




This presentation is the property of Microsoft and is intended for informational and educational purposes only. You may use, copy, and distribute this presentation for your personal, non-commercial purposes. You may not modify, alter, or create derivative works from this presentation without the prior written consent of Microsoft. You may not use this presentation to misrepresent, defame, or disparage Microsoft or its products, services, or affiliates. You may not use this presentation to endorse or promote any other products, services, or organizations without the prior written consent of Microsoft.

By using this presentation, you agree to abide by these terms. If you do not agree, you must not use this presentation. Microsoft reserves the right to change these terms and conditions at any time without notice. Microsoft disclaims any and all warranties, express or implied, relating to this presentation, including but not limited to the accuracy, completeness, timeliness, or suitability of the information contained herein. Microsoft is not liable for any damages, losses, or liabilities arising from your use of or reliance on this presentation.

Please review the terms of use posted in the content library.



Batteries included: SQL DevOps with SQL projects

Drew Skwiers-Koballa
Principal Product Manager, Microsoft

Microsoft SQL – Ground to Cloud to Fabric

One SQL for the era of AI

Develop once, deploy anywhere

GA



SQL Server 2025

SQL

Azure SQL

GA



SQL database in Fabric

T-SQL

Developers



Engine



Tools



Fabric



AI



Copilots

Foundations of DevOps



Culture: Communication and collaboration



Product: Delivering value efficiently



Process: Repeatable actions and consistent outcomes



Tools: Technology aiding culture, product, and process



Branch
Build
Commit
Ship



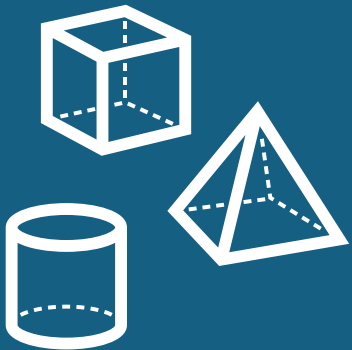
SQL Database Project



Database settings



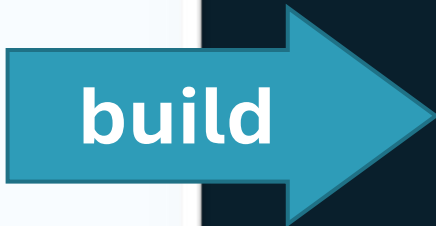
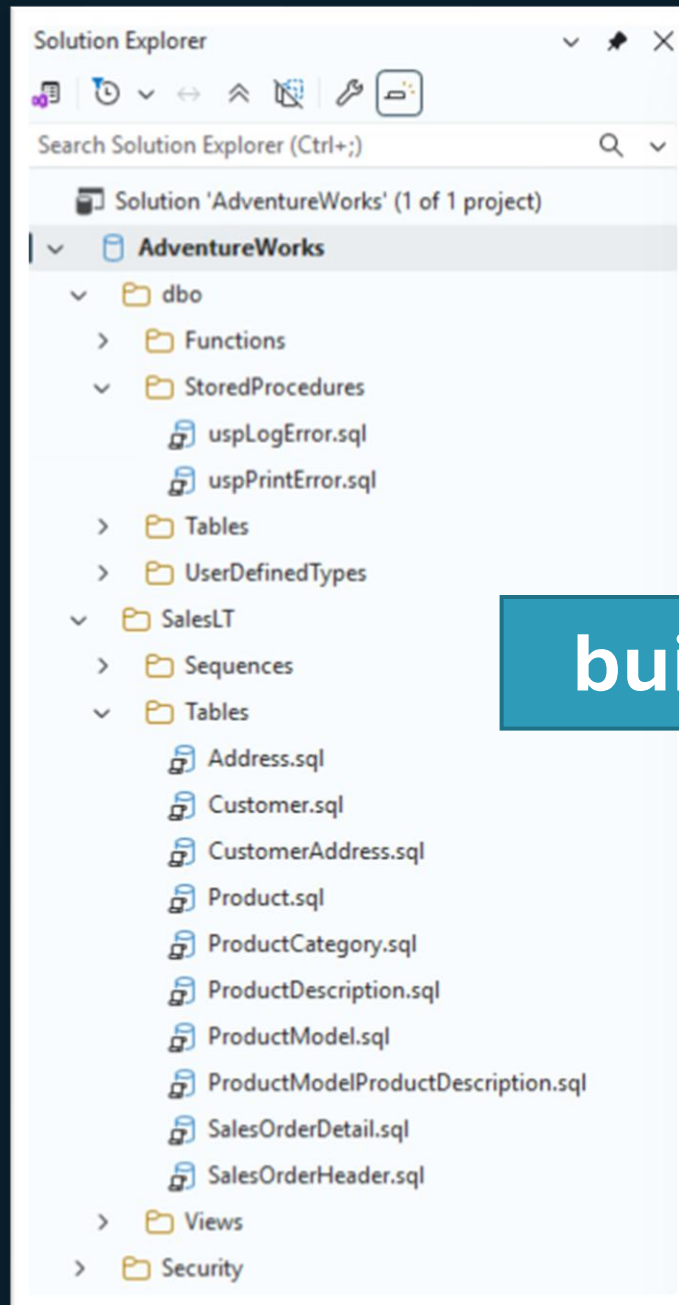
Pre-deployment script



Database objects
(tables, stored
procs, indexes,
etc.)



Post-deployment script



SQL projects evolution

Original SQL project format

- “build” process validates syntax and references (*.NET Framework*)
- Dacpac file output can be deployed to new and existing databases
- Link multiple projects or dacpacs together to reference same or other databases (database references)
- Standard code analysis (antipattern check) with 14 default rules

Microsoft.Build.Sql “SDK-style”

- ← all of that, plus
- Build cross-platform with vanilla “dotnet build”
- Upgrade available syntax right from project file, no additional download needed
- Includes all *.sql files in the folder by default
- Package references for database references and custom code analysis rules

EXPLORER

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

You can clone a repository locally.

Clone Repository

To learn more about how to use Git and source control in VS Code read [our docs](#).

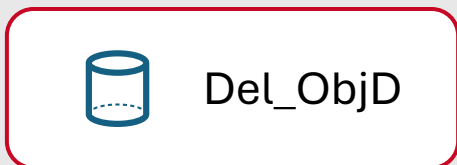
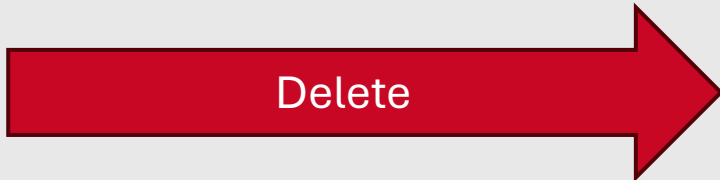
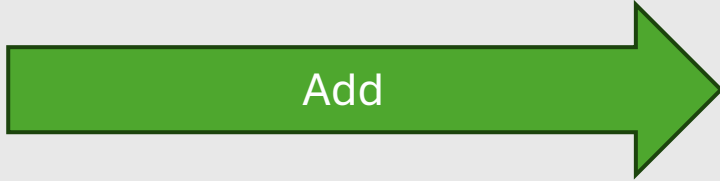
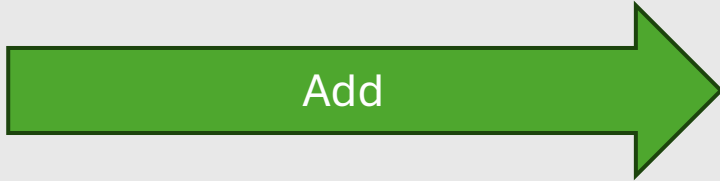
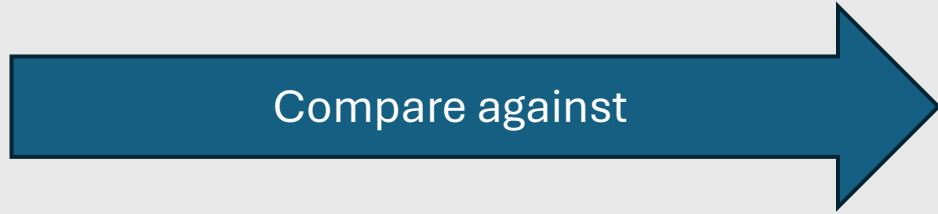
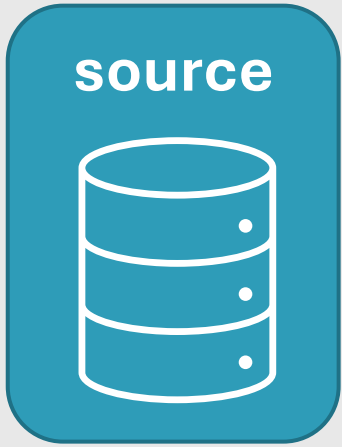
OUTLINE

TIMELINE



- Open Chat Ctrl + Alt + I
- Show All Commands Ctrl + Shift + P
- Open Recent Ctrl + R
- Open File Ctrl + O
- Open Folder Ctrl + K Ctrl + O

Commit



Fabric CI/CD

The screenshot shows the Fabric Source Control interface. At the top, there are navigation options: "View deployment pipeline", "Create app", "Manage access", and "Workspace settings". Below this is a search bar for "Source control" with a "Filter by keyword" input and a "Filter" dropdown. The main area is divided into two sections: a table of tasks and a "Source control" panel.

| Task | Owner | Refreshed | Ne ref |
|------|------------|-------------|--------|
| — | Anna Ho... | — | — |
| — | Anna Ho... | — | — |
| — | Contoso... | 7/22/202... | N/A |
| — | Anna Ho... | — | — |
| — | Anna Ho... | — | — |
| — | Contoso... | 10/25/20... | N/A |
| — | Anna Ho... | — | — |
| — | Anna Ho... | — | — |
| — | Anna Ho... | — | — |

The "Source control" panel shows the current branch as "planelanding" and a list of changes. The changes list includes:

- Contoso PDFs RAG (green plus icon)
- RAG-with-resumes (green plus icon)
- api1 (checked, green plus icon)
- ContosoDB (checked, green plus icon)
- dev (checked, green plus icon)
- Notebook 1 (checked, red minus icon)

At the bottom of the panel are "Commit" and "Undo" buttons.

Every database under source control

The screenshot shows the SQL Explorer interface. The "ReferenceData" view is active, displaying a SQL query:

```
1 MERGE INTO [dbo].[Countries] AS target
2 USING (VALUES
3 ('US', 'United States'),
4 ('CA', 'Canada'),
5 ('MX', 'Mexico'),
6 ('JA', 'Japan'),
7 ('UK', 'United Kingdom')
8 ) AS source ([CountryCode], [CountryName])
9 ON target.[CountryCode] = source.[CountryCode]
10 WHEN MATCHED THEN
11 UPDATE SET [CountryName] = source.[CountryName]
12 WHEN NOT MATCHED BY TARGET THEN
13 INSERT ([CountryCode], [CountryName]) VALUES (sou
```

The Explorer pane on the left shows the folder structure: AdventureWorks > dbo > SalesLT > Queries > My queries > Shared queries > ReferenceData. A context menu is open over the "ReferenceData" file, with the following options:

- Duplicate
- Rename
- Delete
- Set as Pre-deployment Script
- Set as Post-deployment Script (highlighted with a red box)
- Move to My queries

EXPLORER

- ZAVAFINANCIALLOANAPPS
 - .vs
 - .vscode
 - bin
 - dbo
 - StoredProcedures
 - Tables
 - account_ownership.sql
 - accounts.sql
 - borrowers.sql
 - collaterals.sql
 - credit_bureaus.sql
 - credit_scores.sql
 - income_verifications.sql
 - loan_applications.sql
 - Views
 - vw_active_portfolio_overview.sql
 - vw_borrower_credit_summary.sql
 - vw_loan_application_details.sql
 - obj
- ZavaFinancialLoanApps.sqlproj

OUTLINE

TIMELINE

loan_applications.sql X

dbo > Tables > loan_applications.sql

Connect to MSSQL

```

1 CREATE TABLE [dbo].[loan_applications] (
2     [app_id] INT IDENTITY (1, 1) NOT NULL,
3     [application_channel] NVARCHAR (50) NULL,
4     [borrower_id] INT NULL,
5     [decision_status] NVARCHAR (30) NULL,
6     [lender_id] NVARCHAR (50) NULL,
7     [product_id] NVARCHAR (50) NULL,
8     [purpose_code] NVARCHAR (30) NULL,
9     [requested_amount] BIGINT NULL,
10    [requested_term_months] BIGINT NULL,
11    [submitted_utc] DATETIME2 (6) NULL,
12    [underwriting_model_version] NVARCHAR (50) NULL,
13    CONSTRAINT [PK_loan_applications] PRIMARY KEY CLUSTERED ([app_id] ASC)
14 );
15
16
17 GO
18
19

```



Choose from predesigned task flows or add a task to build one
Select from one of Microsoft's predesigned task flows or add a task to start building one yourself.

Select a predesigned task flow | Add a task

Import a task flow

| Name | Status | Type | Task | Owner | Refreshed | Next refresh | Endorsemei | Sensitivity | Included in app |
|--------------------|--------|------------------|------|-----------------|-----------|--------------|------------|------------------|-----------------|
| LoanApplicationApp | | SQL database | — | Drew Skwiers... | — | — | — | Confidential\... | ⊙ |
| LoanApplicationApp | | SQL analytics... | — | Drew Skwiers... | — | — | — | Confidential\... | ⊙ |

Build

Checking as early as possible:

Is the syntax correct for the version of SQL?

Do all the referenced objects exist?

Does the code meet certain quality standards?

And creating a reusable output artifact.

EXPLORER

- ZAVAFINANCIALLOANAPPS
 - .vs
 - .vscode
 - bin
 - dbo
 - StoredProcedures
 - Tables
 - account_ownership.sql
 - accounts.sql
 - borrowers.sql
 - collaterals.sql
 - credit_bureaus.sql
 - credit_scores.sql
 - income_verifications.sql**
 - loan_applications.sql
 - Views
 - vw_active_portfolio_overview.sql
 - vw_borrower_credit_summary.sql
 - vw_loan_application_details.sql
 - vw_pending_income_verification.sql
 - obj
 - .gitignore

OUTLINE

- TIMELINE income_verifications.sql
 - new pending income verification v... 58 mins
 - initial commit Drew Skwiers-Koballa 1 hr
 - File Saved

income_verifications.sql

dbo > Tables > income_verifications.sql

```

1 CREATE TABLE [dbo].[income_verifications] (
2     [incv_id] INT IDENTITY (1, 1) NOT NULL,
3     [borrower_id] INT NOT NULL,
4     [app_id] INT NOT NULL,
5     [verification_method] NVARCHAR (50) NULL,
6     [verification_provider] NVARCHAR (100) NULL,
7     [verification_status] NVARCHAR (30) NULL,
8     [verification_date] DATETIME2 (7) NULL,
9     CONSTRAINT [PK_income_verifications] PRIMARY KEY CLUSTERED ([incv_id] ASC)
10 );
11
12
13 GO
14
15

```

Design

SR0001: Avoid SELECT *

SR0008: Consider SCOPE_IDENTITY instead of @@IDENTITY

SR0009: Avoid types of variable length size 1 or 2

SR0013: Output parameter isn't populated in all code paths

SR0014: Data loss might occur when casting

Naming

SR0011: Avoid using special characters in object names

SR0012: Avoid using reserved words for type names

SR00016: Avoid sp_ as a prefix for stored procedures

Performance

SR0004: Avoid using columns that do not have indexes as test expressions in IN predicates

SR0005: Avoid using patterns that start with "%" in LIKE predicates

SR0006: Move a column reference to one side of a comparison operator to use a column index

SR0007: Use ISNULL(column, default_value) on nullable columns in expressions

SR0015: Extract deterministic function calls from WHERE predicates

```
name: Build SQL project with code analysis
```

```
on:
```

```
  push:
```

```
    branches: [ "main" ]
```

```
    paths:
```

```
      - 'Zava/**'
```

```
  pull_request:
```

```
    branches: [ "main" ]
```

```
    paths:
```

```
      - 'Zava/**'
```

```
jobs:
```

```
  build:
```

```
    runs-on: ubuntu-24.04
```

```
    steps:
```

```
      - uses: actions/checkout@v4
```

```
      - name: Setup .NET
```

```
        uses: actions/setup-dotnet@v4
```

```
        with:
```

```
          dotnet-version: 10.0.x
```

```
      - name: Build
```

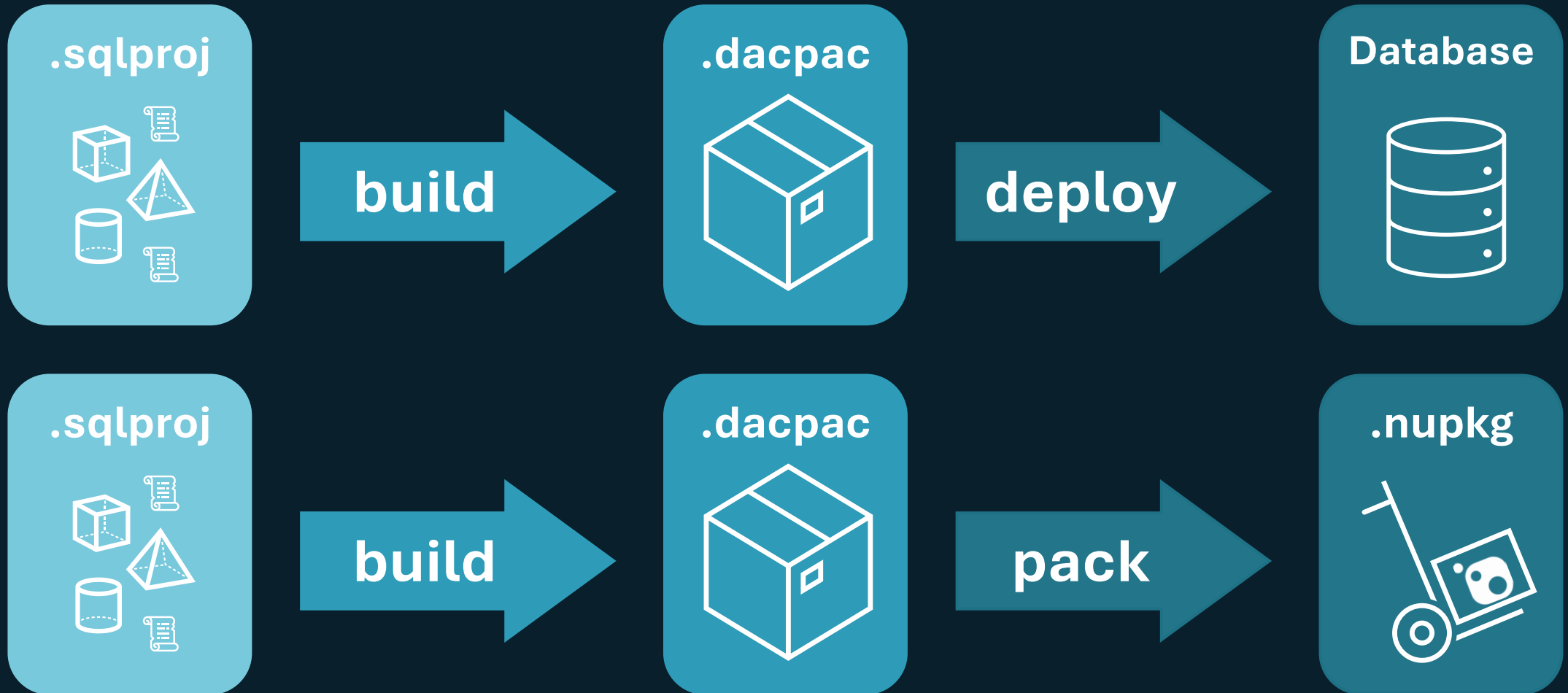
```
        run: dotnet build Zava/Zava.sqlproj /p:RunSqlCodeAnalysis=true
```

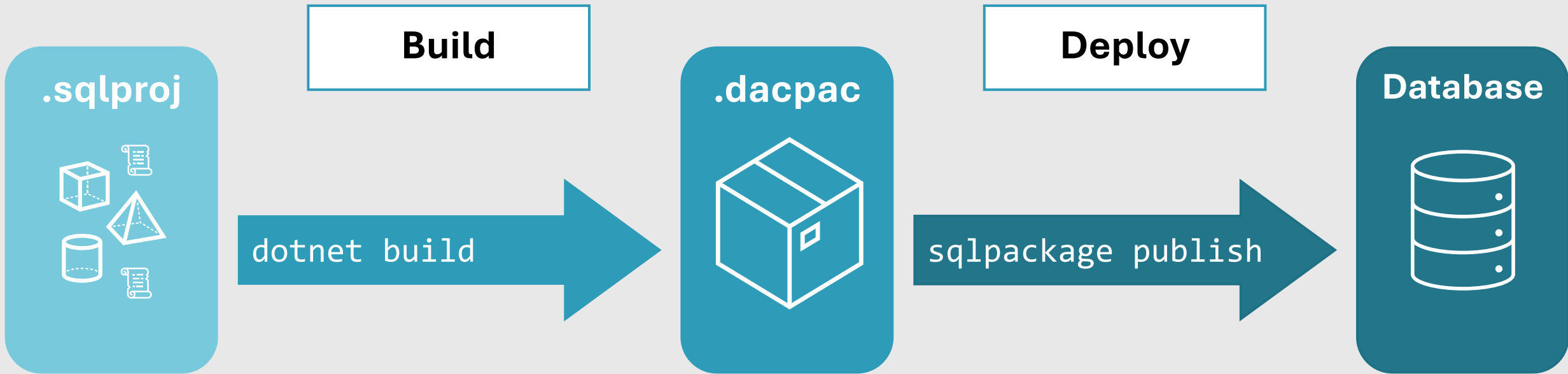
A blue rounded rectangular button with the text "CI automation" in white, bold, sans-serif font.

CI automation

Ship

Delivery options





SQL project contains

```
CREATE TABLE Customer  
(  
    FirstName NVARCHAR(100) NOT NULL,  
    LastName NVARCHAR(100) NOT NULL,  
    Region NVARCHAR(100) NULL  
)
```

SqlPackage deployment

```
ALTER TABLE Customer  
ADD Region NVARCHAR(100)
```

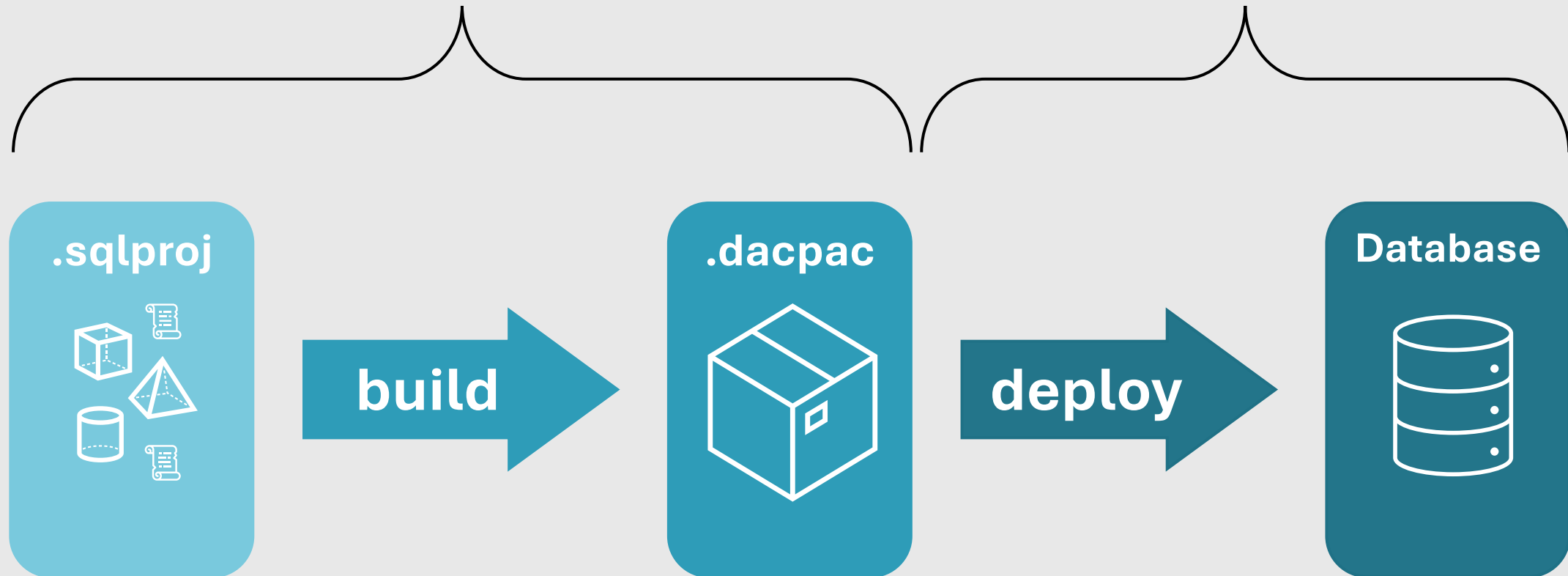
SQLCMD variables - value resolution during deployment

SQLCMD variable unresolved

'\$(EnvironmentName)'

SQLCMD variable set

'staging'



SQL SERVER

CONNECTIONS

- drskwier.database.windows.n...
 - Databases
 - Fabric-LoanApplicationApp

QUERY HISTORY



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS Build - Task ✓ + ▾ □ 🗑️ ... | 🔄 ✕

```
* Executing task: C:\Program Files\dotnet\dotnet.EXE build c:\\Users\\drskwier\\source\\ZavaFinancial\\ZavaFinancialLoanApps\\ZavaFinancialLoanApps.sqlproj /p:NetCoreBuild=true /p:SystemDacpacsLocation="c:\\Users\\drskwier\\.vscode\\extensions\\ms-mssql.sql-database-projects-vscode-1.5.8\\BuildDirectory"
Restore complete (0.8s)
  info NETSDK1057: You are using a preview version of .NET. See: https://aka.ms/dotnet-support-policy
  ZavaFinancialLoanApps netstandard2.1 succeeded (4.5s) -> bin\\Debug\\ZavaFinancialLoanApps.dacpac
Build succeeded in 6.1s
* Terminal will be reused by tasks, press any key to close it.
```

Object Explorer

Connect

- localhost (SQL Server 17.0.1105.2 - REDMOND)
- Databases
 - System Databases
 - Database Snapshots
 - AdventureWorks
 - LoanApplicationApp
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.account_ownership
 - dbo.accounts
 - dbo.borrowers
 - dbo.collaterals
 - dbo.credit_bureaus
 - dbo.credit_scores
 - dbo.income_verifications
 - dbo.loan_applications
 - Dropped Ledger Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Stored Procedures
 - System Stored Procedures
 - dbo.EvaluateBorrowerRisk
 - dbo.GetLatestCreditScores
 - dbo.SubmitLoanApplication
 - Functions
 - Database Triggers
 - Assemblies
 - Types
 - Rules
 - Defaults
 - Plan Guides
 - Sequences
 - Query Store
 - Service Broker
 - Storage
 - Security
 - Security
 - Server Objects
 - Replication
 - Always On High Availability
 - Management
 - SQL Server Agent (Agent XPs disabled)
 - XEvent Profiler



Solution Explorer

Solution 'ZavaFinancialLoanApps' (1 of 1 project)

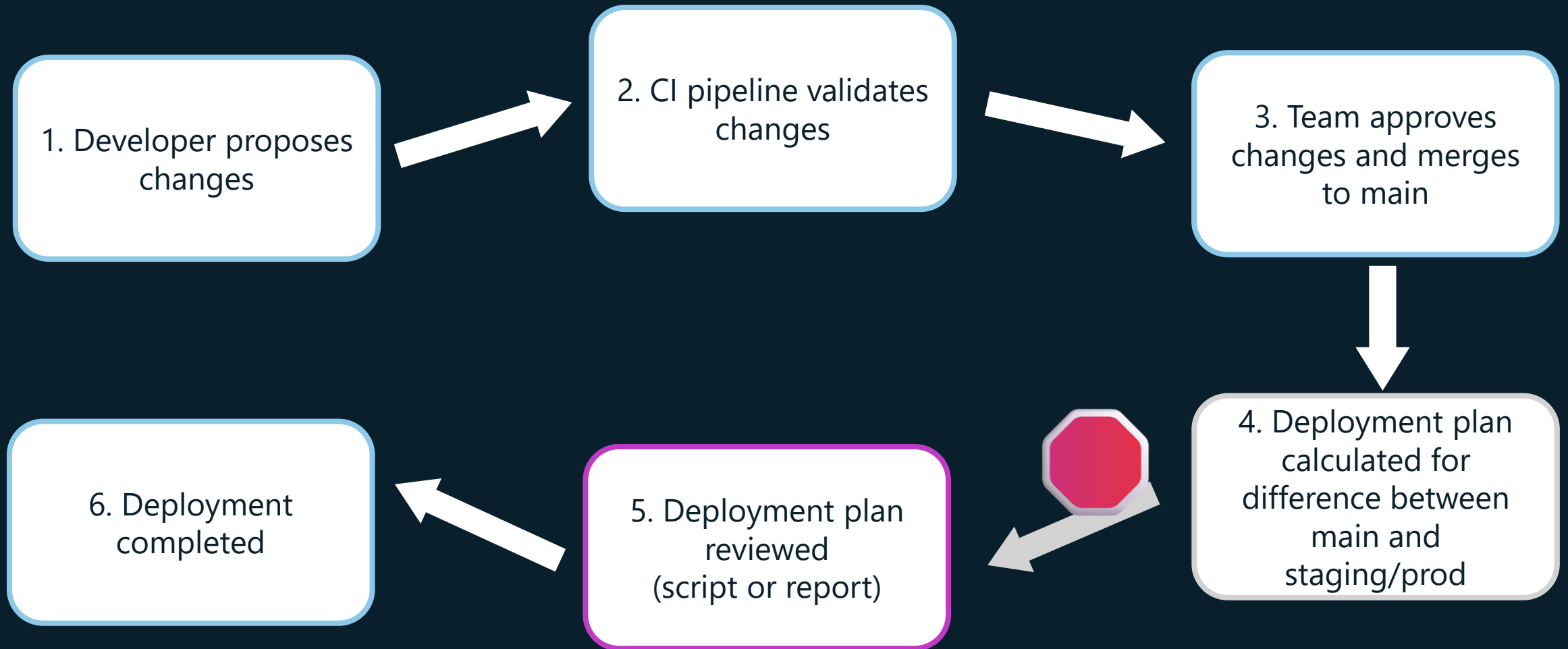
- ZavaFinancialLoanApps
 - dbo
 - StoredProcedures
 - EvaluateBorrowerRisk.sql
 - GetLatestCreditScores.sql
 - SubmitLoanApplication.sql
 - Tables
 - account_ownership.sql
 - accounts.sql
 - borrowers.sql
 - collaterals.sql
 - credit_bureaus.sql
 - credit_scores.sql
 - income_verifications.sql
 - loan_applications.sql
 - Views
 - vw_active_portfolio_overview.sql
 - vw_borrower_credit_summary.sql
 - vw_loan_application_details.sql
 - vw_pending_income_verification.sql

Output

Show output from: Build

```
2>Creating Procedure [dbo].[GetLatestCreditScores]...
2>Creating Procedure [dbo].[SubmitLoanApplication]...
2>Update complete.
2>Updating database
2>Updating database (Complete)
2>Deploying package to database
2>Publish operation completed successfully.
2>==== SQL Database Publish Operation Complete ===
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
===== Build completed at 5:14 PM and took 39.055 seconds =====
===== Publish: 1 succeeded, 0 failed, 0 skipped =====
===== Publish completed at 5:14 PM and took 39.055 seconds =====
```

Gate pipelines for human validation



Summary

Jobs

✓ build

🕒 deploy

Run details

🔗 Usage

📄 Workflow file

Manually triggered 4 minutes ago

| Status | Total duration | Artifacts |
|------------------|----------------|-----------|
| 🕒 Waiting | - | 1 |

🚀 dzsquared requested your review to deploy to ProductionDeployment

2

Review deployments

publish.yml

on: workflow_dispatch



🔄 - +

Deployment protection rules

Reviewers, timers, and other rules protecting deployments in this run

Start all waiting jobs

| Event | Environments | Comment |
|----------------------------|----------------------|---------|
| dzsquared requested review | ProductionDeployment | - |

Artifacts

Produced during runtime

| Name | Size |
|--------------------------|---------|
| 📄 ReviewDeploymentScript | 1.75 KB |

📄 🗑️

1

Branch

Object Explorer

- localhost (SQL Server 17.0.1105.2 - REDMOND\drskw)
- Databases
 - System Databases
 - Database Snapshots
 - AdventureWorks
 - LoanApplicationApp
 - LoanApp-Prototype
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.account_ownership
 - dbo.accounts
 - dbo.borrowers
 - dbo.collaterals
 - dbo.credit_bureaus
 - dbo.credit_scores
 - dbo.income_verifications
 - dbo.loan_applications
 - Dropped Ledger Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Service Broker
 - Storage
 - Security
 - Server Objects
 - Replication
 - Always On High Availability
 - Management
 - SQL Server Agent (Agent XPs disabled)
 - XEvent Profiler

```
SQLQuery3.sql...rskwier (79) | PostDeployme...rskwier (54)
1 SELECT TOP (1000) [acct_id]
2     , [date_opened]
3     , [is_active]
4 FROM [LoanApp-Prototype].[dbo].[accounts]
5
```

100 % No issues found Ln: 1, Ch: 1 SPC CRLF UTF-8

Results Messages

| | acct_id | date_opened | is_active |
|----|---------|-----------------------------|-----------|
| 1 | 1 | 2025-01-03 00:00:00.0000000 | 1 |
| 2 | 2 | 2025-01-07 00:00:00.0000000 | 1 |
| 3 | 3 | 2025-01-10 00:00:00.0000000 | 1 |
| 4 | 4 | 2025-01-14 00:00:00.0000000 | 1 |
| 5 | 5 | 2025-01-17 00:00:00.0000000 | 1 |
| 6 | 6 | 2025-01-21 00:00:00.0000000 | 1 |
| 7 | 7 | 2025-01-24 00:00:00.0000000 | 1 |
| 8 | 8 | 2025-01-28 00:00:00.0000000 | 1 |
| 9 | 9 | 2025-01-31 00:00:00.0000000 | 0 |
| 10 | 10 | 2025-02-04 00:00:00.0000000 | 1 |
| 11 | 11 | 2025-02-07 00:00:00.0000000 | 1 |
| 12 | 12 | 2025-02-11 00:00:00.0000000 | 1 |

Query executed successfully. localhost (17.0 RTM) | REDMOND\drskwier (79) | LoanApp-Prototype | 00:00:00 | Row: 1, Col: 1 | 100 rows

Output

Show output from: Build

Build started at 10:36 PM...
Build has been canceled.
***** Elapsed 20.564 seconds *****

Solution Explorer

Search Solution Explorer (Ctrl+):

- Solution 'ZavaFinancialLoanApps' (1 of 1 project)
- ZavaFinancialLoanApps
 - dbo
 - StoredProcedures
 - EvaluateBorrowerRisk.sql
 - GetLatestCreditScores.sql
 - SubmitLoanApplication.sql
 - Tables
 - account_ownership.sql
 - accounts.sql
 - borrowers.sql
 - collaterals.sql
 - credit_bureaus.sql
 - credit_scores.sql
 - income_verifications.sql
 - loan_applications.sql
 - Views
 - vw_active_portfolio_overview.sql
 - vw_borrower_credit_summary.sql
 - vw_loan_application_details.sql
 - vw_pending_income_verification.sql
 - PostDeploymentScript.sql
- Miscellaneous Files
 - SQLQuery3.sql

Explorer

Search

- LoanApplicationApp
 - dbo
 - Queries
 - My queries
 - ReferenceData
 - ReferenceData



Query, preview, or connect your data

Interact with your data by starting a query, creating database objects with a template, or connecting the database to your app with a GraphQL API. You can also preview your data by opening a table or view.

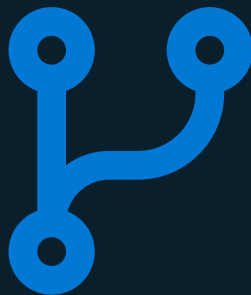
+ New

Stewardship through source control



Branching

- Create a new environment
- Apply the database configuration one or more times
- Standardized process ensures development is against "latest"
- Either create new or continually update the branch environment



Merging

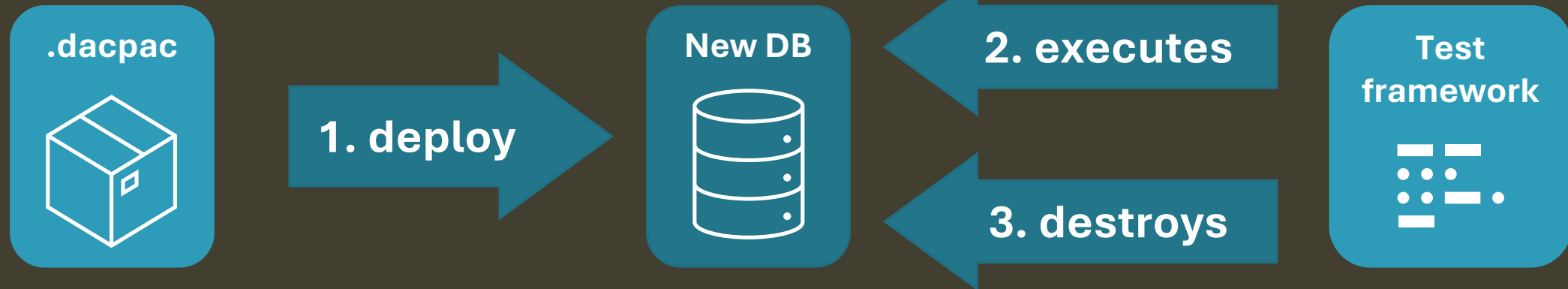
- Not optional!
- Changes from a branch can be examined, tested, and combined with the previous known good state
- Combine code, then apply to environments



Expand test coverage

Leverage ephemeral environments to execute app-integrated tests early and often

Test run




EXPLORER

- TESTCONTAINER-DEMO
 - aspire-apphost
 - console-app
 - data-api
 - database
 - tests
 - .gitignore
 - README.md
 - testcontainer-demo.sln

OUTLINE

TIMELINE

SOLUTION EXPLORER



Open Chat ^ ⌘ I

Show All Commands ⬆ ⌘ P

Open Settings ⌘ ,



Branch

Isolate database changes for safe parallel development

Build

Verify code for quality and correctness early and often

Commit

Represent your database objects as code, regardless of their origin



Ship

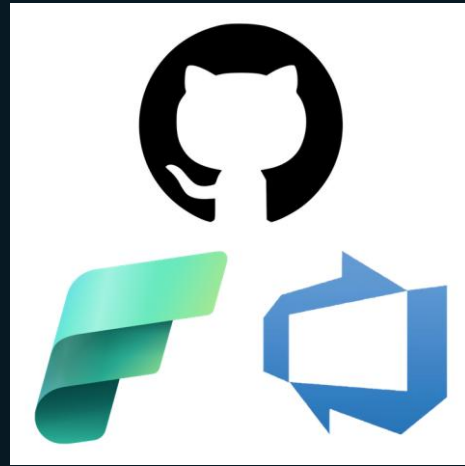
Plan, approve, and execute deployments no matter the diff

SQL projects tooling ecosystem



**SSMS, VS Code,
and more**

Develop, analyze, and
compare database
objects



**GitHub and Azure
DevOps tasks**

Streamline SQL project
deployment from
CI/CD environments



SqlPackage CLI

Automate transposing
between files and
databases



**API, class, and
model packages**

Extensibility for core
APIs



Roadmap: SQL projects

aka.ms/sqlprojects-roadmap

Project capabilities

- Pre-plan script
- Extract database directly to project (CLI)

Tooling

- SSMS schema compare
- VS Code - object rename/refactor

SQL database in Fabric

- Database properties in source control
- Copy data on workspace branch

Resources

1. <https://aka.ms/sqlprojects>
2. <https://aka.ms/sqlprojects-roadmap>
3. <https://aka.ms/sql-devops-workshop>