```powershell
[CmdletBinding()]
param (
    [ValidateNotNullOrEmpty()]
    [string]$WVUMSharedUserGroup = "TestWVUMSharedUsers"
)

#Configure environment
Set-HSCEnvironment

$TodaysDate = Get-Date -format MM/dd/yyyy

$MM = $TodaysDate.Split("/,/")[0]
$dd = $TodaysDate.Split("/,/")[1]
$Date = $MM + $dd

$Added = @()
$users = @()
$UserInfo = @()
$UsersNotFound = @()

Connect-HSCOffice365MSOL

if ($Error.Count -gt 0)
{
    Invoke-HSCExitCommand -ErrorCount $Error.Count
}
<#
$UserInfo = @(
      [PSCustomObject]@{
        status = $user.status
        wvu_id = $user.wvuid
        uid = $user.uid
        FirstName = $user.firstname
        LastName = $user.lastname
        email = $user.email
        HSCEmail = $user.hscemail
  }
)
#>

try {
    $GroupObjectID = Get-MSOLGroup -SearchString $WVUMSharedUserGroup -ErrorAction Stop
    $WVUMUsersInSharedGroup = Get-MSOLGroupMember -GroupObjectId $GroupObjectID.ObjectId -All -ErrorAction Stop
}
catch {
    Write-Warning "Error searching for group/group members"
    Invoke-HSCExitCommand -ErrorAction Stop
}

try {
    $SQLPassword = Get-HSCSQLPassword -Verbose -ErrorAction Stop
    $SQLConnectionString = Get-HSCConnectionString -DataSource hscpowershell.database.windows.net -Database HSCPowerShell -Username HSCPowerShell -SQLPassword $SQLPassword
    Write-Output $SQLConnectionString

    $SQLQuery = 'select * from SharedUserTable'

```

```
    $users = Invoke-SqlCmd -Query $SQLQuery -ConnectionString $SQLConnectionString -ErrorAction Stop
}
catch {
    Write-Warning "SQL Error"
    Invoke-HSCExitCommand -ErrorCount $Error.Count
}


foreach ($user in $users)
{
    if ($user.HSCEmail -eq $false)
    {
        Try
        {
            $HSUserInfo = Get-ADUser $user.uid -Properties *

            if ($($HSUserInfo.extensionAttribute7) -eq "Yes365")
            {
                Write-Output "$($HSUserInfo.name) has exAtt7 set to Yes365"

            }
            else
            {
                $UIDUserObjectId = Get-MsolUser -searchstring $user.uid

                Write-Output "Checking $WVUMSharedUserGroup group share access for $($HSUserInfo.name)..."


                if ($UIDUserObjectId.ObjectId -in $WVUMUsersInSharedGroup.ObjectId)
            {
              Write-Output "$($HSUserInfo.name) already in WVUM Shared Users group"

            }
            else
            {

                    if (($UIDUserObjectId.ObjectId -ne $null) -AND ($($UIDUserObjectId.ObjectId).count -eq 1))
            {
                        Write-Output "Adding $($HSUserInfo.name)  to the WVUM Shared Users group..."

                        Add-MsolGroupMember -GroupObjectId $($GroupObjectID.ObjectId) -GroupMemberType User -GroupMemberObjectId $($UIDUserObjectId.ObjectId)

                        Write-Output "$($HSUserInfo.name) has been added to the security group"

                        $Added += $UserInfo
                    }

                    elseif (($UIDUserObjectId.ObjectId -ne $null) -AND ($($UIDUserObjectId.ObjectId).count -gt 1))
            {
                        Write-Output "Multiple GUID's found.  Attempting to find correct GUID for $($user.firstname) $($user.lastname)"

```

```
106                         $NewObjectID = Get-MsolUser -SearchString
"$($user.firstname) $($user.lastname)"
107
108                         if ($NewObjectId.ObjectId -in
$WVUMUsersInSharedGroup.ObjectId)
109                         {
110                             Write-Output "$($HSUserInfo.name) already in WVUM
Shared Users group"
111                         }
112                         else
113                         {
114                             Write-Output "Adding $($HSUserInfo.name)  to the WVUM
Shared Users group..."
115
116                             Add-MsolGroupMember -GroupObjectId
$($GroupObjectID.ObjectId) -GroupMemberType User -GroupMemberObjectId
$($NewObjectID.ObjectId)
117
118                             Write-Output "$($HSUserInfo.name) has been added to the
security group"
119
120                             $Added += $UserInfo
121                         }
122                     }
123
124                     else
125             {
126                         if (($HSUserInfo.DistinguishedName -like
'*OU=FromNewUsers,OU=DeletedAccounts,DC=HS,DC=wvu-ad,DC=wvu,DC=edu') -OR
($HSUserInfo.DistinguishedName -like '*OU=NewUsers,DC=HS,DC=wvu-ad,DC=wvu,DC=edu'))
127                         {
128
129                         }
130             }
131                 }
132             }
133         }
134         Catch
135         {
136             Try
137             {
138                 $WVUADSamAccountName = Get-ADUser $user.uid -Server wvu-ad.wvu.edu
-properties *
139
140                 if (($WVUADSamAccountName.department -like 'SOM Anesthesiology L4')
-OR ($WVUADSamAccountName.department -like 'SOM Eastern Division L4') -OR
($WVUADSamAccountName.Department -like 'SOM Family Medicine L4') -OR
($WVUADSamAccountName.Department -like 'SOM Ophthalmology L4'))
141                 {
142                     if ($WVUADSamAccountName.ObjectId -in
$WVUMUsersInSharedGroup.ObjectId)
143                 {
144                   Write-Output "$($WVUADSamAccountName.name) already in WVUM Shared
Users group"
145                 }
146                 else
147                 {
148
149                         $ObjectID = Get-MsolUser -SearchString
"$($WVUADSamAccountName.GivenName) $($WVUADSamAccountName.sn)"
```

```
150
151                         #Add-MsolGroupMember -GroupObjectId
$($GroupObjectID.ObjectId) -GroupMemberType User -GroupMemberObjectId
$($ObjectId.ObjectId)
152
153                         $Added += $UserInfo
154                     }
155                 }
156             }
157             Catch
158             {
159                 Write-Output "$($user.uid) not found"
160                 $UsersNotFound += $UserInfo
161             }
162         }
163     }
164 }
165
166 $Added | Export-Excel "C:\Users\microsoft\Documents\GitHub\HSC-PowerShell-
Repository\Update-SharedUserDistributionList\Logs\$(get-date -format yyyy-MM-dd)-
UsersAdded.xlsx" -Append
167 $UsersNotFound | Export-Excel "C:\Users\microsoft\Documents\GitHub\HSC-PowerShell-
Repository\Update-SharedUserDistributionList\Logs\$(get-date -format yyyy-MM-dd)-
UsersNotFound.xlsx" -Append
168
169
170
171 ############################
172 #Removing users from list
173 ############################
174 Write-Output ""
175 Write-Output ""
176 Write-Output ""
177 Write-Output "Removing Users"
178 Write-Output "=============="
179
180 $Removed = @()
181 $UsersToRemovePath =
"\\hs\public\tools\SharedUsersRpt\RemovedSharedUsers\RemovedSharedUsers$Date.xlsx"
182
183 if (Test-Path $UsersToRemovePath)
184 {
185     $UsersToRemove = Import-Excel -Path $UsersToRemovePath
186
187     foreach ($user in $UsersToRemove)
188     {
189         $RemoveUser = Get-MsolUser -SearchString $user.uid
190
191         if ($RemoveUser.ObjectId -in $WVUMUsersInSharedGroup.ObjectId)
192         {
193
194             Remove-MsolGroupMember -GroupObjectId $($GroupObjectID.ObjectId) -
GroupMemberType User -GroupMemberObjectId $($RemoveUser.ObjectId)
195
196             Write-Output "$($User.uid) was successfully removed"
197
198             $Removed += $UserInfo
199         }
200         else
201         {
```

```
            Write-Output "$($User.uid) is not in $WVUMSharedUserGroup"
        }
    }
}
else
{
    Write-Output "File to remove users does not exist. No users to remove today"
}

if ($Removed -ne "$null")
{
    $Removed | Export-Excel "C:\Users\microsoft\Documents\GitHub\HSC-PowerShell-Repository\Update-SharedUserDistributionList\$(get-date -format yyyy-MM-dd)-UsersRemoved.xlsx" -Append
}
##################
#end remove users
##################



Write-Output ""
Write-Output ""
Write-Output "Total users added: $($Added.count)"
Write-Output "Total users removed: $($Removed.count)"

Invoke-HSCExitCommand -ErrorCount $Error.Cout
```