



TimescaleDB: analisi di time-series di consumi energetici

Candidata: Luisa Bottiglieri

Relatore: Prof.ssa Sonia Bergamaschi

Correlatore: Phd Luca Gagliardelli

Time-series data

- Sono dati che rappresentano come un sistema, un processo o un comportamento cambia nel tempo.
- Sono: time-centric, append-only, recent
- Sovrascritture e riempimento di dati mancanti: effettuati raramente

Usati in ambiti in cui vengono raccolte grandi quantità di dati in operazioni di **monitoraggio**, rappresentano una tipologia di dato in continua crescita:

- in una giornata con misurazioni al secondo: $60*60*24 = 86\ 400$ righe
- in un anno: $86\ 400*365 = 31\ 536\ 000$ righe

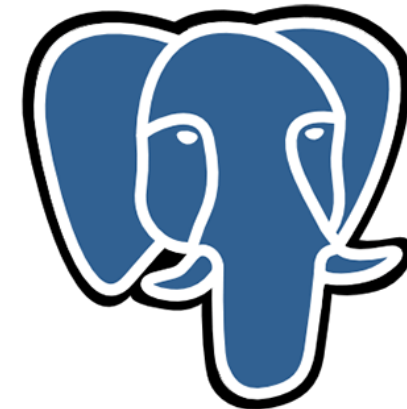
pod	ora	corrente	volt
5246	2020-08-29 00:00:00	1.504	23
5246	2020-08-29 00:00:01	1.504	23
5246	2020-08-29 00:00:02	1.504	23
5246	2020-08-29 00:00:03	1.504	23
5246	2020-08-29 00:00:04	1.504	23
5246	2020-08-29 00:00:05	1.504	23
5246	2020-08-29 00:00:06	1.504	23
5246	2020-08-29 00:00:07	1.504	23
5246	2020-08-29 00:00:08	1.504	23
5246	2020-08-29 00:00:09	1.504	23
5246	2020-08-29 00:00:10	1.504	23

TimescaleDB

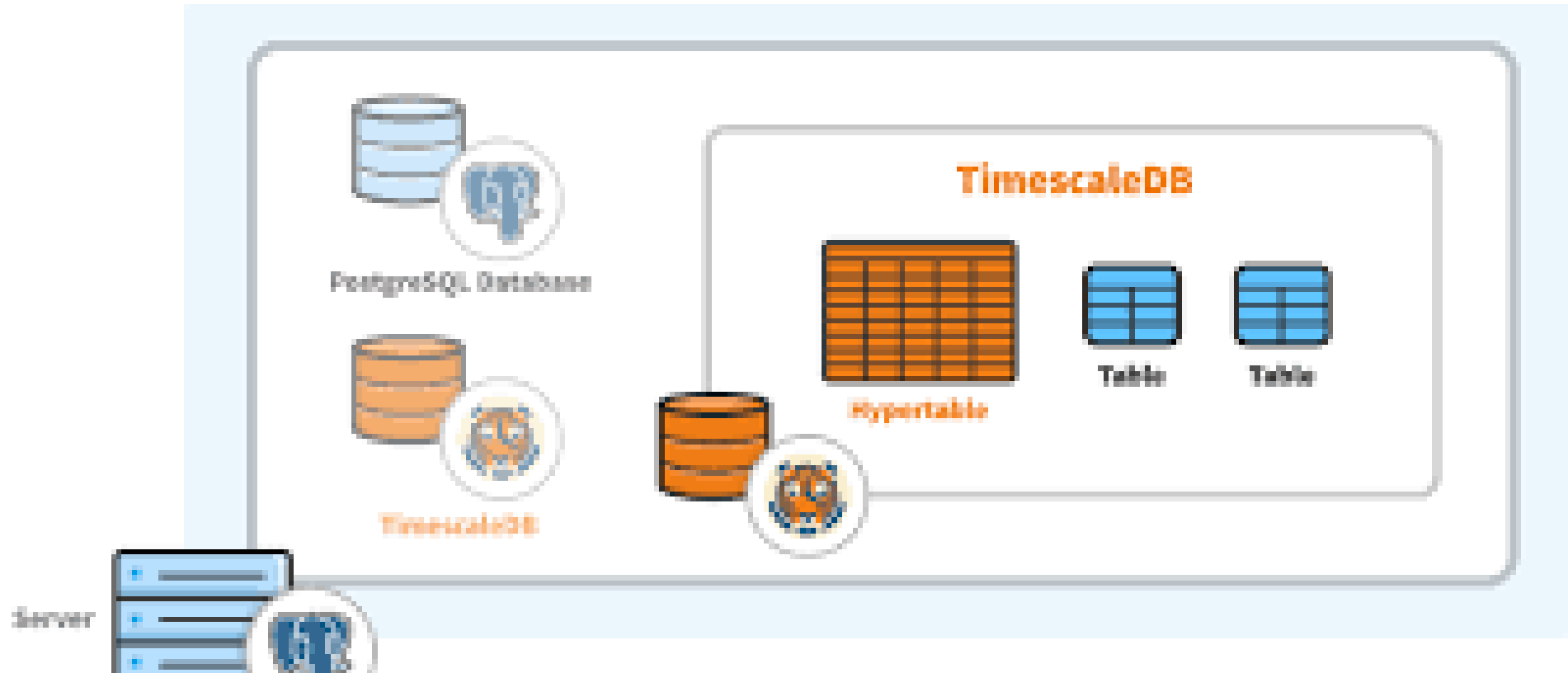
- È un DBMS open-source sviluppato dall'azienda Timescale per time-series data ottimizzato per l'inserimento veloce dei dati e per le query complesse.
- È un estensione di **PostgreSQL** → facilità d'uso
- È affidabile perché progettato sulla base di PostgreSQL, già ampiamente consolidato
- Introduce architettura **hypertables** e **chunks**



TIMESCALE

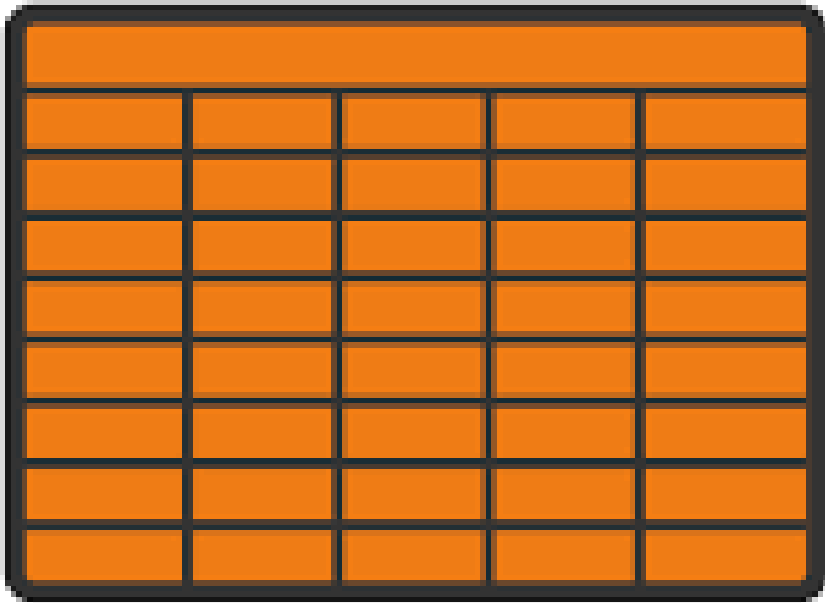


PostgreSQL

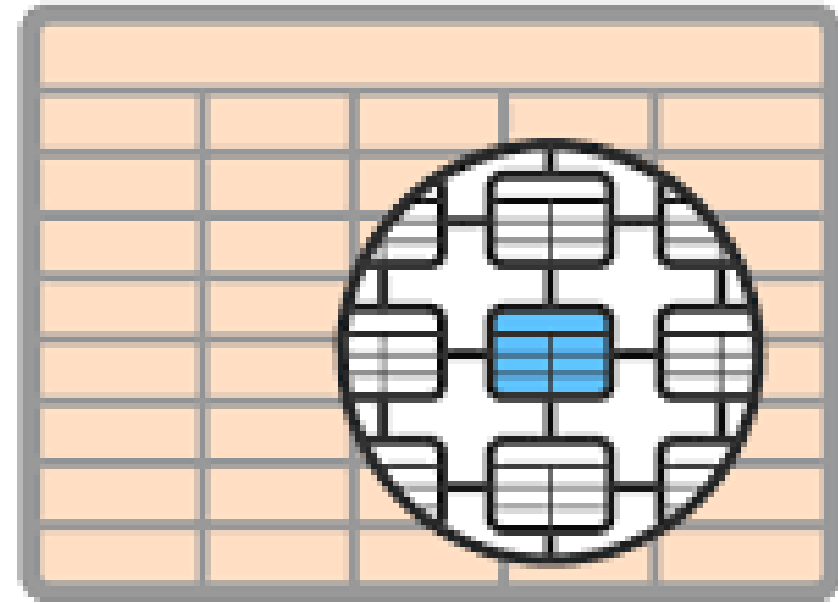


Hypertables

- Una **hypertable** è una tabella normale di PostgreSQL trasformata.
- È definita da uno schema standard con i nomi e i tipi delle colonne con almeno una colonna che specifichi un valore temporale e una colonna, opzionale, che specifichi una chiave di partizione ulteriore.
- Si gestisce con i comandi di PostgreSQL.



Hypertable



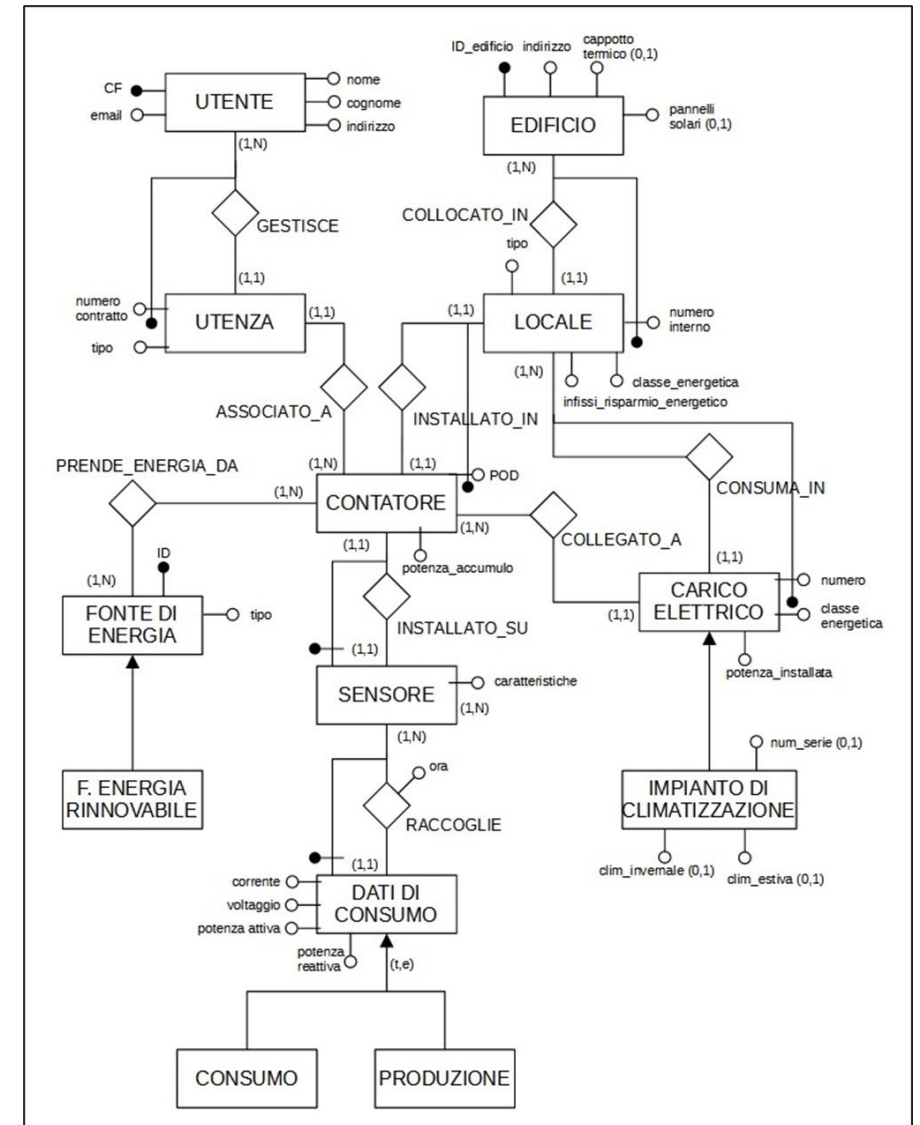
Chunk

Chunks

I chunks sono delle tabelle figlie della tabella hypertable. Ogni chunk corrisponde a uno specifico intervallo temporale e a una regione dello spazio della chiave di partizione

Time-series di consumi energetici

- Time-series (dati di consumo) raccolti da un sensore installato sul contatore.
- Dati come potenza, energia, corrente sono raccolti ogni secondo oppure ogni quarto d'ora
- Per analisi più complete si tiene traccia anche delle caratteristiche di costruzione dell'edificio e dei carichi elettrici



Gestione time-series con hypertable

```
CREATE TABLE consumi1 (  
  POD varchar(50) NOT NULL,  
  ora TIMESTAMP NOT NULL,  
  corrente double precision,  
  voltaggio double precision,  
  potenza_attiva double precision,  
  potenza_reattiva double precision,  
  PRIMARY KEY (POD, ora)  
  FOREIGN KEY (POD) REFERENCES sensore (POD) );
```

Tabella SQL normale

1. Attivare l'estensione "timescaledb" al database:

```
CREATE EXTENSION IF NOT EXISTS timescaledb;
```

2. Creare la hypertable:

```
SELECT create_hypertable('consumi1', 'ora');
```

Estensione ad hypertable con TimescaleDB

Interrogazioni sui time-series

Funzione `time_bucket()`: permette di raggruppare i risultati delle interrogazioni secondo intervalli di tempo desiderati.

Esempio: calcolare il valore medio della potenza attiva ogni cinque minuti per ogni POD.

```
SELECT pod, time_bucket('5 minutes', ora) AS
cinque_minuti,
avg(potenza_attiva) AS potenza_attiva
FROM consumi1
GROUP BY pod, cinque_minuti
ORDER BY cinque_minuti DESC
```

pod	cinque_minuti	potenza_attiva
00000100	2020-08-29 23:55:00	84.99453333333331
00000104	2020-08-29 23:55:00	124.91302666666665
00000108	2020-08-29 23:55:00	0
00000114	2020-08-29 23:55:00	157.21337999999997
00000115	2020-08-29 23:55:00	248.32111999999998

Confronto consumi annuali di due appartamenti

I due appartamenti si trovano nello stesso condominio.

```
SELECT time_bucket('30 days', data_lettura) as mese,  
avg(energia_attiva)*1000  
  
FROM consumicasa1  
  
WHERE (date_part('hour',data_lettura) >= 8 AND  
date_part('hour',data_lettura) < 19  
  
AND date_part('dow', data_lettura) BETWEEN 1 AND 5)  
  
AND date_part('year', data_lettura) = 2019  
  
GROUP BY mese;
```

Time_bucket() specifica di raggruppare il risultato per mese, il resto della query seleziona la fascia oraria e l'anno

```
CREATE TABLE consumicasa1_2019 AS  
  
SELECT mese, sum(consumi_fascia1_kw) AS  
energia_attiva_f1_kw,  
  
sum(consumi_fascia2_kw) AS energia_attiva_f2_kw,  
sum(consumi_fascia3_kw) AS energia_attiva_f3_kw  
  
FROM consumi2019  
  
GROUP BY mese  
  
ORDER BY mese ASC;
```

Si procede analogamente per l'altro appartamento.

```

SELECT s.POD, e.ID_edificio,
e.indirizzo, l.numero_interno,
l.tipo, l.classe_energetica,
e.cappotto_termico, l.infissi_risp_en
FROM sensore s, locale l, edificio e,
consumicasa1 c
WHERE s.num_interno_locale =
l.numero_interno
AND l.ID_edificio = e.ID_edificio AND
s.POD = c.POD
GROUP BY s.POD, ...

```

Query analoga anche per il secondo appartamento

pod	id_edificio	indirizzo	numero_interno	
IT000E00000001	2	via Giacomo Leopardi, 34, Modena	1	
tipo	classe_energetica	cappotto_termico	infissi_risp_en	pannelli_solari
appartamento	C	t	f	f

(1 row)

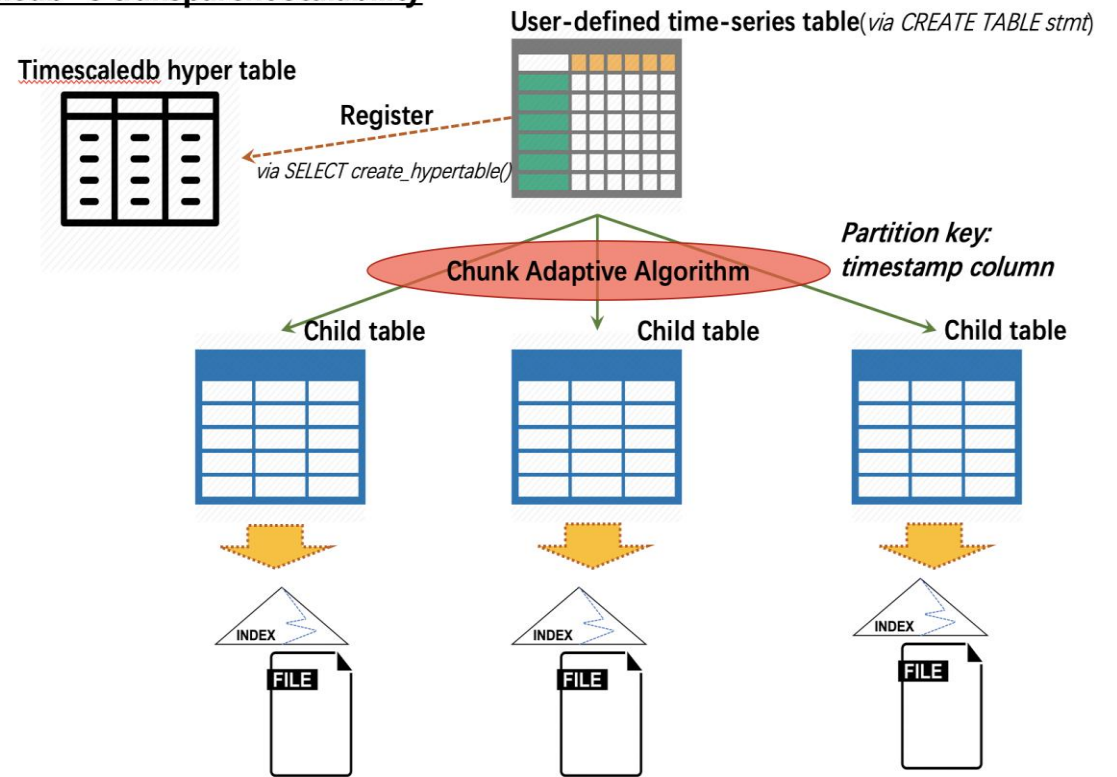
pod	id_edificio	indirizzo	numero_interno	
IT000E00000002	2	via Giacomo Leopardi, 34, Modena	2	
tipo	classe_energetica	cappotto_termico	infissi_risp_en	pannelli_solari
appartamento	A	t	t	f

(1 row)

Quando i dati diventano «tanti»

- La **dimensione** dei chunks per garantire l'accesso veloce dei dati in memoria: TimescaleDB dimensiona i chunks in modo tale che i nuovi dati vengano **inseriti** nelle tabelle più recenti in memoria, mentre i chunks più vecchi vengono compressi.
- **Scalabilità**: grazie al partizionamento del tempo sia per lo **scaling up** di un singolo nodo di dati sia per **scaling out** futuro per l'aggiunta di nuovi componenti
- Per rendere l'analisi dei dati più completa e fluida, TimescaleDB introduce la possibilità di riempire, al momento dell'interrogazione, valori mancanti con interpolazioni o copiando il valore precedente, usando la funzione **time_bucket_gapfill()**

Timescaledb' s transparent scalability





Grazie per l'attenzione

