

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования
Московский технический университет связи и информатики

Кафедра информационных систем

ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ № 1 - 2
по дисциплине
РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ И ПРИЛОЖЕНИЯ
«Работа с данными из удаленных источников в РИС
склада предприятия»
Вариант № 30

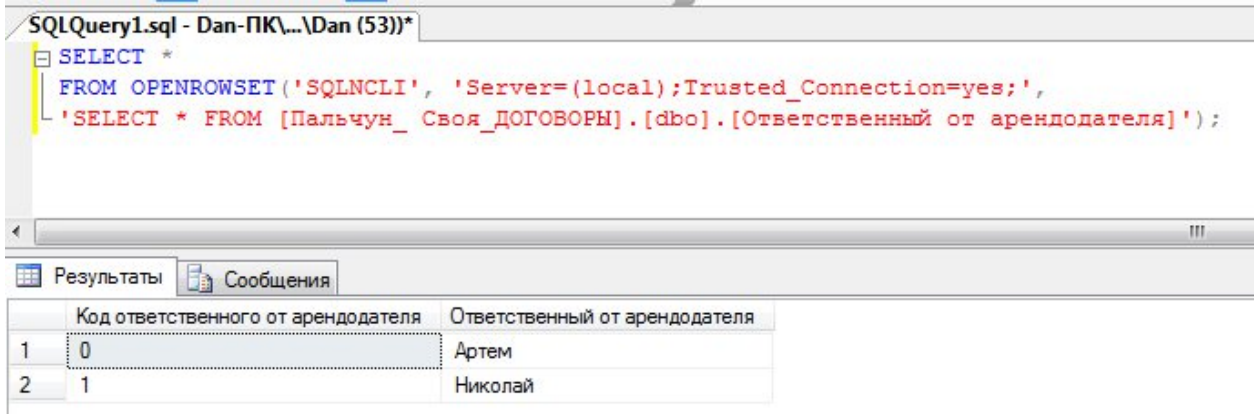
Выполнил: Пальчун Д.А.
группа БЭИ1802

Принял: Ванина М.Ф.,
доцент, к.т.н.

Москва, 2021

Нерегламентируемые соединения с внешними источниками данных

Отдел аренды компании Сириус планирует открыть филиал для работы в новом городе. Для того, чтобы выдавать аренду, необходимо знать какие ответственные лица есть у компании, эту информацию можно получить из центрального офиса, посредством sql запроса к серверу базы данных с функцией OPENROWSET():



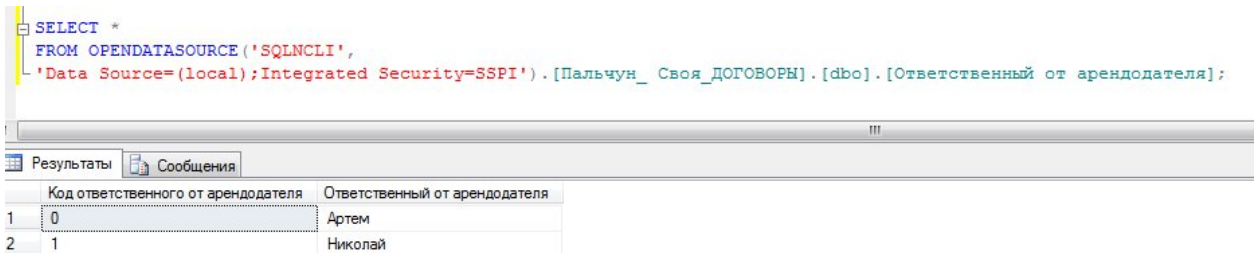
The screenshot shows a SQL query window titled 'SQLQuery1.sql - Dan-ПК\...\Dan (53))*'. The query is as follows:

```
SELECT *
FROM OPENROWSET('SQLNCLI', 'Server=(local);Trusted_Connection=yes;',
'SELECT * FROM [Пальчун_Своя_ДОГОВОРЫ].[dbo].[Ответственный от арендодателя]');
```

Below the query window, the 'Results' tab is active, displaying a table with two columns: 'Код ответственного от арендодателя' and 'Ответственный от арендодателя'. The table contains two rows of data:

	Код ответственного от арендодателя	Ответственный от арендодателя
1	0	Артем
2	1	Николай

SQL запрос к серверу можно также сделать с функцией OPENDATASOURCE():



The screenshot shows a SQL query window titled 'SQLQuery1.sql - Dan-ПК\...\Dan (53))*'. The query is as follows:

```
SELECT *
FROM OPENDATASOURCE('SQLNCLI',
'Data Source=(local);Integrated Security=SSPI') . [Пальчун_Своя_ДОГОВОРЫ].[dbo].[Ответственный от арендодателя];
```

Below the query window, the 'Results' tab is active, displaying a table with two columns: 'Код ответственного от арендодателя' and 'Ответственный от арендодателя'. The table contains two rows of data:

	Код ответственного от арендодателя	Ответственный от арендодателя
1	0	Артем
2	1	Николай

Для того, чтобы все прошло успешно, необходимо разрешить нерегламентированные соединения с помощью:

```
GO
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Ad Hoc Distributed Queries', 1;
GO
RECONFIGURE;
GO
```

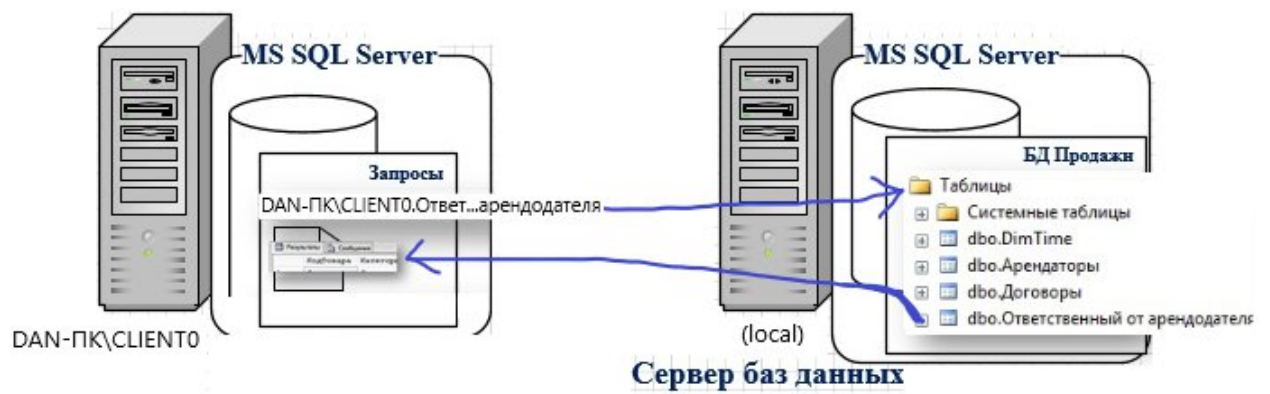


Рис. 1 – Архитектурная модель решения задачи.

Связанные серверы

Одним запросом взаимодействие с центральным офисом ограничиться не может, поэтому следует связать с серверы филиала и центрального отдела.

Создание связанного сервера

Выбор страницы

- Общие
- Безопасность
- Параметры сервера

Сценарий Справка

Связанный сервер: DAN-ПК\CLIENT

Тип сервера:

- ☒ Сервер SQL Server
- ☐ Другой источник данных

Поставщик: Microsoft OLE DB Provider for SQL Server

Название продукта:

Источник данных:

Строка поставщика:

Расположение:

Каталог:

Соединение

Сервер: (local)

Соединение: Dan-ПК\Dan

[Просмотреть свойства соединения](#)

Выполнение

Готово

Связанный сервер: имя, которое будет использоваться для обозначения связанного сервера.

OK Отмена

Так как все заключенные договоры хранятся в центральном офисе, их было бы удобно видеть из филиалов. Чтобы не запрашивать их постоянно, это можно было бы сделать посредством установки связи между серверами и обращением к ним с помощью SQL запросов, например:

```
SELECT *
FROM [(local)].[Пальчун Своя ДОГОВОРЫ].[dbo].[Договоры];
```

	Идентификатор арендатора	Номер помещения	Номер договора аренды	Договор действует/закр...	TimeKey	Код периоди...
1	0	1	1	1	20201025	0
2	1	2	2	1	20201026	1
3	2	3	3	1	20201027	2
4	3	4	4	1	20201028	0
5	4	5	5	1	20201029	1
6	5	6	6	1	20201030	2
7	0	7	7	1	20201031	0
8	1	8	8	1	20201101	1
9	2	9	9	1	20201102	2
10	3	10	10	1	20201103	0
11	4	11	11	1	20201104	1
12	5	12	12	1	20201105	2
13	0	13	13	1	20201106	0
14	1	14	14	1	20201107	1
15	2	15	15	1	20201108	2
16	3	16	16	1	20201109	0
17	4	17	17	1	20201110	1
18	5	18	18	1	20201111	2
19	0	19	19	1	20201112	0
20	1	20	20	1	20201113	1
21	2	21	21	1	20201114	2
22	3	22	22	1	20201115	0
23	4	23	23	1	20201116	1
24	5	24	24	1	20201116	2
25	0	25	25	1	20201117	0

Запрос успешно выполнен.

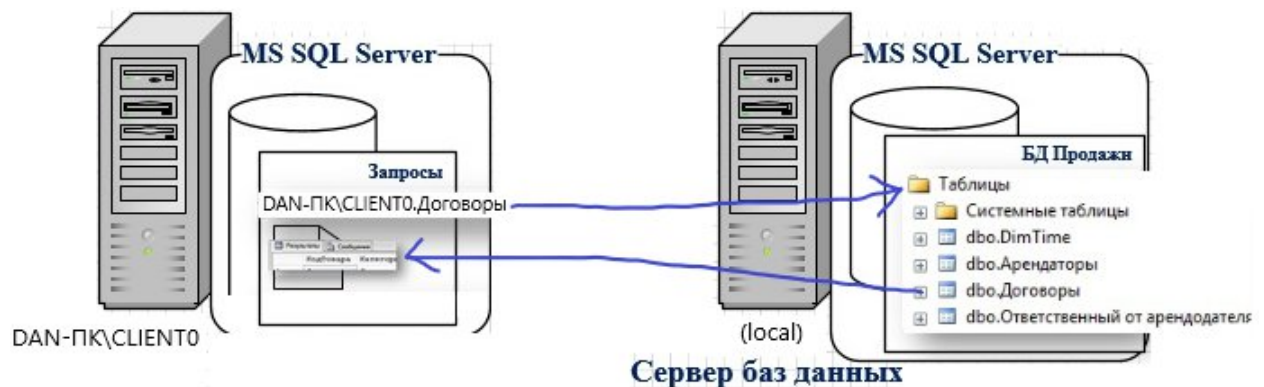


Рис. 2 – Архитектурная модель решения задачи.

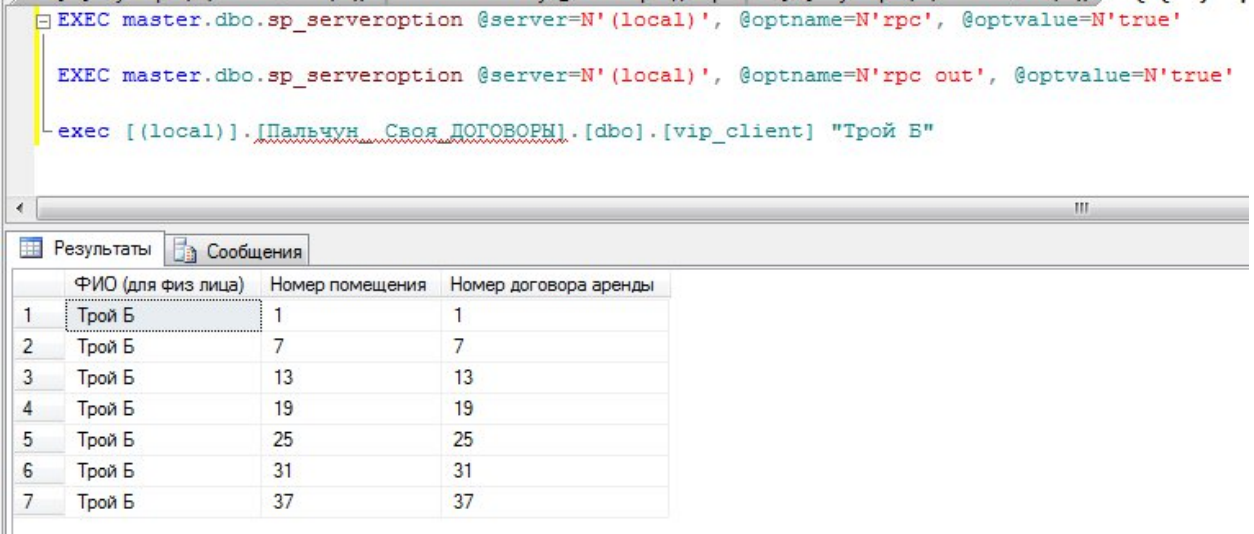
На основном сервере хранятся все клиенты, а также процедура, показывающая клиента и договор в котором он участвует, его площадь которую он арендует:

```
use [Пальчун_Своя_ДОГОВОРЫ]
go
create procedure vip_client
@C nvarchar(50)
as
select [ФИО (для физ лица)], [Номер помещения], [Номер договора аренды]
from [dbo].[Арендаторы] a inner join [dbo].[Договоры] d
on a.[Идентификатор арендатора]=d.[Идентификатор арендатора]
where [ФИО (для физ лица)]=@C;
```

Подобная процедура была бы очень полезна и для филиала. Для того, чтобы обращение к процедуре прошло успешно, необходимо ввести:

```
EXEC master.dbo.sp_serveroption @server=N'(local)', @optname=N'rpc', @optvalue=N'true'
EXEC master.dbo.sp_serveroption @server=N'(local)', @optname=N'rpc out', @optvalue=N'true'
```

Реализовать запрос к данной процедуре можно следующим образом:



The screenshot shows a SQL Server Enterprise Manager window. The top pane displays a query that sets server options for 'rpc' and 'rpc out' to 'true' on the local server, and then executes the 'vip_client' procedure on the 'Пальчун_Своя_ДОГОВОРЫ' database with the parameter 'Трой Б'. The bottom pane shows the results of the query in a table with three columns: 'ФИО (для физ лица)', 'Номер помещения', and 'Номер договора аренды'. The results show seven rows of data for the client 'Трой Б'.

	ФИО (для физ лица)	Номер помещения	Номер договора аренды
1	Трой Б	1	1
2	Трой Б	7	7
3	Трой Б	13	13
4	Трой Б	19	19
5	Трой Б	25	25
6	Трой Б	31	31
7	Трой Б	37	37