Replacing Oracle with Postgres: How To Successfully Migrate Your Legacy Databases

AUTHORED BY:

Matthew Lewandowski
Field CTO, EDB

Marc Linster, Ph.D.
Chief Technology Officer, EDB

POWER TO POSTGRES

Contents

INTRODUCTION

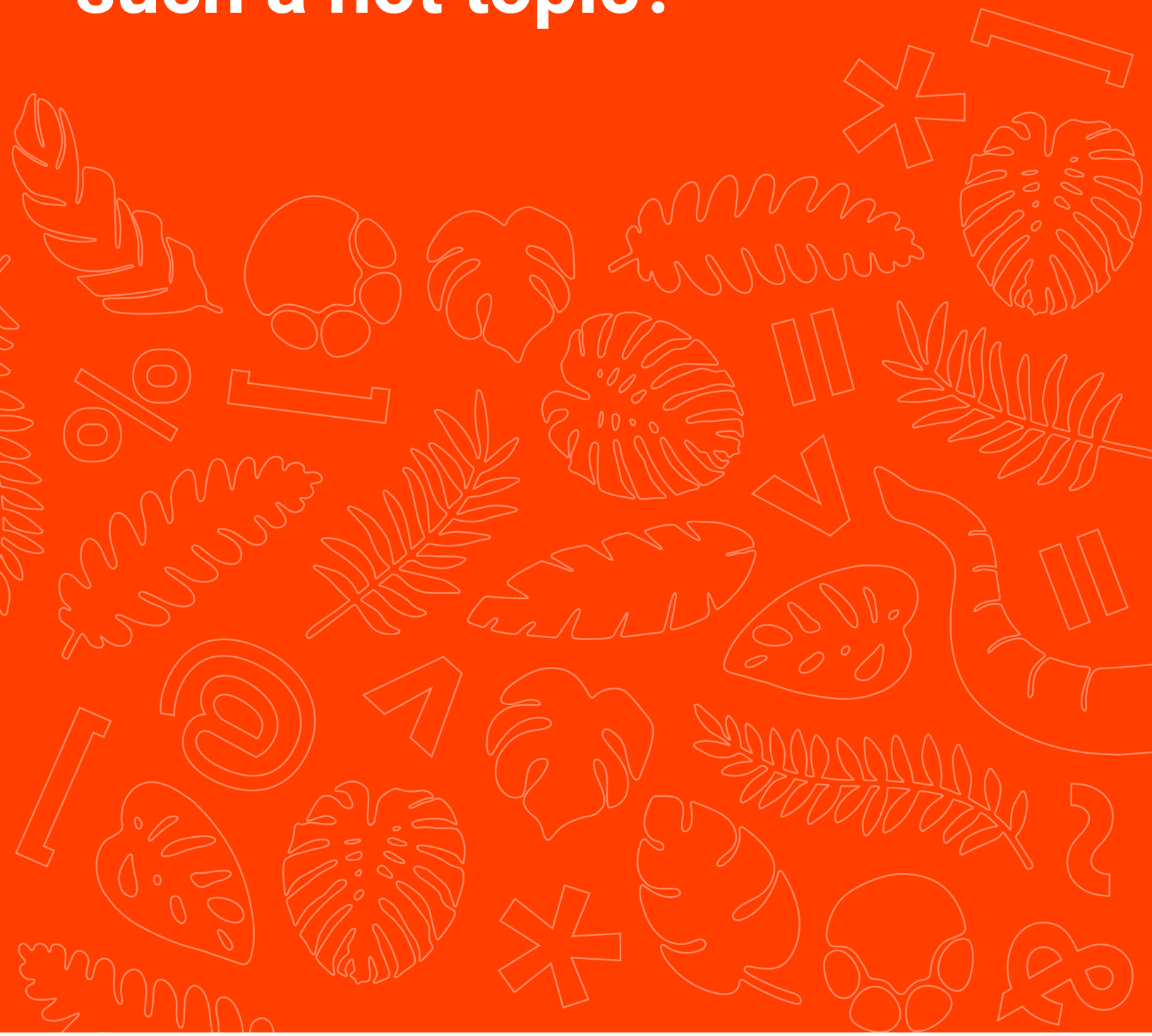
This white paper focuses on the most popular source and target for database migrations: moving from Oracle to Postgres. Oracle is the #1 legacy database, and its extremely onerous license policies are driving the majority of migration demand.

Postgres is the logical target for the migrations. With a constant stream of innovations reflected in annual releases, Postgres has achieved three major database of the year awards from DBengines.com, recognition as the #1 database in StackOverflow's annual developer survey, and the position of #1 database on the Cloud Native Computing Foundation's tech radar. Not only is it clear that Postgres is winning the hearts and minds of innovation drivers, but its small footprint makes it an ideal solution in containers too (see Datadog survey).

The principles and approaches described in this paper are applicable to other source/target combinations as well. You'll find:

- A quick review of the business drivers and migration approaches;
- A dive into the migration journey and its challenges;
- The best tools to get off Oracle quickly;
- How to start taking advantage of Postgres' innovation, agility, and cost effectiveness.

1. Why is legacy database migration such a hot topic?



Why is legacy database migration such a hot topic?

Traditionally, database license cost was listed as the primary reason driving organizations away from legacy databases—such as IBM's DB2, Microsoft's SQL Server, and Oracle—to open source databases such as Postgres. IDC and others have developed **great business cases**, and **Gartner** predicted that open source databases such as Postgres can handle the majority of workloads.

While replacing Oracle with Postgres can yield cost reductions of upwards of 80%, agility, innovation, microservices, and the move to the Cloud have recently emerged as the dominant drivers.

Microservices, or the move away from monolithic databases, has become a key incentive for the move away from legacy databases. Because monoliths typically support many applications, it's difficult, if not near impossible, to deploy changes, and adjust scaling for individual services.

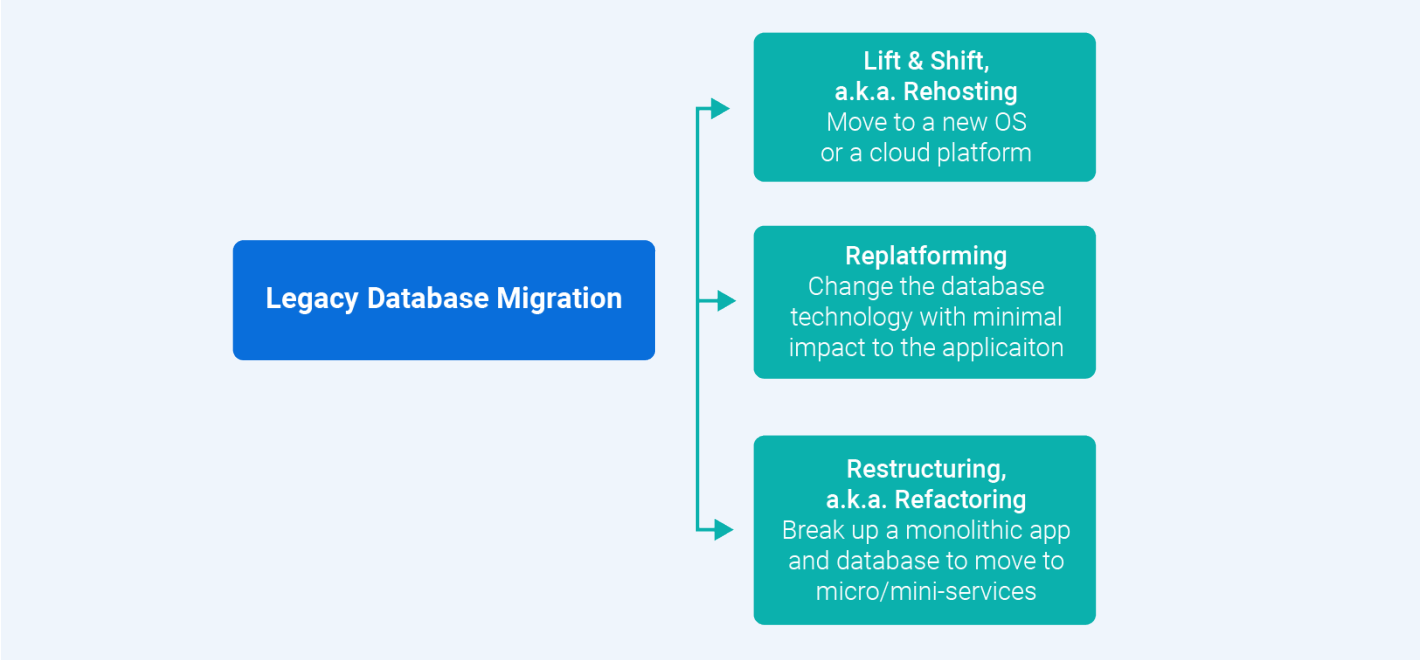
From an IT leadership perspective, we see license constraints and the desire to move away from proprietary data centers as two major drivers. Onerous legacy licenses drive cost up while constraining innovation and agility. For example, not every license is portable to every virtualization platform, every Cloud, or every operating system—without even talking about scaling up and scaling down.

2. Different types of database migrations

Different types of database migrations

Legacy database migrations come in several forms1:

- Lift and shift
- Replatforming
- Restructuring



The term “Lift and shift” or Rehosting is typically used to describe the move to a new host platform, but without changing the underlying software stack. For example, one can lift and shift an existing application that uses an Oracle database on Linux in the data center to a cloud IaaS, or a PaaS service that supports a managed version of the Oracle database.

Replatforming refers to a migration that exchanges the underlying database platform, with minimal or no changes to the application. For example, a migration of an application that uses an Oracle database to the Oracle compatible EDB Postgres Advanced Server (EPAS) database is considered replatforming, as the application typically does not need to be modified.

Restructuring or Refactoring is a more radical approach, where a monolithic legacy application is transformed into multiple smaller applications, which often impacts the backend database. The database can be broken up, or be equipped with services interfaces to support a more modular application architecture. The restructuring approach is typically taken during microservices transformations that result in container and Kubernetes based architectures.

Migration	Pros	Cons	When Considered
Lift and shift	• Fast, easy, limited technology risk • Easy way to move from the datacenter to the cloud • No changes to the application or the database	• Does not resolve legacy license issues or reduce license cost • Does not improve agility or innovation	• End of life applications that need to move out of the data center • The move out of the DC is more pressing than the need to innovate or reduce software cost
Replatforming	• Easy way to reduce database license cost and eliminate platform restrictions • Minimal changes to the application and existing integrations • Take advantage of Postgres innovation	• May require minor application changes and retesting	• High license cost databases with longer-term strategic role • Use of database logic (stored procedures, packages) or proprietary features
Restructuring	• Move to open source platform • Highest degree of agility • Supports move to K8s and containers	• Significantly longer migration cycle with higher technology risk	• Applications that are part of larger scale digital transformation with a business case supporting extensive redesign

3. Migration techniques and technologies

Migration techniques and technologies

A database migration from legacy to open source includes:

- Transforming the schema from a proprietary vendor's extended version of the SQL standard to a version more compliant with standards
- Rewriting data type definitions
- Rewriting queries and stored procedures
- Copying data
- Updating application APIs to use open source JDBC, .NET, ODBC, as most vendors have extended the standard protocols with proprietary extensions
- Verify that the migrated database meets all the non-functional requirements related to performance, manageability, high availability, and integration with enterprise security requirements.

Executing these transformations manually can be a very daunting task. Except for very small databases without any business logic and extremely simple application logic, we would not recommend this approach. It is too expensive and too error-prone.

That is why two automated approaches are well established today:

The translation approach uses automated tools to rewrite (or translate) the definitions, queries and stored procedures from the proprietary database to the open source database. The translation approach is used by AWS's Schema Conversion Tool (SCT), Ispirer's MnMTK, and the open source tool ORA2PG.

- The **native compatibility approach** extends the open source databases' capabilities and creates a native implementation of the proprietary vendor's extensions of the SQL standard, including the APIs and protocols. For example, EDB's Postgres Advanced Server (EPAS) has a native implementation for Oracle's procedural SQL language PL/SQL, which includes packages and Oracle's proprietary driver extensions to ODBC, JDBC, .NET and OCI. This allows code that was written for the Oracle database to run directly on EPAS with minimal changes. The native implementation approach is used by EDB as part of EPAS to achieve compatibility with the Oracle database. The open source tool ORAFCE also attempts to provide some level of compatibility with Oracle, although not to the same degree as EPAS.

Approach	Pros	Cons	When Considered
Manual Transformation	• Targets open source Postgres	• Significant effort required • Significant risk of error	• Very small, non-mission critical databases
Translation	• Targets open source Postgres	• Limited ability to accommodate all proprietary capabilities • Potentially requires significant database logic rewrites • Often requires application modification as non-standard/ proprietary APIs and SQL may not be supported in Postgres	• Simple databases using only standard SQL queries. • Databases and applications that do not use stored procedures, packages, or proprietary data types • Application interfaces limited to standard APIs
Native Compatibility	• Migrates majority of proprietary capabilities (> 90%) • Minimal application rewrites	• Targets proprietary version of Postgres (EDB Postgres Advanced Server)	• Databases with business logic (stored procedures, packages, queries) • Databases targeted by Oracle DBLink

3.1 Comparing the migration techniques

Take for example one of our large media customers: they migrated their Oracle database containing 10,938 database objects (tables, stored procedures, packages, etc.) in 35 person days, including data transfer and testing, as 91% of all the objects were supported by EDB's native compatibility. Only two packages, which included approx 9% of the code needed to be re-architected for compatibility with EPAS.

In the same application, only the storage objects (tables and views) and some functions were open source Postgres compatible using the translation approach with ORA2PG. The rearchitecting approach for 65% of the code was estimated to be between 1.5 and 2 person years of effort.

In another example, a telecom provider was migrating a database with 15 procedures and nine functions, all defined in one custom Oracle package. The logic made extensive use of DBMS_LOB, DBMS_SESSION, DBMS_XMLGEN.convert(), and the PIPELINED table function.

This customer targeted a migration directly to open source Postgres, which required reworking all the business logic in a migration project, requiring 35 person days of effort using the translation approach. Using EPAS' native compatibility approach, this migration could have been executed in approximately three person days, with minimal impact to the application and with minimal retesting.

4. Legacy database migrations and the cloud

Legacy database migrations and the cloud

Data center closures and migration to the cloud are key drivers for database migrations. When migrating databases to the cloud, the user is left with several options, even after deciding to move to Postgres:

- Private cloud, such as OpenShift, Nutanix, or VMWare
- Public Cloud Infrastructure as a Service (IaaS), such as AWS EC2 or GCP
- Public Cloud Kubernetes Platforms, such GKE, AKS, or EKS
- Public Cloud Database as a Service (DBaaS), such as RDS Postgres, Aurora for Postgres, Azure Database for Postgres, Google's Cloud SQL for Postgres, or EDB BigAnimal

While they all implement the Postgres API, there are key differences in migration capabilities. EDB's native compatibility with Oracle is available on OpenShift, VMWare, all IaaS and K8s platforms, and on EDB BigAnimal. RDS, Aurora, Azure Database for Postgres, Google's Cloud SQL for Postgres are limited to thetranslation approach.

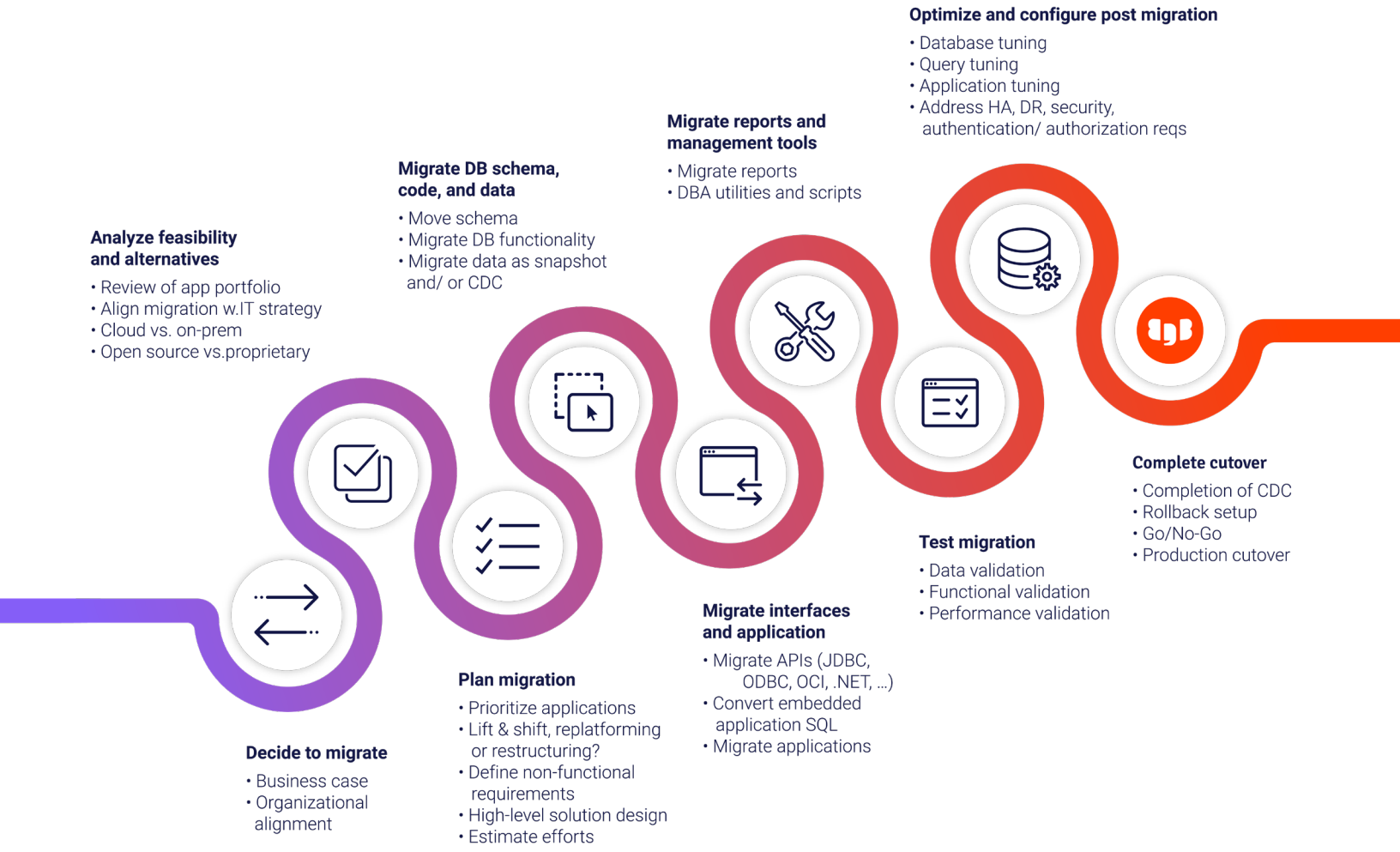
From a non-functional perspective, IaaS, VMWare, OpenShift and K8s platforms provide the highest level of control for performance, manageability, and integration - but they also require in-house deployment and management resources. Leading DBaaS platforms address these issues, but they limit configurability for non-functional requirements.

Data gravity makes it extremely important to think about the target platform early in the process. Even if you start a migration project with a simple database that is well served with the translation approach to migration, a later stage application may require capabilities that are only available as part of the native compatibility approach, such as legacy/proprietary database APIs or database links from other Oracle databases.

The same is true when selecting the cloud platform. Initial migrations may be served well by a DBaaS provided by a Cloud Service Provider, such as AWS or Azure, for whom Postgres is just another software platform that they operate. More advanced applications may need greater access to tuning parameters, as is provided on IaaS platforms, or the services of a Postgres specialist such as EDB.

5. The nine steps of the migration journey

The nine steps of the migration journey



EDB has been migrating databases for over 15 years. While most of our migrations have been from Oracle or SQL Server to Postgres, we have also executed major migrations from DB2.

Experience shows that the enterprise migration journey follows nine steps:

Step	Focus	Outcome
Decide to migrate	• Business case • Business priority	• Decision to undertake migration at scale • Get off legacy databases
Analyze feasibility and alternatives	• Review the application portfolio • Align the migration with the IT strategy and priorities • Decide if we go to public cloud	• Large scale plan to migrate • Migration targets: which cloud, which database (open source/closed source)? • Organizational alignment

Plan migration	• Prioritize applications • Lift & shift, replatforming or restructuring? • Define non-functional requirements • High-level solution design • Estimate effort	• Prioritized list of applications and databases, each identified as lift & shift, replatform, restructure, or leave behind • Performance, HA, and management integration requirements • Migration architecture and high-level post-migration architecture • High level effort estimate, skills requirements, and staffing plan
Migrate DB, code and data	• Move schema • Migrate DB functionality • Migrate data as snapshot and/ or CDC	• Functional database with all or some of the data • Clear understanding of the need for CDC and migration cutover timelines • Valid proof of concept that definitions, code and data can be migrated • Understanding of any gaps and effort assessment to close the gaps
Migrate interfaces and application	• Migrate APIs (JDBC, ODBC, OCI, .NET, …) • Convert embedded application SQL • Migrate applications	• Running application that meets functional requirements • App and database are integrated
Migrate reports and management tools	• Migrate reports • DBA utilities and script	• DBA tools (data loading and other management) are functional • Reports generated through SQLPlus or other means, e.g., Tableau, Qlik
Test migration	• Data validation • Functional validation • Performance validation	• Proof that the data is migration is working correctly • Proof that the code and the APIs are working correctly • Proof that migrated database and code meet performance requirements
Optimize and configure post migration	• Database tuning • Query tuning • Application tuning • Address HA, DR, security, authentication/ authorization reqs	• Updated indexing strategy • Validation of non-functional requirements • SOP (Standard Operating Procedures) and DevOps automation
Complete cutover	• Completion of CDC • Rollback setup • Go/No-Go • Production cutover	• Completed migration

6. What makes Oracle migrations hard

What makes Oracle migrations hard

A survey of 1,500 respondents from the EDB Postgres downloads page shows that data definitions are usually easy to migrate, but stored procedures, APIs, and the data are increasingly difficult. While migrating data may initially appear easy, mapping of data types can be challenging, and the incremental data migration—a.k.a. Change data capture (CDC) that is required for larger databases—can be impossible without the right tooling.

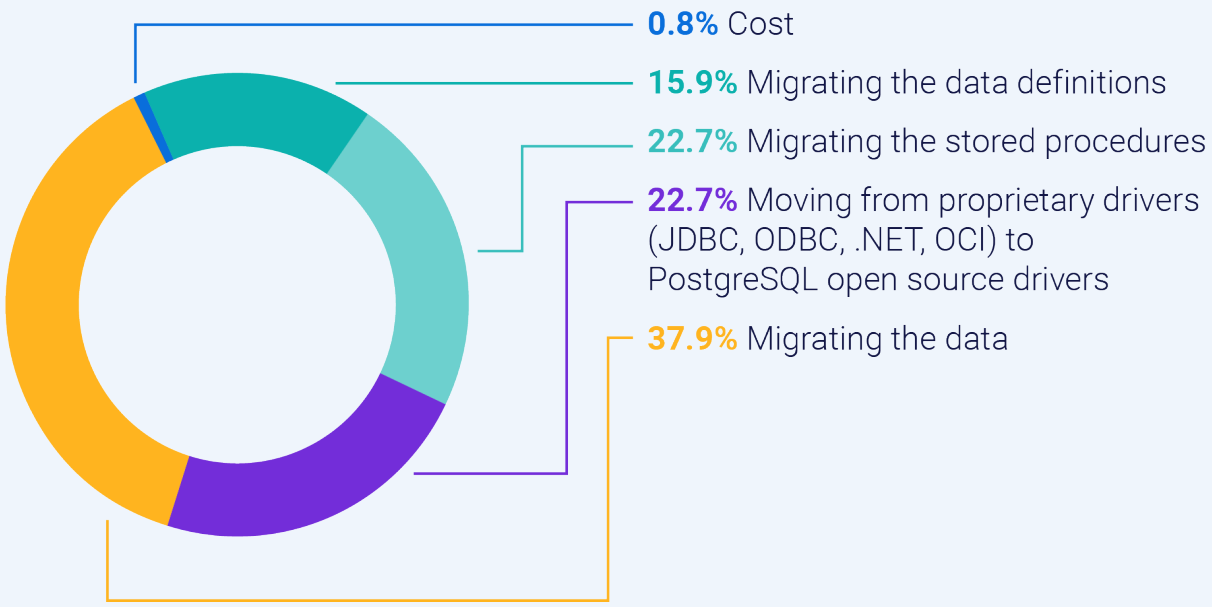


Figure 1: A survey of 1,500 from EDB's Postgres download page (July 2020)

Almost a quarter of the respondents mentioned that migrating from Oracle's proprietary drivers to open source Postgres was the second most difficult thing to do. This is because:

- There is no open source equivalent for Oracle's OCI driver or for Oracle's Pro*C interface
- The 'standard' drivers (JDBC, ODBC, .Net) have been heavily extended to support calling stored procedures, in/out parameters, and cursors.

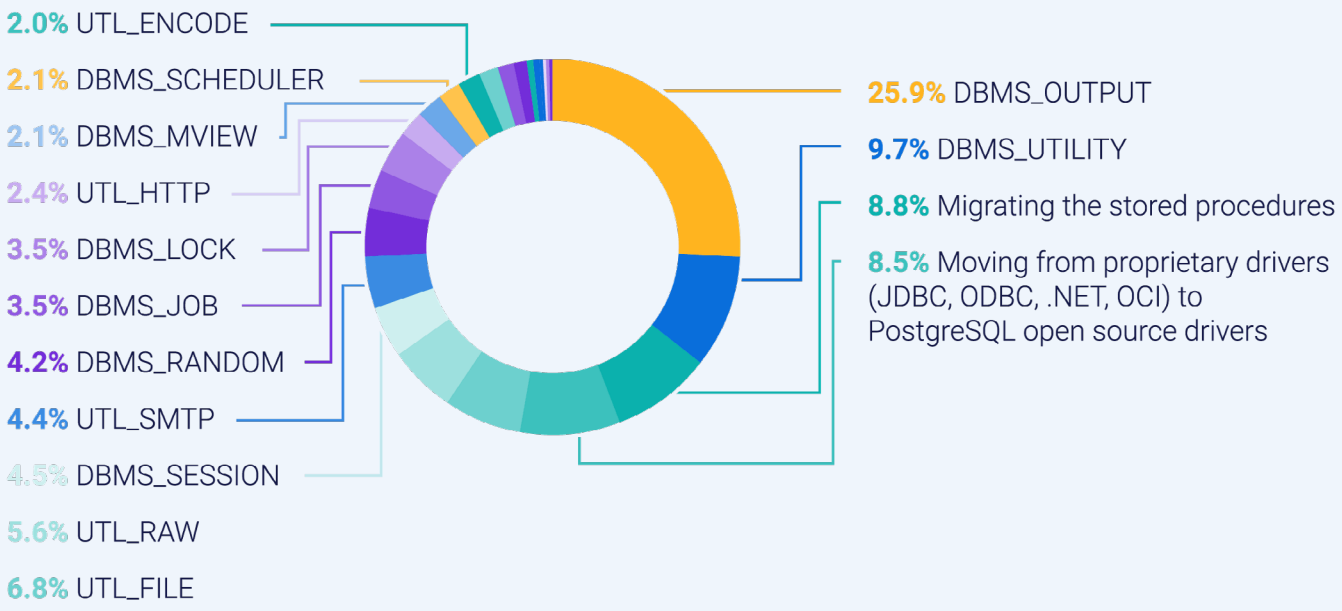
Replacing the proprietary drivers with their nearest open source brethren implies significant modifications in the application logic.

Business logic, mostly stored procedures, is generally seen as a major migration obstacle during manual migrations or when one uses the translation approach. EDB's migration portal is a public website that allows anybody to assess their Oracle DDL for migratability to EDB Postgres Advanced Server. Since January 2019, we have analyzed over 18 million DDL constructs (CREATE TABLE, CREATE STORED PROCEDURE, etc.) for our customers and helped them migrate to EPAS.

This analysis reveals important data that should influence a migration plan:

- 14% of all schemas had at least one reference to PRAGMA AUTONOMOUS_TRANSACTION
- 14% of all schemas had at least one HINT
- 32% of all schemas referred to at least one of the EDB supported Oracle packages, with DBMS_OUTPUT, DBMS_SQL, DBMS_UTILITY, and DBMS_LOB representing the majority of those packages

Occurence of EDB-supported Oracle Packages

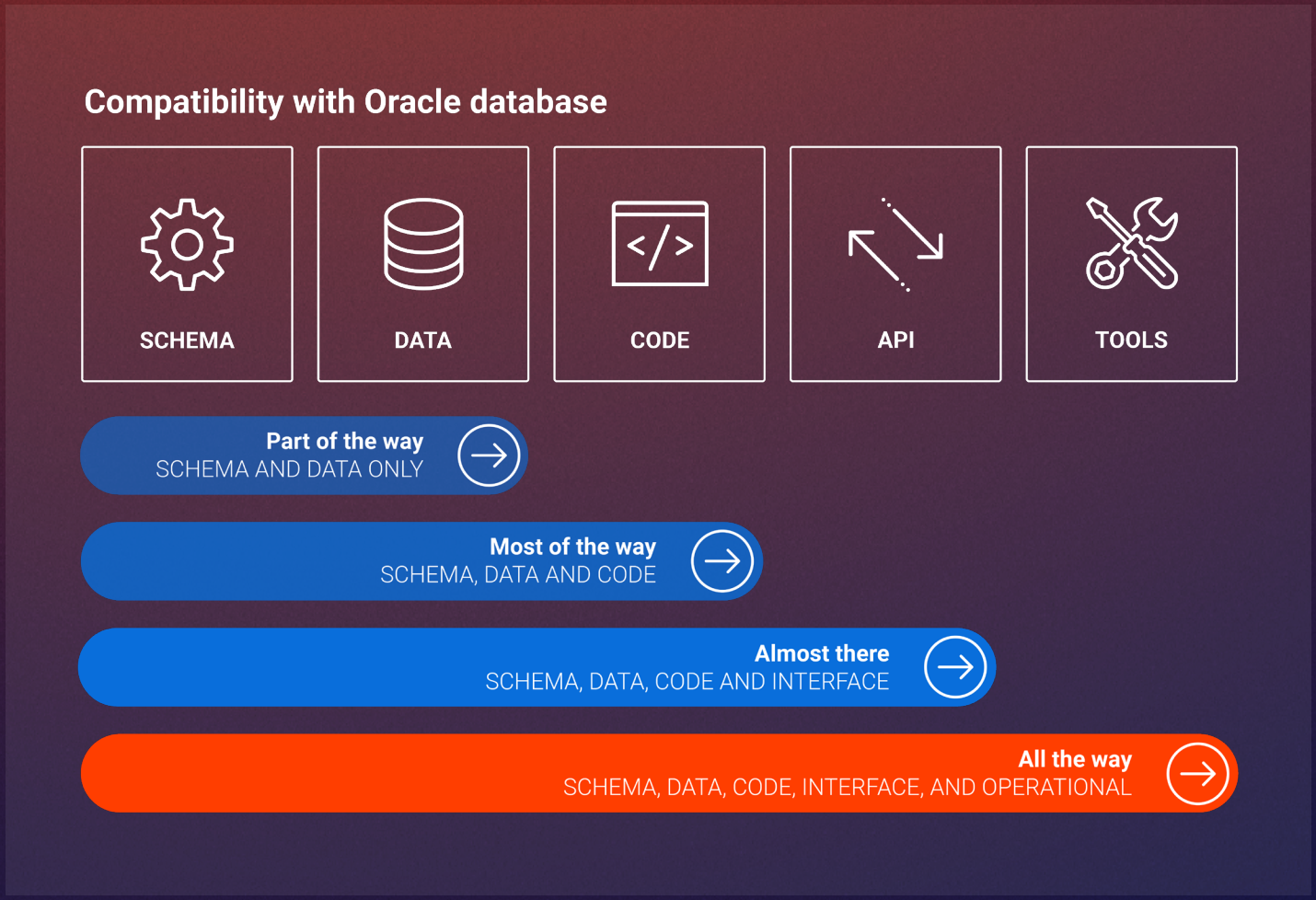


Some of these elements, such as PRAGMA_AUTONOMOUS_TRANSACTION or HINT, are virtually impossible to reproduce in Postgres, which means that a translation approach will require extensive rewrites.

7. What EDB brings to the table

What EDB brings to the table

EDB has enabled legacy database migrations for over 15 years. We have assembled a systematic set of tools to make migration easier, predictable, and risk free. We have chosen the native compatibility approach, and we have learned that successful migrations are not limited to the data and stored procedures, but that a complete business solution requires the APIs and operational tools.



- EDB Postgres Advanced Server is at the heart of our approach. EPAS is an Oracle Database compatible distribution of Postgres that natively understands PL/SQL, packages, Oracle-specific data types, DBA specific views etc.
- Oracle compatible JDBC, ODBC, .NET, OCI and Pro*C drivers for the database
- The EDB Migration Portal is a website that allows users to upload Oracle database DDL definitions, and migrate them to EPAS by leveraging EPAS native compatibility, rewrite rules, and well-defined work arounds

- The Migration Toolkit (MTK) migrates data extremely fast using parallel high-speed techniques that can leverage DBMS links and OCI
- EDB Replication Server (EPRS) is used for incremental data migration, a.k.a. Change Data Capture, which is key when migrating larger data sets (> 100 GB) without significant downtime.
- LiveCompare makes it easy to validate the consistency of a migrated data set
- Management and high availability tools that can replace Oracle Enterprise Manager, key aspects of Golden Gate, RAC, and Data Guard.

7.1 EDB Postgres Advanced Server - the heart of the native compatibility approach

EPAS introduces significant native capabilities that cover a majority of Oracle database features, and is being continuously improved via new features. Compatibility with Oracle is provided by EPAS in the following areas:

• Oracle specific and syntax compatible database object types
• Oracle specific data types
• Oracle PL/SQL support as a built-in native procedural language
• Oracle-like data dictionary views (i.e., ALL_, DBA_, USER_ views)
• Oracle-like built in PL/SQL packages

EPAS does this through in-depth additions to the Postgres core that are necessary to achieve native compatibility that **does not impact performance**. As mentioned earlier, many capabilities such as PRAGMA-AUTONOMOUS_TRANSACTION, HINT, or Resource Manager, require in-depth changes to the underlying server, as they fundamentally change the behaviour of Postgres.

The compatibility features built into EPAS significantly reduce the amount of time and effort required for a migration from Oracle. More schema, SQL, and code can run in Postgres without modification, which means that less schema, SQL, and code needs to be converted or rewritten. The compatibility features also allow DBAs and developers to work with familiar database schema and code constructs and syntax.

In addition to the compatibility features implemented in EPAS, EDB also provides the following tools that allow DBAs and other Oracle users to work in a manner familiar to them:
• EDB*Plus - a command line database client for EPAS similar to Oracle's SQL*Plus
• EDB*Loader - a command line data loading utility for EPAS similar to Oracle's SQL*Loader

These tools not only ease the transition to Postgres from Oracle by providing users with familiar interfaces, they also allow many existing database management and reporting scripts that have been built to continue to be used with little or no modifications.

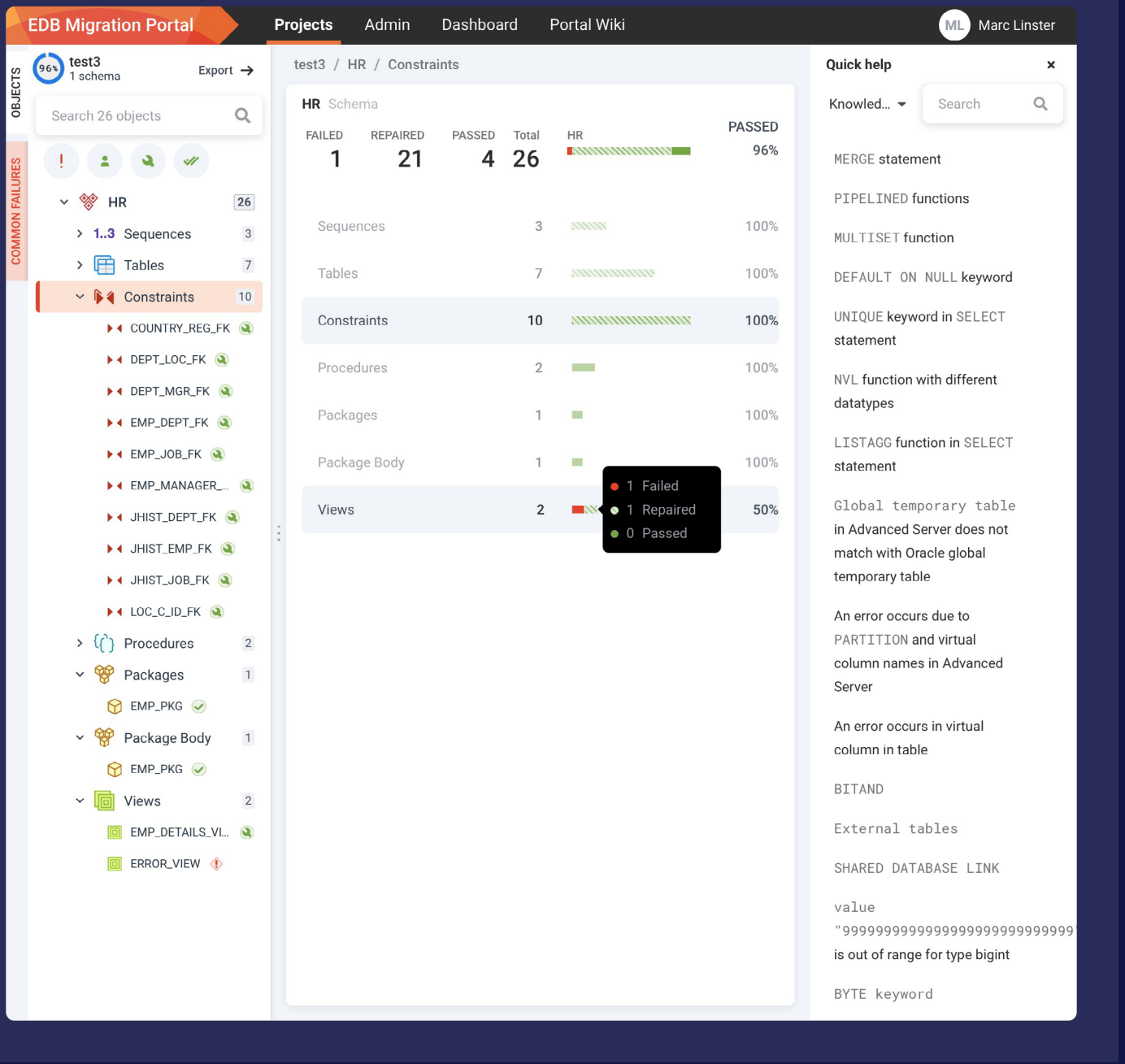
7.2 Natively compatible database drivers

Natively compatible database drivers are the second key piece in migrations that minimize the impact on applications. EDB provides Oracle database compatible JDBC, ODBC, .NET and OCI drivers

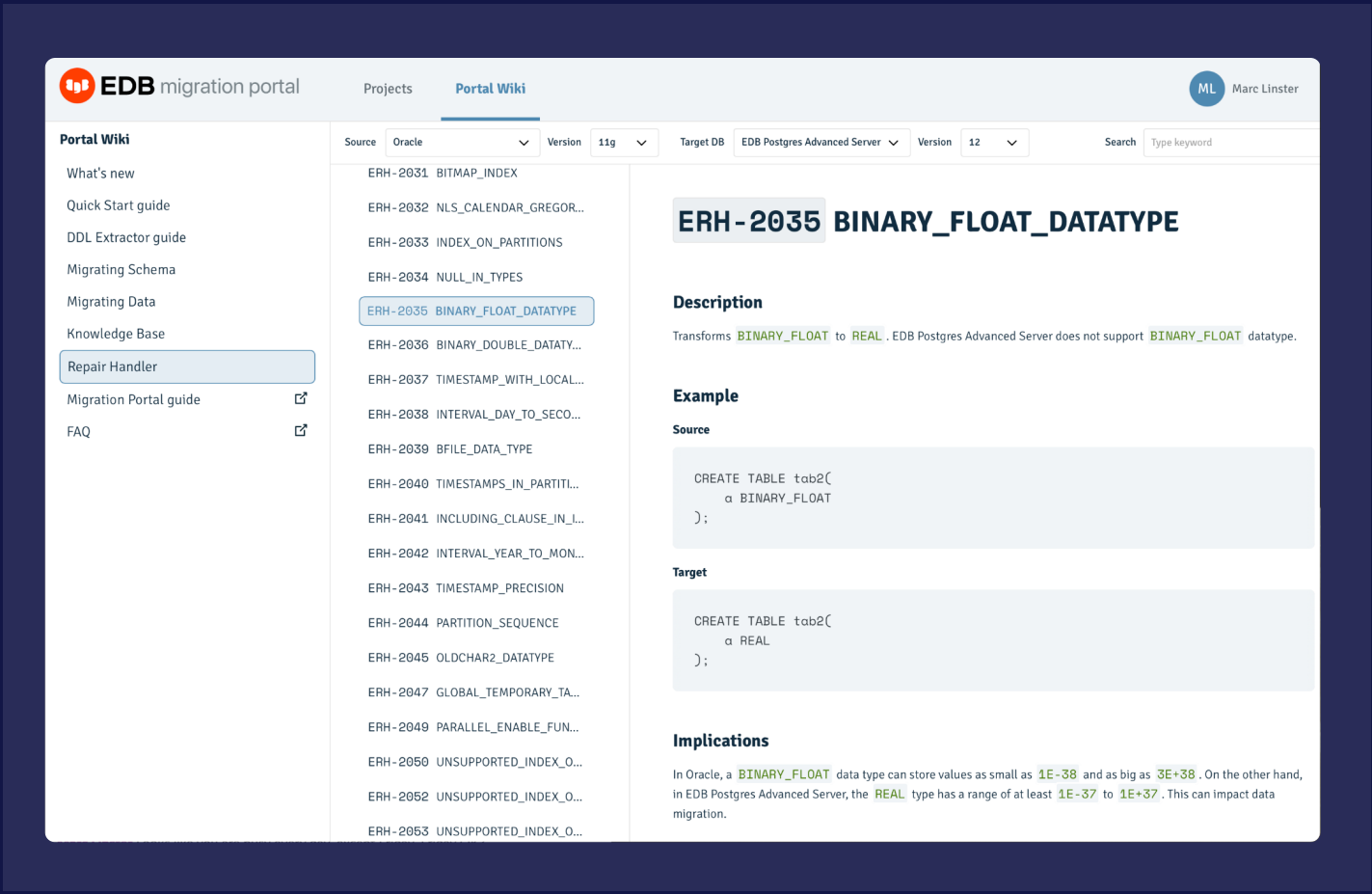
Oracle Compatibility Feature	JDBC	ODBC	.NET	OCI
PL/SQL Support	✓	✓	✓	✓
REF_CURSOR - enhanced support	✓	✓	✓	✓
User-defined Exceptions - vendor code	✓	✓		✓
Named Parameters - parameter names	✓	✓	✓	✓
Data Types- VARCHAR2 , STRUCT, ARRAYS	✓	✓	✓	✓
STRUCT - Enhanced Manipulation	✓		✓	✓
Multiple INOUT/OUT parameters	✓	✓	✓	✓

7.3 EDB Migration Portal - quick and easy migration of schemas and business logic

The EDB Migration Portal (https://migration.enterprisedb.com) is a graphical tool that analyzes Oracle schemas for compatibility, applies transformation rules and workarounds, and then loads the migrated schema into EPAS onprem, on IaaS, on K8s, or on EDB Cloud.



EDB Migration Portal Repair Handlers



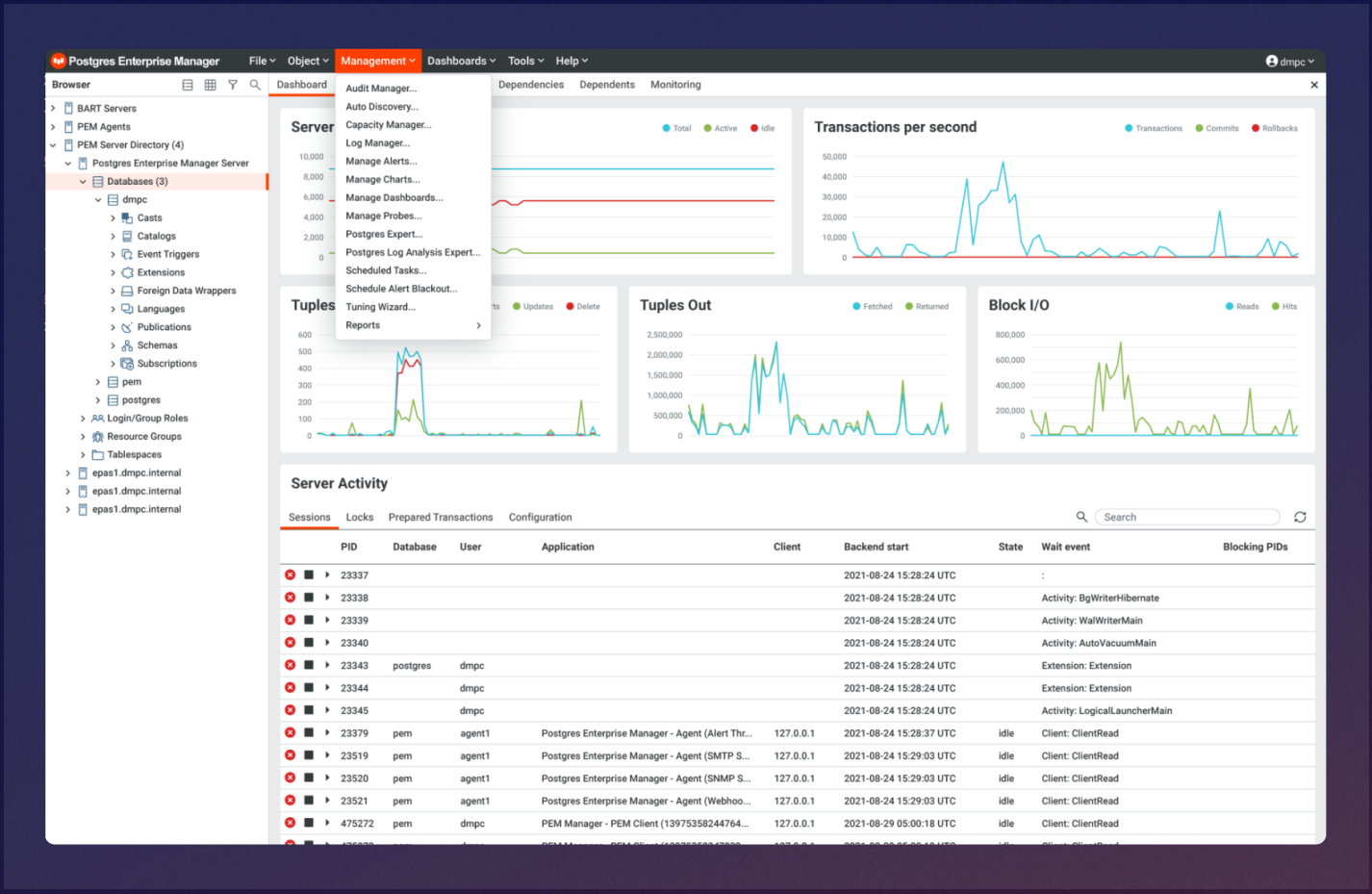
EDB Migration Portal Repair Handlers

7.4 Operational tools for management and high availability

It is important to remember that migration is not just about migrating definitions, code, and data. To make the migrated database a successful business solution, it needs to be operated with the same reliability and efficiency as the legacy solution.

EDB provides an array of management and high availability tools to make sure that non-functional requirements, such as management at scale, minimal downtime, and RPO/RTO/GRO are met effectively.

Postgres Enterprise Manager (PEM) provides a GUI-based tool for managing Postgres installation at large scale, using dashboards, alerts, and analysis tools.



EDB Migration Portal Repair Handlers

Postgres-BDR provides RAC-like high availability for Postgres up to 99.999%, using a mesh-based multi-master architecture running on commodity hardware, available in every cloud.

EDB's Failover Manager and RepMgr make it easy to operate Postgres servers with 99.99% of availability using a streaming replication approach. Barman, EDB's strategic backup and recovery tool, makes sure that customers can recover databases in case of disaster, operator error, or when meeting compliance requirements.

7.5 Experienced Professional Services Team

EDB's Professional Services team has in-depth expertise in migrating legacy databases to Postgres, and in creating the right operational environments to achieve the required high availability and performance goals. Services has a global team of subject matter experts that can support every component of migrations. Professional Services also offers expertise via the Migration Factory, which allows for expedited assessments, and rapid conversion of schema for cost and time optimized migrations. With a full range of packages, custom statement of works, and options for ongoing expertise, EDB Professional Services meets all your deployment needs.

8. Getting started

Getting started

After deciding to get off of legacy platforms, the first question is: Where do we start? Years of experience havehelped us identify key criteria to select applications that are a good starting point for a large scale migration. **Applications that meet the following criteria tend to be prime candidates to prioritize in a large scale migration project:**

- Use of an Object-Relational Mapping tool (ORM), such as Hibernate, or Spring
- Procedures, functions, packages, and triggers are written in PL/SQL, and not in Java
- While we don't expect significant application changes, migrations may require the ability to modify source code, or at least analyze it to architect a suitable workaround
- No use of RAC for scalability
- No need for Flashback
- A test harness to validate functional and non-functional requirements after the migration is generally helpful

A second tier of slightly more involved migrations are often characterized by:

- OCI interface
- Oracle Spatial and XML
- Oracle-specific extensions of .NET and ODBC that are not covered by EDB's drivers

Applications that meet these criteria should only be tackled after the migration team has gained significant experience or working in close collaboration with a migration expert, such as EDB.

- Pro*C interface
- Transaction management control inside PL/SQL (Commit/rollback/ savepoint/exceptions)
- Stored procedures written in Java
- Must have RAC scalability capabilities and Flashback

These guidelines can be used to prioritize applications during the Migration Planning phase, before actually looking at the code and running the schema through the EDB Migration Portal.

EDB's experience shows that approximately 50% of all databases fall into the first category, and can be migrated easily, usually in 10-20 person days, including data transfer and verification. About 25-30% fall into the second category.

9. Summary

Summary

Legacy database migrations, predominantly away from Oracle, are a major concern for enterprises striving for greater agility, cost reduction, and migration to the cloud. Postgres has been the dominant target, and when it is enhanced with native Oracle database compatibility, migrating becomes quick and easy. Other migration techniques that rely on manual transformations or on-the-fly translation approaches tend to involve a lot more risk, and are significantly more labor intensive.

EDB provides a proven methodology and a complete set of migration and operations tools to get you to Postgres quickly, and to make sure that you create a working and reliable business solution.

Download EDB Postgres Advanced Server, or experience native Oracle database compatibility though BigAnimal, EDB's managed database as a service. Use your own examples to evaluate how well EDB's Migration Portal and EDB's Migration Toolkit help you migrate schema, data and business logic.

About EDB

EDB provides enterprise-class software and services that enable businesses and governments to harness the full power of Postgres, the world's leading open source database. With offices worldwide, EDB serves more than 1,500 customers, including leading financial services, government, media and communications and information technology organizations. As one of the leading contributors to the vibrant and fast-growing Postgres community, EDB is committed to driving technology innovation. With deep database expertise, EDB ensures high availability, reliability, security, 24x7 global support and advanced professional services, both on premises and in the cloud. This empowers enterprises to control risk, manage costs and scale efficiently. For more information, visit **www.enterprisedb.com**.

Replacing Oracle with Postgres: How To Successfully Migrate Your Legacy Databases

EnterpriseDB Corporation
34 Crosby Drive
Suite 201
Bedford, MA 01730