



# Microsoft Fabric

## COMMUNITY CONFERENCE

 [FabricConf.com](https://FabricConf.com) | [#FABCON](https://twitter.com/FABCON)

# Using SQL Database Projects

## With Visual Studio SQL Server Data Tools



Mike Diehl, [Mike.Diehl@Improving.Com](mailto:Mike.Diehl@Improving.Com)  
Director of Data Engineering and Business Intelligence

<https://github.com/xhead/FabCon2025>



/mikediehlsqldb



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

imagninet



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

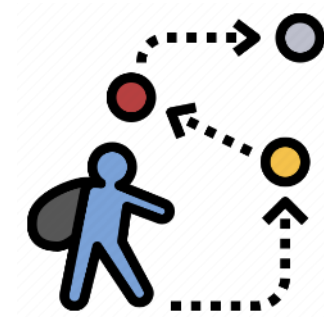
- Visual Studio (Data storage and processing)
- SQL files:
  - CREATE statements
    - Most files are a single CREATE; table scripts can include CREATE INDEX statements and other; separated by GO
  - Other scripts
    - One Pre-Deploy script
    - One Post-Deploy script
    - Call other scripts from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

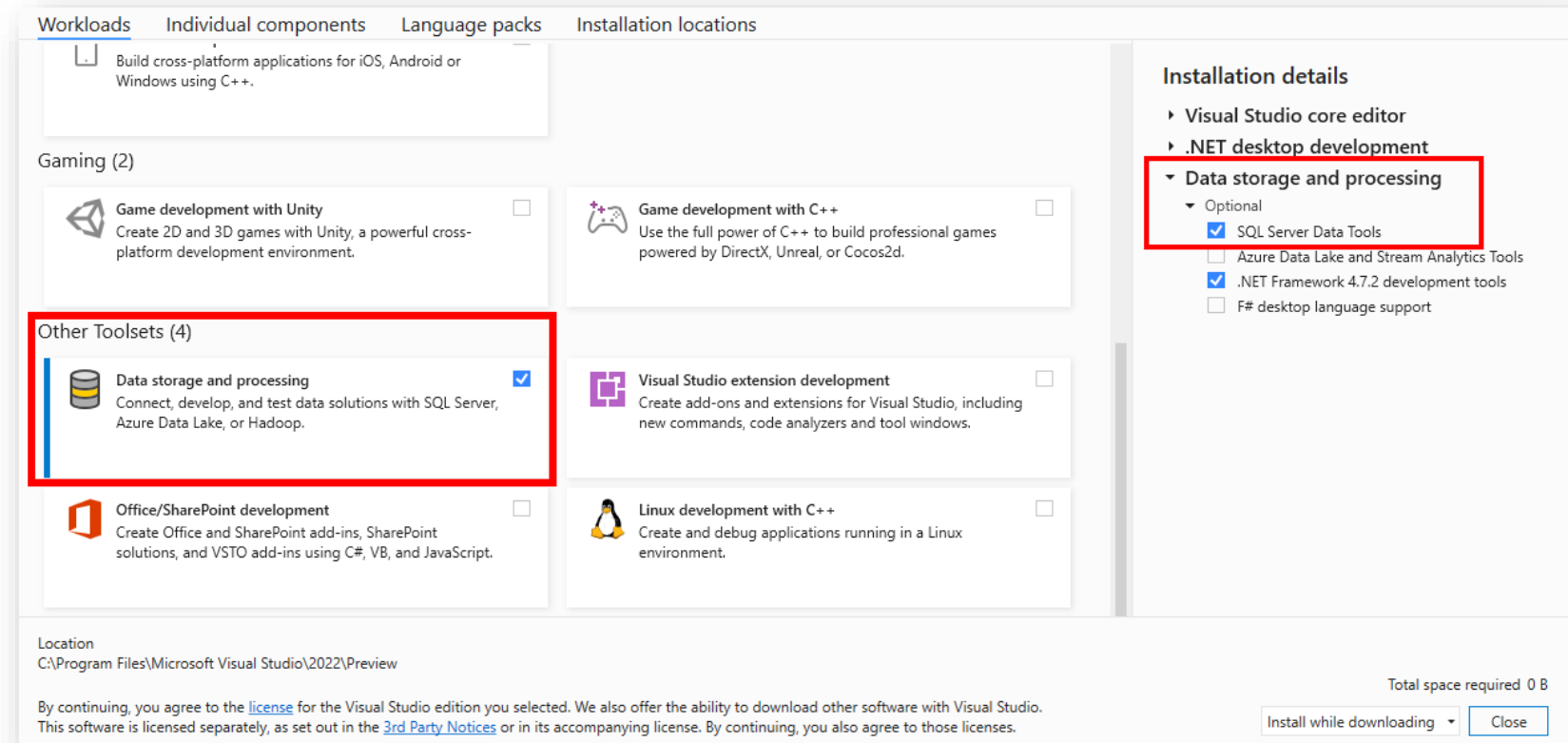
From A to B

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



## SQL Database Projects

Microsoft  [microsoft.com](https://microsoft.com) |  3,820,671 |  (5)

Enables users to develop and publish database schemas for MSSQL Databases

Disable 

Uninstall 

☒ Auto Update 

# Visual Studio bonus value

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

```

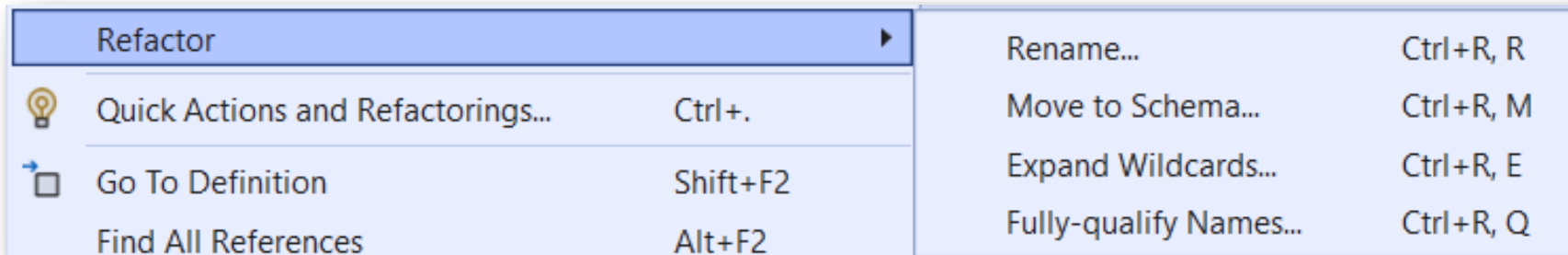
MergeProduct.sql *  Department.sql [Design]
1  Create Proc Staging.MergeProduct
2  As
3
4  Merge dw.Product trg
5      Using Staging.Product src
6      On trg.ProductID = src.ProductID
7  When Not Matched By Target Then
8      Insert (
9          [ProductID]
10         , [Product]
11         , [xProductNumber]
12         , [Color]
13         , [StandardCost]
14         , [ListPrice]
15         , [Model]
16         , [Category]
17     )
18     Values (
19         [ProductID]
20         , [Product]
21         , [ProductNumber]
22         , [Color]

```

Code	Description	Project	File	Line
SQL71501	Procedure: [Staging].[MergeProduct] has an unresolved reference to object [dw].[Product].[xProductNumber].	SampleDB	MergeProduct.sql	11

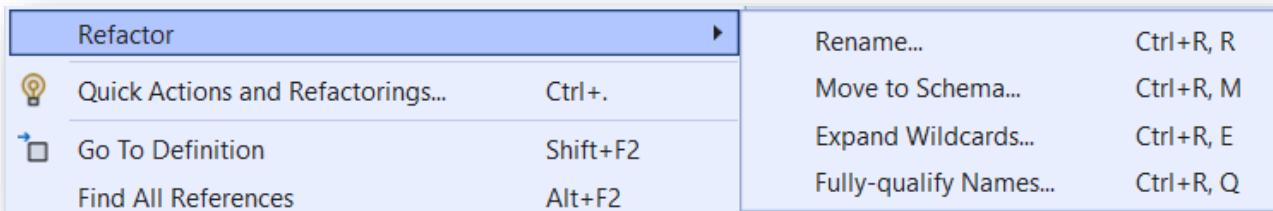
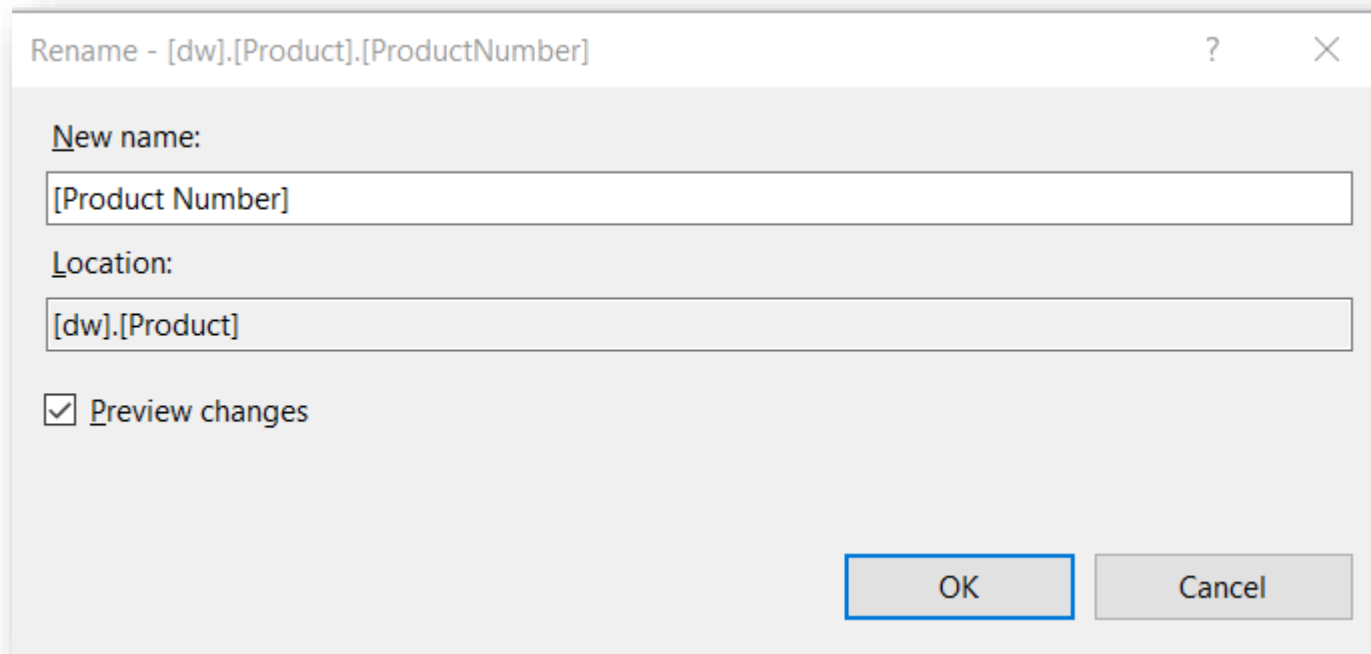
# Visual Studio bonus value

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



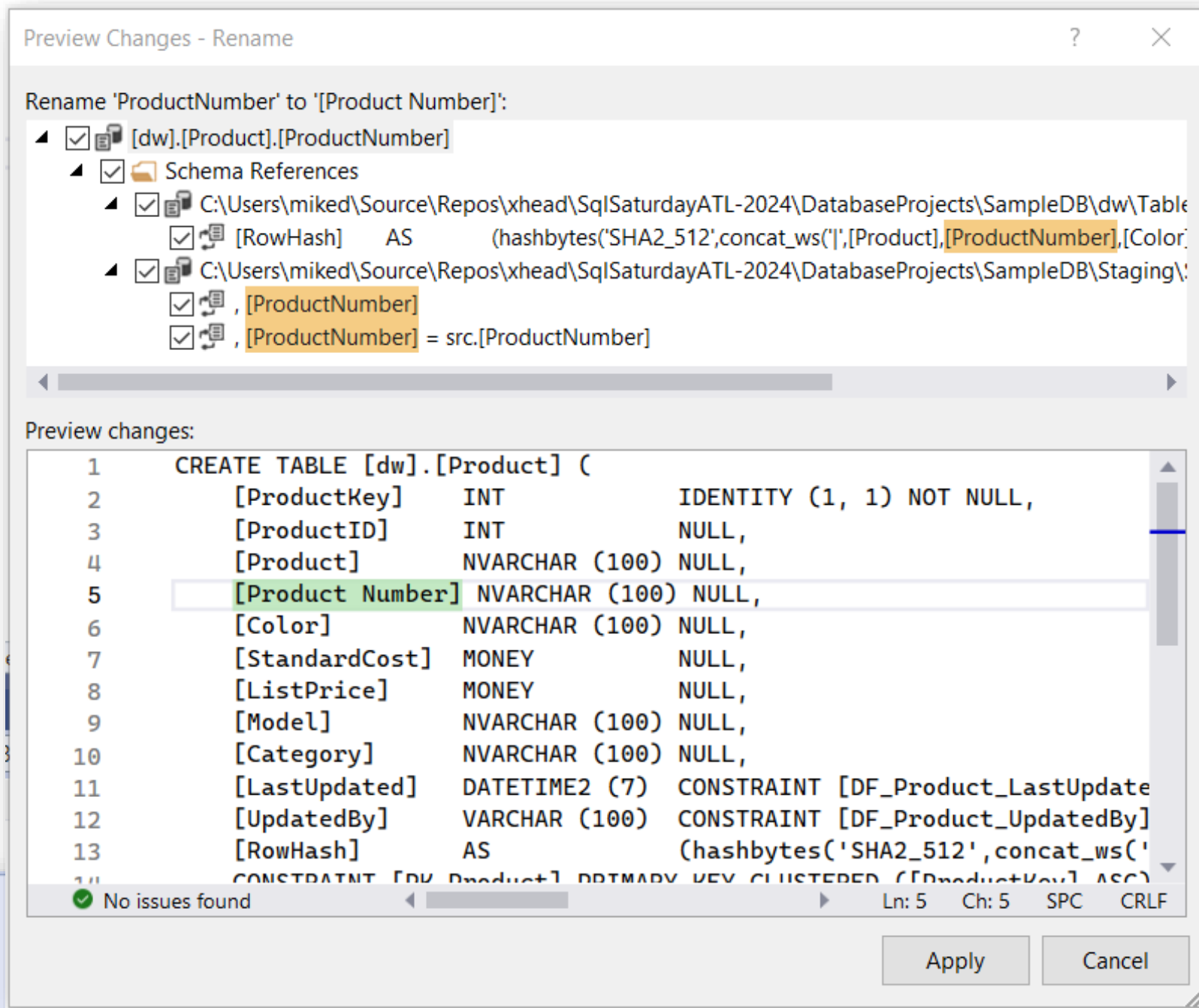
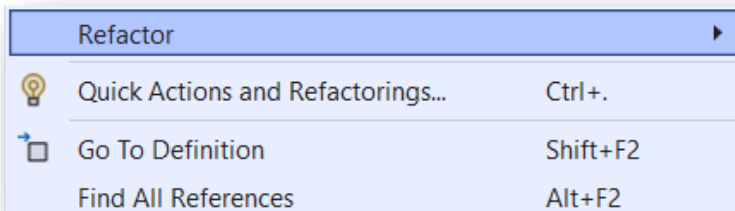
# Visual Studio bonus value

- 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



```
1 CREATE TABLE [dw].[Product] (  
2     [ProductKey] INT IDENTITY (1, 1) NOT NULL,  
3     [ProductID] INT NULL,  
4     [Product] NVARCHAR (100) NULL,  
5     [Product Number] NVARCHAR (100) NULL,  
6     [Color] NVARCHAR (100) NULL,  
7     [StandardCost] MONEY NULL,  
8     [ListPrice] MONEY NULL,  
9     [Model] NVARCHAR (100) NULL,  
10    [Category] NVARCHAR (100) NULL,  
11    [LastUpdated] DATETIME2 (7) CONSTRAINT [DF_Product_LastUpdate  
12    [UpdatedBy] VARCHAR (100) CONSTRAINT [DF_Product_UpdatedBy]  
13    [RowHash] AS (hashbytes('SHA2_512',concat_ws('
```

14 CONSTRAINT [PK\_Product] PRIMARY KEY CLUSTERED ([ProductKey] ASC)

Ln: 5 Ch: 5 SPC CRLF

Apply Cancel

# 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 Database SampleDB.publish.xml

Target Database Settings

Target database connection:  
Data Source=irc-miked-w10;Integrated Security=True;Persist Security Info=Fa [Edit...](#) [Clear](#)

Database name:  
SampleDB\_Dev

Publish script name:  
SampleDB.sql

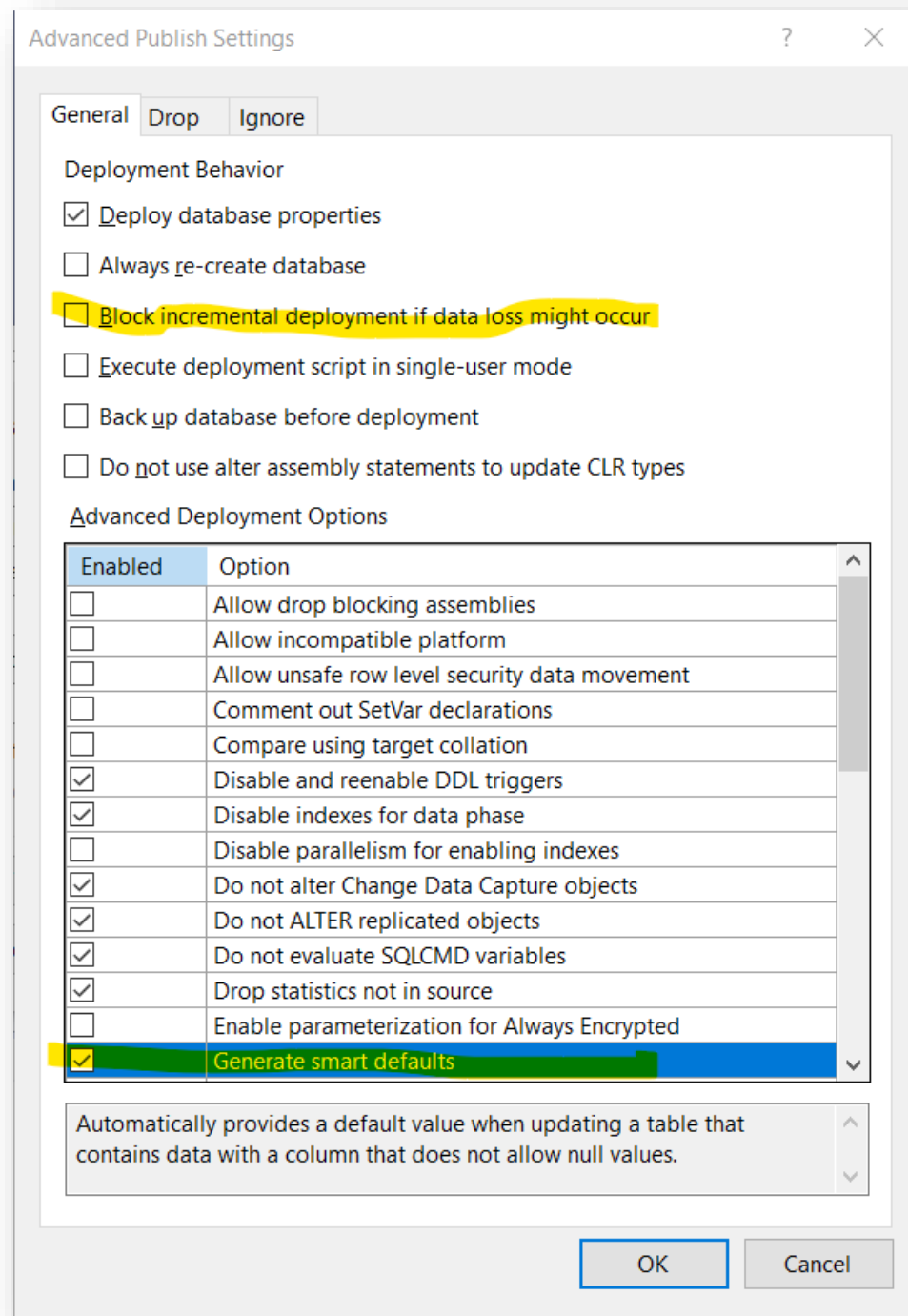
☐ Register as a Data-tier Application  
☐ Block publish when database has drifted from registered version [Advanced...](#)

[Load Profile...](#)

[Save Profile](#) [Save Profile As...](#) [Generate Script](#) [Publish](#) [Cancel](#)

# Publish from Visual Studio

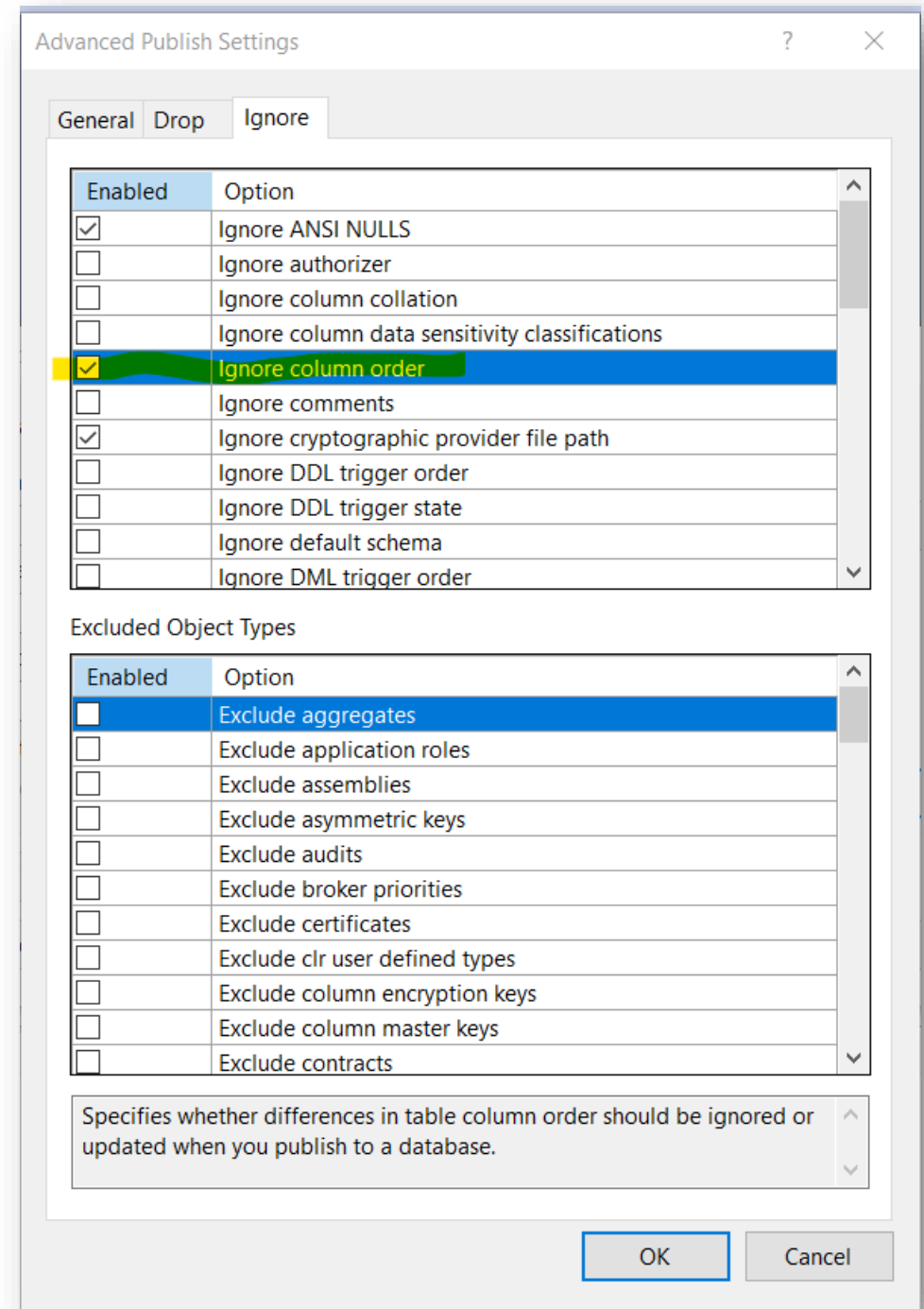
- Rt-click, Deploy
  - Create script
  - Publish directly
- Publish profiles (\*.publish.xml)
- Critical settings in publish profile
  - Use Smart Defaults
  - Ignore column order





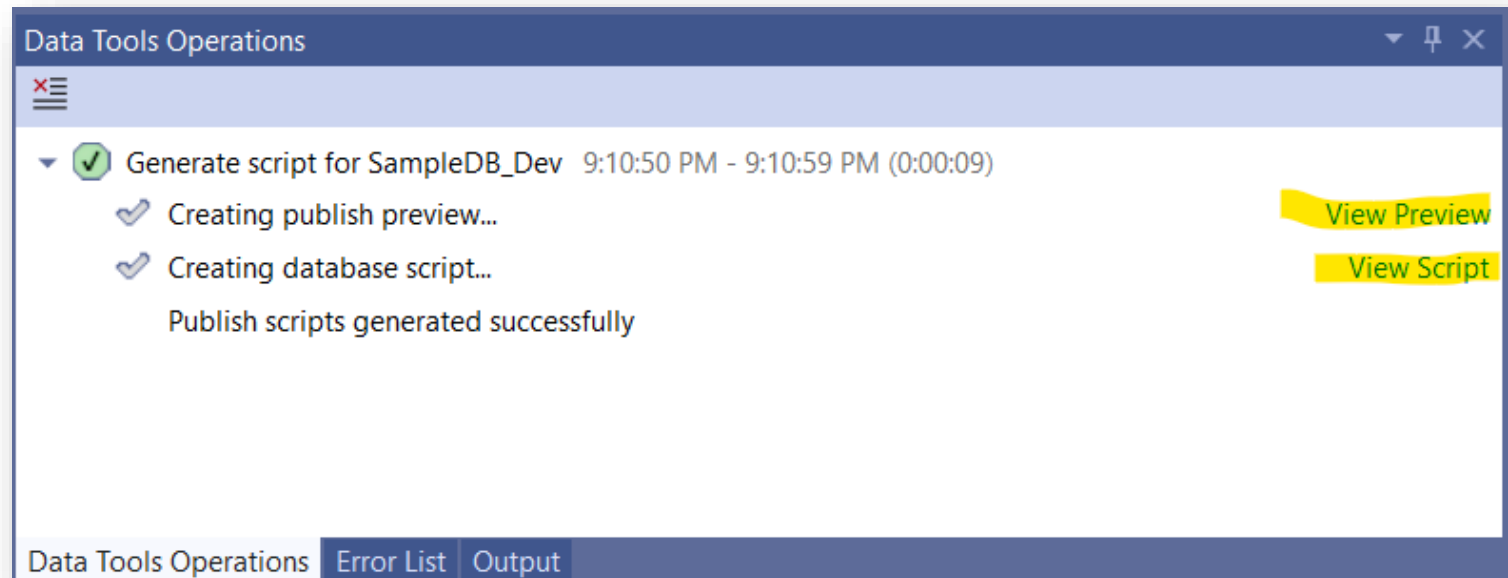
# Publish from Visual Studio

- 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

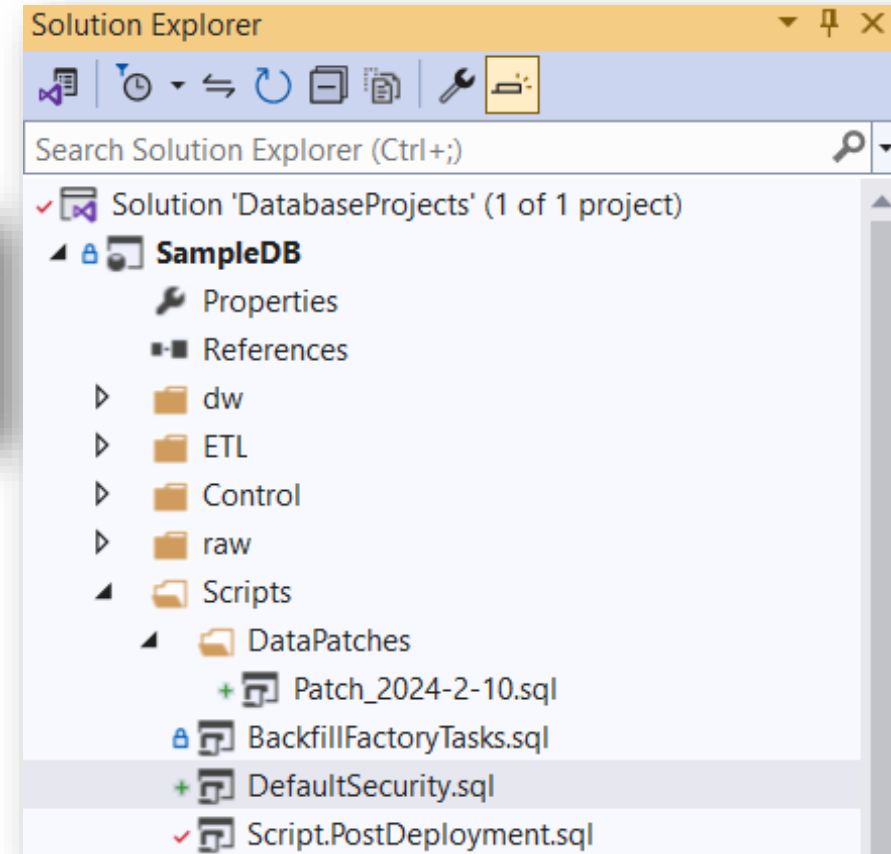
- Only one per project
  - :r .\DefaultSecurity.sql
  - :r .\BackfillReferenceData.sql
  - :r .\Patch\_2024-2-10.sql
- Scripts must be **idempotent**
  - Run one or more times without affecting the desired outcome

# 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

```
Print 'Backfilling Reference.Color';

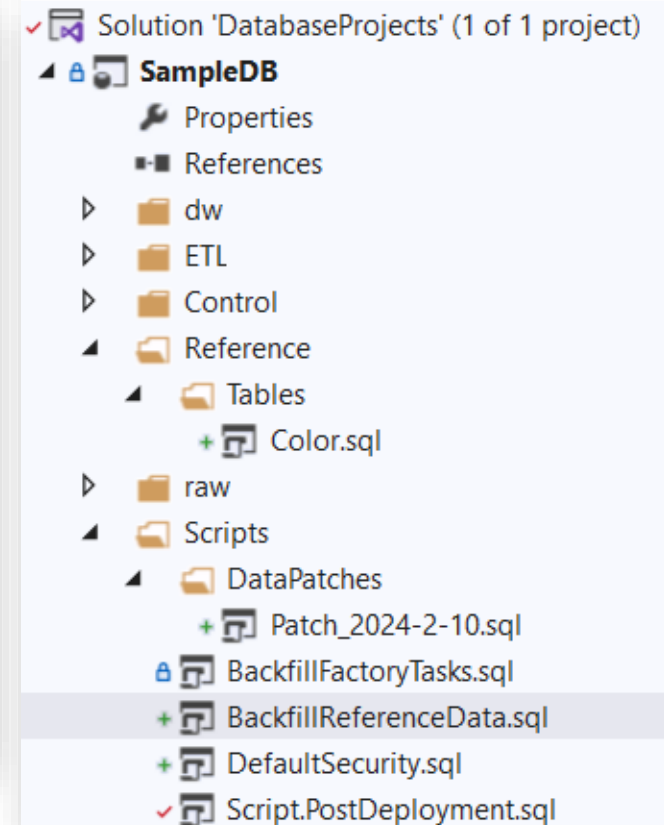
With colors As (
    Select * From (Values (
        ('Blue'),
        ('Red'),
        ('Orange'),
        ('Green'),
        ('Black')
    )) x (Color)
)
Merge Into Reference.Color trg
Using colors src On trg.Color = src.Color

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

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

```
/*
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 variable
Example:      :setvar TableName MyTable
               SELECT * FROM [$(TableName)]
-----
*/
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
```

✓ Solution 'DatabaseProjects' (1 of 1 project)

SampleDB

Properties

References

dw

ETL

Control

raw

Scripts

DataPatches

+ Patch\_2024-2-10.sql

BackfillFactoryTasks.sql

✓ Script.PostDeployment.sql

```
/*  
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)

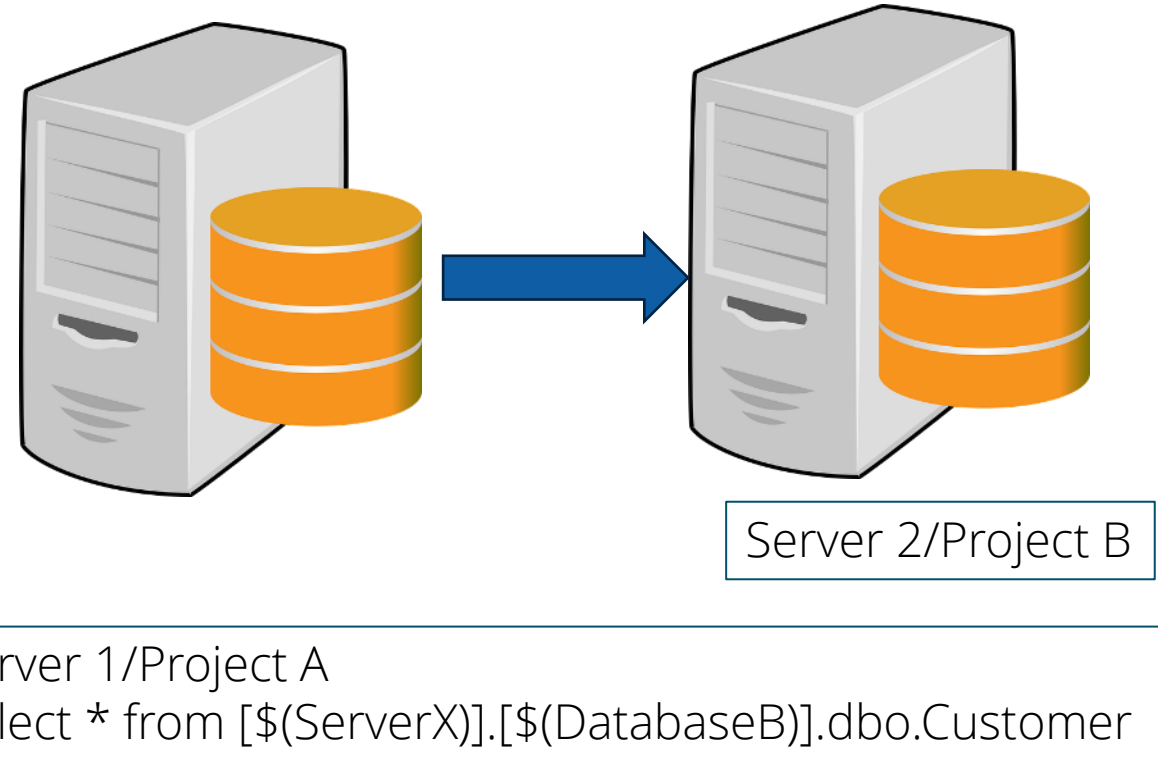


# Database references

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

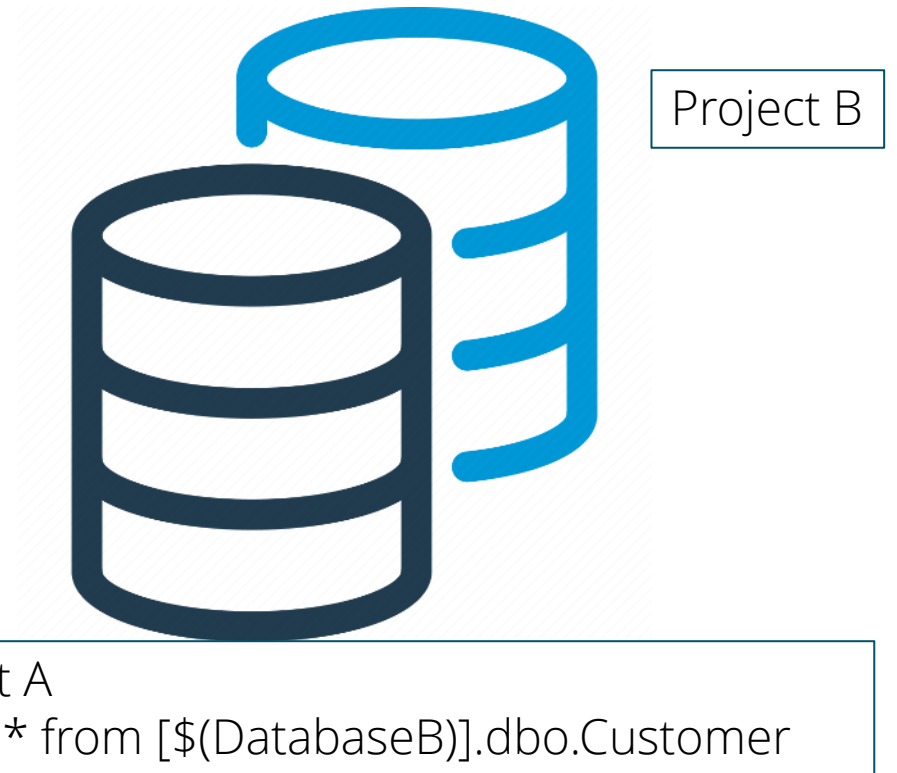
# Database references

- 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



# 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



# 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



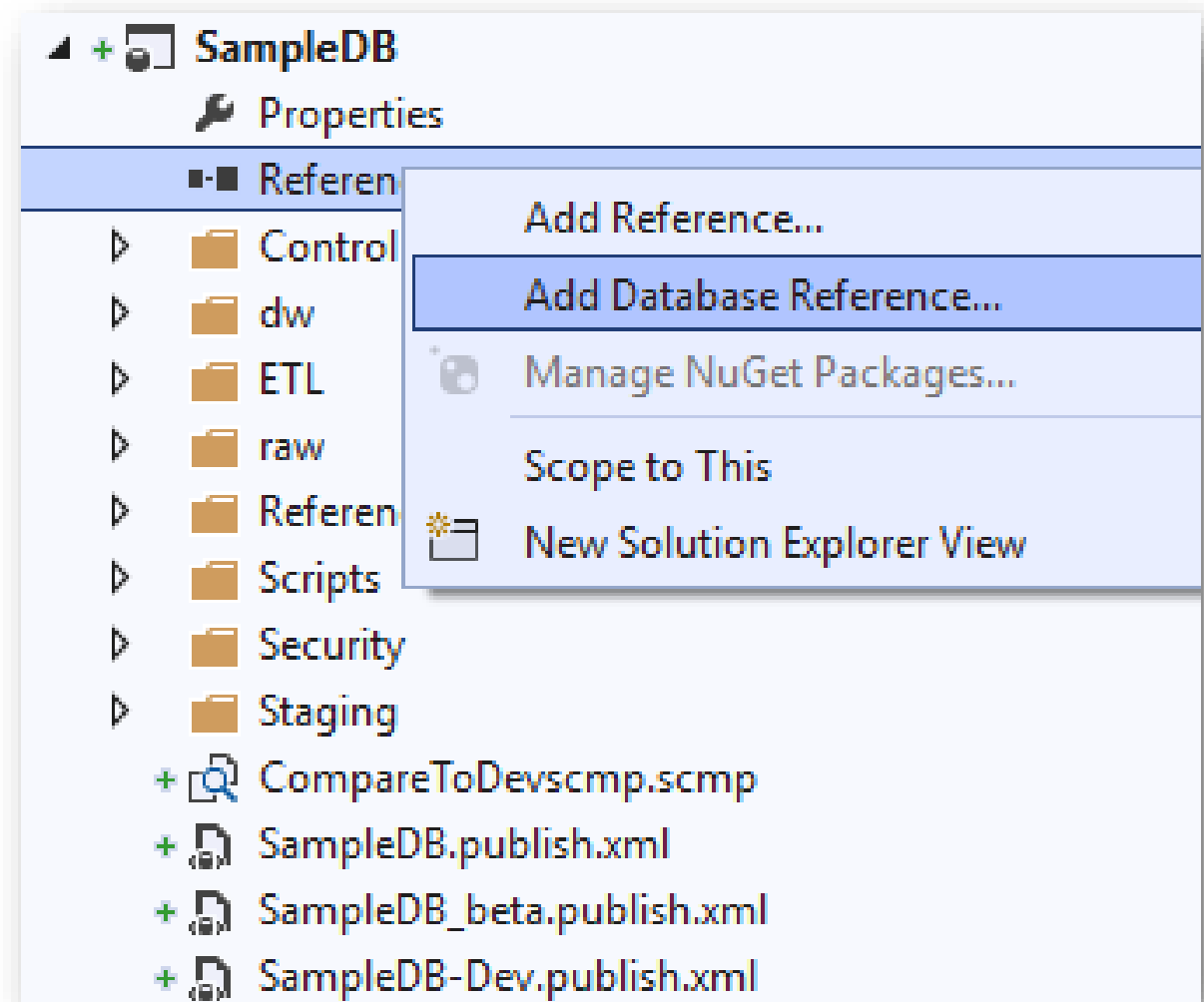
Project A

Project B

Select \* from dbo.Customer

# Database references

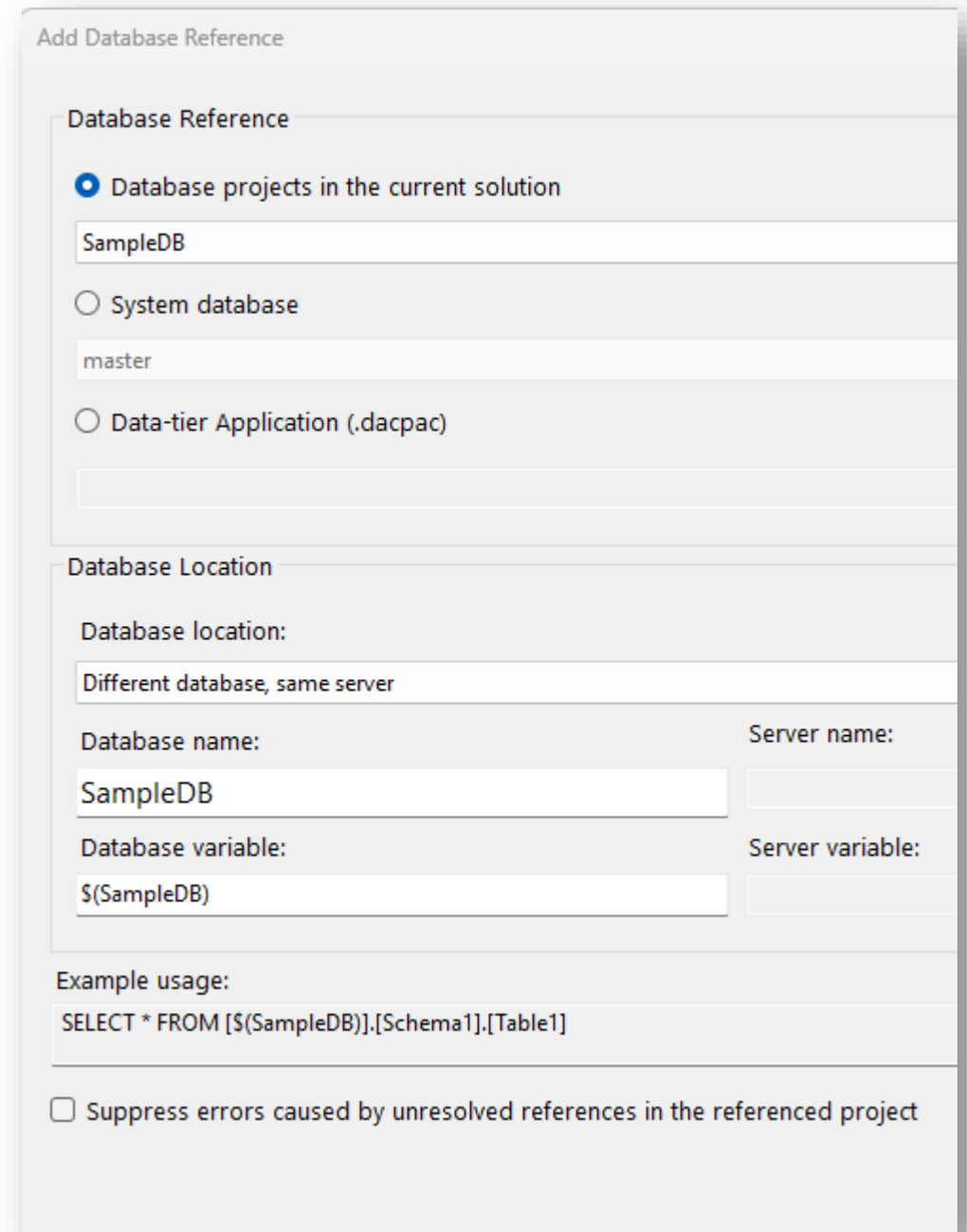
- What is in or out of your control? What can you deploy?
- System databases
  - master, msdb
- Other database projects
- DACPAC files (binary reference)





# 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)



The screenshot shows the 'Add Database Reference' dialog box. It has two main sections: 'Database Reference' and 'Database Location'. In the 'Database Reference' section, the 'Database projects in the current solution' radio button is selected, and 'SampleDB' is entered in the text box. The 'System database' radio button is unselected, with 'master' entered in its text box. The 'Data-tier Application (.dacpac)' radio button is also unselected. The 'Database Location' section has a 'Database location:' dropdown set to 'Different database, same server'. Below this, there are four text boxes: 'Database name:' (SampleDB), 'Server name:' (empty), 'Database variable:' (\$(SampleDB)), and 'Server variable:' (empty). At the bottom, there is an 'Example usage:' section with a text box containing the SQL query: `SELECT * FROM [$(SampleDB)].[Schema1].[Table1]`. Finally, there is a checkbox labeled 'Suppress errors caused by unresolved references in the referenced project' which is currently unchecked.

Add Database Reference

Database Reference

☒ Database projects in the current solution

SampleDB

☐ System database

master

☐ Data-tier Application (.dacpac)

Database Location

Database location:

Different database, same server

Database name: SampleDB

Server name:

Database variable: \$(SampleDB)

Server variable:

Example usage:

SELECT \* FROM [\$(SampleDB)].[Schema1].[Table1]

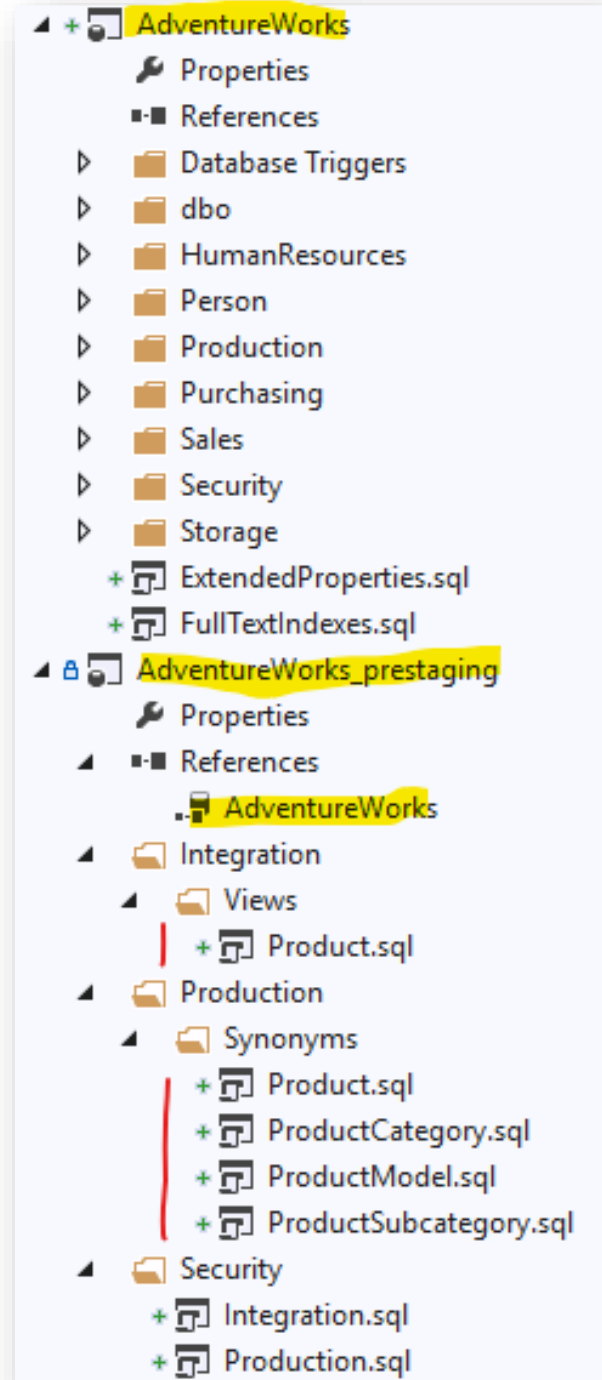
☐ Suppress errors caused by unresolved references in the referenced project

# 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

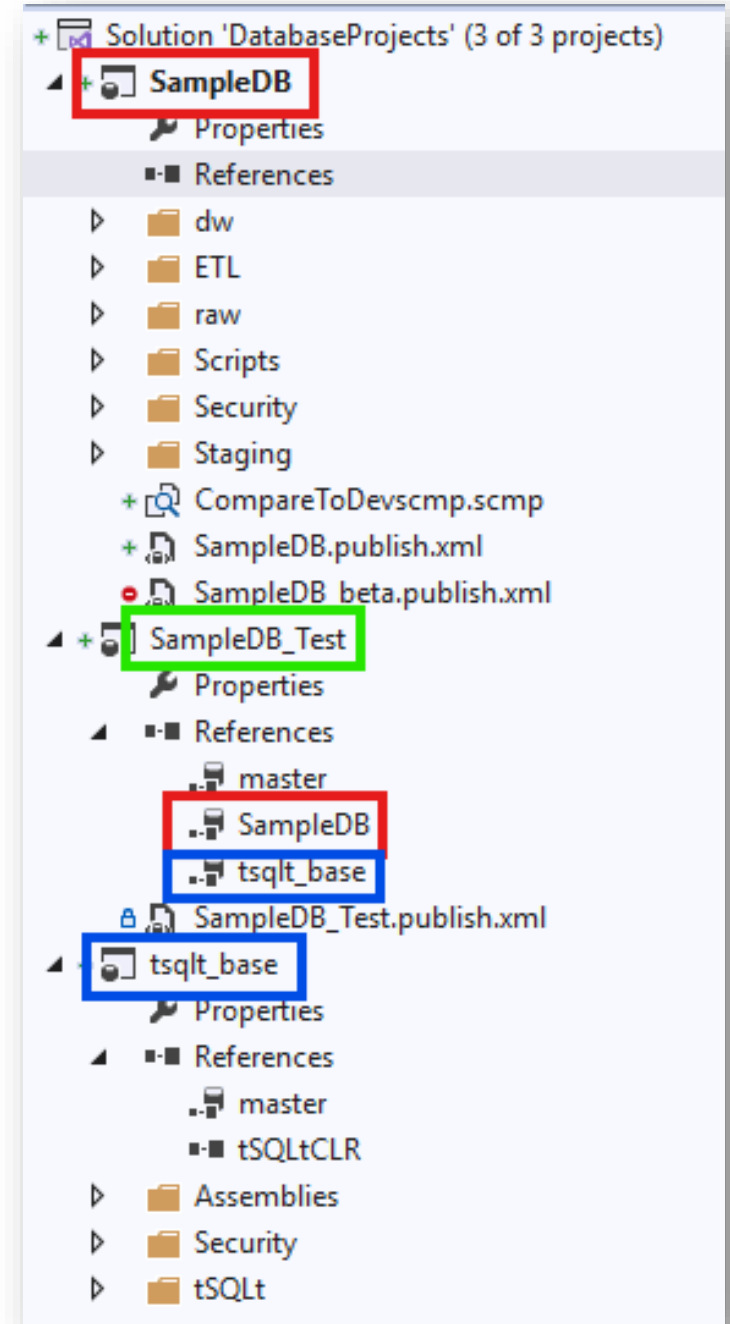
```
create view Integration.Product as
select
    p.ProductID,
    p.ProductNumber,
    p.Name as ProductName,
    p.ListPrice,
    p.StandardCost,
    p.Color,
    p.Weight,
    p.Size,
    pc.Name as ProductCategory,
    ps.Name as ProductSubCategory,
    pm.Name as ProductModel
from Production.Product 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!*

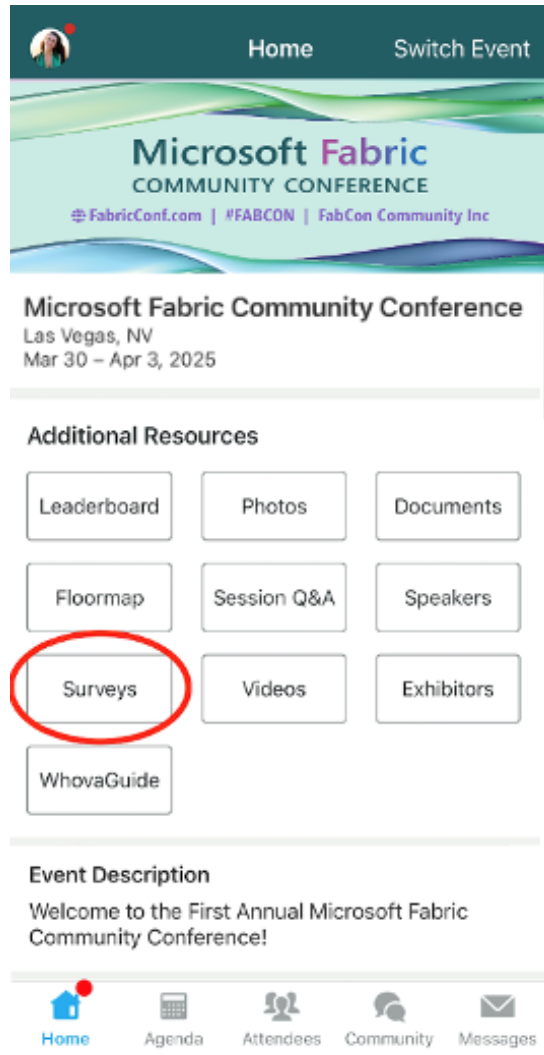


<https://github.com/xhead/FabCon2025>



**/mikediehlsqbi**

**Mike.Diehl@Improving.Com**



## 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](https://NextGen365Events.com)

---

next **GenAI**  
CONFERENCE

**OCT 7-9, 2025** **LOWES SAPPHIRE FALLS RESORT**  
WORKSHOPS: OCT 5, 6 & 10 **ORLANDO, FL**  
 [NextGenAIconf.com](https://NextGenAIconf.com)



**Power Platform**  
COMMUNITY CONFERENCE

**OCT 28-30, 2025** **MGM GRAND**  
WORKSHOPS: OCT 26, 27 & 31 **LAS VEGAS, NV**  
 [PowerPlatformConf.com](https://PowerPlatformConf.com)