



# Database Modernizer Assessment

Source database wordpress

Job docs-sample · Generated 2026-10-07

# Assessment Summary

Source database wordpress · Architecture HYBRID WITH CACHE · Overall risk HIGH

107

query patterns analyzed

3

migration waves

8

source tables migrate

HIGH

overall risk, 2 items

## EXECUTIVE SUMMARY

Analyzed 50 source tables and 107 query patterns across 3 target database(s). Workload split: DynamoDB handles 58 queries (54.2%), Aurora MySQL handles 49 queries (45.8%). Cache layer: ElastiCache fronts 20 hot reads (83.4% of calls) cache-aside and owns no queries. 2 open migration risks (overall: HIGH); 6 more were resolved by the assignment. Other targets evaluated: DocumentDB (consolidated into Aurora MySQL by the reality check), OpenSearch (consolidated into Aurora MySQL by the reality check).

| Engine       | Role             | Workload    | Scope           |
|--------------|------------------|-------------|-----------------|
| DynamoDB     | Migration target | 54.2%       | 0 source tables |
| Aurora MySQL | Migration target | 45.8%       | 0 source tables |
| ElastiCache  | Cache layer      | 83.4% calls | cache-aside     |

8 source tables move to a purpose-built engine; ElastiCache caches 20 hot reads (83.4% of calls) and owns none of the workload. Full detail in the Decision and Engineering reports.



# Workload Coverage and Engine Assignment

107 production query patterns analyzed, 107 in scope



- DynamoDB 54.2% · 58 queries · 97% confidence
- Aurora MySQL 45.8% · 49 queries · 88% confidence
- ElastiCache cache layer · 20 cached reads · 83.4% of calls

## Basis of the analysis

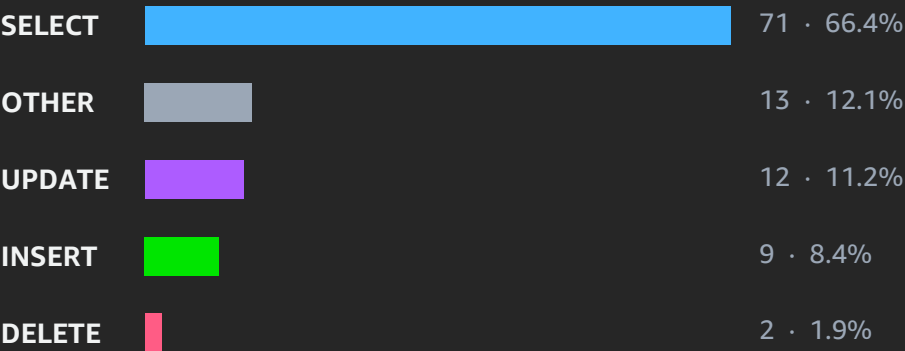
- 107 of 107 query patterns in scope
- 0 tables mapped to a named target, each with its own confidence score
- 1 co-dependency group — table sets that must move together

Pattern detection, scoring and assignment run with no language model. This deck is rendered from the stored results of that run.

# Query Workload Composition

35% key-value lookups · 66% reads · 76% touch a single table

## OPERATION MIX (107 patterns)



### Table fan-out

81 queries touch a single table (75.7%). Only 11 touch three or more (10.3%), and just 2 reach 4.

## WHAT THOSE QUERIES ARE DOING · WHOLE WORKLOAD

| Pattern                                           | Queries | Share |
|---------------------------------------------------|---------|-------|
| Key-value lookups (PK read, ≤5 rows)              | 37      | 34.6% |
| Low-frequency writes (<5 cps)                     | 22      | 20.6% |
| Aggregations (SUM/COUNT/AVG + GROUP BY)           | 21      | 19.6% |
| Low-frequency reads (admin, reporting, health...) | 21      | 19.6% |
| Metadata / config store                           | 18      | 16.8% |
| Complex joins (3+ tables)                         | 13      | 12.1% |
| High frequency reads                              | 6       | 5.6%  |
| Full-text search (LIKE, MATCH, tsvector)          | 5       | 4.7%  |
| Status filters (WHERE status=?, EXISTS, IS NULL)  | 4       | 3.7%  |

### Throughput distribution

222.0 queries/sec across the whole database, 88.9% reads. The top 10 patterns carry 74.4% of throughput; the busiest single query runs at 75.8 rps.



# Engine Confidence and Open Decisions

3 recommended engines · lowest confidence ElastiCache at 76%

| Engine       | Workload       | Confidence | HIGH risks | Role             |
|--------------|----------------|------------|------------|------------------|
| DynamoDB     | 54.2%          | 97%        | 2          | Migration target |
| Aurora MySQL | 45.8%          | 88%        | 0          | Migration target |
| ElastiCache  | 83.4%<br>calls | 76%        | 0          | Cache layer      |

Confidence is the mean fit of the queries routed to each engine (for the cache layer, of the reads it fronts), not a forecast. Steps that need no data migration are reversible at any confidence: the engine already serving the data keeps serving it, so there is nothing to roll back. Migration targets under 50% are sequenced last so they can be re-scoped once the earlier waves have produced real measurements.

**2 of the 2 HIGH risks sit on DynamoDB alone.**

## Scope DynamoDB? **2 of 2 HIGH risks**

Carries the most HIGH risks · 97% confidence · 54.2% of workload

**Scope defined as the matching table subset.**

## Add the no-migration step? **83% of calls cached**

ElastiCache · no data migration · 0 HIGH risks

**No data migration: the engine already serving these reads keeps the data, so the step is reversible.**



# Risk Profile

2 risks: 2 HIGH

HIGH

2

BY TYPE

1 migration complexity · 1 performance degradation

| #        | Engine   | What it is                                                        | Mitigation                                                         | Queries |
|----------|----------|-------------------------------------------------------------------|--------------------------------------------------------------------|---------|
| RISK-001 | DynamoDB | Complex GROUP BY / HAVING / multi-table aggregations. DynamoDB... | Pre-compute aggregates on write (DynamoDB Streams + Lambda), or... | 1       |
| RISK-002 | DynamoDB | Schema design gap: 9 queries assigned to DynamoDB have no...      | Add in-scope access patterns for these queries to the DynamoDB...  | 9       |

Each HIGH risk carries an affected-query count and a documented mitigation. The full mitigation of every risk is in the Engineering Report.

# Migration Sequencing

Wave 1 moves all 107 queries to Aurora at cutover; 45.8% remain there at the end

## WAVE 1

### Aurora MySQL

100% at cutover

88% confidence

0 HIGH risks

The whole source database (MySQL 8.0.45) moves 1:1 to Aurora MySQL: same engine family. The collector reports...

**Gate: Schema and data parity validated against the source database before cutover; decommission...**

## WAVE 2

### ElastiCache

83.4% calls cached

76% confidence

0 HIGH risks

20 hot reads (83.4% of calls), cache-aside in front of Aurora MySQL in wave 2, then DynamoDB once wave 3...

**Gate: Cache hit rate and invalidation verified against Aurora MySQL.**

## WAVE 3

### DynamoDB

54.2% query  
patterns

97% confidence

2 HIGH risks

58 key-value and point-lookup queries (54.2% of query patterns) across 1 table group, respecting co-dependent...

**Gate: Dual-write/backfill validated and query parity confirmed per table group. 7 tables...**

## Sequencing constraints

1 co-dependency group constrains wave boundaries; 4 trade-offs are documented in the Engineering Report. Later waves are re-scoped from wave 1's measurements. One suggested adoption path, not the only one — to modernize in one step instead, adopt the target architecture directly.

