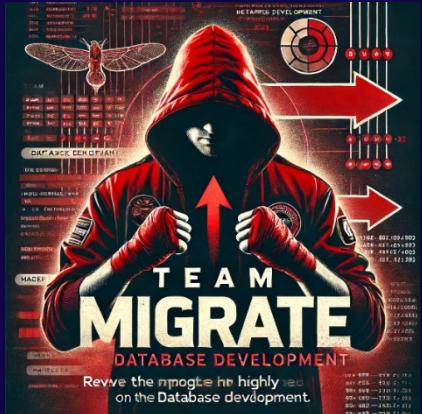
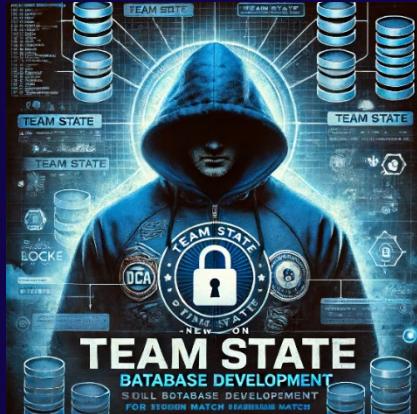


The Database DevOps (CI/CD) Showdown



VS



Tonie Huizer

He/him

Olivier van Steenlandt

He/him



State-based Deployment

Overview

Deploy to Development

- How to deploy?
- Build / Deploy
- Publishing Profiles
- Advanced Configuration

Lab: Deploy your changes

Manual Deployment

- SQL Server Data Tools (GUI)
- Command Line

Build / Deploy Phase

Build

- Validation of code (references, syntax,...)
- Copy build result based on configuration
- Result: .dacpac file

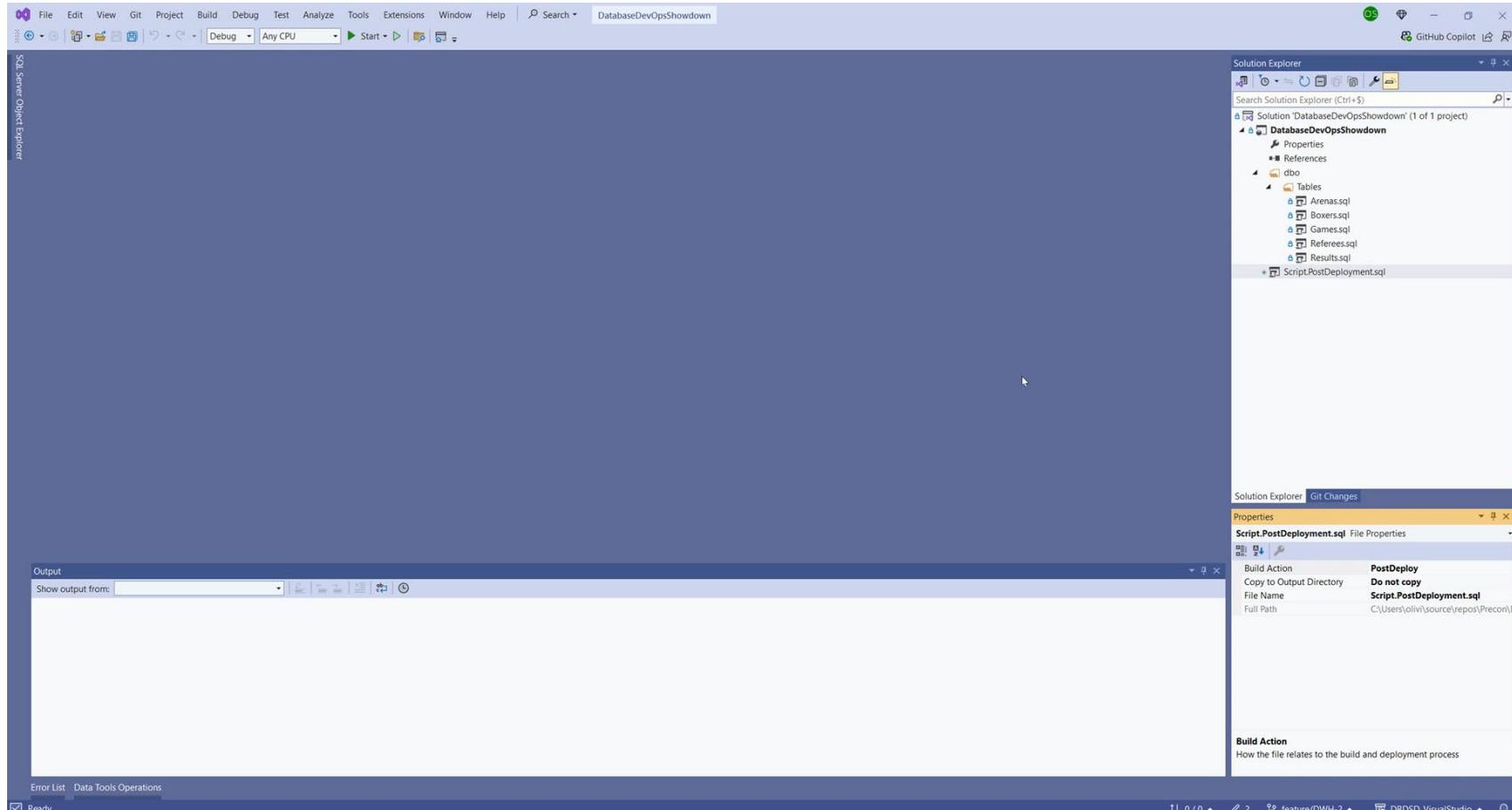
Deploy

- Input: .dacpac file

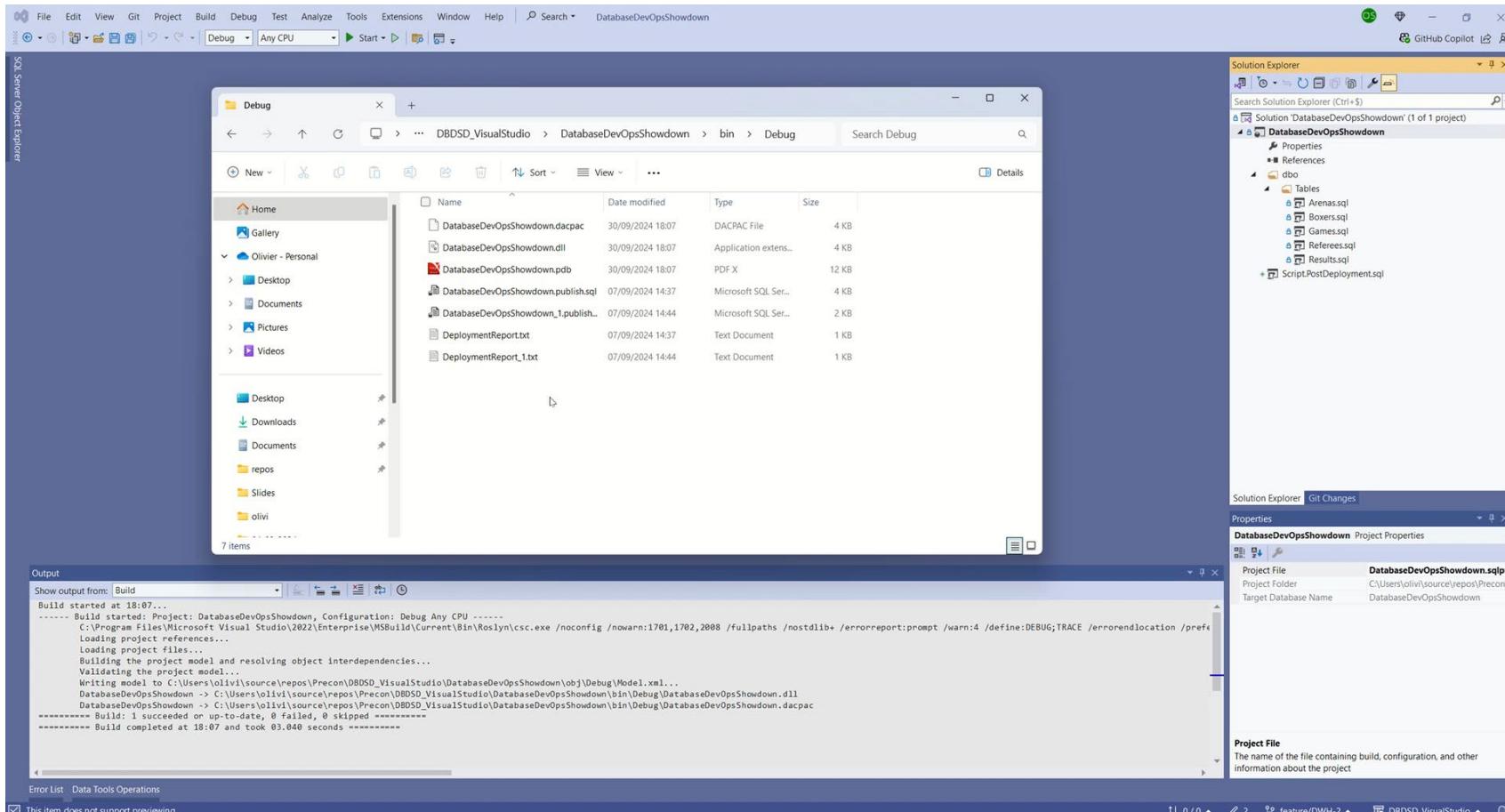
MSBuild (Build Phase)

- MSBuild Engine
- Platform for building applications
- In Visual Studio (GUI)
- MSBuild.exe
- dotnet build

MSBuild In Visual Studio (Build Phase)



.dacpac



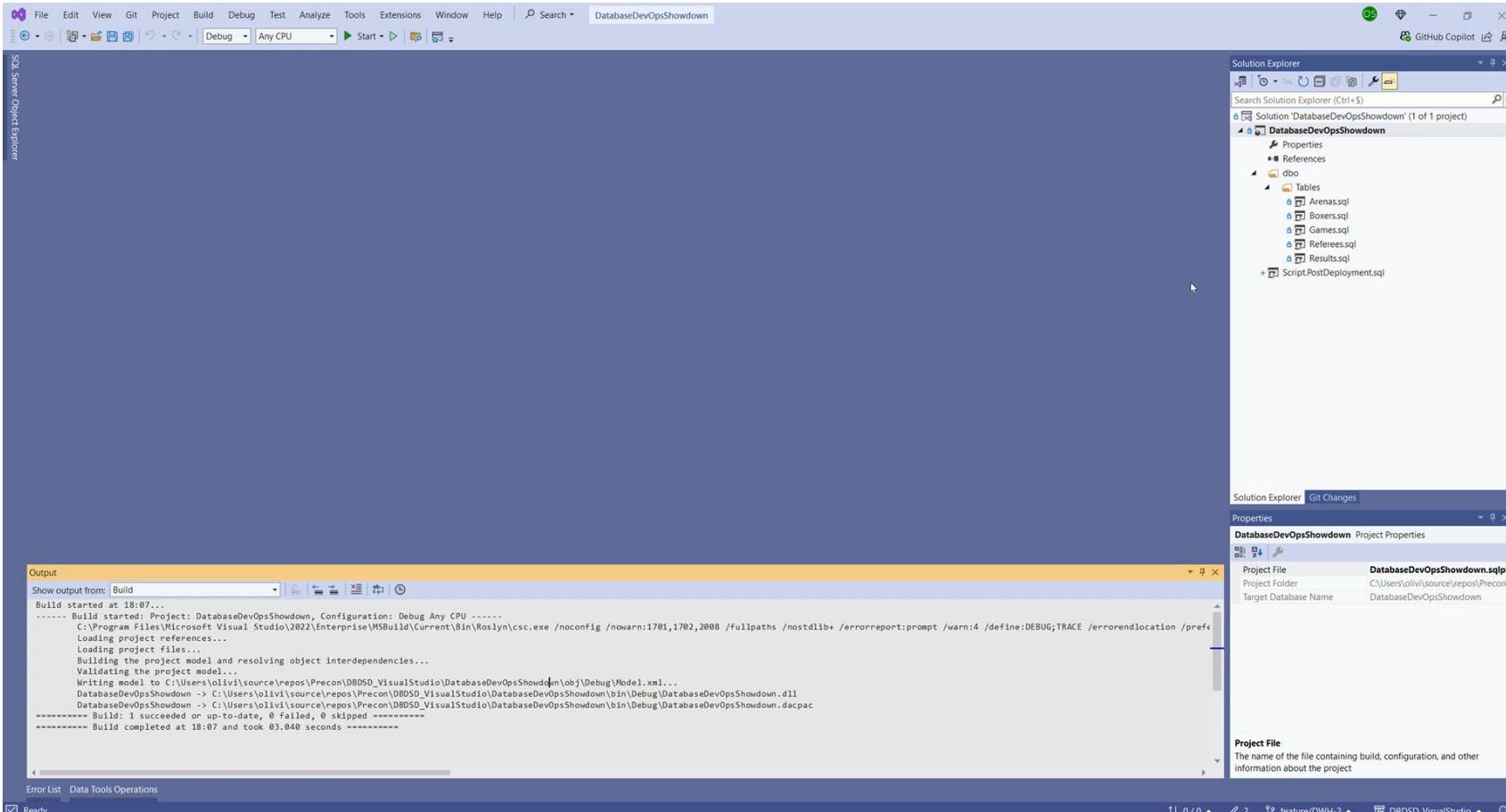
SqlPackage / DacFx (Deployment Phase)

- SqlPackage
 - CLI for DacFx Framework
 - Mechanism for database deployment
 - .dcpac / .bacpac
- In Visual Studio
- SqlPackage.exe

SqlPackage / DacFx (Deployment Phase)

- SqlPackage available for:
 - SQL Server
 - Azure SQL
 - Azure SQL MI
 - Azure Synapse Analytics
 - Microsoft Fabric*

SqlPackage / DacFx (Deployment Phase)



Publishing Profiles

- XML-file
- Deployment Properties

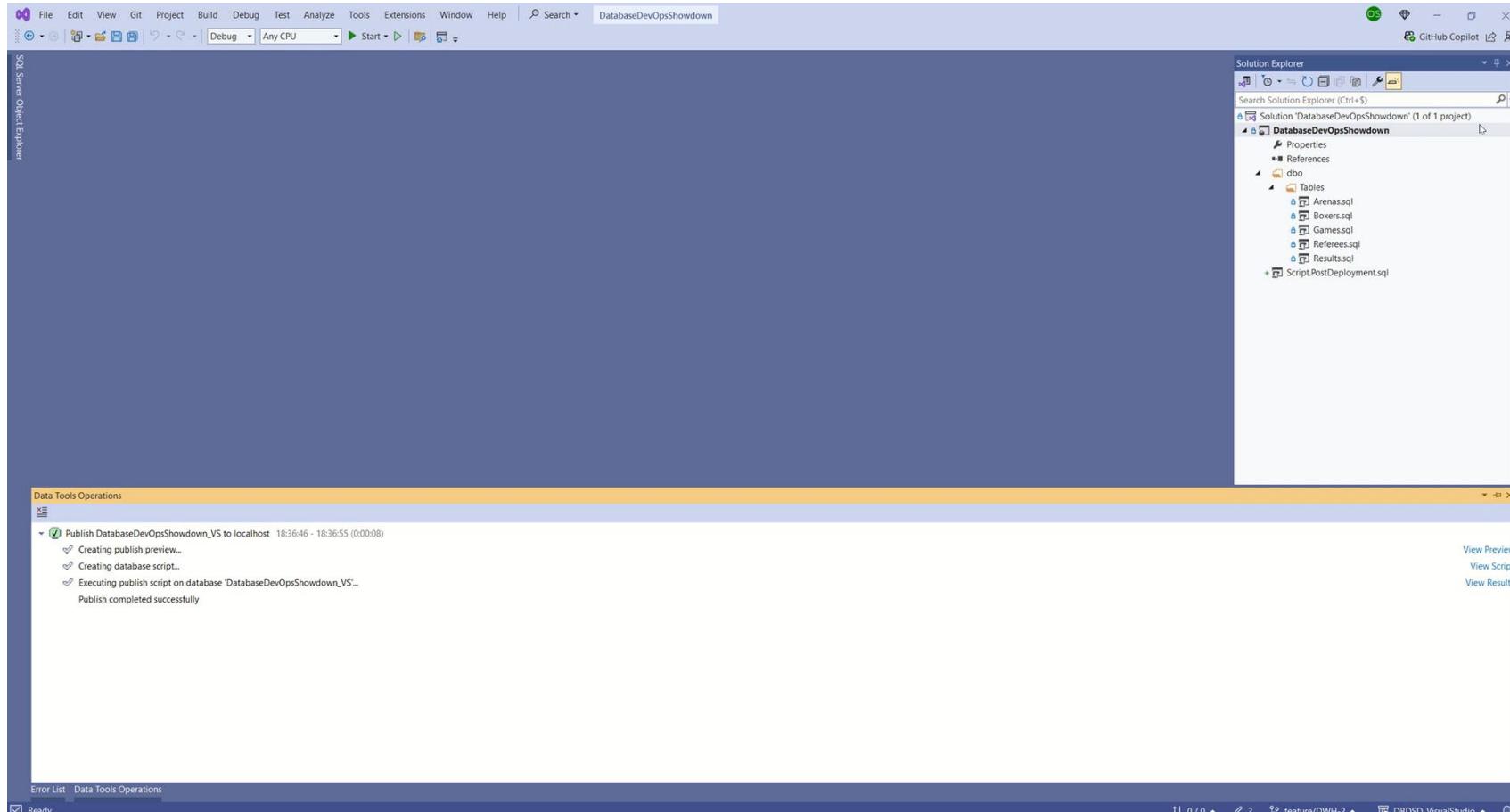
Advanced Deployment Configuration

- Variety of properties
- Most relevant:
 - Block if data loss...
 - Deploy database properties...
 - Drop indexes not in source...

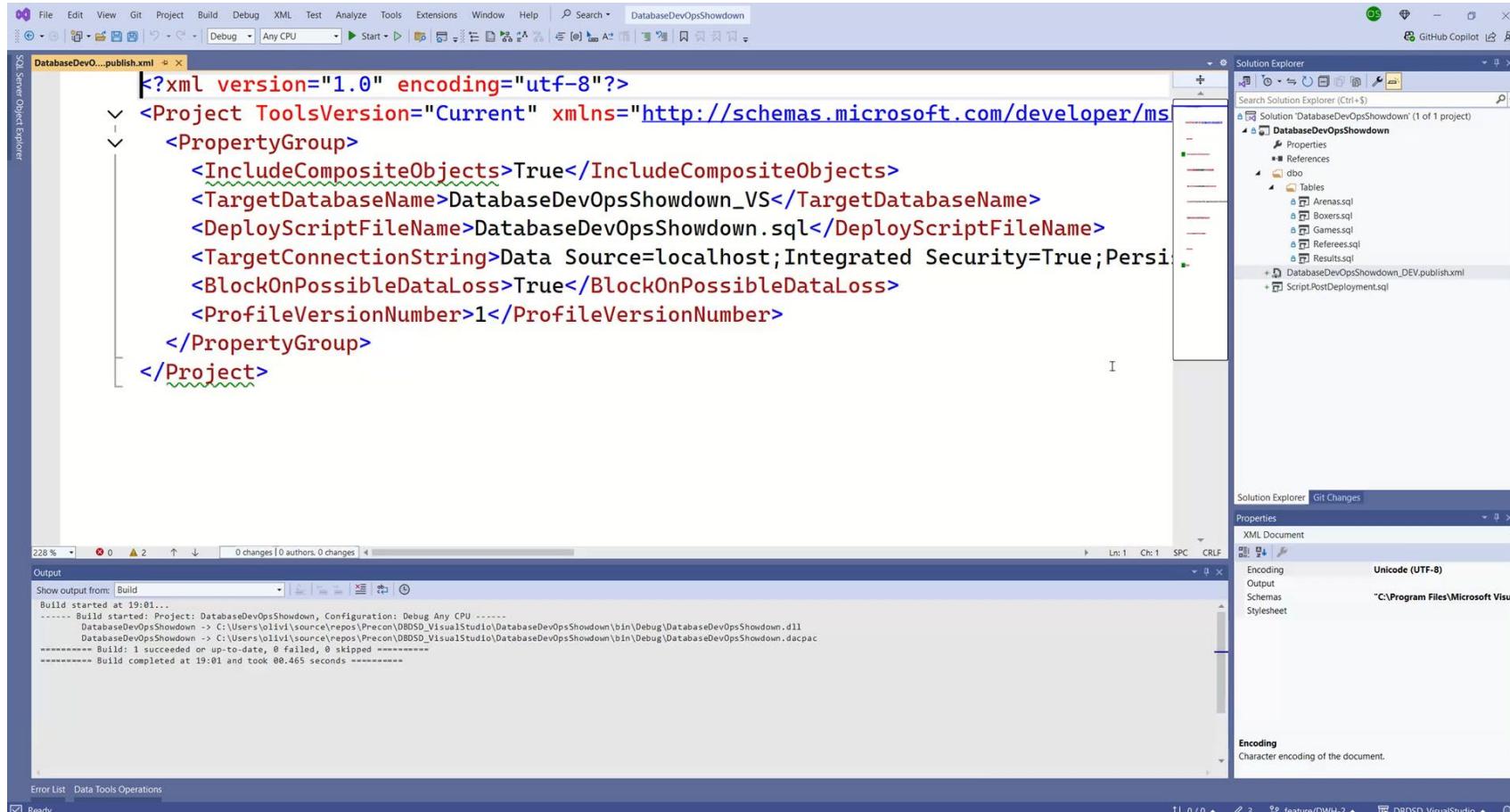
Tales from the trenches

- Deploy database properties...
- Drop indexes not in source...

Publishing Profile



Publishing Profile Usage



Lab

Deploy your changes

Migration-based Deployment

Overview

Deploy to Development

- How to deploy?
- Build vs Deploy
- Deploy
- Advanced Configuration

Lab 3: Deploy your changes

How to manual Deploy

- Flyway Desktop
- Flyway Command Line
 - Grate
 - EF Core

Manual Build vs Deploy Phase

Build

- Flyway Desktop doesn't build
- **Immutable** migrations already created
 - in dev phase (round 2)

Deploy

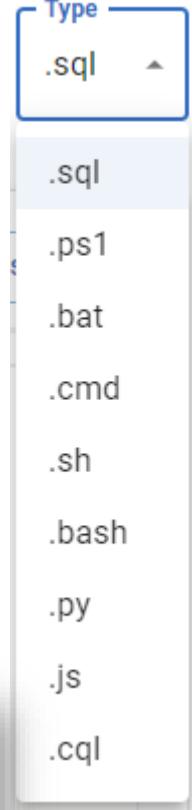
- Input: migrations files (immutable)

The “pre” build step for Flyway Enterprise

- Create a migration
 - With Flyway Desktop
 - Hand crafted
 - SQL and beyond
 - Migration types



Migration Type	Category	Example
SQL	Versioned	V2__Add_new_table.sql
SQL	Versioned (Undo)	U2__Add_new_table.sql
SQL	Repeatable	R__Add_new_table.sql
Java	Versioned	V2__Add_new_table.java
Java	Repeatable	R__Add_new_table.java
Script	Versioned	V1__execute_batch_tool.ps1
Script	Repeatable	R__execute_batch_tool.ps1



The “build” step for Flyway Community / Grate

- Create a migration
 - Hand crafted
 - Via comparison tools; like OpenDbDiff
- Migration types

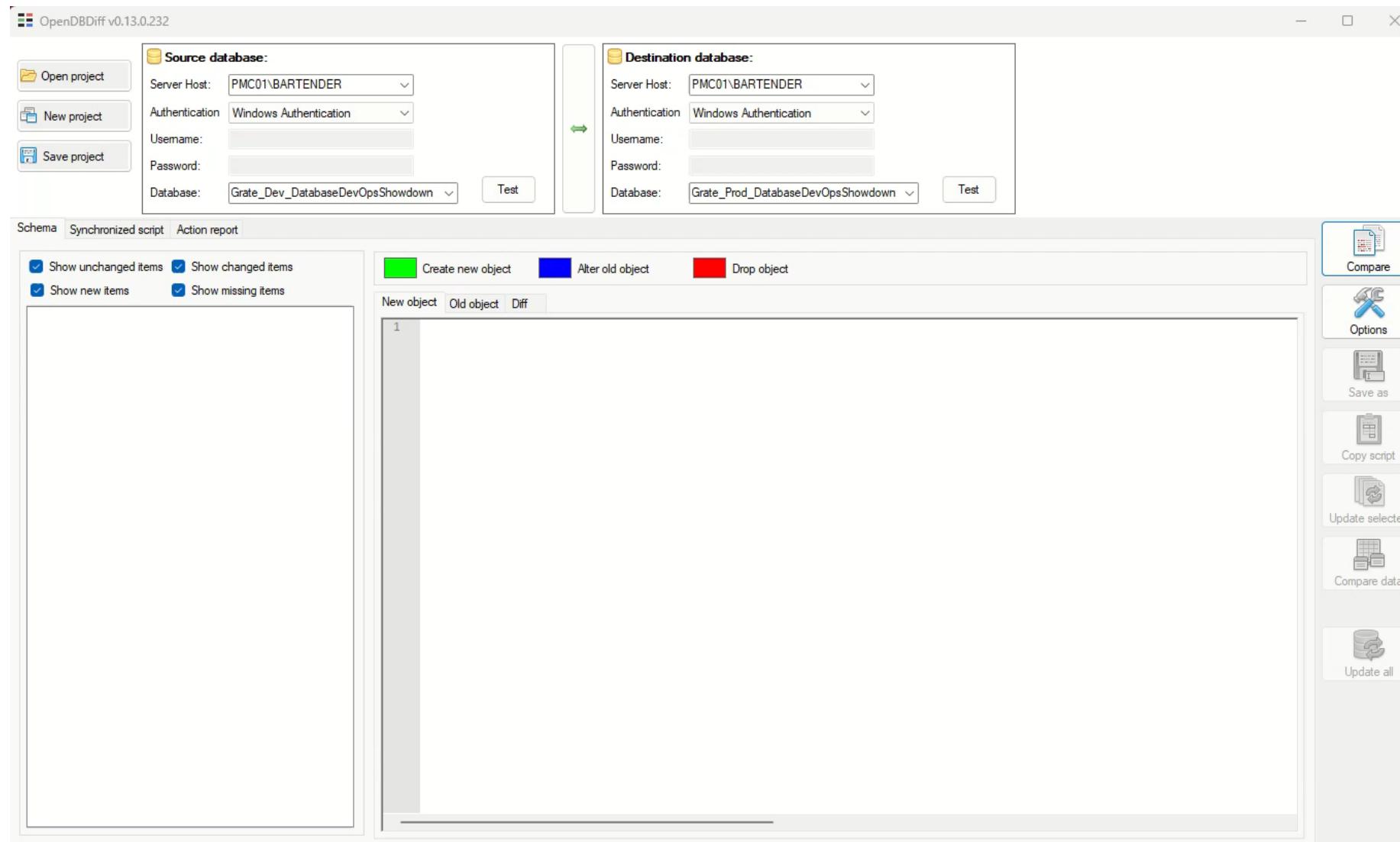


Migration Type	Category	Example
SQL	Versioned	V2__Add_new_table.sql
SQL	Versioned (Undo)	U2__Add_new_table.sql
SQL	Repeatable	R__Add_new_table.sql
Java	Versioned	V2__Add_new_table.java
Java	Repeatable	R__Add_new_table.java
Script	Versioned	V1__execute_batch_tool.ps1
Script	Repeatable	R__execute_batch_tool.ps1

grate has three different types of scripts.

- [Anytime Scripts](#)
- [Everytime Scripts](#)
- [One-Time Scripts](#)

OpenDBDiff - Generate migration files



#PASSDataSummit

Alternative looks on building a Flyway DB



Phil Factor

27 June 2022

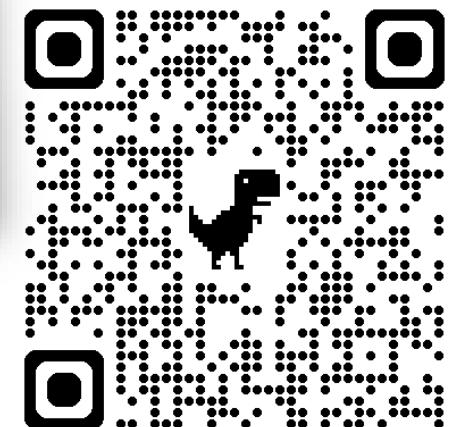
Guest post

This is a guest post from Phil Factor. Phil Factor (real name withheld to protect the guilty), aka Database Mole, has

FLYWAY DATABASE BUILD TASKS DATABASE VERSIONING

Building a Database with Flyway

Flyway, especially Flyway Teams edition, can be used in several different ways to accommodate a database development that was originally based on builds rather than migrations. This article explores four different ways to use Flyway to build a particular version of a database, from the ground up, using a single migration script. It should help teams select the best way to incorporate Flyway into an existing database build system, during development, while benefitting from use of Flyway's versioned migration system for deployments and releases.



Deploy With Flyway

Setting up target environments

The screenshot shows the Flyway Enterprise application interface. The main window displays a list of migrations in the 'devopsshowdown_ent' database. The table includes columns for Category, Version, Description, Type, Undoable, Date migrated, and State. There are four pending migrations listed:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASE...	X	-	Pending
Versioned	002.20241006161137	Ad...	SQL	✓	-	Pending
Versioned	003.20241006191313	Ad...	SQL	✓	-	Pending
Versioned	004.20241006201052	Re...	SQL	✓	-	Pending

A modal dialog is open on the right, titled 'topic/AddReferee'. It shows a dropdown for 'Development database...' and a button 'Configure target database...'. Below this, it says '4 pending migrations' and has a 'Run validate' button. At the bottom, there's a link to 'View Flyway configuration settings'.

First “Activate” Flyway on production

The screenshot shows the Flyway Enterprise UI interface. The top navigation bar includes the Flyway logo, 'Enterprise', and the database name 'devopsshowdown_ent'. The main left panel displays a table of migrations:

Category	Version	Description	Type	Undoable	Deploy Rules
Baseline	001.20241006171030	baseli...	SQL_BASE...	X	
Versioned	002.20241006161137	AddRe...	SQL	✓	
Versioned	003.20241006191313	AddRe...	SQL	✓	
Versioned	004.20241006201052	Remo...	SQL	✓	

A modal window titled 'Flyway' is open in the center, containing the following text and options:

Connect to a test or development environment to...

- See what scripts have already been run against a target
- Preview a deployment script
- Migrate pending scripts to a database to test the build ahead of deployment

Target database: Please select... + Manage target databases

Migrate all

The screenshot shows the Flyway Enterprise application interface. The top navigation bar includes the Flyway logo, 'Enterprise' status, and the database name 'devopsshowdown_ent'. The main left panel displays a table of migrations:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASE...	X	-	Ignored (B)
	001.20241006171030	<< ...	BASELINE	X	2024-10-07 14:...	Baseline
Versioned	002.20241006161137	Ad...	SQL	✓	-	Pending
Versioned	003.20241006191313	Ad...	SQL	✓	-	Pending
Versioned	004.20241006201052	Re...	SQL	✓	-	Pending

The right panel shows a modal for the 'production' target database, indicating '3 pending migrations'. It includes buttons for 'View command (0 parameters)', 'View Dry Run script', 'Migrate' (dropdown), and 'Run migrate'. Below this is an 'Advanced settings' section with a link to 'View Flyway configuration settings'.

#PASSDataSummit

Migrate all – the result

The screenshot shows the Flyway Enterprise UI interface. On the left, there's a sidebar with icons for navigation, search, and configuration. The main area has tabs for 'Flyway' (selected) and 'Enterprise'. Below that, a database connection named 'devopsshowdown_ent' is selected. The central part of the screen displays a table of migration history:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASELINE	X	-	Ignored (I)
	001.20241006171030	<< ...	BASELINE	X	2024-10-07 14:43:35	Baseline
Versioned	002.20241006161137	Ad...	SQL	✓	2024-10-12 13:53:08	Success
Versioned	003.20241006191313	Ad...	SQL	✓	2024-10-12 13:53:08	Success
Versioned	004.20241006201052	Re...	SQL	✓	2024-10-12 13:53:08	Success

To the right of the migration history, a modal window titled 'topic/AddReferee' is open, showing the target database is set to 'production'. It also indicates '0 pending migrations' and provides buttons for 'View command (0 parameters)', 'View Dry Run script', 'Migrate', and 'Run migrate'. A link to 'Advanced settings' and 'View Flyway configuration settings' is also present.

Undo a migration

The screenshot shows the Flyway Enterprise interface for the database 'devopsshowdown_ent'. The left pane displays a table of migrations:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASELINE	X	-	Ignored (B)
	001.20241006171030	<< ...	BASELINE	X	2024-10-07 14:...	Baseline
Versioned	002.20241006161137	Ad...	SQL	✓	2024-10-12 13:...	Success
Versioned	003.20241006191313	Ad...	SQL	✓	2024-10-12 13:...	Success
Versioned	004.20241006201052	Re...	SQL	✓	2024-10-12 13:...	Success

A context menu is open over the fourth migration (version 004), showing the command:

```
flyway migrate -configFiles ="C:\Users\Tonie\source\devopsshowdown_ent\flyway.toml,C:\Users\Tonie\source\devopsshowdown_ent\flyway.user.toml" -workingDirectory="C:\Users\Tonie\source\devopsshowdown_ent" -environment=production
```

The menu includes options like 'View Dry Run script', 'Migrate' (selected), and 'Run migrate'.

Undo multiple migration (cherrypick)

The screenshot shows the Flyway Enterprise application interface. On the left, the migration history table lists several migrations:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASELINE	X	-	Ignored (B)
	001.20241006171030	<< ...	BASELINE	X	2024-10-07 14:...	Baseline
Versioned	002.20241006161137	Ad...	SQL	✓	2024-10-12 13:...	Success
Versioned	003.20241006191313	Ad...	SQL	✓	2024-10-12 13:...	Success
Versioned	004.20241006201052	Re...	SQL	-	2024-10-12 13:...	Undone
Undo	004.20241006201052	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Versioned	004.20241006201052	Re...	SQL	✓	-	Pending

On the right, a modal dialog titled "topic/AddReferee" is open, showing a dropdown for "production" and a message "Configure target database...". It displays "1 pending migration" and a "Run undo" button.

#PASS

Migrate just one version (cherry pick)

The screenshot shows the Flyway Enterprise interface for the database `devopsshowdown_ent`. The left pane displays a migration history table with columns: Category, Version, Description, Type, Undoable, Date migrated, and State. The right pane shows pending migrations for the `production` database.

Migration History:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASEL...	X	-	Ignored
	001.20241006171030	<< ...	BASELINE	X	2024-10-07 14:...	Baseline
Versioned	002.20241006161137	Ad...	SQL	-	2024-10-12 13:...	Undone
Versioned	003.20241006191313	Ad...	SQL	-	2024-10-12 13:...	Undone
Versioned	004.20241006201052	Re...	SQL	-	2024-10-12 13:...	Undone
Undo	004.20241006201052	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Undo	003.20241006191313	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Undo	002.20241006161137	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Versioned	002.20241006161137	Ad...	SQL	✓	-	Pending
Versioned	003.20241006191313	Ad...	SQL	✓	-	Pending
Versioned	004.20241006201052	Re...	SQL	✓	-	Pending

Pending Migrations:

- 3 pending migrations
 - View command (0 parameters)
 - View Dry Run script
 - Undo
 - Run undo
- Advanced settings
 - View Flyway configuration settings
 - View the parameters applied to undo

Migrate to a version (target)

The screenshot shows the Flyway Enterprise application interface. The top navigation bar includes the Flyway logo, Enterprise status, and the database name `devopsshowndown_ent`. The main area displays a migration history table and a detailed view of pending migrations.

Migration History Table:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASELINE	X	-	Ignored
	001.20241006171030	<< ...	BASELINE	X	2024-10-07 14:...	Baseline
Versioned	002.20241006161137	Ad...	SQL	-	2024-10-12 13:...	Undone
Versioned	003.20241006191313	Ad...	SQL	-	2024-10-12 13:...	Undone
Versioned	004.20241006201052	Re...	SQL	-	2024-10-12 13:...	Undone
Undo	004.20241006201052	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Undo	003.20241006191313	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Undo	002.20241006161137	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Versioned	002.20241006161137	Ad...	SQL	✓	2024-10-12 14:...	Success
Versioned	003.20241006191313	Ad...	SQL	✓	-	Pending
Versioned	004.20241006201052	Re...	SQL	✓	-	Pending

Pending Migrations:

- Configure target database... (production)
- 2 pending migrations
 - > View command (0 parameters) [View Dry Run script](#)
 - > Advanced settings
 - View Flyway configuration settings
 - View the parameters applied to migrate

Run Migrate button is highlighted in blue.

Undo to a version (target)

The screenshot shows the Flyway Enterprise application interface. The left pane displays a list of migrations with columns for Category, Version, Description, Type, Undoable, Date migrated, and State. The right pane shows a detailed view of a pending migration for the 'production' database.

Migration History:

Category	Version	Description	Type	Undoable	Date migrated	State
Baseline	001.20241006171030	ba...	SQL_BASE...	X	-	Ignored
	001.20241006171030	<< ...	BASELINE	X	2024-10-07 14:...	Baseline
Versioned	002.20241006161137	Ad...	SQL	-	2024-10-12 13:...	Undone
Versioned	003.20241006191313	Ad...	SQL	-	2024-10-12 13:...	Undone
Versioned	004.20241006201052	Re...	SQL	-	2024-10-12 13:...	Undone
Undo	004.20241006201052	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Undo	003.20241006191313	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Undo	002.20241006161137	UN...	UNDO_SQL	-	2024-10-12 14:...	Success
Versioned	002.20241006161137	Ad...	SQL	✓	2024-10-12 14:...	Success
Versioned	003.20241006191313	Ad...	SQL	✓	2024-10-13 10:...	Success
Versioned	004.20241006201052	Re...	SQL	✓	-	Pending

Pending Migration Dialog:

Configure target database... **production**

1 pending migration

View command (0 parameters) **Run migrate**

View Dry Run script

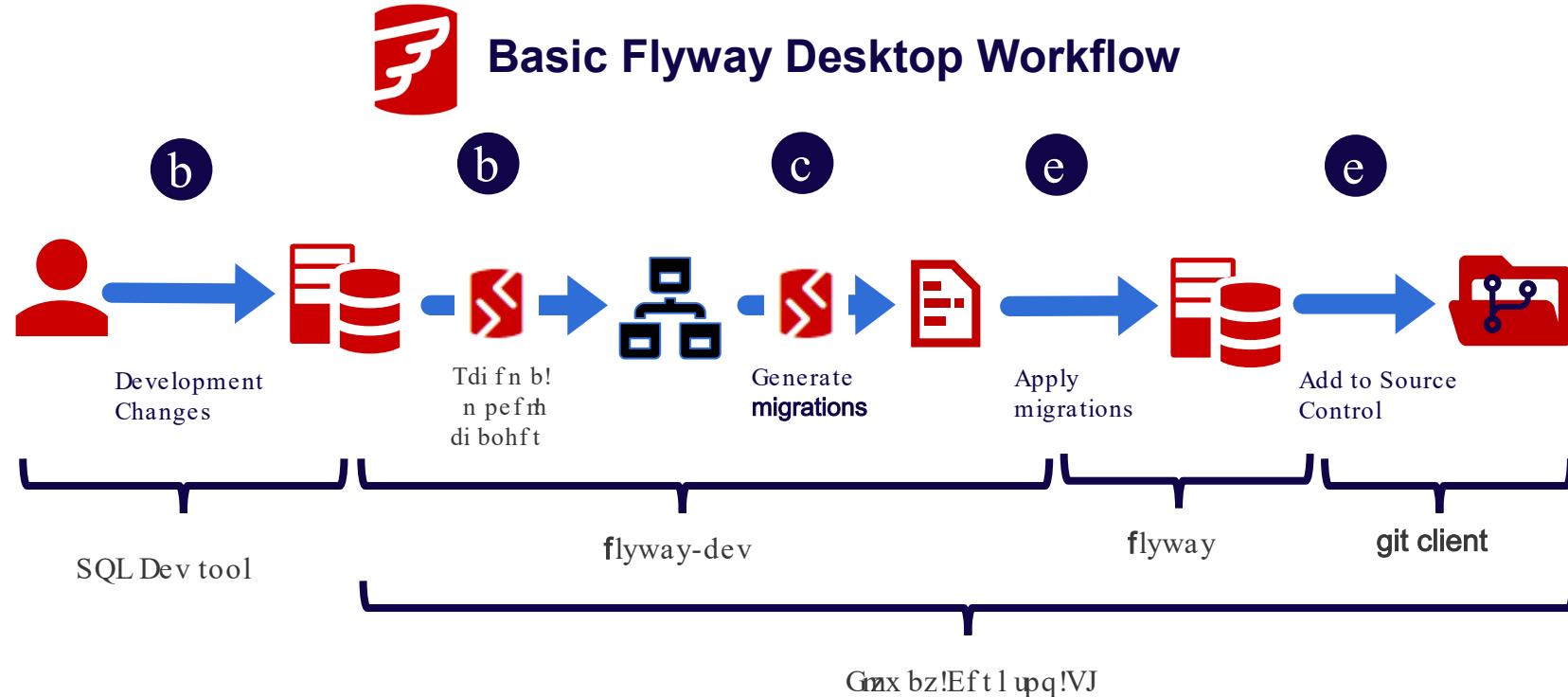
Migrate Advanced settings

View Flyway configuration settings
View the parameters applied to migrate

Flyway target vs cherrypick

- Execution Flow
 - `target`: applies all migrations up to a certain point
 - in order
 - `cherrypick`: allows you to choose specific version(s) to run
 - potentially out of order
- Use Cases:
 - `target`: is often used for controlled, stepwise update / rollback
 - `cherrypick`: applying /undoing specific migrations in isolated scenarios

What just happened?



Flyway CLI – Migrate

The screenshot shows the Flyway Enterprise interface. On the left, a sidebar has icons for Home, Database, Scripts, and Help. The main area has tabs for 'Flyway' (selected) and 'Enterprise'. A search bar includes a 'Search' input, a 'topic/AddReferee' dropdown, and a 'redgate' button. A message 'devopsshowdown_ent' is shown. A checkbox 'Only show pending migrations' is checked, and a note says 'Updated 2 minutes ago'. A table lists pending migrations:

Category	Version	Description	Type	Undoable	Date migrated	State
Versioned	002.20241006161137	Ad...	SQL	✓	-	Pending
Versioned	003.20241006191313	Ad...	SQL	✓	-	Pending
Versioned	004.20241006201052	Re...	SQL	✓	-	Pending

A modal dialog on the right is titled 'topic/AddReferee' and shows 'Configure target database...' with a dropdown set to 'production'. It displays '3 pending migrations' and a 'Run migrate' button. Other buttons include 'View command (0 parameters)', 'View Dry Run script', and 'Advanced settings'.

#PASSDataSummit

Flyway CLI

- flyway
 - migrate
 - -environment=production
 - -configFiles=
 - “C:\Users\Tonie\source\devopsshowdown_ent\flyway.toml,
 - C:\Users\Tonie\source\devopsshowdown_ent\flyway.user.toml”
 - -workingDirectory= “C:\Users\Tonie\source\devopsshowdown_ent”

Flyway CLI

- flyway
 - migrate
 - -environment:
 - -configFiles=
 - “C:\Users\Toni\Documents\flyway\config\mysql\script”
 - C:\Users\Toni\Documents\flyway\config\mysql\script
 - -workingDirectory=“C:\Users\Toniie\script”

- Auth
- Baseline
- Check Command
- Clean
- Info
- List Engines
- Migrate
- Repair
- Snapshot
- Undo
- Validate

Parameters

Page last updated 03 O 2024

Connection

- url
- user
- password
- driver
- connectRetries
- connectRetriesInterval
- initSql
- jdbcProperties

General

- batch
- callbacks
- cherryPick **Flyway Teams**
- configFileEncoding

Advanced Flyway Deployment Configuration

- Callbacks
 - Not only for migrate
- Useful for:
 - Data (un)loading
 - Pre / post processes
 - Custom RDBMS calls
- Grate alternative:
 - “Everytime” scripts

Migrate	Execution
beforeMigrate	Before Migrate runs
beforeRepeatables	Before all repeatable migrations during Migrate
beforeEachMigrate	Before every single migration during Migrate
beforeEachMigrateStatement	Before every single statement of a migration during Migrate
afterEachMigrateStatement	After every single successful statement of a migration during Migrate
afterEachMigrateStatementError	After every single failed statement of a migration during Migrate
afterEachMigrate	After every single successful migration during Migrate
afterEachMigrateError	After every single failed migration during Migrate
afterMigrate	After successful Migrate runs
afterMigrateApplied	After successful Migrate runs where at least one migration has been applied
afterVersioned	After all versioned migrations during Migrate
afterMigrateError	After failed Migrate runs

Lab

Deploy your changes

Deploy

With grate



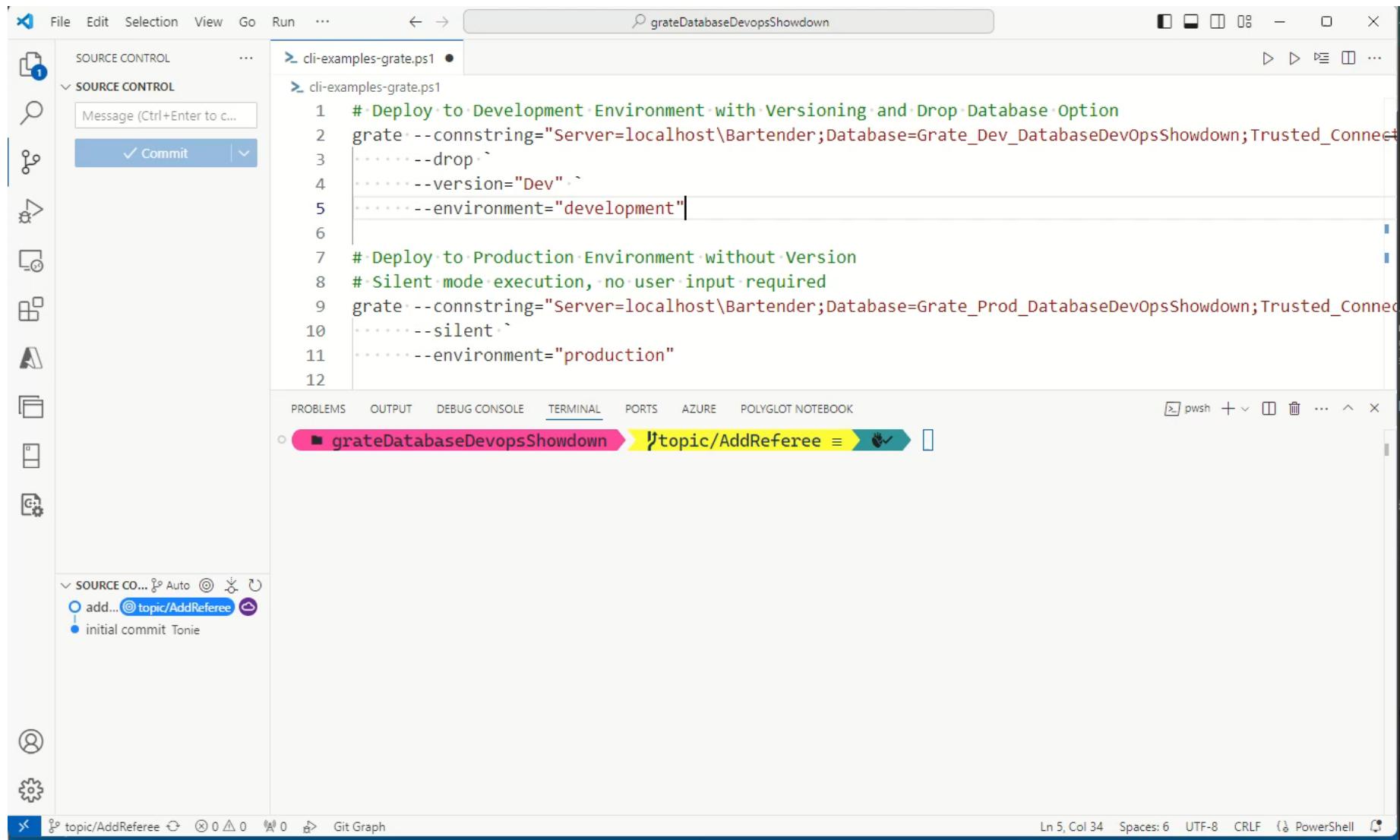
Deploy to production with version

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** grateDatabaseDevopsShowdown
- SOURCE CONTROL:** Shows 1 file changes, 1 commit pending.
- Editor:** A PowerShell script named `cli-examples-grate.ps1` containing the following code:

```
8 # Silent mode execution, no user input required
9 grate --connstring="Server=localhost\Bartender;Database=Grate_Prod_DatabaseDevOpsShowdown;Trusted_Connection=True"
10 .....--silent`-
11 .....--environment="production"
12
13 # Deploy to Production Environment with Specific Version
14 # Silent mode execution with version specified
15 grate --connstring="Server=localhost\Bartender;Database=Grate_Prod_DatabaseDevOpsShowdown;Trusted_Connection=True"
16 .....--silent`-
17 .....--version="Sprint 333" `-
18 .....--environment="production"
```
- Bottom Status Bar:** Shows the current file is `grateDatabaseDevopsShowdown`, the main tab is active, and the status message "Ln 14, Col 47 Spaces: 6 UTF-8 CRLF (PowerShell)".

Deploy to dev, but drop the db first



A screenshot of a terminal window in a development environment. The window title is "grateDatabaseDevopsShowdown". The code in the terminal is:

```
1 # Deploy to Development Environment with Versioning and Drop Database Option
2 grate --connstring="Server=localhost\Bartender;Database=Grate_Dev_DatabaseDevOpsShowdown;Trusted_Connection=True"
3     --drop
4     --version="Dev"
5     --environment="development"
6
7 # Deploy to Production Environment without Version
8 # Silent mode execution, no user input required
9 grate --connstring="Server=localhost\Bartender;Database=Grate_Prod_DatabaseDevOpsShowdown;Trusted_Connection=True"
10    --silent
11    --environment="production"
```

The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, AZURE, and POLYGLOT NOTEBOOK. The TERMINAL tab is selected. Below the terminal is a status bar showing "Ln 5, Col 34" and "Spaces: 6".

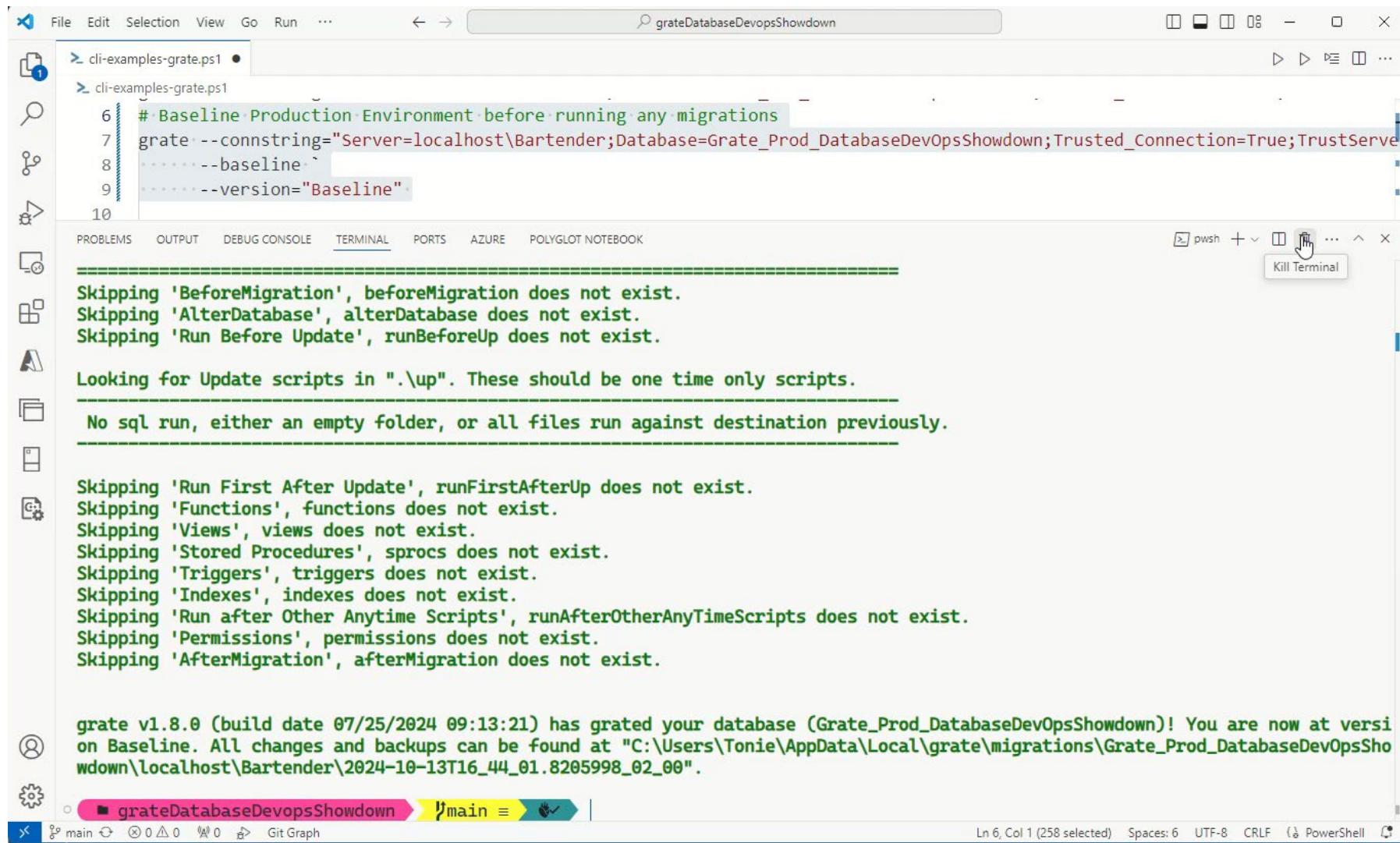
First “Activate” grate on production

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQLQuery12.sql - PMC01\Bartender.Grate_Prod_DatabaseDevOpsShowdown (PMC01\Tonie (65)) - Microsoft SQL Server Management Studio". The menu bar includes File, Edit, View, Query, Project, Tools, SQL Prompt, Window, and Help. The toolbar has various icons for file operations like New Query, Save, Print, and Database navigation. The Object Explorer on the left lists several databases, with "Grate_Prod_DatabaseDevOpsShowdown" expanded to show Tables, Views, and other objects. The main pane displays a T-SQL script:

```
***** Script for SelectTopNRows command from SSMS *****
SELECT TOP (1000) [ArenaID]
      ,[ArenaName]
      ,[Location]
      ,[Capacity]
   FROM [Grate_Prod_DatabaseDevOpsShowdown].[dbo].[Arenas]
```

The Results tab shows a table structure with columns Arenaid, ArenaName, Location, and Capacity, but no data rows. The status bar at the bottom indicates "Query executed successfully." and "0 rows".

First “Activate” grate on production



```
# Baseline Production Environment before running any migrations
grate --connstring="Server=localhost\Bartender;Database=Grate_Prod_DatabaseDevOpsShowdown;Trusted_Connection=True;TrustServerCertificate=True" --baseline --version="Baseline"

Skipping 'BeforeMigration', beforeMigration does not exist.
Skipping 'AlterDatabase', alterDatabase does not exist.
Skipping 'Run Before Update', runBeforeUp does not exist.

Looking for Update scripts in ".\up". These should be one time only scripts.

No sql run, either an empty folder, or all files run against destination previously.

Skipping 'Run First After Update', runFirstAfterUp does not exist.
Skipping 'Functions', functions does not exist.
Skipping 'Views', views does not exist.
Skipping 'Stored Procedures', sprocs does not exist.
Skipping 'Triggers', triggers does not exist.
Skipping 'Indexes', indexes does not exist.
Skipping 'Run after Other Anytime Scripts', runAfterOtherAnyTimeScripts does not exist.
Skipping 'Permissions', permissions does not exist.
Skipping 'AfterMigration', afterMigration does not exist.

grate v1.8.0 (build date 07/25/2024 09:13:21) has grated your database (Grate_Prod_DatabaseDevOpsShowdown)! You are now at version Baseline. All changes and backups can be found at "C:\Users\Tonie\AppData\Local\grate\migrations\Grate_Prod_DatabaseDevOpsShowdown\localhost\Bartender\2024-10-13T16_44_01.8205998_02_00".
```

Grate CLI

- **grate**

- **--connstring=**

```
"Server=localhost\Bartender  
howdown;Trusted_Connection=True"
```

- **--version=2.0**

- **--silent**

- **--drop**

Full Configuration

Option	Default	Purpose
-c -cs --connectionstring --connstring <connectionstring>	-	REQUIRED You now provide an entire connection string. Previous versions of Grate accepted a database name and connection string separately. This is no longer supported and will result in an error. Database names are obsolete.
-a -acs -csa --adminconnectionstring --adminconnstring <adminconnectionstring>	The value provided via --connectionstring, with the target database replaced with a database that can be assumed to be present. For example, "master" for SQL Server.	Used when creating a new database, rather than restoring one.
-f --files --sqlfilesdirectory <sqlfilesdirectory>	. (current directory)	The directory where your SQL scripts are located.
--folders	Default folders as described in Getting started	Folder configuration, see Folder configuration .
-o --output --outputPath <outputPath>	%LOCALAPPDATA%/grate	This is where everything related to the migration is stored, including backups, all items that ran, permission dumps, etc.
--accesstoken <token>	-	Specify an access token to use when connecting to a database.
-ct		

Lab

Deploy your changes

End of this Round

Who's the winner?

