

ARYASOFT BAKIM DESTEK SÜREÇLERİ DÖKÜMANI

Bakım Planları Level 1 – Kurumun İncelenmesi ve Monitoring Entegrasyonu

Anlaşma yapıldıktan sonra kurum sistemlerinin tanınması ve yapımıza dahil edilmnesi kapsamında bu süreçle birlikte çalışmalara başlanır.

Uzaktan bağlantı yapılacaksa VPN bilgileri alınır.

Bir danışman makinesi ve bu makinede administrator yetkili bir domain kullanıcısı istenir.

Makine Özellikleri:

- Windows 10 kurulum
- I5 işlemci ya da daha üstü
- En az 8 GB Ram
- En az 100 GB Storage (Mümkünse SSD)

Versiyon Kontrolleri

Kurumun kullandığı database versiyonu kontrol edilerek güncel sürümle arasındaki fark kontrol edilir.

```
-- Bu script ile SQL Server versiyonu kontrol edilir.
```

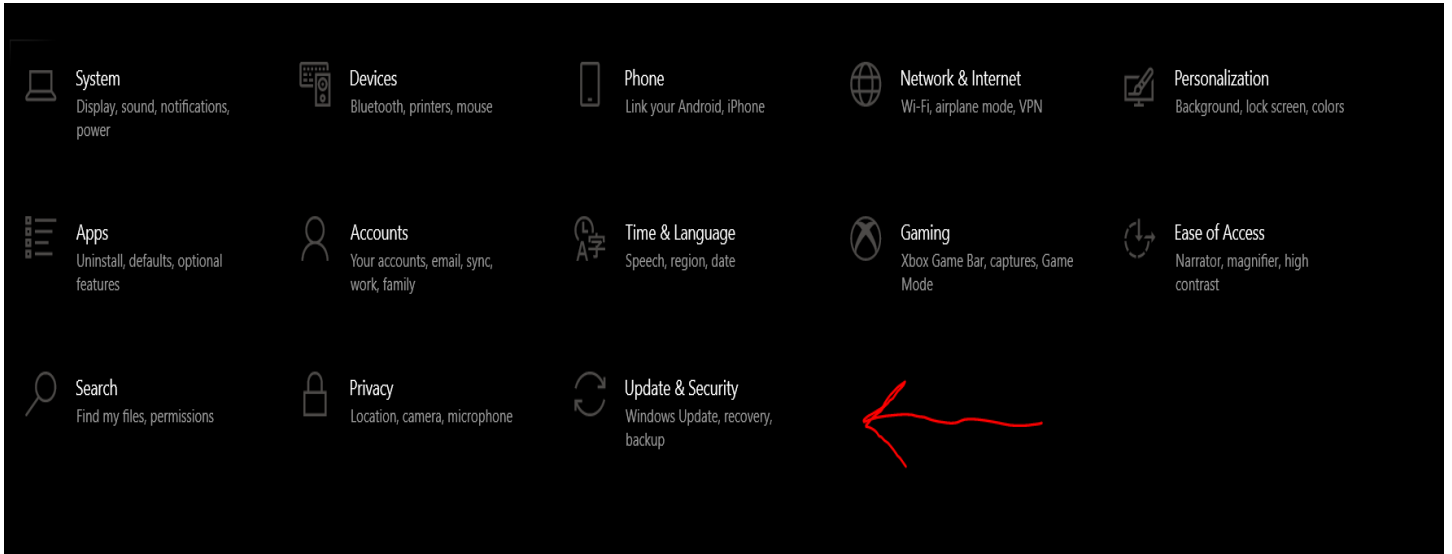
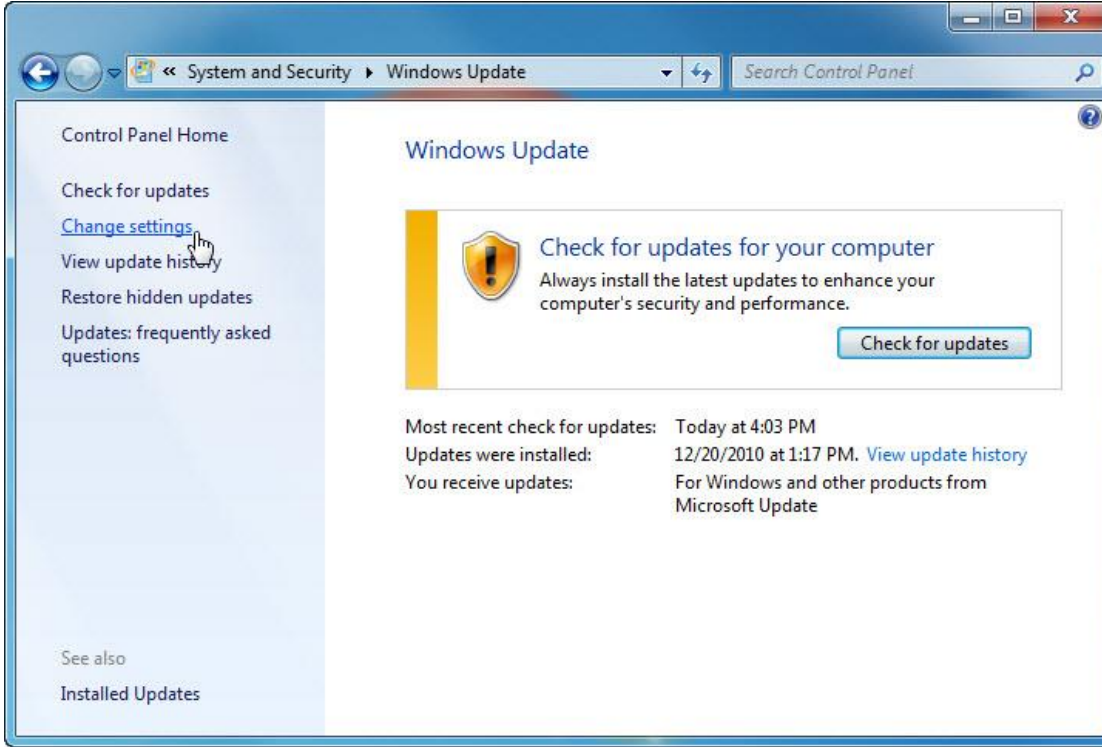
```
SELECT @@VERSION
```

```
-- Bu sitede en son güncel SQL Server versiyonunu görebilirsiniz.
```

```
https://sqlserverbuilds.blogspot.com
```

Update Kontrolleri

Sunucu ve veritabanı sistemi güvenlik updateleri geçilip geçilmediği bekleyen update var mı kontrolü yapılır. Denetim masası üzerinden update kontrolü yapılır. Eğer Windows update yapılmıyorsa kurumun kendi prosedürü olabilir. Bunun prosedür ile ilgili bilgi istenir.



Compatibility Level (Uyumluluk Seviyesi)

Databaselerin uyumluluk modelleri create edildiği sürüm versiyonundan başlamaktadır. DB taşınmalarında backup restore gibi ortam değişikliği ve versiyon yükseltme işlemlerinde güncel versiyon özelliklerini tam anlamıyla kullanabilmek için kontrol edilmesi ya da düzenlenmesi gereken ayardır.

-- Aşağıda script ile veritabanı uyumluluk versiyonlarını en güncel versiyona çeken scripti elde edebiliriz. **(LatestLevelCommand)**

-- Uyumluluk versiyonu yükseldikten sonra hata alıyorsak ve bu hatalar giderilemiyorsa, eski uyumluluk seviyesine geri dönmelidir. **(ComeBackCommand)**

```
select name, compatibility_level , version_name =
```

```
CASE compatibility_level
```

```
WHEN 80 THEN 'SQL Server 2000'
```

```
WHEN 90 THEN 'SQL Server 2005'
```

```
WHEN 100 THEN 'SQL Server 2008/R2'
```

```
WHEN 110 THEN 'SQL Server 2012'
```

```
WHEN 120 THEN 'SQL Server 2014'
```

```
WHEN 130 THEN 'SQL Server 2016'
```

```
WHEN 140 THEN 'SQL Server 2017'
```

```
WHEN 150 THEN 'SQL Server 2019'
```

```
ELSE 'new but not known yet - '+CONVERT(varchar(50),compatibility_level)
```

```
END,
```

```
'ALTER DATABASE [' + name + '] SET COMPATIBILITY_LEVEL = ' + cast(compatibility_level as  
varchar(3)) + ';' AS ComeBackCommand,
```

```
'ALTER DATABASE [' + name + '] SET COMPATIBILITY_LEVEL = ' + cast((SELECT  
MAX(compatibility_level) FROM sys.databases) as varchar(3)) + ';' AS LatestLevelCommand
```

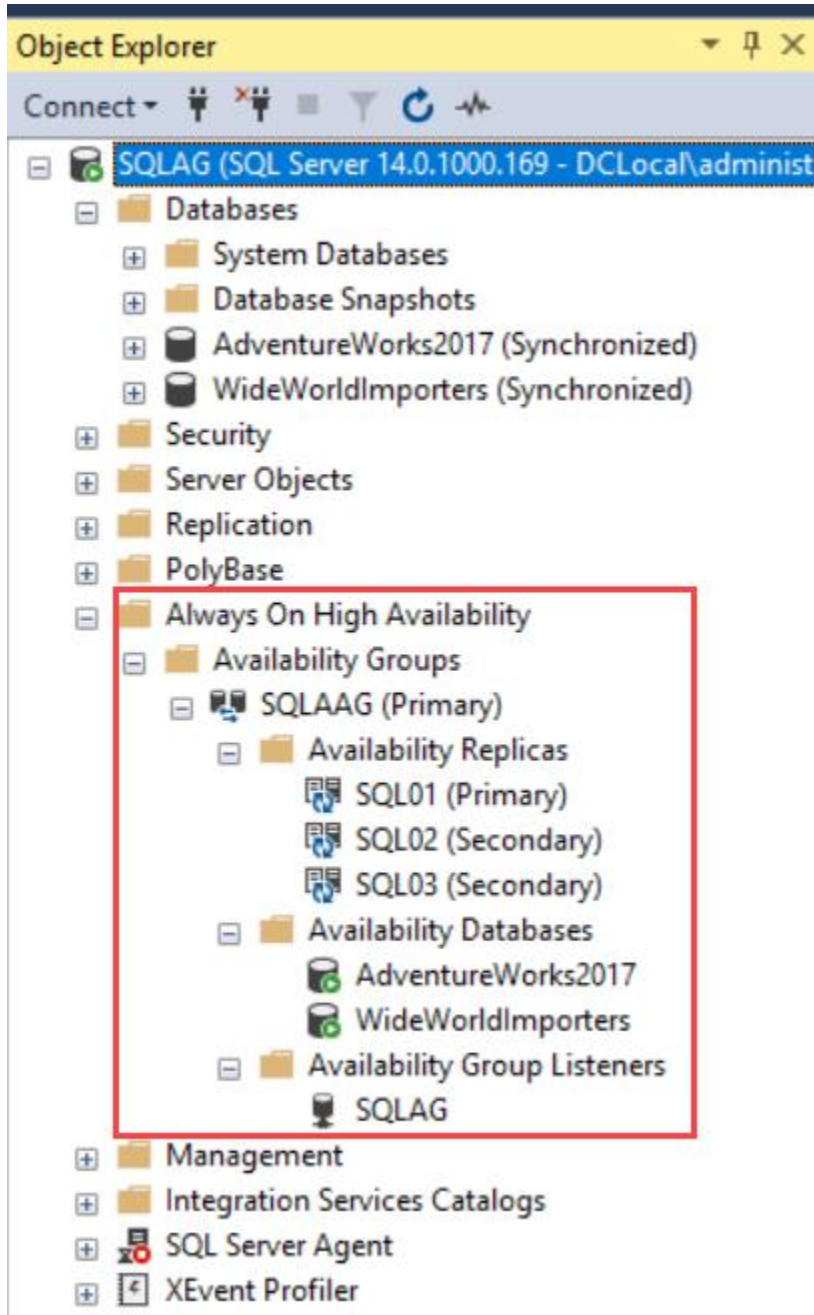
```
from sys.databases
```

High Availability Kontrolleri (Yüksek Erişilebilirlik)

Database sistem mimarisi anlamında yüksek erişilebilirlik var mı kontrolü yapılmalıdır. Eğer bu şekilde database sistemi dizayn edilmişse yapının sağlıklı olarak çalışıp çalışmadığı denetlenir. Primary yazan SQL server AlwaysOn grubu üzerine right click yapıp Show dashboard'a tıklarsak, AlwaysOn durumunu da sorgulamış oluruz.

-- AlwaysOn Object Explorer üzerindeki görüntüsü.

-- AlwaysOn Dashboard görünümü. (Sağlıklı **(Healthy)** ibaresinden başka bir uyarı varsa supervisor danışmanlığına başvurunuz)



SQLAAG: hosted by SQL01 (Replica role: Primary)

Last updated: 12/23/2019 11:58:41 AM
Auto refresh: on

Availability group state: ✔ Healthy
Primary instance: SQL01
Failover mode: Manual
Cluster state: SQL_AAG (Normal Quorum)
Cluster type: Windows Server Failover Cluster

[Start Failover Wizard](#)
[View Always On Health Events](#)
[View Cluster Quorum Information](#)
[Collect Latency Data](#)
[Add/Remove Columns](#)

Availability replica:

Name	Role	Availability Mode	Failover Mode	Seeding Mode	Synchronization State	Issues	Member State
SQL01	Primary	Synchronous co...	Manual	Automatic	Synchronized		Online
SQL02	Secon...	Synchronous co...	Manual	Automatic	Synchronized		Online
SQL03	Secon...	Asynchronous co...	Manual	Automatic	Synchronizing		Online

Group by

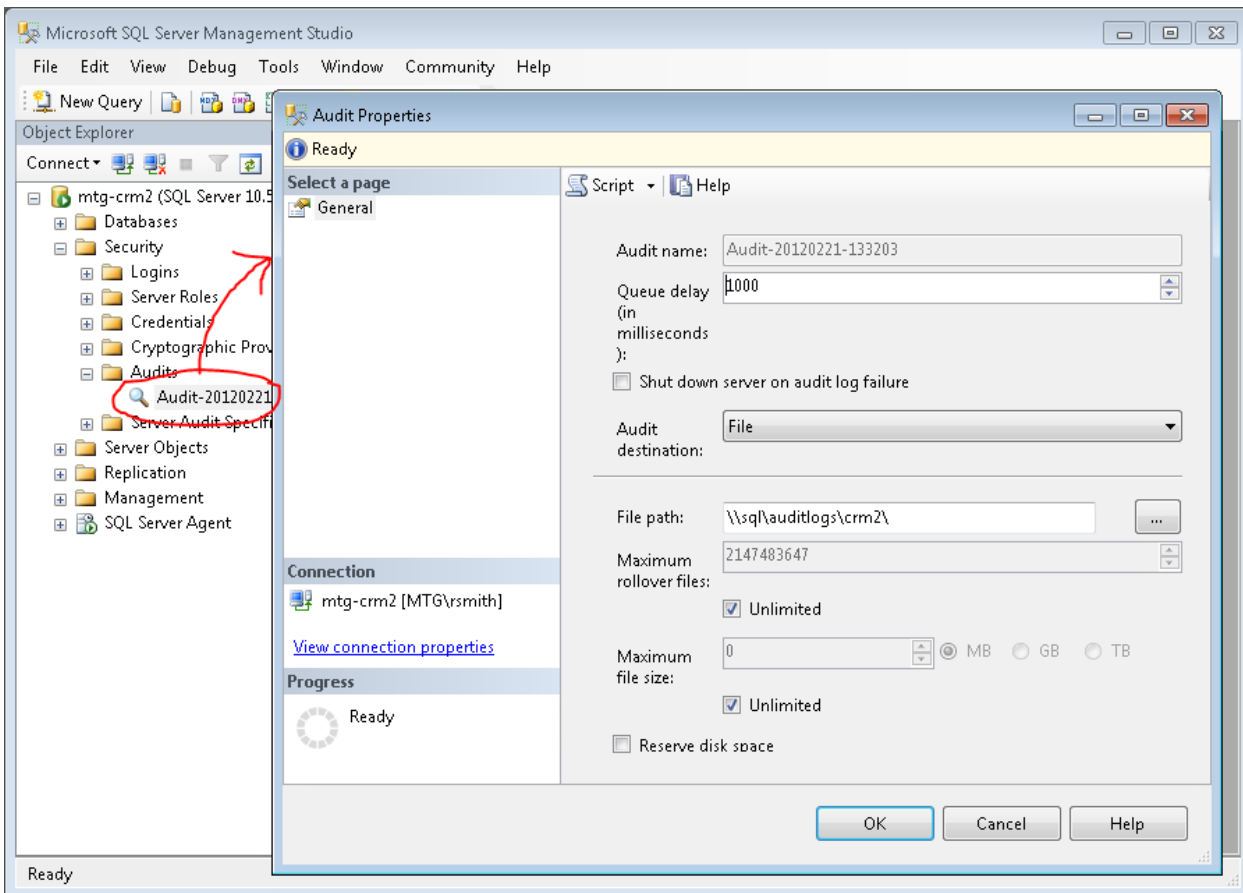
Name	Replica	Synchronization State	Failover Read...	Issues	Synchronizatio...
SQL01					
AdventureWorks2017	SQL01	Synchronized	No Data Loss		0
WideWorldImporters	SQL01	Synchronized	No Data Loss		0
SQL02					
AdventureWorks2017	SQL02	Synchronized	No Data Loss		0
WideWorldImporters	SQL02	Synchronized	No Data Loss		0
SQL03					
AdventureWorks2017	SQL03	Synchronizing	Data Loss		0
WideWorldImporters	SQL03	Synchronizing	Data Loss		0

[Add/Remove Columns](#)

Audit Kontrol ve Konfigürasyonları

Kurumun var olan yapısında Audit logları tutuluyor mu incelenir. Tutuluyorsa önerilecek audit tipleri analizi yapılarak öneri olarak sunulur. Tutulmuyorsa bunun için yapılacak çalışmalar ve önemi hakkında firmaya bilgi verilir.

-- Object Explorer üzerinde Audit kontrol ekranı



-- Audit üzerinde nelerin kayıt altına alınacağını belirleyen ekran.

Server Audit Specification Properties

Ready

Select a page

General

Script Help

Name: ServerAuditSpecification-20161005-e

Audit: Audit-20161005-f-security-log

Actions:

	Audit Action Type	Object Class	Object Schema	Object Name	Principal Name
1	DATABASE_OBJECT_ACCESS_GROUP				
2	SCHEMA_OBJECT_ACCESS_GROUP				
3	DATABASE_OBJECT_CHANGE_GROUP				
4	SCHEMA_OBJECT_CHANGE_GROUP				
5	SERVER_OBJECT_CHANGE_GROUP				
6	DATABASE_OPERATION_GROUP				
7	SERVER_OPERATION_GROUP				
8					

Connection

View connection properties

Progress

Ready

OK Cancel Help

SqlAdminPremium Kurulum ve Konfigürasyonları

DB Monitoring yapısı için geliştirilmiş olan SqlAdminPremium database i oluşturulmalıdır. SQLAdminPremium veritabanının kurulumundan sonra, bu prosedürler kapsamında bize ticket açması sağlanır. Mail vb konfigürasyonları yoksa bu konfigürasyonu, müşteriden aldığımız bilgiler doğrultusunda kendimiz kurarız.

1. SQLAdminPremium'u kurulumunu gerçekleştirecek olan dacpac dosyasını indiririz.
2. Bu dosyayı internet sitesi linkinden ya da SQLAdminPremium github projesi üzerinden indirebiliriz. (Sitemizde ki Link eklenecek)
3. Kurulumu yapacağımız sql server instance üzerinde SQLAdminPremium isimli bir veritabanı oluşturuyoruz.
4. Bu veritabanı üzerine right click yapıp Tasks -> Upgrade Data-tier Application seçeneğine tıklıyoruz.
5. Açılan yeni sayfada Next dedikten sonra gelen sekmede dacpac dosyasını Browse diyerek seçiyoruz ve Next diyoruz.
6. Detect Changes sayfasını tamamlandıktan sonra 'Proceed despite possible loss of changes' seçeneğini tıklayıp devam ediyoruz.
7. Options kısmında 'Rollback on Faliure' seçeneğine tik atarak next diyoruz.
8. Upgrade plan gözden geçirildikten sonra Next, Next, Next diyip kurulumu tamamlıyoruz.
9. Aşağıdaki kodları SQL server query'si olarak çalıştırıp kurulumu tamamlıyoruz.

```
--Önce ilk yoruma kadar olan kısmı çalıştır.
exec [dbo].[sp_CreateDatabaseMail]
@ProfileName = 'SQLAdminMailProfile',
@SmtpServerOutgoingPort =25 ,
@SmtpServer = 'relay.test.com.tr',
@SenderMailAddress = 'sqlnotification@test.com.tr',
@SenderDisplayName = 'SQLSPWEB',
@SenderUsername = 'sqlnotification@test.com.tr',
@SenderPassword = 'password',
@SmtpEnableSsl =0,
@SysmailAccountName = 'mailAddress',
@OverwriteDeployConfig=1 --secondaryde 0 yapmak lazım.

--jobları yaratmak için iki kere çalıştır alttaki komutu neden bakmak lazım :)
EXECUTE [dbo].[sp_CreateJobs] @CreateJobs='Y'

--alertleri yaratmak için
exec sp_CreateAlert
```

Yetki ve Erişim Kontrolleri

- DB Owner

Databaseleri oluşturan kullanıcılar değiştirilerek yetki zafiyetlerinin önüne geçilmesini kapsamaktadır. Ayrıca ilgili user'ın silinmesi durumlarında da sistemde oluşacak hataların önüne geçilmesini kapsamaktadır.

- Job Owner

Jobları oluşturan kullanıcılar değiştirilerek yetki zafiyetlerinin önüne geçilmesini kapsamaktadır. Ayrıca ilgili user'ın silinmesi durumlarında da sistemde oluşacak hataların önüne geçilmesini kapsamaktadır.

- Service Owner

Servisleri yürüten kullanıcılar değiştirilerek yetki zafiyetlerinin önüne geçilmesini kapsamaktadır. Ayrıca ilgili user'ın silinmesi durumlarında da sistemde oluşacak hataların önüne geçilmesini kapsamaktadır.

- Sysadmin

Sistemde sorun oluşması ve aşama aşama sistemin tek bir noktadan yönetilmesi anlamında yetkisiz kişilerin data'lara erişmesinin önlenmesi kapsamında dışarıya veri sızdırılmasını engellemek amacıyla incelenmesi gerekmektedir.

```
-- Burada ilk önemli olan sysadmin yetkilerinin kimlerde olduğudur. Sysadmin yetkisi bulunan user'ların çıkarılıp bunların gerçekten sysadmin yetkisine ihtiyacı olup olmadığı tespit edilip, sysadmin yetkisine ihtiyacı olmayan kullanıcıların yetkileri alınmalıdır
```

```
--Sysadmin yetkisi bulunan kullanıcıların bulunmasını sağlayan script
```

```
select loginname, dbname, sysadmin from sys.syslogins where sysadmin = 1 and loginname not in ('sa', 'BUILTIN\Administrators', 'NT SERVICE\SQLWriter', 'NT SERVICE\Winmgmt', 'NT SERVICE\MSSQLSERVER', 'NT SERVICE\SQLSERVERAGENT')
```

```
-- Sysadmin yetkisini almanı sağlayan script
```

```
USE [master]  
ALTER SERVER ROLE [sysadmin] DROP MEMBER [username];
```


FileGroup Free Space

Veritabanı dosyalarının doluluk oranları incelenir. Aşağıdaki scriptte data'ların bulunduğu dosyaların doluluk oranları bulunabilir. SQL Server üzerindeki bir data dosyası için uygulanan ideal doluluk oranı %80'den fazla olmamalıdır. Yani aşağıdaki script çalıştırıldığında FreeSpaceRatio kısmında %20 altında görülen her değer ideale uygun olmayıp artırılmalıdır. Bu artırım yapılırken disk alanında boşluk önemlidir. Disk alanını aşırı dolduracak (Yine burada da %20 altına inmemeli) bir artırımdan kaçınılmalıdır.

```
-----You can use this script for showing CurrentSize FreeSpace and
FreeSpaceRatio-----

CREATE TABLE #FileSize
(dbName NVARCHAR(128),
  FileName NVARCHAR(128),
  Physcial_Name NVARCHAR(600),
  type_desc NVARCHAR(128),
  CurrentSizeMB DECIMAL(10,2),
  FreeSpaceMB DECIMAL(10,2),
  FreeSpaceRatio DECIMAL(10,2)
);

INSERT INTO #FileSize(dbName, FileName, Physcial_Name, type_desc, CurrentSizeMB,
FreeSpaceMB, FreeSpaceRatio)
exec sp_msforeachdb
'use [?];
  SELECT DB_NAME() AS DbName,
         name AS FileName,
         physical_name,
         type_desc,
         size/128.0 AS CurrentSizeMB,
         size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0 AS
FreeSpaceMB,
         (((size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0) *
100) / (size/128.0)) AS FreeSpaceRatio
FROM sys.database_files
WHERE growth <> 0 and type IN (0,1);';

SELECT *
FROM #FileSize
WHERE dbName NOT IN ('distribution') and type_desc = 'ROWS' and FreeSpaceRatio < 20
order by 1;
```

Log Free Space

Log dosyalarının backup işleminden sonra bırakılarak ilgili alanın şişerek DB makinesinde yer kaplamaması adına inceleme yapılması gerekmektedir. Yine benzer şekilde Log dosyalarının içindeki boşluk %60'dan fazla olmalıdır. Bu değerden fazla olan log dosyalarında kontrol sağlanarak sorunun log backuplardan olmadığı görülürse log dosyası büyütülür.

```
-----You can use this script for showing CurrentSize FreeSpace and
FreeSpaceRatio-----

CREATE TABLE #FileSize
(dbName NVARCHAR(128),
 FileName NVARCHAR(128),
 Physcial_Name NVARCHAR(600),
 type_desc NVARCHAR(128),
 CurrentSizeMB DECIMAL(10,2),
 FreeSpaceMB DECIMAL(10,2),
 FreeSpaceRatio DECIMAL(10,2)
);

INSERT INTO #FileSize(dbName, FileName, Physcial_Name, type_desc, CurrentSizeMB,
FreeSpaceMB, FreeSpaceRatio)
exec sp_msforeachdb
'use [?];
SELECT DB_NAME() AS DbName,
       name AS FileName,
       physical_name,
       type_desc,
       size/128.0 AS CurrentSizeMB,
       size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0 AS
FreeSpaceMB,
       (((size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0) *
100) / (size/128.0)) AS FreeSpaceRatio
FROM sys.database_files
WHERE growth<>0 and type IN (0,1);';

SELECT *
FROM #FileSize
WHERE dbName NOT IN ('distribution') and type_desc = 'LOG' and FreeSpaceRatio < 20
order by 1;
```

Logfile

Vlf count ve varolan alanın büyümeleri kontrol edilerek db hizmetinin durmaması adına metrikler çıkarılır.

```
-- Bu script ile veritabanının VLF sayısını tespit edebiliriz.
-- VLF sayısı 50'den fazla olmamalıdır.

Use [database_name]
DBCC LOGINFO()
```

Maintenance Planlar Kuurlumu (Bakım Planları)

Kuruma ait bakım ve operasyonel anlamda oluşturulmuş olan Maintenance planlar var olup olmadığı kontrol edilir. Eğer yoksa Ola Hallengren'in oluşturduğu MaintainceSolution.sql dosyası ile bakım joblarının kurulumu gerçekleştirilir. Bu dosyaya Ola Hallengren'in internet sitesi, Github sayfası ya da Aryasoft github sayfasın üzerinde "SQLAdmin-Scripts" repository'si üzerinden _OnProgress dosyası içinden ulaşılabilir.

<https://ola.hallengren.com>

<https://github.com/olahallengren/sql-server-maintenance-solution>

https://github.com/aryasoft/SQL-Admin-Scripts/blob/development/_OnProgress/05_OLAMaintenanceSolution_2022-01-02_13_58_13.sql

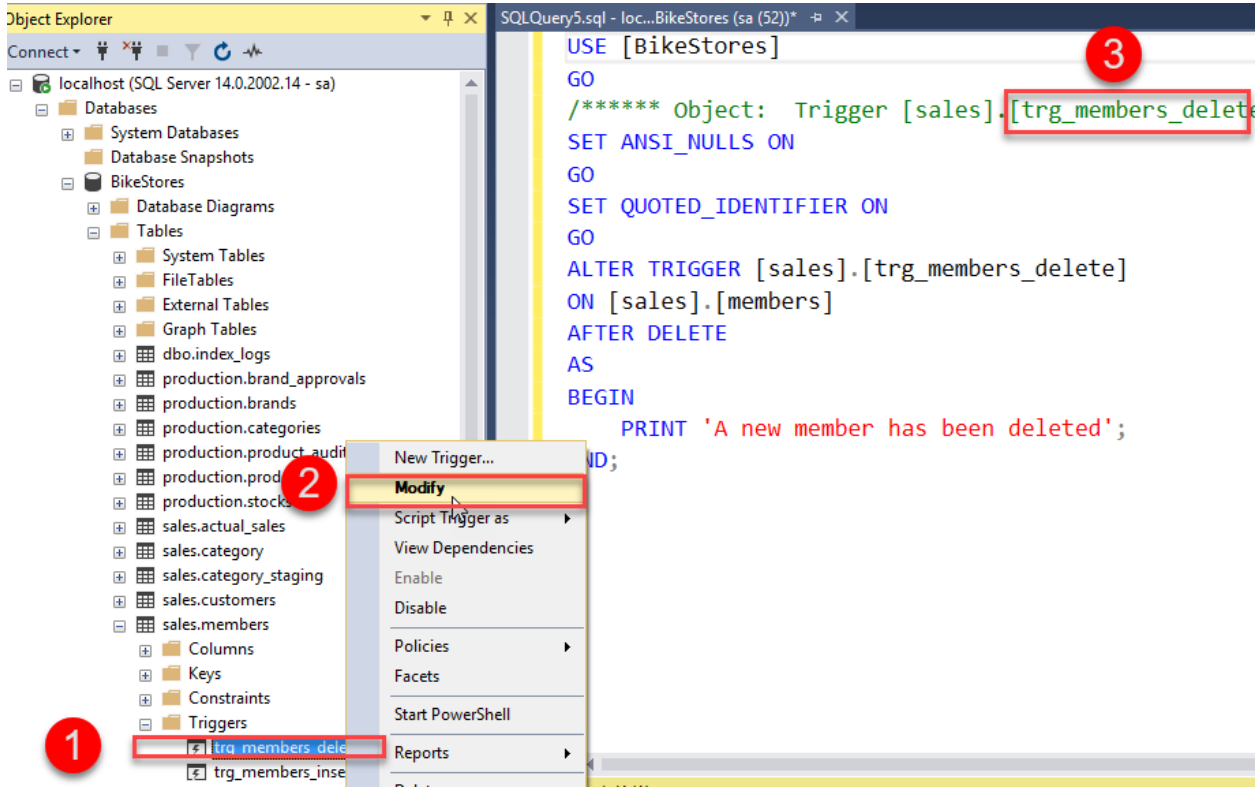
Halihazırda Bulunan Maintenance Planlar Yönetimi (Bakım Planları)

Kurumun daha önceden hazırladığı bakım planları varsa incelemeleri yapılır. Bu bakım planlarının çalışma durumu ve yaptığı işlemler not edilip hatalı olanlar iptal edilip kendi bakım planlarımızın kurulumu yapılır. Her ne kadar SQL Server kendi bakım planlarını yapabilse de bu bakım planlarından istediğimiz performansı alamıyoruz.

- + Security
- + Server Objects
- + Replication
- + PolyBase
- + Always On High Availability
- Management
 - + Policy Management
 - + Data Collection
 - + Resource Governor
 - + Extended Events
 - Maintenance Plans
 - MaintenancePlan_Aryasoft_Musa_Every15Minutes_LogBackups
 - MaintenancePlan_Aryasoft_Musa_Overnight
 - MaintenancePlanForTest
 - + SQL Server Logs
 - + Database Mail
 - + Distributed Transaction Coordinator
- + Integration Services Catalogs
- + SQL Server Agent
- + XEvent Profiler

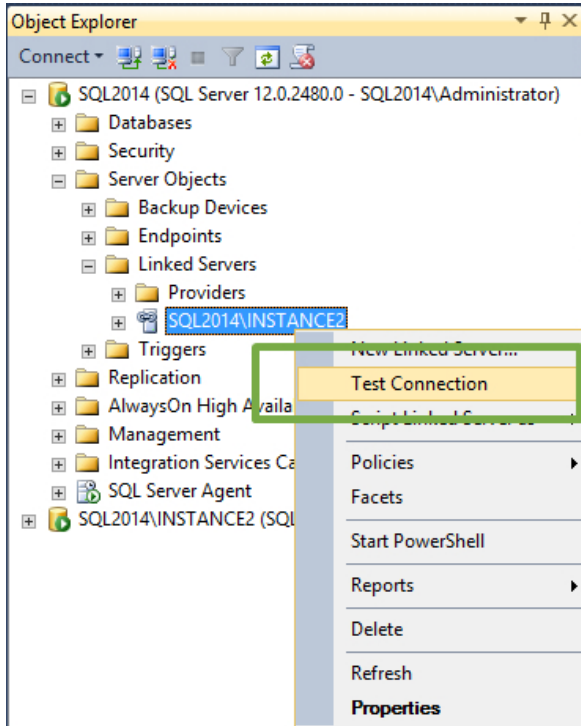
Database Trigger

Database Trigger konfigürasyonu yapılmış mı? Yapıldıysa neden yapılmış ve bunun kaldırılması anlamında alınacak aksiyonlar nelerdir? Trigger'ın yaptığı işlemlerin lock ve deadlockla neden olması sistem kaynaklarını gelen işleme göre yönetim dışı kullanması nedeniyle database hizmetlerine yük getirmektedir.



Linked Server

Kurumun diğer database ortamlarıyla olan haberleşmesini database katmanında yapıp yapmadığı kontrolü yapılır. Bu databasede verilen yetkilerin dışında ardaki linked serverla yetkisiz kişilerin kazanımlar elde etmesinin önüne geçmesi engellenir.



Backup Kontrolleri

Kurumun sahip olduğu database sistemlerinin güncel backupları tutuluyor mu? En son backuplama işlemi ne zaman yapılmış ve bu süreç nasıl yönetilmektedir. İncelemeler yapılarak olası tehlikelere karşı kurumda veri kaybını minimuma indirmek amacıyla kontroller yapılır.

```
-- Backup Backuplar kontrol edilir. Kontrol Sorgusu

SELECT
    s.database_name,
    m.physical_device_name,
    CAST(DATEDIFF(second, s.backup_start_date,
    s.backup_finish_date) AS VARCHAR(100)) + ' ' + 'Seconds' TimeTaken,
    s.backup_start_date,
    s.backup_size/1024/1024 BS_MB,
    s.compressed_backup_size/1024/1024 CBS_MB,
    CASE s.[type]
    WHEN 'D' THEN 'Full'
    WHEN 'I' THEN 'Differential'
    WHEN 'L' THEN 'Transaction Log'
    END AS BackupType,
    s.server_name,
    s.recovery_model
FROM msdb.dbo.backupset s
INNER JOIN msdb.dbo.backupmediafamily m ON s.media_set_id = m.media_set_id
where s.backup_start_date > DATEADD(dd, -7, GETDATE()) and (s.type='D' or
s.type='L')
ORDER BY backup_start_date DESC, backup_finish_date
GO
```

Restore Planları Kontrolleri

Alınan backupların belirli süre aralıklarla test amaçlı açılması işlemine yönelik kurumun çalışmaları var mı kontrolü işlemini kapsamaktadır. Bir restore örneği aşağıda verilmiştir.

```
/* restore to the latest point in time with full and transaction log backups
*/

-- Full Backup restore edilmesi ve üstüne log backup restore örnek script
RESTORE DATABASE [MyDatabaseTest]
FROM DISK = N'D:\Backups\MyDatabase_backup_2021_06_01_000011_4175616.bak'
WITH MOVE N'MyDatabase' TO N'D:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\MyDatabaseTest.mdf',
      MOVE N'MyDatabase_log' TO N'D:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\MyDatabaseTest_log.ldf',
NORECOVERY, -- 'restore with norecovery' allows additional backups to be
applied - database will be unavailable
REPLACE;

-- restore transaction log backups
RESTORE LOG [MyDatabaseTest] FROM DISK =
N'D:\Backups\MyDatabase_backup_2021_06_01_001502_1580793.trn' WITH
NORECOVERY;
RESTORE LOG [MyDatabaseTest] FROM DISK =
N'D:\Backups\MyDatabase_backup_2021_06_01_011501_4325128.trn' WITH
NORECOVERY;
```

Server Konfigürasyonları Kontrolü

Server konfigürasyonlarında best practicelara uygun olmayan ayarlar bulunmakta mı eğer bu tür ayarlar varsa işlemler niçin yapılmış kontrolleri yapılır. Best practice'lere göre düzenlemeler yapmak kurum kazanımlarını olumlu yönde etkileyecek mi analizi yapılır. Best practice'ler ortamdan ortama değişiklik gösterebilmektedir. Bu konuda bir değişiklik yapmadan önce kesinlikle ve kesinlikle supervisor'ınızdan destek alınız.

```
-- SQL Server konfigürasyon ayarlarının görüntülenmesi

USE master;
GO

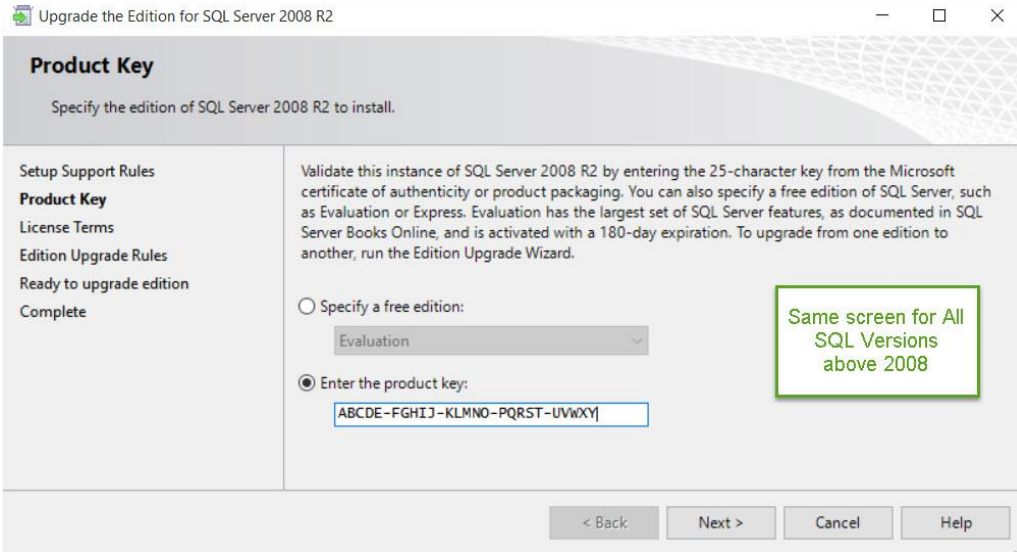
EXEC sp_configure 'show advanced option', '1'; --Enable
advanced options
RECONFIGURE;

EXEC sp_configure --Show all the options

EXEC sp_configure 'show advanced option', '0'; --Always
disable advanced options
RECONFIGURE;
```

Lisans Kontrolü

Kurumun var olan kurulumları sahip olduğu lisans gereksinimlerini karşılamakta mıdır kontrolleri yapılır.



Bakım Planları Level 2 – Monitoring Uyarılarının İncelenmesi ve Kurumdaki Yapıların İyileştirilmesi ve Geliştirilmesi

Kurum bizim süreçlerimize entegre olduktan sonra ölçeklendirilmesi, kurum sistemlerinin sağlıklı bir şekilde hizmet verebilmesi ve yapının geliştirilebilmesi adına bu fazda aksiyonları alınır.

Kullanılmayan İndex ve Objelerin Tespiti

Sistemde daha önceden oluşturulmuş ancak maliyet anlamında performans vermeyen ve kullanılmayan index ve objeler tespit edilerek kuruma bilgi verilmesi objelerin disable edilmesi ve sonrasında kaldırılarak hem alan hemde performans kazanımı sağlanması süreçleridir.

```
--Kullanılmayan indeksleri bulmak için çalıştırılacak query

USE [database_name]
GO SELECT TOP 10 DB_NAME(id.database_id) AS databaseName, id.statement AS TableName,
id.equality_columns, id.inequality_columns, id.included_columns, gs.last_user_seek,
gs.user_seeks, gs.last_user_scan, gs.user_scans, gs.avg_total_user_cost * gs.avg_user_impact *
(gs.user_seeks + gs.user_scans) AS ImprovementValue FROM sys.dm_db_missing_index_group_stats gs
INNER JOIN sys.dm_db_missing_index_groups ig ON gs.group_handle = ig.index_group_handle INNER
JOIN sys.dm_db_missing_index_details id ON id.index_handle = ig.index_handle ORDER BY
avg_total_user_cost * avg_user_impact * (user_seeks + user_scans) DESC GO
--Sorgulara göre SQL tarafyndan yapılmayı öngörülen indeksler
SELECT TOP 10 DB_NAME(id.database_id) as databaseName,
id.statement as TableName,
id.equality_columns,
id.inequality_columns,
id.included_columns,
gs.last_user_seek,
gs.user_seeks,
gs.last_user_scan,
gs.user_scans,
gs.avg_total_user_cost * gs.avg_user_impact * (gs.user_seeks + gs.user_scans) as
ImprovementValue
FROM sys.dm_db_missing_index_group_stats gs
INNER JOIN sys.dm_db_missing_index_groups ig ON gs.group_handle = ig.index_group_handle
INNER JOIN sys.dm_db_missing_index_details id ON id.index_handle = ig.index_handle
ORDER BY avg_total_user_cost * avg_user_impact * (user_seeks + user_scans) DESC
GO
```

Ola Hallengren IndexOptimize - USER_DATABASES Job ayarları

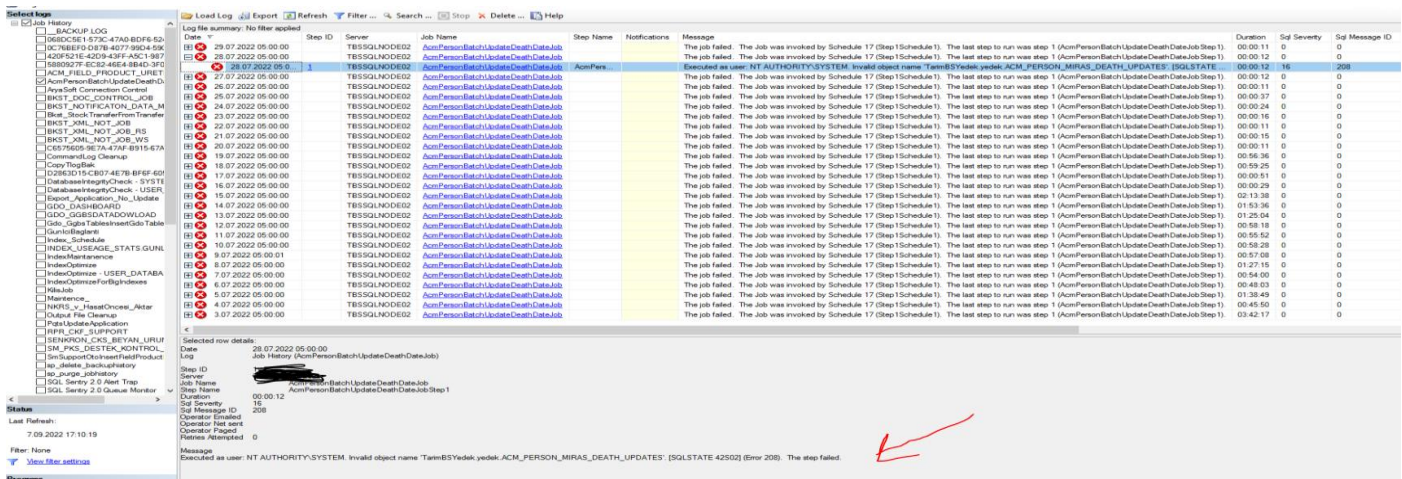
```
EXECUTE dbo.IndexOptimize
@Databases = 'ALL_DATABASES',
@Indexes= 'ALL_INDEXES',
@FragmentationLow = NULL,
@FragmentationMedium =
'INDEX_REORGANIZE,INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE',
@FragmentationHigh = 'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE',
@FragmentationLevel1 = 5,
@FragmentationLevel2 = 30,
@PartitionLevel = 'Y',
@LogToTable = 'Y',
@Delay = 1,
@FillFactor = 90,
@MaxDOP = 0,
@SortInTempdb = 'Y',
@LockTimeout = 300
```

Kullanılmayan Bakım Planları ve Jobların Tespiti

Kurumda daha önceden oluşturulmuş bakım planları ve joblar varsa bunların kontrollü olarak kaldırılması süreci yönetilir. Bu jobları SQLAgent altında Job Activity Monitor üzerinden tespiti sağlanabilir. Bu konuda supervisor'ınızdan ve ayrıca çalışmakta olduğunuz müşteri ile iletişimde olun kaldırılmaması gereken bir job'ı kaldırmak istemezsiniz.

Hata Loglarının Kontrolü

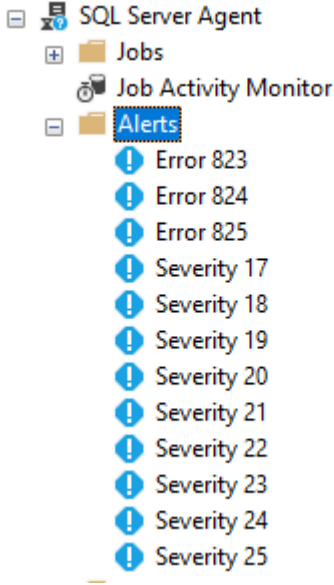
Bir önceki incelemeden itibaren sistemde oluşan hatalar incelenerek analiz edilir. Hataların devam etmesi durumunda aksiyon planları oluşturularak uygulanır ve sorun çözülür. Bu jobları SQLAgent altında Job Activity Monitor üzerinden tespiti sağlanabilir. Burada hata veren job'ın üstüne right click yapılarak View History seçilir ve error log okunarak kontrol sağlanır.



The screenshot displays the SQL Server Job Activity Monitor interface. The main window shows a list of jobs, including 'AcumPersonBatchUpdateDeathDataJob'. The 'View History' button is clicked, opening a window showing the job's execution history. The history shows that the job failed on 28.07.2022 05:00:00 with the error message: 'Executed as user: NT AUTHORITY\SYSTEM. Invalid object name 'TemsB5Yedek.yedek.ACM_PERSON_MIRAS_DEATH_UPDATES'. [SQLSTATE 42502] (Error 208). The step failed.'

SQL Server Alert Kontrolü

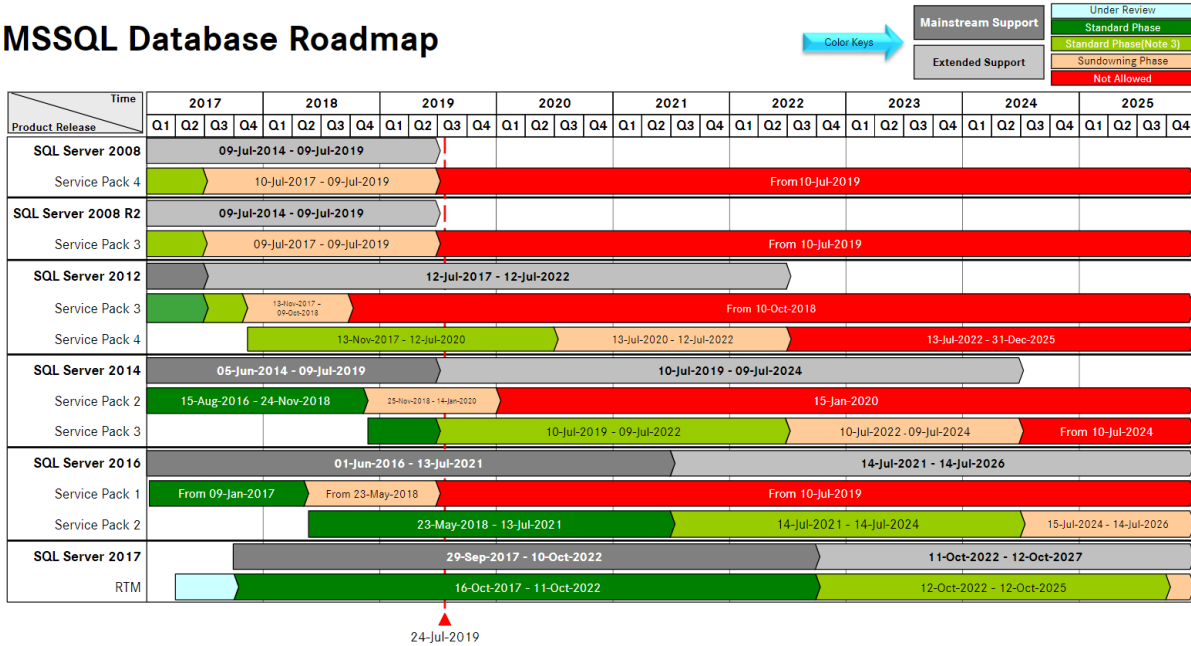
SQL Server üzerinde bulunan alert'ların incelemesi yapılır konfigürasyon ayarları yapıldı mı kontrol edilir.



Yol Haritası Oluşturulması

Varolan işletim sistemi ve database versiyonları upgrade ve update işlemlerinin DB, işletim sistemi bazında end of life (Artık güncelleme gelmeyeceği) olacağı zaman takvimlerinin çıkartılması gereklidir. Bu takvim çıkarıldığında ürünler end of life olmadan yeni versiyonlara yükseltilmelidir. Bu durumun takibinin yapılması sürecin çıkartılması ve işlemler için hem bizim kurumumuzda hemde ilgili kurumda bu takvimin paylaşılması işlemidir. İlgili takvim kurum bizimle süreci yürütmek istediğinde detaylandırılabilir.

MSSQL Database Roadmap



Versiyon Update ve Upgradelerinin Planlanması ve Kurumun Bilgilendirilmesi

Versiyonların incelenerek sektörün takip edilmesi ve ürün yaşam döngüsünün sağlıklı bir şekilde yönetilmesi gerekmektedir. Roadmap çıkartıldığı durumda prosedürel olarak kuruma hafıza kazandırmak adına bu işlemlerin sürecinin yürütülmesi ve kurumun haberdar olması sağlanır. Artıları eksileri detaylandırılmalıdır.

Monitoring Sisteminden Gelen Hataların Giderilmesi

Monitoring database devreye alındıktan sonra gelen uyarıların incelenerek en kısa sürede çözülmesi çözüm üretilmiyorsa daha üst seviyeye iletilmesi sürecidir. Bu hataların çözümlerini oluşturduğumuz destek platformu üzerinden takip edip hata giderildikten sonra ise rapor hazırlayarak sonucu yazabiliriz.

Aylık Yapılan Kontrol Operasyonlarının Takibi

Sistem sürekliliği anlamında yapılan sistemdeki bozulmaların önüne geçmek ve taramaların sistemin sağlıklı devam edebilmesine yönelik yapılması gereklidir. Tüm sistemin düzenli olarak denetlenmesi ve monitoring uyarılarından bağımsız izlenmesi sistemlerin sağlıklı yaşam döngüsünü sürmesi adına önemlidir.

Çalışma listesi bir sonraki sayfadadır...

Çalışma Kontrol Listesi Başlangıç

Backup (Yedekleme)

Backuplar kontrol edilecek. Full, Diff ve Log backuplar firma backup politikası dahilinde alınıyor mu kontrol edilecek.

-- Backup Kontrol Sorgusu

SELECT

```
s.database_name,
m.physical_device_name,
CAST(DATEDIFF(second, s.backup_start_date,
s.backup_finish_date) AS VARCHAR(100)) + ' ' + 'Seconds' TimeTaken,
s.backup_start_date,
    s.backup_size/1024/1024 BS_MB,
    s.compressed_backup_size/1024/1024 CBS_MB,
CASE s.[type]
WHEN 'D' THEN 'Full'
WHEN 'I' THEN 'Differential'
WHEN 'L' THEN 'Transaction Log'
END AS BackupType,
s.server_name,
s.recovery_model
FROM msdb.dbo.backupset s
INNER JOIN msdb.dbo.backupmediafamily m ON s.media_set_id = m.media_set_id
where s.backup_start_date > DATEADD(dd, -7, GETDATE()) and (s.type='D' or s.type='L')
    ORDER BY backup_start_date DESC, backup_finish_date
GO
-----
```

Security

Loglar makine event viewer (Başlat tuşuna bastıktan sonra "event viewer" ya da run kısmına "eventvwr" yazılarak açılır) ve sql server loglarından (SSMS -> Object Explorer -> Management -> SQL Server Logs -> Current) kontrol edilecek veritabanı ile ilgili hatalar incelenip kritik hatalar raporlanacak.

Capacity (Kapasite)

Veritabanı boyutları ve disk alanı kontrol edildiğinde disk alanı şu anda ihtiyaçları karşılayabilir durumdadır.

-- Kapasite kontrolü için veritabanı boyutlarını veren sorgu

with fs as

```
( select database_id, type, size * 8.0 / 1024/1024 size
from sys.master_files where database_id IN
( SELECT DISTINCT
DB_ID(dbcs.database_name) AS veritabaniid
FROM master.sys.availability_groups AS AG
LEFT OUTER JOIN master.sys.dm_hadr_availability_group_states as agstates
ON AG.group_id = agstates.group_id
INNER JOIN master.sys.availability_replicas AS AR
ON AG.group_id = AR.group_id
INNER JOIN master.sys.dm_hadr_availability_replica_states AS arstates
ON AR.replica_id = arstates.replica_id AND arstates.is_local = 1
INNER JOIN master.sys.dm_hadr_database_replica_cluster_states AS dbcs
ON arstates.replica_id = dbcs.replica_id
LEFT OUTER JOIN master.sys.dm_hadr_database_replica_states AS dbrs
ON dbcs.replica_id = dbrs.replica_id AND dbcs.group_database_id = dbrs.group_database_id
WHERE ISNULL(arstates.role, 3) = 1
union all
select DB_ID(name) veritabaniid from sys.databases where replica_id is null and database_id >4 ) )
select
    name,
    (select sum(size) from fs where type = 0 and fs.database_id = db.database_id) +
    (select sum(size) from fs where type = 1 and fs.database_id = db.database_id) DBSizeGB
from sys.databases db
where (select sum(size) from fs where type = 1 and fs.database_id = db.database_id) is not null
order by DBSizeGB desc
-----
```

System Metric

Sistemin çalışmasını engelleyecek bir query ya da yük var mı kontrol edilecek. Sistem kaynakları bu yapıda devam etmeli mi daha fazla kaynağa ihtiyaç var mı ya da hali hazırdaki kaynaklar gereğinden fazla mı tespit edilecek.

-- Sistemde canlı olarak çalışan querylerin sisteme yaptığı yükü kontrol eden sorgu

```
-----  
SELECT DB_NAME(ST.DBID) AS [DATABASENAME],  
MG.REQUESTED_MEMORY_KB ,  
MG.IDEAL_MEMORY_KB ,  
MG.REQUEST_TIME ,  
MG.GRANT_TIME ,  
MG.QUERY_COST ,  
MG.DOP ,  
ST.[TEXT],  
QP.QUERY_PLAN  
FROM SYS.DM_EXEC_QUERY_MEMORY_GRANTS AS MG  
CROSS APPLY SYS.DM_EXEC_SQL_TEXT(PLAN_HANDLE) AS ST  
CROSS APPLY SYS.DM_EXEC_QUERY_PLAN(MG.PLAN_HANDLE) AS QP  
ORDER BY MG.REQUESTED_MEMORY_KB DESC ;  
-----
```

Tavsiye edilen indexler

Sql server'ın bize önerdiği indexler. Bu indexler üretildiğinde performans artışı bekliyoruz. Ama SQL Server bu konuda her zaman haklı olmayabiliyor. Her index maliyeti ile beraber gelir. Bu yüzden ayrıntılı incelemeler sonucu bu indexler oluşturulmalıdır.

```
-- Missing indexes with CREATE statement for it
SELECT
    DB_NAME(database_id)
    ,MID.[statement] AS ObjectName
    ,MID.equality_columns AS EqualityColumns
    ,MID.inequality_columns AS InequalityColms
    ,MID.included_columns AS IncludedColumns
    ,MIGS.last_user_seek AS LastUserSeek
    ,MIGS.avg_total_user_cost
    * MIGS.avg_user_impact
    * (MIGS.user_seeks + MIGS.user_scans) AS Impact
    ,N'CREATE NONCLUSTERED INDEX <Add Index Name here> ' +
    N'ON ' + MID.[statement] +
    N' ( ' + MID.equality_columns
    + ISNULL(', ' + MID.inequality_columns, N'') +
    N') ' + ISNULL(N'INCLUDE ( ' + MID.included_columns + N');', '')
    AS CreateStatement
FROM sys.dm_db_missing_index_group_stats AS MIGS
INNER JOIN sys.dm_db_missing_index_groups AS MIG
    ON MIGS.group_handle = MIG.index_group_handle
INNER JOIN sys.dm_db_missing_index_details AS MID
    ON MIG.index_handle = MID.index_handle
WHERE
    MIGS.last_user_seek >= DATEDIFF(month, GetDate(), -1)
    and database_id>4
ORDER BY database_id, Impact DESC
```

Maliyetli sorgular (Expensive queries)

SQL Server üzerinde çalışan bütün sorgular arasında en çok maliyet ve kaynak harcadığımız sorgular bu Script ile bulunur. Bu sorguların incelenmesi, eğer yapılabilir ise optimize edilmesi ya da ihtiyaç duyulan indekslerin oluşturulması gerekmektedir.

--EXPENSIVE QUERIES

```
select top 20 coalesce (db_name(st.dbid), db_name(convert (int, pa.value)), 'Empty') as DBName,

    qs.last_execution_time as LastExecutionTime,
    SUBSTRING(st.text, (qs.statement_start_offset/2)+1,
        ((CASE qs.statement_end_offset
            WHEN -1 THEN DATALENGTH(st.text)
            ELSE qs.statement_end_offset
            END - qs.statement_start_offset)/2) + 1) AS StatementText,
    st.text as ProcedureTextOrBatchText,
    qs.execution_count as ExecutionCount,
    (qs.total_worker_time/1000) as CPUTimeTotal,
    ((qs.total_worker_time/1000)/qs.execution_count) as CPUTimeAvg,
    (qs.total_elapsed_time/1000) as DurationTimeTotal,
    ((qs.total_elapsed_time/1000)/qs.execution_count) as DurationTimeAvg,
    qs.total_physical_reads as PhysicalReadsTotal,
    (qs.total_physical_reads/qs.execution_count) as PhysicalReadsAvg,
    qs.total_logical_reads as LogicalReadsTotal,
    (qs.total_logical_reads/qs.execution_count) as LogicalReadsAvg
from sys.dm_exec_query_stats qs
    cross apply sys.dm_exec_sql_text(sql_handle) st
    cross apply sys.dm_exec_query_plan(plan_handle) qp
    cross apply sys.dm_exec_plan_attributes(qs.plan_handle) pa
where attribute = 'dbid'
    and execution_count > 1000
--order by CPUTimeTotal desc
order by CPUTimeAvg desc

--order by DurationTimeTotal desc
--order by DurationTimeAvg desc
--order by PhysicalReadsTotal desc
--order by PhysicalReadsAvg desc
--order by LogicalReadsTotal desc
--order by LogicalReadsAvg desc
```

Database

Sistemde herhangi bir sorgu Lock (Kilitlenme) ya da latency (Gecikme) sebebi mi?

Eğer lock veya latency yapan bir sorgu varsa bunu neden yaptığı kontrol edilir.

Kaynağı tespit edildikten sonra kill edilme ihtimali varsa kill edilir eğer yoksa bitmesi beklenir.

-- Lock ve Latency kontrolü yapacağımız sorgu

select

```
b.query_plan as QueryPlan,  
es.session_id as SPID,  
er.database_id,  
db_name(er.database_id) AS DBName,  
a.text,  
wait_time,  
wait_type,  
blocking_session_id,  
er.cpu_time as CPU,  
er.logical_reads,  
*
```

from

```
sys.dm_exec_sessions es  
inner join sys.dm_exec_requests er on es.session_id = er.session_id  
CROSS APPLY sys.dm_exec_query_plan (Plan_Handle) as b  
CROSS APPLY fn_get_sql(SQL_HANDLE) AS a
```

where er.status in ('running', 'suspended')

```
order by CPU DESC
```

Version

Veritabanı versiyon kontrolü yapılır. Eğer SQL Server versiyonu güncel versiyondan farklı ise update (Güncelleme) planı hazırlanmalıdır.

Güncel versiyon kontrolü öğrenmek için bu siteye bakabilirsiniz: <https://sqlserverbuilds.blogspot.com>

-- Version Kontrol Sorgusu

Select @@Version

Service

SQL Server servisinin çalışıp çalışmadığı SQL Server 2019 Configuration Manager ya da run içine "**services.msc**" yazılarak ve burada çıkan ekranlarda SQL Server (MSSQLSERVER) -> State -> Running kontrol edilerek çalıştığı teyit edilir.

Çalışma Kontrol Listesi Bitiş

Delete Planlarının Oluşturulması

Database serverında dosya ve database içinden veri, obje silineceği zaman silme prosedürüne girerek belirli süre o obje erişime kapatılır ancak varlığını devam ettirir. Kurumun belirlediği zaman diliminde izlenerek ilgili objenin kaldırılması işlemi yapılır. (Bu objeler tablo, index, job, stored procedure, function, alert, maintenance plan, user .vb olabilir)

Index Maintenance ve Monitoring İşlemleri

Kurumun politikaları ve hizmet verme sıklığına bağlı olarak izleme işleminin ve sistemin ölçeklendirilmesi ve buna göre bakımlarının planlanması ve yürütülmesi gerekmektedir. Kurum metrikleri toplanarak kurum hafızasına sahip olduğunda kesintiye sebep olmadan sistem durumunun izlenmesi işlemi yapılır.

Kuruma Yönelik Prosedürlerin Oluşturulması

Kurumun iş bilgisine sahip olunarak yapılan işlemlerin prosedürel olarak bir dayanağa bağlanması ve sistemin bu süreçleri düzenleyen dokümanlar takip edilerek yönetilmesi ve kurumun iş bilgisinin ve kazanımlarının korunması amacıyla süreçler prosedürel dokümanlarla desteklenir.

Bakım Planları Level 3 – Kurumla olan İlişğin Kesilmesi ve Ticket Entegrasyonumuzun Bitirilmesi

Kurumla Yapılan Anlaşmanın Sonlanması tarihi netleştikten sonra sürece başlanır. Kurumun ticket açmasına ve sistemimize dahil olmasına yarayan süreçler kontrollü olarak ilgili sistemlerden kaldırılır. Kurumla ilgilenen kişi veya kişilerin iş planları gözden geçirilir. İş planları düzenlenmesi yapılır.

SqlAdminPremium Veritabanının Kaldırılması

Monitoring ve sistemlerin takibi kapsamında kurulumu ve konfigürasyonu yapılmış olan tarafımızca geliştirilmiş olan izleme veritabanı, ilgili ortamdan uninstall edilir.

```
use [SQLAdminPremium];  
exec sp_UninstallSQLAdmin;
```

Monitoring Sistemimize Ticket Açan Mail Operasyonlarının Disable Edilmesi

Sistemlerde oluşan durumların destek sistemimizde yeni ticketlar üretmesini önlemek amacıyla ilgili mail adresinin gönderdiği maillerin sistemimizde tanımlı olan yetkileri kaldırılır. Hata_Op operatörü disable edildi mi kontrol edilir.

Destek Aryasoft Adresinden İlgili Mail Adreslerinden Mail Almanın Engellenmesi

Monitoring mail adresi destek sisteminde yetkileri alınarak disable durumuna çekilir.

İlgili Ekibin İş Bölümünün Düzenlenmesi

Kurumla ilgilenen kişi veya ekibin iş planlaması ve takvimi incelenerek ekipler arasında iş organizasyonu tekrardan düzenlenir.