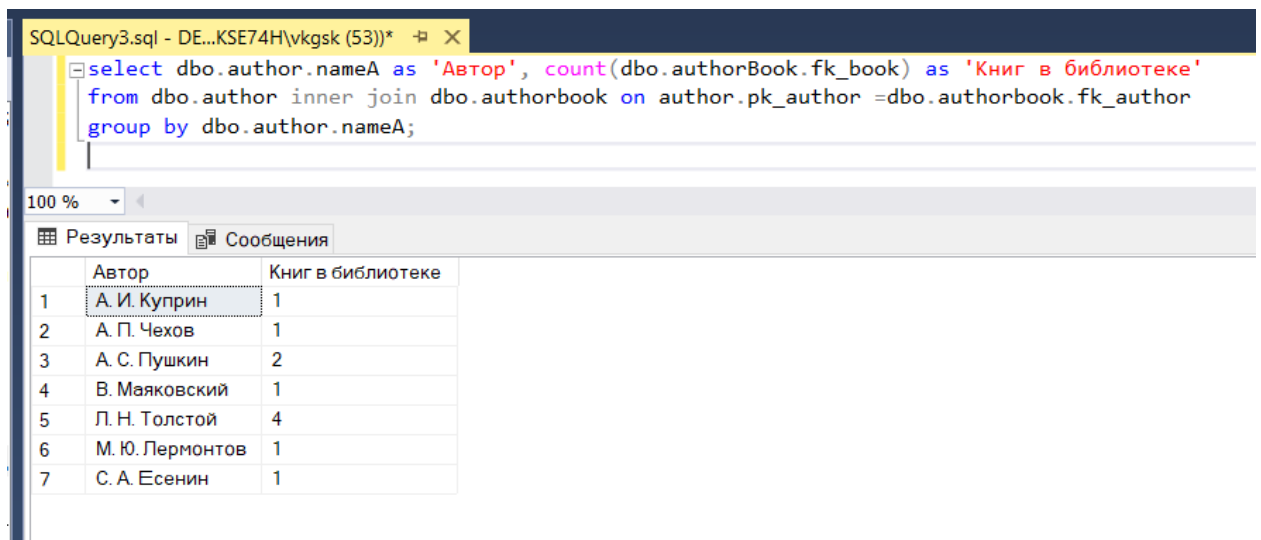


Запрос 1.

Посмотрим количество представленных книг в библиотеке разных авторов.

```
select dbo.author.nameA as 'Автор', count(dbo.authorBook.fk_book) as  
'Книг в библиотеке'  
from dbo.author inner join dbo.authorbook on author.pk_author  
=dbo.authorbook.fk_author  
group by dbo.author.nameA;
```



The screenshot shows a SQL query window with the following text:

```
SQLQuery3.sql - DE...KSE74H\vkgsk (53))*  
select dbo.author.nameA as 'Автор', count(dbo.authorBook.fk_book) as 'Книг в библиотеке'  
from dbo.author inner join dbo.authorbook on author.pk_author =dbo.authorbook.fk_author  
group by dbo.author.nameA;
```

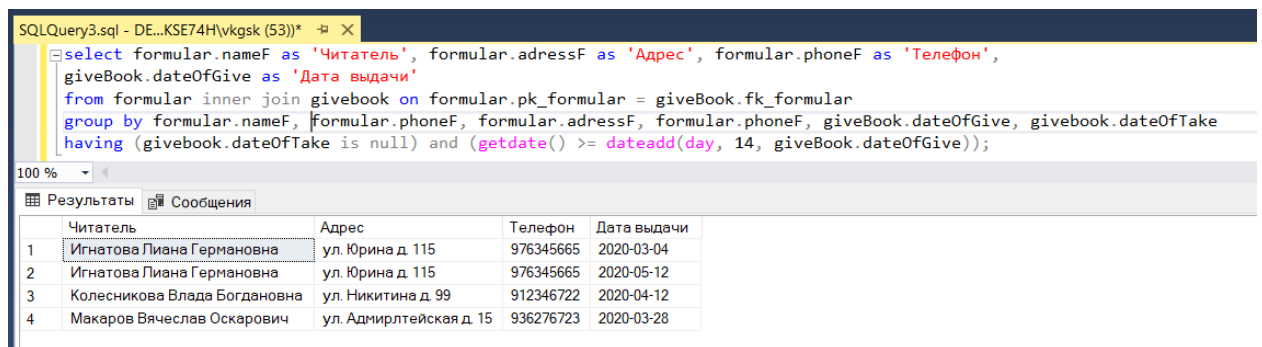
Below the query window, the 'Results' tab is selected, displaying a table with 7 rows and 2 columns: 'Автор' and 'Книг в библиотеке'.

	Автор	Книг в библиотеке
1	А. И. Куприн	1
2	А. П. Чехов	1
3	А. С. Пушкин	2
4	В. Маяковский	1
5	Л. Н. Толстой	4
6	М. Ю. Лермонтов	1
7	С. А. Есенин	1

Запрос 2.

Выведем всех задолжников библиотеки. Задолжником считается читатель, который не вернул книгу в течение двух недель.

```
select formular.nameF as 'Читатель', formular.adressF as 'Адрес',  
formular.phoneF as 'Телефон',  
giveBook.dateOfGive as 'Дата выдачи'  
from formular inner join givebook on formular.pk_formular =  
giveBook.fk_formular  
group by formular.nameF, formular.phoneF, formular.adressF,  
formular.phoneF, giveBook.dateOfGive, givebook.dateOfTake  
having (givebook.dateOfTake is null) and (getdate() >= dateadd(day, 14,  
giveBook.dateOfGive));
```



The screenshot shows a SQL query editor with the following query:

```
select formular.nameF as 'Читатель', formular.adressF as 'Адрес', formular.phoneF as 'Телефон',  
giveBook.dateOfGive as 'Дата выдачи'  
from formular inner join givebook on formular.pk_formular = giveBook.fk_formular  
group by formular.nameF, formular.phoneF, formular.adressF, formular.phoneF, giveBook.dateOfGive, givebook.dateOfTake  
having (givebook.dateOfTake is null) and (getdate() >= dateadd(day, 14, giveBook.dateOfGive));
```

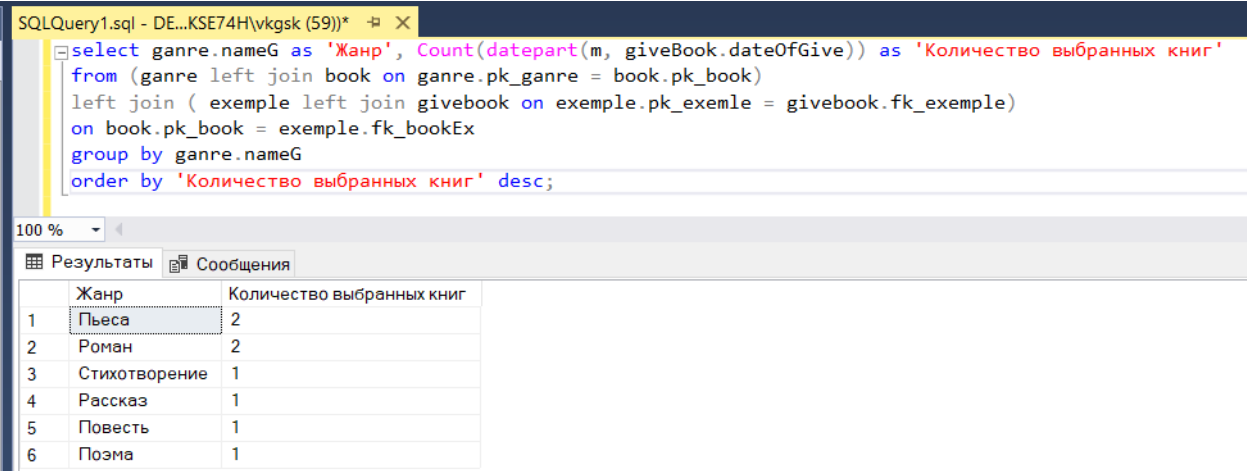
Below the query, the results are displayed in a table with 5 columns: Читатель, Адрес, Телефон, and Дата выдачи. The table contains 4 rows of data.

	Читатель	Адрес	Телефон	Дата выдачи
1	Игнатова Лиана Германовна	ул. Юрина д 115	976345665	2020-03-04
2	Игнатова Лиана Германовна	ул. Юрина д 115	976345665	2020-05-12
3	Колесникова Влада Богдановна	ул. Никитина д 99	912346722	2020-04-12
4	Макаров Вячеслав Оскарович	ул. Адмирлтейская д 15	936276723	2020-03-28

Запрос 3.

Узнаем какие жанры пользуются популярностью в библиотеке.

```
select ganre.nameG as 'Жанр', Count(datepart(m, giveBook.dateOfGive)) as  
'Количество выбранных книг'  
  
from (ganre left join book on ganre.pk_ganre = book.pk_book)  
  
left join ( exemple left join givebook on exemple.pk_exemle =  
givebook.fk_exemple)  
  
on book.pk_book = exemple.fk_bookEx  
  
group by ganre.nameG  
  
order by 'Количество выбранных книг' desc;
```



The screenshot shows a SQL query window with the following text:

```
SQLQuery1.sql - DE...KSE74H\vkgsk (59))*  X  
select ganre.nameG as 'Жанр', Count(datepart(m, giveBook.dateOfGive)) as 'Количество выбранных книг'  
from (ganre left join book on ganre.pk_ganre = book.pk_book)  
left join ( exemple left join givebook on exemple.pk_exemle = givebook.fk_exemple)  
on book.pk_book = exemple.fk_bookEx  
group by ganre.nameG  
order by 'Количество выбранных книг' desc;
```

Below the query window, the 'Results' tab is active, displaying a table with 2 columns: 'Жанр' and 'Количество выбранных книг'. The table contains 6 rows of data.

	Жанр	Количество выбранных книг
1	Пьеса	2
2	Роман	2
3	Стихотворение	1
4	Рассказ	1
5	Повесть	1
6	Позма	1

Запрос 4.

Выведем читателей, которые любят новые книги. Под новой книгой будем подразумевать год издания не позднее 2015 года.

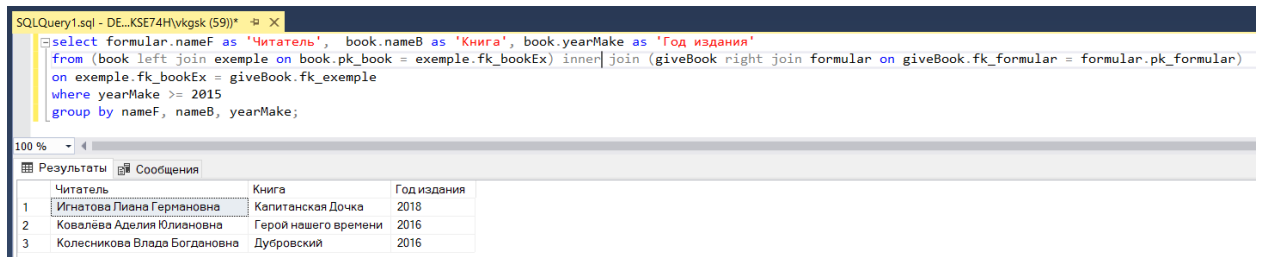
```
select formular.nameF as 'Читатель',    book.nameB as 'Книга',
book.yearMake as 'Год издания'

from (book left join exemple on book.pk_book = exemple.fk_bookEx) inner
join (giveBook right join formular on giveBook.fk_formular =
formular.pk_formular)

on exemple.fk_bookEx = giveBook.fk_exemple

where yearMake >= 2015

group by nameF, nameB, yearMake;
```



SQLQuery1.sql - DE...KSE74H\vkgsk (59))

```
select formular.nameF as 'Читатель',    book.nameB as 'Книга', book.yearMake as 'Год издания'
from (book left join exemple on book.pk_book = exemple.fk_bookEx) inner join (giveBook right join formular on giveBook.fk_formular = formular.pk_formular)
on exemple.fk_bookEx = giveBook.fk_exemple
where yearMake >= 2015
group by nameF, nameB, yearMake;
```

100 %

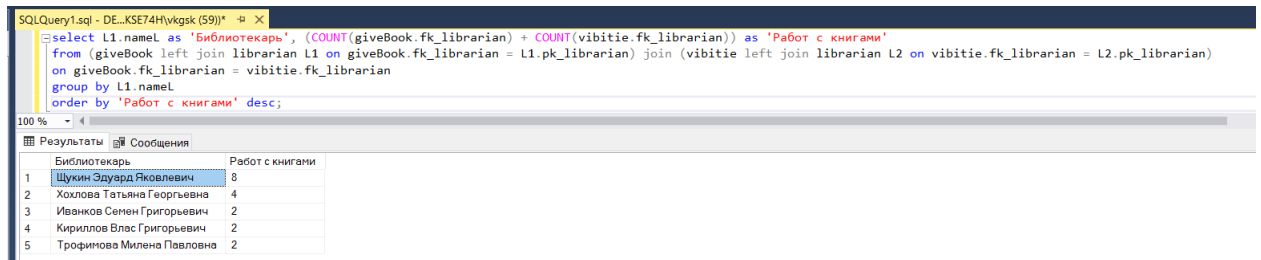
Результаты

	Читатель	Книга	Год издания
1	Игнатьева Лияна Германовна	Капитанская Дочка	2018
2	Ковалёва Аделия Юлиановна	Герой нашего времени	2016
3	Колесникова Влада Богдановна	Дубровский	2016

Запрос 5.

Узнаем, сколько работ с книгами совершил библиотекарь, т.е. сколько книг он выдал/принял и сколько книг списал.

```
select L1.nameL as 'Библиотекарь', (COUNT(giveBook.fk_librarian) +  
COUNT(vibitie.fk_librarian)) as 'Работ с книгами'  
  
from (giveBook left join librarian L1 on giveBook.fk_librarian =  
L1.pk_librarian) join (vibitie left join librarian L2 on vibitie.fk_librarian =  
L2.pk_librarian)  
  
on giveBook.fk_librarian = vibitie.fk_librarian  
  
group by L1.nameL  
  
order by 'Работ с книгами' desc;
```



SQLQuery1.sql - DE...KSE74H\vkgsk (59) * ❏ ✕

```
select L1.nameL as 'Библиотекарь', (COUNT(giveBook.fk_librarian) + COUNT(vibitie.fk_librarian)) as 'Работ с книгами'  
from (giveBook left join librarian L1 on giveBook.fk_librarian = L1.pk_librarian) join (vibitie left join librarian L2 on vibitie.fk_librarian = L2.pk_librarian)  
on giveBook.fk_librarian = vibitie.fk_librarian  
group by L1.nameL  
order by 'Работ с книгами' desc;
```

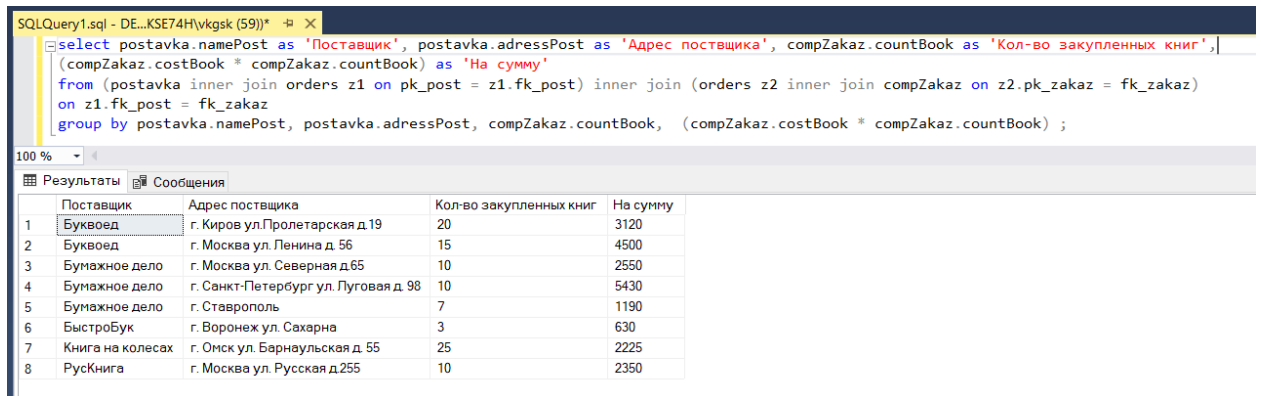
100 %

	Библиотекарь	Работ с книгами
1	Щукин Эдуард Яковлевич	8
2	Хохлова Татьяна Георгиевна	4
3	Иванков Семен Григорьевич	2
4	Кириллов Влас Григорьевич	2
5	Трофимцова Милена Павловна	2

Запрос 6.

Выведем поставщиков, у которых совершались заказы: в каком и на какую сумму.

```
select поставка.namePost as 'Поставщик', поставка.adressPost as 'Адрес  
поставщика', compZakaz.countBook as 'Кол-во закупленных книг',  
(compZakaz.costBook * compZakaz.countBook) as 'На сумму'  
from (поставка inner join orders z1 on pk_post = z1.fk_post) inner join  
(orders z2 inner join compZakaz on z2.pk_zakaz = fk_zakaz)  
on z1.fk_post = fk_zakaz  
group by поставка.namePost, поставка.adressPost, compZakaz.countBook,  
(compZakaz.costBook * compZakaz.countBook) ;
```



SQLQuery1.sql - DE...KSE74H(vkgsk (59))

```
select поставка.namePost as 'Поставщик', поставка.adressPost as 'Адрес поставщика', compZakaz.countBook as 'Кол-во закупленных книг',  
(compZakaz.costBook * compZakaz.countBook) as 'На сумму'  
from (поставка inner join orders z1 on pk_post = z1.fk_post) inner join (orders z2 inner join compZakaz on z2.pk_zakaz = fk_zakaz)  
on z1.fk_post = fk_zakaz  
group by поставка.namePost, поставка.adressPost, compZakaz.countBook, (compZakaz.costBook * compZakaz.countBook) ;
```

100 %

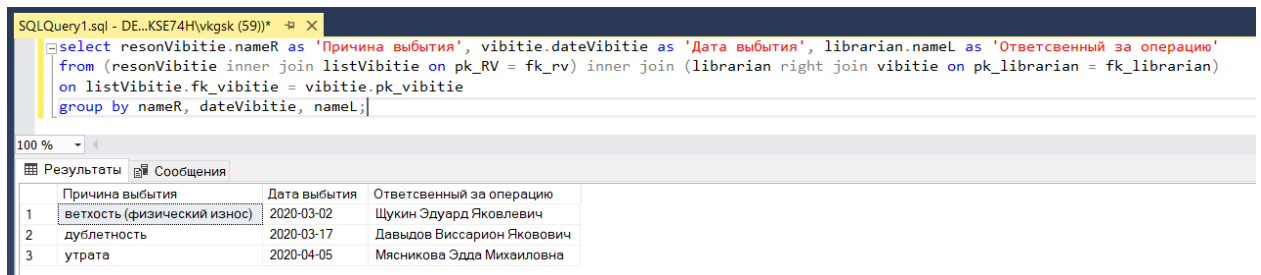
Результаты Сообщения

	Поставщик	Адрес поставщика	Кол-во закупленных книг	На сумму
1	Буквоед	г. Киров ул. Пролетарская д.19	20	3120
2	Буквоед	г. Москва ул. Ленина д. 56	15	4500
3	Бумажное дело	г. Москва ул. Северная д.65	10	2550
4	Бумажное дело	г. Санкт-Петербург ул. Пуговая д. 98	10	5430
5	Бумажное дело	г. Ставрополь	7	1190
6	БыстроБук	г. Воронеж ул. Сахарна	3	630
7	Книга на колесах	г. Омск ул. Барнаульская д. 55	25	2225
8	РусКнига	г. Москва ул. Русская д.255	10	2350

Запрос 7.

Узнаем, кто из библиотекарей списывал книги и по какой причине.

```
select resonVibitie.nameR as 'Причина выбытия', vibitie.dateVibitie as  
'Дата выбытия', librarian.nameL as 'Ответственный за операцию'  
  
from (resonVibitie inner join listVibitie on pk_RV = fk_rv) inner join  
(librarian right join vibitie on pk_librarian = fk_librarian)  
  
on listVibitie.fk_vibitie = vibitie.pk_vibitie  
  
group by nameR, dateVibitie, nameL;
```



SQLQuery1.sql - DE...KSE74H\vkgsk (59) * X

```
select resonVibitie.nameR as 'Причина выбытия', vibitie.dateVibitie as 'Дата выбытия', librarian.nameL as 'Ответственный за операцию'  
from (resonVibitie inner join listVibitie on pk_RV = fk_rv) inner join (librarian right join vibitie on pk_librarian = fk_librarian)  
on listVibitie.fk_vibitie = vibitie.pk_vibitie  
group by nameR, dateVibitie, nameL;
```

100 %

Результаты Сообщения

	Причина выбытия	Дата выбытия	Ответственный за операцию
1	ветхость (физический износ)	2020-03-02	Щукин Эдуард Яковлевич
2	дублетность	2020-03-17	Давыдов Виссарион Яковович
3	утрата	2020-04-05	Мясникова Эдда Михайловна

Создадим процедуру, которая будет обновлять статус посетителя библиотеки. Если человек имеет на руках книгу более 14 дней, ему присваивается статус 0, если меньше 14 или сдал все книги в библиотеку – 1. Для реализации добавим столбец statusF в таблицу formular.

```
ALTER PROCEDURE [dbo].[up_stat]
AS
BEGIN
    SET NOCOUNT ON;

    update formular set statusF = 0 from formular inner join givebook on formular.pk_formular = giveBook.fk_formular
    where (givebook.dateOfTake is null) and (getdate() >= dateadd(day, 14, giveBook.dateOfGive))
    update formular set statusF = 1 from formular inner join givebook on formular.pk_formular = giveBook.fk_formular
    where givebook.dateOfTake is not null

    select nameF, adressF, statusF from formular
END
```

Сообщения

Выполнение команд успешно завершено.

Рисунок – Хранимая процедура №2

SQLQuery30.sql - D:\KSE741\vkysk (db) | SQLQuery24.sql - D:\KSE741\vkysk

exec [up_stat]

100 %

Результаты | Сообщения

	nameF	adressF	statusF
1	Игнатова Лиана Германовна	ул. Юрина д. 115	1
2	Моисеева Данна Рудольфовна	ул. М. Горького д. 67	1
3	Щербакова Клавдия Гордеевна	ул. Попова д. 89	1
4	Колесникова Влада Богдановна	ул. Никитина д. 99	0
5	Ковалёва Аделия Юлиановна	ул. Взлетная д. 56	1
6	Шарапов Лаврентий Рубенович	ул. Юрина д. 3	1
7	Макаров Вячеслав Оскарович	ул. Адмирлтейская д. 15	0
8	Брагин Иосиф Протасьевич	ул. Кирова д. 7	1
9	Борисов Моисей Лукъевич	ул. Молодежная д. 55	1
10	Субботин Дмитрий Арсеньевич	ул. Приречная д. 5	1
11	Артемьев Лука Николаевич	ул. Приречная д. 7	0

Рисунок – Выполнение запроса

Также необходима процедура, которая будет формировать список на списание книг по причине ветхости определенного года.

```

ALTER PROCEDURE [dbo].[old_list]
-- Add the parameters for the stored procedure here
@numDoc int, @year int, @librarian int
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select @numDoc = (REPLICATE('0', 8 - len(@numDoc)) + @numDoc)

INSERT INTO [dbo].[vibitie]
([numVibitie]
,[dateVibitie]
,[fk_librarian])
VALUES
(@numDoc, GETDATE(), @librarian)

INSERT INTO [dbo].[listVibitie]
select vibitie.pk_vibitie, 2, exemple.pk_exemple
from (vibitie join listVibitie on vibitie.pk_vibitie = listVibitie.fk_vibitie)
join (book join exemple on book.pk_book = exemple.fk_bookEx) on pk_exemple = fk_exemple
where yearMake between (@year-1) and @year

select * from listVibitie
END

```

Рисунок – Хранимая процедура №3

SQLQuery12.sql - D:\KSE74H\vkgsk (73)* SQLQuery1

exec[old_list] 3479, 2012, 4

select * from vibitie

99 %

Результаты Сообщения

	pk_LV	fk_vibitie	fk_rv	fk_exemple
1	1	1	2	11
2	2	1	2	12
3	3	3	5	15
4	4	6	1	17
5	10	1	2	11
6	11	1	2	11

	pk_vibitie	numVibitie	dateVibitie	fk_librarian
1	1	2343	2020-03-02	2
2	2	1553	2020-03-17	3
3	3	1554	2020-03-17	3
4	4	1434	2020-03-21	1
5	5	2321	2020-03-23	4
6	6	8372	2020-04-05	5
7	7	9843	2020-04-06	9
8	8	2983	2020-04-18	6
9	9	4323	2020-04-18	6
10	10	2383	2020-05-05	7
11	11	5555	2020-10-06	3
12	12	5555	2020-10-06	3
13	13	5555	2020-10-06	3
14	14	5555	2020-10-06	3
15	15	5555	2020-10-06	5
16	16	6733	2020-10-06	5
17	17	3479	2020-10-09	4
18	18	3479	2020-10-09	4

Рисунок – Выполнение запроса

Т.к. дата возврата не должна быть раньше даты выдачи, создадим Триггер для проверки введенной даты. Если дата записана неправильно, то будем считать, что книга до сих пор находится на руках у читателя.

```
-- =====
ALTER TRIGGER [dbo].[change_date]
ON [dbo].[giveBook]
AFTER INSERT,UPDATE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    update giveBook set dateOfTake = null where dateOfGive > dateOfTake

END
```

Рисунок – Триггер №4

И триггер на проверку того, что один экземпляр книги был выдан одному читателю библиотеки и эта книга не была списана ранее, иначе такая запись удаляется. Работоспособность триггеров проверим после импорта данных в таблицы.

```
ALTER TRIGGER [dbo].[del_dupl]
ON [dbo].[giveBook]
AFTER INSERT,UPDATE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here

    declare @tmpTable table (pk_dupl int not null)

    insert into @tmpTable (pk_dupl)

    select pk_GV from (giveBook join exemple on fk_exemple = pk_exemple)
    join (listVibitie join vibitie on fk_vibitie = pk_vibitie) on pk_exemple = listVibitie.fk_exemple
    where dateVibitie < dateOfGive
    union all
    select g1.pk_GV from giveBook g1, giveBook g2
    where g1.fk_exemple = g2.fk_exemple and g1.dateOfGive = g2.dateOfGive
    and (g1.dateOfTake > g2.dateOfGive or g1.dateOfTake is null) and g1.pk_GV <> g2.pk_GV

    delete giveBook where pk_GV in (select pk_dupl from @tmpTable)

    select * from giveBook

END
```

Рисунок – Триггер №5

Проверяем корректность выдачи книг. Т.е. экземпляр книги не может быть выдан в момент нахождения книги на руках.

exec[right_data]

99 %

Результаты Сообщения

	dateOfGive	dateOfTake	fk_exemple	comment
26049	2020-02-03	2020-08-19	2519	Корректно
26050	2020-05-13	2020-09-09	3978	Наложение дат 27958
26051	2020-05-17	2020-08-05	1615	Корректно
26052	2020-05-21	2020-08-02	675	Корректно

exec[right_data]

99 %

Результаты Сообщения

	dateOfGive	dateOfTake	fk_exemple	comment
27957	2020-01-10	2020-09-19	2540	Корректно
27958	2020-05-12	2020-07-08	6274	Корректно
27959	2020-02-25	2020-08-02	1233	Корректно
27960	2020-04-28	NULL	9224	Корректно

exec[right_data]

99 %

Результаты Сообщения

	dateOfGive	dateOfTake	fk_exemple	comment
29681	2020-05-27	NULL	3305	Корректно
29682	2020-05-18	2020-09-09	8180	Наложение дат 32844
29683	2020-09-29	NULL	4323	Корректно
29684	2020-03-31	NULL	3702	Корректно

```
ALTER PROCEDURE [dbo].[right_data]
-- Add the parameters for the stored procedure here

AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select g1.dateOfGive, g1.dateOfTake, g1.fk_exemple,
case g1.dateOfTake when (select g3.dateOfTake from giveBook g3 where g3.dateOfGive > g3.dateOfTake )
then CONCAT('Наложение дат', ' ', (select top 1 g2.pk_GV from giveBook g2
where g1.fk_exemple = g2.fk_exemple and g1.dateOfGive > g2.dateOfGive and g2.dateOfTake is null))
else 'Корректно'
end 'comment'
from giveBook g1

END
```

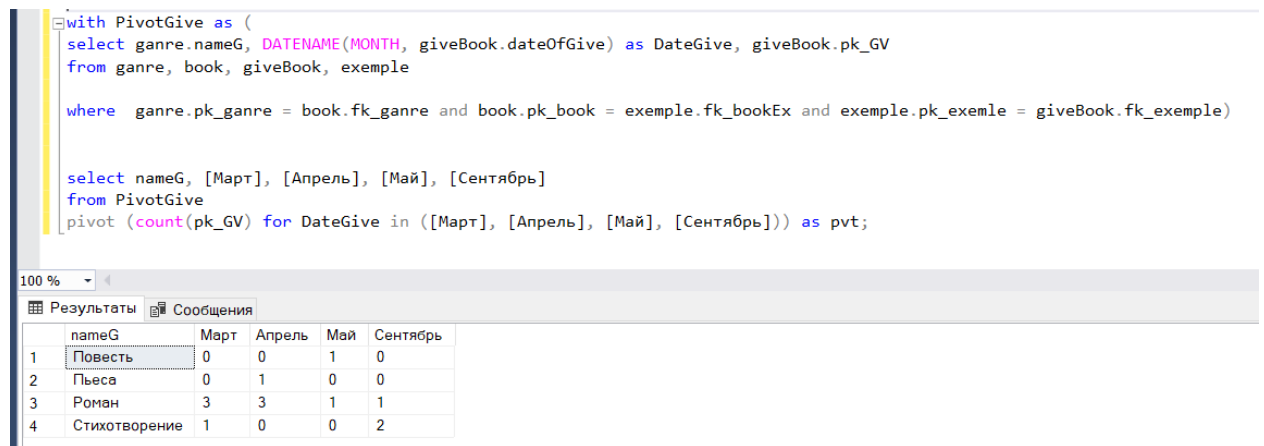
Защита

Запрос 1

```
with PivotGive as (  
select ganre.nameG, DATENAME(MONTH, giveBook.dateOfGive) as DateGive, giveBook.pk_GV  
from ganre, book, giveBook, exemple
```

```
where ganre.pk_ganre = book.fk_ganre and book.pk_book = exemple.fk_bookEx and  
exemple.pk_exemle = giveBook.fk_exemple)
```

```
select nameG, [Март], [Апрель], [Май], [Сентябрь]  
from PivotGive  
pivot (count(pk_GV) for DateGive in ([Март], [Апрель], [Май], [Сентябрь])) as pvt;
```



The screenshot shows a SQL query window with the following code:

```
with PivotGive as (  
select ganre.nameG, DATENAME(MONTH, giveBook.dateOfGive) as DateGive, giveBook.pk_GV  
from ganre, book, giveBook, exemple  
  
where ganre.pk_ganre = book.fk_ganre and book.pk_book = exemple.fk_bookEx and exemple.pk_exemle = giveBook.fk_exemple)  
  
select nameG, [Март], [Апрель], [Май], [Сентябрь]  
from PivotGive  
pivot (count(pk_GV) for DateGive in ([Март], [Апрель], [Май], [Сентябрь])) as pvt;
```

Below the query window, the 'Results' tab is active, displaying the following table:

	nameG	Март	Апрель	Май	Сентябрь
1	Повесть	0	0	1	0
2	Пьеса	0	1	0	0
3	Роман	3	3	1	1
4	Стихотворение	1	0	0	2

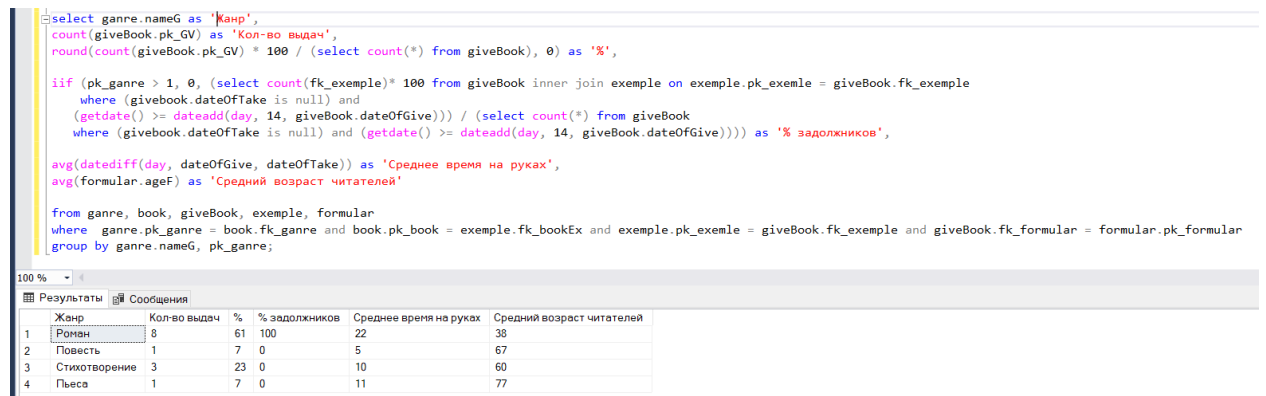
Запрос 2

```
select genre.nameG as 'Жанр',
count(giveBook.pk_GV) as 'Кол-во выдач',
round(count(giveBook.pk_GV) * 100 / (select count(*) from giveBook), 0) as '%',

iif (pk_genre > 1, 0, (select count(fk_exemple)* 100 from giveBook inner join exemple on
exemple.pk_exemple = giveBook.fk_exemple
where (givebook.dateOfTake is null) and
(getdate() >= dateadd(day, 14, giveBook.dateOfGive))) / (select count(*) from giveBook
where (givebook.dateOfTake is null) and (getdate() >= dateadd(day, 14,
giveBook.dateOfGive)))) as '% задолжников',

avg(datediff(day, dateOfGive, dateOfTake)) as 'Среднее время на руках',
avg(formular.ageF) as 'Средний возраст читателей'

from genre, book, giveBook, exemple, formular
where genre.pk_genre = book.fk_genre and book.pk_book = exemple.fk_bookEx and
exemple.pk_exemple = giveBook.fk_exemple and giveBook.fk_formular = formular.pk_formular
group by genre.nameG, pk_genre;
```



The screenshot shows a SQL query in the Enterprise Manager editor and its results in the Results pane. The query is a complex SELECT statement with multiple joins and calculations. The Results pane shows a table with 6 columns: Жанр, Кол-во выдач, %, % задолжников, Среднее время на руках, and Средний возраст читателей. The data is grouped by genre.

	Жанр	Кол-во выдач	%	% задолжников	Среднее время на руках	Средний возраст читателей
1	Роман	8	61	100	22	38
2	Повесть	1	7	0	5	67
3	Стихотворение	3	23	0	10	60
4	Пьеса	1	7	0	11	77