ORACLE TO POSTGRESQL · MIGRATION ASSESSMENT

Demonstration Estate (Oracle 11g)

Schema **RECON_TEST** · Oracle 11.2.0.2.0 · extracted 2026-08-23 12:39:06
Prepared by DevCrafter · 2026-08-23 · engine: pgrecon

B
MIGRATION GRADE

Executive summary

This assessment covers a PL/SQL codebase of 32 objects and 162 lines running on Oracle 11.2.0.2.0. The analysis identified 41 findings across the codebase and assigns an overall grade of B, indicating a real project with identified hot spots rather than a trivial or a highly complex migration. The severity mix is composed of 8 high, 17 medium, 12 low, and 4 informational findings, showing that while most issues are of moderate or minor consequence, a meaningful subset requires focused remediation before a PostgreSQL migration can proceed with confidence.

The high-severity findings touch on constructs that typically require the most deliberate rework in a PostgreSQL migration, including a database link, package-level state, UTL_FILE usage, a compound trigger, an autonomous transaction, a LONG or LONG RAW column, and a BFILE column. These are not evenly distributed across the codebase; a small number of objects account for a disproportionate share of findings and weighted complexity, which is a favorable condition for planning since remediation effort can be concentrated rather than spread thin across the estate.

41	8	17	108
FINDINGS	HIGH	MEDIUM	EXPECTED PERSON-DAYS

Effort range: **87–148** person-days (4.2–7.0 person-months), expected 108 — including testing and stabilization. Assumptions are listed with the estimate; this is a scoping instrument to be calibrated per engagement, not a quote.

Estate overview

32	162	5/5	11
OBJECTS	PL/SQL LINES	DDL PARSED	UNITS DEEP-PARSED

OBJECT TYPE	COUNT
TABLE	9
INDEX	6
TRIGGER	4
SEQUENCE	3
VIEW	2
PROCEDURE	2
TYPE BODY	1
TYPE	1
SYNONYM	1
PACKAGE BODY	1
PACKAGE	1
FUNCTION	1

Largest tables by rows (from optimizer statistics)

TABLE	ROWS
AUDIT_LOG	2
DEPT	2
EMP	2
INVOICES	0
LEGACY_NOTES	0
LEGACY_REFS	0
MV_DEPT_SALARIES	0
PRODUCT_SPECS	0

License exposure

What the database reports about its own edition, hardware, and use of separately licensed features. These are the source system's facts; contract interpretation and pricing belong in the engagement, next to the actual agreements.

Express	3	1
EDITION	CPU CORES	CPU_COUNT IN FORCE

No separately licensed option shows usage in the database's own statistics: the estate runs on the edition's base feature set.

Migration hot spots

The ten objects concentrating the most remediation weight; these decide the project's shape and deserve design attention first.

OBJECT	FINDINGS	WEIGHT	DRIVEN BY
RECON_TEST.PKG_LEDGER	7	12.3	Package-level state (5), Package initialization block (2), PL/SQL collection type (2), +4 more
RECON_TEST.ARCHIVE_NOTES	9	8.3	UTL_FILE usage (3), ROWID in code (1.5), DBMS_LOB usage (1.5), +6 more
RECON_TEST.TRG_EMP_AUDIT	2	5.5	Autonomous transaction (4), Transaction control inside PL/SQL (1.5)
RECON_TEST.PKG_LEDGER (body)	1	5.0	Package-level state (5)
RECON_TEST.TRG_EMP_SAL_GUARD	2	4.3	Compound trigger (4), SYSDATE (0.3)
RECON_TEST.LEGACY_NOTES.BODY	1	4.0	LONG or LONG RAW column (4)
RECON_TEST.LOOPBACK	1	3.0	Database link (3)
RECON_TEST.LEGACY_REFS.SCAN_DOC	1	3.0	BFILE column (3)
RECON_TEST.PRODUCT_SPECS.SPEC	1	3.0	XMLTYPE column (3)
RECON_TEST.T_MONEY	1	2.5	User-defined object type (2.5)

Findings and remedies

HIGH Database link R-OBJ-01 · extension: postgres_fw

> Cross-database access needs postgres_fw or oracle_fw on the PostgreSQL side, plus credential management. Every statement using the link must be found and repointed.

RECON_TEST.LOOPBACK	to //localhost:1521/XEPDB1 as RECON_TEST

HIGH Package-level state R-PKG-01

> PostgreSQL has no package variables. Scalars map to session settings via set_config and current_setting (text only); composites and collections need per-session temporary tables. Every reader and writer of the state changes.

RECON_TEST.PKG_LEDGER	2 package-level declaration(s), first at line 2
RECON_TEST.PKG_LEDGER (body)	1 package-level declaration(s), first at line 2

HIGH UTL_FILE usage R-SYS-01 · extension: orafce

> PostgreSQL functions have no supported server-side file API. Move file handling to the application, or accept the operational cost of an untrusted language such as plpython.

RECON_TEST.ARCHIVE_NOTES	UTL_FILE (first at line 16)

HIGH Compound trigger R-TRG-01

> PostgreSQL has no compound triggers. Split the timing sections into separate trigger functions and move statement-level state into a transition table or a session variable.

RECON_TEST.TRG_EMP_SAL_GUARD	on EMP

HIGH Autonomous transaction R-TRG-02

> PostgreSQL functions cannot commit independently of their caller. Rework audit-style autonomous transactions with a dblink loopback, a message table drained by a worker, or pg_background.

RECON_TEST.TRG_EMP_AUDIT	PRAGMA AUTONOMOUS_TRANSACTION (first at line 5)

HIGH LONG or LONG RAW column R-TYPE-01

> LONG has been deprecated since Oracle 8. Map LONG to text and LONG RAW to bytea; any code reading the column piecewise must be rewritten, and only one LONG per table exists to find.

RECON_TEST.LEGACY_NOTES.BODY	LONG

HIGH BFILE column R-TYPE-07

> BFILE points at a file outside the database and PostgreSQL has no counterpart. Store a path or object key plus the file in application storage, and move the DIRECTORY grants and BFILENAME plumbing into the application.

RECON_TEST.LEGACY_REFS.SCAN_DOC	BFILE

MEDIUM User-defined object type R-OBJ-05

> Composite types carry the attributes, but member functions do not exist in PostgreSQL: each method becomes a standalone function taking the composite as its first argument, and every call site changes.

RECON_TEST.T_MONEY	object type

MEDIUM Package initialization block R-PKG-02

> Oracle runs the block once per session on first reference, which PostgreSQL cannot replicate. Extract it into an init function guarded by a session flag and call it lazily from each entry point. Detection is heuristic; confirm by eye.

RECON_TEST.PKG_LEDGER	BEGIN blocks exceed subprograms by 1 (possible initialization block)

MEDIUM Dynamic SQL R-SRC-01

> EXECUTE IMMEDIATE maps to PL/pgSQL EXECUTE, but bind syntax and error handling differ. Each dynamic statement needs a manual port and a test.

RECON_TEST.PKG_LEDGER	EXECUTE IMMEDIATE (first at line 27)

MEDIUM GOTO statement R-SRC-02

PL/pgSQL has no GOTO. The control flow must be restructured with loops, exceptions, or early returns.

RECON_TEST.FIND_MANAGER	GOTO (first at line 14)

MEDIUM BULK COLLECT R-SRC-03

PL/pgSQL has no BULK COLLECT; array_agg into an array or a plain set-based statement usually replaces the whole loop.

RECON_TEST.PKG_LEDGER	BULK COLLECT (first at line 24)

MEDIUM CONNECT BY hierarchical query R-SRC-04

Rewrite as a recursive CTE (WITH RECURSIVE). LEVEL, SYS_CONNECT_BY_PATH, and ORDER SIBLINGS BY all need manual equivalents.

RECON_TEST.V_ORG_CHART	CONNECT BY in view definition

MEDIUM Transaction control inside PL/SQL R-SRC-14

PostgreSQL functions cannot COMMIT or ROLLBACK; only procedures called outside a transaction block can. Each routine that controls transactions must become a procedure or hand that control back to the caller.

RECON_TEST.TRG_EMP_AUDIT	COMMIT or ROLLBACK (first at line 15)

MEDIUM PL/SQL collection type R-SRC-16

Nested tables, varrays, and associative arrays map to PostgreSQL arrays, jsonb, or temporary tables depending on use; collection methods such as EXTEND, COUNT, and FIRST have no direct equivalents and every reader changes.

RECON_TEST.PKG_LEDGER	collection type declaration (first at line 20)

MEDIUM Empty string treated as NULL R-SRC-18

> Oracle treats '' as NULL; PostgreSQL treats it as a real value. v := '' assigns NULL in Oracle but an empty string in PostgreSQL, NVL(x, '') stops collapsing, and col = '' can start matching rows. Nothing errors: behavior changes silently. Decide per site whether NULL or empty was meant, and put the data path under test.

RECON_TEST.ARCHIVE_NOTES	empty-string literal (first at line 4)

MEDIUM ROWID in code R-SRC-19

> ctid is not a ROWID: it moves on update and vacuum and must not be stored or compared. Fetch-by-rowid loops rewrite to primary-key access or WHERE CURRENT OF; ROWID-typed variables and parameters change type with them.

RECON_TEST.ARCHIVE_NOTES	ROWID (first at line 3)

MEDIUM DBMS_LOB usage R-SYS-04

> Piecewise LOB calls mostly disappear: text and bytea are handled whole, with substring and length built in. Streaming access beyond 1 GB needs large objects and the lo_ API.

RECON_TEST.ARCHIVE_NOTES	DBMS_LOB (first at line 14)

MEDIUM Global temporary table R-TAB-01 · extension: pgtt

> PostgreSQL temporary tables are per session and vanish with it; there is no shared definition with private rows. Rework to on-the-fly temp tables or use the pgtt extension.

RECON_TEST.STAGING_ROWS	GLOBAL TEMPORARY

MEDIUM Sequence-assigned key via trigger R-TRG-03

> The pre-12c idiom of filling a key from a sequence in a BEFORE INSERT trigger becomes a plain identity column or a DEFAULT nextval() in PostgreSQL; the trigger disappears.

RECON_TEST.AUDIT_LOG_BI	NEXTVAL in trigger (first at line 5)
RECON_TEST.INVOICES_BI	NEXTVAL in trigger (first at line 5)

MEDIUM XMLTYPE column R-TYPE-02

PostgreSQL xml stores documents but has no XML index type and none of the XMLType member functions; XQuery support differs. Inventory how the application queries the XML before mapping.

RECON_TEST.PRODUCT_SPECS.SPEC	XMLTYPE

MEDIUM TIMESTAMP WITH LOCAL TIME ZONE column R-TYPE-03

Values are normalized to the session time zone on read, which timestamptz does not replicate exactly. Verify every reader assumes UTC or one fixed zone before mapping to timestamptz.

RECON_TEST.INVOICES.CREATED_AT	TIMESTAMP(6) WITH LOCAL TIME ZONE

MEDIUM ROWID or UROWID column R-TYPE-06

A stored ROWID is a physical row address, and PostgreSQL has no stable one: ctid changes on update and vacuum. Replace the column with a real key reference; whatever wrote and followed these addresses changes with it.

RECON_TEST.LEGACY_REFS.REF_ADDR	ROWID

LOW Synonym R-OBJ-06

PostgreSQL has no synonyms. Same-database synonyms usually dissolve into search_path; cross-schema renames become views or schema qualification at the call sites.

RECON_TEST.STAFF	for RECON_TEST.EMP

LOW Optimizer hint R-PERF-01 · extension: pg_hint_plan

PostgreSQL ignores optimizer hints; pg_hint_plan can pin plans but is a last resort. A hinted statement was tuned by hand against Oracle's planner, so budget a fresh look at its plan and indexes on PostgreSQL instead of a plain rewrite.

RECON_TEST.ARCHIVE_NOTES	optimizer hint (first at line 8)

LOW Old-style outer join (+) R-SRC-05

> The (+) operator does not exist in PostgreSQL. Rewrite as ANSI LEFT or RIGHT JOIN; semantics match except for a few multi-condition corner cases worth testing.

RECON_TEST.V_EMP_DEPT	uses (+) outer join

LOW SYSDATE R-SRC-11

> SYSDATE is evaluated at call time; now() is fixed at transaction start. Map to statement_timestamp() or clock_timestamp() where the difference matters, for example inside long transactions and loops.

RECON_TEST.ARCHIVE_NOTES	SYSDATE (first at line 19)
RECON_TEST.PKG_LEDGER	SYSDATE (first at line 7)
RECON_TEST.TRG_EMP_SAL_GUARD	SYSDATE (first at line 8)

LOW DBMS_OUTPUT usage R-SYS-05 · extension: orafce

> Map PUT_LINE to RAISE NOTICE for diagnostics. If callers consume the output programmatically, that pattern needs a real interface instead.

RECON_TEST.ARCHIVE_NOTES	DBMS_OUTPUT (first at line 19)

LOW NUMBER without precision R-TYPE-04

> Bare NUMBER accepts up to 38 significant digits. Map to numeric, and check whether application code assumes integer behavior that a typed bigint would express better.

RECON_TEST.AUDIT_LOG.AUDIT_ID	NUMBER with no precision
RECON_TEST.INVOICES.INVOICE_ID	NUMBER with no precision
RECON_TEST.MV_DEPT_SALARIES.HEADCOUNT	NUMBER with no precision
RECON_TEST.MV_DEPT_SALARIES.TOTAL_SAL	NUMBER with no precision
RECON_TEST.V_ORG_CHART.DEPTH	NUMBER with no precision

INFO Invalid object R-OBJ-08

> Already broken before any migration. Decide per object: fix it, or exclude it from scope in writing so the estimate does not pay for dead code.

RECON_TEST.ARCHIVE_NOTES	status INVALID
RECON_TEST.PKG_LEDGER	status INVALID

INFO Swallowed exception R-SRC-07

> WHEN OTHERS THEN NULL hides every failure including the ones a migration introduces. Worth removing before the port so the test phase can actually see errors.

RECON_TEST.ARCHIVE_NOTES	WHEN OTHERS THEN NULL (first at line 22)

INFO Table without optimizer statistics R-STAT-01

> num_rows is null, so data-volume estimates for this table are unreliable. Gather statistics before trusting sizing numbers.

RECON_TEST.STAGING_ROWS	no statistics gathered

Risk outlook

The overall risk profile is manageable but not negligible. The presence of database links, autonomous transactions, a compound trigger, and Oracle-specific data types such as LONG RAW and BFILE indicates areas where behavioral parity with PostgreSQL will require careful redesign rather than mechanical translation. The medium-severity findings, which form the largest single category, include dynamic SQL, transaction control inside PL/SQL, sequence-assigned keys via triggers, and several Oracle-specific data types, all of which are well-understood migration patterns but collectively add to the overall effort. Low and informational findings, including NUMBER columns without precision and instances of SYSDATE usage, are lower risk but should be addressed for long-term maintainability and consistency.

Recommended approach

Given the severity mix and the concentration of findings in a small number of hot spot objects, a phased approach is recommended. The first phase should target the highest-weighted objects, which

carry the largest share of findings and complexity, resolving the high-severity constructs such as the database link, autonomous transaction, compound trigger, and Oracle-specific data types in those locations first. Subsequent phases can address the remaining medium-severity patterns across the broader codebase, followed by cleanup of low and informational items. This sequencing aligns remediation effort with the estimated range of 87.4 to 147.8 person-days, with an expected effort of 107.5 person-days, and allows early phases to validate the approach on the most consequential objects before extending it to the rest of the estate.

Effort estimate

COMPONENT	PERSON-DAYS
baseline and environment	5.0
schema conversion	1.6
finding remediation	59.8
PL/SQL porting by volume	0.8
data movement	0.0
Development subtotal	**67.2**

With testing and stabilization: **87 / 108 / 148** person-days (low / expected / high).

Assumptions

- calibration profile 'default': the public engine's conservative defaults, unchanged
- stored code ports at 200 lines per person-day including unit tests
- repeated findings of one rule cost a severity-dependent fraction of the first fix
- testing and stabilization runs 1.3x to 2.2x development, 1.6x expected
- static estimate from schema facts alone: workload, data quality, and application coupling are not visible here

Appendix: findings detail

Each rule lists up to twelve objects here so repeated findings stay readable at estate scale; the findings CSV delivered with this report carries every object with its evidence.

RULE	SEVERITY	OBJECT	EVIDENCE
`R-OBJ-01`	high	RECON_TEST.LOOPBACK	to //localhost:1521/XEPDB1 as RECON_TEST

R-PKG-01	high	RECON_TEST.PKG_LEDGER	2 package-level declaration(s), first at line 2
R-PKG-01	high	RECON_TEST.PKG_LEDGER (body)	1 package-level declaration(s), first at line 2
R-SYS-01	high	RECON_TEST.ARCHIVE_NOTES	UTL_FILE (first at line 16)
R-TRG-01	high	RECON_TEST.TRG_EMP_SAL_GUARD	on EMP
R-TRG-02	high	RECON_TEST.TRG_EMP_AUDIT	PRAGMA AUTONOMOUS_TRANSACTION (first at line 5)
R-TYPE-01	high	RECON_TEST.LEGACY_NOTES.BODY	LONG
R-TYPE-07	high	RECON_TEST.LEGACY_REFS.SCAN_DOC	BFILE
R-OBJ-05	medium	RECON_TEST.T_MONEY	object type
R-PKG-02	medium	RECON_TEST.PKG_LEDGER	BEGIN blocks exceed subprograms by 1 (possible initialization block)
R-SRC-01	medium	RECON_TEST.PKG_LEDGER	EXECUTE IMMEDIATE (first at line 27)
R-SRC-02	medium	RECON_TEST.FIND_MANAGER	GOTO (first at line 14)
R-SRC-03	medium	RECON_TEST.PKG_LEDGER	BULK COLLECT (first at line 24)
R-SRC-04	medium	RECON_TEST.V_ORG_CHART	CONNECT BY in view definition
R-SRC-14	medium	RECON_TEST.TRG_EMP_AUDIT	COMMIT or ROLLBACK (first at line 15)
R-SRC-16	medium	RECON_TEST.PKG_LEDGER	collection type declaration (first at line 20)
R-SRC-18	medium	RECON_TEST.ARCHIVE_NOTES	empty-string literal (first at line 4)
R-SRC-19	medium	RECON_TEST.ARCHIVE_NOTES	ROWID (first at line 3)
R-SYS-04	medium	RECON_TEST.ARCHIVE_NOTES	DBMS_LOB (first at line 14)
R-TAB-01	medium	RECON_TEST.STAGING_ROWS	GLOBAL TEMPORARY
R-TRG-03	medium	RECON_TEST.AUDIT_LOG_BI	NEXTVAL in trigger (first at line 5)
R-TRG-03	medium	RECON_TEST.INVOICES_BI	NEXTVAL in trigger (first at line 5)
R-TYPE-02	medium	RECON_TEST.PRODUCT_SPECS.SPEC	XMLTYPE
R-TYPE-03	medium	RECON_TEST.INVOICES.CREATED_AT	TIMESTAMP(6) WITH LOCAL TIME ZONE
R-TYPE-06	medium	RECON_TEST.LEGACY_REFS.REF_ADDR	ROWID

R-OBJ-06	low	RECON_TEST.STAFF	for RECON_TEST.EMP
R-PERF-01	low	RECON_TEST.ARCHIVE_NOTES	optimizer hint (first at line 8)
R-SRC-05	low	RECON_TEST.V_EMP_DEPT	uses (+) outer join
R-SRC-11	low	RECON_TEST.ARCHIVE_NOTES	SYSDATE (first at line 19)
R-SRC-11	low	RECON_TEST.PKG_LEDGER	SYSDATE (first at line 7)
R-SRC-11	low	RECON_TEST.TRG_EMP_SAL_GUARD	SYSDATE (first at line 8)
R-SYS-05	low	RECON_TEST.ARCHIVE_NOTES	DBMS_OUTPUT (first at line 19)
R-TYPE-04	low	RECON_TEST.AUDIT_LOG.AUDIT_ID	NUMBER with no precision
R-TYPE-04	low	RECON_TEST.INVOICES.INVOICE_ID	NUMBER with no precision
R-TYPE-04	low	RECON_TEST.MV_DEPT_SALARIES.HEADCOUNT	NUMBER with no precision
R-TYPE-04	low	RECON_TEST.MV_DEPT_SALARIES.TOTAL_SAL	NUMBER with no precision
R-TYPE-04	low	RECON_TEST.V_ORG_CHART.DEPTH	NUMBER with no precision
R-OBJ-08	info	RECON_TEST.ARCHIVE_NOTES	status INVALID
R-OBJ-08	info	RECON_TEST.PKG_LEDGER	status INVALID
R-SRC-07	info	RECON_TEST.ARCHIVE_NOTES	WHEN OTHERS THEN NULL (first at line 22)
R-STAT-01	info	RECON_TEST.STAGING_ROWS	no statistics gathered

Produced with the pgrecon assessment engine from an offline extraction; no connection was made to the source database. Findings are deterministic and reproducible from the same inventory. DevCrafter · devcrafterai.com