Оглавление

[LabWork 2 2](#_Toc529722671)

[Tasks 1: 3](#_Toc529722672)

[**TASK1 .Extra option:** 5](#_Toc529722673)

[TASK2. Create the HumanResources Database and the InternetSales Database: 8](#_Toc529722674)

[**TASK 2.1 ADDITIONAL** 10](#_Toc529722675)

[TASK 2.2 ADDIT 15](#_Toc529722676)

[**TASK 2.3 ADDIT** 16](#_Toc529722677)

Для большей наглядности я каждое задание начинал с нового листа. И чтобы можно было разобраться – сделал оглавление.

# LabWork 2

Prerequisites:

At least 1 running SQL server

At list 2 attached not system disk in the OS – to place users DBs files (ALL files of custom DBs)

Tools to work with SQL server

Pay attention to best practices and performance optimization for SQL server and databases.

## Tasks 1:

1. Reconfigure TEMPDB to another database files location and options (use custom folder):

**Main DB file:**

Size: 10 MB

File growth: 5 MB

Maximum size: Unlimited

File location: on the attached disk

**DB log file:**

Size: 10 Mb

File growth: 1 MB

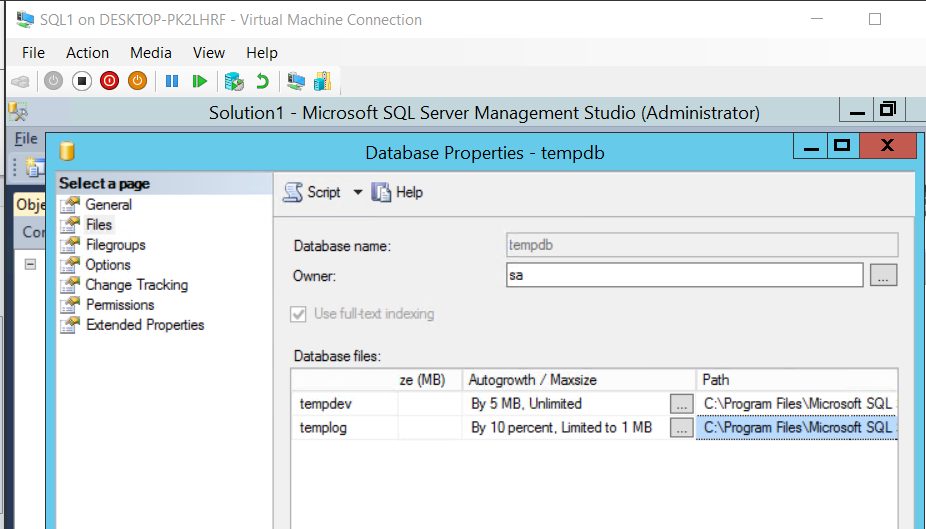
Maximum size: Unlimited

File location: on the attached disk

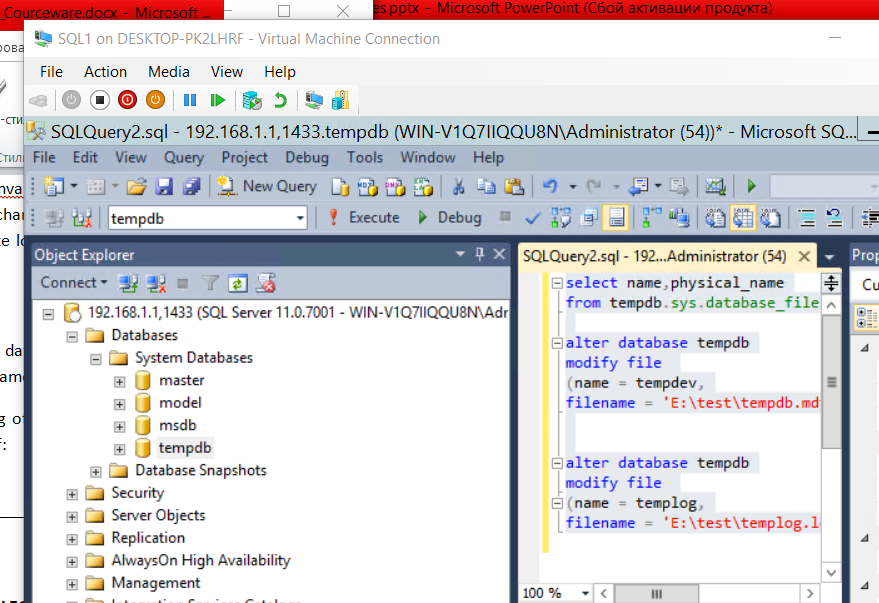
Restart SQL to apply changes

Delete old files

Видно, что размеры, прирост менять можно прямо тут. Менять же Path из данного окна невозможно(можно менять только при создании бд)



Смена пути БД



КОД:

select name,physical\_name

from tempdb.sys.database\_files

alter database tempdb

modify file

(name = tempdev,

filename = 'E:\test\tempdb.mdf');

alter database tempdb

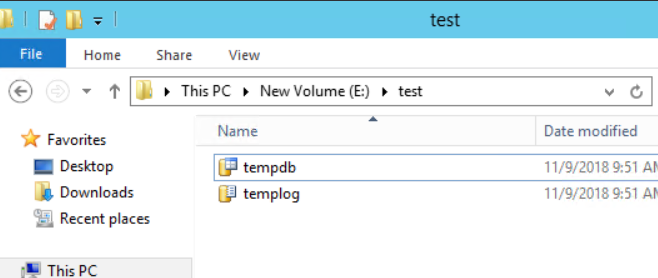
modify file

(name = templog,

filename = 'E:\test\templog.ldf');

Перезапуск бд.

Результат:



**TASK1 .Extra option:**

Script it using PowerShell:

Code should be wrapped in PowerShell script, started from student local PC, Password isn’t provided as a plain text, errors capturing is active, code returns DB files locations before execution and after, check the existing files with the same name in the target location, and check free space on disk. To check DB with TSQL:

SELECT name, physical\_name,size,max\_size,growth

FROM sys.master\_files

WHERE database\_id = DB\_ID(N'tempdb');

**Проблема, которая у меня произошла:**

Я сдаюсь. У меня баг, который я пытался пофиксить много часов. Локально код работает. Я также могу подключиться удаленно к повершелу. Но когда я пытаюсь подключиться удаленно к **повершелу, а потом** запустить этот скрипт – ничего не выходит. Ощущение что мой пк не успевает подключиться к другой машине и просто продолжает исполнять код. Пробовал ставить паузы, задержки – ничего не поменялось.

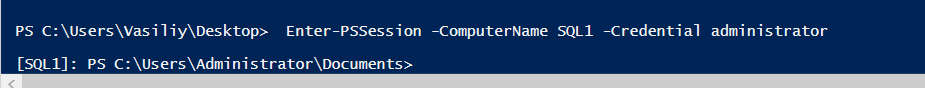
Если же я удаленно **уже подключился к повершелу и запустил скрипт**- все работает.

**Вот Код если я уже подключен удаленно, т.е все работает**

**КОД:**

#вот я успешно подключился

Enter-PSSession -ComputerName SQL1 -Credential administrator



#вот я запустик код и все отработало хорошо

cls

Enter-PSSession -ComputerName SQL1 -Credential administrator

#try – как попытка проверки на ошибки

try{

write-host "before installation"

Get-ChildItem E:\test

#залазим в SQL и меняем по заданию файлы

Invoke-Sqlcmd -Query "

alter database tempdb

modify file

(name = tempdev,

size = 10MB, maxsize= Unlimited, FILEGROWTH = 5MB

);

alter database tempdb

modify file

(name = templog,

size = 10MB, maxsize= Unlimited, FILEGROWTH = 1MB

);

alter database tempdb

modify file

(name = tempdev,

filename = 'E:\test\extraoption\tempdb.mdf');

alter database tempdb

modify file

(name = templog,

filename = 'E:\test\extraoption\tempdb.ldf');

"

#-credential (get-credential) - togda ne pobit, skoree vsego izza try i catch

#-serverinstance 192.168.1.1 - togda ne pobit, skoree vsego izza try i catch

write-host "after installation"

Get-ChildItem E:\test\extraoption

#лезем в SQL 2 раз и вытягиваем параметры, которые написаны в селекте

$needsize = Invoke-Sqlcmd -Query "SELECT name, physical\_name,size,max\_size,growth

FROM sys.master\_files

WHERE database\_id = DB\_ID(N'tempdb');"

#переменная size – будет считать ток сумму того, скок занимают файлыбазы SQL и сравнинвают с физическим место

$size = $needsize | Measure-Object -Property size -Sum

if ($size.sum -ge $disk.FreeSpace ){write-host "da, mesta hvataet" -foregroundcolor green} else {write-host "net, mesta ne hvataet" -foregroundcolor red}

#переменная диск- скок физически свободного мета

#переменная диск- скок физически свободного мета

$disk = Get-WmiObject Win32\_LogicalDisk -Filter "DeviceID='E:'" |

Select-Object Size,FreeSpace

}

#первая попытка отладчика, без фактической отладки.

catch {

write-host "The error:"

write-host $\_ -ForegroundColor Yellow

write-host "Call Admin! "}

#проверка на совпадания имени

$location="E:\test\extraoption"

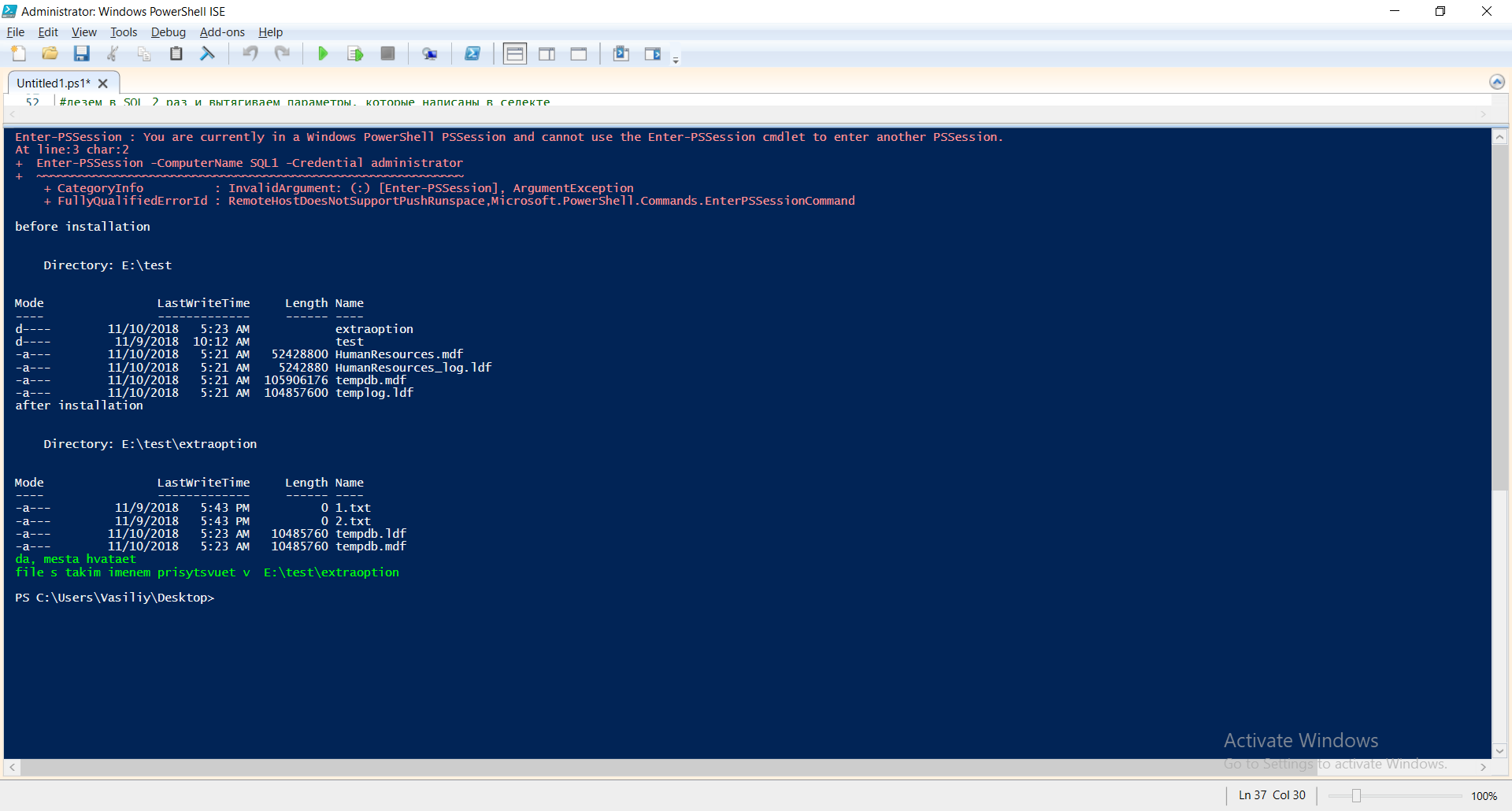
Set-Location $location

Get-ChildItem | foreach {if ($\_.name -eq "1.txt") {Write-Host "file s takim imenem prisytsvuet v $location" -foregroundcolor green} }

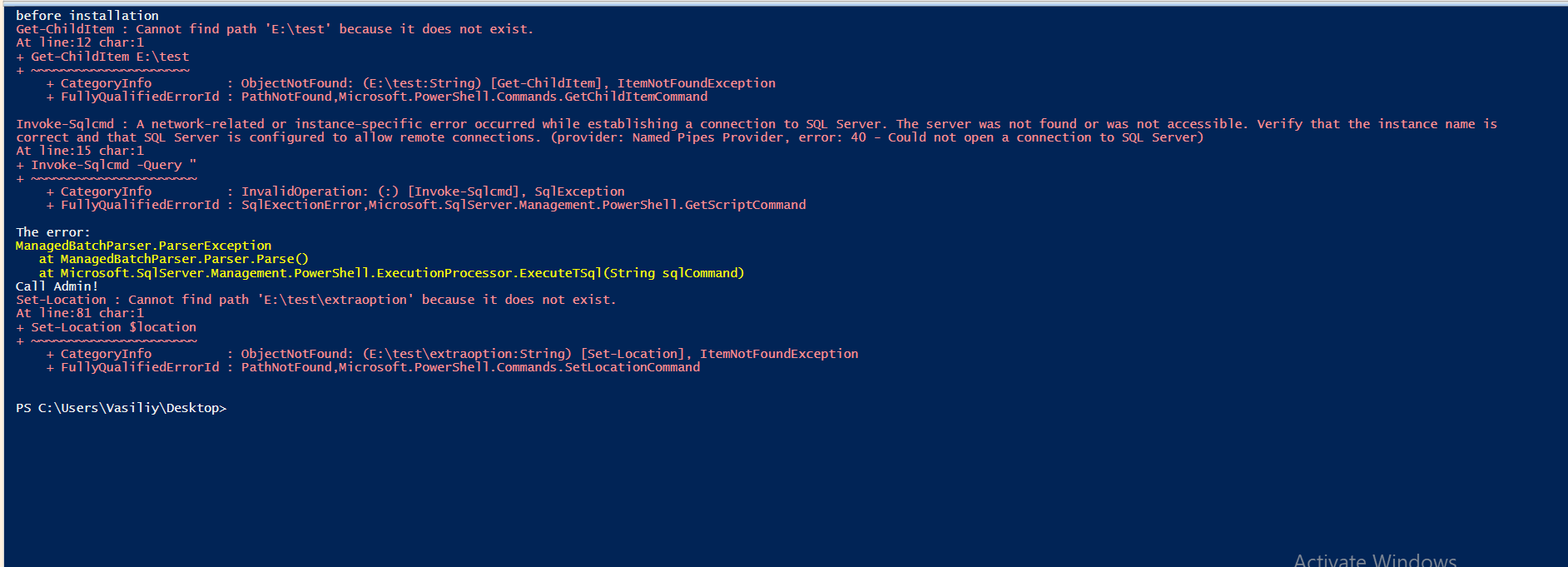
exit

**Результаты:**

Если я уже был подключен удаленно, то код отрабатывает хорошо

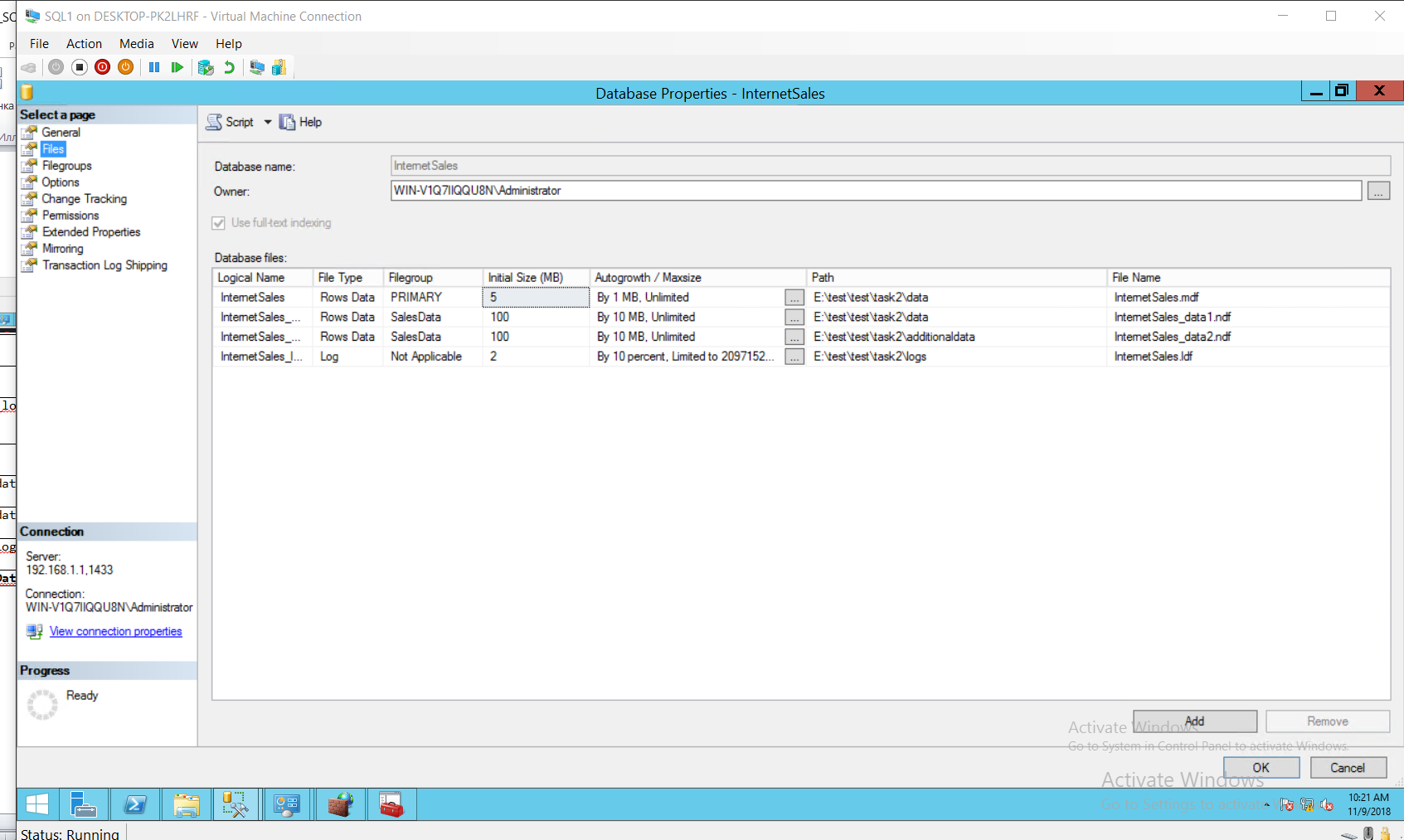


**А вот если я пытаюсь подключиться к удаленке, а потом запустить код. Видно, что код попадает в положение catch, т.е. try не отрабатывает.**

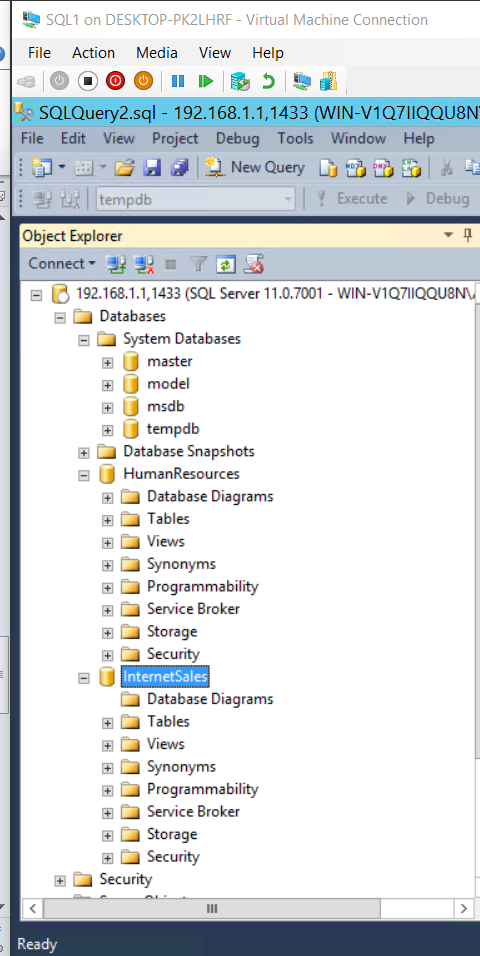


## TASK2. Create the HumanResources Database and the InternetSales Database:

Тут задаем параметры по заданию



И видим созданные бд



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Logical Name | Filegroup | Initial Size | Growth | Path |
| HumanResources | PRIMARY | 50 MB | 5MB/Unlimited | D:\Data\HumanResources.mdf |
| HumanResources\_log |  | 5 MB | 1 MB/Unlimited | D:\Logs\HumanResources.ldf |
| InternetSales | PRIMARY | 5 MB | 1 MB / Unlimited | D:\Data\InternetSales.mdf |
| InternetSales\_data1 | SalesData | 100 MB | 10 MB / Unlimited | D:\Data\InternetSales\_data1.ndf |
| InternetSales\_data2 | SalesData | 100 MB | 10 MB / Unlimited | D:\AdditionalData\InternetSales\_data2.ndf |
| InternetSales\_log |  | 2 MB | 10% / Unlimited | D:\Logs\InternetSales.ldf |
| Make the **SalesData** filegroup the default filegroup | | | | |

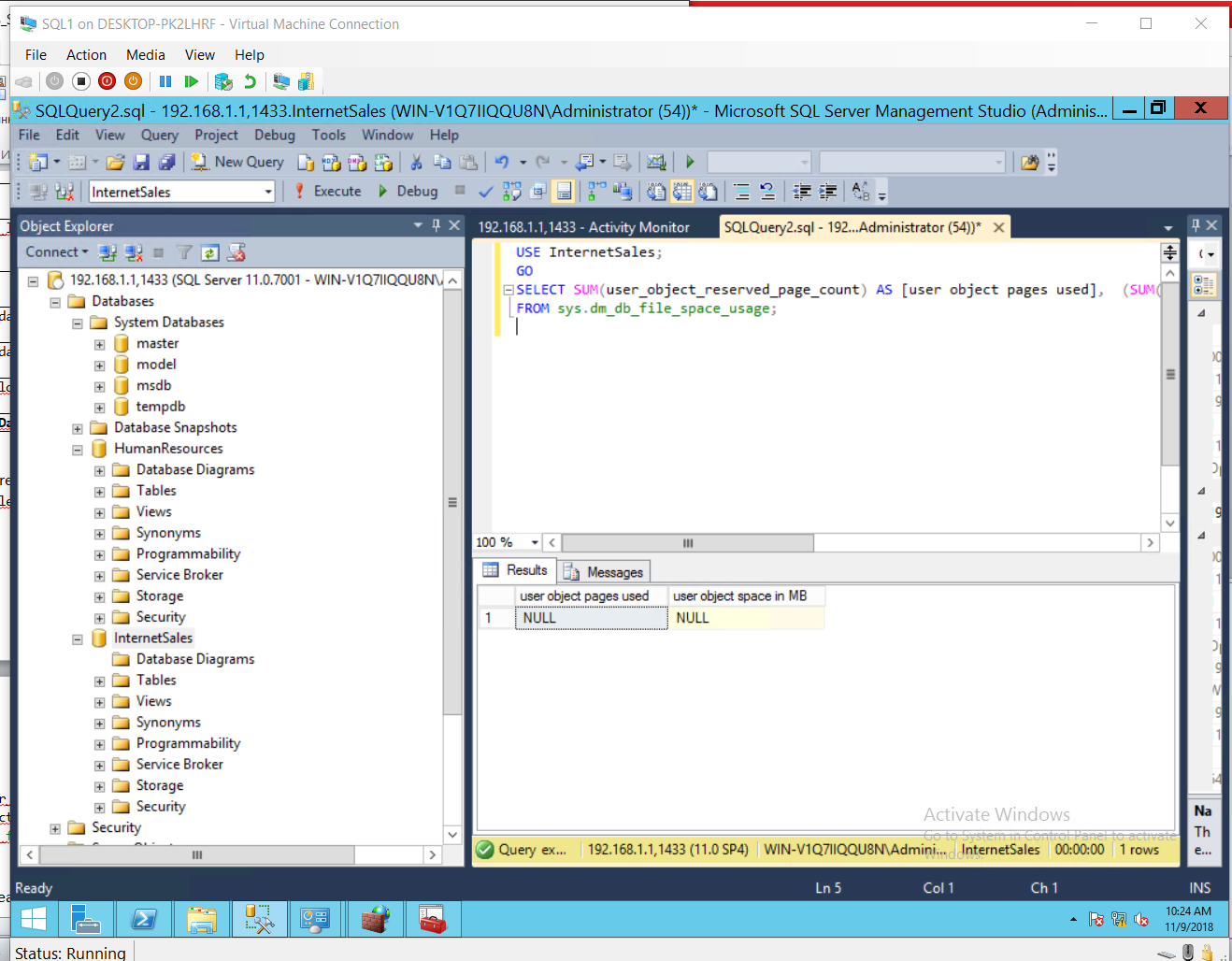
Check usage of created DBs space:

USE InternetSales;

GO

SELECT SUM(user\_object\_reserved\_page\_count) AS [user object pages used], (SUM(user\_object\_reserved\_page\_count)\*1.0/128) AS [user object space in MB]

FROM sys.dm\_db\_file\_space\_usage;



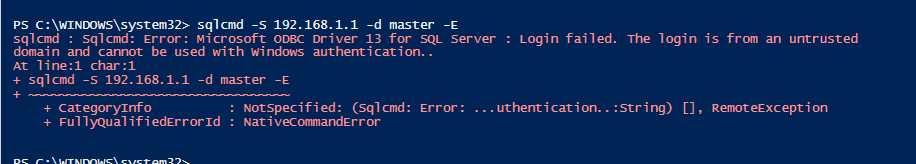
**TASK 2.1 ADDITIONAL**

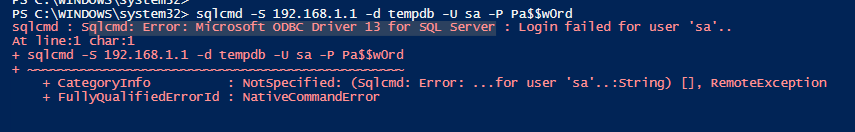
1. Wrap DBs creation in PS Script, lab task 2 should be executed as non-stop script with Human readable output, all scripts start from Student local PC. Check if DBs with such names already exist, if yes – drop them with their files, check if everything was succesfull, and create new DBs. NO REMOTE PS SESSIONS or invoke-expressions on remote PC.

Исходя из задания получается, что это должен быть PS скрипт, но без поднятия сессий или вызовов на удаленном компе. Я вижу только одно решение – попробую sqlcmd запрос всунуть в PS скрипт. Получится и скрипт есть и нет настройки удаленной сессии.

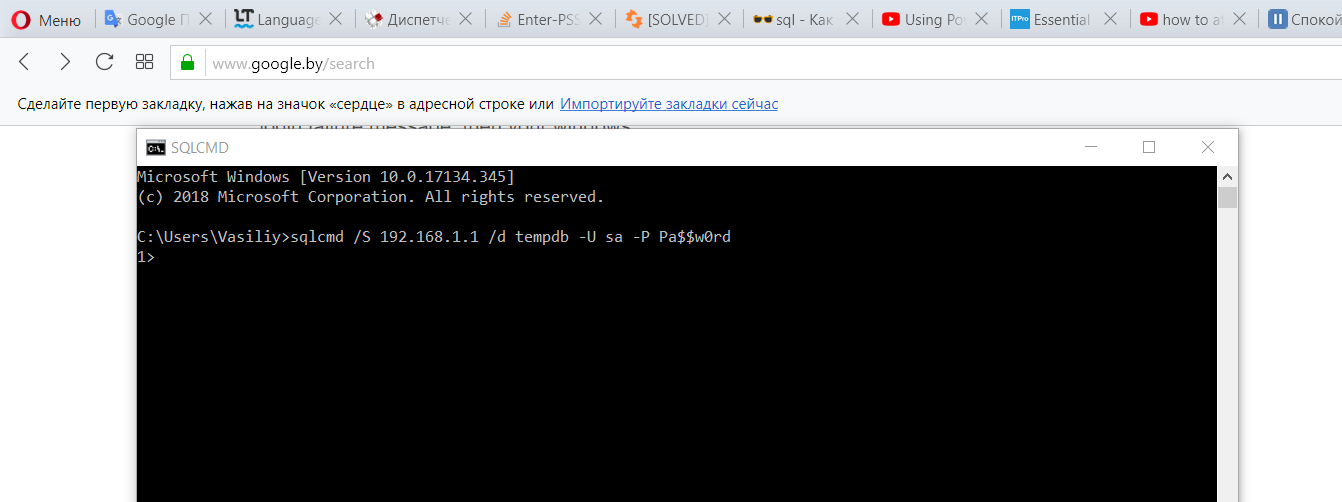
**Неудачные попытки:**

C хоста подключение через PS





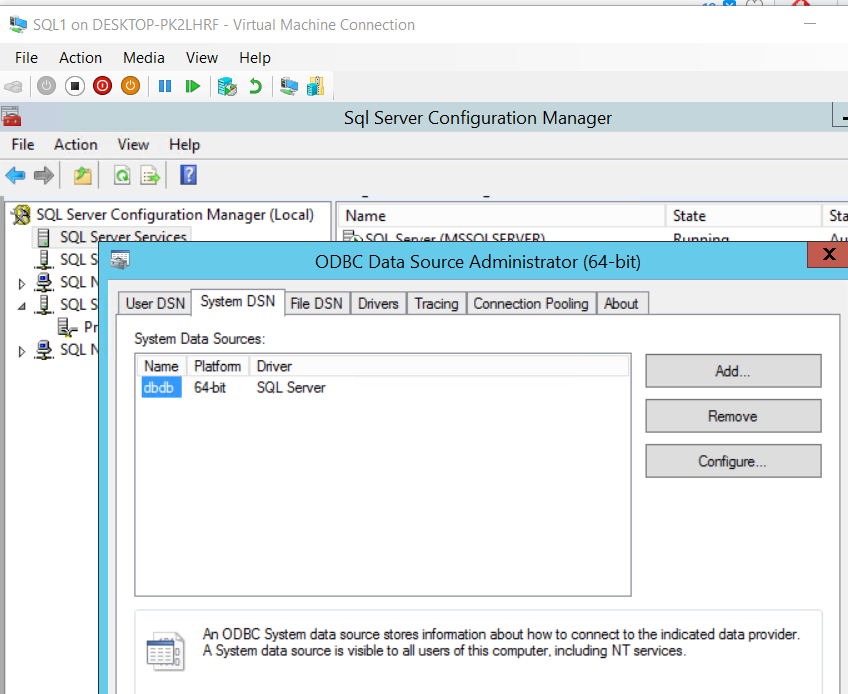
С хоста подключение через sqlcmd. Это странно.

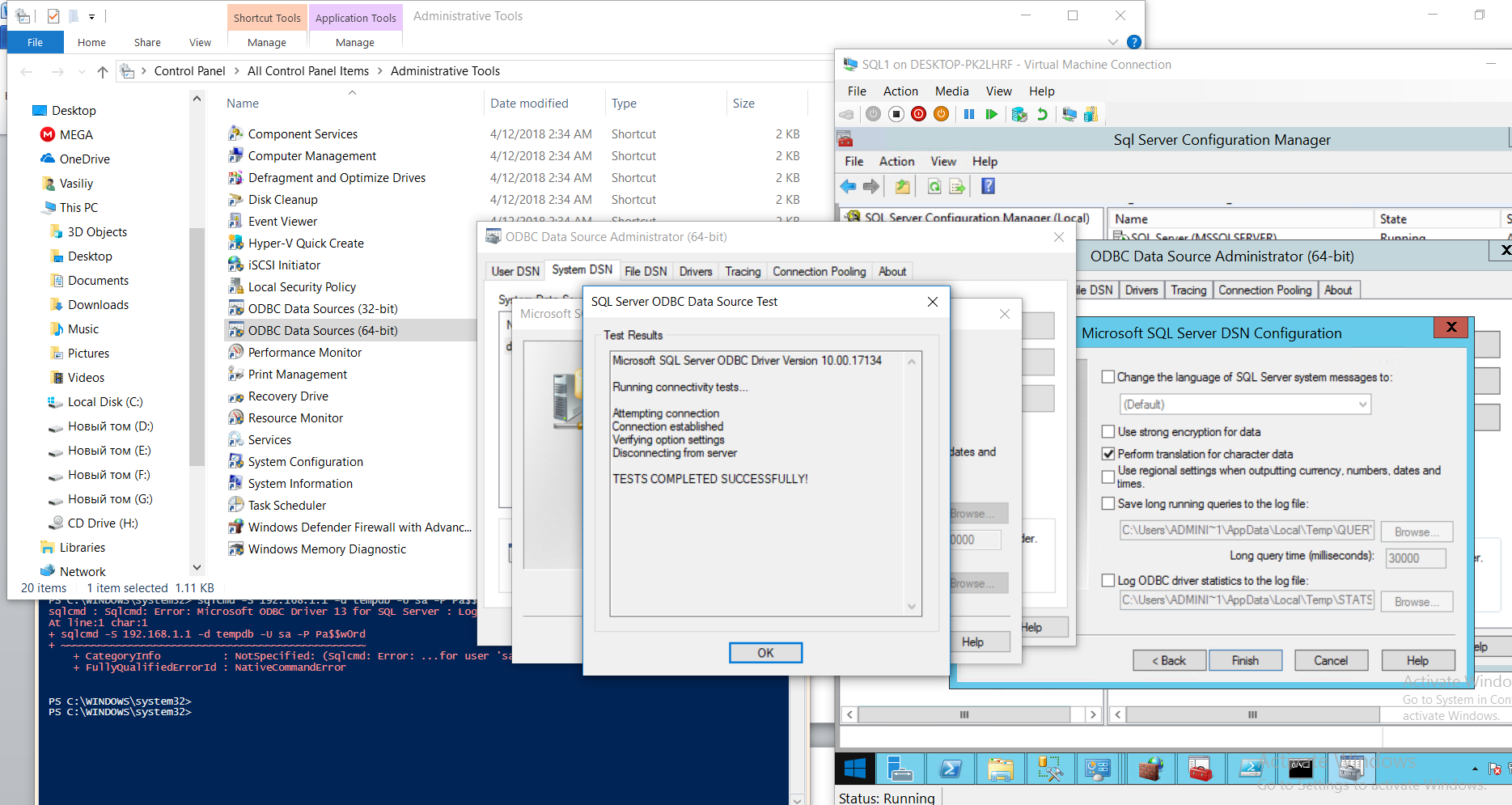


Это очень странно. Вроде тоже самое действие, только из разных источников. Должноработать.

Пишет ошибку ODBC, попробуем разобраться что это.

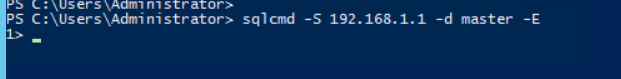
Добавлял всех в разрешение на подключение. Устанавливал ODBC драйвер везде. Не помогло.





Как я понял ODBC– это для программистов, чтобы программа через ODBC API стягивала к себе инфу из бд.

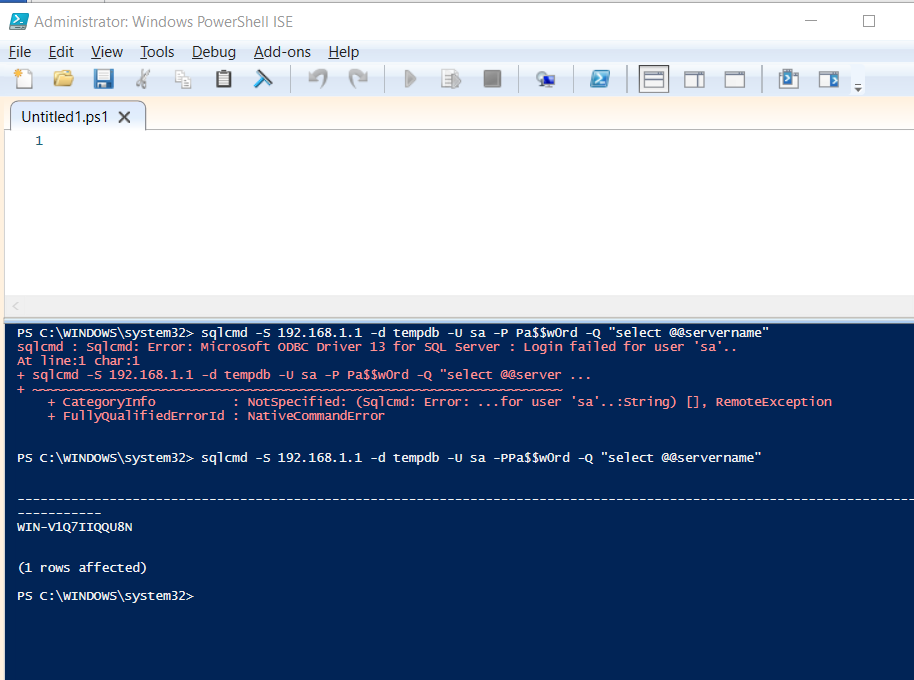
С удаленного пк с самогосебя на себя норм. Что опять же странно.



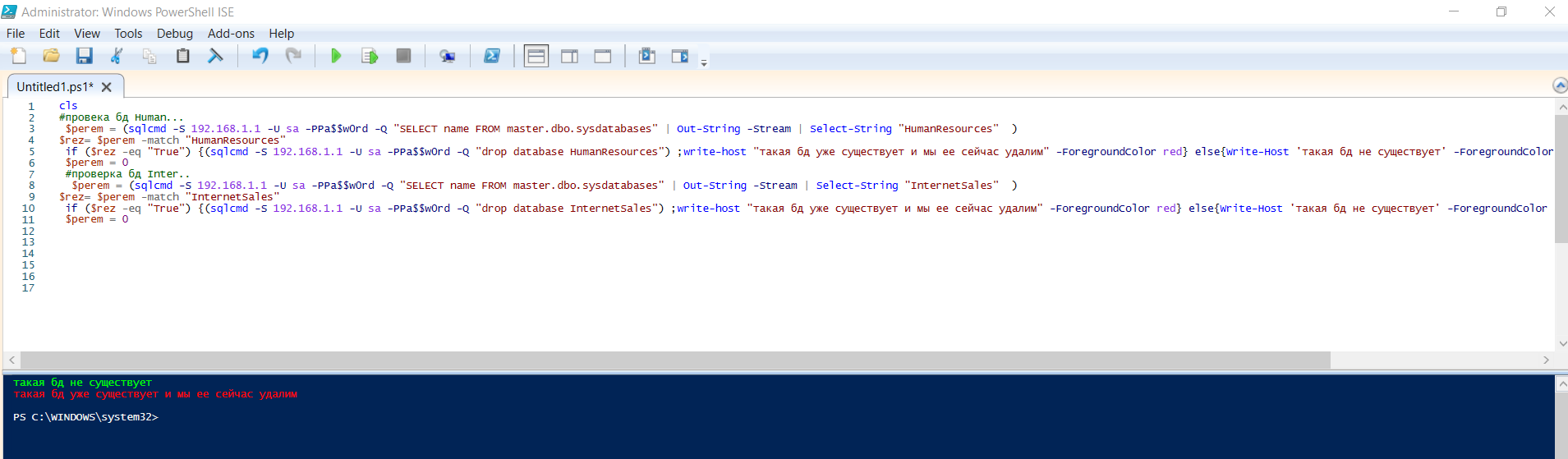
**УДАЧНАЯ ПОПЫТКА**

Решение для подключения было найдено. (Роман Богатырев подсказал). И оно очень странное. Надо с хостовой машины иписать при подключение не –P Pa$$w0rd, а –Ppa$$w0rd. Это похоже на какой-то баг команды от майкрософта , но это работает.

Демонстрация бага (Слитная и раздельная команда). Это подключение с хоста к вм



**Часть 1 Проверка 1ой части скрипта. Удаление существующих таблиц- отрабатывает. Визуально в бд это увидел.**



**Finalcode**

cls

#провека бд Human...

$perem = (sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "SELECT name FROM master.dbo.sysdatabases" | Out-String -Stream | Select-String "HumanResources" )

$rez= $perem -match "HumanResources"

if ($rez -eq "True") {(sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "drop database HumanResources") ;write-host "такая бд уже существует и мы ее сейчас удалим" -ForegroundColor red} else{Write-Host 'такая бд не существует' -ForegroundColor Green}

$perem = 0

#проверка бд Inter..

$perem = (sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "SELECT name FROM master.dbo.sysdatabases" | Out-String -Stream | Select-String "InternetSales" )

$rez= $perem -match "InternetSales"

if ($rez -eq "True") {(sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "drop database InternetSales") ;write-host "такая бд уже существует и мы ее сейчас удалим" -ForegroundColor red} else{Write-Host 'такая бд не существует' -ForegroundColor green}

$perem = 0

#созда бд Human

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "create database HumanResources

on

PRIMARY

(

Name=HumanResourcesdev,

FileName='E:\test\extraoption\Humanresources.mdf',

size=50mb,

Filegrowth=5mb,

MaxSize=Unlimited

)

log on

(

Name=Humanresourceslog,

size=5mb,

FileName='E:\test\extraoption\Humanresources.log',

Filegrowth=5mb,

MaxSize=Unlimited

)

"

#созда бд Internetsales

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "create database InternetSales

on

PRIMARY

(

Name=HumanResourcesdev,

FileName='E:\test\extraoption\InternetSales.mdf',

size=50mb,

Filegrowth=5mb,

MaxSize=Unlimited

)

log on

(

Name=Humanresourceslog,

size=5mb,

FileName='E:\test\extraoption\InternetSales.log',

Filegrowth=5mb,

MaxSize=Unlimited

)

"

## TASK 2.2 ADDIT

1. Using PowerShell, create and run script from local PC, which deploys:

Data Base “PCDRIVE” (with parameters like HumanResources)

DB contains tables with results of command:

Get-PhysicalDisk | select -Property FriendlyName,BusType,HealthStatus,Size,MediaType

Data types and names for columns could be selected by yourself

Compare pages and files sizes before and after filling data to DB (PS Script should show it in Human readable format).

Save logging of script actions with start and stop time of execution. (Decide what should be logged – it is your own choice).

cls

write-host "before script" -ForegroundColor Green

Invoke-Command -Credential administrator -ComputerName 192.168.1.1 -ScriptBlock{ Get-ChildItem E:\test\2.2}

#созда бд PCDRIVE

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "create database PCDRIVE

on

PRIMARY

(

Name=PCDRIVEdev,

FileName='E:\test\2.2\PCDRIVEmdf',

size=50mb,

Filegrowth=5mb,

MaxSize=Unlimited

)

log on

(

Name=PCDRIVElog,

size=5mb,

FileName='E:\test\2.2\PCDRIVElog',

Filegrowth=5mb,

MaxSize=Unlimited

)

"

#созд таблицу

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

create table PCDRIVE.dbo.PhysicalDisk

(

FriendlyName varchar(50) null,

BusType varchar(10) NUlL,

HealthStatus varchar(10) Null,

Size varchar(50) null,

MediaType binary(10) null

);

"

#вывести парам жд вм ,просто посмотреть что есть

Invoke-Command -Credential administrator -ComputerName 192.168.1.1 -ScriptBlock{ Get-PhysicalDisk |

select -Property FriendlyName,BusType,HealthStatus,Size,MediaType}

#создать переменную $friendlyname, которая массивом раскидается по таблице dbo.PhysicalDisk в столбике friendlyname

$friendlyname = Invoke-Command -Credential administrator -ComputerName 192.168.1.1 -ScriptBlock{ Get-PhysicalDisk |

select -Property FriendlyName}

#разбить переменную $friendlyname на несколько переменных , каждая со своих ssd/hdd

$friendlyname0=$friendlyname[0].FriendlyName

$friendlyname1=$friendlyname[1].FriendlyName

#один ссд - в 1ый кортеж в 1 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([friendlyname]) values ('$friendlyname0')"

#второй hdd - в 2ый кортеж в 1 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([friendlyname]) values ('$friendlyname1')"

#создать переменную $Butstype, которая массивом раскидается по таблице dbo.PhysicalDisk в столбике Butstype

$BusType = Invoke-Command -Credential administrator -ComputerName 192.168.1.1 -ScriptBlock{ Get-PhysicalDisk |

select -Property BusType}

#разбить переменную $Butstype на несколько переменных , каждая для своей шины для каждого диска

$Bustype0=$BusType[0].Bustype

$Bustype1=$BusType[1].Bustype

#первая шина в 1 кортеж 2 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([Bustype]) values ('$Bustype0')"

#вторая шина в 2 кортеж 2 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([Bustype]) values ('$Bustype1')"

#создать переменную $HealthStatus, которая массивом раскидается по таблице dbo.PhysicalDisk в столбике HealthStatus

$HealthStatus = Invoke-Command -Credential administrator -ComputerName 192.168.1.1 -ScriptBlock{ Get-PhysicalDisk |

select -Property HealthStatus}

#разбить переменную $Butstype на несколько переменных , каждая для своей шины для каждого диска

$HealthStatus0=$HealthStatus[0].HealthStatus

$HealthStatus1=$HealthStatus[1].HealthStatus

#1ый хелсчек в 1 кортеж 3 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([HealthStatus]) values ('$HealthStatus0')"

#2ый хелсчек в 2 кортеж 3 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([HealthStatus]) values ('$HealthStatus1')"

#создать переменную $size, которая массивом раскидается по таблице dbo.PhysicalDisk в столбике Size

$Size = Invoke-Command -Credential administrator 192.168.1.1 -ScriptBlock{ Get-PhysicalDisk |

select -Property Size}

#разбить переменную $Size на несколько переменных

$Size0= $Size[0].Size

$Size1= $Size[1].size

#1ый Size в 1 кортеж 4 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([Size]) values ('$Size0')"

#2ый Size в 2 кортеж 4 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([Size]) values ('$Size1')"

#создать переменную $MediaType, которая массивом раскидается по таблице dbo.PhysicalDisk в столбике MediaType

$MediaType = Invoke-Command -Credential administrator 192.168.1.1 -ScriptBlock{ Get-PhysicalDisk |

select -Property MediaType}

#разбить переменную $MediaType на несколько переменных

$MediaType0= $MediaType[0].PSComputerName

$MediaType1= $MediaType[1].PSComputerName

#1ый MediaType в 1 кортеж 5 атрибут, если бд была пустая

sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([MediaType]) values ('$MediaType0')"

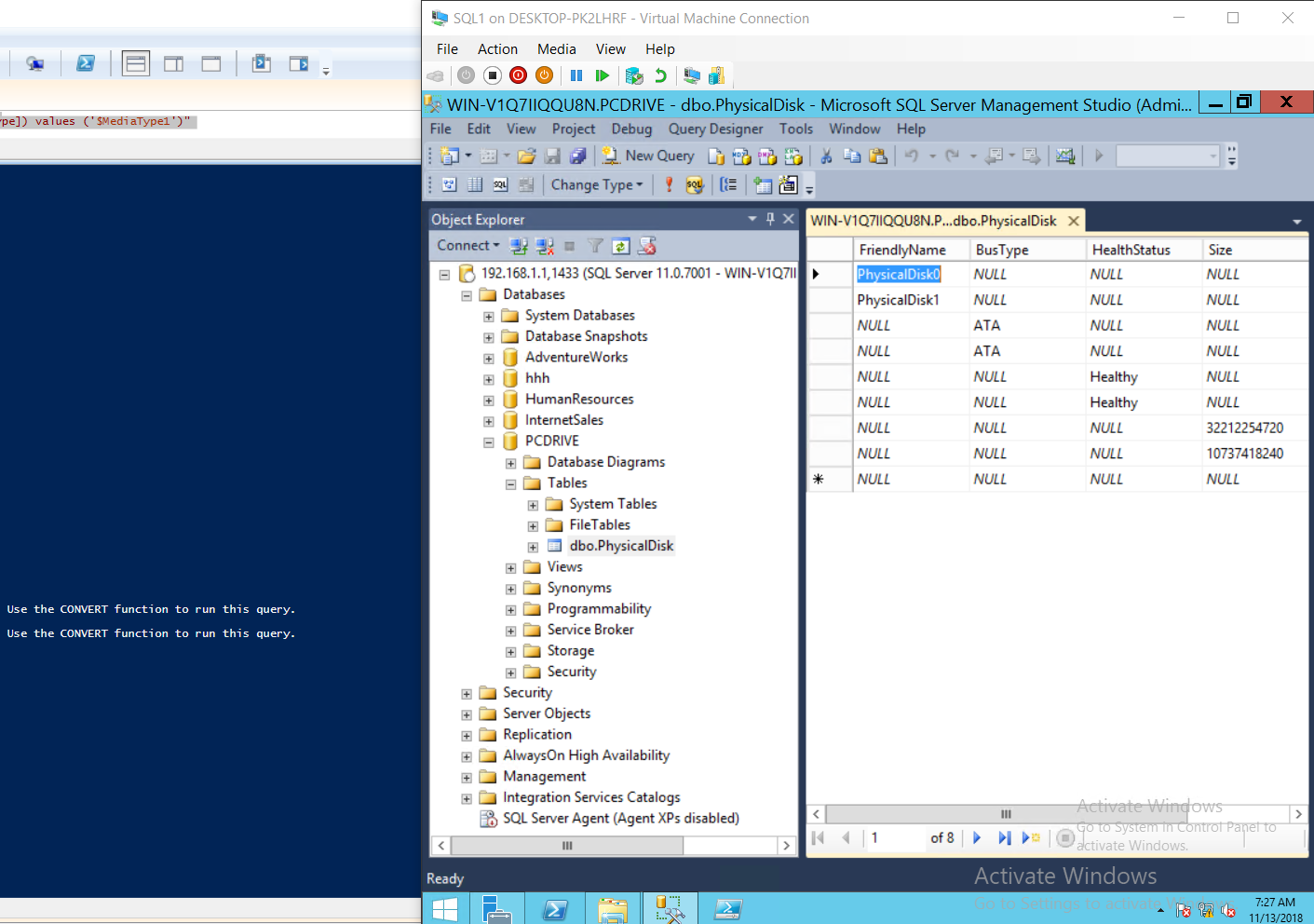
#2ый MediaType в 2 кортеж 5 атрибут, если бд была пустая

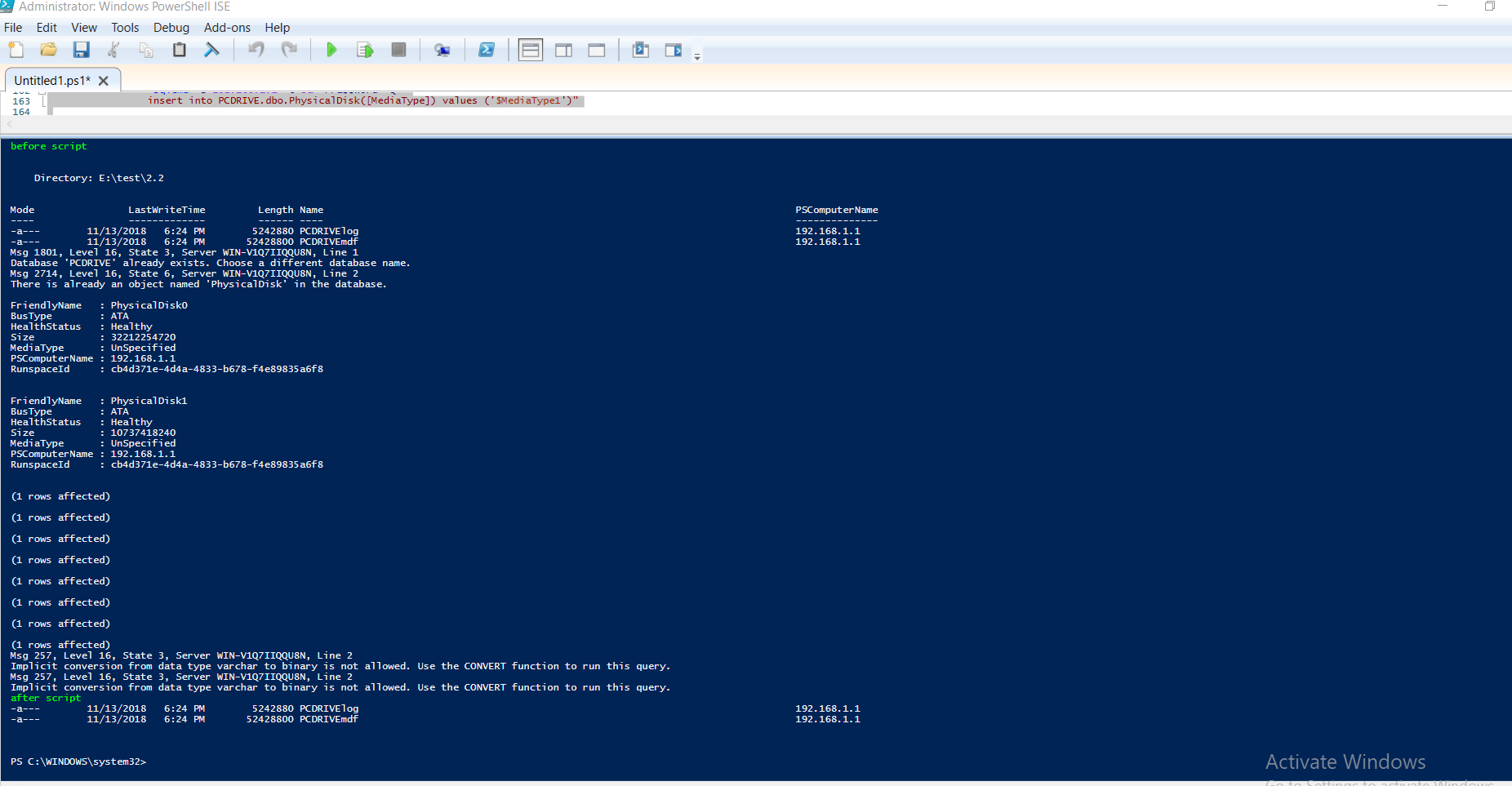
sqlcmd -S 192.168.1.1 -U sa -PPa$$w0rd -Q "

insert into PCDRIVE.dbo.PhysicalDisk([MediaType]) values ('$MediaType1')"

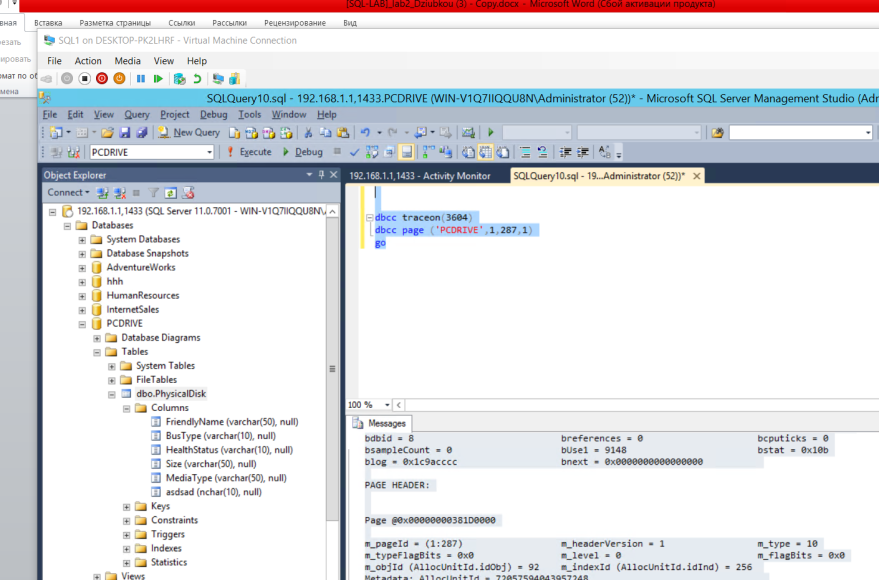
write-host "after script" -ForegroundColor Green

Invoke-Command -Credential administrator -ComputerName 192.168.1.1 -ScriptBlock{ Get-ChildItem E:\test\2.2}





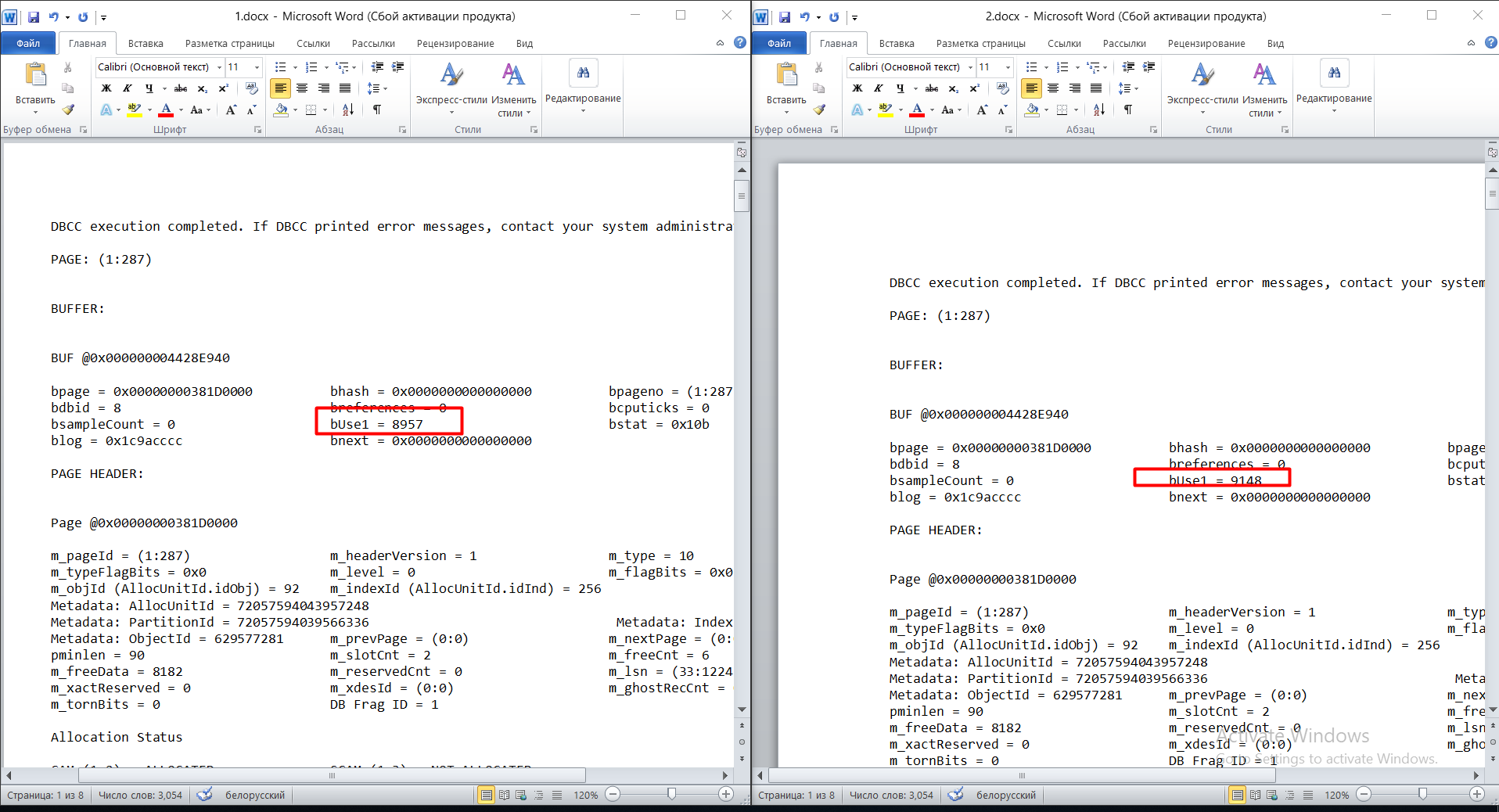
Сравниная параметр Pages , видим что внутри что-то меняется. Это происходит после внесения изменений



dbcc traceon(3604)

dbcc page ('PCDRIVE',1,287,1)

go



**TASK 2.3 ADDIT**

Attach database “AdventureWorks” from the provided source. Place files in the new created folder on Data disk. Answer the question if Database could be attached if you create a new database using provided files. Make screenshots of the process for report.

Создали БД – нашли ее файлики. Удалили бд . И пробуем приатачить эти файлы, которые мы скопировали перед удалением.

