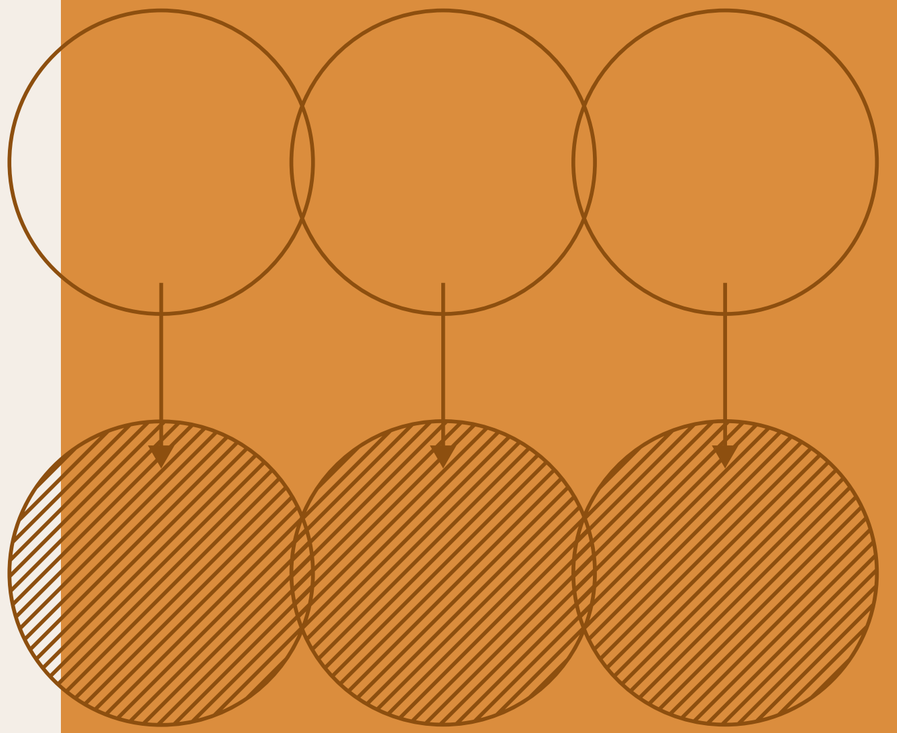
influxdb

Seminário



Wesley Silveira, Felipe Oliveira, Mateus Alexandre, Bruno Máximo

SUMÁRIO

INTRODUÇÃO AO BANCO DE DADOS

Criado pela Empresa InfluxData, o InfluxDB é um *time series Database (Banco de Dados de série temporal)*, sendo uma plataforma de base opensource que permite a coleta, armazenamento, monitoramento e visualização dos dados.

Um *time series Database* refere-se a um banco de dados criado para lidar com métricas e eventos ou medições com registro de data e hora.

https://www.influxdata.com/time-series-database/#use-cases

influxdb

- Registro temporal em segundos, milissegundos ou micro segundos;
- Organização em formato de coluna para definições de medições, conjunto de tags e campos;
- Não apresenta limitação de tags, visto que seu design é voltado para séries temporais;
- É adequado para lidar com fluxo de dados com carimbo de tempo.

https://www.influxdata.com/time-series-database/#use-cases

influxdb

TIPOS DE DADOS

A Anatomia de um Dado no InfluxDB

- Medição
- Etiquetas
- Campos
- Data/Hora

Integração com o Grafana:

Preços das ações da Apple Inc. (AAPL) e Intel Inc. (INTC)

Os Tipos de Fields(campos)

- **Float**
- **Intenger**
- **Boolean**
- **String**

Inferência de tipo

InfluxDB pode inferir o tipo de dados com base no valor inserido. Por exemplo, se você inserir "10", ele pode ser tratado como um inteiro. Se você inserir "10.5", ele pode ser tratado como um float.

COMPARAÇÃO COM O SQL

INFLUXDB

- mais read/write por segundo
- voltado para time series
- otimizado para tempo

SQL

- consultas mais complexas
- grande ecossistema
- ótimo em operações acid

INFLUXDB

MEASUREMENTS

- TIMESTAMP
- TAGS
- FIELDS

```
temperatura,cidade=Recife,sensor_id=s1 temperatura=29.5 1721556000000000000
temperatura,cidade=Recife,sensor_id=s1 temperatura=29.7 1721556060000000000
temperatura,cidade=Salvador,sensor_id=s2 temperatura=28.1 1721556000000000000
temperatura,cidade=Salvador,sensor_id=s2 temperatura=28.3 1721556060000000000
temperatura,cidade=Recife,sensor_id=s3 temperatura=30.0 1721556120000000000
```

(LINE PROTOCOL)

SQL

TABELAS

- COLUNAS
- TIPOS
- CHAVES PRIMÁRIAS

data_hora	cidade	sensor_id	temperatura
2025-07-21 10:00:00	Recife	s1	29.5
2025-07-21 10:01:00	Recife	s1	29.7
2025-07-21 10:00:00	Salvador	s2	28.1
2025-07-21 10:01:00	Salvador	s2	28.3
2025-07-21 10:02:00	Recife	s3	30.0

Comparação com o SQL
Inserção de dados

INFLUXDB

LINE PROTOCOL

```
temperatura,cidade=Recife,sensor_id=s1 temperatura=29.5 1721556000000000000
temperatura,cidade=Recife,sensor_id=s1 temperatura=29.7 1721556060000000000
temperatura,cidade=Salvador,sensor_id=s2 temperatura=28.1 1721556000000000000
temperatura,cidade=Salvador,sensor_id=s2 temperatura=28.3 1721556060000000000
temperatura,cidade=Recife,sensor_id=s3 temperatura=30.0 1721556120000000000
```

SQL

INSERT INTO

```sql
CREATE TABLE temperatura (
    id SERIAL PRIMARY KEY,
    data_hora TIMESTAMP,
    cidade VARCHAR(50),
    sensor_id VARCHAR(10),
    temperatura FLOAT
);

INSERT INTO temperatura (data_hora, cidade, sensor_id, temperatura) VALUES
('2025-07-21 10:00:00', 'Recife',   's1', 29.5),
('2025-07-21 10:01:00', 'Recife',   's1', 29.7),
('2025-07-21 10:00:00', 'Salvador', 's2', 28.1),
('2025-07-21 10:01:00', 'Salvador', 's2', 28.3),
('2025-07-21 10:02:00', 'Recife',   's3', 30.0);
```

Comparação com o SQL

Consulta com Filtro Temporal

INFLUXQL

```
SELECT MEAN(temperatura) AS media_temperatura
FROM temperatura
WHERE cidade = 'Recife'
  AND time > now() - 10m;
```

FLUX

```
from(bucket: "meu_bucket")
  |> range(start: -10m)
  |> filter(fn: (r) =>
    r._measurement == "temperatura" and
    r.cidade == "Recife" and
    r._field == "temperatura"
  )
  |> mean()
```

SQL

```
SELECT AVG(temperatura) AS media_temperatura
FROM temperatura
WHERE cidade = 'Recife'
  AND data_hora > NOW() - INTERVAL '10 minutes';
```

Comparação com o SQL
Agrupamento por Tempo

INFLUXQL

```
SELECT MEAN(temperatura) AS media_temperatura
FROM temperatura
WHERE cidade = 'Recife'
  AND time > now() - 5m
GROUP BY time(1m)
```

FLUX

```
from(bucket: "meu_bucket")
  |> range(start: -5m)
  |> filter(fn: (r) =>
    r._measurement == "temperatura" and
    r.cidade == "Recife" and
    r._field == "temperatura"
  )
  |> aggregateWindow(every: 1m, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

SQL

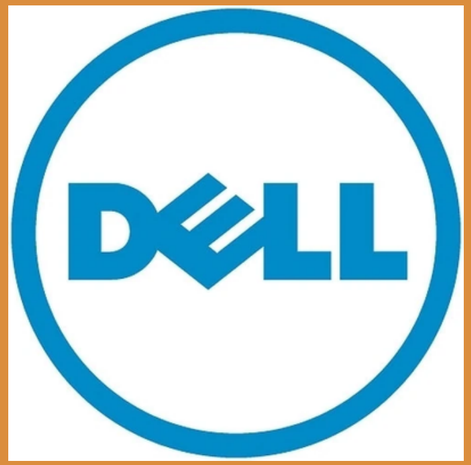
```
SELECT
  date_trunc('minute', data_hora) AS intervalo,
  AVG(temperatura) AS media_temperatura
FROM temperatura
WHERE cidade = 'Recife'
  AND data_hora > NOW() - INTERVAL '5 minutes'
GROUP BY intervalo
ORDER BY intervalo;
```

CASOS DE USO

CASOS DE USO

FONTE: HTTPS://WWW.INFLUXDATA.COM/CUSTOMERS/

DELL TECHNOLOGIES



- MONITORAMENTO DE PLATAFORMA ECS
- GERAÇÃO DE ALERTAS
- VISUALIZAÇÃO DE ESTADO DO SISTEMA

FONTE: HTTPS://WWW.INFLUXDATA.COM/CUSTOMERS/

SAP

- CLOUD FOUNDRY - MONITORAMENTO MULTICLOUD
- TESTES DE PERFORMANCE

FONTE: HTTPS://WWW.INFLUXDATA.COM/CUSTOMERS/

IBM

- IBM TRUSTEER - PREVENÇÃO DE FRAUDES
- POWER SYSTEMS - BENCHMARKING DE SERVIDORES

FONTE: HTTPS://WWW.INFLUXDATA.COM/CUSTOMERS/

OBRIGADO!

docker-compose.yml

```
version: '3.8'

services:
  influxdb:
    image: influxdb:2.7
    container_name: influxdb
    ports:
      - "8086:8086"
    volumes:
      - influxdb-data:/var/lib/influxdb2
    environment:
      - DOCKER_INFLUXDB_INIT_MODE=setup
      - DOCKER_INFLUXDB_INIT_USERNAME=admin
      - DOCKER_INFLUXDB_INIT_PASSWORD=admin123
      - DOCKER_INFLUXDB_INIT_ORG=MinhaOrganizacao
      - DOCKER_INFLUXDB_INIT_BUCKET=meu_bucket
      - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=meu_token_superseguro
    restart: unless-stopped

volumes:
  influxdb-data:
```