

ANOVA CLOUD



# Database Modernization Discovery

A structured discovery instrument for scoping SQL Server → PostgreSQL programs — estate, workloads, licensing, applications, operations, and constraints.

Discovery questionnaire · Working document

Complete with your AnovaCloud architect during the discovery engagement.

Bring evidence, not opinions — every rating should point to an artifact, owner, or system.

## How to use this questionnaire

- Work through this with the people who own the answers — architects, data engineers, security, and business sponsors. One respondent per section rarely knows enough.
- Answer for **today's reality**, not the roadmap. Roadmaps go in the notes, clearly labeled.
- Where a question doesn't apply, mark N/A and say why — a wrong answer is worse than a blank one.
- Your AnovaCloud architect will challenge ratings in the workshop. Evidence wins arguments.

Rate each statement 1–5 where a scale is shown. **1** = not true / absent. **5** = fully true / mature and evidenced. Use the notes lines to cite the artifact, system, or person behind the rating — ratings without evidence are revisited in the workshop.

## A Estate inventory

List every in-scope database server. One row per instance — attach a spreadsheet for large estates.

- A.1** For each instance, record: hostname, SQL Server version & edition, database count, total data size, workload type (OLTP/DW/mixed), and business criticality.

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

- A.2** Instances past end-of-support (or approaching it within 18 months) are identified.

☐ Yes ☐ No ☐ N/A ☐ Unknown

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

- A.3** License position is documented: core counts, Software Assurance status, renewal dates, true-up history.

☐ Yes ☐ No ☐ N/A ☐ Unknown

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

- A.4** Virtualization / cloud placement of each instance is known (on-prem, Azure VM, RDS, etc.).

☐ Yes ☐ No ☐ N/A ☐ Unknown

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

- A.5** Orphaned or unknown databases (no owner, no documented purpose) are flagged separately.

☐ Yes ☐ No ☐ N/A ☐ Unknown

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

## B Workload profiling

What the databases actually do — measured, not assumed.

**B.1** Peak and average workload (transactions/sec, batch windows) is measured for tier-1 databases.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**B.2** Procedural code volume is inventoried: stored procedures, functions, triggers, CLR assemblies per database.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**B.3** Problematic features are flagged: linked servers, replication, Service Broker, SSIS/SSRS dependencies, filestream.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**B.4** Query performance baselines exist (top queries by CPU/IO, captured before any move).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**B.5** Maintenance windows and downtime tolerance are documented per application.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**B.6** Seasonality and growth projections (data volume, user growth) are available.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**B.7** Cross-database dependencies (cross-db queries, distributed transactions) are mapped.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**B.8** Non-production environment topology (dev/test/UAT/DR) mirrors production closely enough to validate against.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

## C Licensing & cost

The economic case lives or dies here.

**C.1** Current annual SQL Server spend (licenses, SA, support) is quantified.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**C.2** Renewal / true-up dates and the cost of the next renewal are known.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**C.3** Target PostgreSQL operating cost is modeled (managed service or self-hosted, per wave).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**C.4** Dual-running costs during migration are budgeted (source + target in parallel).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

**C.5** Conversion engineering effort is estimated per complexity tier — not as a single lump sum.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_

**C.6** The business case has an owner who will defend it through the program.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_

## D Application coupling

Applications decide migration difficulty more than databases do.

**D.1** Every application connecting to each database is inventoried with its owner.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_

**D.2** Connection methods are known: drivers, ORMs, connection strings, service accounts.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_

**D.3** SQL dialect dependencies (T-SQL specifics, proprietary functions) are assessed per application.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_

**D.4** SSIS packages, SSRS reports, and SQL Agent jobs are inventoried with replacement plans started.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_

**D.5** Third-party / vendor applications with SQL Server dependencies are identified; vendor support for PostgreSQL is confirmed.

■ Yes ■ No ■ N/A ■ Unknown

Evidence / notes: \_\_\_\_\_

**D.6** Applications can be changed on the migration timeline (release trains accommodate remediation).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
 \_\_\_\_\_

**D.7** Test coverage exists to prove application parity after migration (automated or well-defined manual).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
 \_\_\_\_\_

## E HA, DR & operations

PostgreSQL operations must be designed, not discovered.

**E.1** Current HA/DR topology (Always On, mirroring, log shipping) and RPO/RTO are documented.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
 \_\_\_\_\_

**E.2** Target PostgreSQL HA design is drafted (streaming replication, failover, managed-service equivalents).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
 \_\_\_\_\_

**E.3** Backup, point-in-time recovery, and restore testing have defined owners and schedules.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
 \_\_\_\_\_

**E.4** Monitoring and alerting can extend to PostgreSQL (metrics, logs, slow-query analysis).

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
 \_\_\_\_\_

**E.5** DBA / operations staff have (or will get) PostgreSQL training during the program.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
 \_\_\_\_\_

**E.6** Patching, minor-version upgrades, and major-version upgrade paths are planned.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_**E.7** Security baselines (encryption, auditing, access reviews) are defined for the target estate.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

## **F** Migration constraints

The non-technical facts that shape the plan.

**F.1** Cutover windows and blackout periods (fiscal close, peak season) are calendared.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_**F.2** Data residency and sovereignty requirements for the target platform are confirmed.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_**F.3** Rollback expectations are agreed: how far back, how fast, who decides.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_**F.4** Stakeholders accept a wave-based program (months) over a big-bang cutover.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_**F.5** Program governance exists: wave entry/exit criteria, named wave owners, escalation path.

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

---

**F.6** Knowledge transfer is funded — your team operates PostgreSQL afterward, not us forever.

---

■ 1 ■ 2 ■ 3 ■ 4 ■ 5 (1 = not true / absent → 5 = fully true / mature)

---

Evidence / notes: \_\_\_\_\_  
\_\_\_\_\_

---

## Complexity scoring

Score each in-scope database 1–5 on the six factors. **Average  $\leq 2.0$ :** early waves — prove the factory. **2.1–3.4:** middle waves with extra validation.  **$\geq 3.5$ :** late waves; consider simplification or re-platforming before migration.

| Complexity factor   | Low (1)                     | Medium (3)                   | High (5)                           |
|---------------------|-----------------------------|------------------------------|------------------------------------|
| Procedural code     | < 50 simple procs           | 50–500 procs, some complex   | > 500 procs / heavy CLR            |
| SQL Server features | Engine + Agent only         | Replication / linked servers | Service Broker / CLR / filestream  |
| App coupling        | 1–2 modern apps, good tests | Several apps, partial tests  | Many apps / vendor apps / no tests |
| SSIS / SSRS / jobs  | None or trivial             | Moderate, replaceable        | Heavy, business-critical           |
| HA/DR needs         | Standard RPO/RTO            | Tight RTO, multi-site        | Zero-downtime, active-active       |
| Downtime tolerance  | Weekend windows OK          | Short planned windows        | Near-zero required                 |

### Next steps

- Score each database 1–5 on the six factors above; average  $\geq 3.5$  goes late in the wave plan,  $\leq 2.0$  goes first. Your architect validates scoring in the assessment.
- Typical next step: a fixed-scope **Database Migration Assessment** — automated inventory, complexity scoring, wave plan, and TCO model.
- Bring this completed questionnaire plus a database inventory export to your discovery call.  
**[anovacloud.com/contact/](https://anovacloud.com/contact/)**

**AnovaCloud** — Engineering the AI-Native Enterprise. Advisory · Engineering · Agentic AI · Modernization · Academy · Workforce. Frisco, Texas. **Talk to an Architect:** [anovacloud.com/contact/](https://anovacloud.com/contact/)