

Have you ever wondered how Timescale handles time-series data?

TimescaleDB introduces Hypertables, a feature that automatically partitions data based on time. While Hypertables behave like regular Postgres tables, they offer specialized functionality tailored for storing and managing timeseries data.

According to Timescale's documentation:

"With hypertables, Timescale makes it easy to improve insert and query performance by partitioning time-series data on its time parameter. Behind the scenes, the database performs the work of setting up and maintaining the hypertable's partitions. Meanwhile, you insert and query your data as if it all lives in a single, regular PostgreSQL table."

Now, the next question is: **How do we convert regular tables into hypertables?**

The process is straightforward. Let's consider a typical scenario where we have a regular table capturing stock data:

```
CREATE TABLE stocks_real_time (  
  time TIMESTAMPTZ NOT NULL,  
  symbol TEXT NOT NULL,  
  price DOUBLE PRECISION NULL,  
  day_volume INT NULL  
);
```

To convert this table into a hypertable, we use the `create_hypertable` function:

```
SELECT create_hypertable('stocks_real_time', by_range('time'));
```

In the `BY RANGE` clause, you can specify the partitioning interval. If you omit this, the default chunk interval is set to 7 days.

```
timescaledb=# SELECT h.table_name, c.interval_length  
timescaledb=# FROM _timescaledb_catalog.dimension c  
timescaledb=# JOIN _timescaledb_catalog.hypertable h  
timescaledb=# ON h.id = c.hypertable_id;  
 table_name | interval_length  
-----+-----  
stocks_real_time | 004800000000  
(1 row)
```

If you wish to define an interval during creation, you can do so using the `interval` parameter in the `BY RANGE` clause:

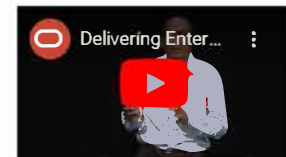
```
SELECT create_hypertable('stocks_real_time', by_range('time', INTERVAL '5 minutes'));
```

To modify the interval after creation, employ the `set_chunk_time_interval` function:

```
SELECT set_chunk_time_interval('stocks_real_time', INTERVAL '5 minutes');
```

```
timescaledb=#  
timescaledb=# SELECT h.table_name, c.interval_length  
timescaledb=# FROM _timescaledb_catalog.dimension c  
timescaledb=# JOIN _timescaledb_catalog.hypertable h  
timescaledb=# ON h.id = c.hypertable_id;  
 table_name | interval_length  
-----+-----  
stocks_real_time | 3600000000
```

FOLLOW US



(1 row)

By default, indexes are automatically generated on the time column, sorted in descending order, when creating a hypertable. However, it's advisable to create additional indexes based on the queries used.

In **conclusion**, understanding and utilizing hypertables is crucial for gaining insights into how Timescale efficiently manages time-series data.

Narasimha Rao Karanam

Narasimha Rao Karanam is a Service Delivery Manager, Database and Cloud Practice. Out of 19 Years in Industry experience, He has been with Doyensys for 17 Years supporting Oracle Database, Packaged Applications, Cloud Technologies. He handled Strategic customers and challenging environments, takes and handles new initiatives, always love to share what he learn. Connect with him at LinkedIn [[linkedin.com/in/narasimha-rao-karanam-513b5b5](https://www.linkedin.com/in/narasimha-rao-karanam-513b5b5)]

