AS OF 22 SEPTEMBER 2026

PostgreSQL as DepositMatch's system of record

DepositMatch needs a system of record for its reconciliation workspace: each client firm's imports and source rows, the invoices and deposits they become, match suggestions, reviewer decisions, exceptions, and the audit log behind all of it. The goals are a review log a firm can explain at month-end, exceptions that stay owned, and a pilot with one store to operate. The store must commit an import and its derived rows atomically under constraints, record who decided what and keep the history, scope access by role to the column, encrypt data in transit and at rest, recover to a point in time within fifteen minutes, run the worker's job queue, and take a read replica later. We evaluated PostgreSQL 16 against those seven capabilities on the public record. It meets all seven, six from its own documentation and encryption at rest from the Amazon RDS hosting the architecture names. Restore time and replica lag under an import burst wait on a drill and a test.

1. Required capabilities

- Each capability comes from a project document, named in the last column

CAPABILITY	SERVES	COMES FROM
An import and its derived rows commit together or not at all, under constraints	a review log a firm can explain	the decision record and the import-integrity standard
Every decision and correction is recorded with who and when, and history is kept	auditability as usability	the auditability standard and the principles
Access scoped by role down to the column	protecting financial records	the financial-data standard
Encryption in transit by the software, at rest by the hosting	records unreadable if traffic or storage leaks	the financial-data standard and the architecture document
Point-in-time recovery within fifteen minutes	the disaster-recovery target	the architecture document

CAPABILITY	SERVES	COMES FROM
A job queue inside the store for several workers	one store to operate	the architecture document and the decision record
A read replica later, without a schema change	reports that do not slow imports	the decision record

Sources [2]

2. What it is

- An open-source object-relational database, ACID-compliant since 2001, governed by a community with no controlling vendor, and the most-used database in the 2025 Stack Overflow Developer Survey

object-relational, transactional
SQL, ACID, constraints, extensions

community governed
committers and a core team, no controlling vendor

most-used database, 2025 survey
55.6% of respondents

one major release a year
five years of support each, version 16 until 2028

no licence fee
use, copy, modify, distribute

Sources [3]

3. Capability alignment

- Two documented cautions bear on role scoping: superusers always bypass row security and owners do unless the table forces them, and a tenant set with plain SET persists across a pooled connection where SET LOCAL resets it
- Encryption at rest and the five-minute log interval come from the Amazon RDS hosting the architecture names; every other row is the software's own documentation
- Outside the public record: certifications and running cost belong to the hosting, cross-server sharding is not native, and raw CSV rows fit the core JSONB type

CAPABILITY	WHAT THE RECORD SHOWS	STATUS
Atomic commit and constraints	All-or-nothing transactions, six constraint types including foreign keys	Met
Attribution and history	Trigger functions writing user, time, and the changed row to an audit table	Met
Role scope to the column	Column-level privileges and row security since 9.5, SET LOCAL behind a pool	Met
Encryption	TLS from the software, AES-256 at rest from the hosting	Met
Point-in-time recovery	Continuous archiving, hosting uploads logs every 5 minutes	Met
In-store job queue	SELECT FOR UPDATE SKIP LOCKED for queue tables	Met
Read replica later	Hot standby, read-only and eventually consistent, replicas with a lag metric on the hosting	Met

Sources [4, 5, 6, 8]

4. Competitive landscape

- MySQL 8.4 has no native row-level security, and the documented alternative is views with WHERE clauses plus triggers; a document database supports multi-document transactions, and its own maker cautions against relying on them in place of schema design

CANDIDATE	ATOMIC COMMIT	ROLE SCOPE	AT REST	PITR	REPLICA
PostgreSQL 16	Met	Met	via hosting	via hosting	Met
MySQL 8.4 Community	InnoDB ACID model	Unmet, no native row security	not researched	not researched	not researched
Document database (MongoDB)	Transactions since 4.0	not researched	not researched	not researched	not researched
Amazon RDS for PostgreSQL 16	Inherits	Inherits	AES-256	Logs every 5 minutes	Replicas, lag metric

Sources [7]

5. Open questions

- Each answer has a home: the scheduled restore drill, the decision's review trigger, or a test

OPEN QUESTION	WHY IT MATTERS	WHERE THE ANSWER COMES FROM
Does a point-in-time restore complete within the four-hour recovery-time target?	The record carries the recovery point and no restore duration	The quarterly restore drill before paid launch
What replica lag would a reporting replica see during an import burst, should reporting load grow?	Decides whether firm-level reports can leave the primary	The decision's review trigger, then a planned test
If per-firm isolation moves into row security, does throughput hold behind the API's connection pool?	Decides whether isolation lives in the database or the application	A planned test, only if row security is adopted

Sources [9]

Sources

1. Scope and Summary: PostgreSQL 16, the system-of-record role, the accepted decision this informs; 33 sources · component-profile-postgresql#scope, component-profile-postgresql#summary, component-profile-postgresql#sources
2. Need and Required Capabilities: the goals and the seven capabilities with their origins · component-profile-postgresql#need, component-profile-postgresql#required-capabilities
3. What It Is: object-relational, ACID-compliant since 2001, community governed, 55.6% of 2025 survey respondents, one major a year, five years of support, 16 until 2028, no-fee licence · component-profile-postgresql#what-it-is
4. Capability Alignment: transactions and constraints; triggers with current_user and now(); column privileges and row security since 9.5 (superusers always bypass, owners unless forced, SET LOCAL); TLS, AES-256 via hosting; archiving, logs every 5 minutes; SKIP LOCKED; hot standby; Met on all seven; verdict Fit · component-profile-postgresql#capability-alignment
5. Other capabilities: JSONB in core; no cross-server sharding; logical replication since 10 · component-profile-postgresql#capability-alignment
6. Operations and pricing: certifications and running cost not published · component-profile-postgresql#technology-and-operations, component-profile-postgresql#pricing-and-licensing
7. Competitive Landscape: MySQL 8.4 (InnoDB ACID, no native row-level security, GPL); MongoDB (transactions since 4.0); Amazon RDS for PostgreSQL 16 (PostgreSQL 13 to 17, AES-256, logs every 5 minutes, replicas with a lag metric, Multi-AZ standby) · component-profile-postgresql#competitive-landscape
8. Confidence: Medium; design-defining claims rest on the maker alone · component-profile-postgresql#confidence
9. Open Questions: restore within the 4-hour target, replica lag under an import burst, row-security throughput behind the pool; drill, review trigger, test · component-profile-postgresql#open-questions