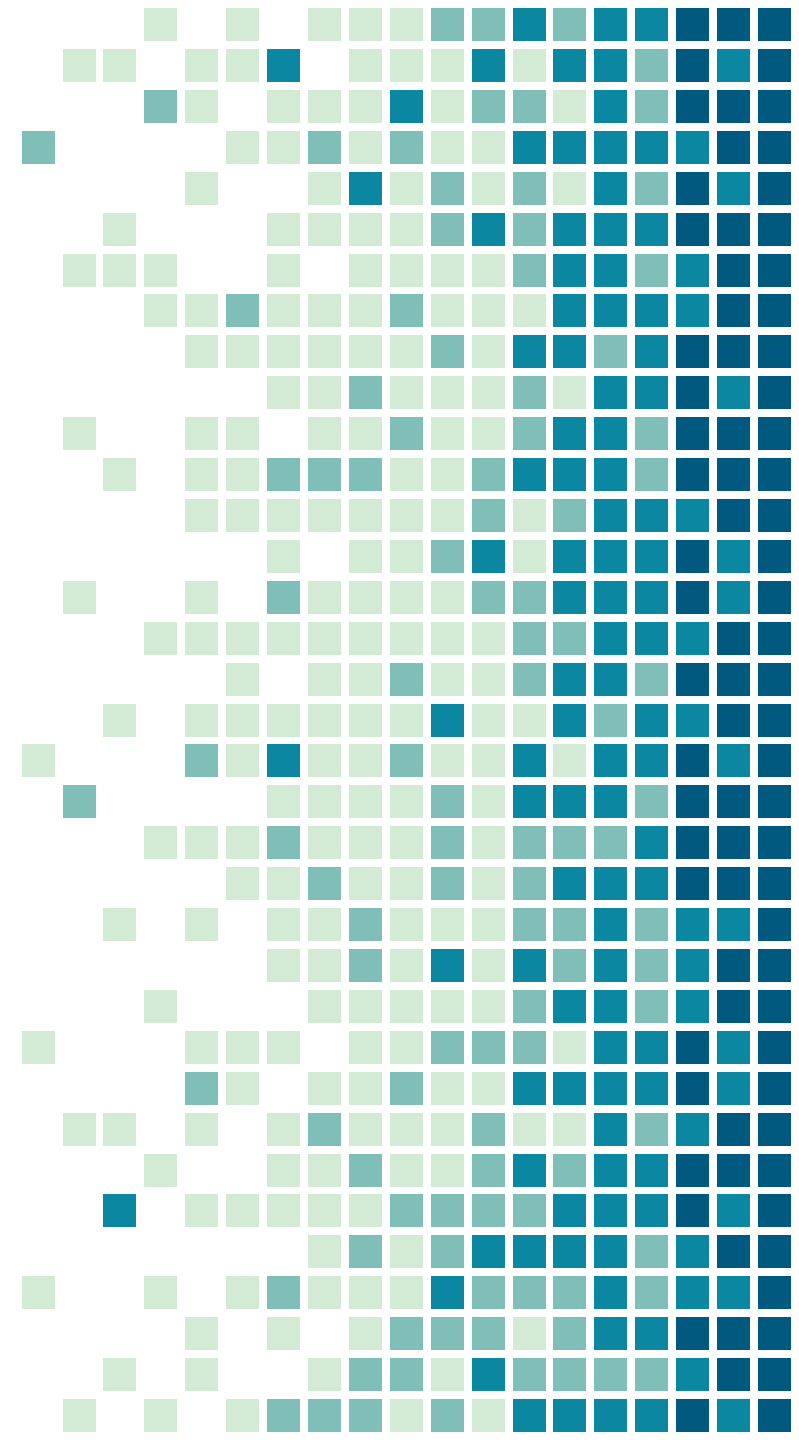


Database Deployment Methods Demystified

Deborah Melkin

NESQL

January 8, 2025



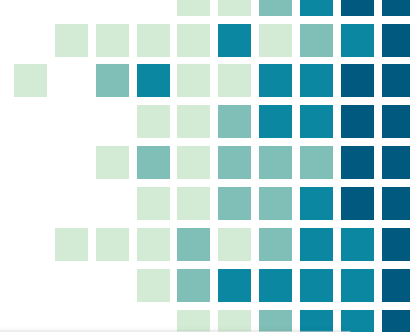
About Me

- 25 years as a DBA
- Mainly work with SQL Server
- Mainly work with OLTP
- Data Platform WIT co-leader
- WITspiration cofounder
- Redgate Community Ambassador
- Microsoft MVP – Data Platform

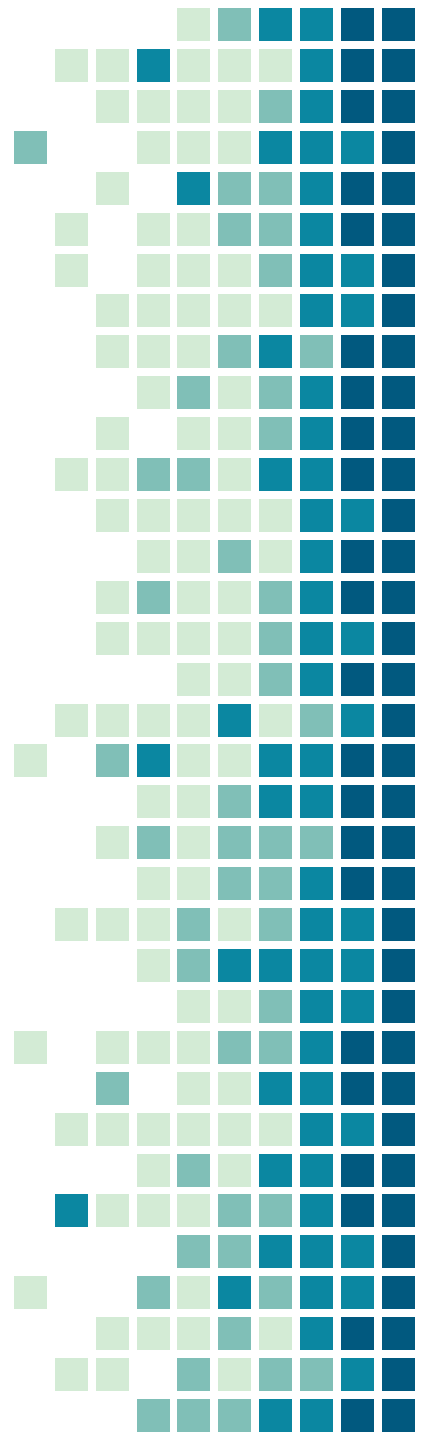


About Me – The Important Stuff

- I'm a long time member of the alto section.
- I go to bluegrass jams regularly.
- I've been learning guitar and now mandolin.
- I am a bit of a musical theater geek.
- My husband and I do geeky things with our dog.



Why Talk about Database Deployment Methods?

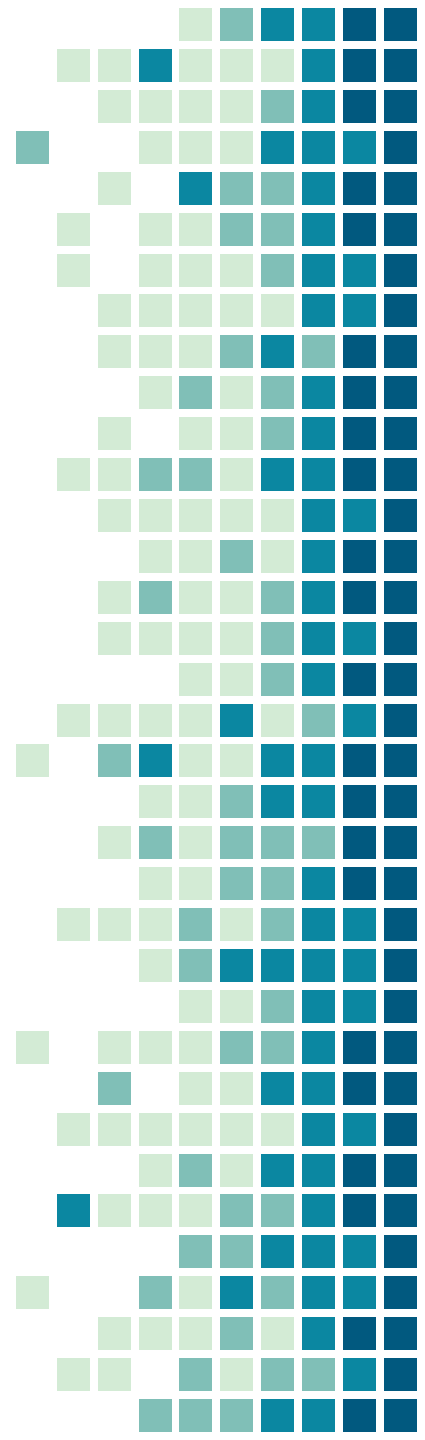


Not just “DevOps” ...

- We all have code to deploy
- This affects everyday workflow
- “DataOps”

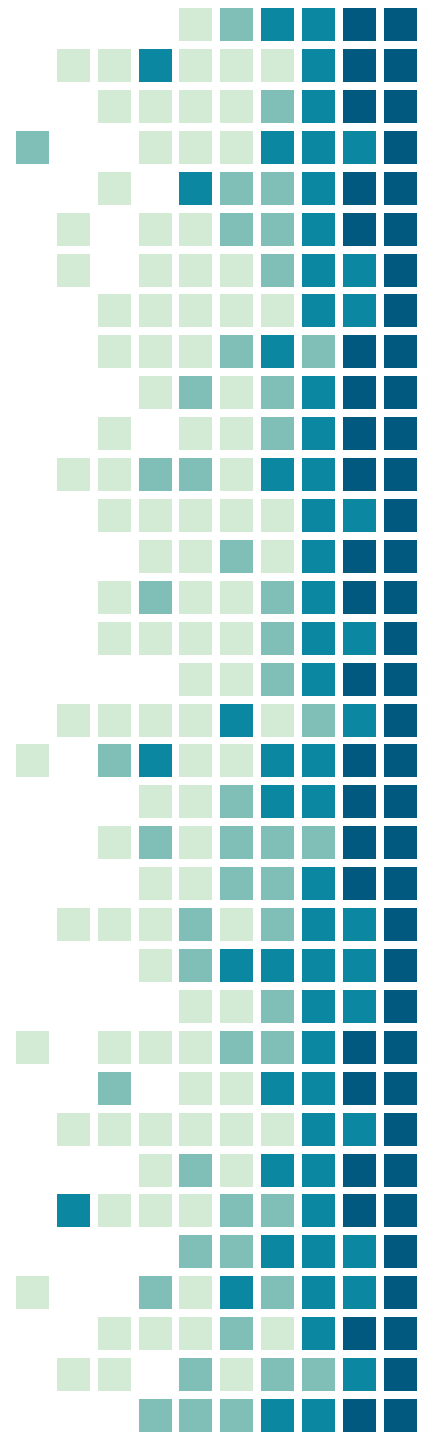
It doesn't matter whether you are Agile or Waterfall or
Waterfall pretending to be Agile...

if you have database changes, this matters to you!



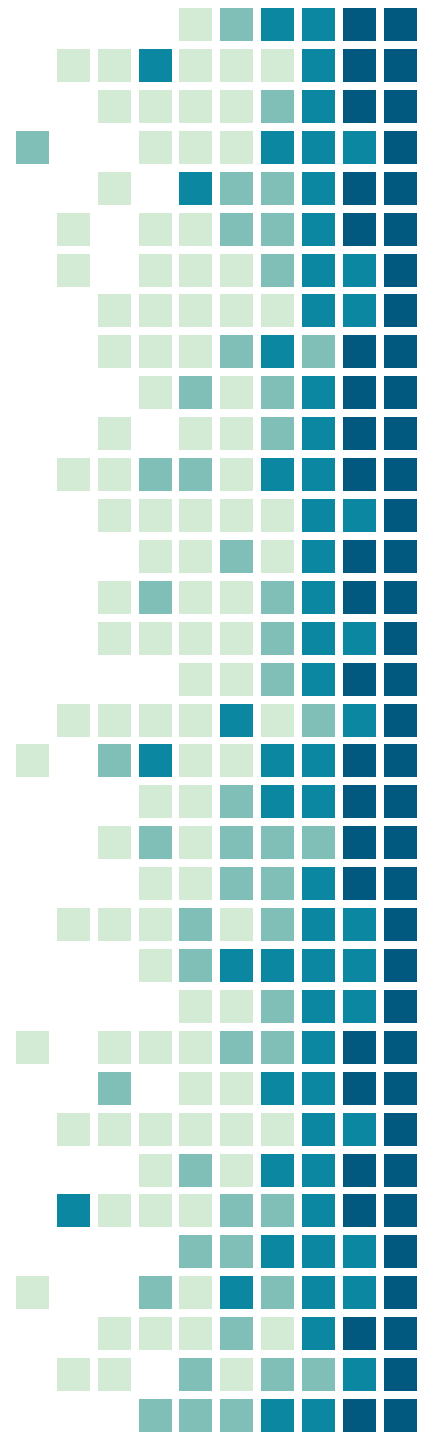
A little expectation setting ...

- This is not a How-To session
 - No Azure DevOps or GitHub Actions
- Skill Level – 200
 - There may be some things which seem “basic”
 - But there’s a lot that goes into what makes it at that level
 - Appreciation for the work that goes into making some of these things seem “basic”



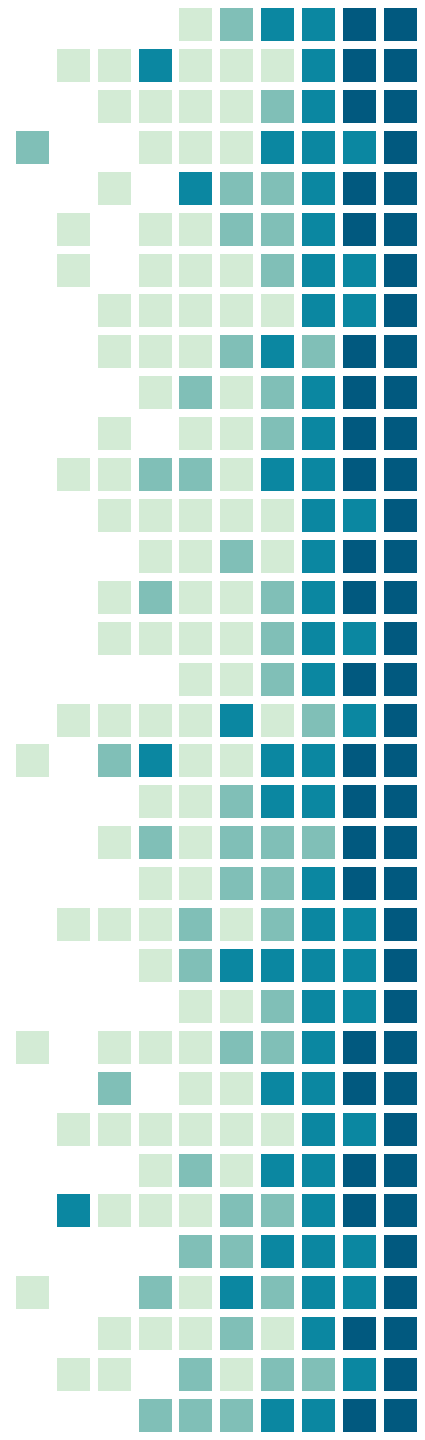
Related Terms

- “Source of Truth”
 - The artifact used to determine what the database should contain like depending on version or features that have been released.
- “Database as Code”
 - The idea that the database development should follow the same guidelines as the general Software Development Life Cycle (SDLC) principles.



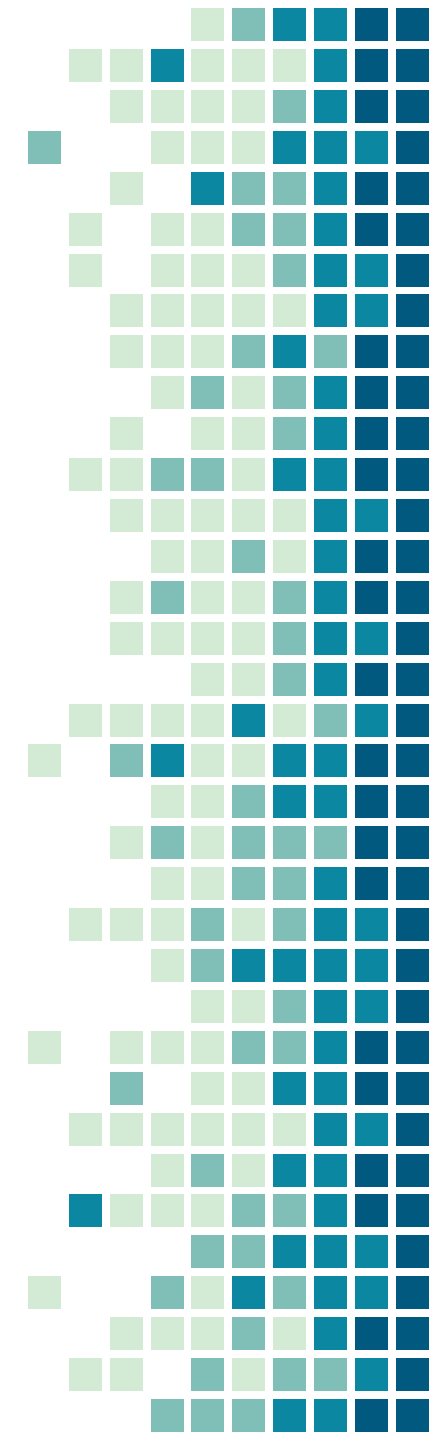
Requirements

- Should be able to create a new database or upgrade an existing one
- A newly created or upgraded existing database must be the same
 - Caveat: The upgraded database may have deprecated objects that have not been dropped. But strategy must exist for deprecation.
- Deployments must be re-runnable
 - "idempotent"

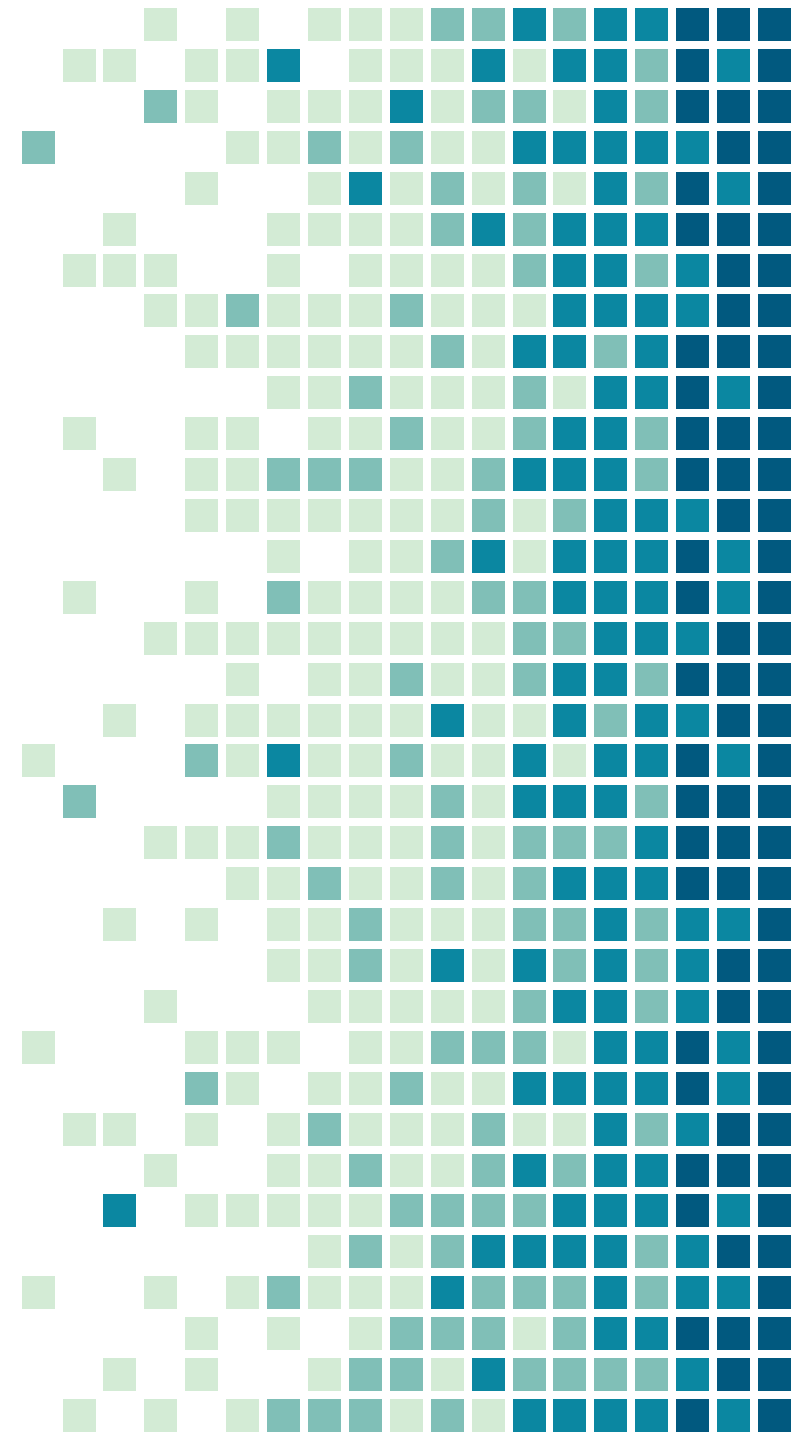


Types of deployments

- Migration based deployment
- State based deployment

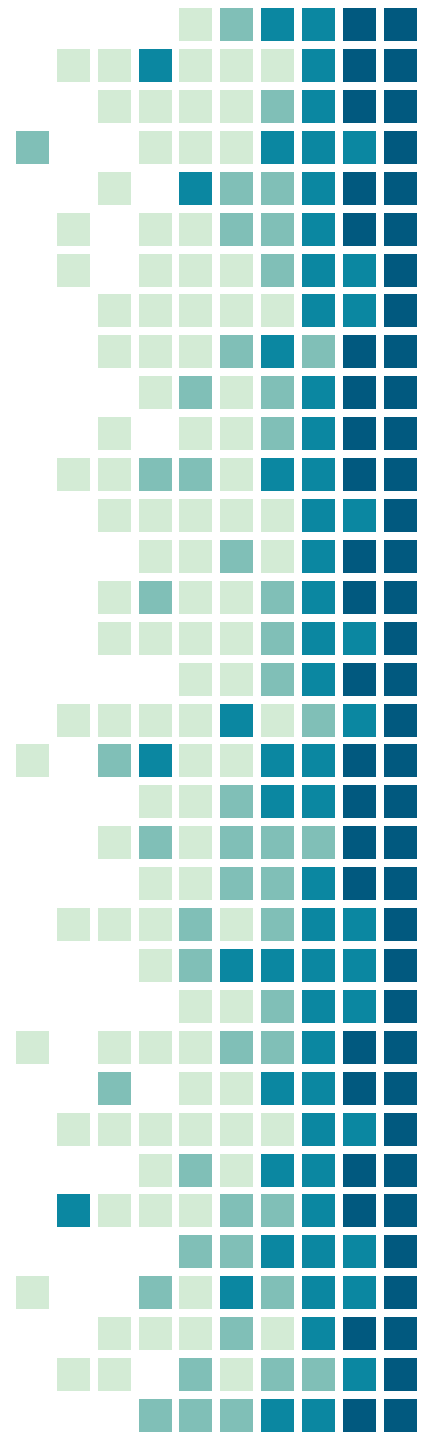


Migration Based Deployment



What is a Migration Based Deployment

- Series of scripts executed to deploy database changes
- The scripts are created manually (or mostly manually.)



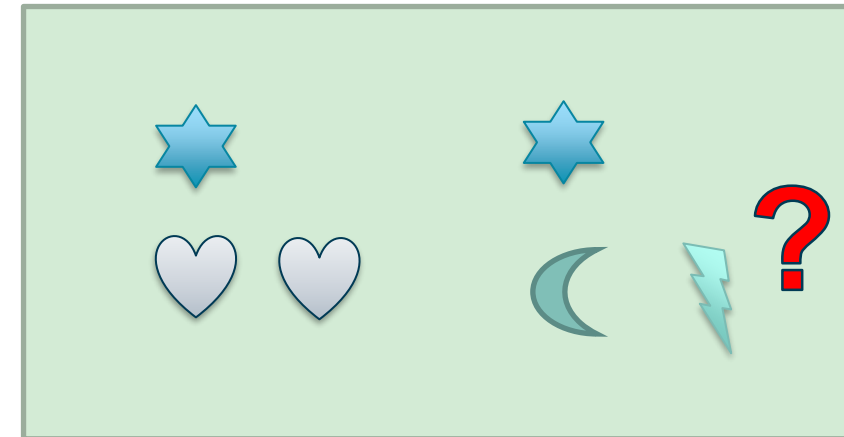


Source

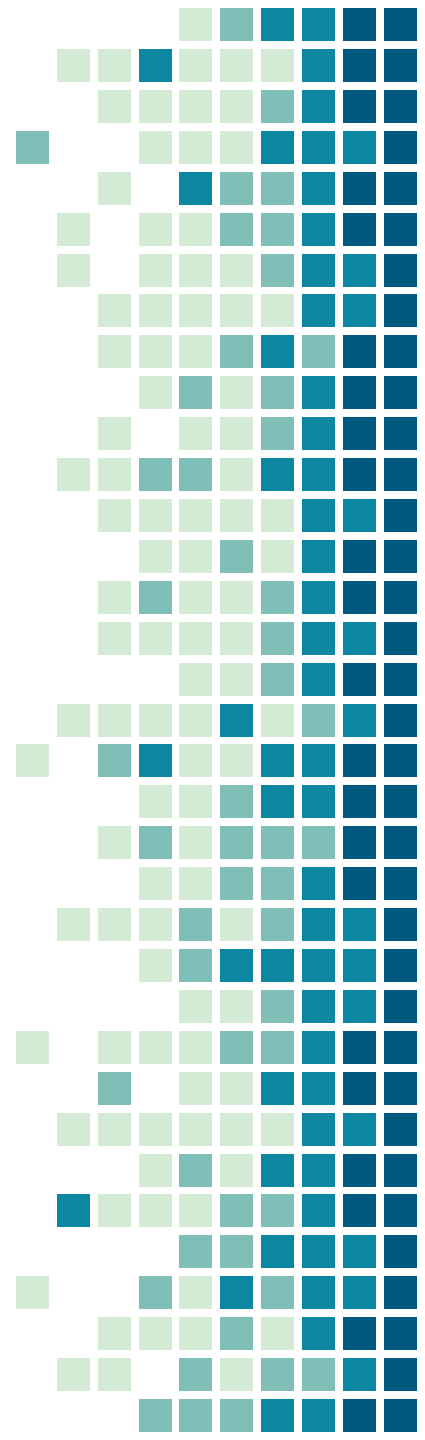


Target A

- ~~Add 1 Blue Star~~
Make sure there are 4 blue stars
- ~~Add 1 Pink Heart~~
Make sure there are 3 Pink Hearts
- ~~Add 1 Green Moon~~
Make sure there is 1 Green Moon

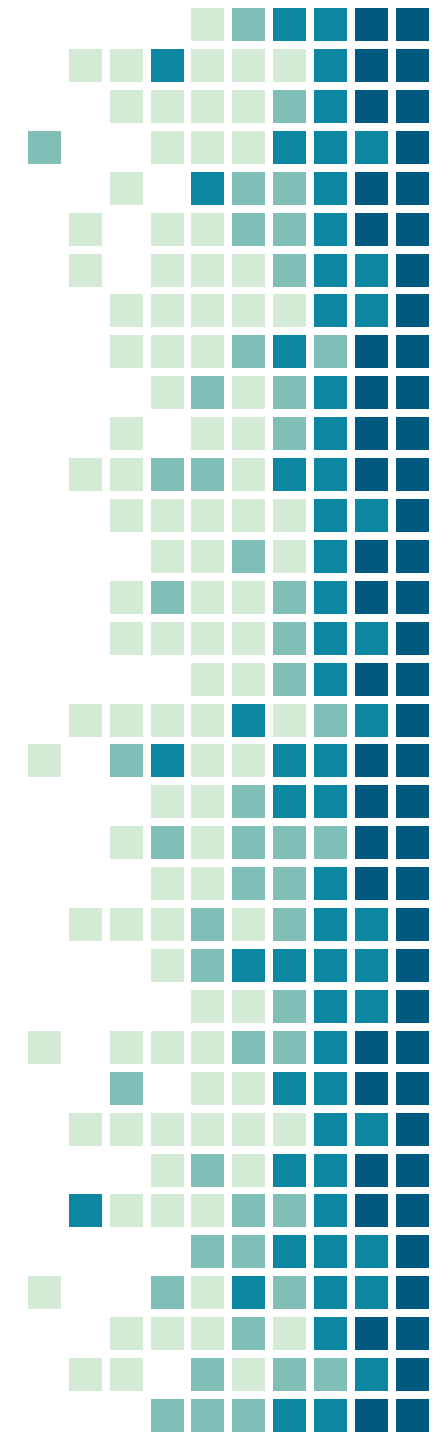


Target B

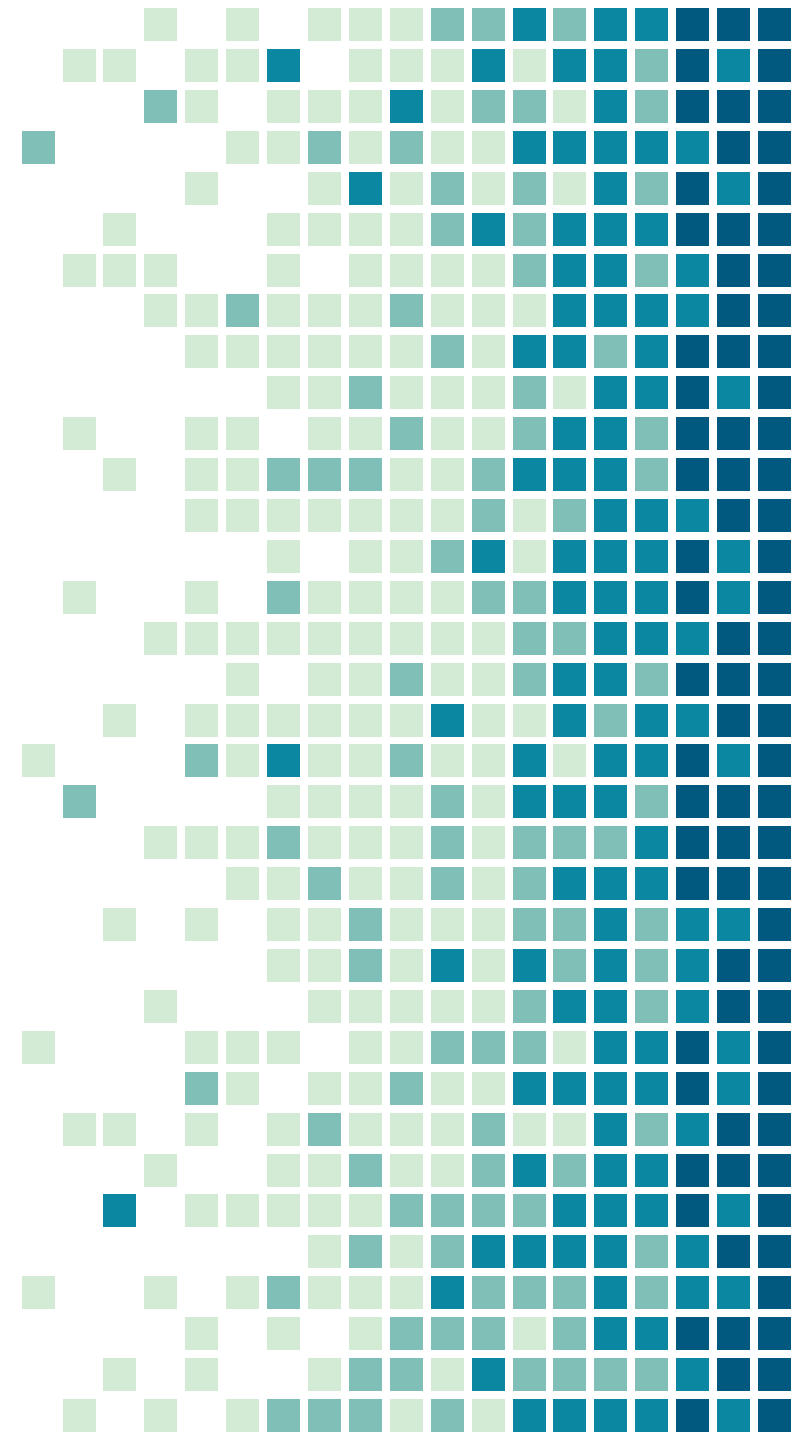


Tools that can be used:

- 3rd Party Tools:
 - Liquibase
 - Flyway
- Home Grown Tools:
 - PowerShell\dbatools
 - sqlcmd

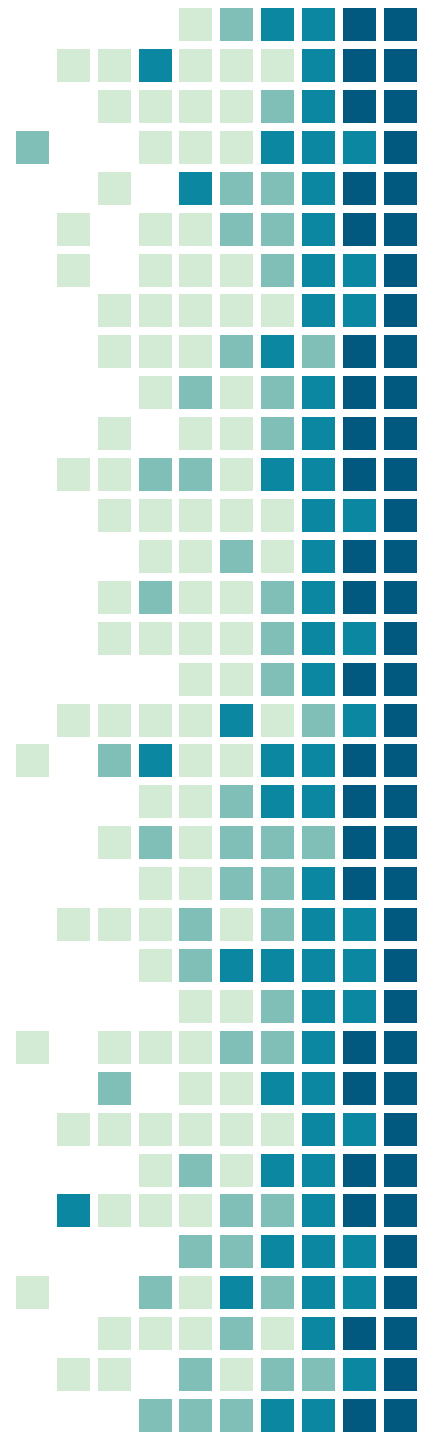


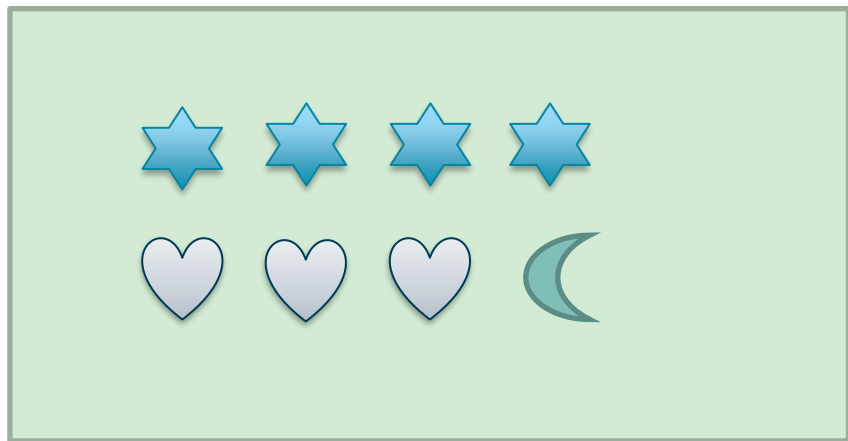
State Based Deployment



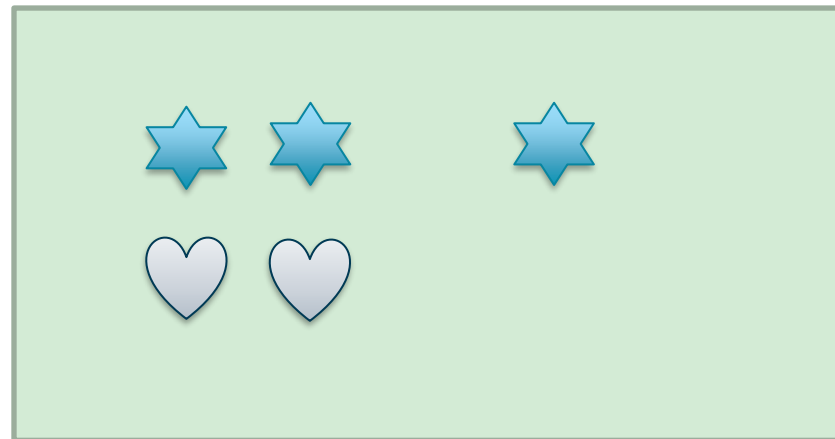
What is a State Based Deployment

- Comparison between target database and source that determines what the differences are, creates the SQL needed for the updates, then executes the changes
 - Examples: dacpac\bacpac

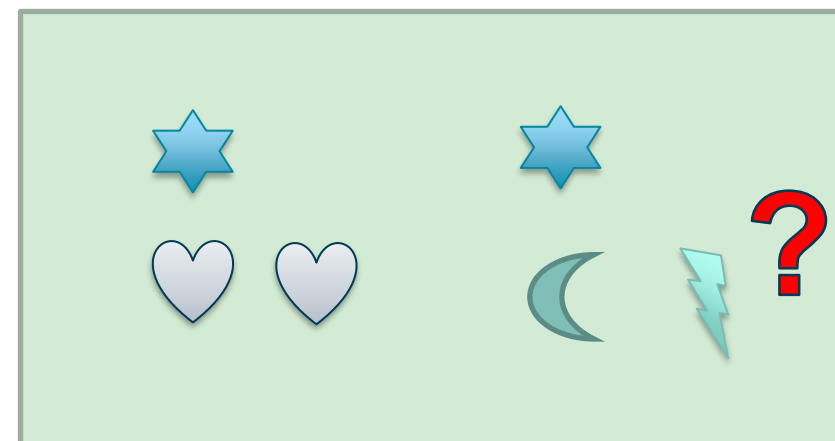




Source



Target A



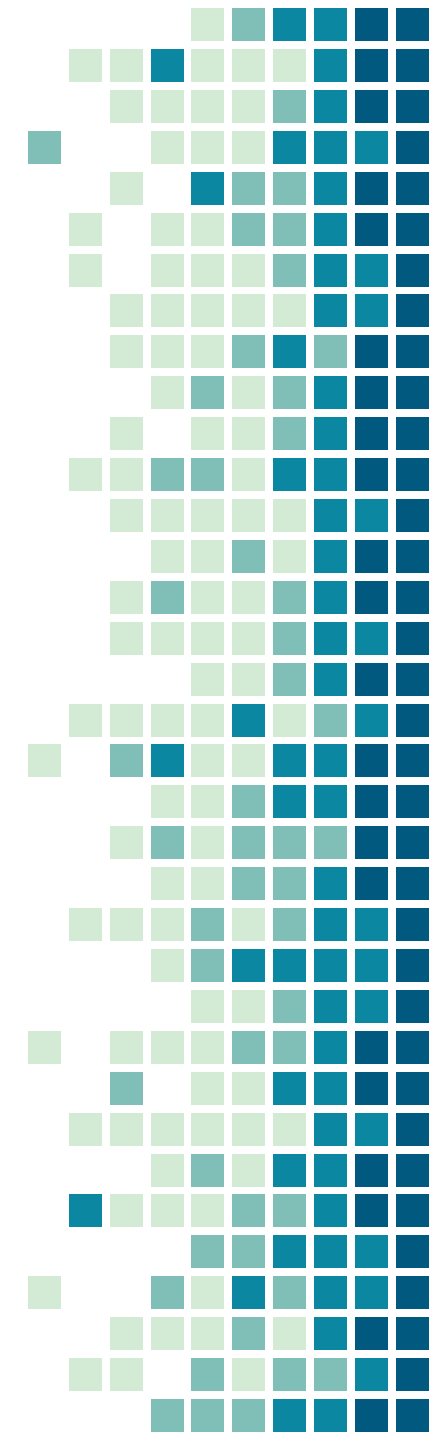
Target B

- Run Compare against Source and Target
- Apply changes



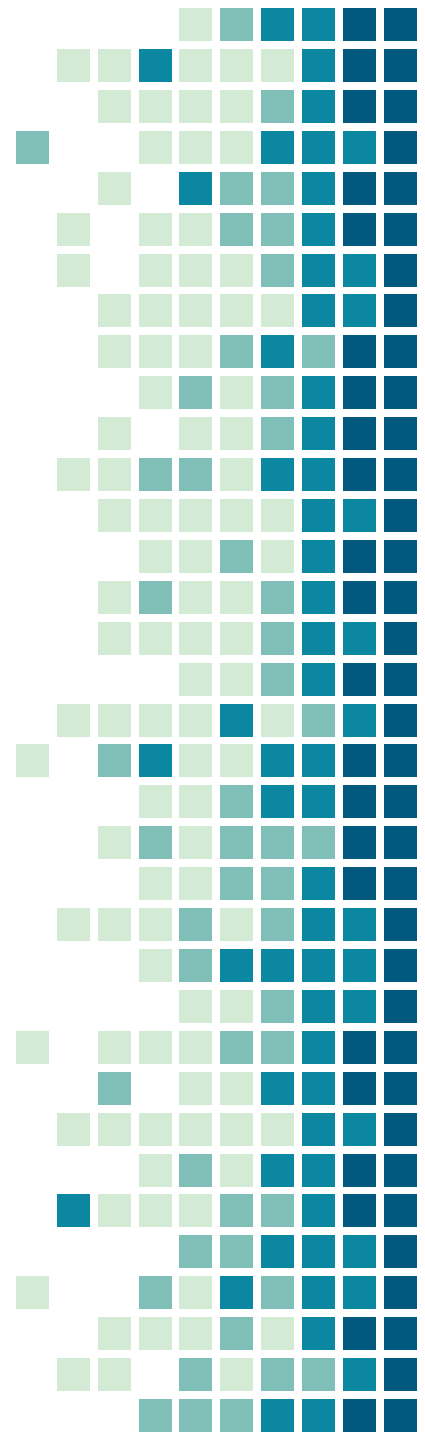
Tools that can be used:

- 3rd Party Tools
 - sqlpackage.exe\SSDT\DacFX
 - Redgate SQL Change Automation
- Home Grown Tools:
 - dbatools.io (Publish-DbuDacPackage)
 - Internal company comparison tools



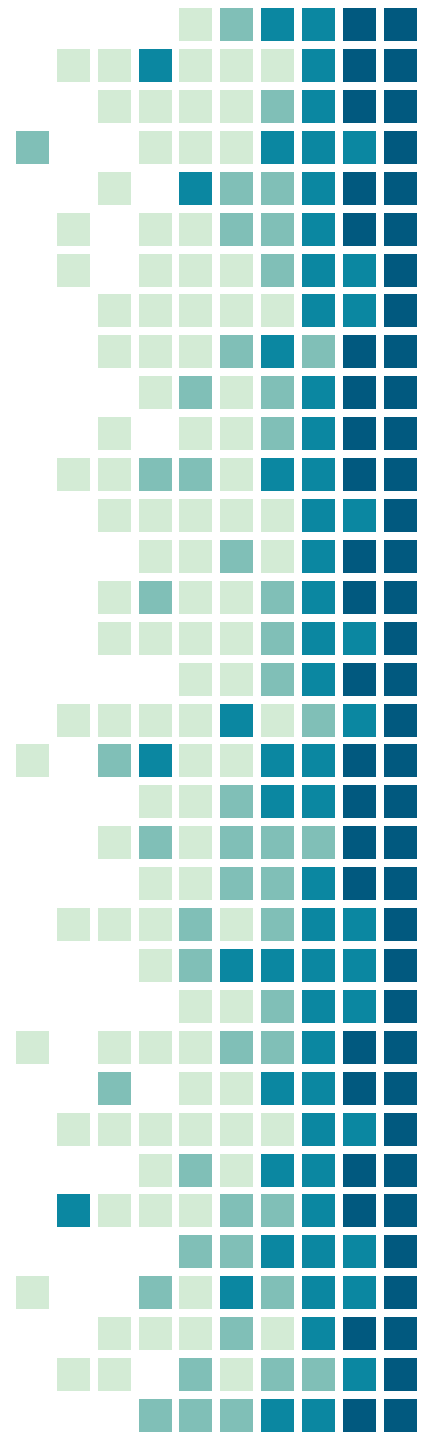
Pros of Migration Deployments

- You determine everything
 - Folder structure
 - Naming convention
 - Execution order of scripts
 - Tool used for deployment
- Flexibility
- The code checked in is the code that will be run



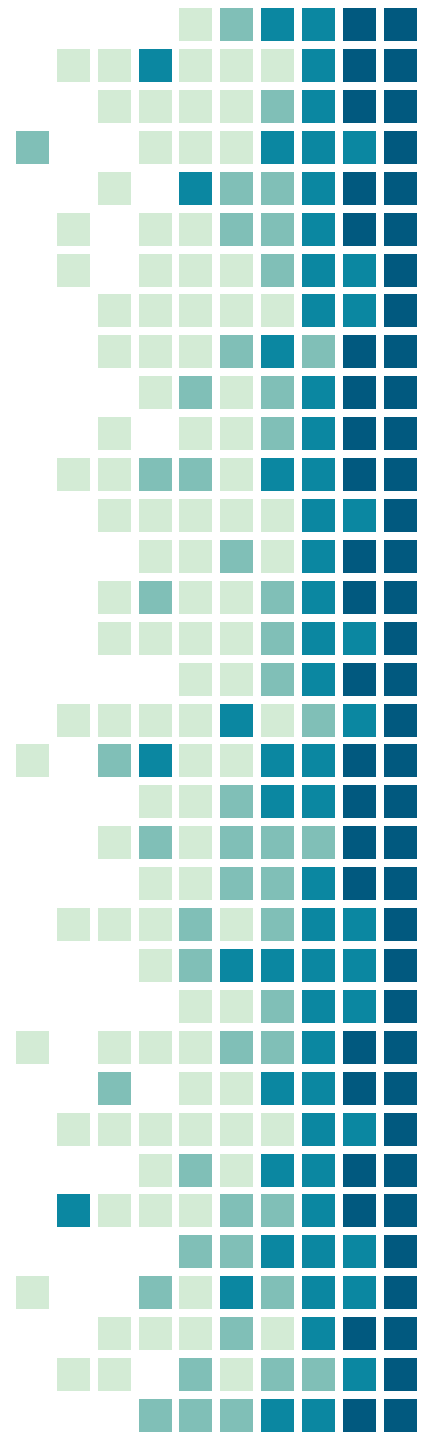
Pros of State Based Deployments

- Configurable
- Can identify schema drifts
- Automatically designed for new installs and upgrades
- Database Projects
 - Structure for code is built in
 - Object validation
 - Cross database dependencies
 - Code Analysis built-in



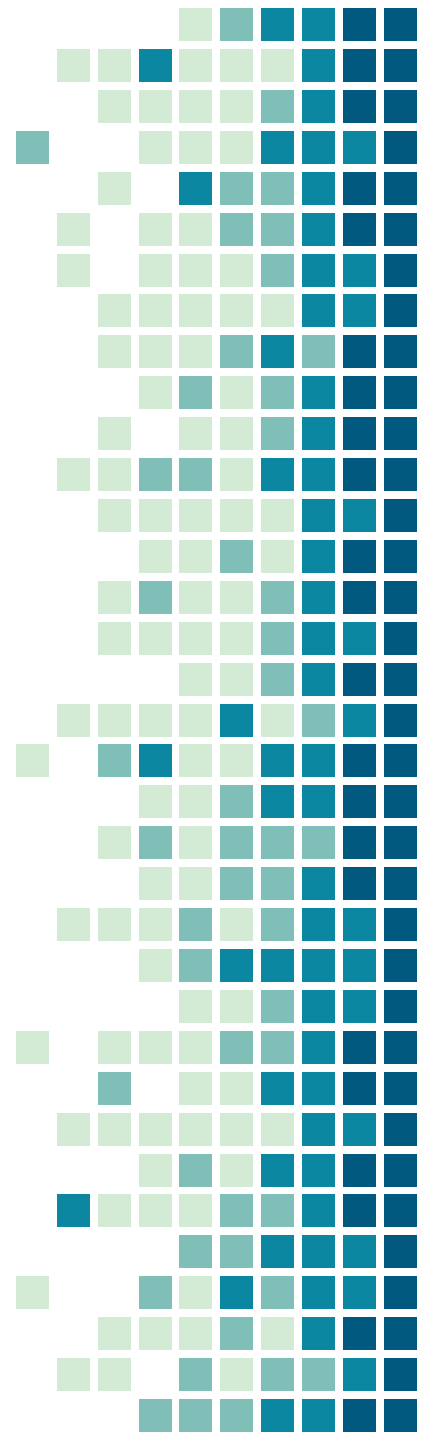
Cons of Migration Deployments

- You determine everything
 - Folder structure
 - Naming convention
 - Execution order of scripts
 - Tool used for deployment
- Schema drifts need to be handled manually
- Need to determine how to handle new installs vs upgrades

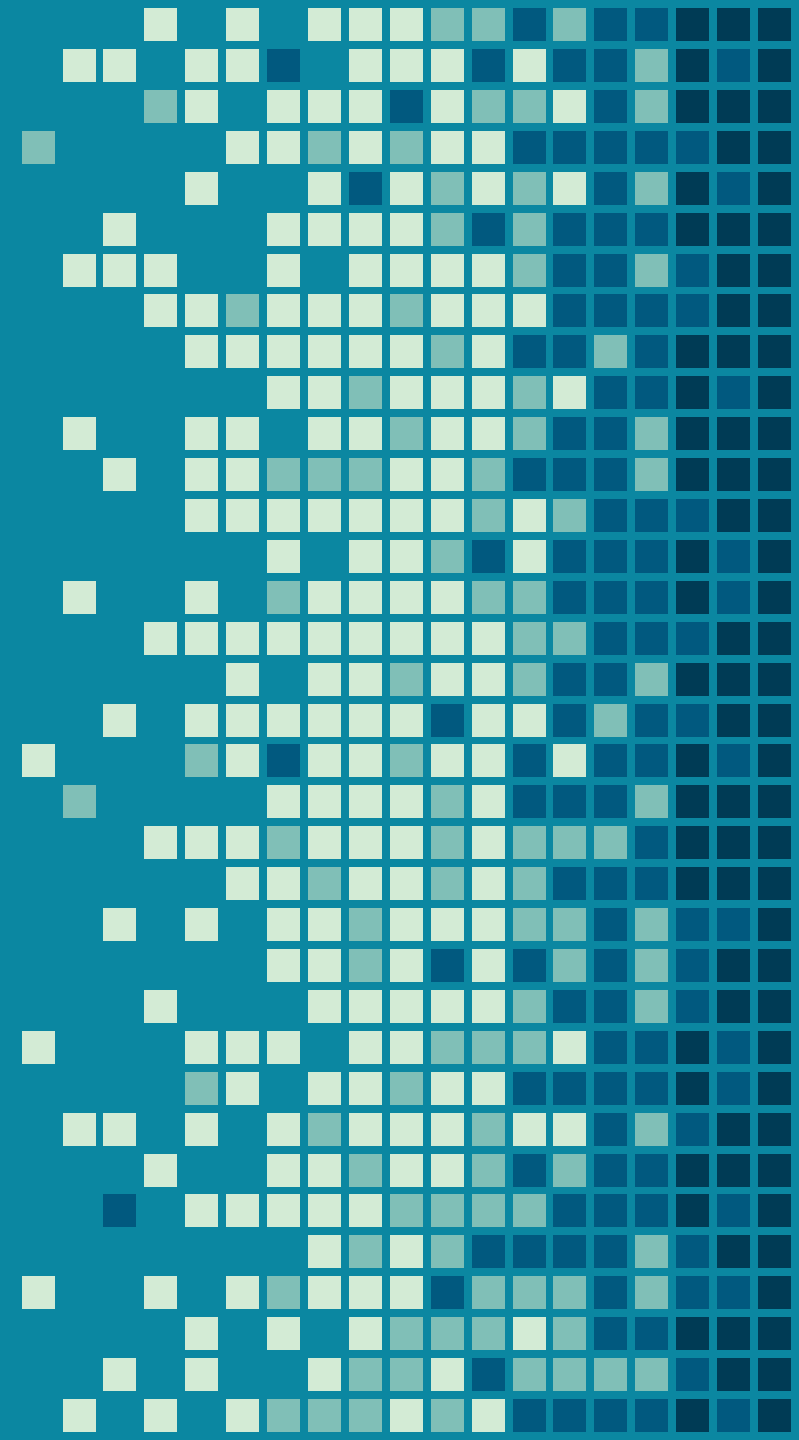


Cons of State Based Deployments

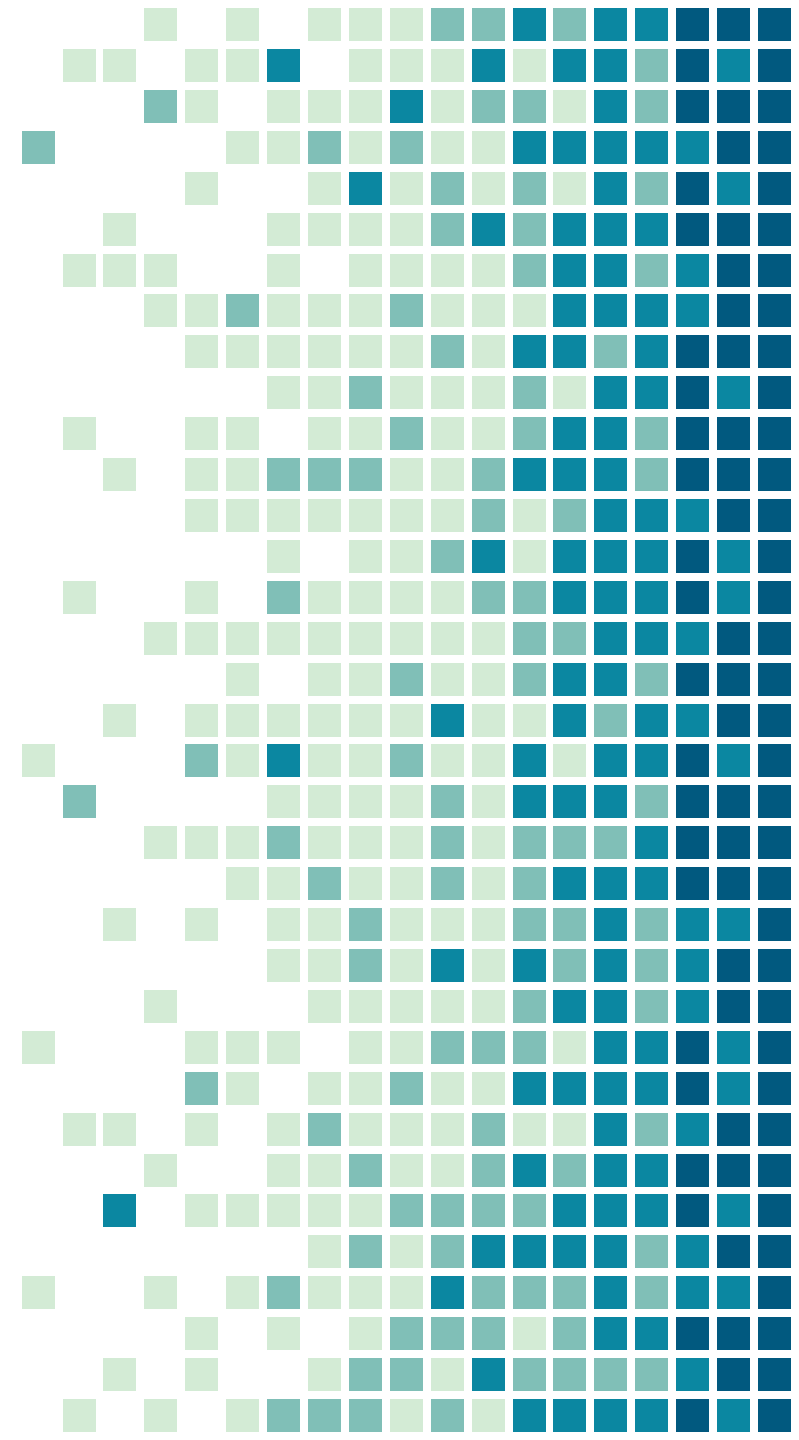
- Can't handle all situations
 - Migration scripts still required
- May take longer to complete when there are a large number of objects and dependencies
- Script of changes can differ from target database to target database
- Database Projects:
 - Restrictions on how code can be written (CREATE only, etc.)
 - SQL Server only



“*Let me show
you what I’m
talking about...*

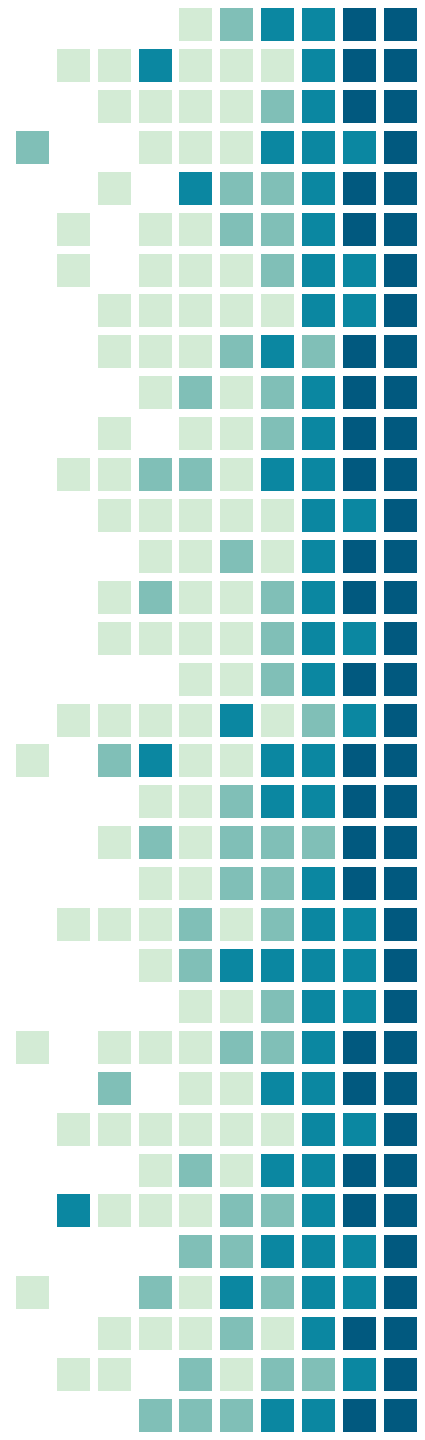


Considerations



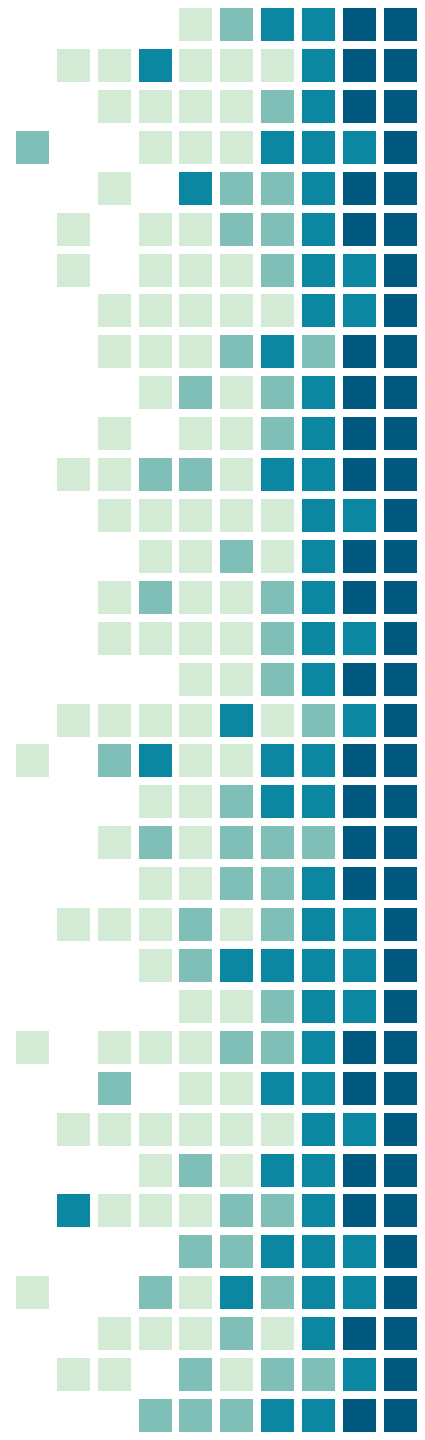
Things to Consider

- Source Control
 - Folder structure
 - Naming Conventions
- Database Instances\Environments
 - Individual developer instances
 - Shared environments
 - Where do the database instances live?
 - Security Roles



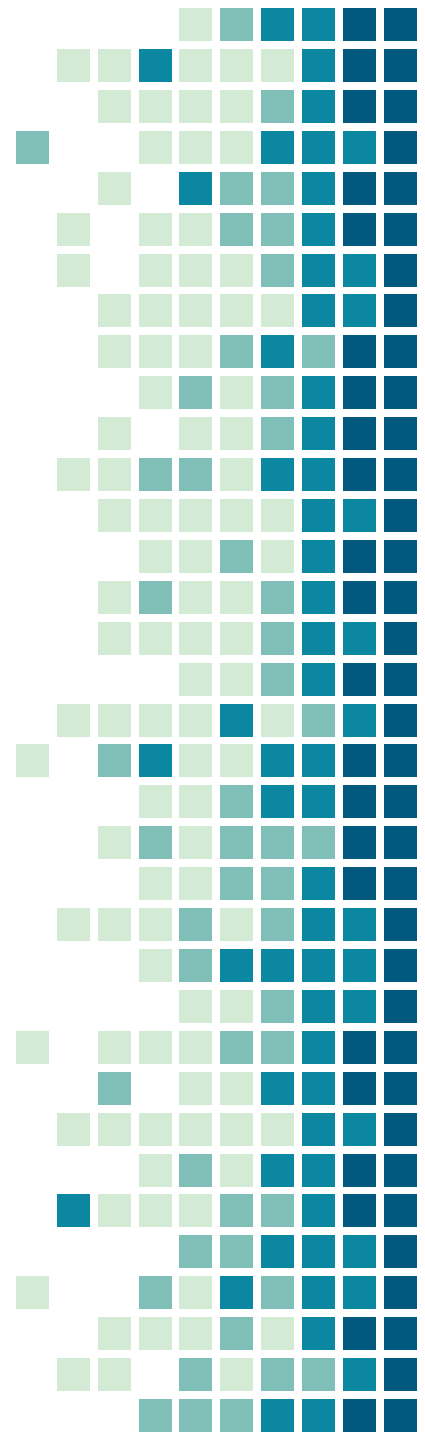
Things to Consider (cont'd)

- Release Processes
 - CI\CD vs Installers?
 - Feature Releases vs. Versioned Releases
 - “Chain of Command”, i.e. who’s responsible for the process?
- Deployment Window
- Data Changes
- Cross Database Dependencies



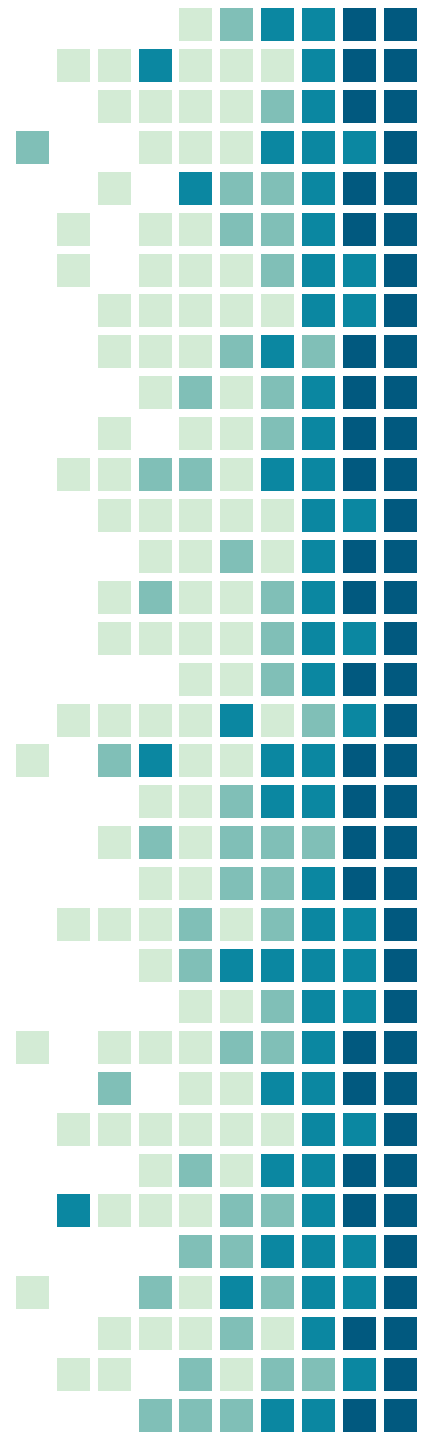
Things to Consider (cont'd)

- Installs vs Upgrades
- Testing
- Client Customization\Schema Drift
- Error handling during deploys\Upgrade "Rollback"



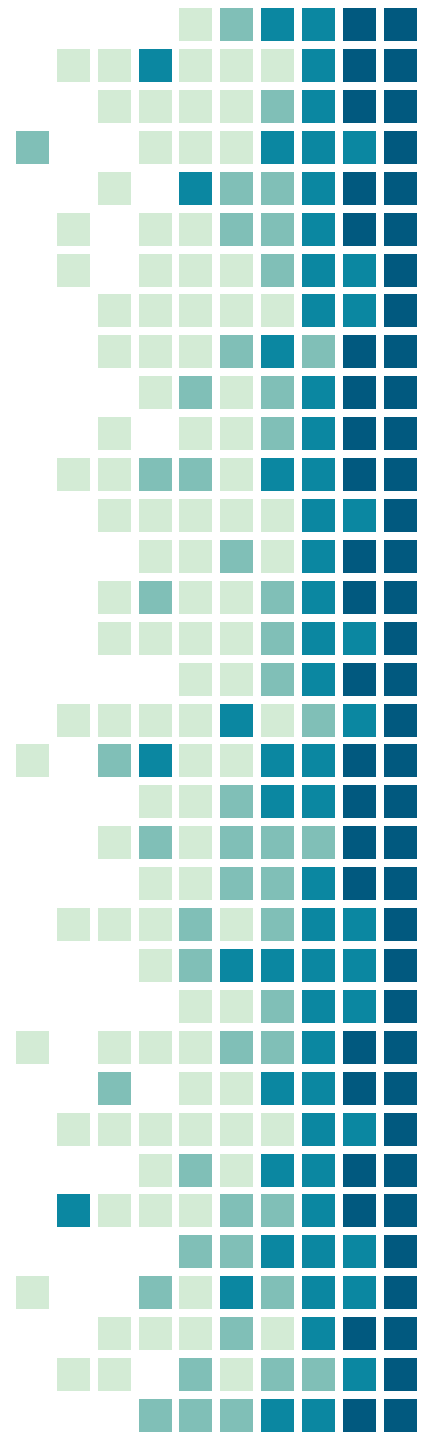
Leaving with more Questions than Answers?

- Understanding how migration based deployments and state based deployments work is the easy part
- Setting up the deployment strategy is hard.



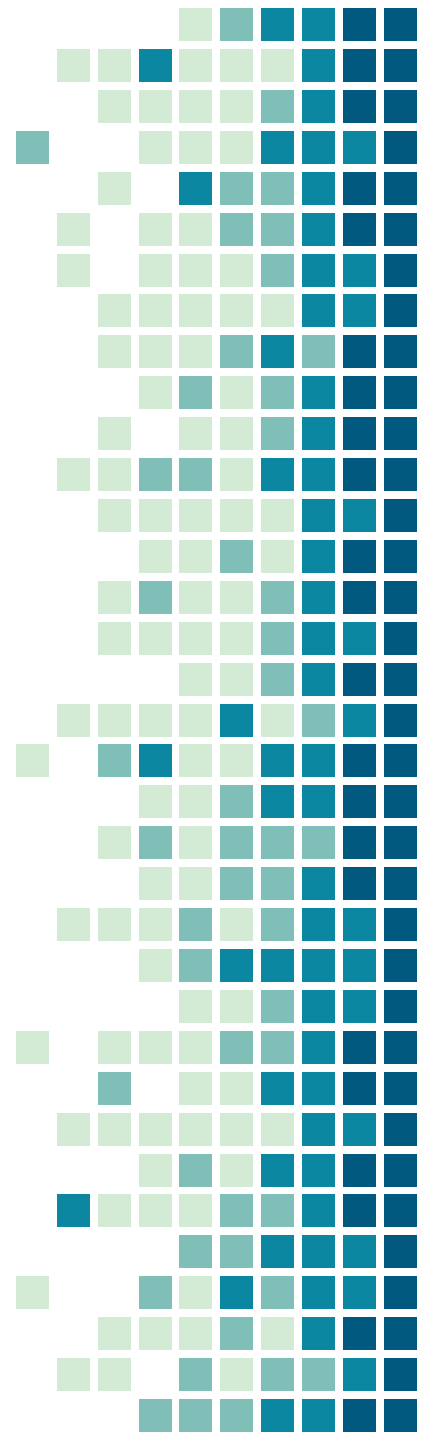
Resources – Tools

- **Flyway:** <https://flywaydb.org/>
- **Liquibase:** <https://www.liquibase.org/>
- **sqlpackage.exe:** <https://learn.microsoft.com/en-us/sql/tools/sqlpackage/sqlpackage?view=sql-server-ver16>
- **Database Projects:** <https://learn.microsoft.com/en-us/visualstudio/data-tools/creating-and-managing-databases-and-data-tier-applications-in-visual-studio?view=vs-2022>



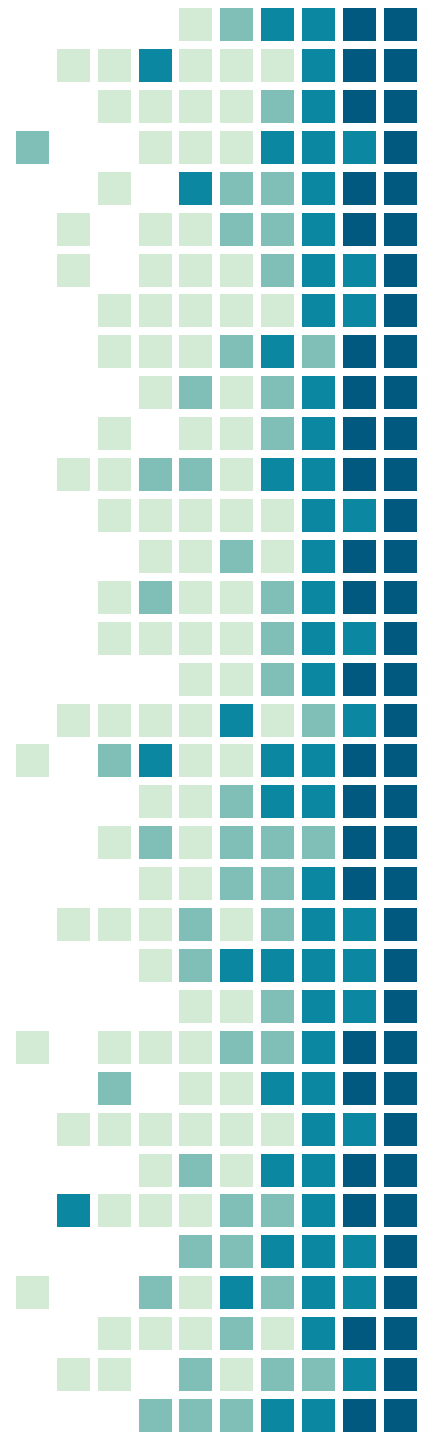
Resources – General Blog Posts

- <https://www.dbmaestro.com/blog/database-devops/state-driven-vs-migration-driven>
- <https://littlekendra.com/2019/01/23/state-vs-migration-for-database-source-control-the-core-question/>
- <https://catalyzesql.com/blog/2023/08/09/database-devops-core-concepts-migration-scripts/>
- <https://blog.greglow.com/2018/02/02/devops-should-databases-use-state-based-migration-based-deployment/>



Resources (cont'd)

- <https://www.gatevnotes.com/continuous-deployment-sql-server-data-tools-project-template/>
- <https://eitanblumin.com/2022/07/12/ssdt-which-version-of-my-database-was-deployed/>
- <https://eitanblumin.com/2022/07/05/should-the-database-and-application-projects-be-in-the-same-repository/>
- <https://www.oliviervs.be/database-projects/database-projects-deploying-changes/>



Any other Questions?

- Social Media handles:
 - @dgmelkin (@dataplatfom.social) (.bsky.social)
- Email:
 - dgmelkin@gmail.com
- Blog:
 - DebtheDBA.wordpress.com
- Github:
 - <https://github.com/DebtheDBA/DBDeployments>



Thanks for coming!

