figure 2.23

The query result may look like the one shown in Figure 2.24.

Results | Messages

	Товар	1	2	3	4	5	6	7	8	9	10	11	12
1	Видеомагнитофон	0	0	0	0	0	0	0	0	37	0	0	0
2	Компьютер	0	0	0	0	0	0	0	0	67	47	0	0
3	Магнитофон	0	0	0	0	0	0	0	0	96	0	0	0
4	Монитор	0	0	0	0	0	0	0	0	129	73	0	0
5	Принтер	0	0	0	0	0	0	0	0	41	0	0	0
6	Стереосистема	0	0	0	0	0	0	0	0	50	0	0	0
7	Телевизор	0	0	0	0	0	0	0	0	132	96	0	0

Figure 2.24

Request 19

Create a list of delivered goods. For each product in this list, the following data must be indicated: contract number, product name, number of units, unit price, delivery date, month name and year number.

The request text is shown in Figure 2.25.

```
USE delivery

SELECT Поставлено.НомерДоговора, Поставлено.Товар,
       Поставлено.Количество, Поставлено.Цена,
       Договоры.ДатаДоговора,
       DATENAME(month,ДатаДоговора) AS Месяц,
       YEAR(ДатаДоговора) AS Год
       FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора
```

Figure 2.25

The query result (fragment) may look like the one shown in the figure 3.26. The query can be saved in a file named SQLQuery19_1.sql

Results | Messages

	НомерДоговора	Товар	Количество	Цена	ДатаДоговора	Месяц	Год
16	4	Телевизор	56	990.56	1999-09-23 00:00:00.000	September	1999
17	5	Видеомагнитофон	17	850.12	1999-09-24 00:00:00.000	September	1999
18	5	Магнитофон	33	585.67	1999-09-24 00:00:00.000	September	1999
19	5	Монитор	44	590.23	1999-09-24 00:00:00.000	September	1999
20	5	Телевизор	14	860.33	1999-09-24 00:00:00.000	September	1999
21	6	Компьютер	32	1850.24	1999-10-01 00:00:00.000	October	1999
22	6	Монитор	51	520.95	1999-10-01 00:00:00.000	October	1999
23	6	Телевизор	34	810.15	1999-10-01 00:00:00.000	October	1999
24	7	Компьютер	15	1234.56	1999-10-02 00:00:00.000	October	1999
25	7	Монитор	22	389.75	1999-10-02 00:00:00.000	October	1999
26	7	Телевизор	62	900.58	1999-10-02 00:00:00.000	October	1999

Figure 2.26

As can be seen from the query result, the requirement to include the name of the month in the query result is formally fulfilled by using the built-in DATENAME () function. However, such month names are not always easy to read. There may be a requirement to replace them with Russian-speaking, Ukrainian-speaking, etc. This problem can be solved by developing a custom function that converts the names of the months. This approach is somewhat more time consuming. Another solution to the problem is to use the CASE function of the Transact-SQL language in the query. The text of such a query is shown in Figure 2.27. The result of the query (fragment) can be as shown in Figure 2.28. The query can be saved in a file named SQLQuery19_2.sql

```
USE delivery

SELECT Поставлено.НомерДоговора, Поставлено.Товар,
       Поставлено.Количество, Поставлено.Цена,
       Договоры.ДатаДоговора,
       Месяц = CASE MONTH(ДатаДоговора)
                WHEN 1 THEN 'январь'
                WHEN 2 THEN 'февраль'
                WHEN 3 THEN 'март'
                WHEN 4 THEN 'апрель'
                WHEN 5 THEN 'май'
                WHEN 6 THEN 'июнь'
                WHEN 7 THEN 'июль'
                WHEN 8 THEN 'август'
                WHEN 9 THEN 'сентябрь'
                WHEN 10 THEN 'октябрь'
                WHEN 11 THEN 'ноябрь'
                WHEN 12 THEN 'декабрь'
                ELSE '?????????'
                END,
       YEAR(ДатаДоговора) AS Год
       FROM Поставлено, Договоры
WHERE Договоры.НомерДоговора = Поставлено.НомерДоговора
```

Figure 2.27

Results | Messages

	НомерДогово...	Товар	Количество	Цена	ДатаДоговора	Месяц	Год
16	4	Телевизор	56	990.56	1999-09-23 00:00:00.000	сентябрь	1999
17	5	Видеомагнитофон	17	850.12	1999-09-24 00:00:00.000	сентябрь	1999
18	5	Магнитофон	33	585.67	1999-09-24 00:00:00.000	сентябрь	1999
19	5	Монитор	44	590.23	1999-09-24 00:00:00.000	сентябрь	1999
20	5	Телевизор	14	860.33	1999-09-24 00:00:00.000	сентябрь	1999
21	6	Компьютер	32	1850.24	1999-10-01 00:00:00.000	октябрь	1999
22	6	Монитор	51	520.95	1999-10-01 00:00:00.000	октябрь	1999
23	6	Телевизор	34	810.15	1999-10-01 00:00:00.000	октябрь	1999
24	7	Компьютер	15	1234.56	1999-10-02 00:00:00.000	октябрь	1999
25	7	Монитор	22	389.75	1999-10-02 00:00:00.000	октябрь	1999
26	7	Телевизор	62	900.58	1999-10-02 00:00:00.000	октябрь	1999

Figure 2.28

Saving work results

Save query files: SQLQuery01_1.sql;
SQLQuery01_2.sql; SQLQuery02.sql;

SQLQuery03.sql;
SQLQuery04.sql;
SQLQuery05.sql;
SQLQuery06.sql;
SQLQuery07.sql;
SQLQuery08.sql;
SQLQuery09.sql;
SQLQuery10.sql;
SQLQuery11.sql;
SQLQuery12.sql;
SQLQuery13.sql;
SQLQuery14.sql;
SQLQuery15.sql;
SQLQuery16.sql;
SQLQuery17.sql;
SQLQuery18_1.sql; SQLQuery18_2.sql;
SQLQuery19_1.sql; SQLQuery19_2.sql

Report requirements:

1) briefly describe the main stages of the work;

2) for each of the implemented requests, provide a request condition, query text and query execution result (in the form of a table, figure, screen form, etc.).