

PostgreSQL Enterprise Features

Michael Banck <michael.banck@credativ.de>

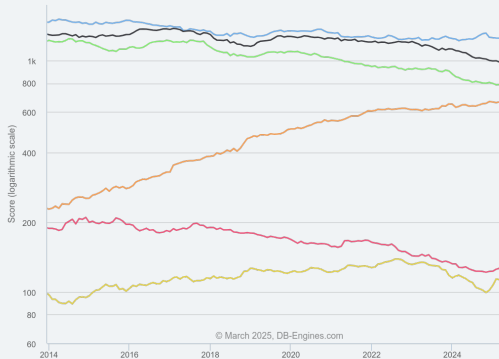
CLT2025, 22.3.2025

- Gegründet 1999 in Mönchengladbach
- Enge Verbindung zur Open-Source Community
- 40 Open-Source Spezialisten
- Beratung, Entwicklung, Schulungen, Support (3rd-Level / 24x7)
- Open-Source-Infrastruktur mit Debian, Kubernetes und Proxmox
- Open-Source-Datenbank PostgreSQL
- DevOps mit Ansible, Puppet, Terraform und andere
- Seit 2025 wieder unabhängiges, Inhaber-geführtes Unternehmen

“PostgreSQL: The World’s Most Advanced Open Source Relational Database”

Popularity Contest

DB-Engines Ranking



https://db-engines.com/en/ranking_trend/relational+dbms

Popularity Contest

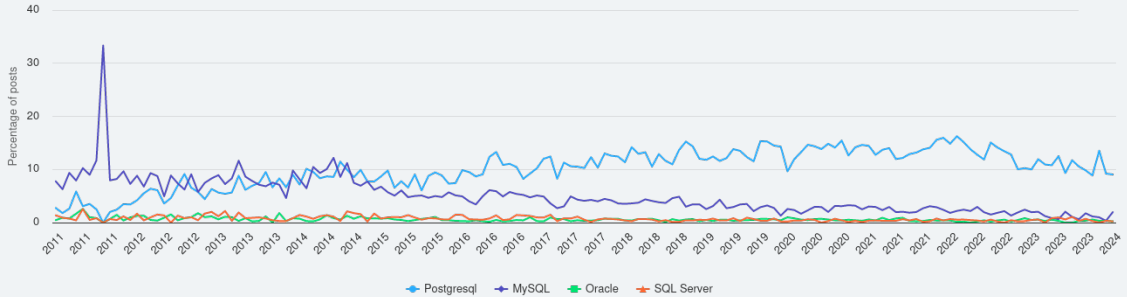
Stack Overflow 2023 Developer Survey



<https://survey.stackoverflow.co/2023/>

Popularity Contest

Hacker News Hiring Trends



Highcharts.com

<https://www.hntrends.com/>

- Erweiterbares, Objekt-Relationales Datenbanksystem
- Entstanden als Forschungsprojekt in Berkeley
- Community-basierte Entwicklung als Open-Source Projekt seit Mitte der 90er
- "Postgres Global Development Group", 7 köpfiges Core Team, ca. 30 Committer
- Hersteller-Unabhängig, kommerzieller Support von mehreren Firmen erhältlich
- Viele (auch proprietäre) Forks

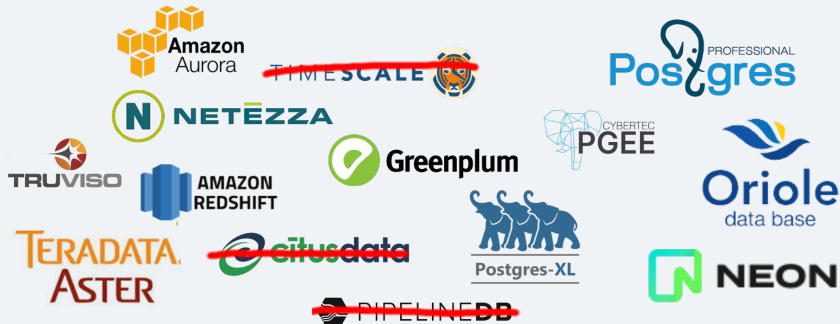
PostgreSQL

Forks



PostgreSQL

Forks



- Einige Forks sind inzwischen Erweiterungen für unmodifiziertes Postgres

PostgreSQL Wire Protokoll

Die universale Datenbank API



- Google Cloud Spanner Announcement Google Cloud Next '21:
 - "PostgreSQL has emerged as the "API" for operational databases"



Cloud
Spanner



CockroachDB



Amazon Aurora



AlloyDB for PostgreSQL



CRATE.IO



yugabyteDB



ClickHouse



QuestDB



Materialize



Yellowbrick

Postlite

<https://dbdb.io/browse?compatible=postgresql>

Top Feature-Requests 2009



- Einfache, eingebaute Replikation
- In-Place Upgrades
- Administration/Monitoring
- Treiber-Qualität/Wartung
- Erweiterungs-Management
- Locales/Collationen pro Spalte
- Materialisierte und Aktualisierbare Sichten
- Autonome Transaktionen
- Parallelisierte Queries
- Index-Only Scans
- Merge/Upsert Anweisung
- Automatische Partitionierung
- Hot Standby
- Rekursive Abfragen und Window Funktionen

Top Feature-Requests 2009 – Stand 2025



- Einfache, eingebaute Replikation
- In-Place Upgrades
- Administration/Monitoring
- Treiber-Qualität/Wartung
- Erweiterungs-Management
- Locales/Collationen pro Spalte
- Materialisierte und Aktualisierbare Sichten
- Autonome Transaktionen
- Parallelisierte Queries
- Index-Only Scans
- Merge/Upsert Anweisung
- Automatische Partitionierung
- Hot Standby
- Rekursive Abfragen und Window Funktionen

- Donald Feinberg, Gartner:
 - *“Postgres functionality has increased greatly and is now more than sufficient to run both mission-critical and non-mission-critical applications.”*
- Noel Yuhanna, Forrester:
 - *“PostgreSQL has the second-largest open source community; has competitive technology and features and continues to expand its growth across various industries.”*
 - *“Performance, integration, security, unpredictable workloads, and high availability are companies’ top data management challenges.”*

- Matt Aslett, 451 Group:
 - “PostgreSQL is a proven database for enterprise relational application workloads”
 - “Increased commercial offerings and cloud-based functionality are driving adoption”
- Erik Anderson, Lead Software Engineer at Bloomberg:
 - “At Bloomberg, we have millions of data feeds and trillions of data points dating back over 100 years. [We] have been extremely pleased with ... TimescaleDB’s capability to accommodate our workload [...] If you are looking to support large scale time-series datasets, then TimescaleDB is a good fit.”

- Planbare Major- und Patch-Releases, lange Wartungszyklen
- Fehler-Toleranz und Daten-Konsistenz
- Unternehmens-relevante Sicherheits-Features
- Interoperabilität und Erweiterbarkeit
- Replikation und Hochverfügbarkeit
- Service-Kontinuität
- Integriertes Operations, Monitoring und Backup

Enterprise Anti-Features

- Keine pro-Core oder pro-Knoten Lizenzierung, keine Lizenz-Audits

PostgreSQL Database Management System

(formerly known as Postgres, then as Postgres95)

Portions Copyright (c) 1996–2025, PostgreSQL Global Development Group

Portions Copyright (c) 1994, The Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

- `pgsql-general` Mailingliste, Slack-Channel (englisch)
- `pgsql-bugs` “Bug-Tracker”
- *“When I submitted a bug to the list, usually within an hour or two I would get an email back saying “confirmed that that’s a bug, I’m gonna look at it” and for the first three or four months, I never submitted a bug that I didn’t have a fix installed and running within 24 hours of submitting the initial post. And after my experience with Sybase and MySQL/MaxDB, it was totally amazing.”*

- PostgreSQL-Dokumentation hat einen hervorragenden Ruf
- Dokumentation steht ebenfalls unter Postgres-Lizenz und ist komplett frei
- Features werden nur mit vollständiger Dokumentation akzeptiert
- Dokumentation wird pro Major-Release gepflegt
- Keine Übersetzung ins Deutsche vorhanden

Planbare Major- und Patch-Releases Lange Wartungszyklen

Verlässliche und lange supportete Versionen

Major-Releases



- Eine Major-Version pro Jahr, Release üblicherweise in September/Oktober
- Zeit-basierter Code-Freeze (Q1), anschließende Beta-Phase
- Release erfolgt wenn keine schweren Bugs mehr vorhanden sind

Version	Release Datum	Version	Release Datum
17	26. September 2024	11	18. Oktober 2018
16	14. September 2023	10	5. Oktober 2017
15	13. Oktober 2022	9.6	29. September 2016
14	30. September 2021	9.5	7. Januar 2016
13	24. September 2020	9.4	18. Dezember 2014
12	3. Oktober 2019	9.3	9. September 2013

- Release Management Team seit 2016

Verlässliche und lange supportete Versionen

Minor/Patch/Point-Releases



- Major-Releases werden 5 Jahre lang supportet (sog. Backbranches)
 - Finales Minor-Release danach
- Vierteljährliche, planbare Point-Releases für kritische/sicherheitsrelevante Bugs
- Jeweils am zweiten Donnerstag des zweiten Monats eines Quartals
- Zeitplan: <https://www.postgresql.org/developer/roadmap/>
- Security Team behandelt Sicherheitsvorfälle
- Im Notfall außerordentliche Point-Releases
 - 17.4, 16.8, ... Release am 20. Februar 2025
 - 17.2, 16.6, ... Release am 21. November 2024
 - 14.4 Release am 16. Juni 2022
 - 10.3, 9.6.8, ... Release am 1. März 2018

- Red Hat / CentOS / Rocky (aktuell RHEL 7/8/9)
 - <http://yum.postgresql.org>
- SUSE (aktuell SLE 15)
 - <http://zypp.postgresql.org>
- Debian (aktuell 11/12) / Ubuntu (aktuell 20.04/22.04/24.04/24.10)
 - <http://apt.postgresql.org>
- Jeweils mehrere Architekturen (x86-64, arm64, teilweise ppc64el, s390x)
- Jeweils alle gängigen Erweiterungen

Fehler-Toleranz und Daten-Konsistenz

- “I manage thousands of databases (PostgreSQL, SQL Server, and MySQL), and this past weekend we had a massive power surge that knocked out two APC cabinets. [...] Long story short, every single PostgreSQL machine survived the failure with **zero** data corruption. I had a few issues with SQL Server machines, and virtually every MySQL machine has required data cleanup and table scans and tweaks to get it back to “production” status.”
- “I had exactly the same experience 3 years ago. Complete power failure (the stand-by generator took fire) in one small datacenter (around 500 machines). We had Oracle, SQL Server, DB2, MySQL, Progress, and of course PostgreSQL. The only database engine that restarted with no operation required was PostgreSQL. There were very minimal problems with Oracle (typing recover on some instances), but we had quite a few problems with the other engines.”

<http://archives.postgresql.org/pgsql-advocacy/2011-04/msg00085.php>

- Write-Ahead-Log sichert Transaktionen gegen Abstürze
 - Automatisches Nachspielen der Transaktionslogs bei Crash-Recovery
- Commit-Log speichert Status von Transaktionen
 - “Instantaneous Transaction Rollback / Accelerated Database Recovery”
- Synchrone Replikation zu Standbys / Disaster-Recovery Standort möglich
 - Optional pro Transaktion
- Daten-Checksummen sichern gegen Storage-Fehler ab
 - https://github.com/credativ/pg_checksums
- Prüfung der Index/Daten-Konsistenz (amcheck Erweiterung)
- Regressions-, Isolations und WAL-Konsistenz-Checks während Entwicklung

Unternehmens-relevante Sicherheits-Features

Unternehmens-relevante Sicherheits-Features



- Authentifizierung
 - Quell-IP/Nutzer/Datenbank basiert
 - LDAP/GSS (Active Directory)
 - SSL-Zertifikate
 - SCRAM-SHA-256
- Zugriffskontrolle innerhalb der Datenbank
 - Spaltenbasierte Grants
 - Row-Level Security (RLS)
 - SELinux Erweiterung (sepgsql)
- Logging und Auditierung
 - Postgres Protokollierung ist sehr flexibel
 - PGAudit Erweiterung für Auditierung

<https://www.postgresql.eu/events/pgconfde2023/sessions/session/4583/slides/385/mbanck-postgresql-bsi.pdf>

Unternehmens-relevante Sicherheits-Features – STIG

UNCLASSIFIED



Crunchy Data Postgres 16 Security Technical Implementation Guide

Version: 1

Release: 1

13 Jun 2024

XSL Release 1/25/2022 Sort by: STIGID

Description: This Security Technical Implementation Guide is published as a tool to improve the security of Department of Defense (DOD) information systems. The requirements are derived from the National Institute of Standards and Technology (NIST) 800-53 and related documents. Comments or proposed revisions to this document should be sent via email to the following address: disa.stig_spt@mail.mil.

https://www.crunchydata.com/files/stig/PGSQL_16_STIG_V1R1.pdf

- Revisionssichere Protokollierung via Syslog (DBAs haben nur Lesezugriff)
- Verbindungs-Logging, Logging von (langsamen) Abfragen, DDL/DML Logging
- Per-Nutzer/Datenbank Logging
- PGAudit-Erweiterung erlaubt feingranulare Auditierung

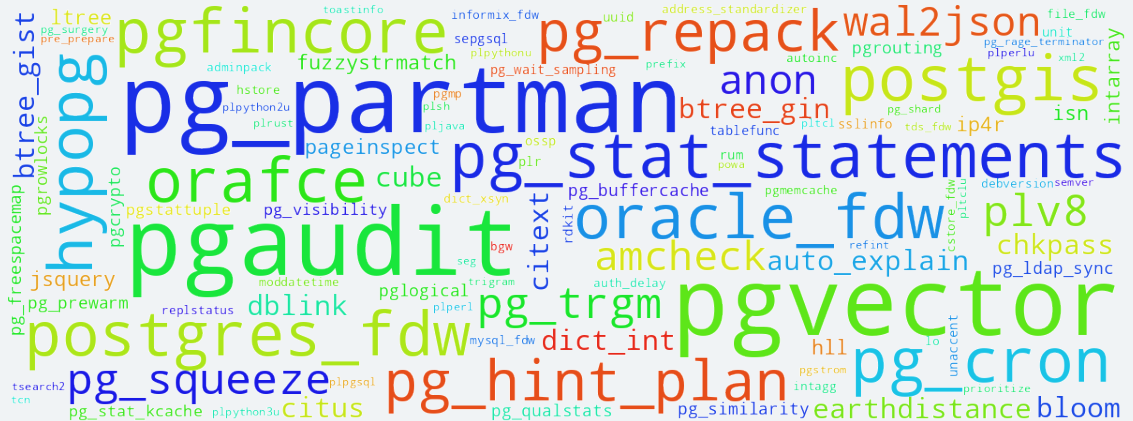
- Konfigurierbare Audit-Ereignisse werden in Logdatei geschrieben
- Per-Nutzer oder Per-Objekt Auditierung möglich
- `pgauditlogtofile` Erweiterung erlaubt separates Audit-Log

```
CREATE ROLE auditor;  
SET pgaudit.role = 'auditor';  
CREATE TABLE accounts  
(  
    id INT,  
    name TEXT,  
    password TEXT,  
    description TEXT  
);  
GRANT SELECT (password) ON public.accounts TO auditor;
```

Interoperabilität und Erweiterbarkeit

- Federation via Foreign Data Wrappers (FDW) SQL/MED-Standard
 - Transparenter (Schreib/Lese)-Zugriff auf entfernte Datenbanken/Datenquellen
 - Insbesondere zu anderen Postgres-Instanzen (`postgres_fdw`)
 - Andere SQL-Datenbanken: Oracle, MSSQL, MySQL, Informix, SQLAlchemy
 - Schema Import (`IMPORT FOREIGN SCHEMA`)
 - Join/Predicate Push-Down (`WHERE`-Bedingungen)
- Change-Data-Capture (CDC) via Logical Decoding
 - Debezium/Kafka Integration
- Logische Replikation von Teilen einer Datenbank in andere Postgres-Instanzen
- FerretDB (<https://github.com/FerretDB/FerretDB>) als MongoDB Proxy

- Reines SQL oder zusätzlich C-basierte Bibliothek
- Mächtige API und Hooks
- Große, wachsende Anzahl
 - Prozedurale Sprachen
 - Zusätzliche Daten-Typen
 - Zusätzliche Index-Typen
 - Zusätzliche Storage-Engines
 - Änderungen im Planner/Optimizer
 - Foreign-Data-Wrapper
 - Protokollierung/Auditierung
 - Administrative Hilfen



- `pgvector` - Vector Support für AI Anwendungen
- `timescale` - Zeitreihen-basierte Erweiterung für Echtzeit-Analytik
- `pgaudit` - Auditierung von Ereignissen
- `orafce` - Oracle Kompatibilität
- `postgis` - Geographische Objekte (Spatial)
- `pg_partman` - Automatisiertes Partitionierungs-Management
- `pg_search/pg_trgm` - BM25 Volltextsuche / Ähnlichkeitssuche
- `pg_hint_plan` - Optimizer-Hints
- `pg_repack/pg_squeeze` - Online Reorganisierung von Tabellen

Replikation und Hochverfügbarkeit

- Transaktionslog-Streaming an Standby
- Lesende Abfragen auf Standby möglich (Hot-Standby)
- Quorum-basierte synchrone Replikation, optional pro Transaktion
- Konsistente lesende Abfragen von synchronen Standbys
- Switchover, Switchback, Promote und Remastering
- Kaskadierende und/oder Zeitverzögerte Replikation
- Einschränkungen:
 - Erfordert gleiche Major-Version auf Primary und Standby
 - Nur gesamte Instanz replizierbar, keine einzelnen Datenbanken oder Tabellen
 - Keine Orchestrierung von Switchover innerhalb Postgres

- Schutz gegen Hardware/Software-Ausfall
 - CPU defekt
 - Ausfall Netzwerk-Karte
 - Kernel-Panik
 - Absturz des Postgres-Prozesses
- Wartung beeinträchtigt Betrieb nicht (Switchover)
 - Neustart Postgres-Prozess nach Patching oder Konfiguration-Änderung
 - Aktualisierung Betriebssystem

- Pacemaker/PostgreSQL Automatic Failover (PAF)
- Patroni
- repmgr
- pg_auto_failover
- Kubernetes Postgres-Operatoren
 - Zalando Operator
 - Crunchy Operator
 - Percona Operator
 - CloudNativePG
 - StackGres Operator
- Client-basiertes Failover durch Angabe mehrerer Hosts
 - PgJDBC
 - libpq

- Agent und Kommandozeilen-Client `patronictl`
- Konfiguriert Instanzen und Replikation (Bot-Pattern)
 - Flexibles Standby Cloning via Replikation oder Restore von Backups
 - Ermöglicht Switchover
 - Switch-Back Recovery alter Primaries via `pg_rewind`-Integration
- Verwendet einen Distributed Consensus Store (etcd, Consul, Zookeeper)
 - Raft-Algorithmus für Leader-Election
 - Vermeidung von Split-Brain, ungerade Anzahl an Knoten
 - Time-to-Live, Demote wenn Leader Lock nicht aktualisiert werden kann
 - Postgres Cluster-Status, Transaktions-Log Position
- Bietet REST-API für Status, Health-Checks und Konfigurations-Änderungen
- Deployment in Containern, Kubernetes, Bare-Metal oder via Distributions-Paketen

Deployment-Szenarien:

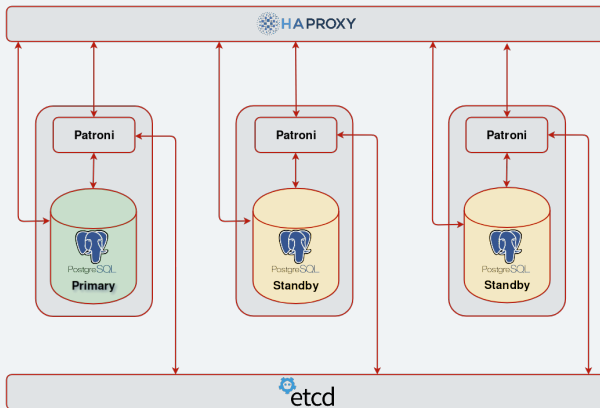
- Externer DCS-Cluster
 - 3 Patroni Knoten, jeweils mit PostgreSQL
- 3 Patroni Knoten, jeweils mit DCS
 - Davon 2 oder 3 PostgreSQL Knoten
- Standby-Cluster für Multi-DC Deployments

Optionale Features:

- Verwendung eines Watchdog-Devices
- HAProxy für Master/Replica Service-Endpunkte
 - HTTP-Check REST-API auf /primary bzw. /replica
- vip-manager für VIP Service-Endpunkte
 - Pollt DCS-Status der lokalen Node und setzt/entfernt VIP je nach Leader-Status

Hochverfügbarkeit

Patroni



<https://www.postgresql.eu/events/pgconfeu2024/sessions/session/5892/slides/544/patroni-deployment-patterns.pdf>

Service-Kontinuität

- pgBouncer: PostgreSQL Verbindungs-Proxy
 - Pooler
 - Queuing
 - Router
- Hält eingehende Verbindungen mit **PAUSE/RESUME**
 - Nach Beendigung aller Verbindungen kann Postgres neu gestartet werden
 - Anwendung sieht nur verzögerte Verbindung, keine Fehlermeldung
 - Benötigt kurzlebige Verbindungen/Transaktionen
- Kann eingehende Verbindungen je nach Konfiguration an verschiedene Postgres-Datenbanken routen
 - Switchover bei z.B. Major-Upgrades
- Ermöglicht selber On-Line Neustart mit Socket-Übernahme
- Kann Load-Balancing von lesenden Verbindungen über mehrere Standbys

Service-Kontinuität

Near-Zero-Downtime Major Upgrades



- Logische Replikation
 - Benötigt doppelte Hardware / doppelten Speicherplatz
 - Primärschlüssel auf allen Tabellen erforderlich
 - Direkter Anwendungs-Switchover
 - `pg_createsubscriber` (ab PG17) macht Read Replica zu Logischem Standby
 - <https://www.postgresql.eu/events/pgconfeu2023/sessions/session/4791/slides/439/2023.pgconf.eu%20Zero%20Downtime%20PostgreSQL%20Upgrades.pdf>
- In-Place Upgrades mit `pg_upgrade`
 - Benötigt keine Primärschlüssel aber zweites Datenverzeichnis
 - Hardlink- (ohne Switchback Möglichkeit) oder Reflink-Modus (XFS, Btrfs)
 - Downtime ab ca. 10s
 - Skaliert mit Anzahl der Datenbank-Objekte, nicht -Größe

Folgende Operationen benötigen keine langanhaltende exklusive Locks oder Table Rewrites:

- Spalten mit **NULL** oder **DEFAULT** hinzufügen
- Spalten entfernen
- Constraints entfernen oder **NOT VALID** erstellen und dann validieren
- Nebenläufig Indexe erstellen (`[CREATE INDEX|REINDEX] CONCURRENTLY`)
- Fremdschlüssel erstellen
- Unique-Constraint via nebenläufigem Index erstellen
- Tabellen-Reorganisierung mit `pg_repack/pg_squeeze`

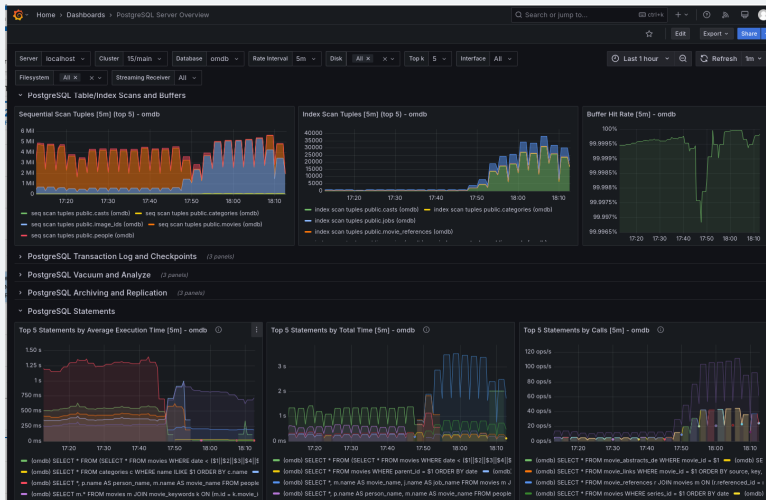
Integrated Operations, Monitoring und Backup

PostgreSQL Managed Service Provider

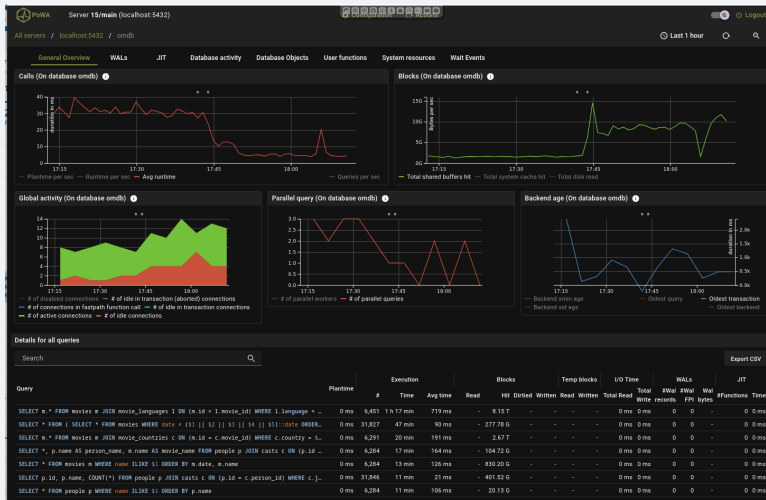


- Web-basierte Administration
 - pgAdmin4
 - DBeaver
- Hoch-Verfügbarkeit
 - Patroni
 - Pacemaker
 - pg_auto_failover
- Monitoring
 - Prometheus/Grafana
 - Icinga
- Logdatei-Analyse
 - pgBadger
- Backups
 - pgBackRest
 - Barman
 - WAL-G
- Pooling
 - PgBouncer
 - PgPool-II
 - PgCat
 - pgagroal
- Workload-Analyse
 - PoWA

Grafana Monitoring Dashboard



Postgres Workload Analyzer (PoWA)



Fehlende Enterprise-Features

Fehlende Enterprise-Features



- Multi-Master Replikation
- Transparent Data Encryption (TDE)
- Flexible Account/Passwort-Richtlinien
- Einheitliche und Integrierte Enterprise-Oberfläche

Fragen?