Microsoft Fabric

COMMUNITY CONFERENCE

FabricConf.com | #FABCON

Microsoft Fabric
COMMUNITY CONFERENCE

Using SQL Database Projects

With Visual Studio SQL Server Data Tools

Mike Diehl, Mike.Diehl@Improving.Com
Director of Data Engineering and Business Intelligence

https://github.com/xhead/FabCon2025

/mikediehlsqlbi

Whova

The official event app for the
Microsoft Fabric Community Conference

Join the event app to access:

- Event announcements
- Personalized agenda, session details
- Speaker & attendee profiles
- Networking, meet-ups, messages
- Event documents

Event Invitation Code: FABCON2025

1997 imaginet

improving

18 OFFICES

Can you answer these questions?

- What did this database look like six months ago?
- Can I change the name of a column or table and not break something?
- Are there any broken references in this database?
- Can I deploy this database to a new database?
- Can I reliably upgrade the objects in an existing database?

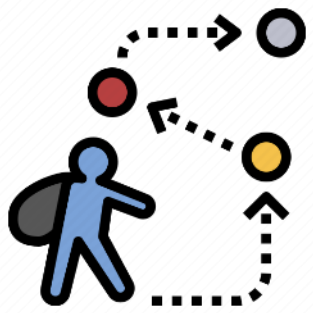
Database Projects

- Visual Studio (Data storage and processing)
- SQL files:
  - CREATE statements
    - Most files are a single CREATE; table scripts can include CREATE INDEX statements and other; separated by GO
  - Other scripts
    - One Pre-Deploy script
    - One Post-Deploy script
    - Call other scripts from these two scripts

Desired State Configuration (DSC)

The source code defines the desired end-state.

At deployment, the actions to move from current state to desired state are determined and executed



Migration-based configuration

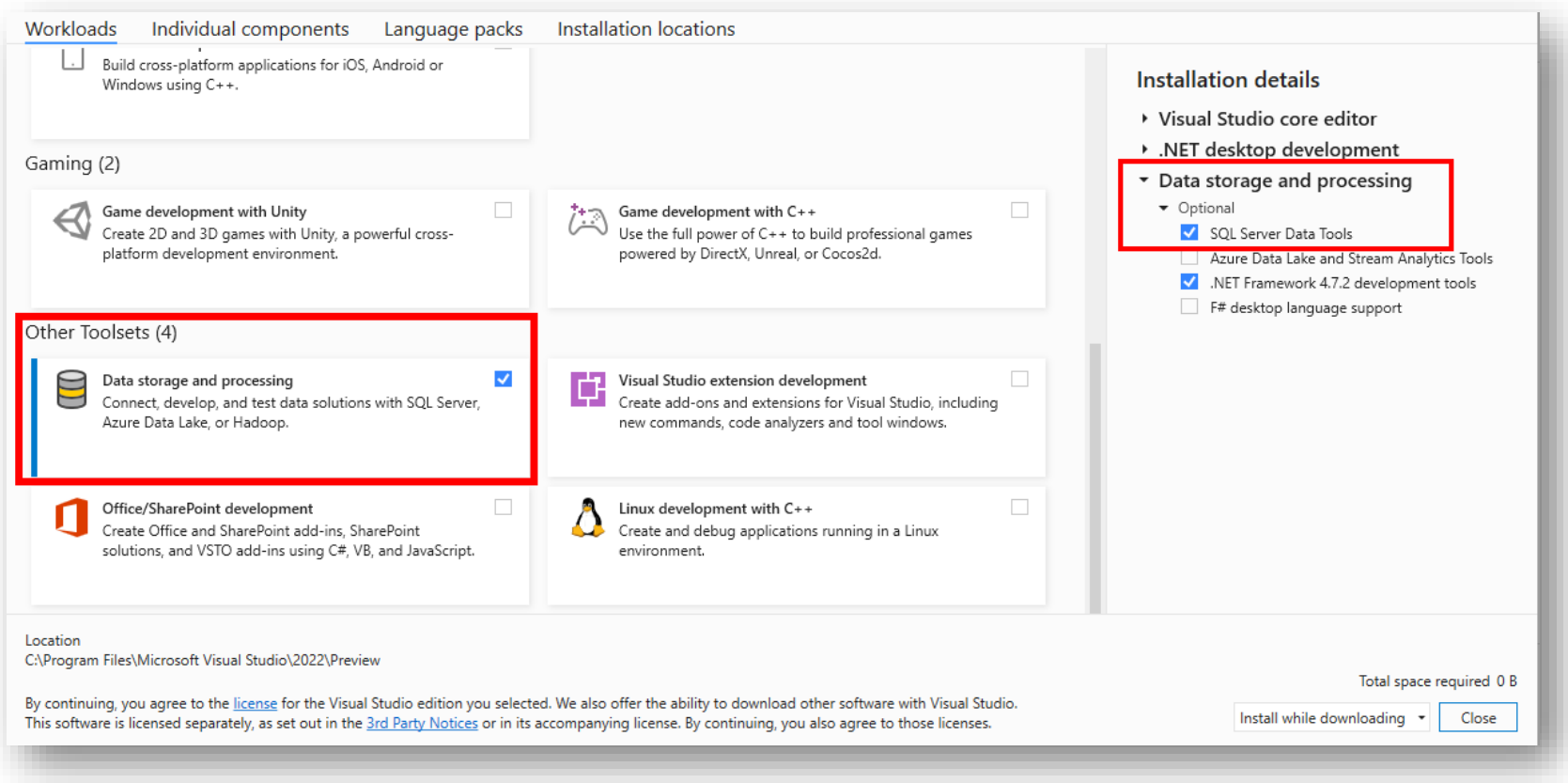
Each migration specifies actions to move from state to state

From A to B

From B to C

Software

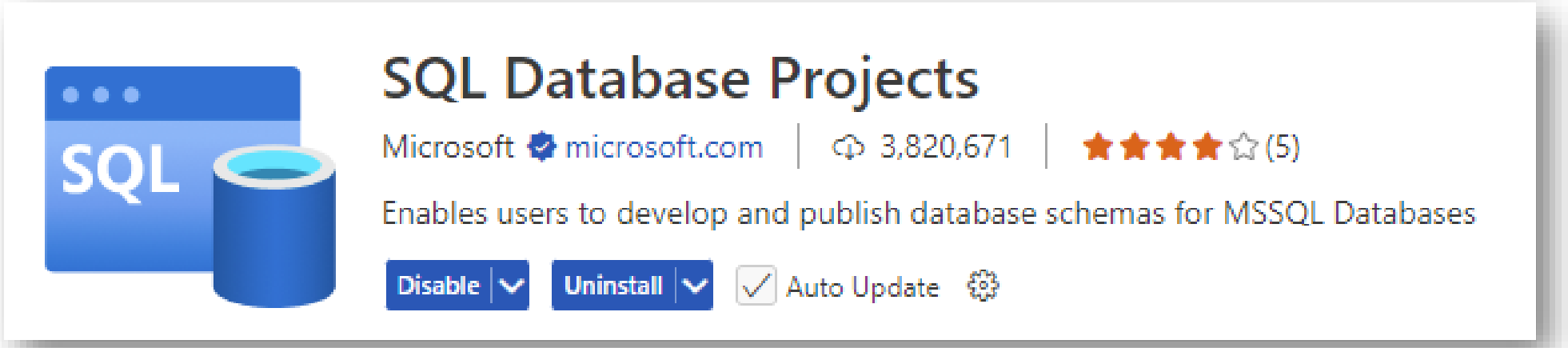
- Visual Studio (Community Edition)
  - Other toolsets
  - Data storage and processing



Software

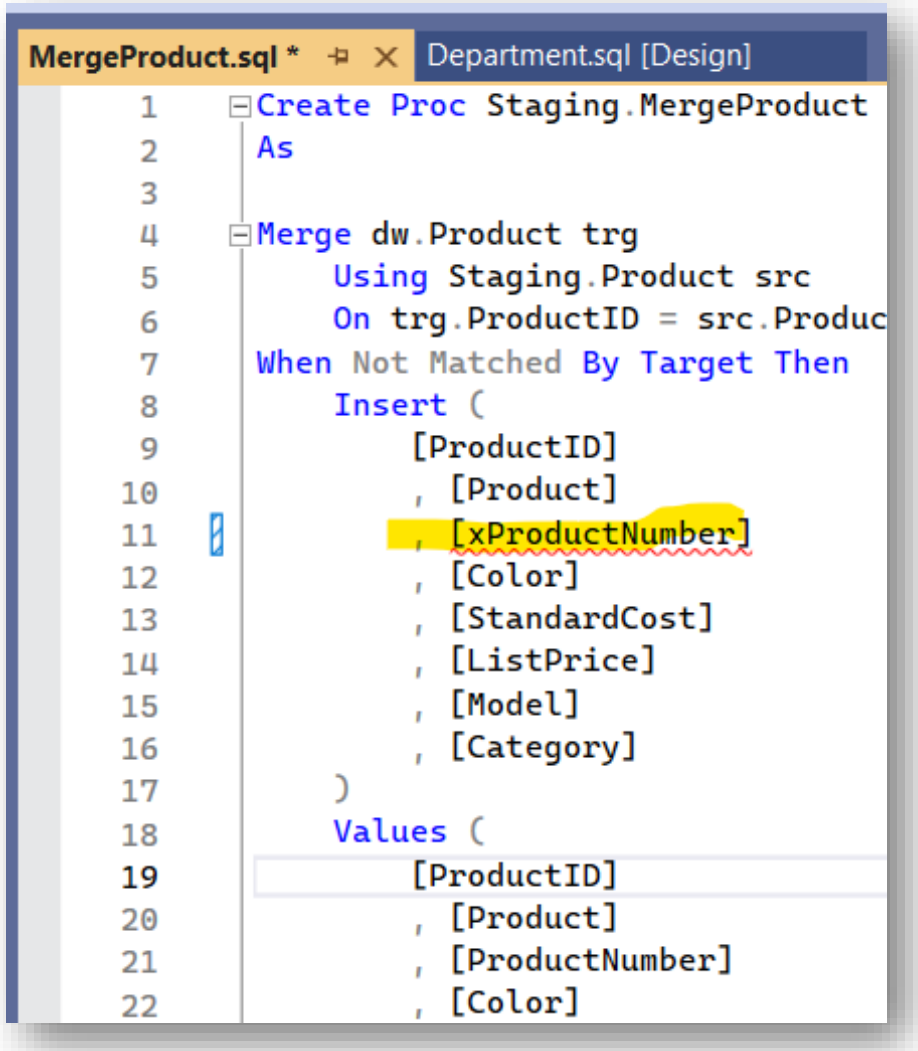
- Visual Studio Code: extension for SQL Database Projects

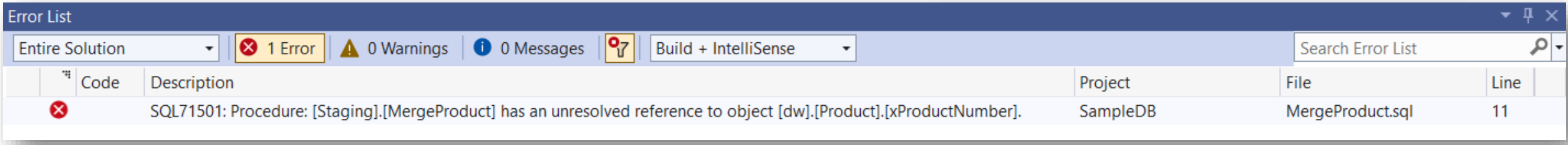
Visual Studio Code



Visual Studio bonus value

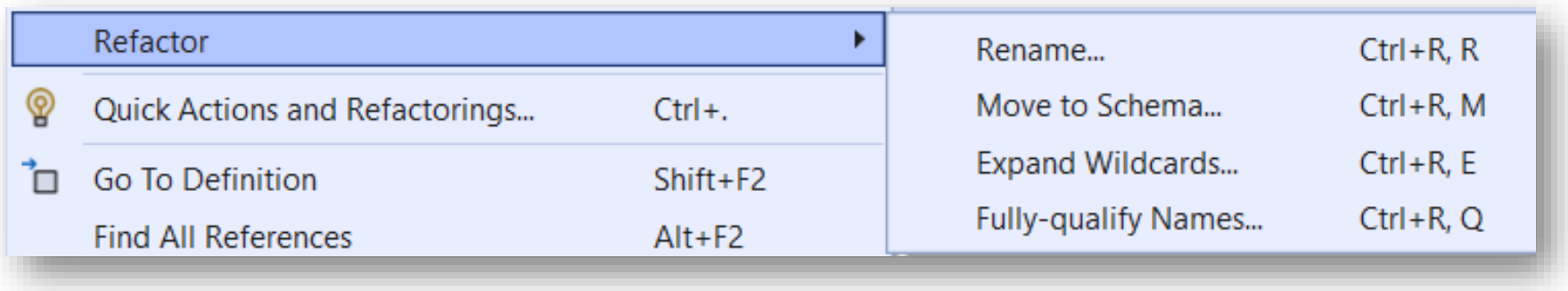
- Build project
  - Detects syntax errors and invalid references
  - Sanity check before attempting to deploy
  - DACPAC file generated





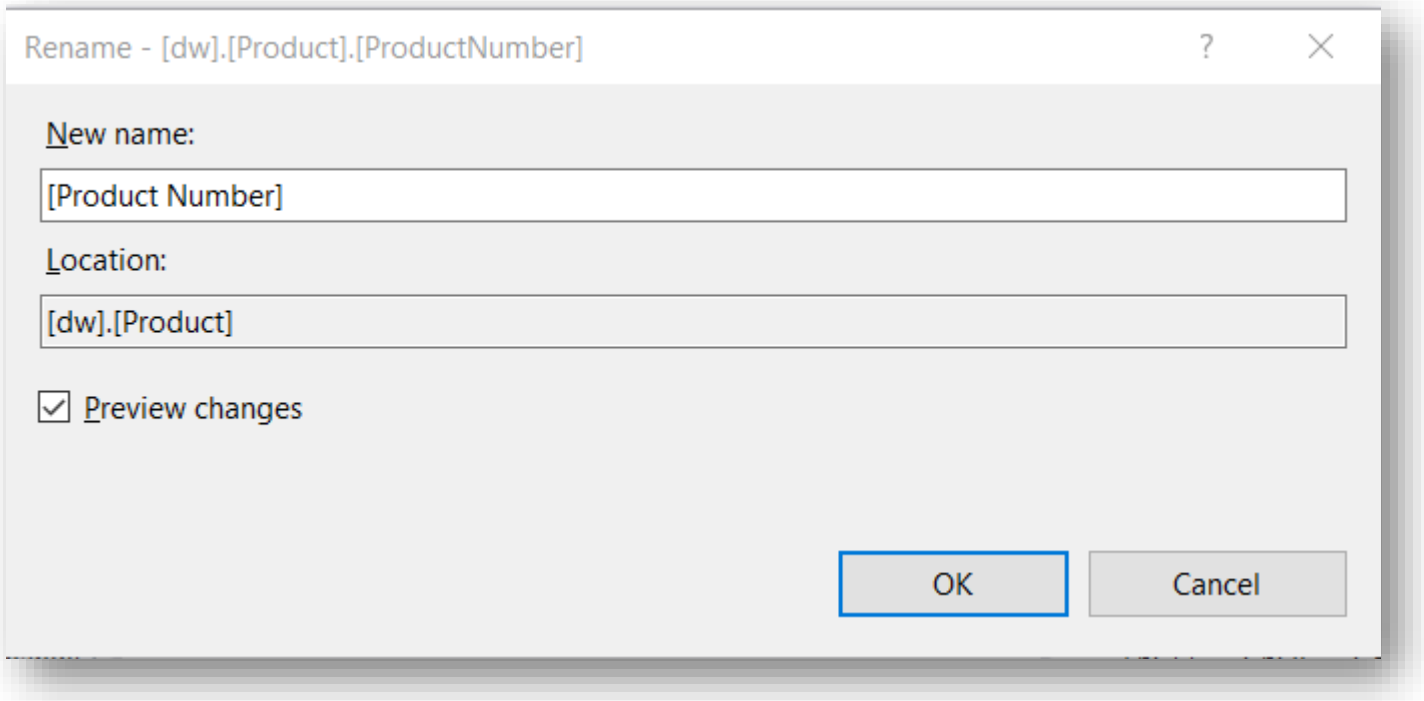
Visual Studio bonus value

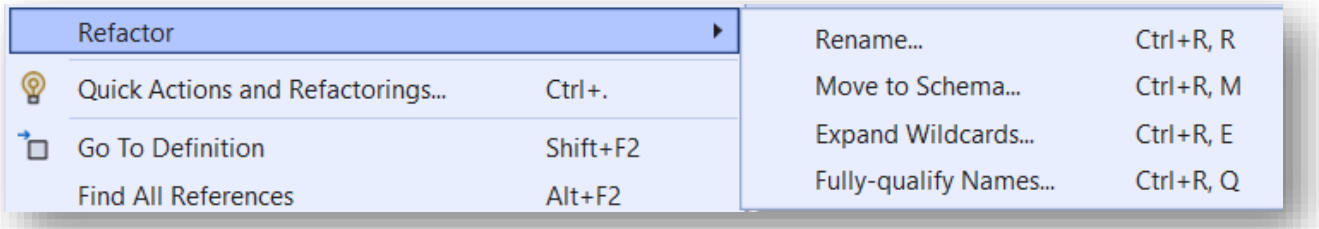
- Context menu
  - Refactor
  - Go To Definition
  - Find All References



Visual Studio bonus value

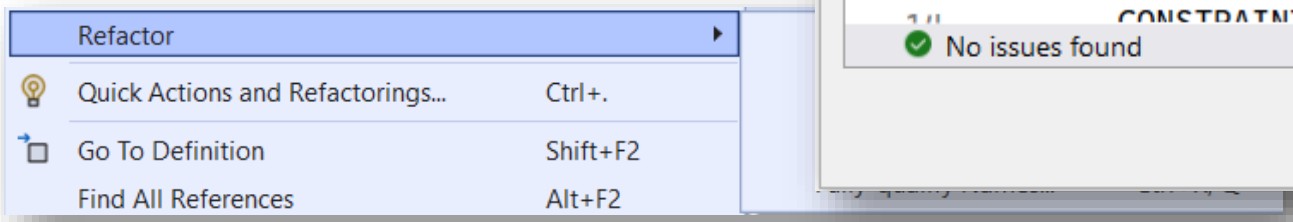
- Context menu
  - Refactor
  - Critical to avoiding data loss on deployment

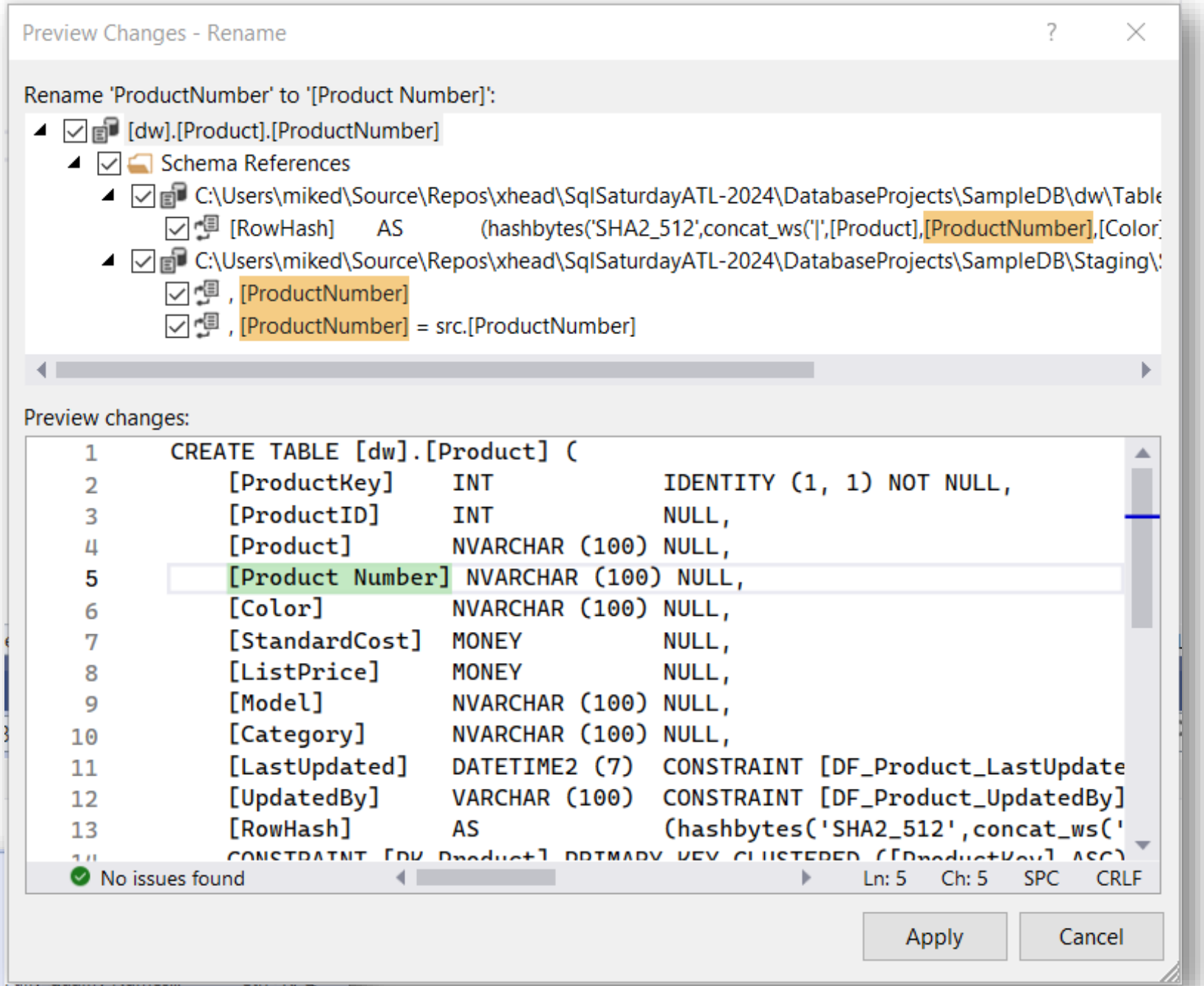




Visual Studio

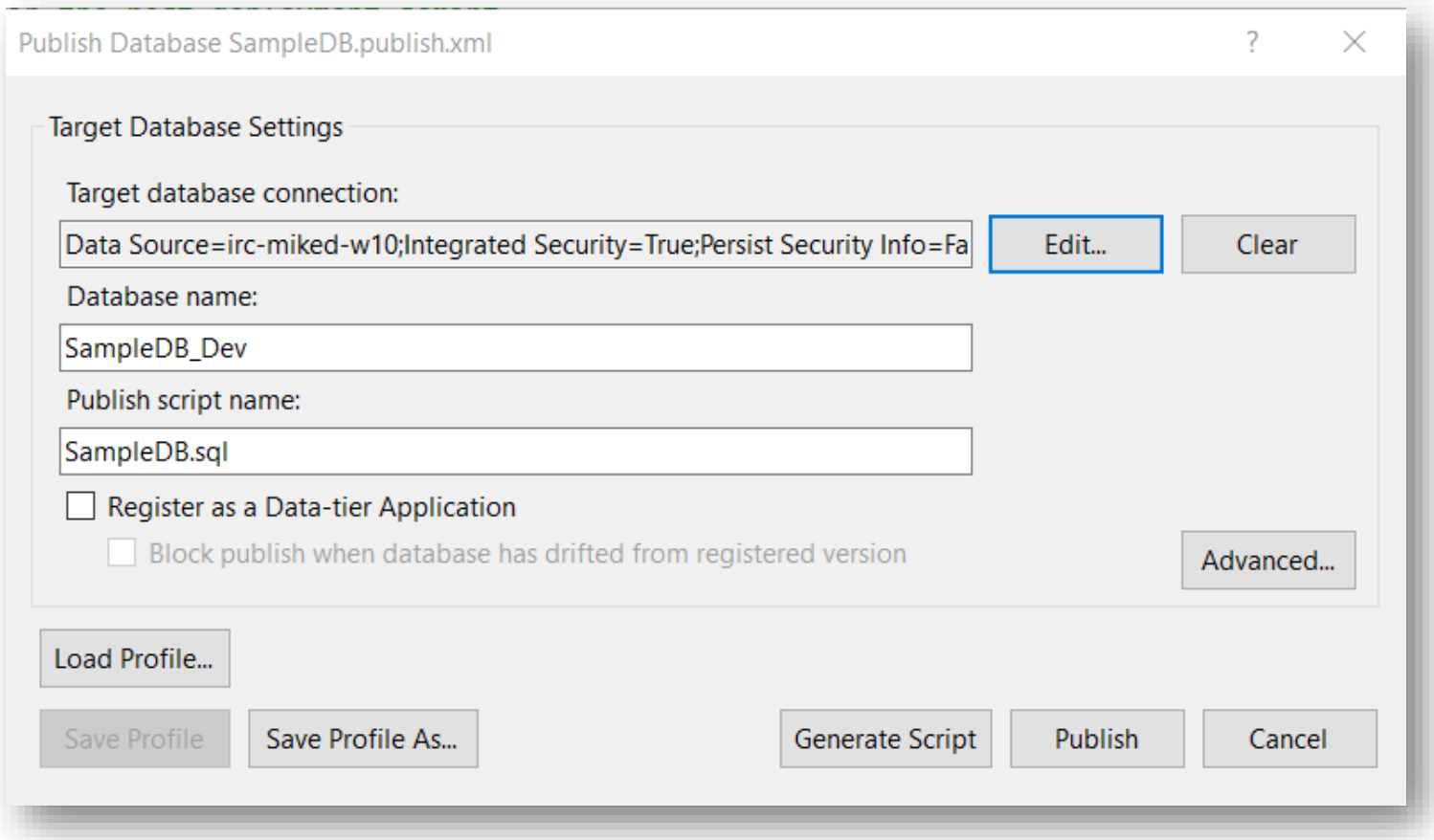
- Context menu
  - Refactor
  - Critical to avoiding data loss on deployment
  - Sp_rename vs ALTER TABLE DROP/ADD
- Refactoring tracked in SampleDB.refactorlog





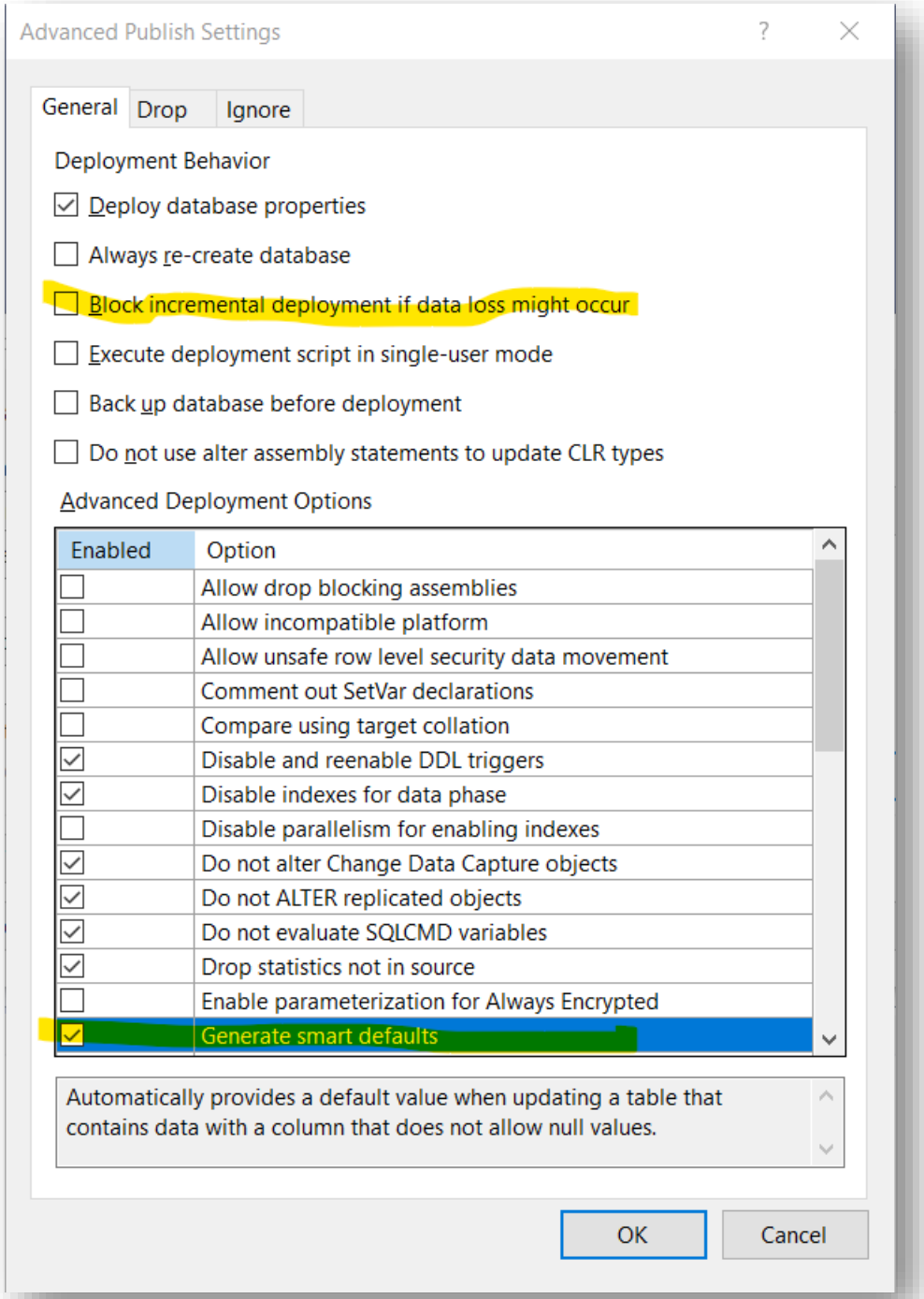
Publish from Visual Studio

- Rt-click, Deploy
  - Create script
  - Publish directly
- Publish profiles (*.publish.xml)
- Critical settings in publish profile
  - Use Smart Defaults
  - Ignore column order



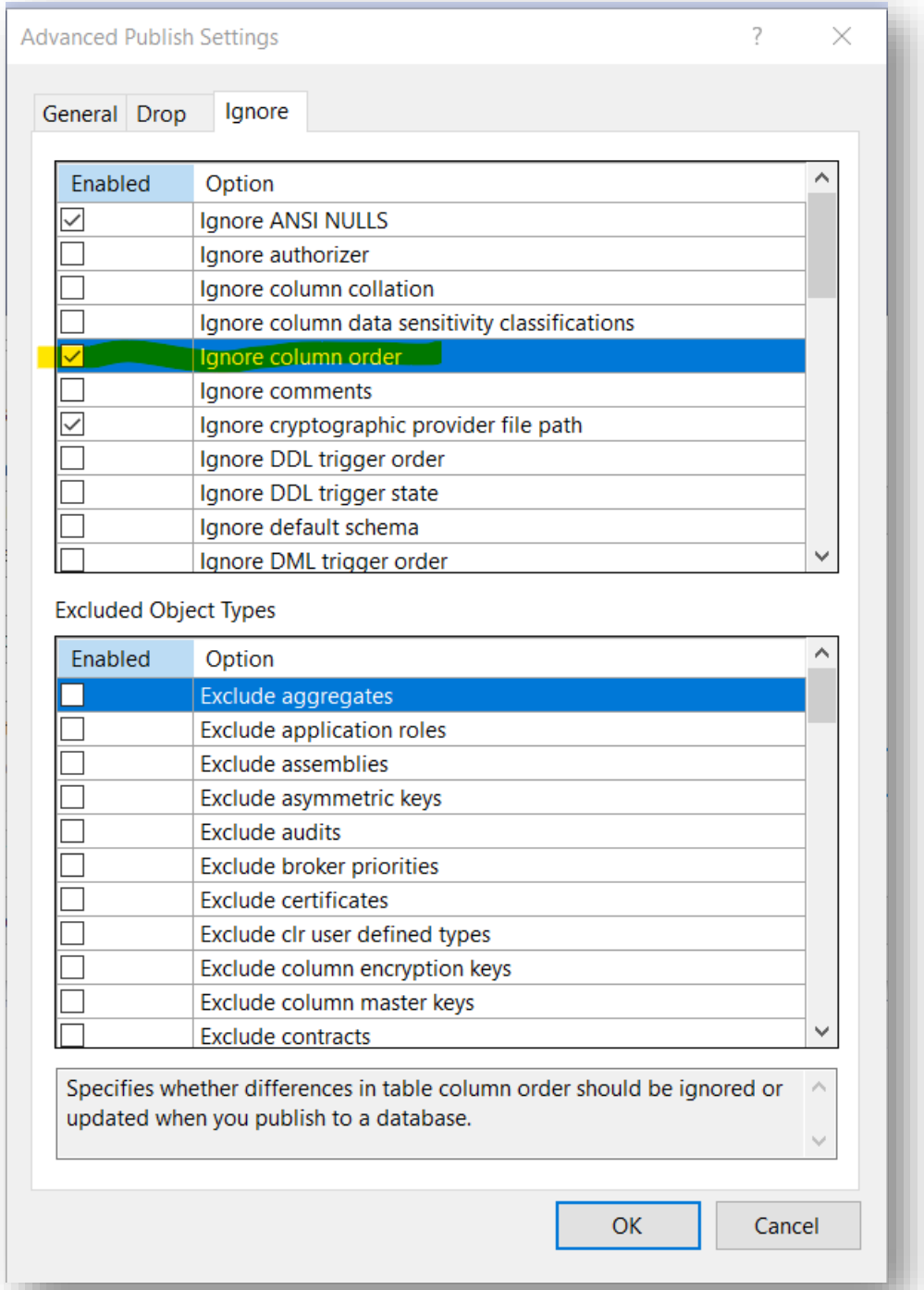
Publish from Visual Studio

- Rt-click, Deploy
  - Create script
  - Publish directly
- Publish profiles (*.publish.xml)
- Critical settings in publish profile
  - Use Smart Defaults
  - Ignore column order



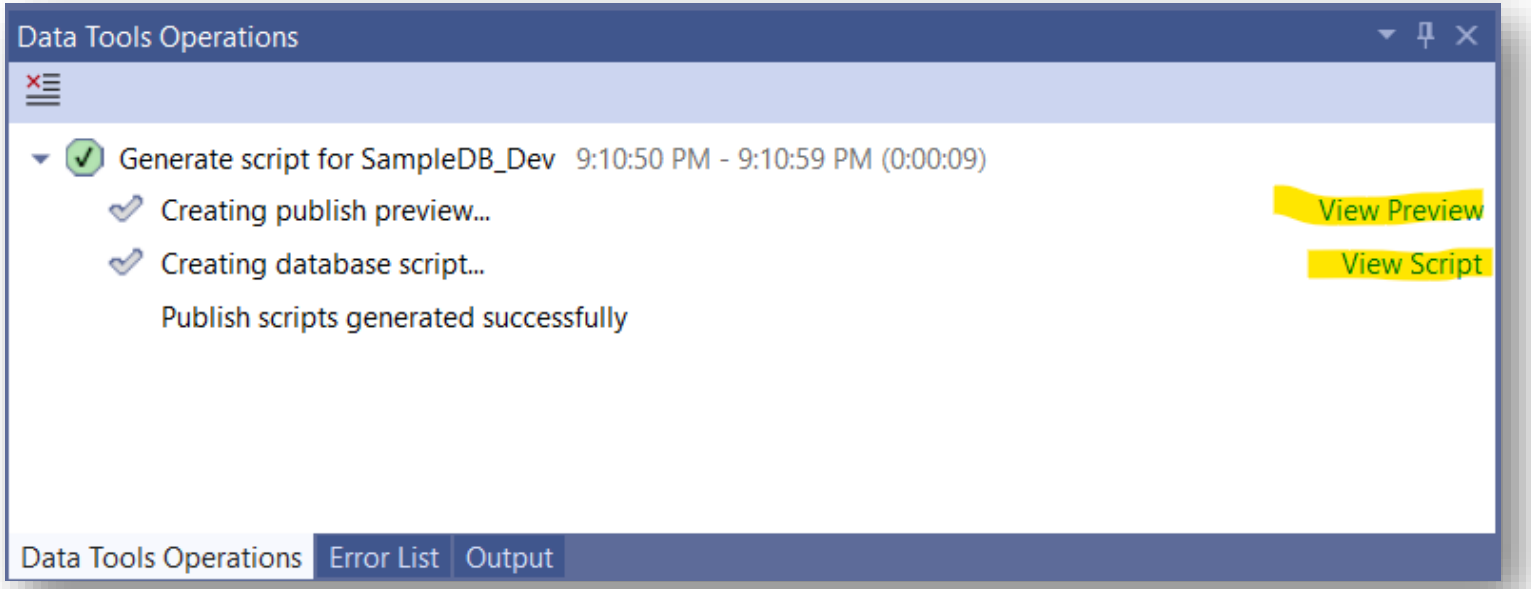
Publish from Visual Studio

- Rt-click, Deploy
  - Create script
  - Publish directly
- Publish profiles (*.publish.xml)
- Critical settings in publish profile
  - Use Smart Defaults
  - Ignore column order



Publish from Visual Studio

- Rt-click, Deploy
  - Create script
  - Publish directly
- Preview contains summary of actions and potential warnings
  - Avoid table rebuilds



SQLPACKAGE.EXE

- Command line utility for deploying DACPAC files to SQL database
- Uses DACPAC file and publish.xml file
- Command line options
  - Generate script
  - Deploy changes
- Used by Azure DevOps deployment tasks or other CI/CD processes

Pre-/Post-deploy scripts

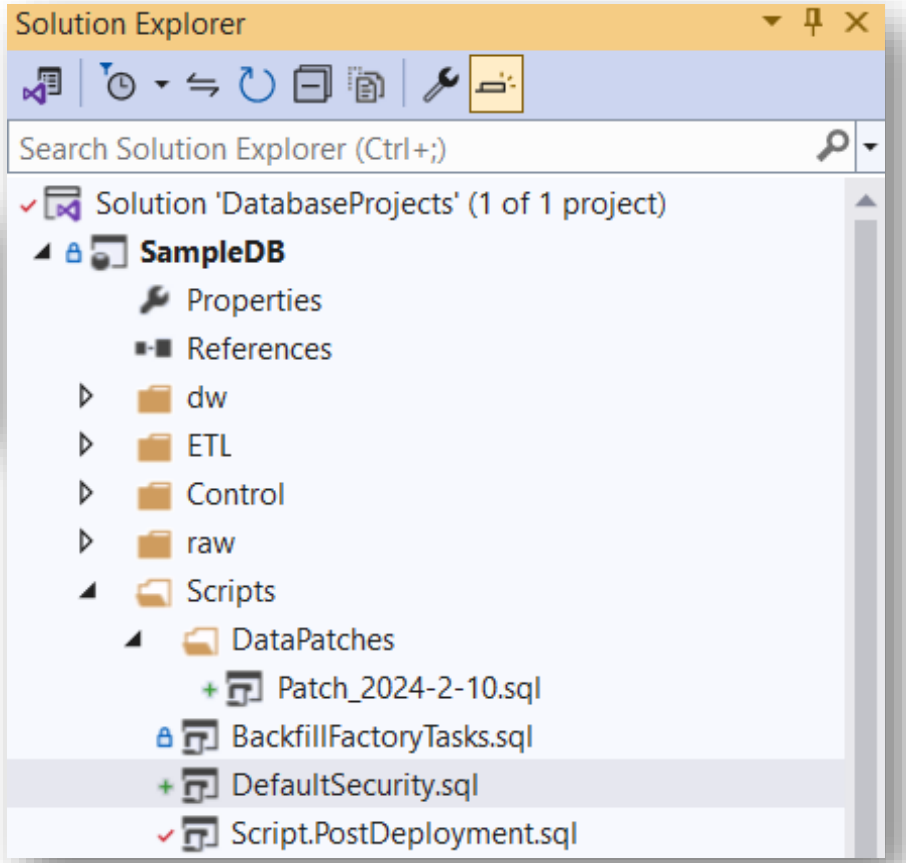
- Only one per project
  - :r .\DefaultSecurity.sql
  - :r .\BackfillReferenceData.sql
  - :r .\Patch_2024-2-10.sql
- Scripts must be idempotent
  - Run one or more times without affecting the desired outcome

SQLCMD syntax

- :r – run file
- :setVar – set variable
- $(variables)

Default security

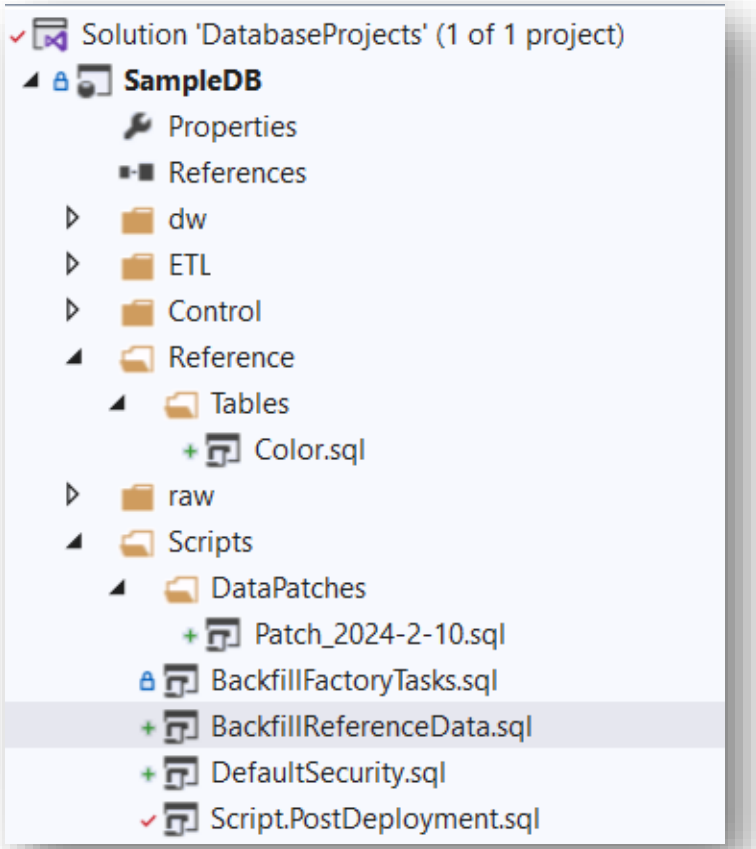
```
If Not Exists(Select * From sys.sysusers Where name = 'StandardLogin')
Begin
    Create User StandardLogin For Login StandardLogin
End
Alter Role SampleRole Add Member StandardLogin
```



Backfilling Reference Data

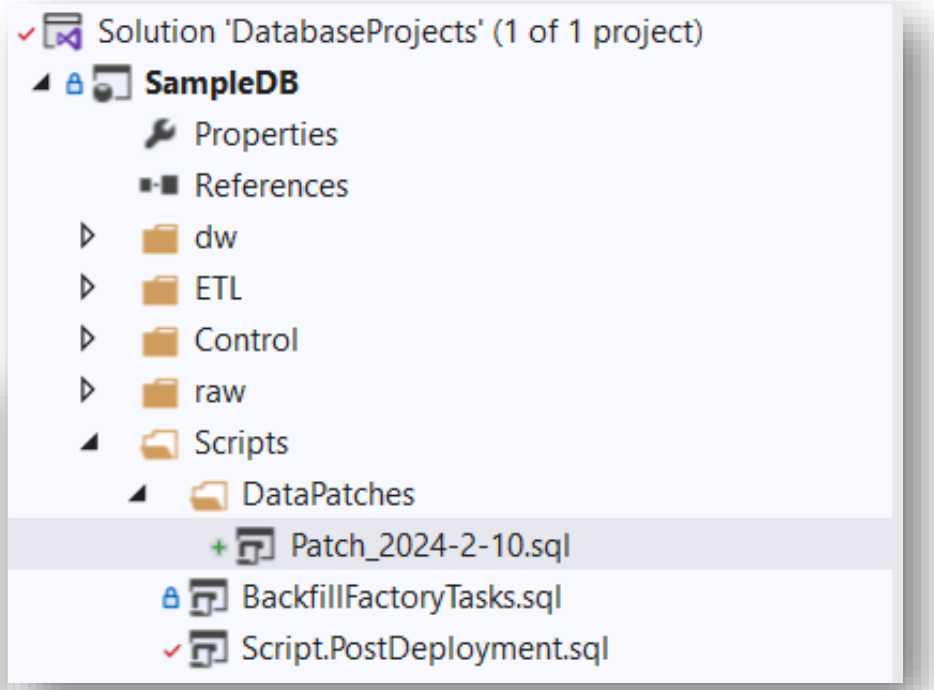
```sql
Print 'Backfilling Reference.Color';

With colors As (
    Select * From (Values (
    ('Blue'),
    ('Red'),
    ('Orange'),
    ('Green'),
    ('Black')
    )) x (Color)
)
Merge Into Reference.Color trg
Using colors src On trg.Color = src.Color

When Not Matched By Target Then
    Insert (Color) Values (src.Color)

When Not Matched By Source Then
    Delete
;
Print 'Complete'
```

```sql
/*
Post-Deployment Script Template
--------------------------------------
 This file contains SQL statements that
 Use SQLCMD syntax to include a file in
 Example:      :r .\myfile.sql
 Use SQLCMD syntax to reference a varia
 Example:      :setvar TableName MyTabl
               SELECT * FROM [$(TableNa
--------------------------------------
*/
Go
:r .\BackfillFactoryTasks.sql
Go
:r .\BackfillReferenceData.sql
Go
:r .\Datapatches\Patch_2024-2-10.sql
Go
```



Executing Data Patches

```
:SetVar Patchname Patch_2024-2-10

If Not Exists(Select * From Control.ChangeTracking Where Change = '$(PatchName)')
Begin

    Print 'Patching - $(PatchName)'

    -- do something
    -- delete from dw.MyTable where BadData = 1

    Insert Into Control.ChangeTracking (Change, AppliedOn)
    Values ('$(patchName)', GetDate())

    Print 'Patch complete.'

End
```




```
/*
Post-Deployment Script Template
--------------------------------------------------------------------------------------
 This file contains SQL statements that will be appended to the build script.
 Use SQLCMD syntax to include a file in the post-deployment script.
 Example:      :r .\myfile.sql
 Use SQLCMD syntax to reference a variable in the post-deployment script.
 Example:      :setvar TableName MyTable
               SELECT * FROM [$(TableName)]
--------------------------------------------------------------------------------------
*/
Go
:r .\BackfillFactoryTasks.sql
Go
:r .\Datapatches\Patch_2024-2-10.sql
Go
```

Importing from an existing database

- Create project then do Schema Compare
- Databases *should* be self-contained
- Often exposes errors/problems
  - Missing objects, bad names
  - Three-part and four-part names
- Replace external references with CREATE SYNONYM and $(variables)

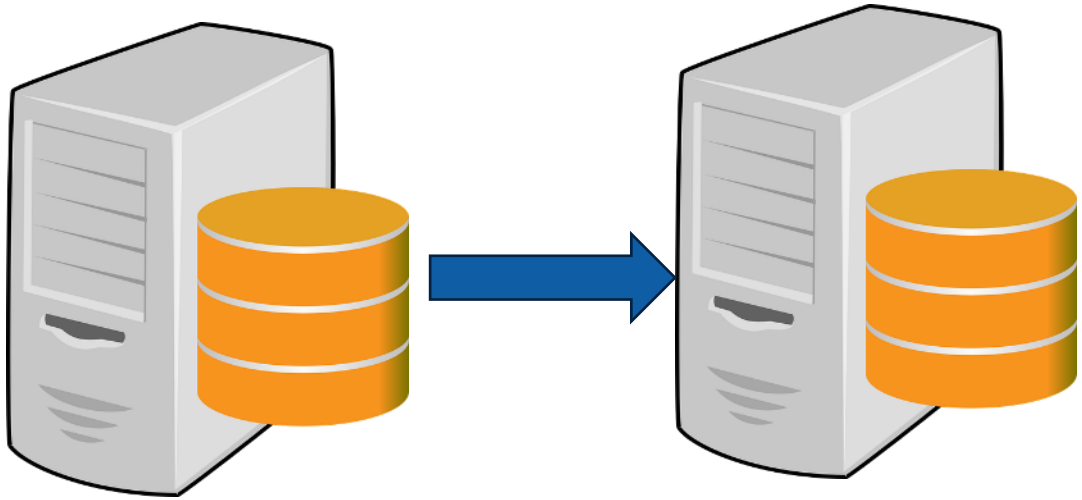
https://www.pabloreinoso.com/

Database references

- What is in or out of your control? What can you deploy?
- System databases
  - master, msdb
- Other database projects
- DACPAC files (binary reference)

Database references

- What is in or out of your control? What can you deploy?
- System databases
  - master, msdb
- Other database projects
- DACPAC files (binary reference)
- Target:
  - Different server, different database



Server 2/Project B

Server 1/Project A
Select * from [$(ServerX)].[$(DatabaseB)].dbo.Customer

Database references

- What is in or out of your control? What can you deploy?
- System databases
  - master, msdb
- Other database projects
- DACPAC files (binary reference)
- Target:
  - Different server, different database
  - Same server, different database

Project B

Project A
Select * from [$(DatabaseB)].dbo.Customer

Database references

- What is in or out of your control? What can you deploy?
- System databases
  - master, msdb
- Other database projects
- DACPAC files (binary reference)
- Target:
  - Different server, different database
  - Same server, different database
  - Same server, same database

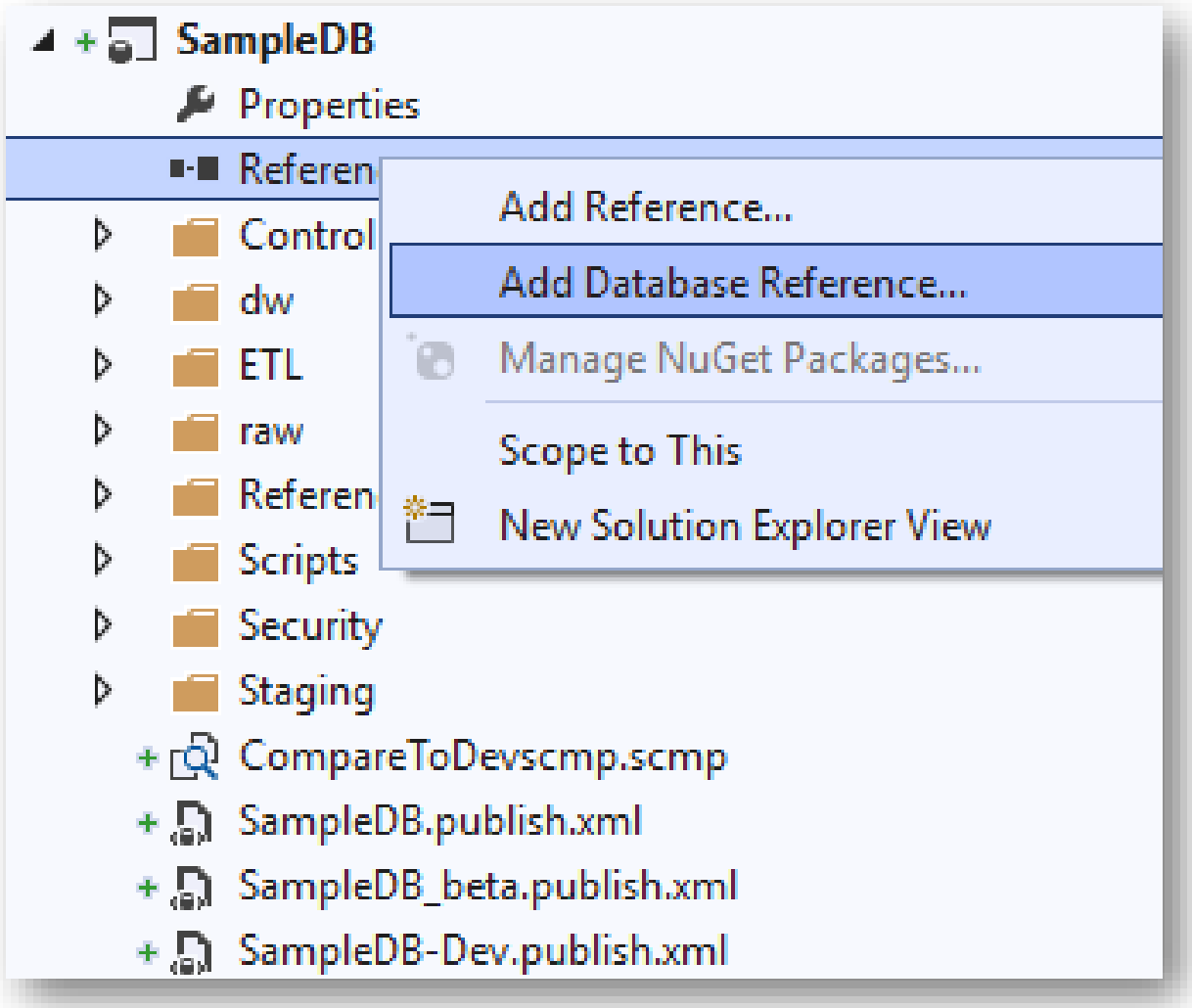
Project A

Project B

Select * from dbo.Customer

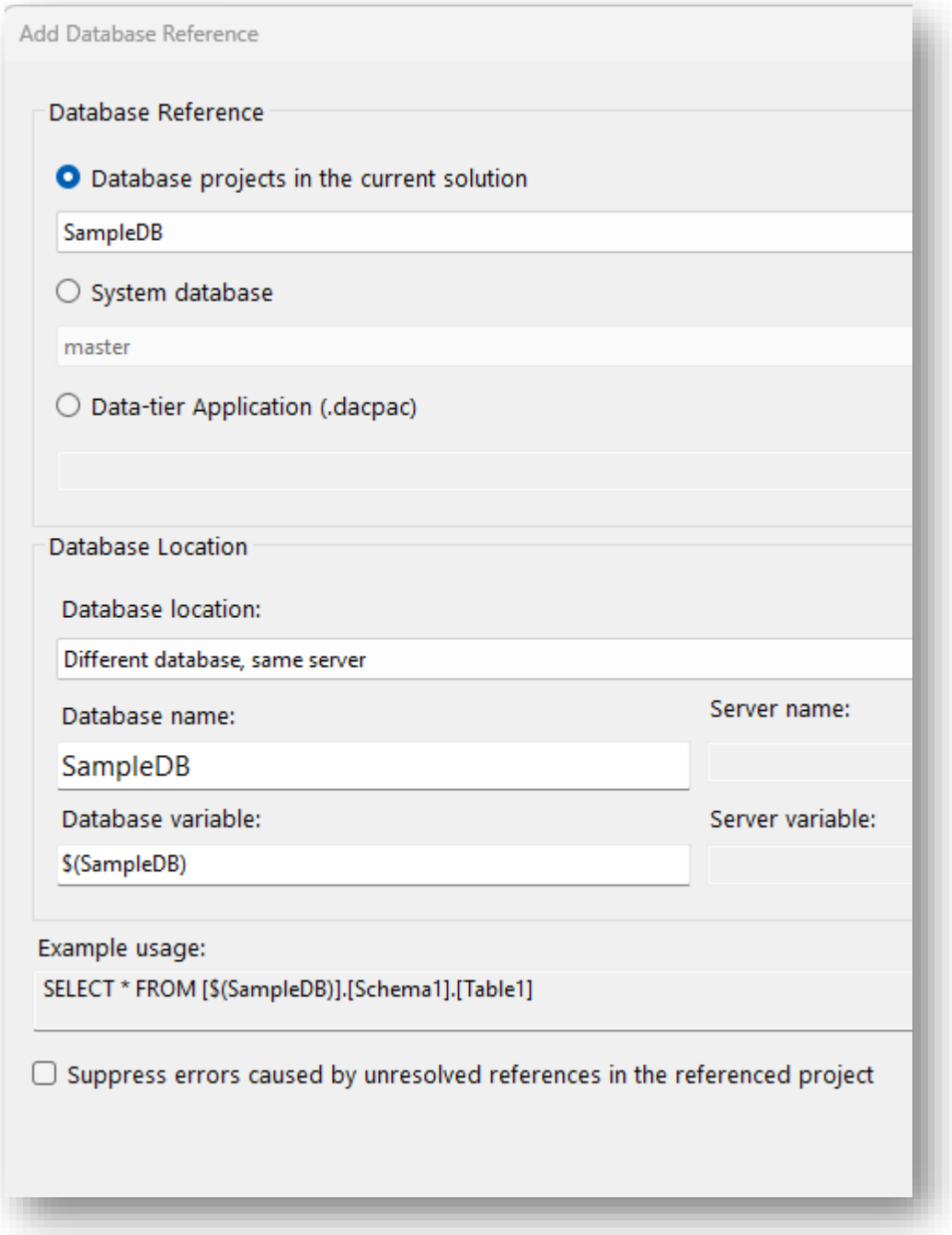
Database references

- What is in or out of your control? What can you deploy?
- System databases
  - master, msdb
- Other database projects
- DACPAC files (binary reference)



Database references

- What is in or out of your control? What can you deploy?
- System databases
  - master, msdb
- Other database projects
- DACPAC files (binary reference)

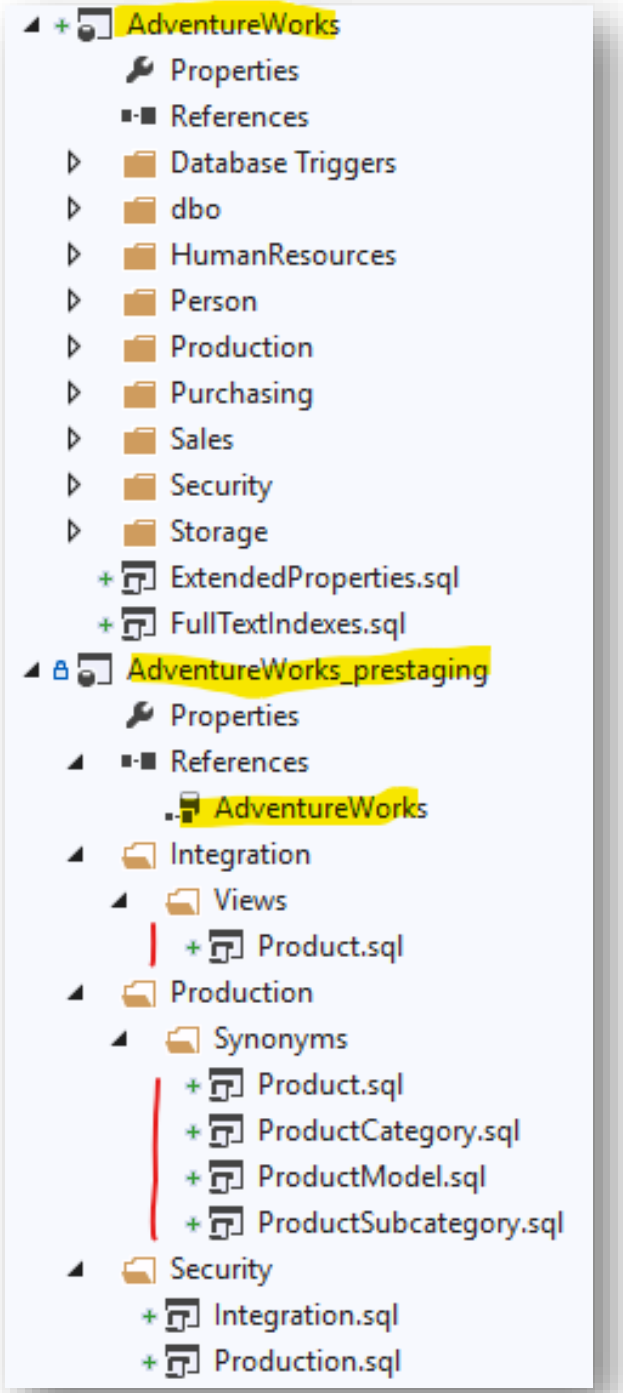


Database reference scenarios

- Cross-reference to another database outside of your control
- Same or different server, different database (dependency)
- CREATE SYNONYM to specific tables, views or sprocs

```
CREATE SYNONYM [Production].[Product]
FOR [$(AdventureWorks)].[Production].[Product];
```

- Deploy only our own objects in our own database
- Schema compare on original database



Database reference scenarios

- Cross-reference to another database outside of your control
- Same or different server, different database (dependency)
- Use schemas for clarity
- Synonym provides API to referenced database

```
CREATE SYNONYM [Production].[Product]
FOR [$(AdventureWorks)].[Production].
[Product];
```

- Source query works in both databases

```
create view Integration.Product as
select
    p.ProductID,
    p.ProductNumber,
    p.Name as ProductName,
    p.ListPrice,
    p.StandardCost,
    p.Color,
    p.Weight,
    p.Size,
    pc.Name as ProductCategory,
    ps.Name as ProductSubCategory,
    pm.Name as ProductModel
from Production.Product p
    join Production.ProductSubcategory ps
        on p.ProductSubcategoryID = ps.ProductSubcategoryID
    join Production.ProductCategory pc
        on ps.ProductCategoryID = pc.ProductCategoryID
    join Production.ProductModel pm
        on p.ProductModelID = pm.ProductModelID
```

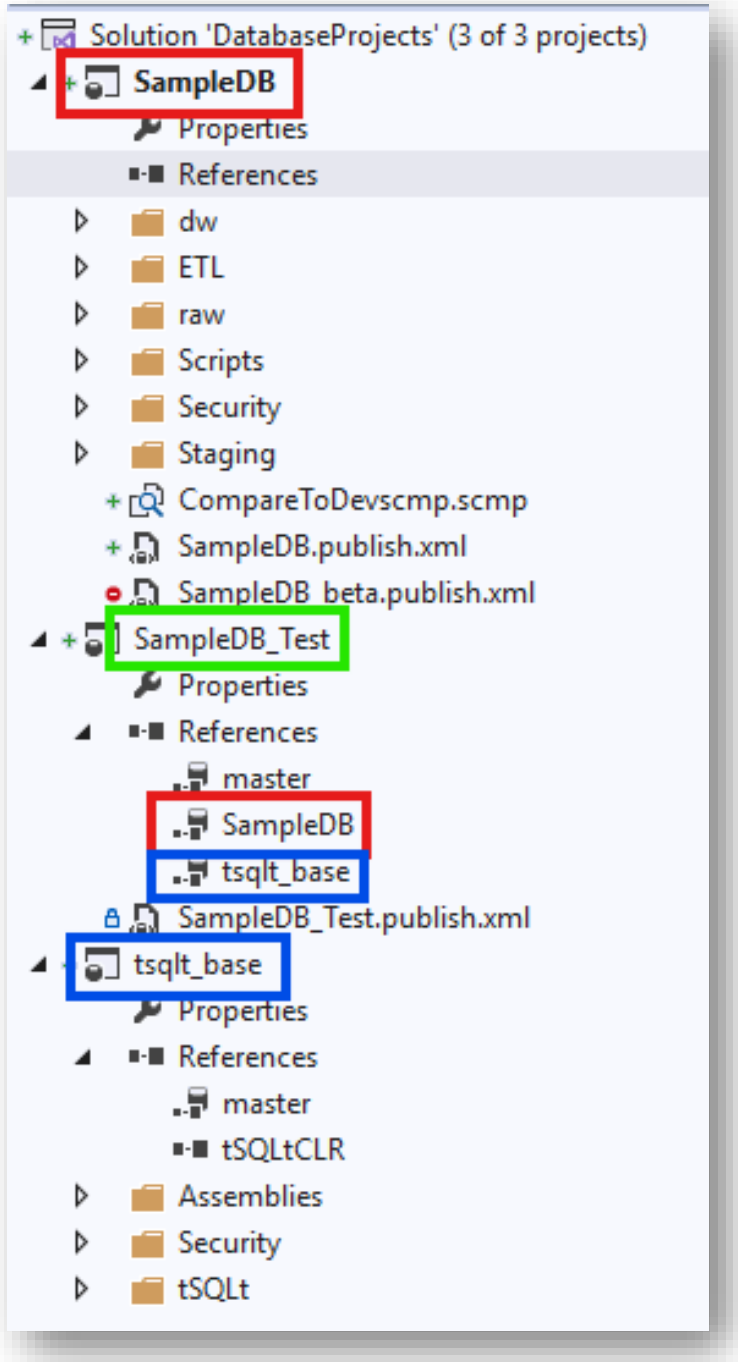
Database reference scenarios

- Cross-reference to another database outside of your control
- Same server, same database (inheritance)
  - No synonyms necessary
  - Deploy our own objects in the referenced database

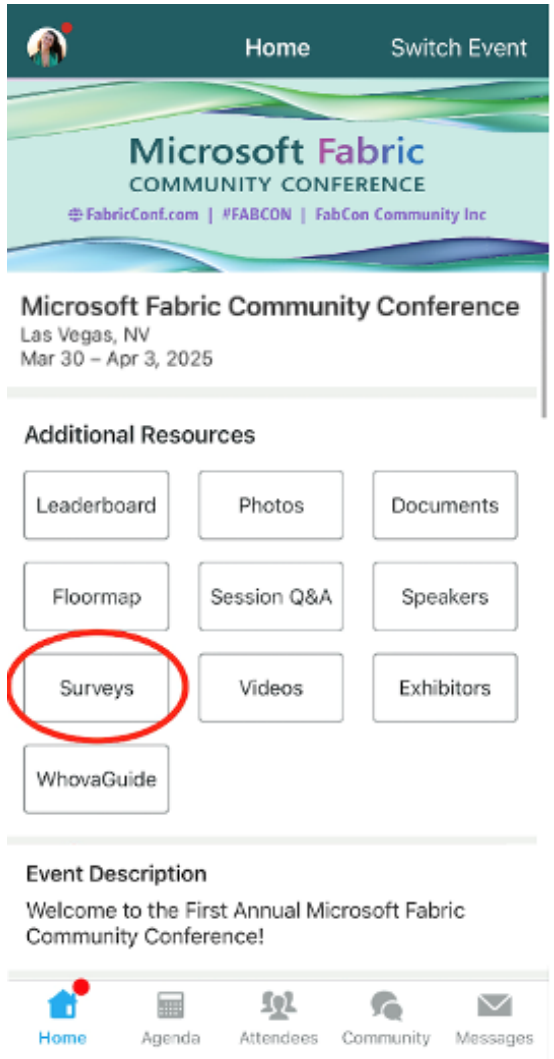
  - Don't clobber other objects in the target database
  - Our own objects could get clobbered by another deployment

Database reference scenario – Unit Testing

- Database project (SUT - system under test) SampleDB
- Unit test framework project contains framework objects (TDD) tsqlt_base
- Test database project SampleDB_Test
  - References database project under test and unit test framework project
    - Both references are “same server, same database”
  - Contains test objects (sprocs)
- Deploy test project and execute unit tests in unit test environment or part of a build
  - SUT database and TDD framework objects get deployed
  - Execute unit tests for pass/fail
- Deploy SUT normally in production environments
  - Clean deploy - no TDD or unit tests included



Session Feedback Surveys



We really want to hear from YOU!

In the pursuit of making next year's Microsoft Fabric Community Conference even better, we want to hear your feedback about this session.

Here's How -

- *Simply go to the Whova App on your smartphone*
- *Scroll down on the Microsoft Fabric Community Conference Homepage to 'Additional Resources' to click "Surveys'.*
- *Click Session Feedback.*
- *Scroll down to find this session title.*
- *Complete the session feedback survey.*
- *Finally, click 'Submit'*

It's just that easy!

https://github.com/xhead/FabCon2025

/mikediehlsqlbi

Mike.Diehl@Improving.Com

Microsoft Fabric
COMMUNITY CONFERENCE

UPCOMING 2025 EVENTS

NextGen 365 Events *presents*
Microsoft 365
COMMUNITY CONFERENCE

MAY 6-8, 2025 MGM GRAND
WORKSHOPS: MAY 4, 5 & 9 LAS VEGAS, NV
NextGen365Events.com

---

next GenAI
CONFERENCE

OCT 7-9, 2025 LOWES SAPPHIRE FALLS RESORT
WORKSHOPS: OCT 5, 6 & 10 ORLANDO, FL
NextGenAIconf.com

---

empOWER everyone
Power Platform
COMMUNITY CONFERENCE

OCT 28-30, 2025 MGM GRAND
WORKSHOPS: OCT 26, 27 & 31 LAS VEGAS, NV
PowerPlatformConf.com