



Oracle® to Postgres®: A technical migration playbook

What runs as written, where the documented limits actually are, and how to scope an Oracle migration before committing to a cutover date.

What this playbook covers

This is written for the people who will run the migration: DBAs, migration architects, and the application owners whose code talks to Oracle. It answers three questions in order: what EDB officially supports, what runs without a rewrite on EDB Postgres AI (EDB PG AI) and where the documented limits are.

It spends more space on limits than most vendor documents do. That is deliberate. On an Oracle estate the migration estimate is the decision, and an estimate built on an incomplete picture of the edges is the thing that turns a nine-month project into an eighteen-month one.

A note on timelines

This playbook publishes no Oracle migration timeline. Several figures are in circulation internally and they do not reconcile with one another. Rather than pick one, the honest position is that timeline follows from the procedural inventory and the downtime tolerance, both of which come out of an assessment.

Ask for the assessment; treat any number offered before it as a sales figure rather than an engineering one.

What EDB officially supports

COMPONENT	WHAT IT DOES	ORACLE SCOPE
EDB Migration Toolkit	Schema and data migration into PostgreSQL or EDB PG AI	Oracle 11g, 12c, 18c and 19c as source. Targets PostgreSQL and EDB PG AI, versions 14–18.
EDB Migration Portal	Web-based schema assessment and conversion	Oracle is its only source platform: Oracle 11g, 12c, 18c and 19c into EDB PG AI with Oracle compatibility, versions 14–18.
EDB PG AI Oracle compatibility	The Oracle-compatible target	Delivered by Enterprise Postgres (Oracle Compatible), a distinct server build running in Oracle compatibility mode, not an extension on community PostgreSQL.
EDB Replication Server	Ongoing synchronization for near-zero-downtime cutover	Oracle 11g R2 to 19c. Documented single-master Oracle to PostgreSQL, Oracle to EDB PG AI and EDB PG AI to Oracle configurations.

Two platform limits to establish on day one

Oracle Real Application Clusters (RAC) and Exadata are not tested with and not officially supported by either Migration Toolkit or Replication Server. If either is in your estate, that is a scoping conversation to have at the start rather than a surprise in stage three. Source versions stop at 19c: there is no documented support for 21c, 23ai or 26ai. Both facts are in EDB's own documentation; neither is a reason not to migrate, but both change a project plan.



Construct-by-construct: what runs as written

Ordered by how much the choice of target matters.

ORACLE CONSTRUCT	EDB PG AI EQUIVALENT	WHAT TO EXPECT
PL/SQL packages	SPL packages	The one that decides the project. Community PostgreSQL has no package concept, so on that target every package becomes a redesign. EDB PG AI's Oracle compatibility supports them structurally.
Exception handling	SPL exception semantics	Semantically aligned, not hand-adjusted. Edge cases still need review; the shape of the code survives.
Cursor logic	SPL cursors	Carries across. Cursor-heavy procedural code is a performance review target, not a correctness one.
Data dictionary views	Oracle-compatible catalog views	Admin scripts and tooling that query the Oracle data dictionary largely keep working; confirm coverage for the views they read. Routinely the most underestimated dependency in an estate.
MERGE, Oracle SQL syntax, built-in packages	Native in Oracle compatibility mode	Supported in Oracle compatibility mode. Built-in packages are a documented set: check the ones your code calls against it.
Applications written against OCI	EDB OCL Connector	Open Client Library for Oracle Call Interface (OCI) applications: the path that avoids rewriting the data-access layer.
SQL*Plus scripts, SQL*Loader jobs	EDB*Plus, EDB*Loader	Operational tooling, not just the database. EDB*Plus commands are SQL*Plus-compatible; EDB*Loader supports a subset of SQL*Loader parameters, so review control files before relying on them.
Obfuscated PL/SQL (Oracle wrap)	EDB*Wrap on the target	Wrapped source cannot be assessed; see section 04.

Compatibility surfaces per EDB's Oracle-compatibility documentation. Coverage per object varies with how much Oracle-specific behavior the code depends on; confirm against a schema assessment of your own estate.



Where the edges are

Every item below is documented by EDB, in the Migration Portal and DDL Extractor notes (the driver ceiling is a Migration Toolkit note). You will meet them during assessment. Pricing them now is cheaper than discovering them in remediation.

Objects that cannot be assessed

- **Wrapped objects.** Anything obfuscated with Oracle wrap must be made available in clear text before extraction. If the original source is gone, that object is a rewrite, not a conversion: find out early.
- **Silently removed during assessment.** Materialized-view backing objects, queues, nested tables, XMLType tables, types depending on SYSTEM schemas, indexes backing primary-key and unique constraints, and unsupported system grants. EDB's documentation notes these are not flagged as removed in the assessment output, so reconcile the object count yourself rather than trusting the delta.

Partial coverage

- **System privileges.** Only CREATE DATABASE LINK, CREATE PUBLIC DATABASE LINK, DROP PUBLIC DATABASE LINK and EXEMPT ACCESS POLICY are supported.
- **ALTER statements.** Only ALTER PROFILE DEFAULT, ALTER TABLE and ALTER TRIGGER are processed.
- **Profiles.** Only Oracle's password-related limits are supported. Resource limits are extracted but not migrated.
- **Passwords.** Not migrated. Users arrive without them, so plan the credential reset as a cutover task with its own security sign-off.

Case sensitivity, and why it costs more than it should

Choosing Migration Portal's Oracle default case option, which preserves Oracle's uppercase naming, can lower the compatibility ratio, with three documented failure patterns: unquoted references to quoted-uppercase objects, %TYPE case mismatches, and TYPE-referenced column types.

Decide this before assessment rather than after, because it depresses the headline number in a way that looks like a compatibility problem and is actually a configuration one.



Extraction-time issues

- **Invalid objects.** The DDL Extractor will not extract objects in an INVALID state. Compile or exclude them first.
- **Flashback and reserved prefixes.** BIN\$ recycle-bin objects and PG_-prefixed schemas are not extracted.
- **Oracle 11g roles.** Roles and role grants cannot be extracted from 11g.
- **Busy systems.** Extraction on a heavily loaded source can hit ORA-01555 snapshot too old. Schedule accordingly, or raise UNDO_RETENTION.
- **Driver ceiling.** With ojdbc7.jar, migrations of more than about 2.1 billion records (Integer.MAX_VALUE) can finish incomplete. Use ojdbc8.jar.

Sources: EDB Migration Portal documentation, Known issues, limitations, and notes; EDB Migration Toolkit documentation, Supported platforms and databases (enterprisedb.com/docs, October 2026).

Where ora2pg fits

ora2pg is mature, free, community-maintained, and carries years of accumulated edge-case handling. If you are evaluating an Oracle migration, you already know about it. Here is the honest comparison.

What the converter decides

ora2pg converts Oracle into community PostgreSQL. That target has no packages, no Oracle-compatible data dictionary views, and different exception and cursor semantics. The conversion itself succeeds; the procedural tier still lands as hand-rewriting work.

What the target decides

EDB PG AI changes the destination, not the converter. With its Oracle compatibility, packages, exception handling, cursors, dictionary views and OCI-linked applications have somewhere compatible to land, which is what moves that work from rewriting to review.



The case for the other answer

If your estate is thin on PL/SQL (a few hundred lines of procedural code, no packages, no OCI-linked applications, no tooling that reads the Oracle data dictionary), then community PostgreSQL with ora2pg is very likely the right answer, and a compatibility layer is a cost you do not need.

The procedural inventory in stage one tells you which situation you are in. That is the number to get first.

The five stages

The same five stages, in the same order, wherever code and schema have to be converted.

STAGE	WHAT HAPPENS	EXIT CRITERION
1 • Assess	Schema and complexity analysis; procedural inventory; identification of RAC/Exadata dependencies, wrapped objects and the case-handling choice.	A scoped object inventory with effort classification.
2 • Transform	Schema and procedural conversion into EDB PG AI. Migration Portal for schema assessment and conversion; Migration Toolkit for schema and data.	Converted schema that builds cleanly on the target.
3 • Remediate	Resolution of the edge cases conversion leaves behind: semantics, performance characteristics, application-level expectations.	No unresolved conversion exceptions.
4 • Move	Bulk transfer, then EDB Replication Server for ongoing synchronization until cutover.	Reconciled row counts and checksums.
5 • Validate	Integrity, functional correctness and performance verification under representative load.	Signed-off test evidence, including regulatory validation where in scope.

Where the effort actually sits

Tooling compresses stage 2 far more than it compresses stage 3. On a package-heavy Oracle estate, remediation is where estimates go wrong most often, and it is human engineering effort, not a tooling problem to be solved by buying a better converter. Staff accordingly.



Separating the move from the switch

EDB Replication Server supports ongoing Oracle-to-EDB PG AI and Oracle-to-PostgreSQL synchronization. That makes the data move and the cutover two separate decisions rather than one event: source and target run in parallel until the switch is a business call rather than an engineering one.

Systems that can take an outage

Bulk transfer, verify, switch. Simplest path, shortest project.

Systems that cannot

Bulk transfer, then replicate continuously, reconcile, and cut over inside a planned window with the source still live behind you.

Systems that must prove it first

Dual-run against production traffic, compare outputs, then promote. Standard practice in regulated environments, and the reason the governance runbook exists as a companion to this document.

RAC and Exadata, again

Replication Server is not tested with or officially supported on them either. EDB's documentation says it, like Migration Toolkit, may work against a single persistent node; confirm that with EDB rather than assume it. Otherwise the near-zero-downtime path in this section needs a different design. Raise it in stage one.



Where to start

1

Inventory the procedural tier

Count and classify packages, procedures, functions, triggers, and the application code paths that depend on Oracle-specific behavior. This is the number that drives every estimate that follows, and it decides whether you need a compatibility layer at all.

2

Establish the platform facts

Source version, RAC or Exadata presence, case handling, wrapped objects, OCI-linked applications. Five answers, and together they determine whether the standard path applies.

3

Decide your downtime tolerance

Agree it with the business before scoping. It selects the cutover design in section 07.

4

Run a schema assessment

Scope against your real estate rather than a planning range. Reconcile the object count against the silently removed classes in section 04.

Request an assessment

Go to enterprisedb.com and request a migration assessment, or contact your EDB account team. If you received this document from an EDB representative, reply to them directly: they can schedule the assessment review in step 4.

enterprisedb.com

Sources

- EDB Migration Toolkit documentation: supported platforms and databases; object migration support (enterprisedb.com/docs/migration_toolkit/latest).
- EDB Migration Portal documentation: supported platforms; known issues, limitations, and notes, including the EDB DDL Extractor (enterprisedb.com/docs/migration_portal/latest).
- Enterprise Postgres (Oracle Compatible) and EDB OCL Connector documentation: Oracle compatibility (SPL, packages, catalog views, MERGE, built-in packages, EDB*Plus, EDB*Loader, DRITA, EDB*Wrap) and Open Client Library (enterprisedb.com/docs/epas/latest, enterprisedb.com/docs/ocl_connector/latest).
- EDB Replication Server documentation: certified versions and permitted Oracle configurations, single-master only (enterprisedb.com/docs/eprs/latest).
- Oracle press release, October 14, 2025: Oracle AI Database 26ai replaces Oracle Database 23ai (oracle.com/news).



Get the procedural inventory first.

Versions and platform limits in this playbook reflect EDB product documentation as of October 2026. Verify them before committing to a project plan.

This playbook is engineering planning guidance and does not constitute a contractual commitment, a performance guarantee, or legal, audit, or compliance advice. Supported versions and platform limitations are per EDB product documentation at time of writing; verify against current documentation before making a purchasing decision. Oracle, Oracle Real Application Clusters, Exadata, SQL*Plus, and SQL*Loader are trademarks or registered trademarks of Oracle Corporation and/or its affiliates. Postgres and PostgreSQL are registered trademarks of the PostgreSQL Community Association of Canada and used with their permission. EDB Postgres® AI is a trademark of EnterpriseDB Corporation. © EnterpriseDB Corporation 2026. All rights reserved.

EDB Postgres® AI is the first open, enterprise-grade sovereign data and AI platform, with a secure, compliant, and fully scalable environment, on premises and across clouds. Supported by a global partner network, EDB Postgres AI unifies transactional, analytical, and AI workloads, enabling organizations to operationalize their data and LLMs where, when, and how they need them. For more information, visit enterprisedb.com.

