

# MANAGING INFRASTRUCTURE SERVICES

MS SQL SERVER 2022



OCTOBER 2025

Submitted to  
Mr Richi FURMAH

Prepared By  
Rajeev KHOODEERAM





# Contents

|                                                   |           |
|---------------------------------------------------|-----------|
| <b>1 PROJECT SUMMARY</b>                          | <b>2</b>  |
| <b>2 INSTALLATION</b>                             |           |
| <b>(Steps 1-3)</b>                                | <b>3</b>  |
| 2.1 Installation of Windows Server 2025 . . . . . | 3         |
| 2.2 Installation of SQL Server 2022 . . . . .     | 3         |
| 2.3 Logins . . . . .                              | 4         |
| <b>3 CONFIGURATION</b>                            |           |
| <b>(steps 4-7)</b>                                | <b>5</b>  |
| 3.1 Configuring TCP port . . . . .                | 5         |
| 3.2 Mail configuration . . . . .                  | 5         |
| 3.3 Operator configuration . . . . .              | 6         |
| 3.4 SSMS Query . . . . .                          | 6         |
| <b>4 DATABASE AND SQL USER / LOGIN</b>            |           |
| <b>(Steps 8-11)</b>                               | <b>7</b>  |
| 4.1 Creating database Project . . . . .           | 7         |
| 4.2 SQL Login on Windows Server . . . . .         | 7         |
| 4.3 SQL User in Project . . . . .                 | 7         |
| 4.4 SMSS Query . . . . .                          | 8         |
| <b>5 EMPLOYEE SCHEMA</b>                          |           |
| <b>(Steps 12-19)</b>                              | <b>9</b>  |
| 5.1 Employee table . . . . .                      | 9         |
| 5.2 Non-clustered indexes . . . . .               | 9         |
| 5.3 SMMS report - r3 . . . . .                    | 10        |
| 5.4 Maintenance plan . . . . .                    | 10        |
| 5.5 User project-access . . . . .                 | 10        |
| 5.6 User project-update . . . . .                 | 11        |
| 5.7 SMSS Report - r5 . . . . .                    | 11        |
| 5.8 Deleting trigger . . . . .                    | 12        |
| <b>6 BACKUP &amp; RESTORE</b>                     |           |
| <b>(Steps 20 - 25)</b>                            | <b>13</b> |
| 6.1 Full backup of Project database . . . . .     | 13        |
| 6.2 Maintenance plan . . . . .                    | 13        |
| 6.3 Project restore . . . . .                     | 14        |
| 6.4 Maintenance plan for errors . . . . .         | 15        |
| 6.5 Check database for errors . . . . .           | 17        |
| 6.6 Performance Data Warehouse . . . . .          | 18        |
| <b>7 PUBLISHER &amp; DISTRIBUTOR</b>              |           |
| <b>(Steps 26 - 31)</b>                            | <b>20</b> |
| 7.1 Replication . . . . .                         | 20        |
| 7.2 SMSS - R11 . . . . .                          | 20        |
| 7.3 SQL Server Agent . . . . .                    | 21        |
| 7.4 ARFA instance - HeadOffice . . . . .          | 22        |
| 7.5 Bulk Export . . . . .                         | 23        |
| 7.6 SMSS - R12 . . . . .                          | 23        |

|          |                                                |           |
|----------|------------------------------------------------|-----------|
| <b>8</b> | <b>MIRRORING</b>                               | <b>25</b> |
| 8.1      | Database mirroring - Manual failover . . . . . | 25        |
| 8.2      | SMSS - R13 . . . . .                           | 25        |
| 8.3      | Mirroring . . . . .                            | 25        |
| 8.4      | Database BranchOffice . . . . .                | 25        |
| 8.5      | SMSS - R14 . . . . .                           | 26        |
| <b>9</b> | <b>REPLICATION</b>                             | <b>27</b> |
| 9.1      | Replicating databases . . . . .                | 27        |
| 9.2      | Testing replication setup . . . . .            | 27        |
| 9.3      | Using SQL Server profiler . . . . .            | 27        |
| 9.4      | Performance counters . . . . .                 | 29        |
| 9.5      | Creating alerts . . . . .                      | 30        |
| 9.6      | SMSS - R17 . . . . .                           | 32        |

# CHAPTER 1

## PROJECT SUMMARY

This project reviews the key topics related to database server administration in SQL Server 2019. Ensure that the following steps are performed on a clean installation of Windows Server 2019.

You are required to install a new SQL Server 2019 database server that will be used by developers within your organization. All server maintenance will be your responsibility and you must communicate/coordinate with the company developer in order to create the appropriate structures and objects on the database server. You will connect to the SQL Server 2019 computer as the Administrator Windows account (or sa account), but all other connections will be done using a separate account.

For this project, I would be using the following :

- UTM Virtualisation (*since I am using a MAC host*)
- Windows Server 2025 as Operating System
- SQL Server 2022



Figure 1.1: Working environment for this project



# CHAPTER 2

## INSTALLATION (Steps 1-3)

### 2.1 Installation of Windows Server 2025

**Step #1 :** Install a new virtual machine with Windows Server 2019. Ensure that this new virtual machine is called DBSERVER and is a domain controller for the domain yourname.com (where yourname is your last name). 1 mark

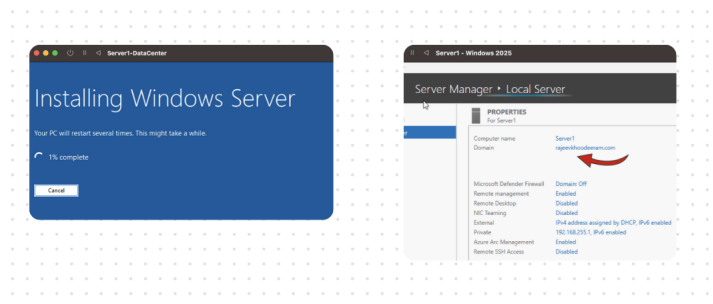


Figure 2.1

### 2.2 Installation of SQL Server 2022

**Step #2 :** Install a default instance of SQL Server 2019/22. 1 mark

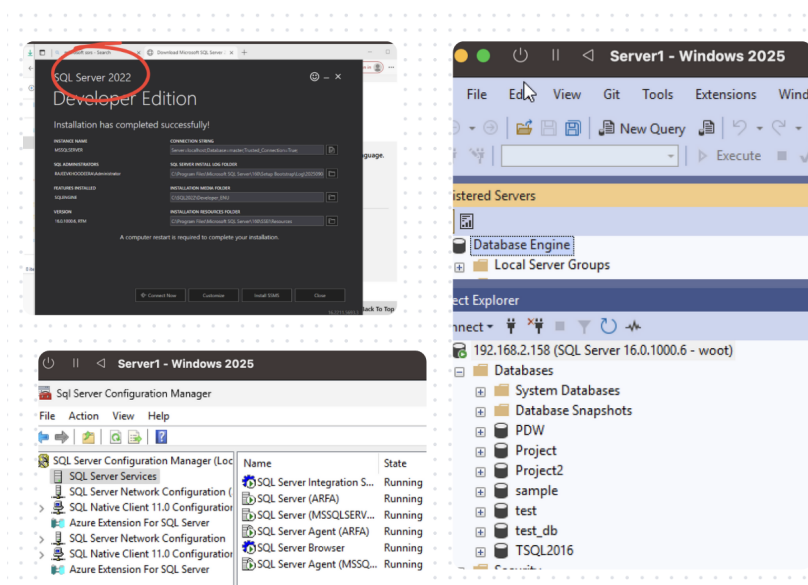


Figure 2.2

## 2.3 Logins

**Step #3 :** Ensure that Windows and SQL logins can be used. Give the sa account a password of Secret555. 1 mark

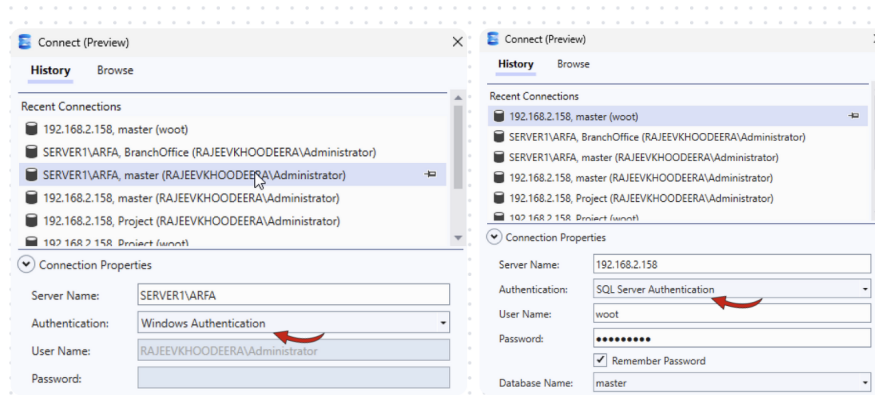


Figure 2.3

# CHAPTER 3

## CONFIGURATION (steps 4-7)

### 3.1 Configuring TCP port

**Step #4.** Allow TCP connections on TCP port 1433 and ensure that all services (including the SQL Server Agent) are set to automatically start each time the server is booted. 2 marks.

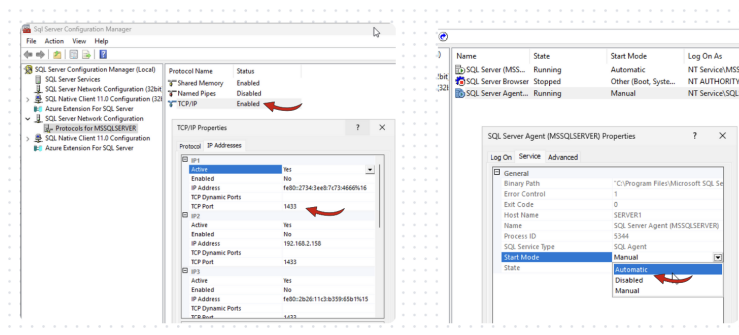


Figure 3.1: TCP port for MSSQL

### 3.2 Mail configuration

**Step #5.** Set up a fake Database Mail profile and account that can be used to send email (the details and name are irrelevant). 1 mark

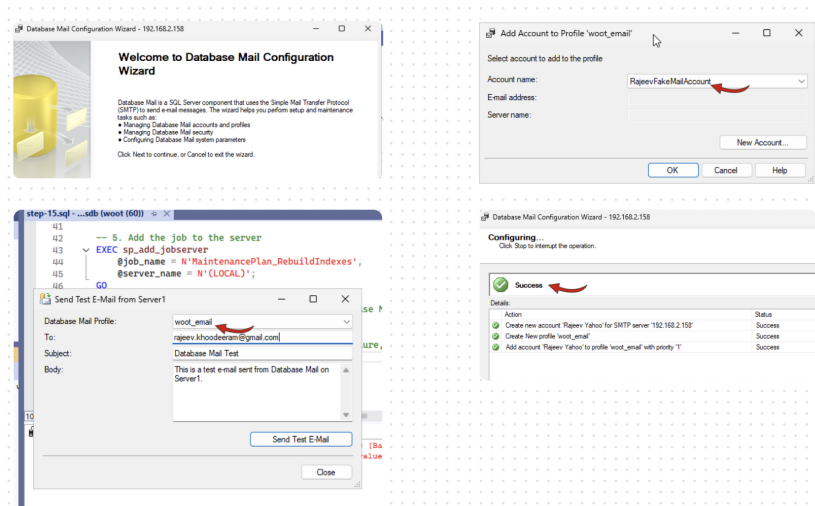


Figure 3.2: Creating a fake database mail



### 3.3 Operator configuration

**Step #6.** Configure an operator for yourself that lists your triOS email address and availability times (be creative here).

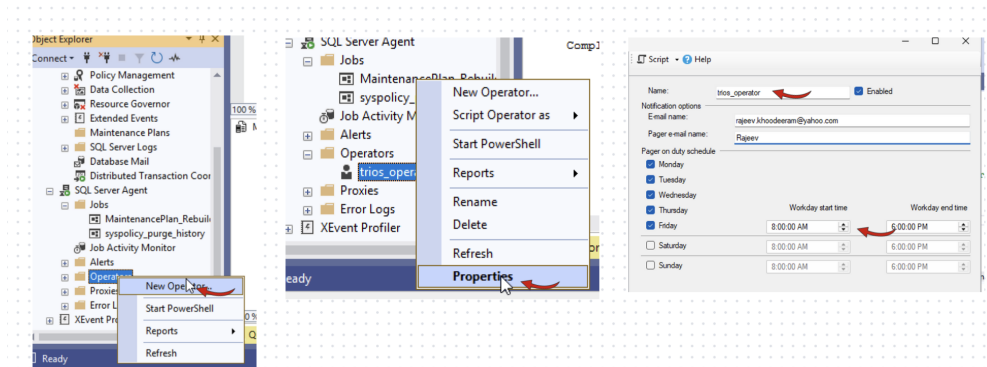


Figure 3.3: Creating a trios operator

### 3.4 SSMS Query

**Step #7.** Create a folder in your virtual machine called *C : \Project*. Next, open a new query in SSMS. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\1.rpt*.

- `SELECT @@SERVERNAME AS 'Server Name'`
- `SELECT DEFAULT_DOMAIN()[DomainName]`
- `SELECT * FROM sys.dm_server_services`
- `SELECT * FROM sys.tcp_endpoints`
- `SELECT * FROM msdb.dbo.sysmail_account`
- `SELECT * FROM msdb.dbo.sysoperators`

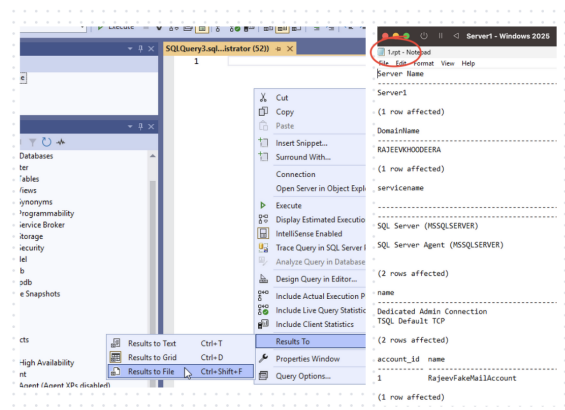


Figure 3.4: Generating RPT 1

# CHAPTER 4

## DATABASE AND SQL USER / LOGIN (Steps 8-11)

### 4.1 Creating database Project

**Step #8.** Create a new database called Project. Ensure that the location of database files optimizes database performance (do not use the default file location). 1 mark.

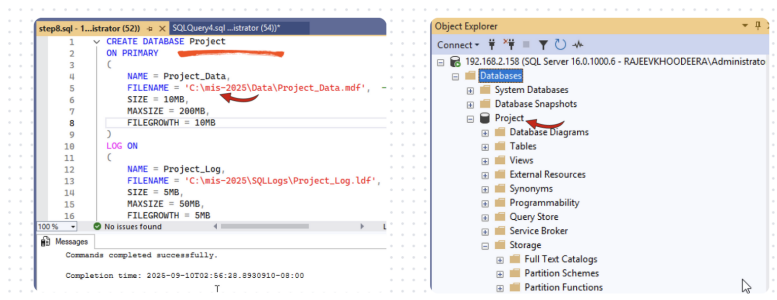


Figure 4.1: Creating database Project

### 4.2 SQL Login on Windows Server

**Step #9.** Create a SQL login for the developer on the server called woot with a password of Secret555. 1 mark

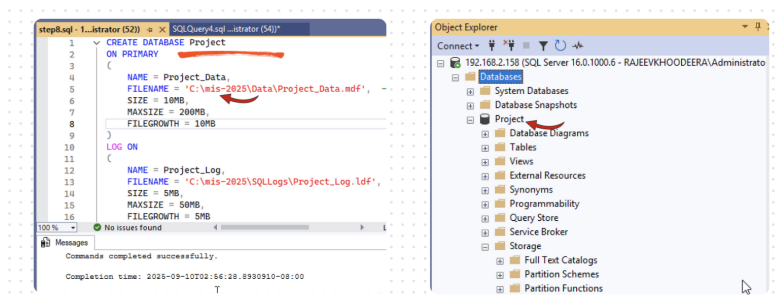
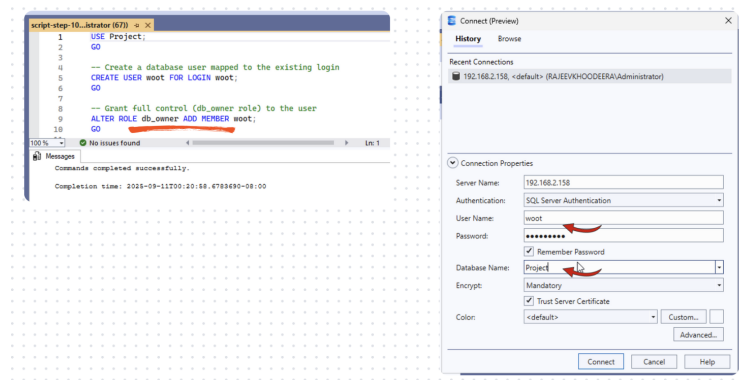


Figure 4.2: Creating database Project

### 4.3 SQL User in Project

**Step #10.** Create a SQL user in the Project database called woot for the developer and ensure that the developer can perform any functions within the Project database. 2 marks

Figure 4.3: Creating database **Project**

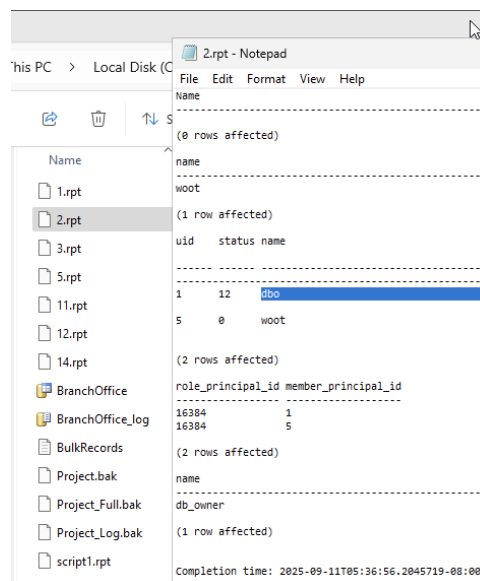
## 4.4 SMSS Query

**Step #11.** Open a new query in SSMS. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\2.rpt*.

```
SELECT db.name AS Name, type\_desc AS Type, Physical\_Name AS Location
FROM sys.master\_files mf
INNER JOIN sys.databases db
ON db.database\_id = mf.database\_id
```

```
SELECT * FROM sys.sql\_logins WHERE name = 'woot'
```

```
USE Project
GO
SELECT * FROM sys.sysusers WHERE hasdbaccess = 1
SELECT * FROM sys.database\_role\_members
SELECT * FROM sys.database\_principals WHERE name = 'db\_owner'
```

Figure 4.4: Generating report *RPT2* – to be reworked in canva

# CHAPTER 5

## EMPLOYEE SCHEMA (Steps 12-19)

### 5.1 Employee table

**Step #12.** Although the developers will be creating most tables in the database, they have asked you to create a table for them in the Project database called Employee in a new schema called HR. A sample of the type of information that this table will store is shown below – use this information to set up the appropriate fields and constraints (ensure that you use the best data types). Ensure that there is a primary key (you can add a field for one) and that there is a clustered index on the primary key field. 6 marks

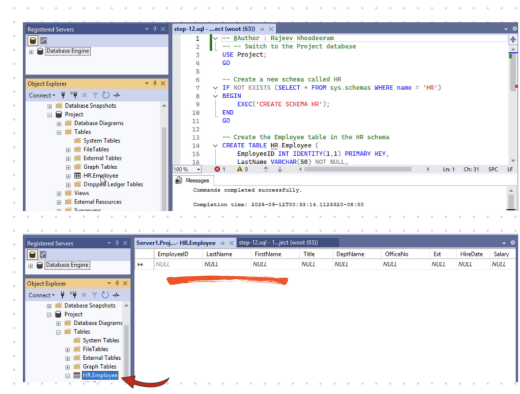


Figure 5.1: Generating report *RPT2* – to be reworked in canva

### 5.2 Non-clustered indexes

**Step #13.** Create 2 separate non-clustered indexes on the last name and office fields. 2 marks

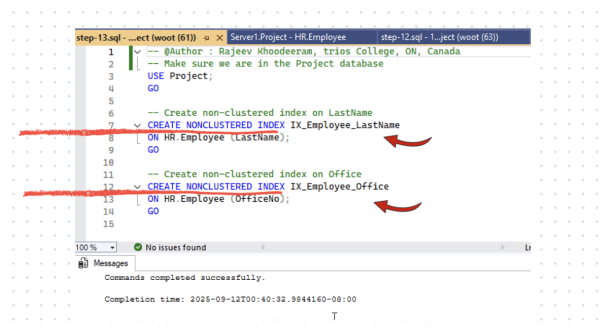


Figure 5.2: Generating report *RPT2* – to be reworked in canva



### 5.3 SMMS report - r3

**Step #14.** Open a new query in SSMS. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\3.rpt*.

- USE Project
- GO
- SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS where TABLE\_NAME='Employee'
- SELECT \* FROM HR.Employee**
- SELECT \* FROM sys.indexes
- WHERE object\_id =
- (SELECT object\_id FROM sys.objects WHERE name = 'Employee')

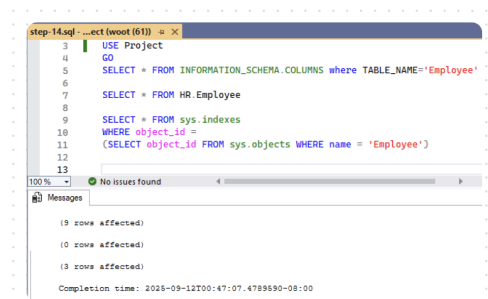


Figure 5.3: Generating report *RPT3*

### 5.4 Maintenance plan

**Step #15.** Create a maintenance plan that rebuilds all indexes on this table and updates all statistics each Sunday at 5:00a.m. Any problems should be emailed to you. Execute your maintenance plan to ensure that it works. Finally, right-click your maintenance plan and choose Modify to view a flowchart of its configuration. Take a screenshot of that configuration and save it as *C : \Project\4.png*.

### 5.5 User project-access

**Step #16.** Create a SQL login and user called project-access that has the password Secret555. Ensure that project-access only has SELECT permissions to the HR.Employee table.

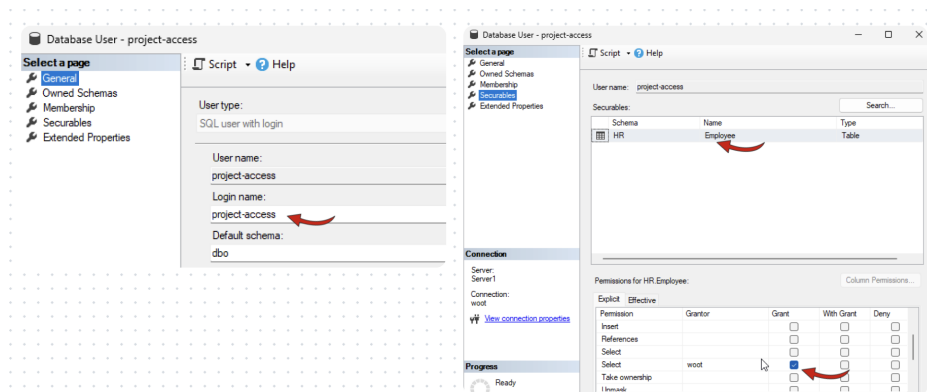


Figure 5.4: Creating user project-access

## 5.6 User project-update

**Step #17.** Create a SQL login and user called project-update that has the password Secret555. Ensure that project-update only has permissions to add, update and delete records in the HR.Employee table.

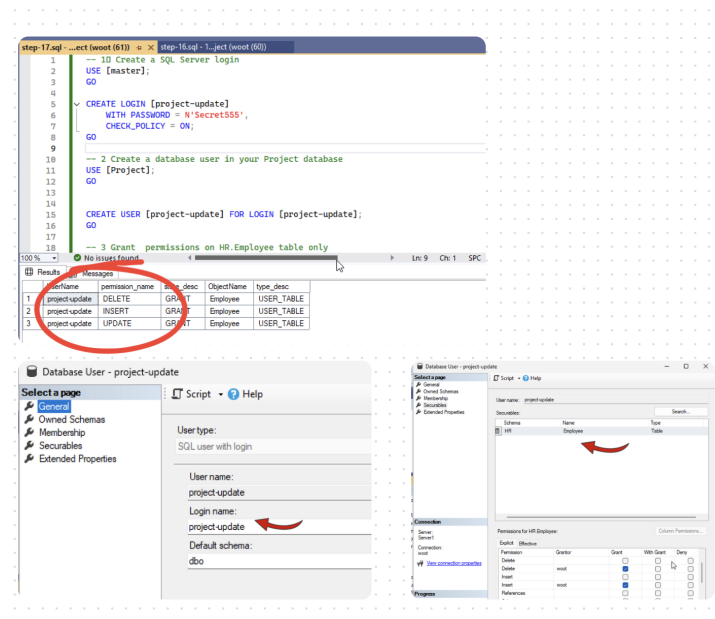


Figure 5.5: Creating user project-update

## 5.7 SMSS Report - r5

**Step #18.** Open a new query in SSMS. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\5.rpt*.

```
SELECT * FROM sys.sql_logins
WHERE name = 'project-access' OR name = 'project-update'

USE Project

GO

SELECT
(
dp.state_desc + ' ' +
dp.permission_name collate latin1_general_cs_as +
' ON ' + '[' + s.name + ']' + ' . ' + '[' + o.name + ']' +
' TO ' + '[' + dpr.name + ']'
) AS GRANT_STMT
FROM sys.database_permissions AS dp
INNER JOIN sys.objects AS o ON dp.major_id=o.object_id
INNER JOIN sys.schemas AS s ON o.schema_id = s.schema_id
INNER JOIN sys.database_principals AS dpr
ON dp.grantee_principal_id=dpr.principal_id
WHERE 1=1 AND o.name IN ('Employee')
ORDER BY dpr.name
```

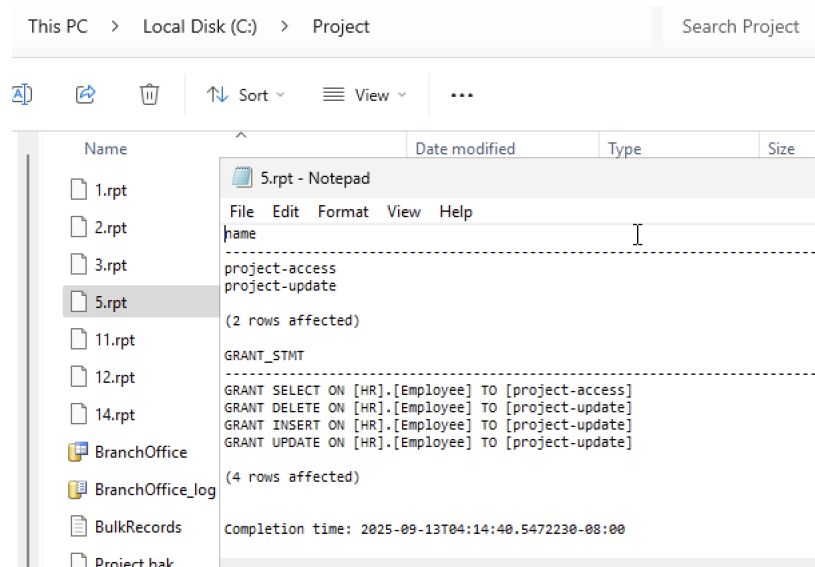


Figure 5.6: Generating SMSS report - 5

## 5.8 Deleting trigger

**Step #19.** Create a trigger that prevents the database developer from deleting tables accidentally in the Project database. Save the query you used to create this trigger to *C : \Project\6.sql*.

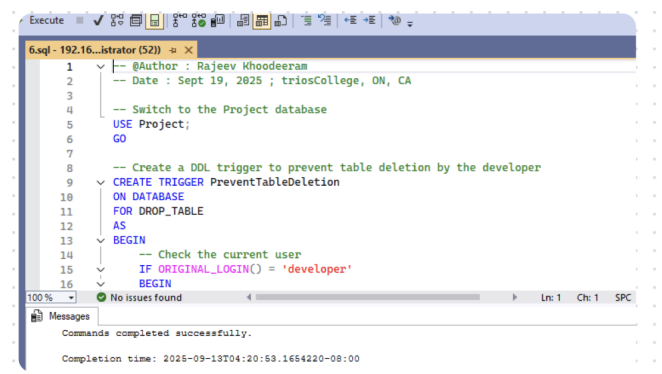


Figure 5.7: Database trigger for developer

# CHAPTER 6

## BACKUP & RESTORE (Steps 20 - 25)

### 6.1 Full backup of Project database

**Step #20.** Manually perform a full backup of your Project database to *C : \Project\Project.bak*.

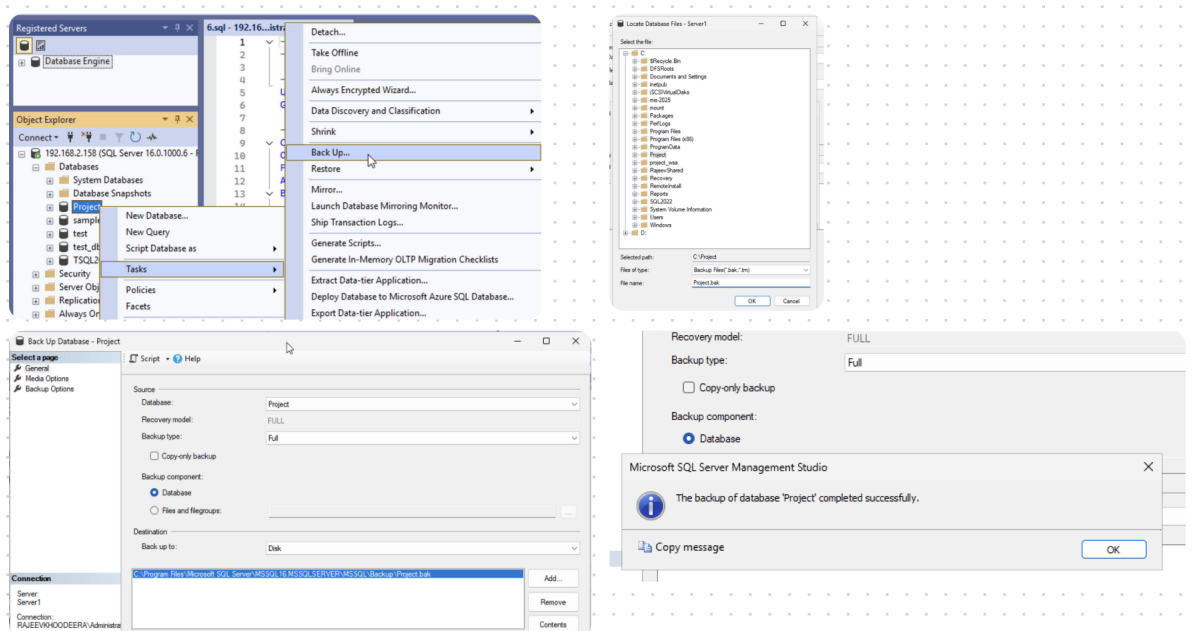


Figure 6.1: Backup of Project database

### 6.2 Maintenance plan

**Step #21.** Ensure that full backups of the Project database are performed every day at 10:00p.m. using a maintenance plan. Execute your maintenance plan to ensure that it works. Finally, right-click your maintenance plan and choose Modify to view a flowchart of its configuration. Take a screenshot of that configuration and save it as *C : \Project\7.png*.

```
-- *****
-- @Author : Rajeev Khoodeeram
-- @Date : Sep 13, 2025; trios College, ON, CA
-- *****
```

```
USE msdb;
GO
```

```
-- =====
-- 1. Create SQL Agent Job
-- =====
```

```

EXEC sp_add_job
@job_name = 'Daily_Full_Backup_ProjectDB_Timestamped',
@enabled = 1,
@description = 'Daily full backup of Project database with timestamp at 10 PM';
GO

-- =====
-- 2. Add Job Step: Perform Timestamped Backup
-- =====
EXEC sp_add_jobstep
@job_name = 'Daily_Full_Backup_ProjectDB_Timestamped',
@step_name = 'FullBackupStep',
@subsystem = 'TSQL',
@command = N'
DECLARE @BackupFile NVARCHAR(500);
SET @BackupFile = ''C:\Project\Project_''
+ REPLACE(CONVERT(VARCHAR, GETDATE(), 120), ':', '' , '')
+ ''.bak'';

BACKUP DATABASE Project
TO DISK = @BackupFile
WITH FORMAT, INIT,
NAME = ''Full Backup of Project Database'';
',
@on_success_action = 1, -- Quit with success
@on_fail_action = 2;    -- Quit with failure
GO

-- =====
-- 3. Create Schedule: Daily at 10:00 PM
-- =====
EXEC sp_add_schedule
@schedule_name = 'Daily_10PM',
@freq_type = 4,          -- Daily
@freq_interval = 1,      -- Every day
@active_start_time = 220000; -- 10:00 PM
GO

-- Attach the schedule to the job
EXEC sp_attach_schedule
@job_name = 'Daily_Full_Backup_ProjectDB_Timestamped',
@schedule_name = 'Daily_10PM';
GO

-- =====
-- 4. Assign Job to SQL Server
-- =====
EXEC sp_add_jobserver
@job_name = 'Daily_Full_Backup_ProjectDB_Timestamped',
@server_name = @@SERVERNAME;
GO

-- =====
-- 5. Optional: Test the Job Immediately
-- =====
-- EXEC msdb.dbo.sp_start_job @job_name = '
--     Daily_Full_Backup_ProjectDB_Timestamped';
GO

```

## 6.3 Project restore

**Step #22.** Create a SQL Query file that can quickly be executed to restore the previous night's backup (without losing any transactions since the previous night's backup). Perform a sample restore using this query to make sure it works. Finally, save the query to *C:\Project\8.sql*.

```
-- *****
-- @Author : Rajeev Khoodeeram
-- @Date   : September 13, 2025; triosCollege, ON, CA
-- *****

-- =====
-- Restore the Project database from previous night's backup
-- without losing transactions since the backup
-- =====

-- Switch to master database
USE master;
GO

-- 1. Set the database to single-user mode (optional, ensures restore can occur)
ALTER DATABASE Project
SET SINGLE_USER
WITH ROLLBACK IMMEDIATE;
GO

-- 2. Restore the full backup from the previous night
-- Replace the file name with the actual backup you want to restore
-- Here, using a timestamped backup pattern, you may need to update the date
    manually
DECLARE @BackupFile NVARCHAR(500);
SET @BackupFile = 'C:\Project\Project.bak'; -- Example: previous night's backup

RESTORE DATABASE Project
FROM DISK = @BackupFile
WITH NORECOVERY, -- Keep database ready for restoring subsequent log backups
REPLACE;         -- Overwrite the existing database
GO

-- 3. Bring the database back online after restoring all backups
--ALTER DATABASE Project
--SET MULTI_USER;
--GO
```

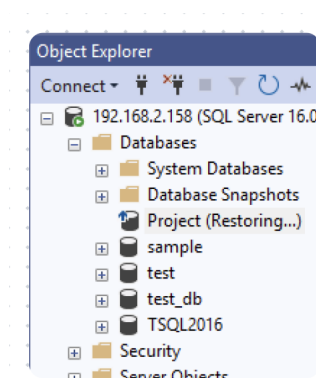


Figure 6.2: Restoring the project database

## 6.4 Maintenance plan for errors

**Step #23.** Ensure that the Project database is automatically checked for errors and then shrunk every Monday and Thursday at 2:00am using a single maintenance plan. Execute your maintenance plan to ensure that it works. Finally, right-click your maintenance plan and choose Modify to view a flowchart of its configuration. Take a screenshot of that configuration and save it as *C : \Project\9.png*.

```
-- *****
-- @Author : Rajeev Khoodeeram
```



```

-- trios College Ontario, Canada
-- @Date : September 13,2025
-- *****

USE msdb;
GO

-- =====
-- 1. Create SQL Agent Job
-- =====

EXEC sp_add_job
@job_name = 'Project_DB_Maintenance',
@enabled = 1,
@description = 'Checks Project DB for errors and shrinks it every Monday &
               Thursday at 2 AM';
GO

-- =====
-- 2. Add Job Step 1: Check for Errors (DBCC CHECKDB)
-- =====

EXEC sp_add_jobstep
@job_name = 'Project_DB_Maintenance',
@step_name = 'CheckForErrors',
@subsystem = 'TSQL',
@command = N'DBCC CHECKDB (Project) WITH NO_INFOMSGS;',
@on_success_action = 3, -- Go to next step
@on_fail_action = 2;    -- Quit with failure
GO

-- =====
-- 3. Add Job Step 2: Shrink Database
-- =====

EXEC sp_add_jobstep
@job_name = 'Project_DB_Maintenance',
@step_name = 'ShrinkDatabase',
@subsystem = 'TSQL',
@command = N'DBCC SHRINKDATABASE (Project);',
@on_success_action = 1, -- Quit with success
@on_fail_action = 2;    -- Quit with failure
GO

-- =====
-- 4. Create Schedule: Mondays & Thursdays at 2:00 AM
-- =====

EXEC sp_add_schedule
@schedule_name = 'MonThu_2AM',
@freq_type = 8, -- Weekly
@freq_interval = 18, -- Monday (2) + Thursday (16) = 18
@active_start_time = 20000; -- 2:00 AM (HHMMSS format)
GO

-- Attach the schedule to the job
EXEC sp_attach_schedule
@job_name = 'Project_DB_Maintenance',
@schedule_name = 'MonThu_2AM';
GO

-- =====
-- 5. Assign Job to SQL Server
-- =====

EXEC sp_add_jobserver
@job_name = 'Project_DB_Maintenance',
@server_name = @@SERVERNAME;
GO

-- =====

```

```
-- 6. (Optional) Run Job Immediately for Testing
-- =====
-- EXEC msdb.dbo.sp_start_job @job_name = 'Project_DB_Maintenance';
GO
```

## 6.5 Check database for errors

**Step #24.** Check your database for errors. Perform a sample repair in single user mode and return to multiuser mode when finished. Save the commands you performed in a file called *C : \Project\10.sql*.

```
-- *****
-- @Author : Rajeev Khoodeeram
-- trios College, ON, CA
-- @Date : September 13, 2025
-- *****

-- =====
-- Step 1: Run DBCC CHECKDB to check for errors
-- =====
DBCC CHECKDB (Project) WITH NO_INFOMSGS;
GO

-- =====
-- Step 2: Set database to SINGLE_USER mode
-- =====
ALTER DATABASE Project
SET SINGLE_USER
WITH ROLLBACK IMMEDIATE;
GO

-- =====
-- Step 3: Perform a sample repair
-- (REPAIR_ALLOW_DATA_LOSS is the most aggressive option,
-- REPAIR_REBUILD is safer if available)
-- =====
DBCC CHECKDB (Project, REPAIR_REBUILD);
-- Example of more aggressive option:
-- DBCC CHECKDB (Project, REPAIR_ALLOW_DATA_LOSS);
GO

-- =====
-- Step 4: Return database to MULTI_USER mode
-- =====
ALTER DATABASE Project
SET MULTI_USER;
GO
```

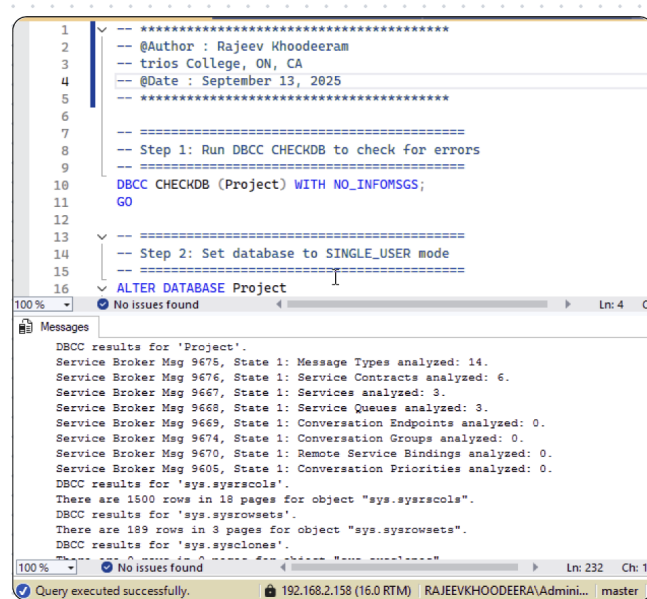


Figure 6.3: Checking database for errors

## 6.6 Performance Data Warehouse

**Step #25.** Configure the Performance Data Warehouse to monitor the key areas of your SQL server (this data should be stored in a database called PDW). After a period of time (such as 1 day), generate the three reports (Disk Usage, Server Activity, Query Statistics). Right-click each report and choose Export > PDF and save it to *C : \Project\* using the default filename. You should have 3 reports in this folder:

- *C : \Project\DiskUsageSummary < date > .pdf*
- *C : \Project\QueryStatisticsHistory < date > .pdf*
- *C : \Project\ServerActivityHistory < date > .pdf*

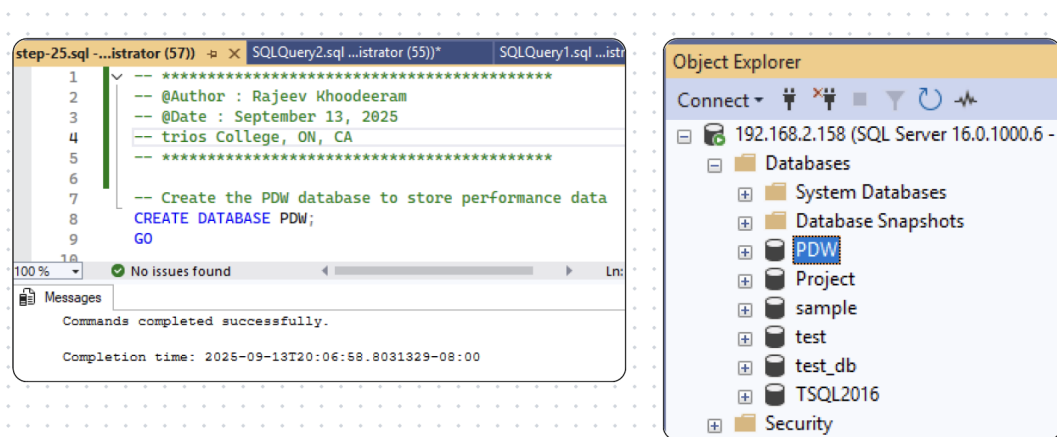


Figure 6.4: Database PDW

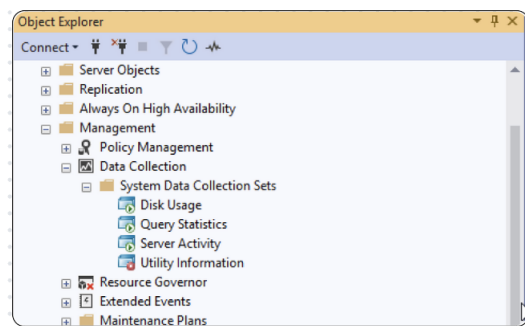


Figure 6.5: Generating reports (Disk usage, Query statistics, Server Activity)

# CHAPTER 7

## PUBLISHER & DISTRIBUTOR (Steps 26 - 31)

### 7.1 Replication

**Step #26.** Enable your system as a publisher and distributor (use the default database name of distribution for your distributor database). Next, configure snapshot replication to back up the Employee table in the Project database to a new database you create called Project2. 2 marks Although Project2 would normally be on another SQL server, you will create Project2 on your own SQL server instance.

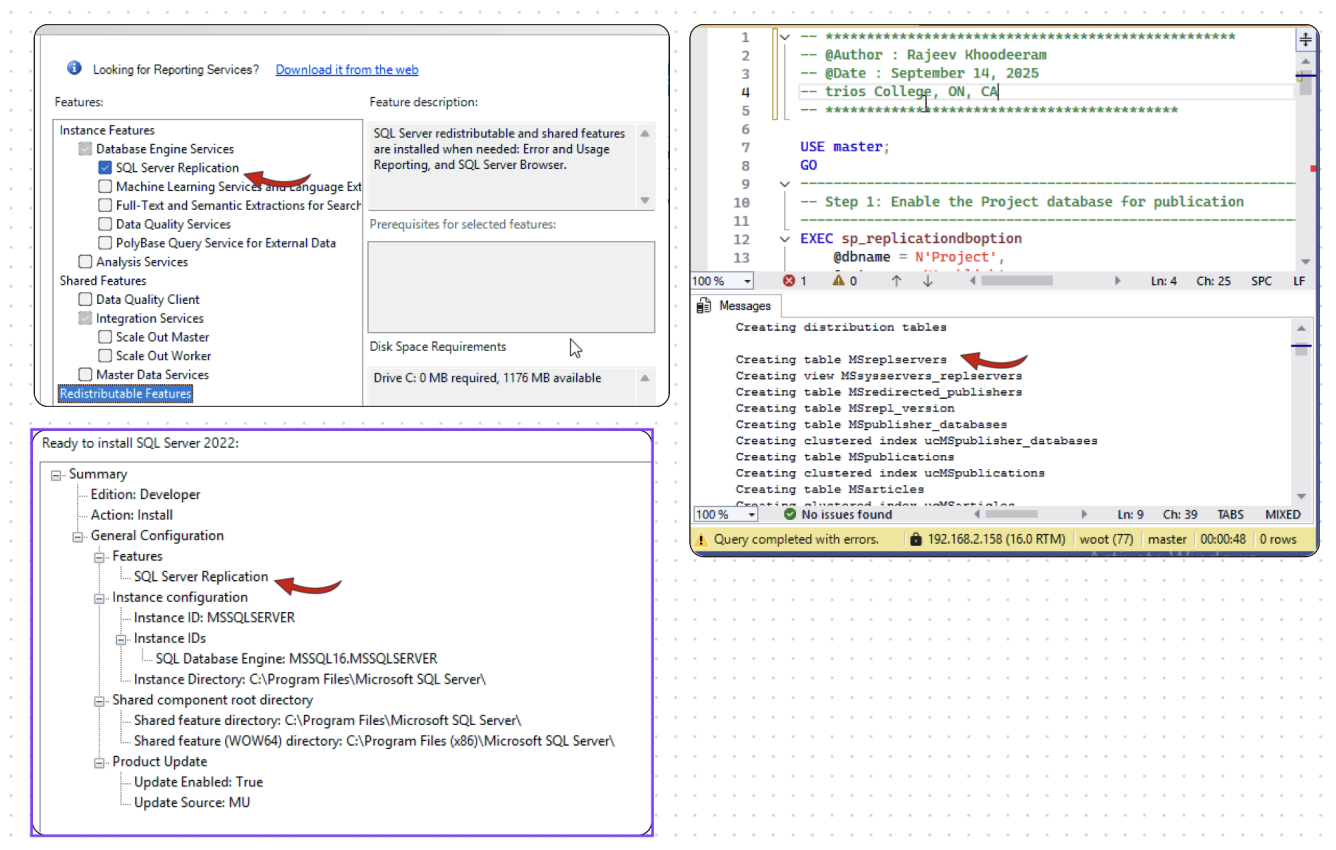


Figure 7.1: Publisher and Distributor configuration

### 7.2 SMSS - R11

**Step #27.** Open a new query in SSMS. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\11.rpt*.

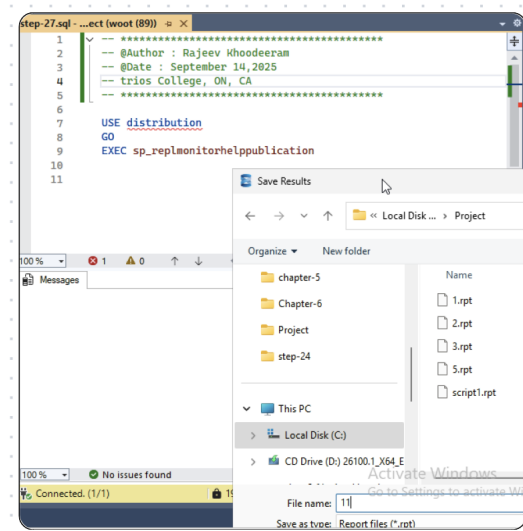


Figure 7.2: Generating SMSS Report 11

## 7.3 SQL Server Agent

**Step #28.** Create a 2nd (named) instance of SQL on your computer called ARFA that allows network access. Ensure that the SQL Server Agent is started and set to start automatically at boot time.

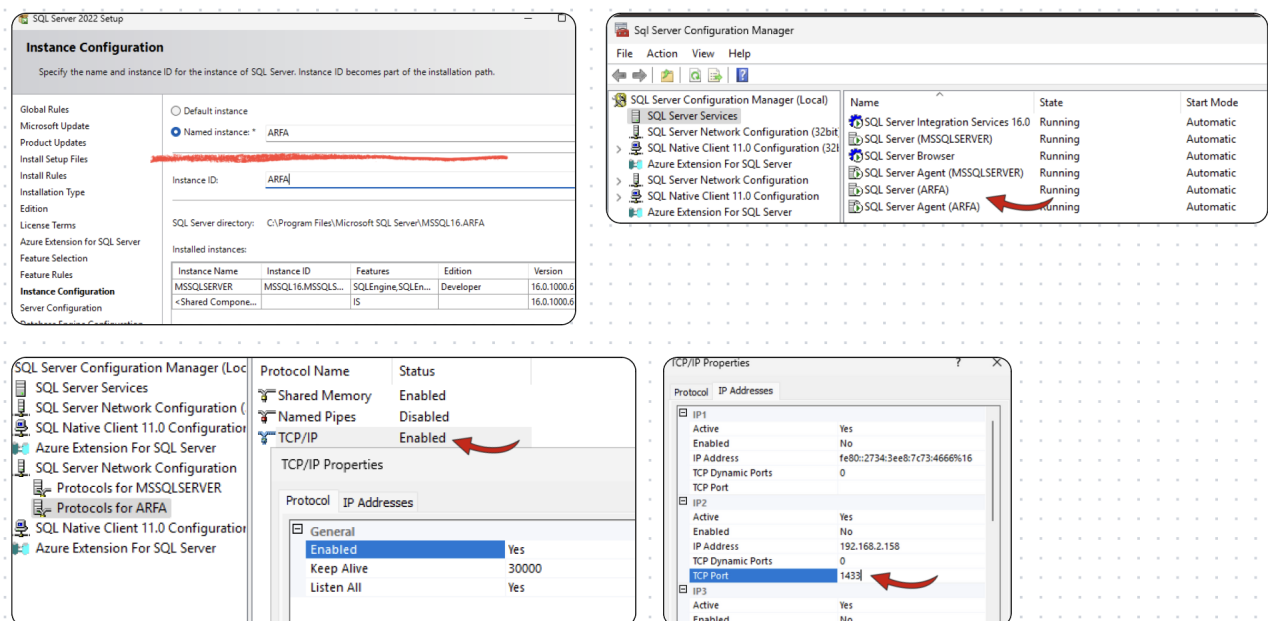


Figure 7.3: Second SQL Instance - ARFA

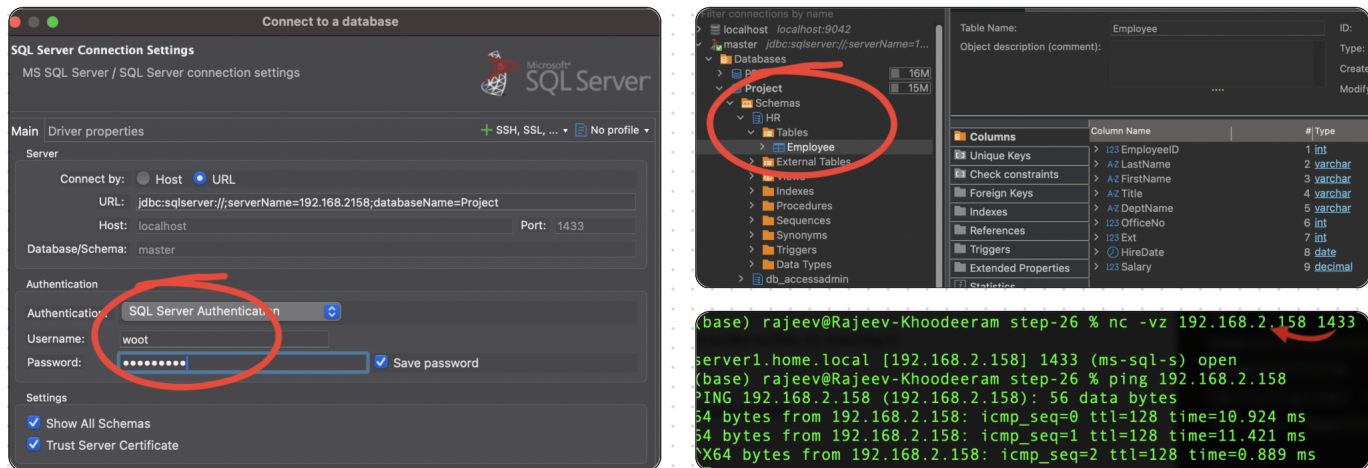


Figure 7.4: Testing remote access to ARFA : Left (Using DBeaver for SQL database management); Right (Using terminal on MAC); SQL Server is on VM UTM on Windows Server

## 7.4 ARFA instance - HeadOffice

**Step #29.** Create a new database on your ARFA instance called HeadOffice (use appropriate locations for the database and log files).

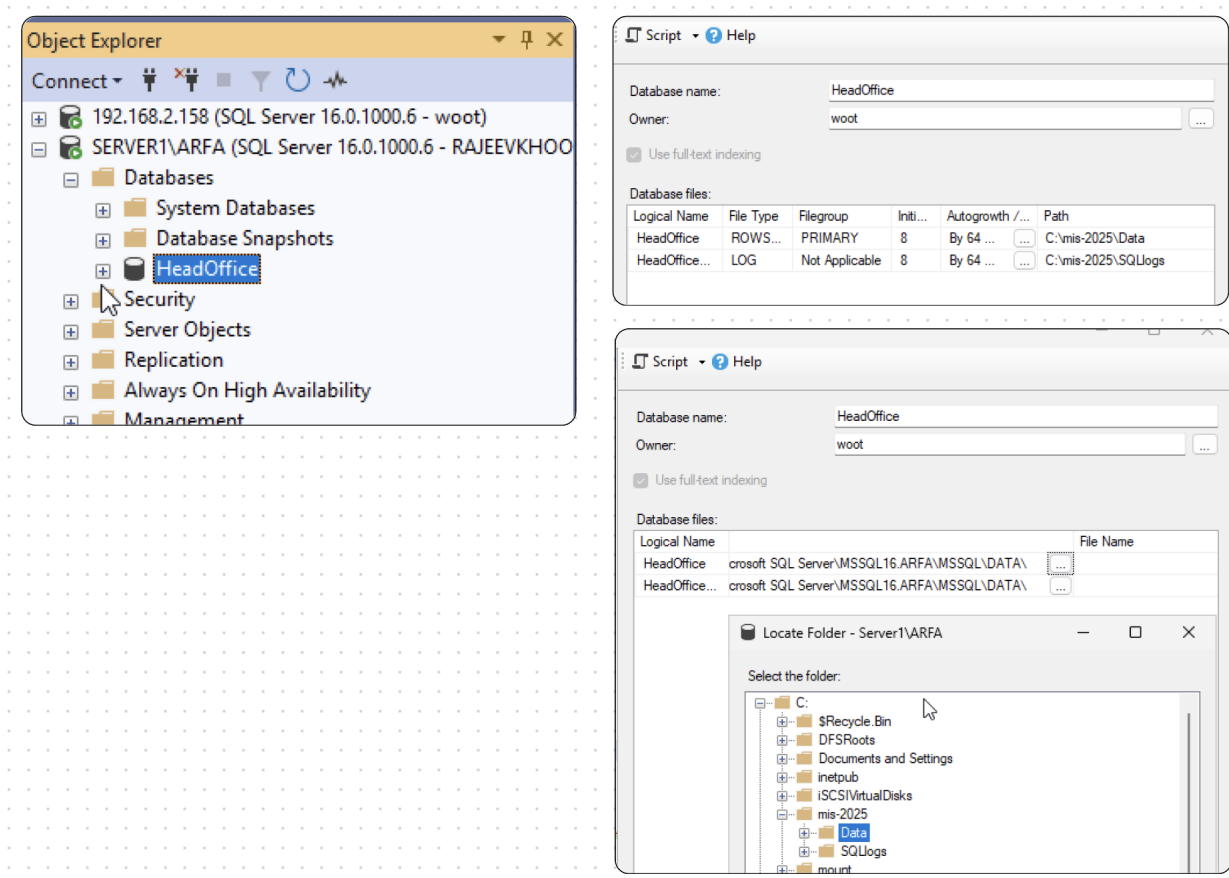


Figure 7.5: Creating HeadOffice database on ARFA



## 7.5 Bulk Export

**Step #30.** Perform a bulk export of the data within that table that you created within your project database to a text file called *C : \Project\BulkRecords.txt*. Following this, bulk import the data into the same table within the HeadOffice database on your ARFA instance.

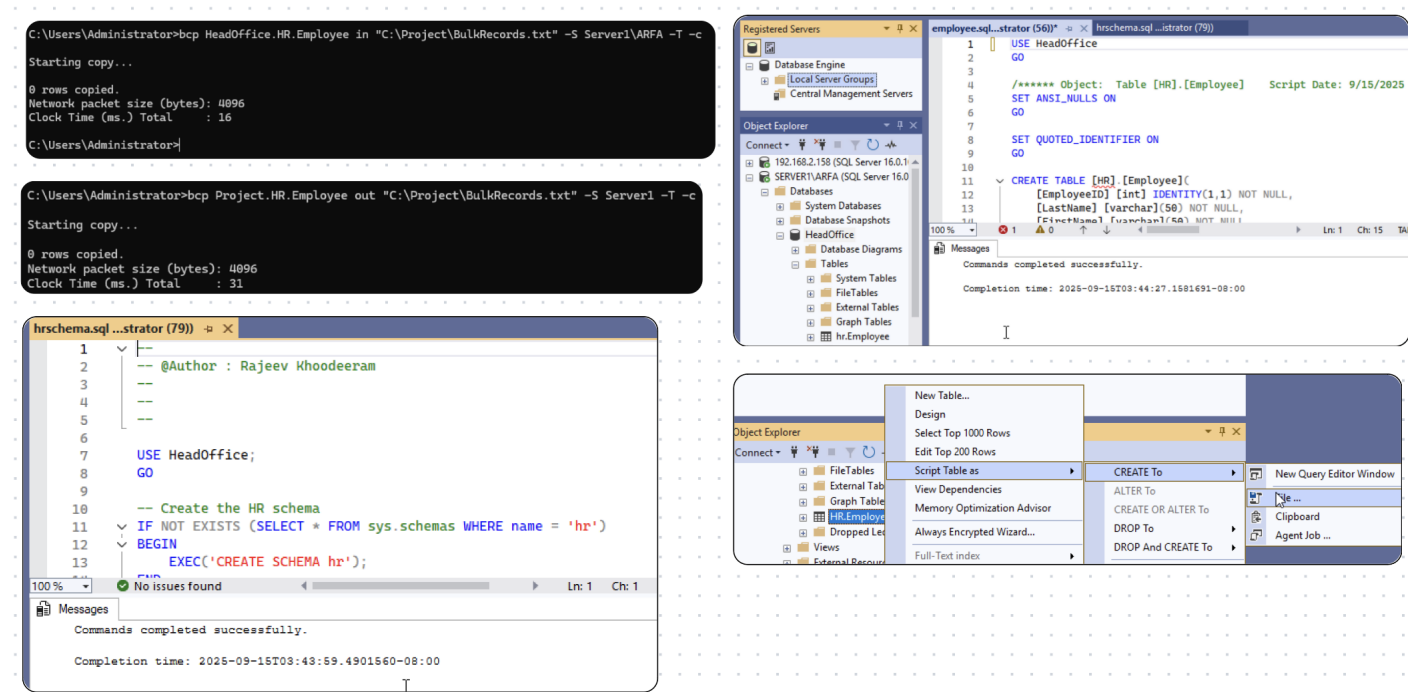
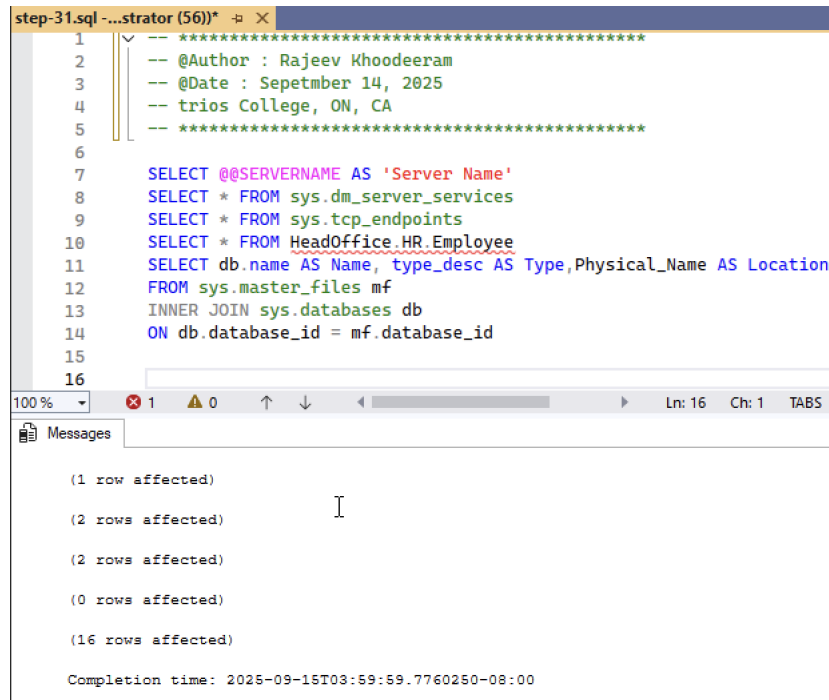


Figure 7.6: Export database to ARFA instance

## 7.6 SMSS - R12

**Step #31.** Open a new query in SSMS that is connected to your ARFA instance. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\12.rpt*.

```
SELECT @@SERVERNAME AS 'Server Name'
SELECT * FROM sys.dm_server_services
SELECT * FROM sys.tcp_endpoints
SELECT * FROM HeadOffice.HR.Employee
SELECT db.name AS Name, type_desc AS Type, Physical_Name AS Location
FROM sys.master_files mf
INNER JOIN sys.databases db
ON db.database_id = mf.database_id
```



The screenshot shows a SQL Developer window titled 'step-31.sql -...strator (56)\*'. The main editor contains a SQL script with 16 lines. Lines 1-6 are comments providing author and date information. Lines 7-16 are SQL statements: a SELECT statement for @@SERVERNAME, followed by three SELECT statements for sys.dm\_server\_services, sys.tcp\_endpoints, and HeadOffice.HR.Employee, and finally a SELECT statement joining sys.master\_files and sys.databases. The Messages pane at the bottom shows the execution results: (1 row affected), (2 rows affected), (2 rows affected), (0 rows affected), and (16 rows affected). The completion time is 2025-09-15T03:59:59.7760250-08:00.

```

1  -- *****
2  -- @Author : Rajeev Khoodeeram
3  -- @Date : Sepetmber 14, 2025
4  -- trios College, ON, CA
5  -- *****
6
7  SELECT @@SERVERNAME AS 'Server Name'
8  SELECT * FROM sys.dm_server_services
9  SELECT * FROM sys.tcp_endpoints
10 SELECT * FROM HeadOffice.HR.Employee
11 SELECT db.name AS Name, type_desc AS Type, Physical_Name AS Location
12 FROM sys.master_files mf
13 INNER JOIN sys.databases db
14 ON db.database_id = mf.database_id
15
16

```

100 % 1 0 16 Ln: 16 Ch: 1 TABS

Messages

(1 row affected)

(2 rows affected)

(2 rows affected)

(0 rows affected)

(16 rows affected)

Completion time: 2025-09-15T03:59:59.7760250-08:00

Figure 7.7: Generating report R12

# CHAPTER 8

## MIRRORING

### 8.1 Database mirroring - Manual failover

**Step #32.** Set up database mirroring for your Project database between your default and 2nd instance that uses manual failover. Perform 2 manual failovers.

### 8.2 SMSS - R13

**Step #33.** Open a new query in SSMS that is connected to your default instance. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\13a.rpt*. Next, repeat this step while connected to your ARFA instance but save the results to *C : \Project\13b.rpt*.

### 8.3 Mirroring

**Step #34.** Remove the mirror to restore your Project database to its original state.

### 8.4 Database BranchOffice

**Step #35.** Create a new database on your 2nd instance called BranchOffice (use appropriate locations for the database and log files).

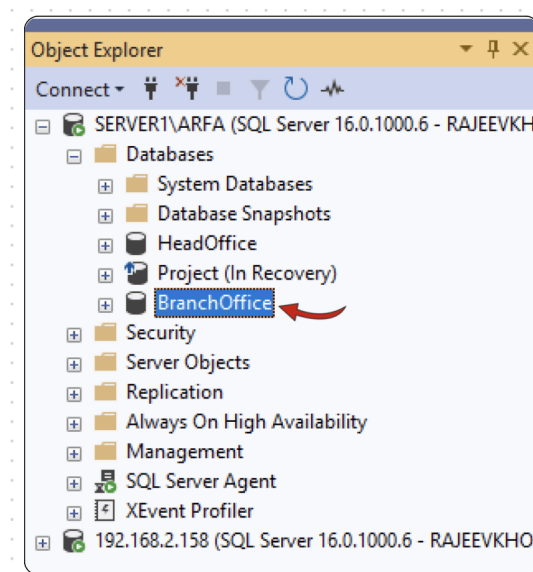


Figure 8.1: Creating database *BranchOffice*

## 8.5 SMSS - R14

**Step #36.** Open a new query in SSMS that is connected to your ARFA instance. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\14.rpt*.

```
SELECT db.name AS Name, type_desc AS Type, Physical_Name AS Location
FROM sys.master_files mf
INNER JOIN sys.databases db
ON db.database_id = mf.database_id
```

# CHAPTER 9

## REPLICATION

### 9.1 Replicating databases

**Step #37.** Set up replication so that the table that you created in the Project database on your default instance is replicated to the BranchOffice database on your ARFA instance using merge replication.

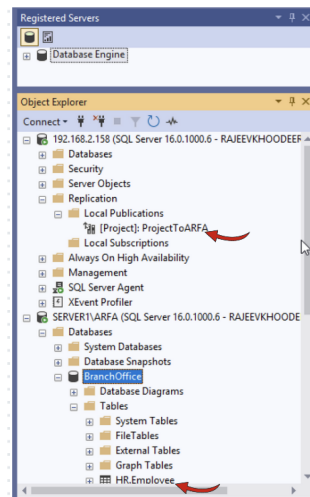


Figure 9.1: Replication of HR.Employee from Project database to BranchOffice database

### 9.2 Testing replication setup

**Step #38.** Add a new record to your BranchOffice database and ensure that it is added to your Project database. When finished, open the Replication Monitor and take a screenshot of your replication status (expand your publication) and save it as *C : \Project\15.png*.

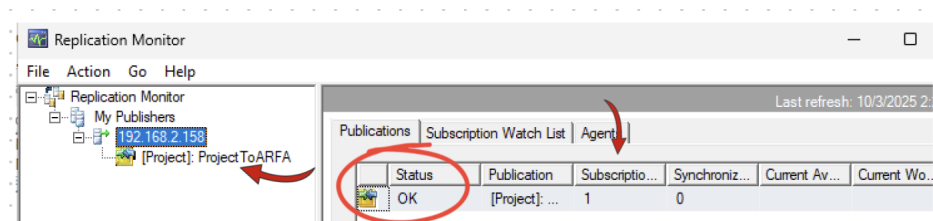


Figure 9.2: Replicating record from BranchOffice to Project database

### 9.3 Using SQL Server profiler

**Step #39.** Use the SQL Server Profiler to capture the existing performance of your Project database in a single trace file called *C : \Project\PerformanceTrace.trc* (no file rollover). Next, use the DTA to analyze the trace file and produce index recommendations (these will be poor recommendations because

this is not a production SQL server). Take a screenshot of the recommendations and save it as *C : \Project\16.png*.

**Step 1: Start SQL Server Profiler and Create a Trace**

- Open SQL Server Profiler (from SSMS → Tools → SQL Server Profiler).
- Connect to your SQL Server instance.
- File → New Trace → give it a name like ProjectTrace.
- In the Trace Properties window:
- General Tab:
- Select your database.
- Set the trace file name to:
- *C : \Project\PerformanceTrace.trc*
- Uncheck "Enable file rollover".
- Click Run to start the trace.

**Step 2: Use the Database Engine Tuning Advisor (DTA)**

- Open Database Engine Tuning Advisor (DTA).
- Connect to your SQL Server instance.
- File → New Session.
- In the Workload section:
- Select File.
- Browse and choose:
- *C : \Project\PerformanceTrace.trc*
- In the Database for Workload Analysis section:
- Select your Project database.
- In Tuning Options:
- Leave defaults (Indexes, Indexed Views, Partitions).
- (Optional) Limit tuning time if you just want quick results.
- Click Start Analysis.

**Step 3: Review the Recommendations**

- DTA will analyze the trace and output:
- Recommended indexes
- Recommended indexed views
- Possible partitioning suggestions
- Since this is not a production system, the recommendations may not be meaningful, but you'll see how the tool works.
- You can also view the Expected Improvement (%).

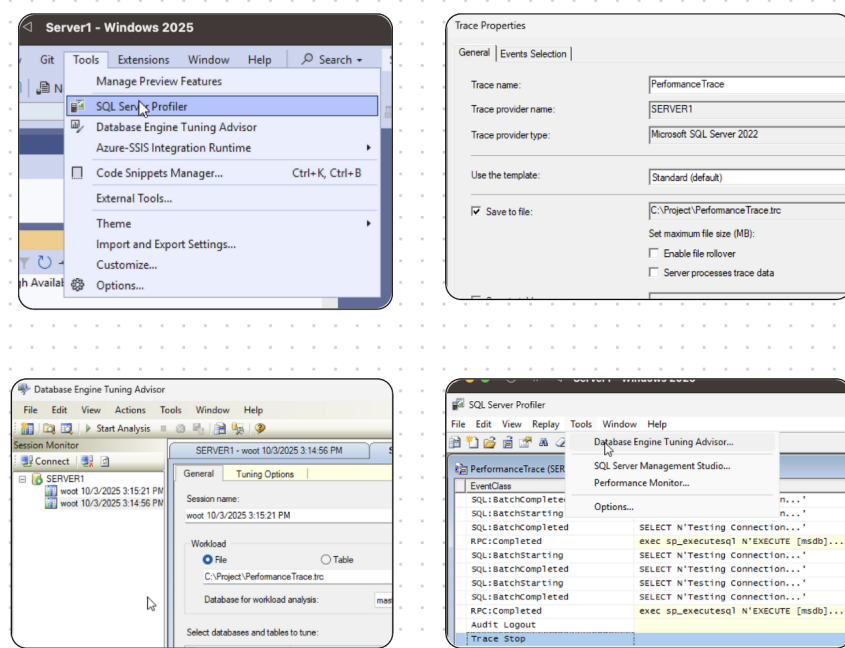


Figure 9.3: Performance trace

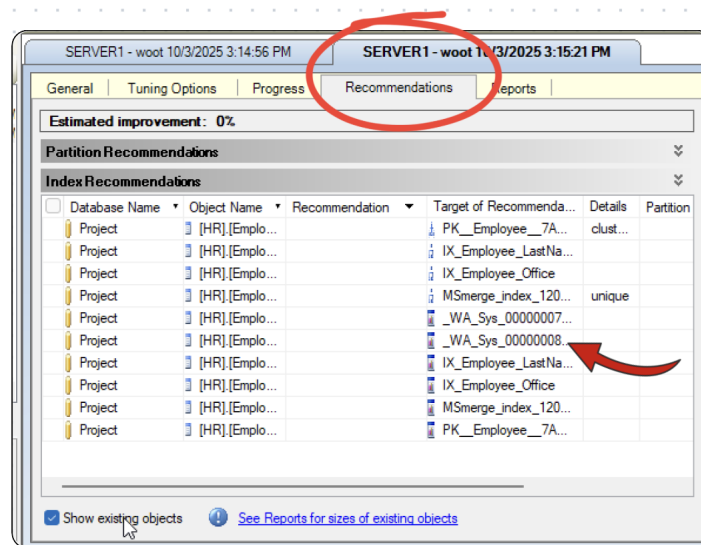


Figure 9.4: Possible recommendations

## 9.4 Performance counters

**Step #40.** Add appropriate performance counters to the Performance Console in Windows and create a baseline for later use in HTML (web page) format. Save the baseline as *C : \Project\Baseline.htm*.





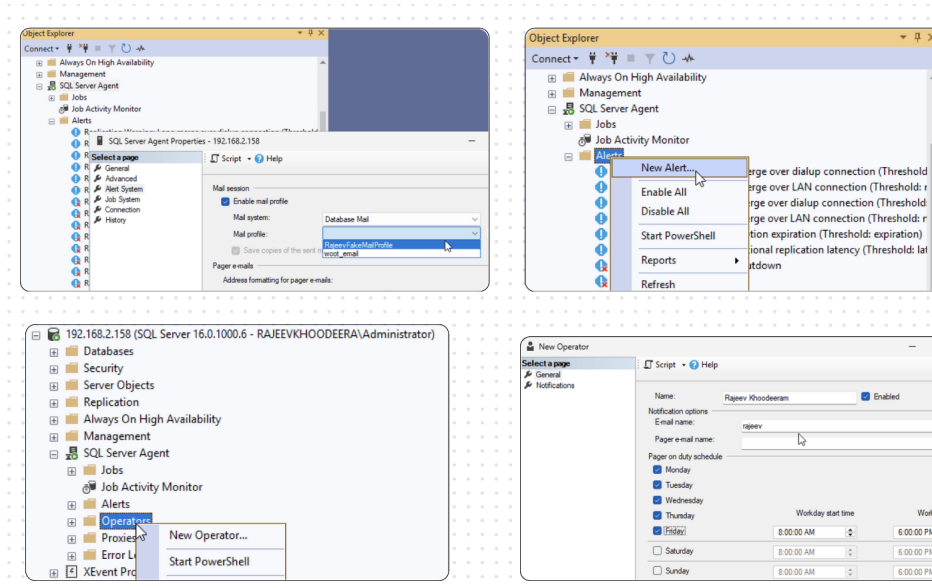


Figure 9.6: Configuring alert and operator

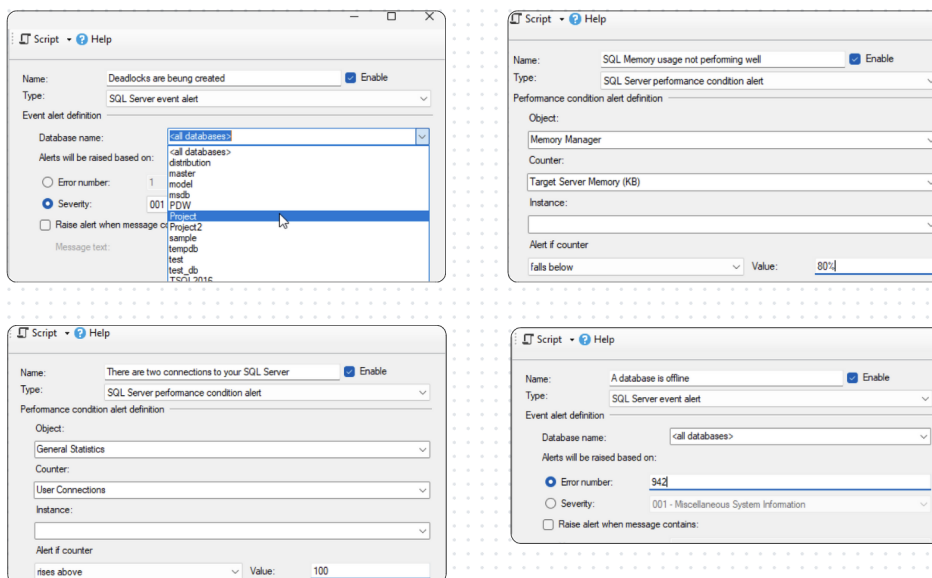


Figure 9.7: Creating Alert 1 - 4

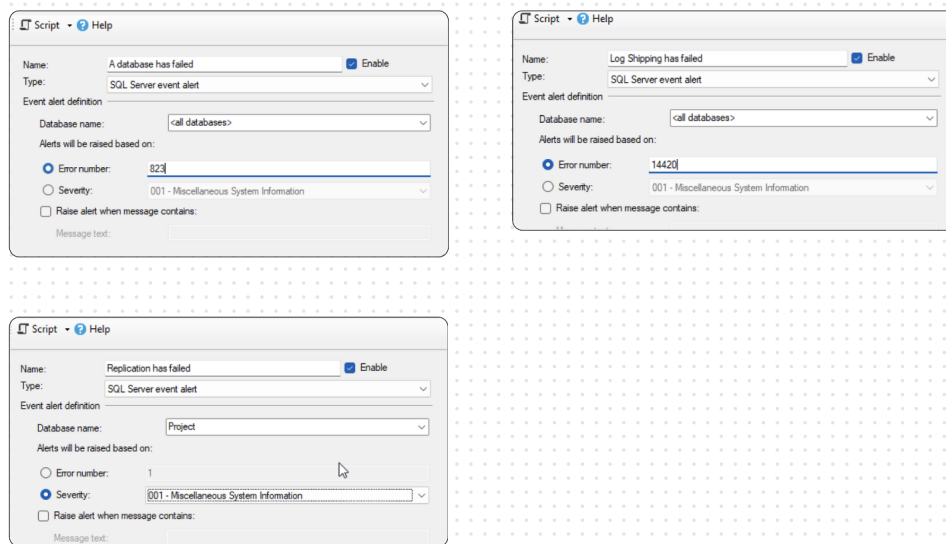


Figure 9.8: Creating Alert 5 - 7

## 9.6 SMSS - R17

**Step #42.** Open a new query in SSMS that is connected to your default instance. Right-click the white background of the query window and click Results To > Results to File. Next, execute the following queries in this window. When prompted to save the results to a file, choose *C : \Project\17.rpt*.

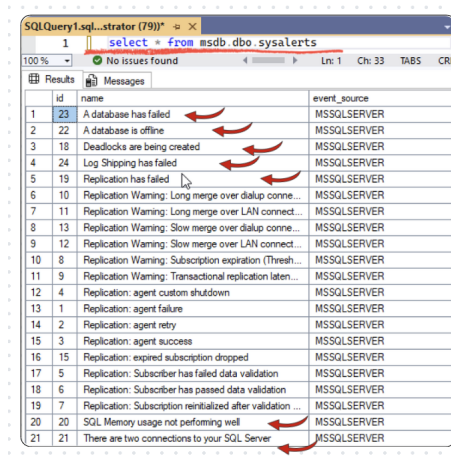


Figure 9.9: Generating SMSS Report - R17