Intro to TimescaleDB

New Users Guide 2019

Timescale

Welcome!

Whether you are just getting started or you are interested in learning more, use this guide to kick-start your journey with TimescaleDB.

TimescaleDB Overview

TimescaleDB is optimized to provide the best possible experience when working with time-series data. By leveraging PostgreSQL's foundation, TimescaleDB provides a familiar operational experience that inherits the reliability of a database suited for mission critical and production deployments.

- 📌 Rich time-series analytics
- 📌 Massive scale and performance
- 📌 Convenience and familiarity of SQL

Products

There are three different tiers of TimescaleDB available. If you are interested in learning more about which option is best for you, contact our sales team at sales@timescale.com.

TimescaleDB

- Open Source (free): time-series SQL database optimized for fast ingest and complex queries, with automatic partitioning and efficient data retention
- Community (free): built on Open Source with advanced time-series analytics and continuous data aggregations
- Enterprise (paid): all Community capabilities plus Data Lifecycle Management (DLM) for retention policies, background analytics, and lowering TCO

Products cont.

Timescale Cloud automatically comes with Timescale Enterprise. If you are interested in learning more, contact our sales team at sales@timescale.com.

Timescale Cloud

- Hosted, fully-managed, HA instances optimized for TimescaleDB with Enterprise capabilities enabled
- Avoid cloud vendor lock-in with the choice of AWS, GCP, and Azure, plus migration and replication across clouds
- Keep your data safe with VPC peering, IP Access Control, SSL-enabled connections, and data encryption
- Transparent, pay as you go pricing model with plans starting at less than $2 / day

Architecture

TimescaleDB is implemented as an extension on PostgreSQL, which means that it runs within an overall PostgreSQL instance.

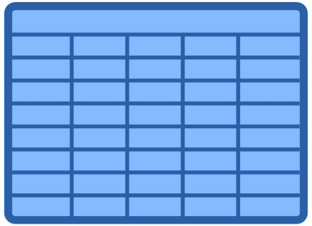
TimescaleDB leverages the high degree of customization available to extensions by adding hooks deep into PostgreSQL's query planner, data model, and execution engine.

Hypertables

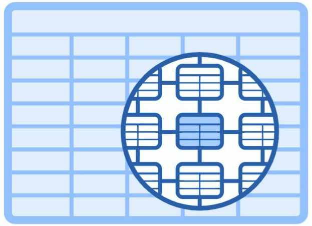
- Abstraction layer and primary point of interaction with your data used for creating tables and indexes, altering tables, inserting data, selecting data

Chunks

- Hypertable are automatically split into chunks; each chunk corresponds to a specific time interval and a region of the partition key’s space



Hypertable



Chunk

Features

TimescaleDB is designed to provide the best possible experience when working with time-series data. Two core components of the database include flexible management and a scalable architecture.

Flexible management ⚙️

- Supports all SQL operations and queries
- Compatible with existing PostgreSQL ecosystem and tooling
- Robust support for data retention policies

Scalable architecture 🏠

- Transparent time/space partitioning for both scaling up (single node) and scaling out (forthcoming)
- High data write rates, right-sized chunks, parallelized operationations

Features cont.

In addition to scaling to accommodate large data workloads, TimescaleDB provides advanced time-series analytic capabilities that allow for fast ingest and complex queries.

Time-series analytics 📈

- Extends SQL to introduce new semantics that make time-series manipulation easier (`time_bucket`, `first`, `last`)
- Additional advanced functions designed for ease-of-use and better query optimization

Fast ingest,complex queries 🔮

- Utilizes time-space partitioning for high ingest rates
- Time-based merge append optimizations that minimize the number of chunks accessed when running queries

Features cont.

Being built on PostgreSQL provides TimescaleDB with many benefits, but a core differentiator is the ability to combine time-series data with important relational metadata and rich geo data, all in the same database using SQL.

Time-series + relational 🔗

- Time-series data in hypertables can be combined with relational data in standard PostgreSQL tables via JOINs
- Avoid managing, upgrading, and implementing two separate database systems to JOIN this data in your application

Time-series + geospatial 🌐

- Leverage PostGIS for geospatial data types and queries combined with time-series analytics
- Use cases are as varied as supply-chain analytics, asset tracking, fleet management, and mapping applications

Getting Started

If you are trying TimescaleDB for the first time, or you are brand new to the time-series space, we are here to help you along the way!

Installation

Download

Explore the available installation methods

Visit the installation guide

Timescale Cloud

Get started on the cloud platform of your choice

Go to Timescale Cloud

Support information

Slack

Connect with our team and the community

Join Slack channel

Timescale docs

For tutorials, tips, and tricks on getting up and running

Visit the docs

Additional Resources

Check out our blog and follow us on Twitter or LinkedIn for the most up-to-date information.

Videos 📺

- What the Heck is Time-Series Data?
- Performant Time-Series Management and Analytics with Postgres
- Getting Started with TimescaleDB in the Cloud

Reading material 📖

- Time-series data: Why (and how) to use a relational database instead of NoSQL
- Working with a dataset: Hello NYC
- TimescaleDB vs. InfluxDB: Purpose built differently for time-series data