|  |  |
| --- | --- |
|  | DBmaestro  Database DevOps Platform  DBmaestro vs Liquibase Pro Comparison |

Table of Contents

[1. What is DBmaestro? 3](#_Toc104275992)

[2. Why DBmaestro? 3](#_Toc104275993)

[3. Our Company 3](#_Toc104275994)

[4. What is this document for? 3](#_Toc104275995)

[5. DBmaestro vs Liquibase Pro Comparison 4](#_Toc104275996)

[5.1 DB Pipeline Automation 4](#_Toc104275997)

[5.1.1 Automated DB Deployments 4](#_Toc104275998)

[5.1.2 Support my DB deployment process 4](#_Toc104275999)

[5.1.3 Support environments with multiple databases 5](#_Toc104276000)

[5.1.4 Support multiple environments per type 5](#_Toc104276001)

[5.1.5 Support DB Pipeline Groups 6](#_Toc104276002)

[5.1.6 Full control of all existent DB Pipelines from one single screen 6](#_Toc104276003)

[5.2 Rollbacks 7](#_Toc104276004)

[5.2.1 Automated Rollbacks 7](#_Toc104276005)

[5.2.2 Backup and auto-restore DB on deployment failure 7](#_Toc104276006)

[5.3 Integrations 8](#_Toc104276007)

[5.3.1 DB disconnected from CI/CD process 8](#_Toc104276008)

[5.4 Issue Prevention 8](#_Toc104276009)

[5.4.1 Find Drifts before new SQL change promotion 8](#_Toc104276010)

[5.4.2 DryRun your SQL scripts 9](#_Toc104276011)

[5.4.3 Performance issues in Prod DB 9](#_Toc104276012)

[5.4.4 Risk of Unintended table/column drops 10](#_Toc104276013)

[5.4.5 Bad Coding Practices 10](#_Toc104276014)

[5.4.6 Deployment Time Slot Enforcement for critical DBs 11](#_Toc104276015)

[5.5 Security 11](#_Toc104276016)

[5.5.1 Password Vault Integration 11](#_Toc104276017)

[5.5.2 Deployments to SQL Server DBs using Windows Authentication 12](#_Toc104276018)

[5.5.3 Security Policies 12](#_Toc104276019)

[5.5.4 PII Data Manipulation Risk 13](#_Toc104276020)

[5.5.5 Segregation of Duties for DB Deployments 13](#_Toc104276021)

[5.5.6 Standard Security with LDAP/AD Authentication 14](#_Toc104276022)

[5.5.7 Reduce operational complexity with Centralized User Management 14](#_Toc104276023)

[5.5.8 Multiple Deployment Agents 14](#_Toc104276024)

[5.6 Auditing 15](#_Toc104276025)

[5.6.1 Difficulties with Audit & Compliance 15](#_Toc104276026)

[5.7 Change Customization 15](#_Toc104276027)

[5.7.1 Building SQL script from DEV environment changes 15](#_Toc104276028)

[5.7.2 Customize deployment depending on DB env 16](#_Toc104276029)

[5.7.3 Exclude certain SQL objects from change promotion 16](#_Toc104276030)

[5.8 Governance 17](#_Toc104276031)

[5.8.1 Need to enforce naming conventions 17](#_Toc104276032)

[5.8.2 DB Governance is very complex 17](#_Toc104276033)

[6. DBmaestro Success Stories 18](#_Toc104276034)

# 

# 

# 1. What is DBmaestro?



DBmaestro is a self-service tool for DevOps teams to automate, secure, and govern Database CI/CD pipelines.

DBmaestro brings agility to your database development, integrates with DevOps processes, boosts productivity, and empowers collaboration.

# 2. Why DBmaestro?



Because with DBmaestro you can:

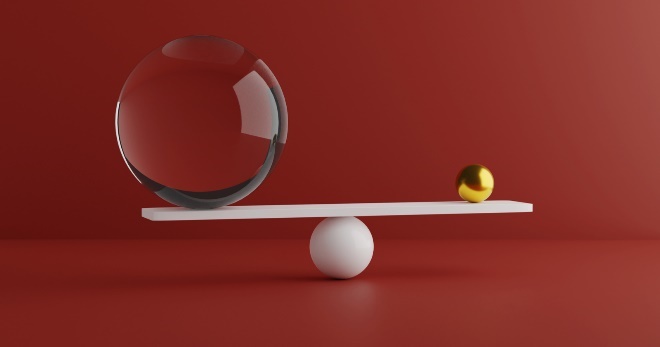
* Boost deployment frequency and speed
* Ensure successful release every time
* Leverage existing tools and processes
* Achieve continuous integration
* Prevent configuration drifts, errors, downtime, and out-of-process changes
* Gain full control and visibility into database changes
* Bring safety, security, and order to database deployments
* Leverage database source control to simplify collaboration
* Automate script creation and shorten feedback loops with the DBA team

# 3. Our Company



We pioneered the database delivery automation market with a disruptive technology approach for automating CI/CD processes in the database. Today, DBmaestro leads in the database delivery automation category, empowering organizations to automate, secure, and govern database delivery pipelines.

# 4. What is this document for?



This document brings to you a detailed comparison of DBmaestro vs Liquibase Pro, where you’ll see the differences and get a very complete understanding of what DBmaestro will deliver and how it excels compared to the competition.

# 

# 5. DBmaestro vs Liquibase Pro Comparison

## 5.1 DB Pipeline Automation

### 5.1.1 Automated DB Deployments

Graphical user interface, application

Description automatically generated

Image 1. DBmaestro can show you a visualization of your full DB Pipeline

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro can automate your Database Delivery process, including upgrades, rollbacks, policy enforcement, etc., giving you full control and visibility from its built-in Web UI or CLI without risking security, in fact making your DBs safer. | **Liquibase Pro**  Liquibase has automation commands. There is a limited centralized Web UI mostly for logging purposes, but it’s only available as a SaaS service, meaning anyone outside your organization can access to it if they learn your credentials. No way to “pilot the plane”, as you cannot trigger deployments from it, and no segregation of duties for your organization roles (DBAs, Developers, Release Managers, etc.) to leverage your least-privilege policies. |

### 5.1.2 Support my DB deployment process

Graphical user interface, application

Description automatically generated

Image 2. DBmaestro supports Deployments by Versions and Task from UI and CLI

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro adjusts to your already existing process. You can create a DB pipeline with just a few clicks. Deciding deployment by version or by task (tickets) is as seamless as choosing the project type from its friendly Web UI. | **Liquibase Pro**  Liquibase supports deployment by version and by task, but no user-friendly setup and no UI visualization for easy configuration and enforcement of this. Aside from that, deployments cannot be triggered from UI either. |

### 5.1.3 Support environments with multiple databases

Graphical user interface, application, chat or text message

Description automatically generated

Image 3. DBmaestro supports multiple databases per DB environment

|  |  |
| --- | --- |
| **DBmaestro**  Your applications are complex. Your applications need to connect to more than one DB to fulfill their tasks (i.e.: CRM app connects to CRM DB, Billing DB and Finance DB). DBmaestro knows this and supports multiple databases per environment out-of-the-box to make things simple for you. | **Liquibase Pro**  Liquibase doesn't support the concept of a Database environment, that can have 1 or more databases. Instead, it only supports the concept of a database. Comparisons can be done only between 2 databases, and you must deploy in one database at a time. No way to deploy to a whole environment and all databases that belong to it in one go. |

### 5.1.4 Support multiple environments per type

A screenshot of a computer

Description automatically generated with medium confidence

Image 4. DBmaestro supports multiple DB environments per environment type

|  |  |
| --- | --- |
| **DBmaestro**  You have critical applications that many Dev Teams concurrently work on. Besides that, and because of time limitations, in UAT your Dev Teams test unrelated features with different business users at the same time, creating bottlenecks or getting into collisions that impact your timeline for business deliverables. And in PROD, sometimes you have DBs that serve other countries. That’s why DBmaestro supports multiple DB environments for all environment types: DEV, QA, UAT, PRE-PROD and PROD. | **Liquibase Pro**  Liquibase doesn't support the concept of Database Environment Type out-of-the-box. No seamless creation of database pipelines with many environments per environment type. |

### 5.1.5 Support DB Pipeline Groups

Graphical user interface, text, application

Description automatically generated

Image 5. DB Pipelines in DBmaestro are called Projects and they can be put in Project Groups

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro can group Projects (DB Pipelines) and set DBmaestro user permissions based on both Projects and Project Groups, to make things simpler for you to manage. | **Liquibase Pro**  Liquibase does not support DB Pipeline Groups. |

### 5.1.6 Full control of all existent DB Pipelines from one single screen

Graphical user interface, application

Description automatically generated

Image 6. DBmaestro Home Page shows you all DB Pipelines you have and can control

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro supports orchestration of all your DB Pipelines from one single screen. This lets you know about of every DB Pipeline there is under your control, what DB Engine it’s related to, when last activity happened, latest deployed and available versions, etc. This screen belongs to DBmaestro Server, which can be installed on-premises or in your cloud of choice. | **Liquibase Pro**  Liquibase has no support to control all your existent DB Pipelines from one single screen. There is an optional SaaS Web UI that is available online where you can have your logs saved to, but you cannot control your DB Pipelines from there, and you cannot be sure all Pipelines of your organization show up in there, as it’s not mandatory. Your team may even opt not to use it for security reasons, as it’s published on the internet. |

## 5.2 Rollbacks

### 5.2.1 Automated Rollbacks

Graphical user interface, application

Description automatically generated

Image 7. DBmaestro supports Rollbacks from UI and CLI

|  |  |
| --- | --- |
| **DBmaestro**  Minimize downtime with automated rollbacks that will let you undo your previously introduced SQL changes, letting DBmaestro undo them for you or by using your own downgrade SQL scripts | **Liquibase Pro**  Liquibase supports rollbacks. They can only be triggered from command line, not from UI. |

### 5.2.2 Backup and auto-restore DB on deployment failure

Graphical user interface

Description automatically generated

Image 8. DBmaestro Backup/Auto Restore functionality

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro supports backups and auto-restores in case deployment fails. This functionality can be set up as optional or default behavior, as you decide. | **Liquibase Pro**  Liquibase has no backup/restore functionality in case of deployment failure. |

## 5.3 Integrations

### 5.3.1 DB disconnected from CI/CD process

Graphical user interface, application, Teams

Description automatically generated

Image 9. Jenkins Pipeline Stages for an integration with DBmaestro and Jira

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro powerful automation commands can be integrated to any CI/CD orchestrator. No need to provide DB credentials from CI/CD orchestrator, a DBmaestro user token is used instead. Execution will be blocked if DBmaestro user doesn’t have the permissions to do the deployment. | **Liquibase**  Liquibase can be integrated to CI/CD tools with automation commands. But CI/CD tool needs to directly send the database connection info and database credentials to connect and deploy to databases. |

## 5.4 Issue Prevention

### 5.4.1 Find Drifts before new SQL change promotion

Graphical user interface, application

Description automatically generated

Image 10. DBmaestro can run drift detection on demand or by default before/after each deploy

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro identifies any unexpected changes introduced into your DB SQL structure that came from outside your deployment process, preventing you from any undesired impact to your database environments. This can be executed standalone or as a deployment pre-step and validates all DBs in your DB environment in one go. | **Liquibase Pro**  Compares 2 databases, or a database to a snapshot, but no option to automatically do it as a pre-step before DB deployment. No support for the concept of database environments that can have more than one database, the comparison needs to be done atomically one DB at a time. No way to this from UI, only from command-line. |

### 5.4.2 DryRun your SQL scripts

Graphical user interface, application

Description automatically generated

Image 11. DBmaestro supports dedicated DryRun environments to PreCheck your SQL scripts

|  |  |
| --- | --- |
| **DBmaestro**  PreCheck your SQL scripts dry-running them in a real database before they are promoted to your next DB environments. Enforced Policies are also checked during DryRun execution. And no need to refresh or clean up DryRun environment, it does it automatically. | **Liquibase Pro**  Liquibase doesn't support the concept or capabilities of a DryRun environment |

### 5.4.3 Performance issues in Prod DB

Graphical user interface, text, application, chat or text message

Description automatically generated

Image 12. Only a few of the DBmaestro out-of-the-box Performance Rules

|  |  |
| --- | --- |
| **DBmaestro**  Over 70 out-of-the-box rules ready to be enforced on your SQL code, to block or warn your developers letting them know about best practices you want them to apply before SQL code promotion. You can also choose to backup and continue. You can duplicate or add more from scratch. You can define time slots for deployments and specific environment types and DB engines to use them on | **Liquibase Pro**  Around 10 out of the box Quality Checks, no database backup option. No web UI to easily manage these checks, no time slot policies, no easy configuration on specific DB environments or DB engines to apply the checks to, and no Rule Verifier like the one DBmaestro's got. |

### 5.4.4 Risk of Unintended table/column drops

Graphical user interface, text, application

Description automatically generated

Image 13. Only a few of the DBmaestro out-of-the-box Data Loss Risk Rules

|  |  |
| --- | --- |
| **DBmaestro**  Prevent any unexpected DROPs in your developers' SQL code to be involuntarily executed. Prevent undesired data loss code like DROP COLUMN, DROP TABLE, TRUNCATE TABLE, etc. | **Liquibase Pro**  Liquibase can only prevent DROPs of tables and columns out of the box. |

### 5.4.5 Bad Coding Practices

Graphical user interface, text, application

Description automatically generated

Image 14. Only a few of the DBmaestro out-of-the-box Coding Standards Rules

|  |  |
| --- | --- |
| **DBmaestro**  Automate best SQL coding practice validation by enforcing out-of-the-box Rules and Policies and any custom policy that you may already be checking for manually in your organization. | **Liquibase Pro**  No out-of-the-box SQL Coding Standards Policies to be enforced |

### 5.4.6 Deployment Time Slot Enforcement for critical DBs

Graphical user interface, application

Description automatically generated

Image 15. DBmaestro out-of-the-box Time Slots Rules

|  |  |
| --- | --- |
| **DBmaestro**  Every software application is different, and that applies to its databases as well. With DBmaestro you can enforce different time slot policies based on day of week and time of day, preventing any impact on your day-to-day operations | **Liquibase Pro**  Liquibase doesn't have the Time Slot Policy functionality |

## 5.5 Security

### 5.5.1 Password Vault Integration

Graphical user interface, application

Description automatically generated

Image 16. DBmaestro out-of-the-box Vault Integration

|  |  |
| --- | --- |
| **DBmaestro**  Out-of-the-box Integration to Vaults from vendors like AWS, Azure and CyberArk will let you keep your database credentials safe and rotate your passwords without affecting your SQL promotions or needing to update anything manually. | **Liquibase Pro**  No out-of-the-box vault integration. Liquibase has no centralized server or security layer to build this Vault integration on top of. |

### 5.5.2 Deployments to SQL Server DBs using Windows Authentication

Graphical user interface, text, application, chat or text message

Description automatically generated

Image 17. DBmaestro supports SQL Server Windows Authentication out of the box

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro does all the heavy lifting for you and makes Windows Authentication completely seamless for your SQL Server DB deployments. | **Liquibase**  For Liquibase to connect to SQL Server using Windows Authentication it needs you to create an IIS web application on a Windows server that you must provision yourself. All this to impersonate the identity supplied by IIS and you must do this whole setup on your own and maintain it yourself. |

### 5.5.3 Security Policies

Graphical user interface, application

Description automatically generated

Image 18. Only a few of the DBmaestro out-of-the-box Security Rules

|  |  |
| --- | --- |
| **DBmaestro**  Automatically validate for any permission violations in your SQL code: Grants, User alterations, Connection changes, Password changes, and every custom security guideline you already manually enforce in your organization | **Liquibase**  Liquibase only supports check for GRANT statements out of the box. |

### 5.5.4 PII Data Manipulation Risk

Graphical user interface, text, application

Description automatically generated

Image 19. Only a few of the DBmaestro out-of-the-box PII Rules

|  |  |
| --- | --- |
| **DBmaestro**  Automatically prevent any operations in PII tables that could affect your sensitive data | **Liquibase Pro**  Liquibase has no PII Data Manipulation risk checks out of the box. |

### 5.5.5 Segregation of Duties for DB Deployments

Graphical user interface, application

Description automatically generated

Image 20. Only a few of the DBmaestro out-of-the-box User Roles.

|  |  |
| --- | --- |
| **DBmaestro**  With DBmaestro you get a very robust security layer, where users can have access only to the specific DB pipelines and environments you grant them access to and can only do the actions you permit them to do. | **Liquibase Pro**  Liquibase has no user permissions layer of its own. There is no way to segregate users based on User Roles or Permissions. Liquibase requests DB credentials as input to deploy changes, and therefore anyone can do any action that the DB credentials allow them to do. |

### 5.5.6 Standard Security with LDAP/AD Authentication

Graphical user interface, application, chat or text message

Description automatically generated

Image 21. DBmaestro Domain Configuration

|  |  |
| --- | --- |
| **DBmaestro**  You can reuse your already existing LDAP/AD authentication infrastructure to login to DBmaestro from Web UI and CLI. | **Liquibase Pro**  Liquibase does not have its own user permissions layer, it just directly receives the DB credentials to connect with. |

### 5.5.7 Reduce operational complexity with Centralized User Management

Graphical user interface, text, application, Teams

Description automatically generated

Image 22. DBmaestro configuration of Directory Service Sync

|  |  |
| --- | --- |
| **DBmaestro**  No need to manually create users in DBmaestro. Automatically populate and sync your DBmaestro userbase from your LDAP/AD instance with their corresponding permissions. | **Liquibase Pro**  Liquibase does not support direct segregation of duties, as it has no user layer of its own. Therefore, there is no sync to LDAP/AD user groups either. |
|  |  |

### 5.5.8 Multiple Deployment Agents

Graphical user interface, application

Description automatically generated

Image 23. DBmaestro supports multiple Deployment Agents

|  |  |
| --- | --- |
| **DBmaestro**  For better security and performance, DBmaestro supports multiple Deployment Agents on separate machines orchestrated from one single DBmaestro Server. An example of usage would be to have a separate DBmaestro Agent to deploy changes in Production from a server node provisioned in the same network than the Production Databases. | **Liquibase Pro**  No support for adding Deployment Agents. A Liquibase instance can only deploy its own changes. |
|  |  |

## 5.6 Auditing

### 5.6.1 Difficulties with Audit & Compliance

Graphical user interface, application

Description automatically generated

Image 24. DBmaestro Audit Report

|  |  |
| --- | --- |
| **DBmaestro**  Auditing Database Deployments doesn't have to be a tedious process. With DBmaestro you get to know what, when, where and who did a deployment, all from one friendly and reliable source. | **Liquibase Pro**  Every database has its own changelog table separately. No consolidated Audit Report, nor a Web UI to see it |

## 5.7 Change Customization

### 5.7.1 Building SQL script from DEV environment changes

Graphical user interface

Description automatically generated

Image 25. DBmaestro DEV Script generation

|  |  |
| --- | --- |
| **DBmaestro**  DBmaestro automatically creates a new Deployment Package from all new changes done in your Development Environment | **Liquibase Pro**  Liquibase has the functionality to find differences between 2 databases and create a script out of it, but can only do it from command line (not from Web UI) and it's a 2-step process: generating the diff and then generating the SQL statements |

### 5.7.2 Customize deployment depending on DB env

Graphical user interface, text, application

Description automatically generated

Image 26. DBmaestro Release Variables screen to customize deployments

|  |  |
| --- | --- |
| **DBmaestro**  Have different actions executed depending on Database Environment. A simple example would be when you want to deploy test data only to a database test environment. | **Liquibase Pro**  Liquibase supports SQL script customization depending on DB environment but does not have a straightforward UI to manage Release Variable values from. |

### 5.7.3 Exclude certain SQL objects from change promotion

Graphical user interface, text, application, email

Description automatically generated

Image 27. DBmaestro Object Exclusion screen

|  |  |
| --- | --- |
| **DBmaestro**  Exclude SQL objects to be promoted, rolled back or even counted as a drift if found modified. | **Liquibase Pro**  SQL objects can be excluded from changelog creation, but it cannot be saved per Database pipeline for reuse and there is no easy Web UI to configure this |

## 5.8 Governance

### 5.8.1 Need to enforce naming conventions

Graphical user interface, text, application

Description automatically generated

Image 28. Only a few of the DBmaestro out-of-the-box Naming Convention Rules

|  |  |
| --- | --- |
| **DBmaestro**  A simple example of this would be checking for name Prefixes or Suffixes depending on SQL object types | **Liquibase Pro**  No out-of-the-box Naming Convention Checks |

### 5.8.2 DB Governance is very complex

Graphical user interface, application

Description automatically generated

Image 29. DBmaestro Customizable Dashboard with metrics like “Deployment Frequency”

|  |  |
| --- | --- |
| **DBmaestro**  Automatically validate SQL code for violations to your custom policies and rules. Also, monitor all the activity of your SQL deployments, deployment frequency and lead time globally, per each DB pipeline, or by groups. | **Liquibase Pro**  No way to have a centralized Dashboard of all deployments and metrics like Deployment Frequency, as every target database keeps its own changelog table |

## 

# 6. DBmaestro Success Stories

|  |  |
| --- | --- |
| Graphical user interface, text, application  Description automatically generated  Graphical user interface, text, application, chat or text message  Description automatically generated  Graphical user interface, text, application  Description automatically generated | Graphical user interface, application  Description automatically generated  Graphical user interface, text, application, chat or text message  Description automatically generated  Graphical user interface, text, application, chat or text message  Description automatically generated |