

# Database Migration Tools Comparison

Graham Pellegrini

August 18, 2025

## Contents

<b>1</b>	<b>Executive Summary</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Bytebase</b>	<b>3</b>
<b>4</b>	<b>Liquibase</b>	<b>3</b>
<b>5</b>	<b>Redgate</b>	<b>4</b>
<b>6</b>	<b>Detailed Comparison</b>	<b>6</b>
6.1	Licensing and Cost (Free Versions Focus) . . . . .	6
6.2	Database Support . . . . .	6
6.3	Feature Comparison . . . . .	6
<b>7</b>	<b>Conclusion</b>	<b>7</b>
<b>8</b>	<b>References</b>	<b>8</b>

# 1 Executive Summary

This document provides a comprehensive comparison of three leading database migration and schema management tools: Bytebase, Liquibase, and Redgate. Each tool offers distinct approaches to database versioning, migration management, and team collaboration, catering to different organizational needs and technical requirements.

# 2 Introduction

The testing was conducted using Microsoft SQL Server and MySQL databases, focusing on the capabilities of the freely available versions of each tool.

## **What we tested:**

- All tools used the same SQL migration scripts
- Tested connections to both SQL Server and MySQL
- Evaluated free/community editions only

## **Tools compared:**

- **Bytebase Community**
- **Liquibase Open Source**
- **Redgate Free Edition**

### 3 Bytebase

Modern web-based tool with a clean interface. Works like a database version of GitHub, great for teams who want visual collaboration.

**What it does well:**

- Easy-to-use web interface
- Connects to multiple database types
- Tracks who changed what and when
- Admin and User permissions configuration
- Free for up to 20 users and 5 databases

**Setup:**

To set up Bytebase, you need to have Docker installed and running on your machine. Once Docker is ready, you can pull the Bytebase Docker image and start the container. The web interface will be accessible through your browser at the specified port. Bytebase code works using a simple folder structure with SQL migration files and a single YAML configuration file (bytebase-config.yaml) to manage database changes. The tool then provides a web service accessible through your browser, from which you can manage your database migrations.

**Testing Outcomes:**

The added dependency on Docker means that if Docker isn't running, Bytebase can't function. This was the main hurdle faced during testing and was an additional learning that had to be tackled. Bytebase then showed its potential by providing a robust migration tracking system and an intuitive web interface for managing database changes. The login process when debugging was tedious as it would constantly prompt for a new admin login and reusing email addresses was not possible. Bytebase effectively managed redundant migrations and provided clear visibility into the migration history. That is processing time for migrations where no changes were made was minimal, showcasing Bytebase's efficiency.

**Overview:**

Bytebase code was simple to learn and interface seems great for teams, the free version is limited for larger projects. Migration tracking is excellent, and the web interface is intuitive. Version control is not as robust as traditional Git-based systems, which may be a drawback for some teams that values simplicity, are comfortable with Docker containers, and don't mind the additional infrastructure setup requirements.

### 4 Liquibase

Command-line tool that's been around for years. Uses text files to track database changes - popular with developers who like automation.

**What it does well:**

- Supports 40+ database types
- Works with any CI/CD pipeline
- Completely free with no user limits
- Very flexible - uses XML, YAML, or SQL files

**Setup:** Liquibase supports three changelog formats: XML, YAML, and SQL files. You can choose any format based on your team's preference. We used XML files for our migrations because they provide better structure validation and error checking compared to plain SQL. The tool uses a master changelog file that references all individual migration files, with configuration managed through a `liquibase.properties` file. Unlike the other tools, Liquibase can be fully accessed and used directly from code/command line without requiring separate applications. Our implementation required manually downloading the correct database libraries (JDBC drivers) and adjusting existing ones to work with our Python virtual environment. This was a very tedious process, especially getting Liquibase to properly identify file paths. The tool is heavily dependent on where you run commands from. Path configurations in the properties file and relative references in XML files must be precisely aligned, making it fragile when running from different directories.

**Testing Results:** Worked with MySQL successfully and showed consistent performance in automated testing (3-4 second execution times). Could not connect to SQL Server due to TCP/IP network restrictions, requiring the automated system to skip Liquibase for SQL Server migrations entirely. When it worked, Liquibase demonstrated reliable execution with proper status reporting ("Database schema is up to date", "No new changesets to execute"). All operations generate detailed log files, making it excellent for debugging and audit trails. This file-based logging is particularly useful for automated environments and troubleshooting.

**Overview:**

*Pros:* Supports the most databases, totally free, great for automation, very mature tool, powerful once configured, fully code-integrated (no separate apps needed), excellent file-based logging for debugging.

*Cons:* Harder to learn, no web interface, may need network setup in corporate environments, tedious setup process, heavily path-dependent, requires manual library management.

*Choose this if:* You need maximum database support, prefer command-line tools, have complex automation needs, and don't mind spending significant time on initial setup and configuration.

## 5 Redgate

Visual tools focused on SQL Server. Think of it as the "Microsoft Office" of database tools - polished and feature-rich.

**What it does well:**

- Excellent visual comparison tools
- Built specifically for SQL Server
- Easy drag-and-drop interface
- Professional-grade features

**Setup:** Unlike Bytebase's web interface or Liquibase's command-line approach, Redgate requires downloading and installing a separate Windows desktop application - you can't just pip install it or use it directly in VS Code. The "visual" part means it shows database schemas side-by-side in a graphical interface where you can see differences highlighted in colors, click to navigate between objects, and use drag-and-drop to synchronize changes. It's a standalone desktop application, not integrated with your development environment. Must download and install the desktop application from Redgate's website. Cannot be installed via pip or used directly within VS Code like Python packages.

**Testing Results:** Connected successfully to SQL Server and showed consistently fast execution times in automated testing (0.6 seconds for SQL Server, 0.5-0.7 seconds for MySQL).

Demonstrated reliable connectivity and quick processing through PowerShell integration across both database platforms.

**Overview:**

*Pros:* Best-in-class for SQL Server, intuitive desktop interface with visual comparisons, professional support, reliable connectivity.

*Cons:* Expensive for full features, mainly SQL Server only, very limited free version (mainly just schema comparison for single users), requires Windows desktop installation (not web-based).

*Choose this if:* You primarily use SQL Server, prefer desktop applications over web interfaces, need visual schema comparison tools, or have budget for premium software.

## 6 Detailed Comparison

### 6.1 Licensing and Cost (Free Versions Focus)

Aspect	Bytebase Community	Liquibase Open Source	Redgate Free Tools
License Type	Free Community Edition	Open Source (Apache 2.0)	Limited Free Edition
User Limitations	Up to 20 users	Unlimited	Single user
Database Instances	Up to 5 instances	Unlimited	Limited
Enterprise Features	Limited RBAC, Basic audit	None	Very limited
Support	Community only	Community only	Limited
Upgrade Path	Seamless to Enterprise	Liquibase Pro available	Full Redgate suite

Table 1: Free Version Licensing Comparison

### 6.2 Database Support

Database	Bytebase	Liquibase	Redgate
PostgreSQL	✓	✓	Limited
MySQL	✓	✓	Limited
SQL Server	✓	✓	✓ (Primary)
Oracle	✓	✓	Limited
MongoDB	✓	✓	No
SQLite	✓	✓	No
Total Supported	15+	40+	10+

Table 2: Database Platform Support

### 6.3 Feature Comparison

Feature	Bytebase	Liquibase	Redgate
Web Interface	✓	Limited	✓
Command Line	✓	✓	✓
Version Control	Git Integration	File-based	Git/TFS Integration
Rollback	✓	✓	✓
Schema Drift	✓	Limited	✓
Data Migration	✓	✓	✓
Multi-environment	✓	✓	✓
RBAC	✓	Enterprise only	✓
API Access	✓	✓	Limited
Audit Trail	✓	Basic	✓

Table 3: Feature Comparison Matrix

## 7 Conclusion

### Quick Summary:

- **Bytebase** - Best for teams who want something modern and easy to use
- **Liquibase** - Best for developers who need maximum database support and automation
- **Redgate** - Best for SQL Server shops with budget for premium tools

**What to consider:** For simple migrations, all three work fine. For complex stuff, Liquibase and Redgate have more features. Small teams should try Bytebase first - it's the easiest to learn. Large companies might prefer Liquibase for its flexibility or Redgate for SQL Server environments. Remember that Liquibase might need extra network setup in corporate environments.

**Bottom line:** Pick based on your database type, team size, and how much complexity you need. Start with the free versions to see what works for your team.

## 8 References

- Bytebase Official Documentation: <https://bytebase.com/docs>
- Liquibase Documentation: <https://docs.liquibase.com>
- Redgate Documentation: <https://documentation.red-gate.com>