Bulk Insert and Reads

ISM 6218

Due on October 22th

The Avengers Team

"We will avenge every problem on our way"

Aitemir Yeskenov (Team Lead)

Nagarjuna Kanneganti

Sai Suraj Argula

Vinay Kumar Reddy Baradi

Table of Contents

Business Process Supported

For this assignment, we showed how to use **bulk insert** as well as **Linked Server, Open Query, OpenRowSet, OpenDataSource** and mapped the operations to the user stories of the XYZ company. We also compare execution plans as part of the summary for this assignment.

Requirements Described

First, bulk insert a set of rows.

Second, use: Linked Server, Open Query, OpenRowSet, OpenDataSource.

- Embed each function in a separate SPROC to support a User story.

Compare performance using execution plans.

Provide a summary of your experience using the alternatives.

First, bulk insert a set of rows.

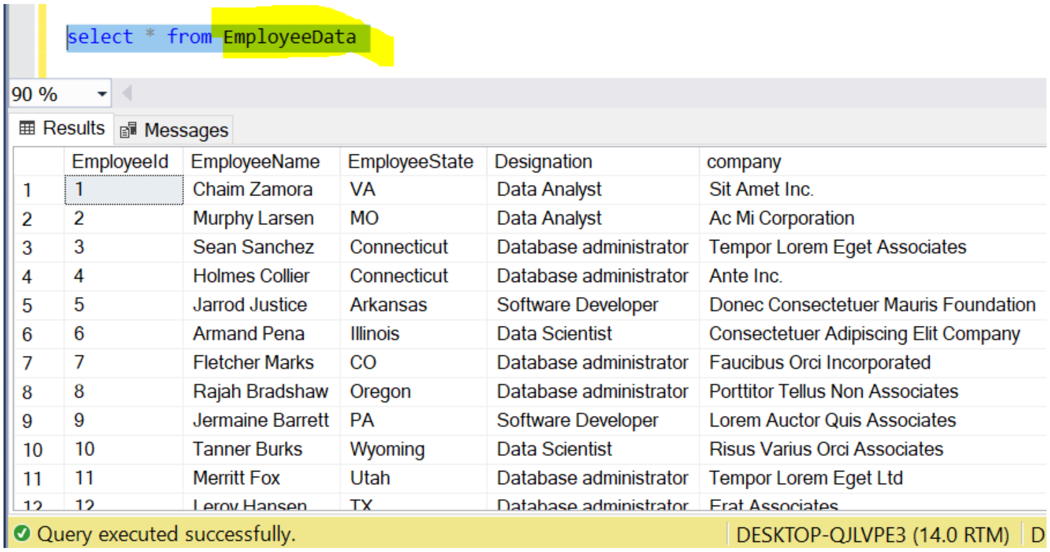
User Story:

Recently XYZ company hired some professionals in the field of Information Technology. HR wants to get the details of hired professionals to be stored in the database to automate the on-boarding process.

HR has Data in a CSV file. We created a EmployeeData table containing details of EmployeeID. Name, State, designation and company they are hired from. We used “Bulk Insert” to insert data from CSV file on local folder to the newly created table in Database.

Creating table

```
drop table if exists EmployeeData
create table EmployeeData (
EmployeeId  int,
EmployeeName varchar(50),
EmployeeState           nvarchar(50),
Designation     nvarchar(50),
company  nvarchar(50)
)
bulk insert EmployeeData from 'C:\Users\barad\Desktop\Employee.csv'
with
(
        FIELDTERMINATOR =',',
        ROWTERMINATOR ='\n',
                firstrow=2
)

select * from EmployeeData
```



```
select * from EmployeeData
```

	EmployeeId	EmployeeName	EmployeeState	Designation	company
1	1	Chaim Zamora	VA	Data Analyst	Sit Amet Inc.
2	2	Murphy Larsen	MO	Data Analyst	Ac Mi Corporation
3	3	Sean Sanchez	Connecticut	Database administrator	Tempor Lorem Eget Associates
4	4	Holmes Collier	Connecticut	Database administrator	Ante Inc.
5	5	Jarrod Justice	Arkansas	Software Developer	Donec Consectetuer Mauris Foundation
6	6	Armand Pena	Illinois	Data Scientist	Consectetuer Adipiscing Elit Company
7	7	Fletcher Marks	CO	Database administrator	Faucibus Orci Incorporated
8	8	Rajah Bradshaw	Oregon	Database administrator	Porttitor Tellus Non Associates
9	9	Jermaine Barrett	PA	Software Developer	Lorem Auctor Quis Associates
10	10	Tanner Burks	Wyoming	Data Scientist	Risus Varius Orci Associates
11	11	Merritt Fox	Utah	Database administrator	Tempor Lorem Eget Ltd
12	12	Leroy Hansen	TX	Database administrator	Erat Associates

Query executed successfully. | DESKTOP-QJLVPE3 (14.0 RTM) | D

Use: Linked Server, Open Query, OpenRowSet, OpenDataSource.

User Story:

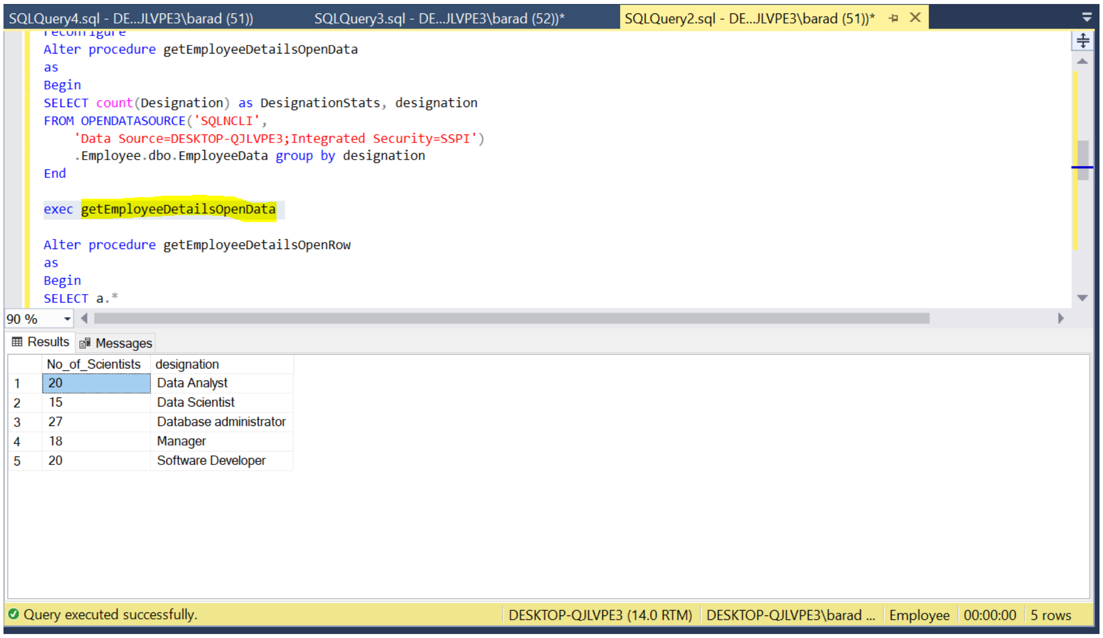
HR of the company XYZ wants to know about the hiring statistics (number of employees) of the various software roles they just hired.

For this we created stored procedures which will select data from the table we just created in the server DESKTOP-QJLVPE3 and group them by designation, using Linked Server, Open Query, OpenRowSet, OpenDataSource

OpenDataSource:

```
Alter procedure getEmployeeDetailsOpenData
as
Begin
SELECT count(Designation) as DesignationStats, designation
FROM OPENDATASOURCE('SQLNCLI',
    'Data Source=DESKTOP-QJLVPE3;Integrated Security=SSPI')
    .Employee.dbo.EmployeeData group by designation
End

exec getEmployeeDetailsOpenData
```

Result:



SQLQuery4.sql - DE...JLVPE3\barad (51))
SQLQuery3.sql - DE...JLVPE3\barad (52))*
SQLQuery2.sql - DE...JLVPE3\barad (51))*

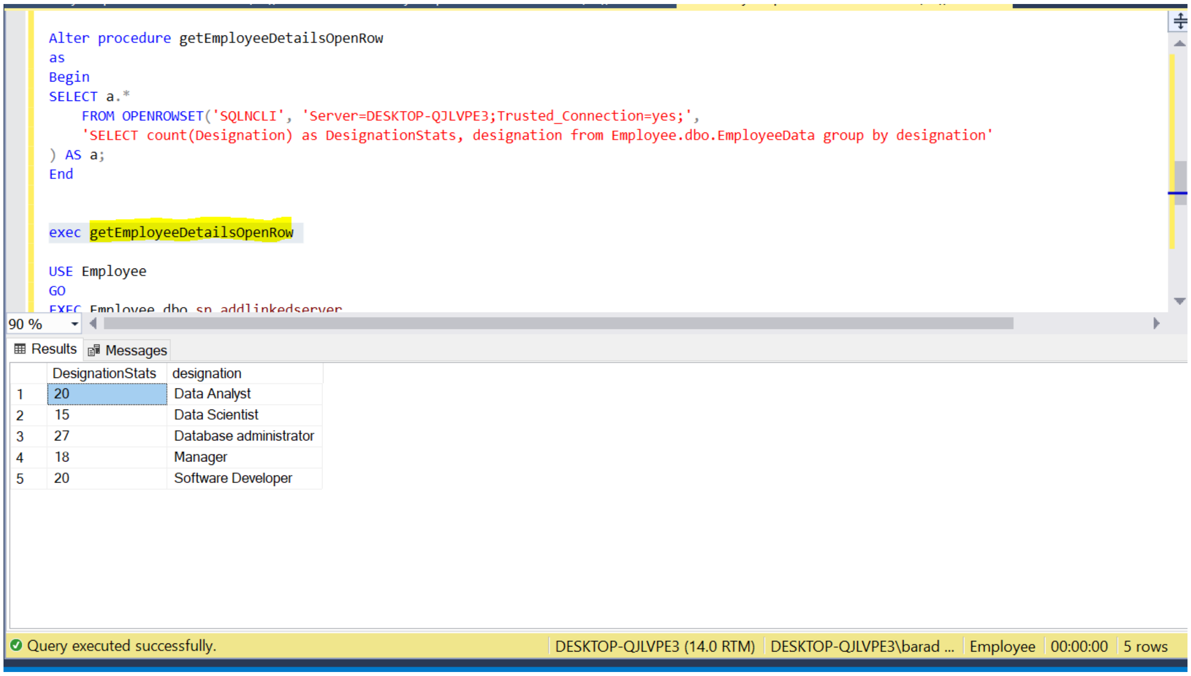
```
reconfigure
Alter procedure getEmployeeDetailsOpenData
as
Begin
SELECT count(Designation) as DesignationStats, designation
FROM OPENDATASOURCE('SQLNCLI',
    'Data Source=DESKTOP-QJLVPE3;Integrated Security=SSPI')
    .Employee.dbo.EmployeeData group by designation
End

exec getEmployeeDetailsOpenData

Alter procedure getEmployeeDetailsOpenRow
as
Begin
SELECT a.*
```

	No_of_Scientists	designation
1	20	Data Analyst
2	15	Data Scientist
3	27	Database administrator
4	18	Manager
5	20	Software Developer

Query executed successfully. | DESKTOP-QJLVPE3 (14.0 RTM) | DESKTOP-QJLVPE3\barad ... | Employee | 00:00:00 | 5 rows

OpenRowSet:

```
Alter procedure getEmployeeDetailsOpenRow
as
Begin
SELECT a.*
        FROM OPENROWSET('SQLNCLI', 'Server=DESKTOP-QJLVPE3;Trusted_Connection=yes;',
        'SELECT count(Designation) as DesignationStats, designation from Employee.dbo.EmployeeData group by designation'
) AS a;
End

exec getEmployeeDetailsOpenRow
```



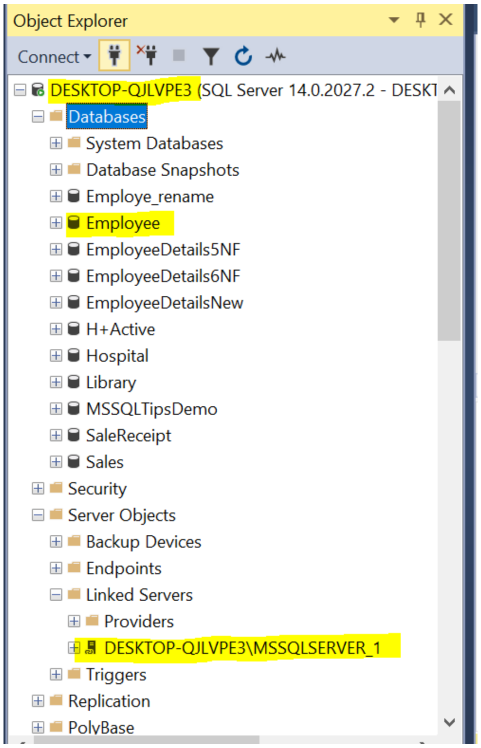
```
Alter procedure getEmployeeDetailsOpenRow
as
Begin
SELECT a.*
    FROM OPENROWSET('SQLNCLI', 'Server=DESKTOP-QJLVPE3;Trusted_Connection=yes;',
    'SELECT count(Designation) as DesignationStats, designation from Employee.dbo.EmployeeData group by designation'
) AS a;
End


exec getEmployeeDetailsOpenRow

USE Employee
GO
EXEC Employee.dbo.sp_addlinkedserver
```

	DesignationStats	designation
1	20	Data Analyst
2	15	Data Scientist
3	27	Database administrator
4	18	Manager
5	20	Software Developer

Query executed successfully. | DESKTOP-QJLVPE3 (14.0 RTM) | DESKTOP-QJLVPE3\barad ... | Employee | 00:00:00 | 5 rows

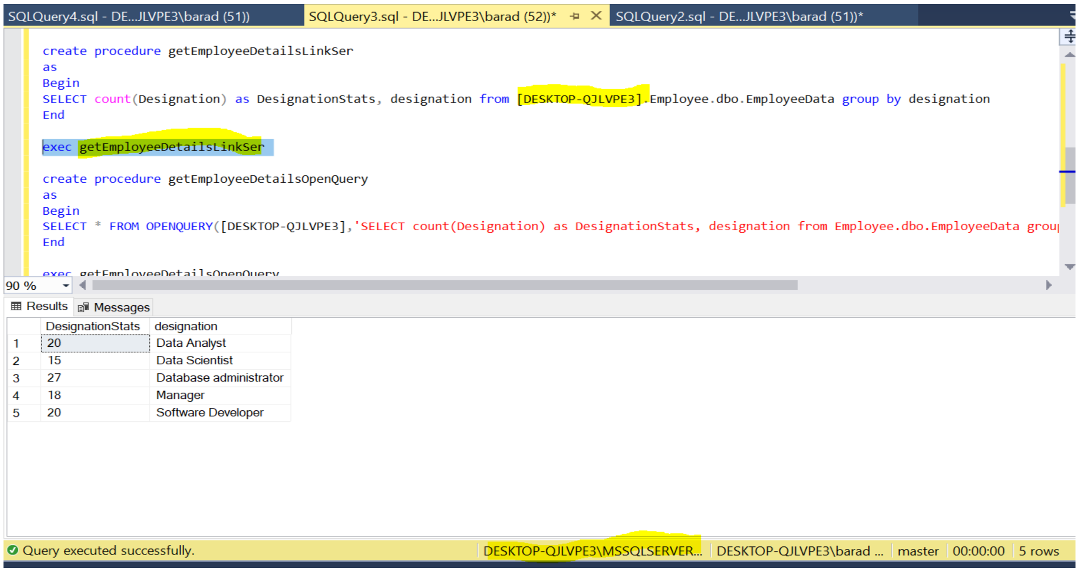
Linked Server:

First we linked two servers QJLVPE3 and QJLVPE3\MSSSQLSERVER_1 and ran the stored procedure which will query data from other server.



- DESKTOP-QJLVPE3 (SQL Server 14.0.2027.2 - DESKT
  - Databases
    - System Databases
    - Database Snapshots
    - Employe_rename
    - Employee
    - EmployeeDetails5NF
    - EmployeeDetails6NF
    - EmployeeDetailsNew
    - H+Active
    - Hospital
    - Library
    - MSSQLTipsDemo
    - SaleReceipt
    - Sales
  - Security
  - Server Objects
    - Backup Devices
    - Endpoints
    - Linked Servers
      - Providers
      - DESKTOP-QJLVPE3\MSSQLSERVER_1
    - Triggers
  - Replication
  - PolyBase

```
create procedure getEmployeeDetailsLinkSer
as
Begin
SELECT count(Designation) as DesignationStats, designation from [DESKTOP-QJLVPE3].Employee.dbo.EmployeeData group by designation
End

exec getEmployeeDetailsLinkSer
```


SQLQuery4.sql - DE...JLVPE3\barad (51))
SQLQuery3.sql - DE...JLVPE3\barad (52))*
SQLQuery2.sql - DE...JLVPE3\barad (51))*
90 %
Results
Messages

```
create procedure getEmployeeDetailsLinkSer
as
Begin
SELECT count(Designation) as DesignationStats, designation from [DESKTOP-QJLVPE3].Employee.dbo.EmployeeData group by designation
End

exec getEmployeeDetailsLinkSer

create procedure getEmployeeDetailsOpenQuery
as
Begin
SELECT * FROM OPENQUERY([DESKTOP-QJLVPE3],'SELECT count(Designation) as DesignationStats, designation from Employee.dbo.EmployeeData grou
End

exec getEmployeeDetailsOpenQuery
```

	DesignationStats	designation
1	20	Data Analyst
2	15	Data Scientist
3	27	Database administrator
4	18	Manager
5	20	Software Developer

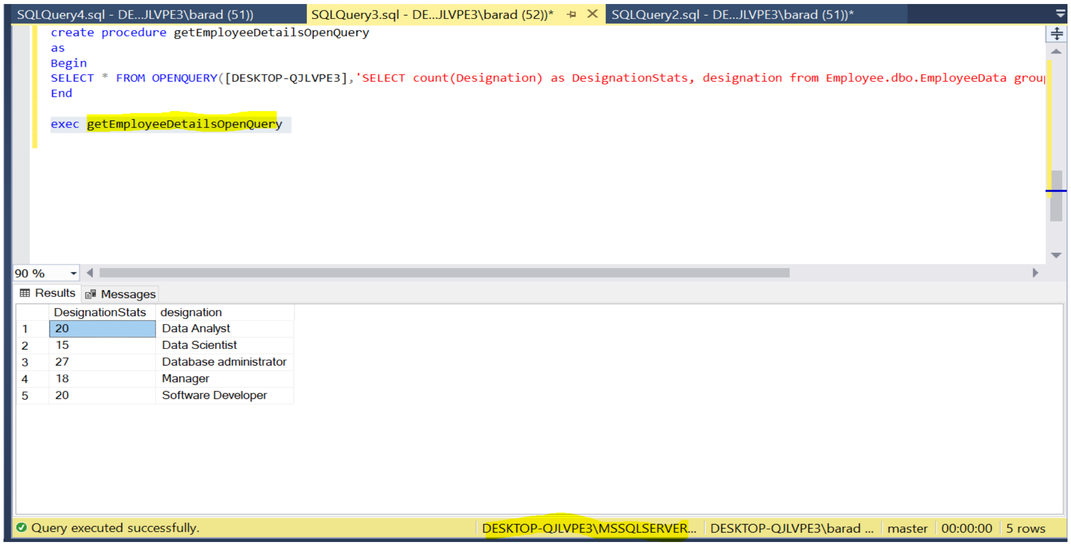
Query executed successfully. DESKTOP-QJLVPE3\MSSQLSERVER... DESKTOP-QJLVPE3\barad ... master 00:00:00 5 rows

OpenQuery:

OpenQuery also needs the Linked Server to be added, but since the LinkedServer is already added in the previous method we don’t have to do it again here.

create procedure getEmployeeDetailsOpenQuery
as
Begin
SELECT * FROM OPENQUERY([DESKTOP-QJLVPE3],
'SELECT count(Designation) as DesignationStats,
designation from Employee.dbo.EmployeeData group by designation')
End

exec getEmployeeDetailsOpenQuery



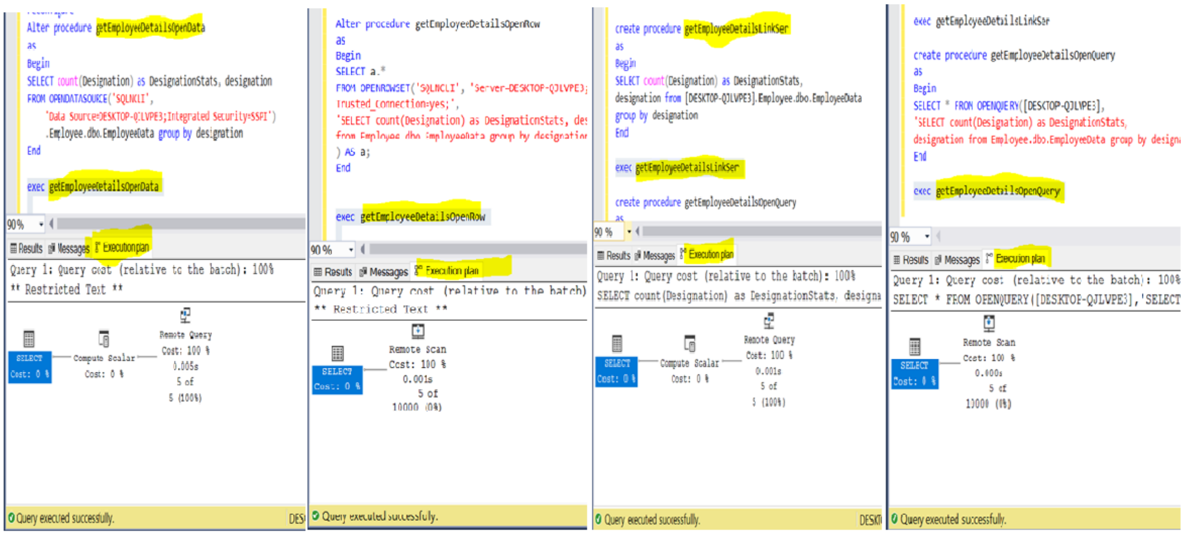
SQLQuery4.sql - DE...JLVPE3\barad (51))
SQLQuery3.sql - DE...JLVPE3\barad (52))*
SQLQuery2.sql - DE...JLVPE3\barad (51))*

```
create procedure getEmployeeDetailsOpenQuery
as
Begin
SELECT * FROM OPENQUERY([DESKTOP-QJLVPE3],'SELECT count(Designation) as DesignationStats, designation from Employee.dbo.EmployeeData grou
End

exec getEmployeeDetailsOpenQuery
```

	DesignationStats	designation
1	20	Data Analyst
2	15	Data Scientist
3	27	Database administrator
4	18	Manager
5	20	Software Developer

```
Query executed successfully.   DESKTOP-QJLVPE3\MSSQLSERVER...   DESKTOP-QJLVPE3\barad ...   master   00:00:00   5 rows
```

Execution Plan Comparison Between OpenQuery, OpenDataSource, OpenRowSet, LinkedServer



```
Alter procedure getEmployeeDetailsOpenData
as
Begin
SELECT count(Designation) as DesignationStats, designation
FROM OPENDATASOURCE('SQLNCLI',
    'Data Source=DESKTOP-QJLVPE3;Integrated Security=SSPI')
    .Employee.dbo.EmployeeData group by designation
End

exec getEmployeeDetailsOpenData
```

```
Query 1: Query cost (relative to the batch): 100%
** Restricted Text **

SELECT        Compute Scalar     Remote Query
Cost: 0 %     Cost: 0 %          Cost: 100 %
                                 0.005s
                                 5 of
                                 5 (100%)

Query executed successfully.
```

```
Alter procedure getEmployeeDetailsOpenRow
as
Begin
SELECT a.*
FROM OPENROWSET('SQLNCLI', 'Server=DESKTOP-QJLVPE3;
Trusted_Connection=yes;',
'SELECT count(Designation) as DesignationStats, des
from Employee.dbo.EmployeeData group by designation
) AS a;
End

exec getEmployeeDetailsOpenRow
```

```
Query 1: Query cost (relative to the batch)
** Restricted Text **

SELECT        Remote Scan
Cost: 0 %     Cost: 100 %
              0.001s
              5 of
              10000 (0%)

Query executed successfully.
```

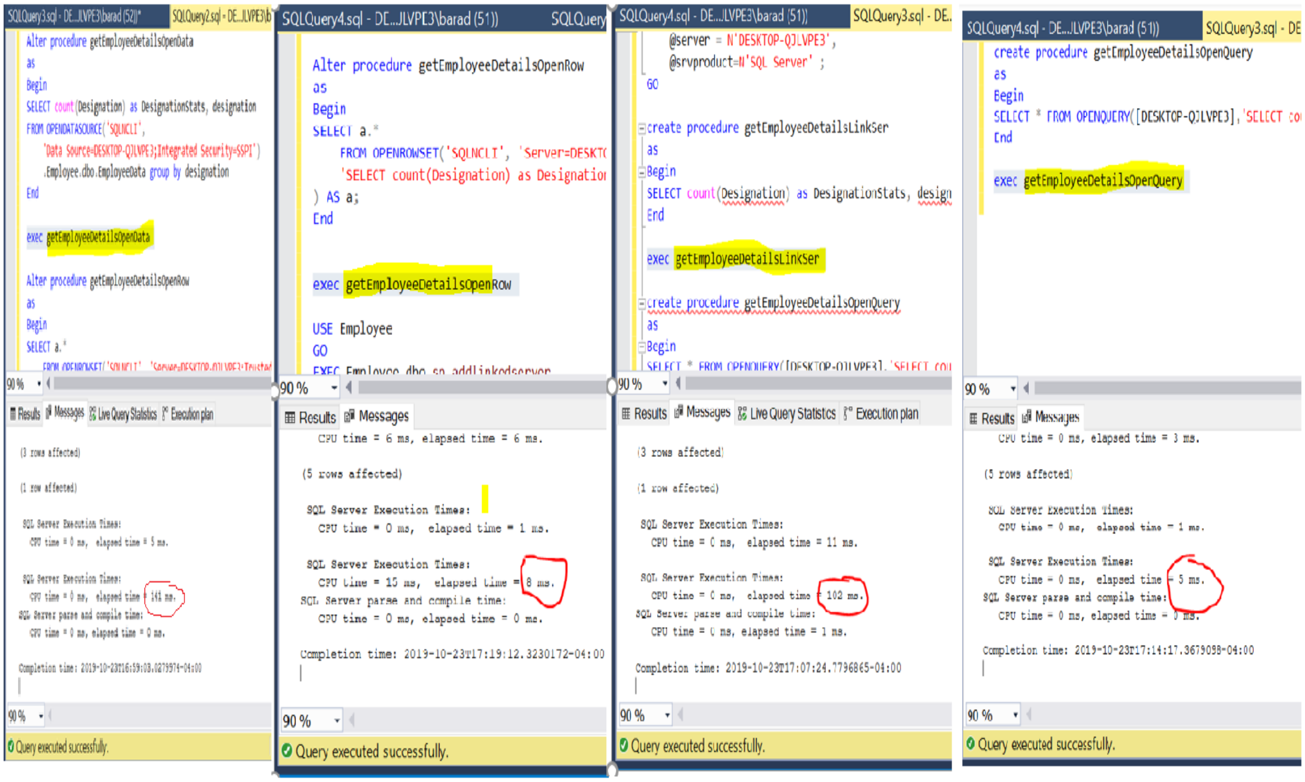
```
create procedure getEmployeeDetailsLinkSer
as
Begin
SELECT count(Designation) as DesignationStats,
designation from [DESKTOP-QJLVPE3].Employee.dbo.EmployeeData
group by designation
End

exec getEmployeeDetailsLinkSer

create procedure getEmployeeDetailsOpenQuery
as
```

```
Query 1: Query cost (relative to the batch): 100%
SELECT count(Designation) as DesignationStats, designa

SELECT        Compute Scalar     Remote Query
Cost: 0 %     Cost: 0 %          Cost: 100 %
                                 0.001s
                                 5 of
                                 5 (100%)

Query executed successfully.
```

```
exec getEmployeeDetailsLinkSer

create procedure getEmployeeDetailsOpenQuery
as
Begin
SELECT * FROM OPENQUERY([DESKTOP-QJLVPE3],
'SELECT count(Designation) as DesignationStats,
designation from Employee.dbo.EmployeeData group by designa
End

exec getEmployeeDetailsOpenQuery
```

```
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM OPENQUERY([DESKTOP-QJLVPE3],'SELECT

SELECT        Remote Scan
Cost: 0 %     Cost: 100 %
              0.000s
              5 of
              10000 (0%)

Query executed successfully.
```

Execution Time Comparison Between OpenQuery, OpenDataSource, OpenRowSet, LinkedServer


SQLQuery3.sql - DE...JLVPE3\barad (52))*
SQLQuery2.sql - DE...JLVPE3\b
90 %
Results
Messages
Live Query Statistics
Execution plan
Query executed successfully.

```
Alter procedure getEmployeeDetailsOpenData
as
Begin
SELECT count(Designation) as DesignationStats, designation
FROM OPENDATASOURCE('SQLNCLI',
    'Data Source=DESKTOP-QJLVPE3;Integrated Security=SSPI')
    .Employee.dbo.EmployeeData group by designation
End

exec getEmployeeDetailsOpenData

Alter procedure getEmployeeDetailsOpenRow
as
Begin
SELECT a.*
    FROM OPENROWSET('SQLNCLI', 'Server=DESKTOP-QJLVPE3;Trusted
```

```
(3 rows affected)

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 5 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 141 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

Completion time: 2019-10-23T16:59:03.0275974-04:00
```

SQLQuery4.sql - DE...JLVPE3\barad (51))
SQLQuery
90 %
Results
Messages
Query executed successfully.

```
Alter procedure getEmployeeDetailsOpenRow
as
Begin
SELECT a.*
    FROM OPENROWSET('SQLNCLI', 'Server=DESKT
    'SELECT count(Designation) as Designatio
) AS a;
End


exec getEmployeeDetailsOpenRow

USE Employee
GO
EXEC Employee.dbo.sp_addlinkedserver
```

```
   CPU time = 6 ms, elapsed time = 6 ms.

 (5 rows affected)

  SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 1 ms.

  SQL Server Execution Times:
   CPU time = 15 ms,  elapsed time = 8 ms.
 SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

 Completion time: 2019-10-23T17:19:12.3230172-04:00
```

```
    @server = N'DESKTOP-QJLVPE3',
    @srvproduct=N'SQL Server' ;
GO

create procedure getEmployeeDetailsLinkSer
as
Begin
SELECT count(Designation) as DesignationStats, design
End

exec getEmployeeDetailsLinkSer

create procedure getEmployeeDetailsOpenQuery
as
Begin
SELECT * FROM OPENQUERY([DESKTOP-QJLVPE3],'SELECT cou
```

```
(3 rows affected)

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 11 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 102 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 1 ms.

Completion time: 2019-10-23T17:07:24.7796865-04:00
```

SQLQuery4.sql - DE...JLVPE3\barad (51))
SQLQuery3.sql - DE
90 %
Results
Messages
Query executed successfully.

```
create procedure getEmployeeDetailsOpenQuery
as
Begin
SELECT * FROM OPENQUERY([DESKTOP-QJLVPE3],'SELECT co
End

exec getEmployeeDetailsOpenQuery
```

```
   CPU time = 0 ms, elapsed time = 3 ms.

 (5 rows affected)

  SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 1 ms.

  SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 5 ms.
 SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

 Completion time: 2019-10-23T17:14:17.3679098-04:00
```

Summary Comparison Between OpenQuery, OpenDataSource, OpenRowSet, LinkedServer:

Function	Execution Time
OpenRowSet	8ms
OpenDataSource	141ms
OpenQuery	5ms
LinkedServer	102ms

- From the above Summary table, the OpenQuery results to be the fastest method to Query a Remote server with 5ms, while OpenDataSource seems to be the slowest to Query a Remote server with 141ms.
- Also, there is a difference in the Execution Plans. OpenRowset and OpenQuery have similar execution plans, whereas OpenDataSource and LinkedServer have similar set of execution plans.
- Execution Plan for OpenDataSource and LinkedServer has an extra step of computing scaler which could be the reason for poor perfomance.
- LinkedServer method and OpenQuery method are different as initally we need to create a link between the servers and then execute the query whereas using other methods we can directly query the remote server.
- Also, using LinkedServer the Connection created to remote servers is permanent whereas in ad-hoc query methods the connection created is temporary or one-time.