Imported with seeder application: 5.920.000 products

Pagination:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

Without AsNoTracking() + List (removed IQueryable)

IQueryable<Product> query = \_dbContext.Products.AsNoTracking(); ->

var query = \_dbContext.Products

.Where(p =>

(!queryParams.MinPrice.HasValue || p.Price >= queryParams.MinPrice.Value) &&

(!queryParams.MaxPrice.HasValue || p.Price <= queryParams.MaxPrice.Value) &&

(string.IsNullOrEmpty(queryParams.Category) || p.Category == queryParams.Category) &&

(string.IsNullOrEmpty(queryParams.Search) ||

p.Name.Contains(queryParams.Search) ||

p.Description.Contains(queryParams.Search)));

A screenshot of a computer

AI-generated content may be incorrect.

Slow Search query, more then 4 seconds.

Implementing Full Text Search (FTS) in sql server. Some articles:

<https://www.sqlshack.com/hands-full-text-search-sql-server/>

<https://www.red-gate.com/simple-talk/databases/sql-server/learn/sql-server-full-text-search-language-features-part-2/>

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A computer screen with text on it

AI-generated content may be incorrect.A black text on a white background

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

// Add health checks for SQL Server and Redis.

builder.Services.AddHealthChecks()

.AddSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"), "SQL Server")

.AddRedis(builder.Configuration.GetSection("Redis")["Connection"], name: "Redis");

[Health checks in ASP.NET Core | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/health-checks?view=aspnetcore-9.0)

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

# Advanced DB strategies

A screenshot of a computer

AI-generated content may be incorrect.

# Adding replication

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

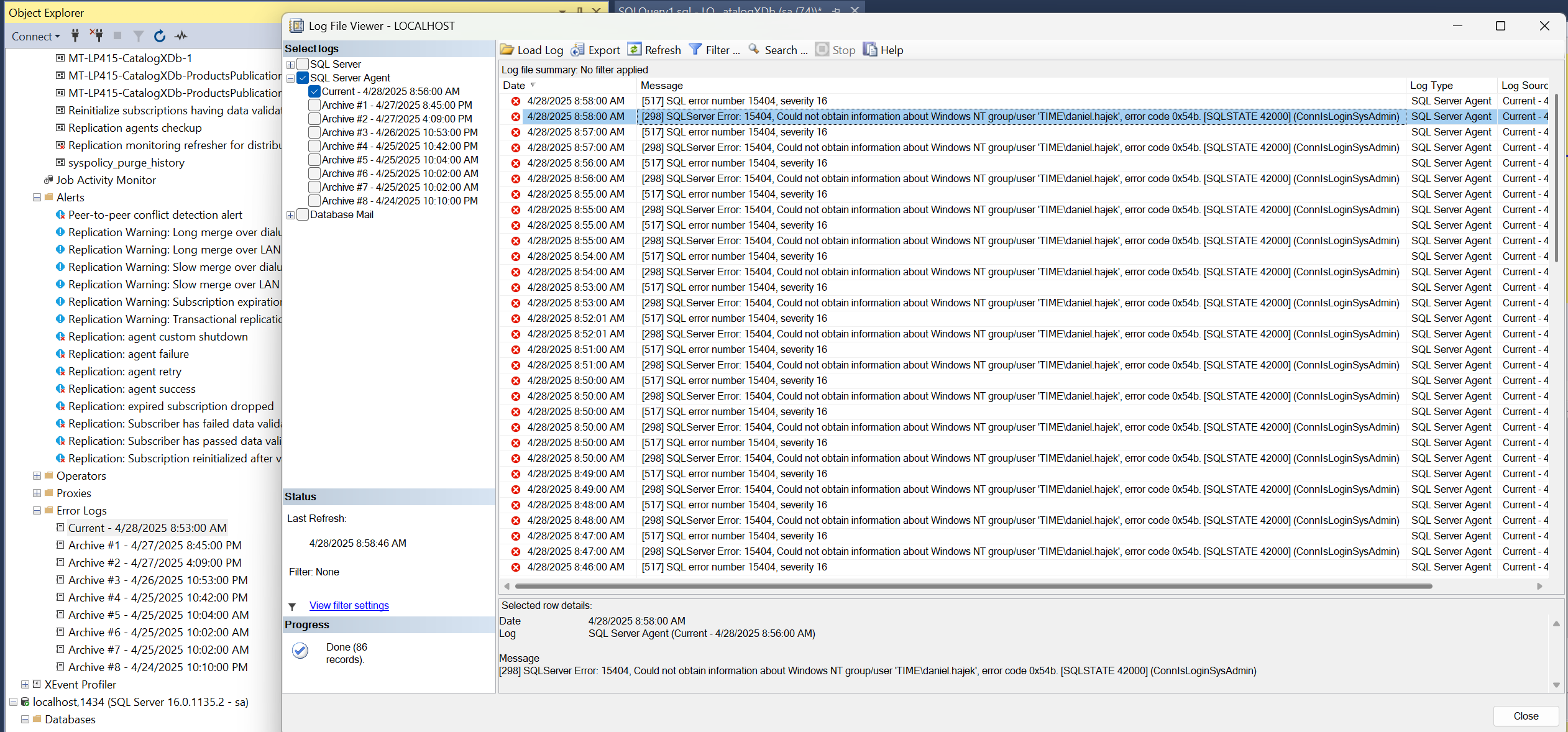
AI-generated content may be incorrect.A screenshot of a computer

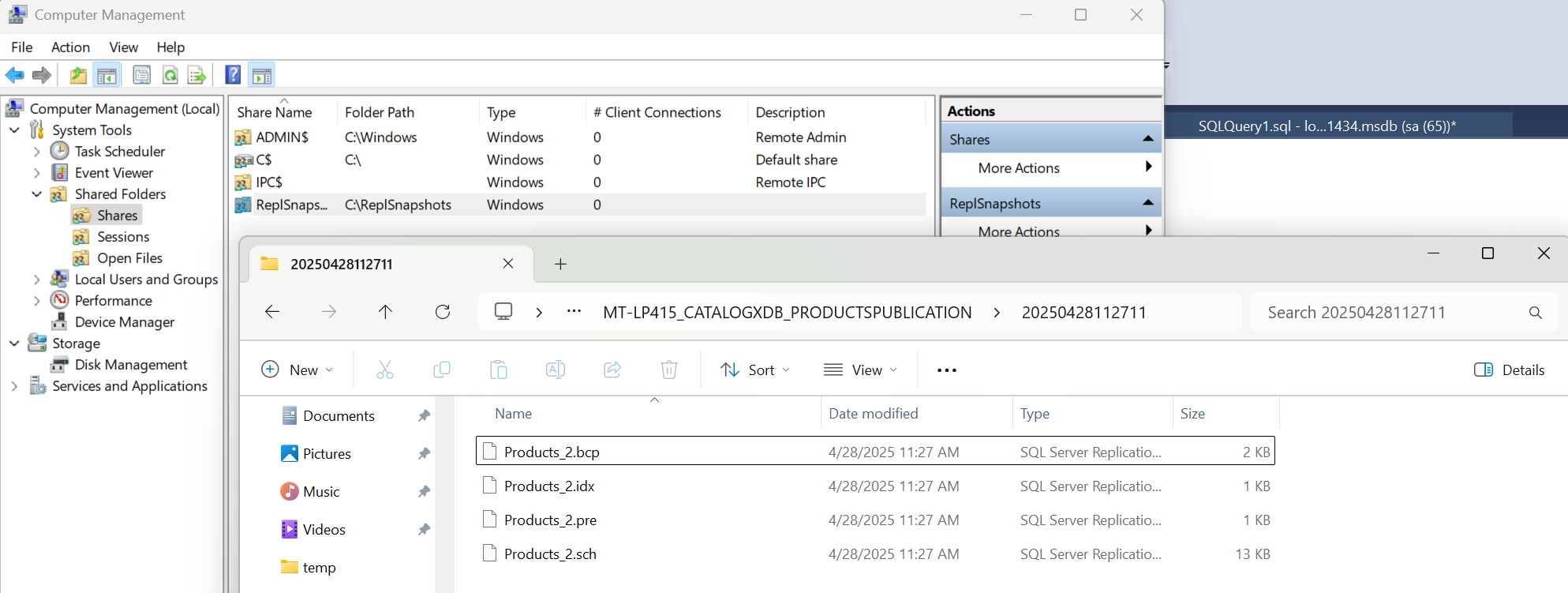
AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.





Post-Mortem step to setup replication on SSMS:

 **Enabled Distribution**

* Configured your own instance (LOCALHOST) as its Distributor (right-click Replication → Configure Distribution).

 **Created the Publication**

* Set up a Transactional Publication on CatalogXDb, adding only the dbo.Products table and enabling schema‐change replication.

 **Created the Push Subscription**

* Pointed it at your subscriber instance (LOCALHOST,1434) and database (CatalogXDb\_Replica), with the initial snapshot to be applied automatically.

 **Switched off Docker/Linux – moved to Windows SQL Server**

* Avoided the “no SQL Server Agent in Linux” limitation by running a Windows-based SQL Server instance with full Agent support.

 **Fixed Snapshot Folder Permissions & UNC Share**

* Created a dedicated folder (e.g. D:\ReplSnapshots)
* Shared it as \\LOCALHOST\ReplSnapshots
* Gave NT SERVICE\MSSQLSERVER and NT SERVICE\SQLSERVERAGENT full NTFS + share rights
* Updated Distributor → Snapshot folder to that UNC path

 **Created a Dedicated SQL Service Login**

* In **master**:

sql

CopyEdit

CREATE LOGIN repl\_agent WITH PASSWORD='…';

ALTER SERVER ROLE sysadmin ADD MEMBER repl\_agent;

* Used this instead of sa or your personal Windows account.

 **Re-assigned All Agent Jobs to repl\_agent**

* In **msdb**, looped through sysjobs to sp\_update\_job ... @owner\_login\_name='repl\_agent'.
* Eliminated all 15404 “unable to query Windows user” errors.

 **Fixed the Proxy on Step 2**

* Removed the old TIME\daniel.hajek proxy and set the replication steps to run under the SQL Agent service account (repl\_agent), eliminating proxy authentication failures.

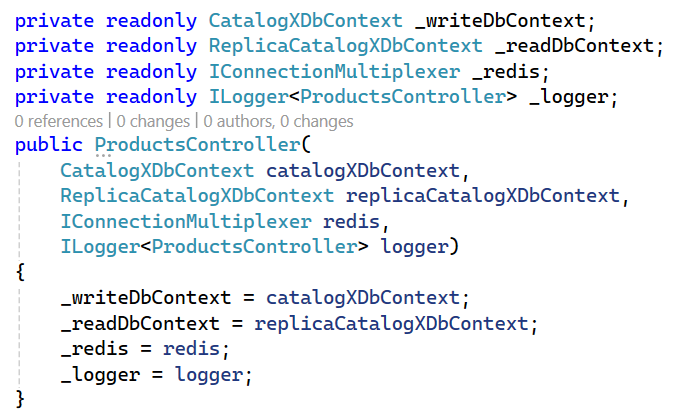
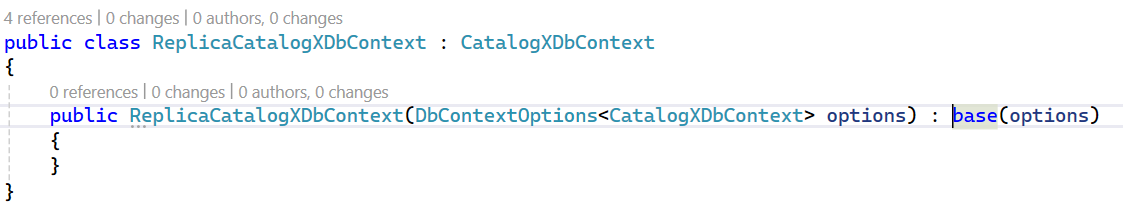
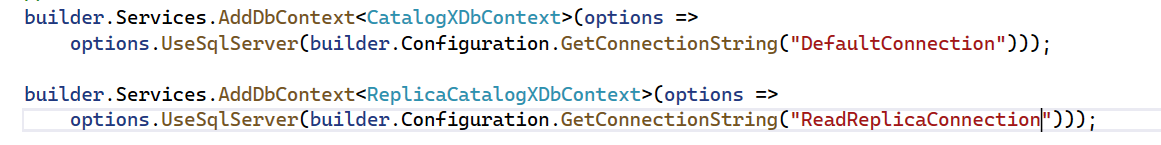
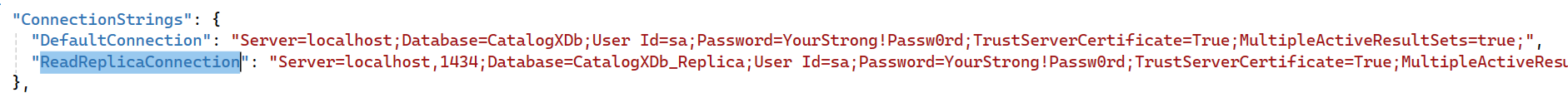
 **Reinitialized Subscriptions & Manually Started Snapshot**

* Right-clicked the publication → Reinitialize All Subscriptions
* Manually kicked off the **Snapshot Agent** job (Step 1) to force snapshot file creation.

 **Verified Snapshot & Distribution**

* Confirmed .bcp and schema scripts appeared under \\LOCALHOST\ReplSnapshots\unc\…
* Watched the **Distribution Agent** apply them to CatalogXDb\_Replica
* Did a test INSERT on Publisher and saw it arrive in the subscriber within seconds

# Adding replica db context



# Testing with k6

