

03 JUNE 2025

Boldly Migrate to PostgreSQL – Introducing credativ-pg-migrator

Many companies these days are thinking about migrating their databases from legacy or proprietary system to [PostgreSQL](#). The primary aim is to reduce costs, enhance capabilities, and ensure long-term sustainability. However, even just the idea of migrating to PostgreSQL can be overwhelming. Very often, knowledge about the legacy applications is limited or even lost. In some cases, vendor support is diminishing, and expert pools and community support are shrinking. Legacy databases are also often running on outdated hardware and old operating systems, posing further risks and limitations.

Problem: How to migrate databases efficiently to PostgreSQL



Josef Machytka and Alexander Wirt at PGconf.de 2025

Many organizations have also over time introduced several different database systems, and their IT environments are now a patchworks of different solutions. In this situation, IT teams are facing dilemmas when it comes to the question of how to efficiently perform all these database migrations. The perspective of utilizing different migration tools for different databases is not very appealing. Each specific tool has its own set of features, limitations, and learning curves. Using multiple tools can result in fragmented workflows and inconsistent results across different databases. This can create a lack of visibility into the overall migration progress and make it difficult to track and manage the various components of the migration process.

Solution: Get help from the credativ-pg-migrator tool

To address these challenges, we created our new tool **credativ-pg-migrator**. The tool simplifies and unifies the migration process from various source database systems to PostgreSQL. It provides a single, consistent interface for migrating applications from different database engines, making it easier for IT teams to manage the migration process and ensuring that all migrations are performed in a standardized way. We already presented this tool on the PostgreSQL Conference Germany 2025.

What is this all about?

The **credativ-pg-migrator** is an offline migration tool built in Python, leveraging stable and well-documented libraries for database access (JDBC, ODBC, or native Python drivers). Python language was chosen because it offers stable libraries for multiple databases across their different versions. The migrator's architecture is modular, comprising a Parser, Planner, Orchestrator, and parallel Workers, enabling efficient data transfer with speed primarily dependent on hardware capabilities. The tool also meticulously logs all migration details into dedicated protocol tables and outputs comprehensive information about each migration step into log file. This ensures that users can easily and precisely check the progress and results of the migration.

Supported foreign databases

In the current version, the tool allows migration of data models (tables, data, indexes, constraints, views) from Oracle, Informix, IBM DB2 LUW, Sybase ASE, MS SQL Server, MySQL/MariaDB, and SQL Anywhere. For Informix, the tool also offers conversion of stored procedures, functions and triggers into PL/pgSQL code. The success rate of this conversion is up to 80 to 90% depending on the writing style of the original code. This is because old databases often allowed code to be written in a very free style. Conversion of stored procedures, functions, and triggers can be relatively easily extended to other databases too. However, to make it working properly, we need to adjust it on real-life migration examples. Every database has some specific use cases and every developer has its own style.

The tool also implements several customization to the migration process. For example, it allows custom replacements for specific source data types based on multiple criteria. This is useful when the source database contains user-defined data types or when it allows in a data model specific legacy design patterns, no longer supported by modern solutions. Users can also configure custom-defined replacements for default values of columns, which is important because SQL functions in legacy databases are frequently different from the ones used in PostgreSQL. And migration of tables data can be limited by custom defined conditions. This is useful for example when the source database contains a lot of historical data, and only the newest records should be migrated to PostgreSQL.

Future enhancements

We are already working on further enhancements to this tool. In coming versions, we add support for partitioning of target tables. This feature will also include pre-migration analysis of the source database. Analysis will check multiple aspects of the source database, including data distribution in large tables to help to choose the best partitioning strategy for the target PostgreSQL database. This is very important feature, because partitioning schemas used on legacy databases, including sub-partitions and sub-sub-partitions, are often not optimal for PostgreSQL.

Are you interested in using this tool for your database migrations? [Contact us](#) to discuss your specific use cases and requirements. We are happy to assist you with your database migrations, including adding new features to the tool to meet your specific needs. When the code is stable enough, we will properly open source it so the whole community can benefit from it. Please reach out to us for more information.

**ABOUT THE AUTHOR****Josef Machytka**[View posts](#) 