splendiddata

Splendid Data white paper

# Automated migration of Oracle® databases to PostgreSQL

Cortex highlights

This white paper is a working document that is subject to review and changes.
Version 1.1 October 2026

# Contents


**Introduction** 3

**Migrating Oracle databases to native PostgreSQL** 4

**Cortex Migration** 6

**Additional services** 10

**Contact** 11




White paper version 1.1 © 2026 Splendid Data

# Introduction

## About Cortex

Cortex is the most sophisticated product available in the market to perform automated migration of your Oracle databases to native PostgreSQL. Our experts that built and continuously improving Cortex have in-depth technical knowhow of both Oracle databases and PostgreSQL as well as parser and compiler techniques.

Translation of the Oracle database Data Objects and Code Objects (PL/SQL to PL/pgSQL) is done semantically in relation to their context (including dependencies throughout the PL/SQL code). All other products in the market are mainly based on “find and replace” without any dependency checks throughout the PL/SQL code, which makes the overall automatic translation of these products unreliable and inefficient, and basically inappropriate to setup a larger scale migration of Oracle databases to PostgreSQL program/project.

Because we believe in database freedom, we always migrate Oracle databases to native PostgreSQL to avoid any vendor lock-in and to deliver maximum freedom of deployment on- premise and/or in the Cloud. You can use Cortex on-premise or on your Azure tenant (see Azure Marketplace). Cortex is 100% secure, well documented and supported. Our Cortex product doesn’t leave a footprint behind, so there will be no vendor lock-in after using Cortex for a migration your Oracle database to PostgreSQL.

Some of our references that are convinced of what we promise and deliver

# Migrating Oracle to native PostgreSQL

Cortex is developed with the aim of automating the migration from Oracle databases to PostgreSQL and providing as much flexibility as possible using settings to achieve the desired and accurate results.

In order to achieve the most accurate results, Cortex performs the translation of the Oracle database Data Objects and Code Objects (PL/SQL to PL/pgSQL) semantically in relation to their context (including dependencies throughout the code).

## What does Cortex migrate to native PostgreSQL?

| |
|---|
| Tables and Columns (incl. defaults) |
| Constraints (incl. unique, primary key, foreign key, not null and check constraints) |
| Indexes (incl. function based) |
| Triggers (incl. before each row, after each row, compound etc.) |
| Partitions and subpartitions |
| Views and materialized views |
| Types (composite types, collection types and subtypes) |
| Synonyms (will be taken into account where applicable) |
| Sequences |
| PL/SQL (packages/package bodies, functions, procedures, global variables, global constants, etc.) |
| References to dblinked tables |
| Spatial and Geographic objects (Oracle Locator and Spatial to PostGIS) |
| XML |
| Merge statements |
| External SQL's |
| "connect by" queries |
| Old Oracle-style outer joins "(+)" |

Cortex is available via container technology (e.g. OCIv2). Depending on the target version of PostgreSQL, starting at version 14 up to the latest release, a Cortex Image is available. From a Cortex Image a Cortex Runtime Container is created.

Cortex has graphical user interface to steer, influence and manage each Oracle to PostgreSQL migration.

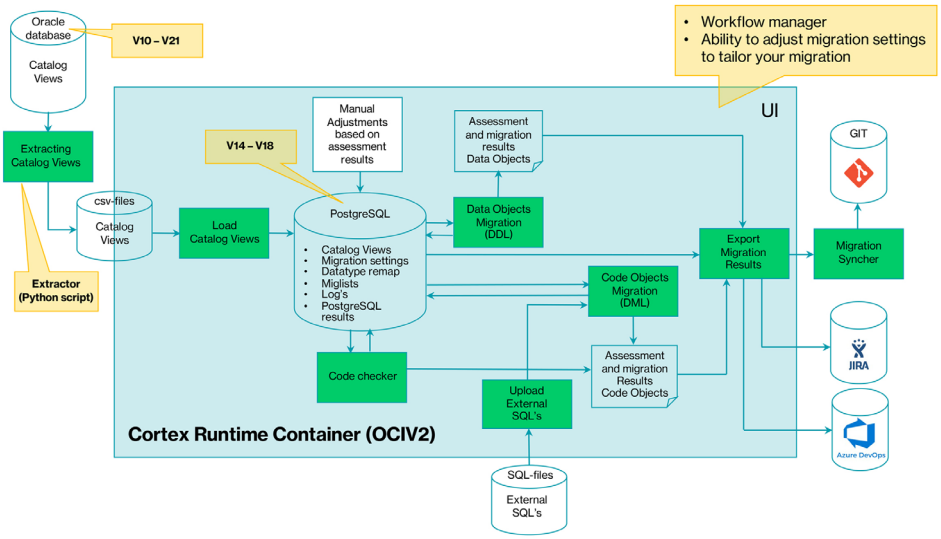
## Benefits of using Cortex for Migrations of Oracle databases to PostgreSQL

- **Proven technology.** Cortex is used by Splendid Data, its partners and customers to migrate Oracle databases to native PostgreSQL.
- **Secure deployment and usage.** Cortex runs on-premises or in your own Azure tenant. It requires no direct connection to the source Oracle databases. A provided Python script extracts the required Oracle Catalog Views into CSV files. No backdoors and fully transparent.
- **Comprehensive documentation and expert support.** Detailed documentation covers installation, configuration, migration workflows, reporting and post-migration activities. Expert assistance is available through a dedicated ticketing system.
- **Standardized and repeatable.** Cortex supports a factory-based migration approach, enabling consistent processes across multiple databases and teams.
- **High degree of automation.** Cortex automates the conversion of Oracle data and code objects to native PostgreSQL, including PL/SQL to PL/pgSQL, substantially reducing manual remediation.
- **Full traceability.** Extensive logging, reporting and version management provide an audit trail of source metadata, migration settings, results, changes and outstanding manual adjustments.
- **Flexible configuration.** Migration settings allow control over PostgreSQL implementation choices, including data type mapping, partitioning, transaction methods, usage of collection types and many more.
- **Version-aware migration.** Cortex accounts for PostgreSQL Community supported releases, generating output appropriate for the selected target PostgreSQL version.
- **Faster, more cost-effective migrations.** Automation reduces repetitive work, improves consistency and enables teams to migrate more databases with fewer resources.
- **Native PostgreSQL, without Cortex dependency.** Migrated objects use native PostgreSQL rather than proprietary compatibility layers. Teams retain control over development, deployment and maintenance using standard PostgreSQL tooling. Cortex leaves no runtime dependency behind... Guaranteed NO VENDOR LOCK-IN after Cortex did its work.
- **TCO between 12 and 18 months is feasible.** Of course this can very case-by-case. In case facing Oracle license renewals / extensions it can be even less than 12 months and when only savings on maintenance fees it could be longer than 18 months.

# Cortex Migration

## How does it work?

Per Oracle database, which is going to be migrated to PostgreSQL, a Cortex Runtime Container is created (spinned). A container contains all the software components (incl. a PostgreSQL database) needed to perform a migration.



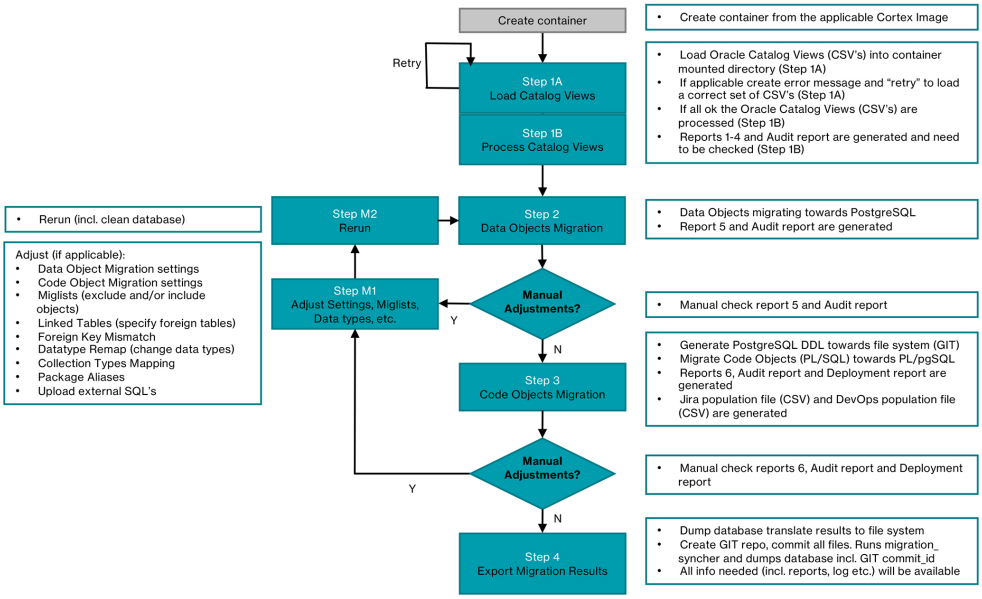

Cortex overview

The PostgreSQL database plays an important role in the container. This database is used to store the Catalog Views of the Oracle database to be migrated, all controls like migration settings, datatype changes (remaps), excluded objects and log data, as well results of the database migrated to PostgreSQL.

| Software Components | Description |
|---|---|
| Extractor | Python script that extracts (by reading) the applicable Oracle Catalog Views and writes the results in csv-files. |
| Loader | Loading the extracted Oracle Catalog Views (csv-files) into a PostgreSQL database |
| Migration Toolkit Interface | Steering the assessment and migration process of the DDL and DML related to the Oracle Catalog Views to PostgreSQL |
| Graphical User Interface | Managing Cortex Images and Containers, and steering the assessment and migration process incl. manual interventions (like: settings, miglists, datatype remap, etc.) |
| Data Objects Migrator | Assessment and migration of the DDL related Oracle Catalog Views to PostgreSQL |
| Code Objects Migrator | Assessment and migration of the DML related Oracle Catalog Views to PostgreSQL |
| Code Checker | Analyses PL/pgSQL code to flag programming errors, bugs, stylistic errors, and suspicious constructs. Open Source Software with Copyright owned by Pavel Stehule (https://github.com/okbob/plpgsql_check). |
| PostgreSQL | Target database used during the assessment and migration. Open Source software, MIT/PostgreSQL licensed |
| PostgreSQL extensions | All kinds of Open Source PostgreSQL extension to support the assessment and migration of the DDL and DML related to the Oracle Catalog Views to native PostgreSQL |
| Migration Syncher | Synchronizing PL/pgSQL code to GIT. Open Source Software with Copyright owned by Splendid Data (https://github.com/splendiddata/migration_syncher). |

Cortex software components

The migration of an Oracle database consists of several steps that can be initiated via graphical interface or the command line interface.




Cortex workflow

Based on the various assessment/migration reports produced during the different steps, insight is gained into the extent to which the migration is automated. This information is valuable because it can be used to make adjustments (e.g. adjust a datatype because Cortex has found a foreign key mismatch, exclude certain objects before the migration, etc.) and/or to change settings used during the migration.

Once changes have been made, a rerun can be initiated. The migration results produced so far are deleted and based on the original Oracle database Catalog Views, the adjustments and the changed settings Cortex will rerun the previous step(s). This process can be repeated until the desired results are achieved. These results can be transferred to your own development environment to complete the migration to perform the manual adjustments. For issue management Cortex produces csv-files for Jira and Azure DevOps.

Cortex literally keeps track of everything (audit trail). It is always clear which settings have been used, what has been adjusted (like datatype remap), which object have been excluded, what went right, what went wrong and why did something went wrong.

Because Cortex takes dependencies between objects into account to an extreme degree, it is important to have a correct migration of the Oracle database Data Objects to PostgreSQL. Based on the reports provided it can be checked if specific adjustments are needed. The “better” the implementation of the Oracle database Data Objects to PostgreSQL is, the “better” the results of the migration of Oracle database Data Objects to PostgreSQL will be.

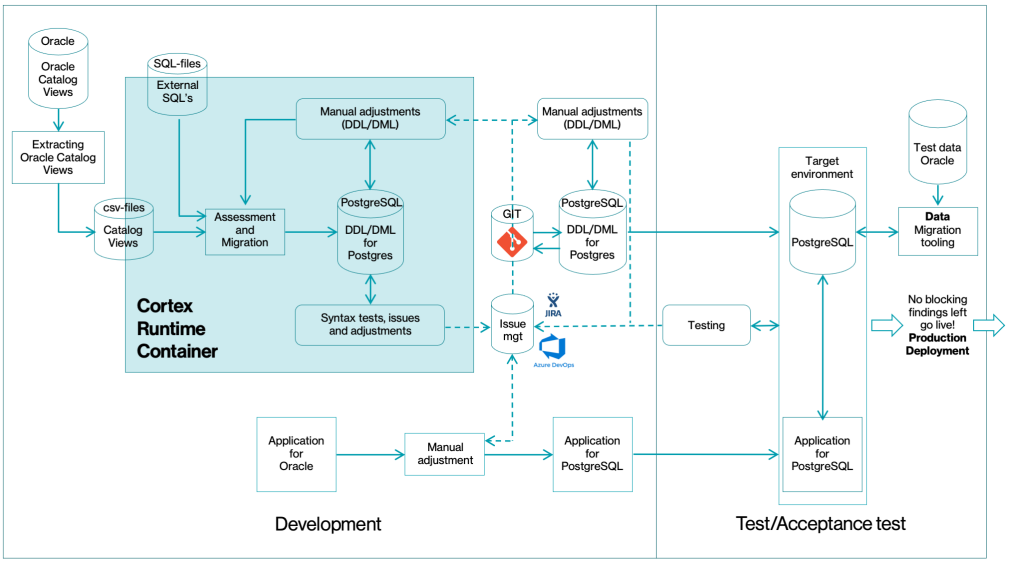
## Continuous improvement and maintenance (Kaizen)

Given the many implementation occurrences of the Oracle database Data Objects and Code Objects, Cortex is continuously improved and maintained, and new Cortex releases will be provided frequently. This is done based on the usage of Cortex and finished migration programs/projects worldwide. Therefore, it is desirable that during the usage of Cortex there are contact moments on regular basis between the clients and partners (both users of the Cortex product) and Splendid Data to provide feedback to improve and maintain Cortex.

This continuous improvement process is based on Kaizen and iterates on the steps:

1. Find unexpected outcomes or problems
2. Create solutions
3. Test solutions
4. Analyse results
5. Standardise
6. Repeat previous steps

## Project flow by using Cortex



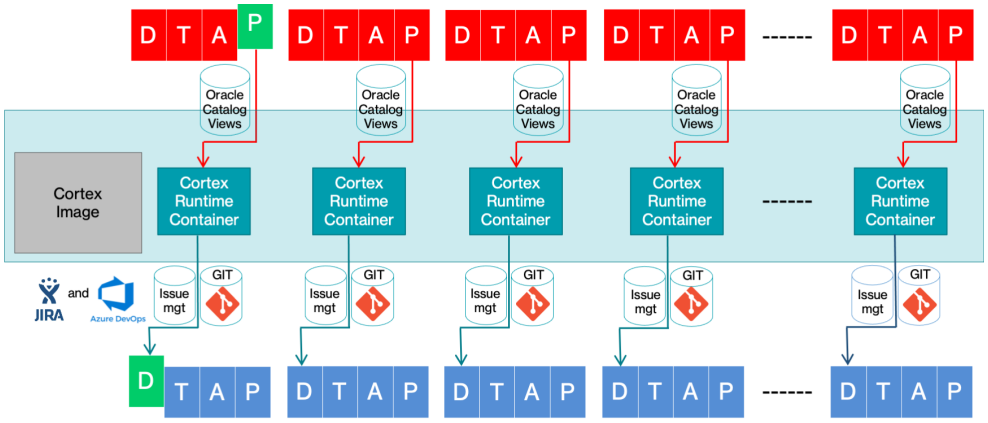

Standard project flow

Each migration of an Oracle database to PostgreSQL is a project on its own. After a migration performed with Cortex the results are available in GIT (version management) and the manual adjustments to be performed are available in Jira or Azure DevOps (issue management).

Besides the database, in almost any case, also the application using the database needs to be adjusted (outside the scope of Cortex). As soon as both database and application are finalized from a development point of view, the next steps like testing, acceptance/ performance test etc. need to be performed. At the stage where no blocking findings (issues) are left, the final step “production deployment” can start.

## Using Cortex in a factory-based approach

For organisations that need to migrate a large number of Oracle databases to PostgreSQL a factory-based approach supported by the usage of Cortex is the only setup to do this in a controlled and unified way.




Cortex factory approach

From a Cortex Image the Cortex Containers are created (spinned). Each container has its own life-cycle. It always starts with uploading the meta data of the applicable Oracle database (production) to be migrated to PostgreSQL. As soon the meta data is available in the container the migration process can start. Always check the reports Cortex produces and use these reports to make adjustments and to change settings to end-up with the best migration results. After finalizing the migration you can transfer your results (incl. all reports) to your own development environment. After everything is transferred, the Cortex container can be stopped and removed.

Cortex promotes a highly standardized approach and it's possible to create an overall factory approach in different regions staffed with your own people and/or staffed with people of your preferred partner(s). Cortex allows running migrations in parallel to feed the different project teams. Cortex avoids any vendor lock-in and any technology push. All objects are migrated to native PostgreSQL and manual adjustments can be made using the IDE tooling of choice.

# Additional services

## Support

Splendid Data provides support during the use of Cortex Image(s) and the Cortex Runtime Container(s) and/or Software Components delivered by Splendid Data separately, in case of questions regarding the functioning.

## Training workshops

Splendid Data provides specific training workshops regarding the use of Cortex in combination with specific hands-on knowledge, by providing insight into the differences between an Oracle database and PostgreSQL, in order to successfully perform the migration from an Oracle database to PostgreSQL.

## Consulting services

If necessary or desired, the migration experts of Splendid Data can be hired for consulting to contribute to the successful migration of your Oracle databases to PostgreSQL.

## Data migration

**Test and acceptance test**
Splendid Data can provide scripting to migrate test data from the Oracle database to the PostgreSQL migrated database based on the PostgreSQL extension Oracle_fdw (Foreign Data Wrapper).

**Production deployment**
Splendid Data can advise you on which tooling is best to use given the data volume, available downtime, deployment (on-premise, cloud managed etc.) and potential special circumstances to consider.

# Contact

## Ready to benefit from PostgreSQL by using Cortex?

Please feel free to contact us if you have any questions left or to setup a meeting for a demo of Cortex.

**Michel Schöpgens**
michel.schopgens@splendiddata.com
+31 85 773 19 99 (office)
+31 6 54 34 30 89 (mobile)

**Pascal Boutin** (for the French speaking markets)
pascal.boutin@splendiddata.com
+33 6 11 69 68 64 (mobile)

**Splendid Data**
Binnenhof 62a
1412 LC NAARDEN
The Netherlands

## Find us online

www.splendiddata.com
www.linkedin.com/company/splendiddata

## About Splendid Data

Splendid Data was founded in 2013 and since then our focus has been purely on PostgreSQL and to pass on its benefits to our clients and partners.

Our specialist team that has many years of in-depth knowledge of Oracle databases and PostgreSQL, enabling us to build **PostgresPURE** – our 100% open source subscription-based software that makes the most of PostgreSQL. At the same time, we developed **Cortex**, our unique solution for the automated migration of Oracle databases (DDL and DML) to native PostgreSQL. And to give organisations a full overview of their Oracle database environments and its suitability for migration, we created **Migration Scoping**, a clever piece of software that takes out all the guess work of a potential migration to PostgreSQL.

We're proud to assist companies of all shapes and sizes to find their way to digital liberty, agility and the power of PostgreSQL. Get in touch today to start your journey to a world without Oracle databases.

Confidential information owned by Splendid Data, to be used by the recipient only. This document, or any part of it, may not be reproduced, copied, circulated and/or distributed nor quoted without prior written approval from Splendid Data.

splendiddata

White paper version 1.1 ©2026 Splendid Data