Stocks & Crypto SQL Show #2

Analyzing NFT transactions with TimescaleDB

Attila Toth, Developer Advocate at Timescale

Timescale

Agenda

01 About the dataset

02 Optimize query speed with continuous aggregates

03 Run queries and analyze NFT transactions

04 Next steps to get started!

01

About the dataset

About the dataset

- What does it include? → Successful sale transactions from OpenSea
- How to get it: tsdb.co/nft-starter-kit
- Schema and other resources: tsdb.co/nft-starter-kit-github
- Let’s have a look inside!

02

Optimize query speed with continuous aggregates

What are continuous aggregates?

- Materialized view for time-series data
- Automatically refreshes data (refresh policies)
- Real-time aggregates: querying data from BOTH caggs and raw hypertable
- Faster long-range, aggregation queries

Create a continuous aggregate for NFT collections

```
CREATE MATERIALIZED VIEW collections_daily
WITH (timescaledb.continuous) AS
SELECT collection_id,
   time_bucket('1 day', time) AS bucket,
   AVG(total_price),
   MAX(total_price),
   MIN(total_price)
FROM nft_sales
GROUP BY collection_id, bucket;
```

Some of the current limitations:

- No window functions
- No JOINs
- Cannot create cont. aggr. on top of another
- ~~No multi-node support~~

Querying continuous aggregates

- Full SQL support over aggregates
- Query planner benefits (hypertable)
- Saves computation resources on time_bucket queries

03

Run queries and analyze NFT transactions

04

Next steps to get started!

Thank you for watching!
Next steps:

01

Learn more about TimescaleDB:

docs.timescale.com

02

Join the TimescaleDB community:

slack.timescale.com

03

Tweet us:

@TimescaleDB

@AttilaTothDev