



Migrating from Oracle®: A migration governance runbook

The stage gates, evidence, and decision rights a regulated Oracle migration has to produce, so the program can be approved, audited, and, if necessary, reversed.

The plan is not the hard part

Most Oracle migrations fail their first governance review not because the technical plan is weak, but because nobody can say who approved what, what the rollback criteria were, or what evidence exists that the migrated system produces the same answers as the one it replaced.

This document is the companion to the technical playbook. The playbook covers what converts. This covers what you have to be able to show — at each stage, to whom, and in what form — for the program to survive a change advisory board, an internal audit, or an external examiner.

What this document is not

There is no regulatory deadline forcing an Oracle migration, and this runbook does not pretend otherwise. Oracle's current Database releases are supported and actively patched. Nothing here is a compliance finding.

This is about governing a large change to a system of record — which is a control problem whatever the source database and whatever the reason for moving.

In this runbook

02 · Roles and decision rights	3
03 · Stage gates	4
04 · Evidence	5
05 · Rollback	6
06 · Validation	7
07 · After cutover	7
08 · Next steps and sources	8

Who decides what

The most common governance gap on a database migration is an implicit assumption that the team doing the work also owns the decision to go live. Separate them explicitly, in writing, before stage one.

ROLE	OWNS	DOES NOT OWN
Migration lead	Technical execution, stage exit recommendations, the issue log.	The go/no-go decision. The rollback trigger.
Application owner	Functional acceptance for their applications; the test cases that constitute "correct."	Infrastructure decisions; the cutover window.
Data owner / steward	Reconciliation acceptance; sign-off that the migrated data is the data.	Performance acceptance.
Change authority (CAB or equivalent)	The go/no-go at each stage gate and at cutover.	Technical design. Test design.
Internal audit / risk	Evidence adequacy — whether what was produced would satisfy an examiner.	Whether the migration proceeds.
Security	Credential handling at cutover, access-control parity, encryption parity.	Functional acceptance.

The segregation that matters most

Whoever performs the migration must not be the person who signs off that it succeeded. On a system of record this is usually already policy; on a database migration it is routinely waived because only the migration team understands the detail. Resist that. If the reviewer needs help reading the evidence, fix the evidence.



What gets approved, and when

Five stages, five gates. Nothing proceeds to the next stage without the named evidence and a recorded decision.

Assess	Transform	Remediate	Move	Validate
Gate 1	Gate 2	Gate 3	Gate 4	Gate 5 · cutover go/no-go
GATE	APPROVES	EVIDENCE REQUIRED	DECISION BY	
G1 — after Assess	Proceeding to conversion, and the effort estimate the budget rests on.	Object inventory with effort classification; platform-facts register (source version, RAC/Exadata, case mode, wrapped objects); reconciliation of assessment output against source object counts.	Change authority	
G2 — after Transform	That the converted schema is a fit basis for remediation.	Converted schema building cleanly on target; conversion exception log; list of objects deferred to remediation.	Migration lead recommends; change authority notes	
G3 — after Remediate	That no unresolved conversion exceptions remain.	Closed exception log with disposition per item; code review records for hand-remediated objects.	Change authority	
G4 — after Move	That the migrated data is the data.	Row-count reconciliation per table; checksum or hash comparison; replication lag record; exception report with dispositions.	Data owner	
G5 — cutover go/no-go	Going live.	Signed functional acceptance per application; performance test evidence under representative load; rollback test evidence; security sign-off on credentials and access parity.	Change authority, on recommendation from application and data owners	

Gate structure maps to the five-stage methodology in the technical playbook: Assess, Transform, Remediate, Move, Validate.



What to keep, and why

The test of an evidence pack is not whether it is thorough. It is whether someone who was not there can reconstruct what happened and why, 18 months later, without asking anyone.

Produced once, at the start

- **Platform-facts register.** Source version, RAC/Exadata presence, case mode, wrapped-object inventory, Oracle Call Interface (OCI)-linked applications. Dated and signed. This is what the estimate was based on; if any of it turns out to be wrong, the estimate is void and that should be visible.
- **Scope statement.** Which schemas, which applications, which environments — and explicitly what is out of scope.
- **Rollback design.** See section 05. Designed before Gate 2, not discovered at cutover.

Produced per stage

- **Conversion exception log.** Every object that did not convert cleanly, with disposition: hand-remediated, redesigned, deferred, or accepted as-is with justification. This is the single most useful artifact in the pack.
- **Reconciliation record.** Row counts and checksums per table, source versus target, with the timestamp of each comparison and the replication lag at that moment.
- **Test evidence.** Which cases, run by whom, against which build, with results. Functional and performance separately — functional parity does not imply performance parity, and when the concurrency implementation changes underneath the code, as it does on a move to Postgres®, that is a real risk rather than a theoretical one.
- **Decision record per gate.** Who approved, on what date, on the basis of which evidence, with any conditions.

Produced at cutover

- **Go/no-go minute.** Attendees, criteria assessed, decision, conditions, and the named person who holds the rollback trigger during the window.
- **Credential-reset record.** Migrated users arrive without their passwords. Migration Portal creates them with none, and Migration Toolkit substitutes a placeholder password for database links unless you supply the real ones. Record who reset what, when, under whose authority, and how the new credentials were distributed.
- **Cutover timeline.** Actual times against planned, including any deviation and who authorized it.



Deciding to go back

A rollback plan that has never been tested is a document, not a control. Two things distinguish a real one: the criteria are quantified in advance, and one named person can invoke it without convening a meeting.

Criteria, set before the window opens

Define the specific, measurable conditions that trigger rollback — reconciliation failure above a stated threshold, a functional defect of stated severity, performance below a stated floor under production load, or elapsed time beyond the window. Quantified in advance, because nobody makes a good judgment call at 4 a.m. with an audience.

Mechanism, tested before the window opens

Whatever the design — source left live and untouched, reverse replication, restore from a verified backup — it must have been exercised end to end at least once in a non-production environment, with evidence. Record how long the tested rollback actually took; that duration is an input to the cutover window, not a footnote.

Authority, named before the window opens

One person holds the trigger, with a named deputy. They can invoke rollback unilaterally within the window. Escalation is for extending the window, not for reversing it.

The decision that gets skipped

Agree in advance what happens to data written to the new system between cutover and rollback. On a transactional system of record, this is the hardest part of any rollback and the part most plans omit. If the answer is “we would replay it,” someone has to have written down how.



Proving the answers are the same

Data reconciliation

Row counts per table are the floor, not the standard. Add checksums or hashes over business-key columns, and reconcile at a point where replication lag is known and recorded. Reconciling against a moving source without noting lag produces a number nobody can defend later.

Performance

Test under representative load, not synthetic load, and not functional load. Oracle and Postgres both use multiversion concurrency, but they implement it differently: Oracle rebuilds older row versions from undo, while Postgres keeps them in the table until vacuum reclaims them, and locking and isolation behavior differ in the details. Code written around the source platform's behavior can be functionally correct and operationally unacceptable. This is the most common post-cutover surprise and the one most likely to trigger a rollback that was otherwise avoidable.

Functional correctness

Application owners define what "correct" means, in test cases, before Gate 2 — not after conversion, when the cases will unconsciously be written around what the converted system does. Pay particular attention to NULL and empty-string handling, which differ between the platforms, and to anything depending on transaction isolation behavior.

Regulatory validation

Where the system supports financial reporting, patient data, cardholder data, or anything else with a named control framework, the relevant validation is an input to Gate 5 — not a post-cutover activity. Confirm with your compliance function which artifacts they need and in what form, at Gate 1, while changing the plan is still cheap.

The first 30 days

- **Hypercare, with an end date.** Elevated monitoring and a named response path, for a defined period with defined exit criteria. Open-ended hypercare becomes normal operations by accident and nobody ever declares the project finished.
- **Keep the rollback path alive, and say for how long.** Decide at Gate 5 how long the source stays available and what it costs to keep it. Decommissioning early to save license cost is a decision with a risk attached; make it deliberately.
- **Watch the plan regressions.** Query plans that were fine in test can degrade as statistics evolve under real load. Baseline the important queries at cutover so you can tell degradation from a bad day.
- **Close the evidence pack.** Final reconciliation, actual-versus-planned timeline, defect log with dispositions, and a decision record for anything accepted rather than fixed. Close it while people still remember; an evidence pack assembled six months later is an exercise in reconstruction.



Where to start

1

Confirm the control framework

Ask your compliance and internal audit functions which control framework governs the target system and what evidence they will want. Do this before scoping — it changes what stages produce, not just what gets filed.

2

Name the decision rights

Fill in section 02 with real names. If two of those roles are the same person, that is the first thing to resolve.

3

Design rollback before you convert

Section 05. Designing it after Gate 2 means designing it around what you have already built.

4

Bring the assessment to a migration architect

The technical assessment and this governance structure are inputs to the same conversation. Review them together.

Request an assessment

Request a migration assessment, or contact your EDB account team. If you received this document from an EDB representative, reply to them directly.

enterprisedb.com/contact

Sources

- Stage structure maps to the five-stage migration methodology described in “Oracle to Postgres: A technical migration playbook” (EDB, 2026).
- EDB Migration Toolkit documentation: supported platforms and databases (Oracle 11g, 12c, 18c, and 19c sources; not tested with or officially supported on Oracle RAC and Exadata) and command options (-useOraCase; database link passwords) (enterprisedb.com/docs/migration_toolkit/latest).
- EDB Migration Portal documentation: supported platforms (Oracle 11g, 12c, 18c, and 19c) and known issues, limitations, and notes (wrapped objects can't be assessed; user passwords aren't migrated; Oracle default case) (enterprisedb.com/docs/migration_portal/latest).
- EDB Replication Server documentation, introduction: Oracle RAC and Oracle Exadata aren't supported (enterprisedb.com/docs/eprs/latest/01_introduction).
- Oracle Lifetime Support Policy: Oracle Technology Products, effective October 2, 2026; Oracle Database Concepts 19c, Data Concurrency and Consistency; PostgreSQL documentation, Concurrency Control and Routine Vacuuming.



Name the decision rights before stage one.

Your control framework, regulator, and internal audit function determine what is actually required. This runbook gives you the structure to show it.

This runbook is provided for internal planning purposes and does not constitute legal, audit, or compliance advice. EDB is not a law firm, an auditor, or a Qualified Security Assessor. Control-framework expectations vary by jurisdiction, industry, and organization; nothing here substitutes for your own compliance, audit, and legal advice. Consult your own compliance, legal, and audit advisors to determine what your control framework requires. Oracle, Oracle Real Application Clusters, and Exadata are registered trademarks of Oracle Corporation. 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.

