



# Scaling Reads in Postgres using Readysset

## A New Era For Caching

PGDay Austria  
05/04/2025

# Agenda



# Agenda

---

- **Intro**
- **How Readysset works**
- **Using Readysset in your environment**
- **Demo**
- **Readysset Cloud**

# Intro



# Intro - Caching

---

- What is Cache



# Intro - Caching

---

- **What is Cache**
- **Maintenance / Invalidation**

*There are only two hard things in computer science: cache invalidation and naming things.*

*– Phil Karlton*

# Intro - Caching

---

- **What is Cache**
- **Maintenance / Invalidation**
- **Shared buffers**
- **Materialized views**
- **External caches e.g. Redis, Memcached**



Modern database scaling platform and  
SQL query caching system PostgreSQL  
(and MySQL).





# Readyset

---

- Based on MIT PhD Thesis - Partial State in Dataflow-  
Based Materialized Views (Noria) by Jon Gjengset
- <https://github.com/readyssettech/readysset>
- Core / Cloud
- No application code changes
- Automatic cache maintenance
- Horizontal Scale
- Wire compatible with PostgreSQL and MySQL

# How Readysset Works



# How Readysset Works

---

- **Snapshotting & CDC (Change Data Capture) / Replication**

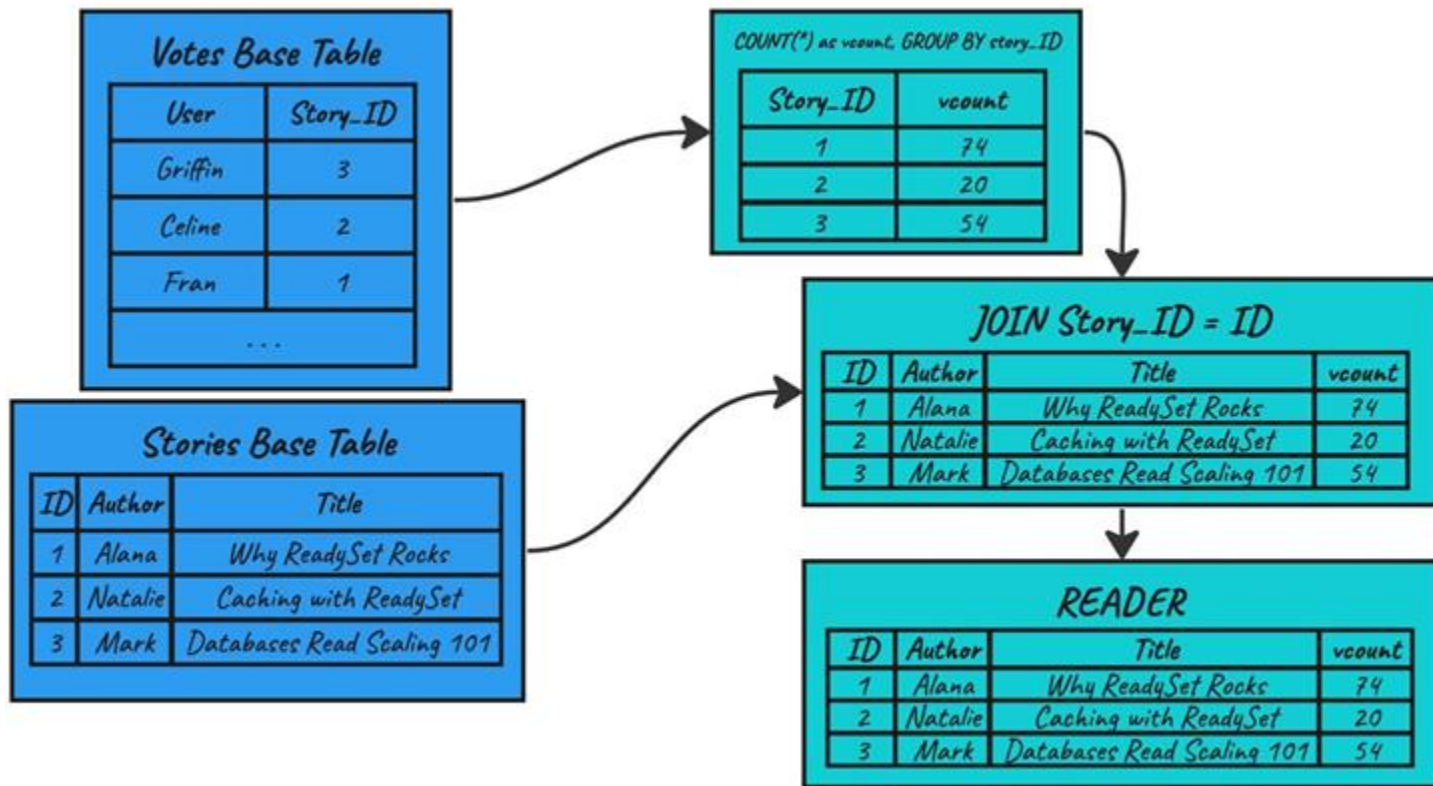
# How Readysset Works

---

- **Snapshotting & CDC (Change Data Capture) / Replication**
- **Dataflow Graph**

```
SELECT id, author, title, vcount
FROM stories JOIN
  (SELECT story_id, COUNT(*) AS vcount
   FROM votes GROUP BY story_id) AS VoteCount
  ON VoteCount.story_id = stories.id
WHERE stories.id = ?;
```

# How Readysset Works - DataFlow Graph

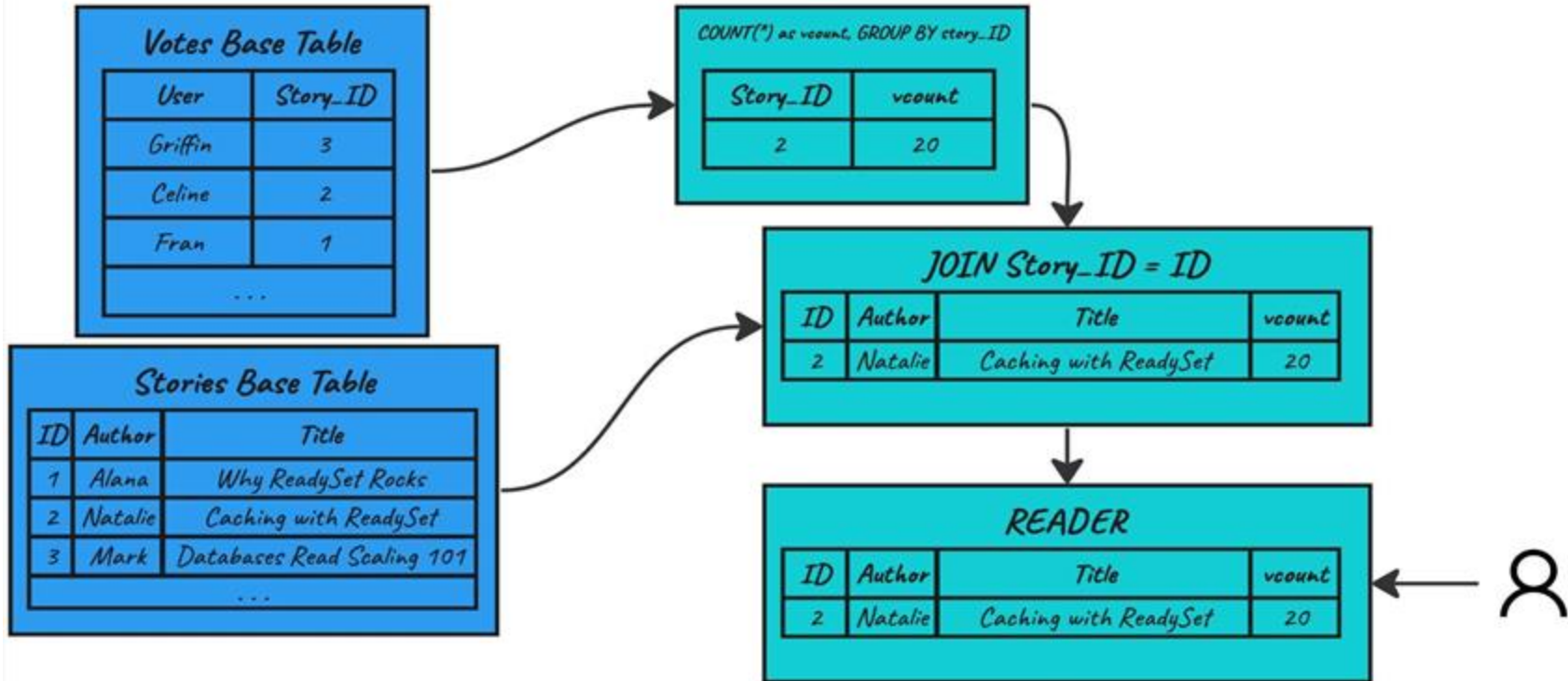


# How Readysset Works

---

- **Snapshotting & CDC (Change Data Capture) / Replication**
- **Dataflow Graph**
- **Partial Materialization**

# How Readysset Works



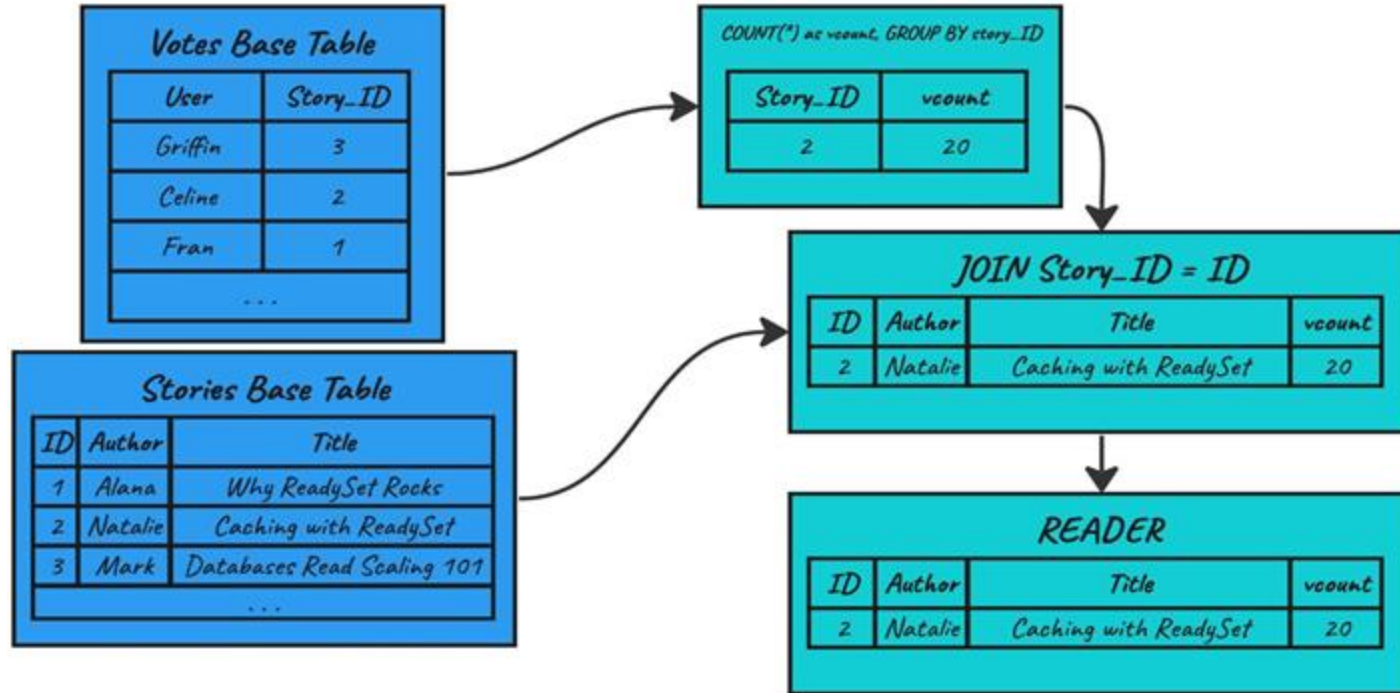
# How Readysset Works

---

- **Snapshotting & CDC (Change Data Capture) / Replication**
- **Dataflow Graph**
- **Partial Materialization**
- **Cache Miss**

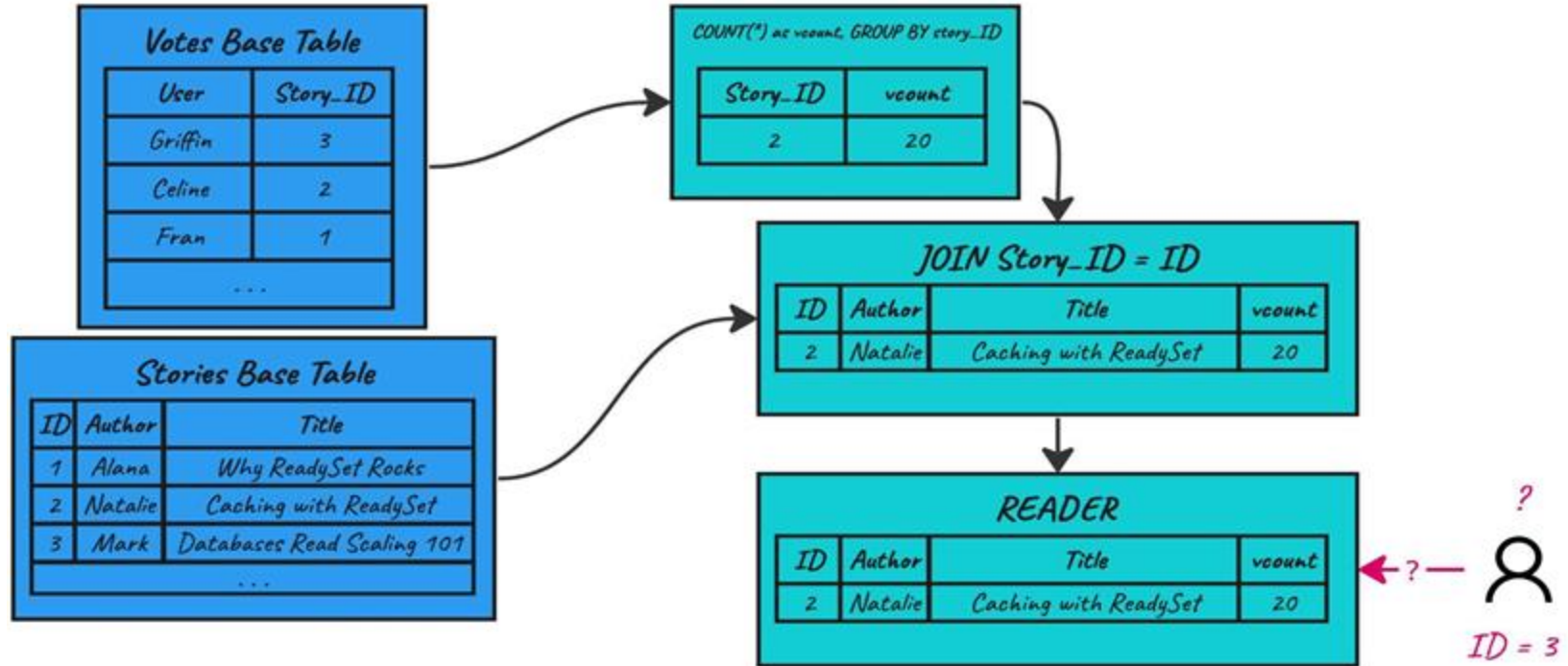


# How Readysset Works

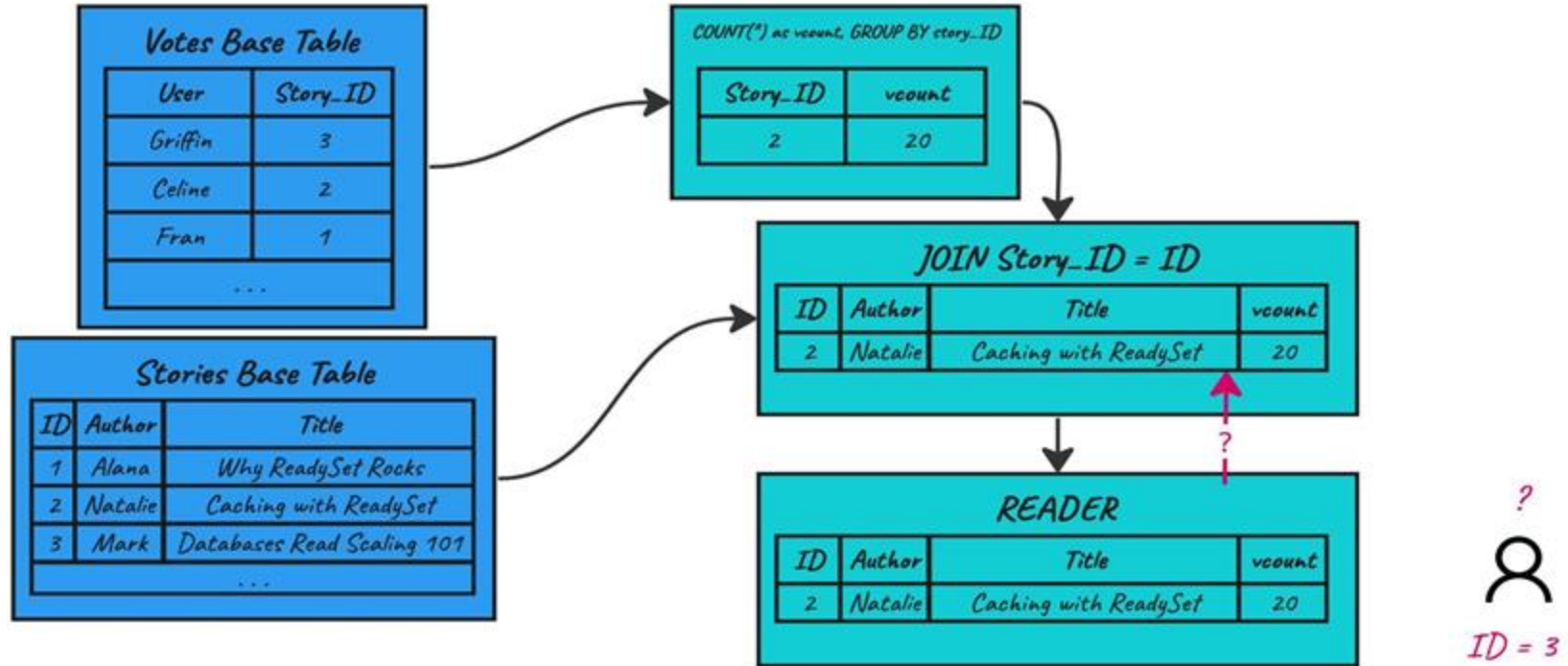


?  
ID = 3

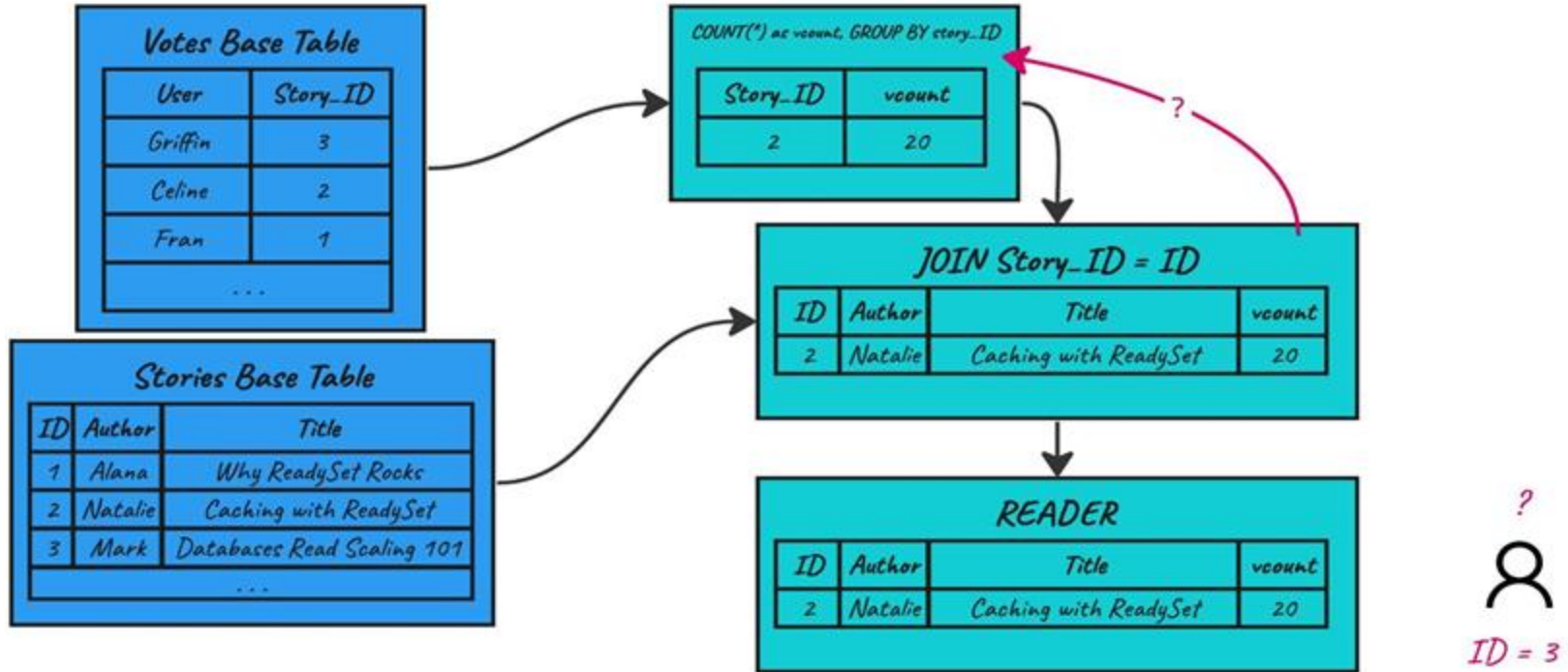
# How Readysset Works



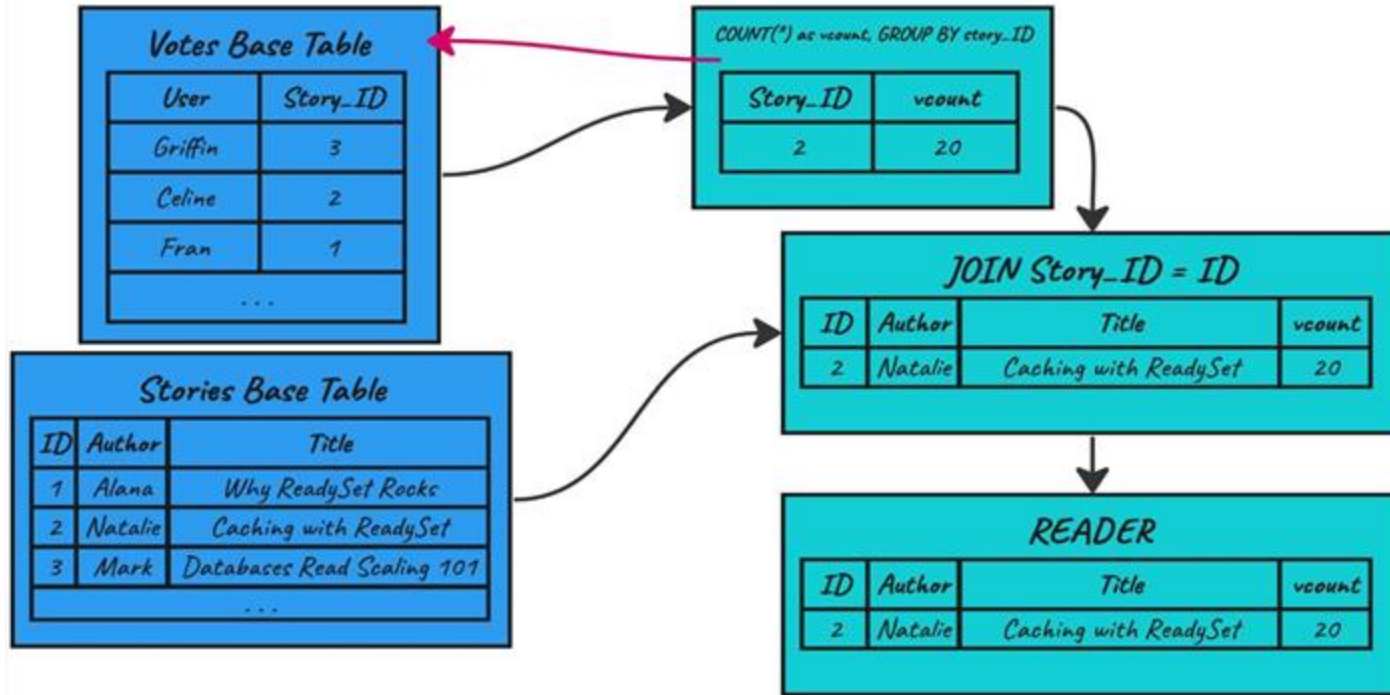
# How Readysset Works



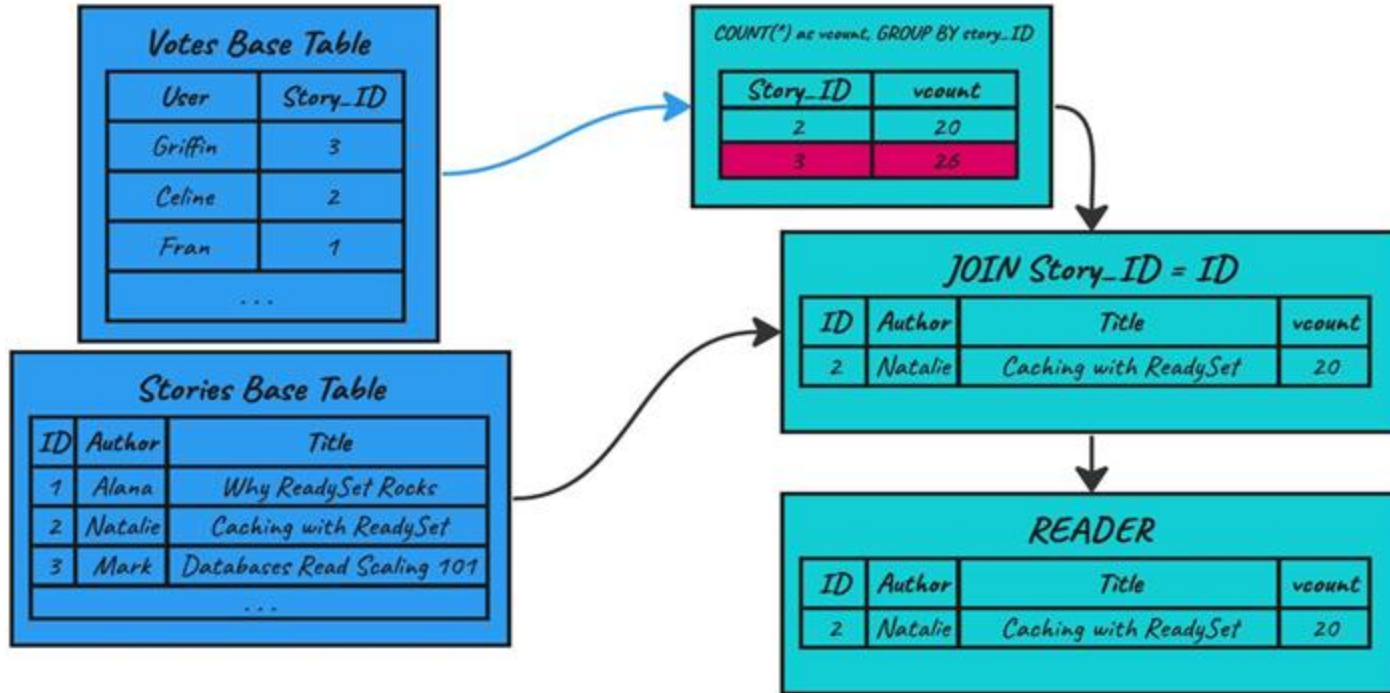
# How Readysset Works



# How Readysset Works



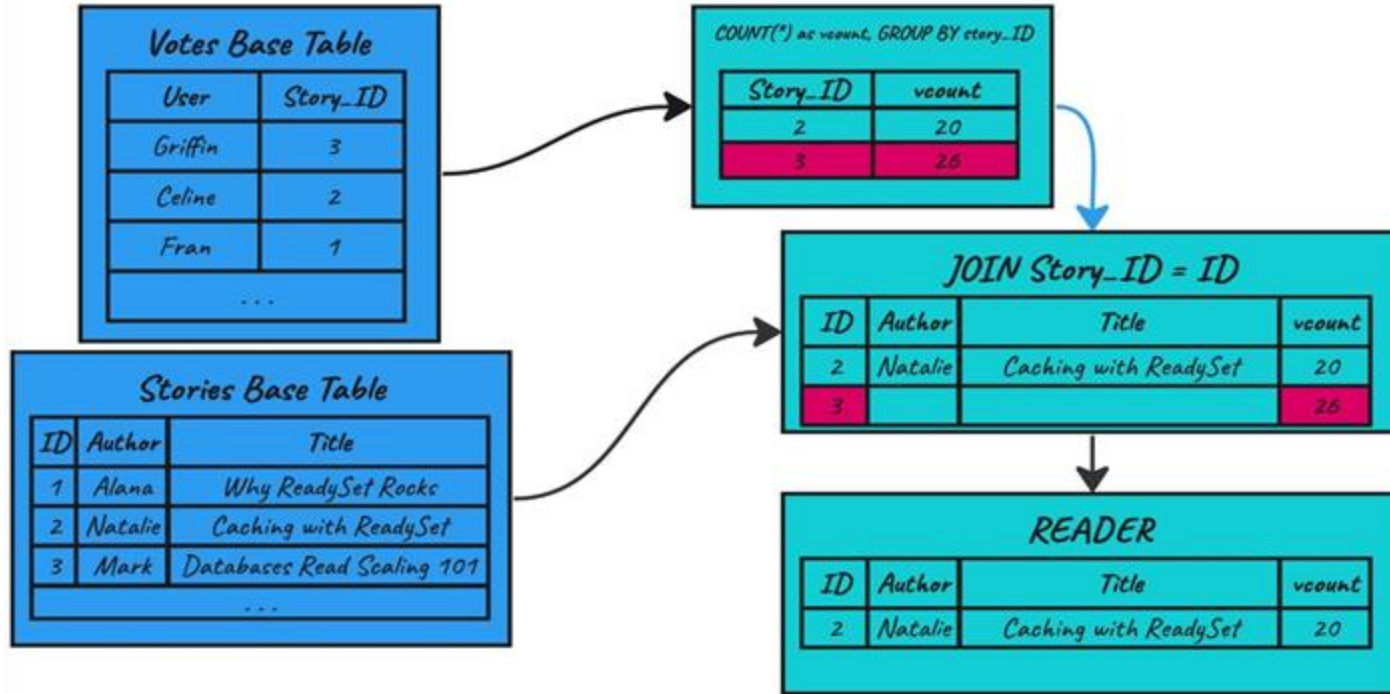
# How Readysset Works



?  
ID = 3

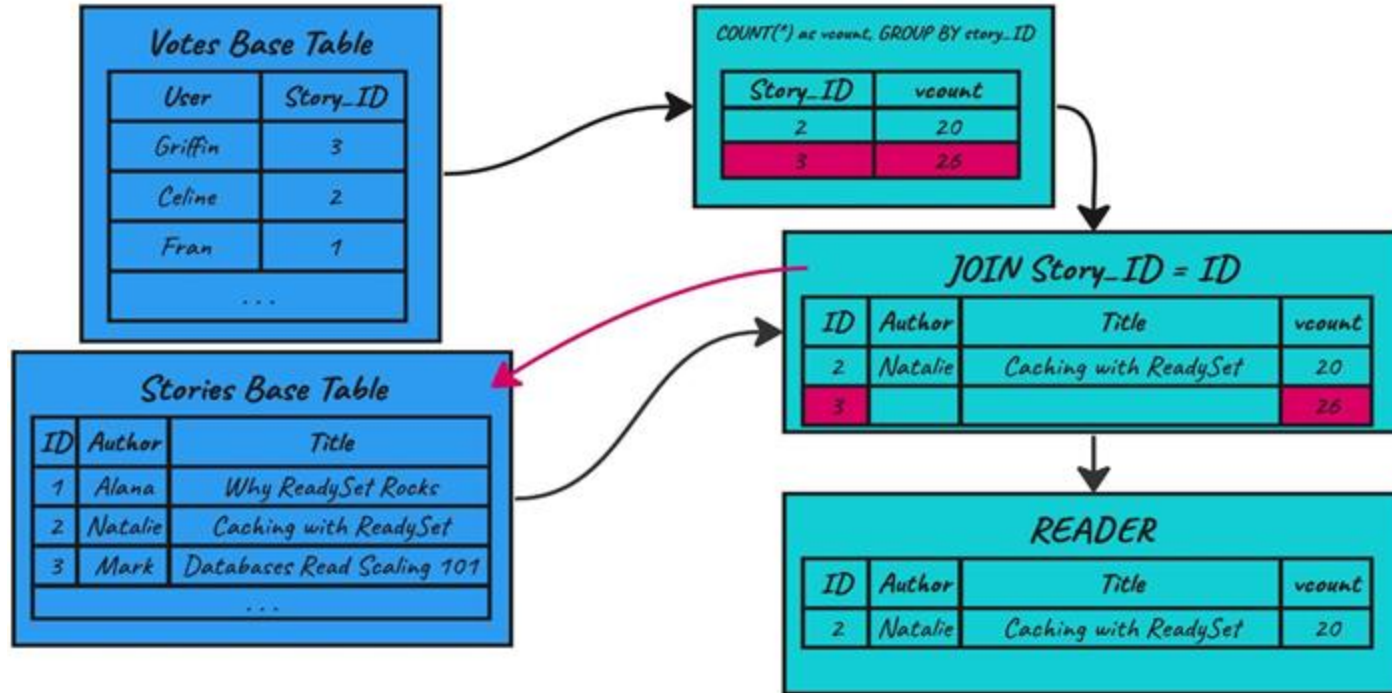


# How Readysset Works



?  
ID = 3

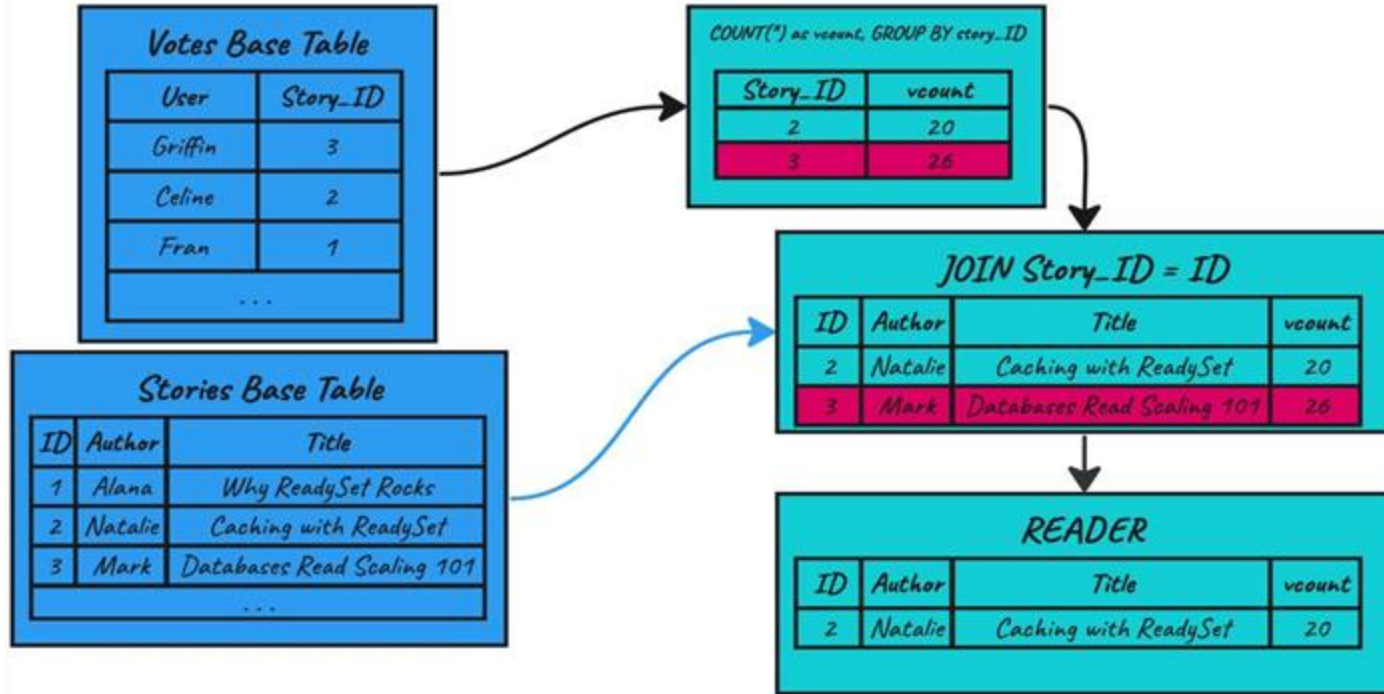
# How Readysset Works



?  
ID = 3

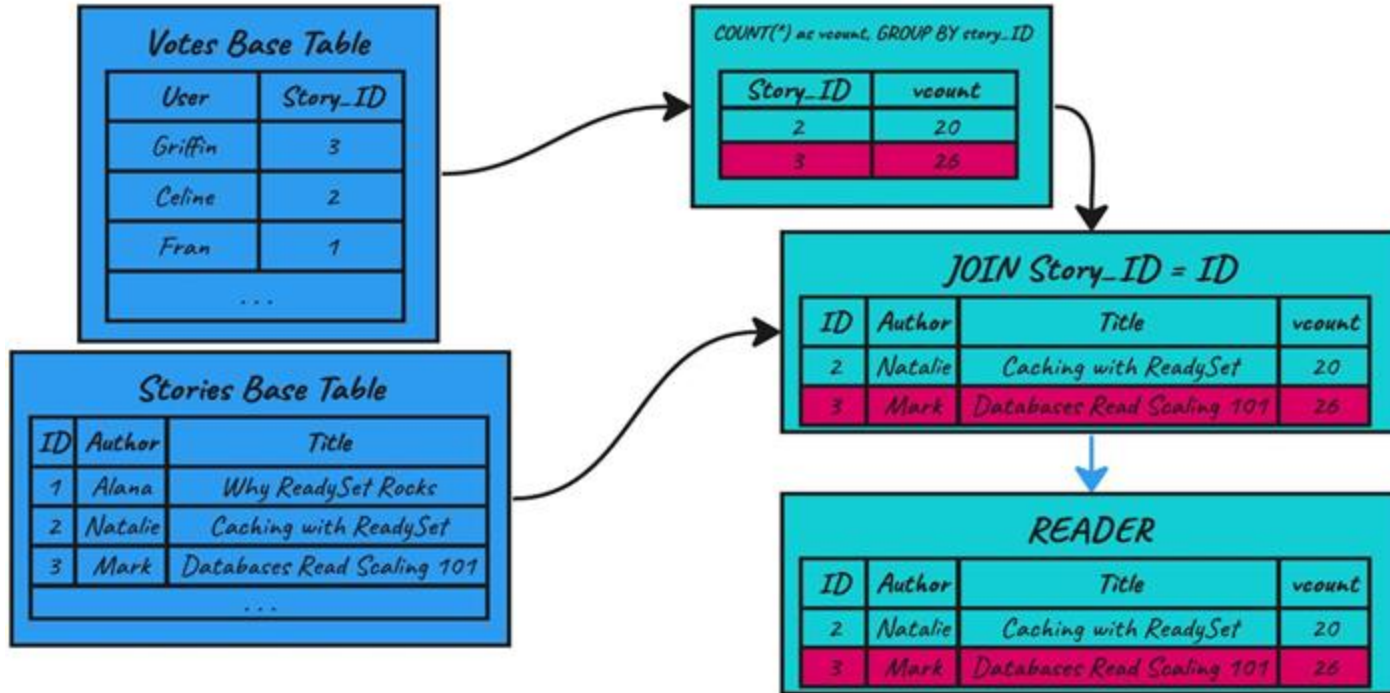


# How Readysset Works



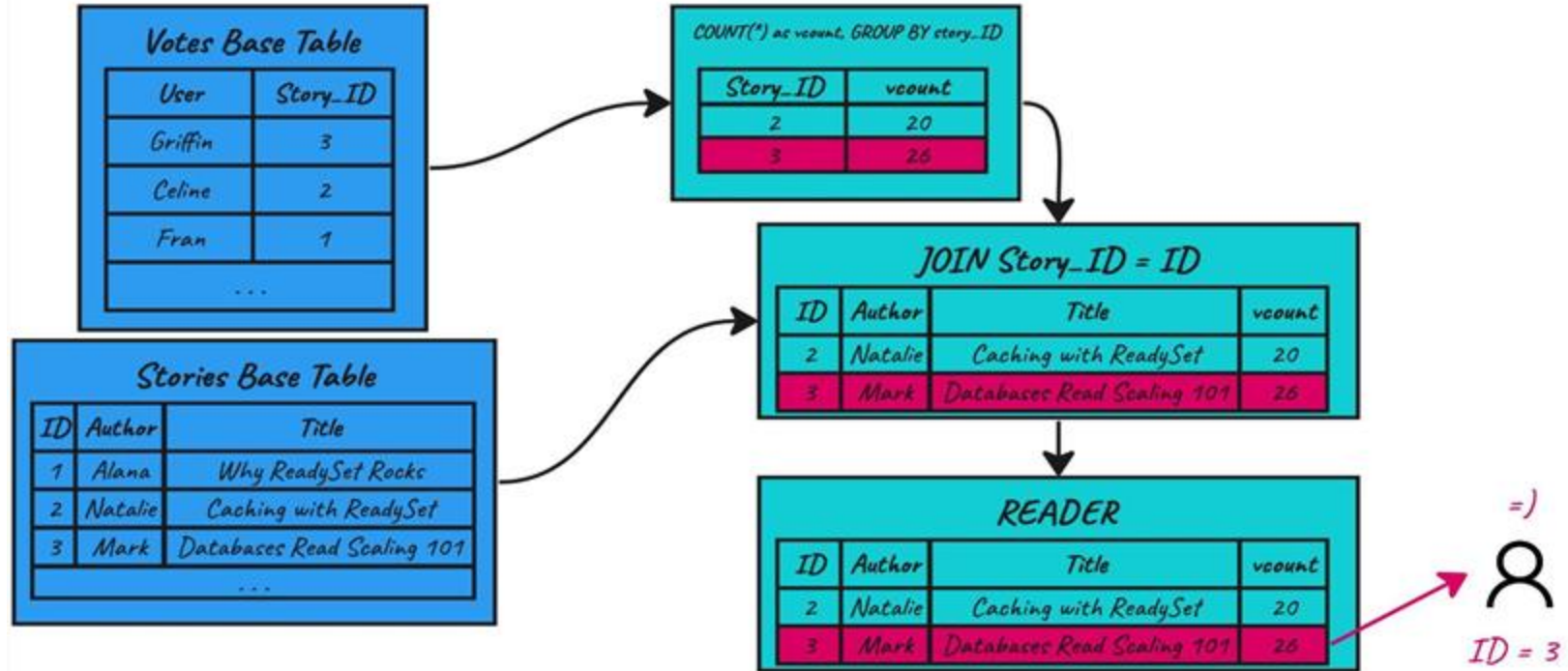
?  
ID = 3

# How Readysset Works



?  
ID = 3

# How Readysset Works

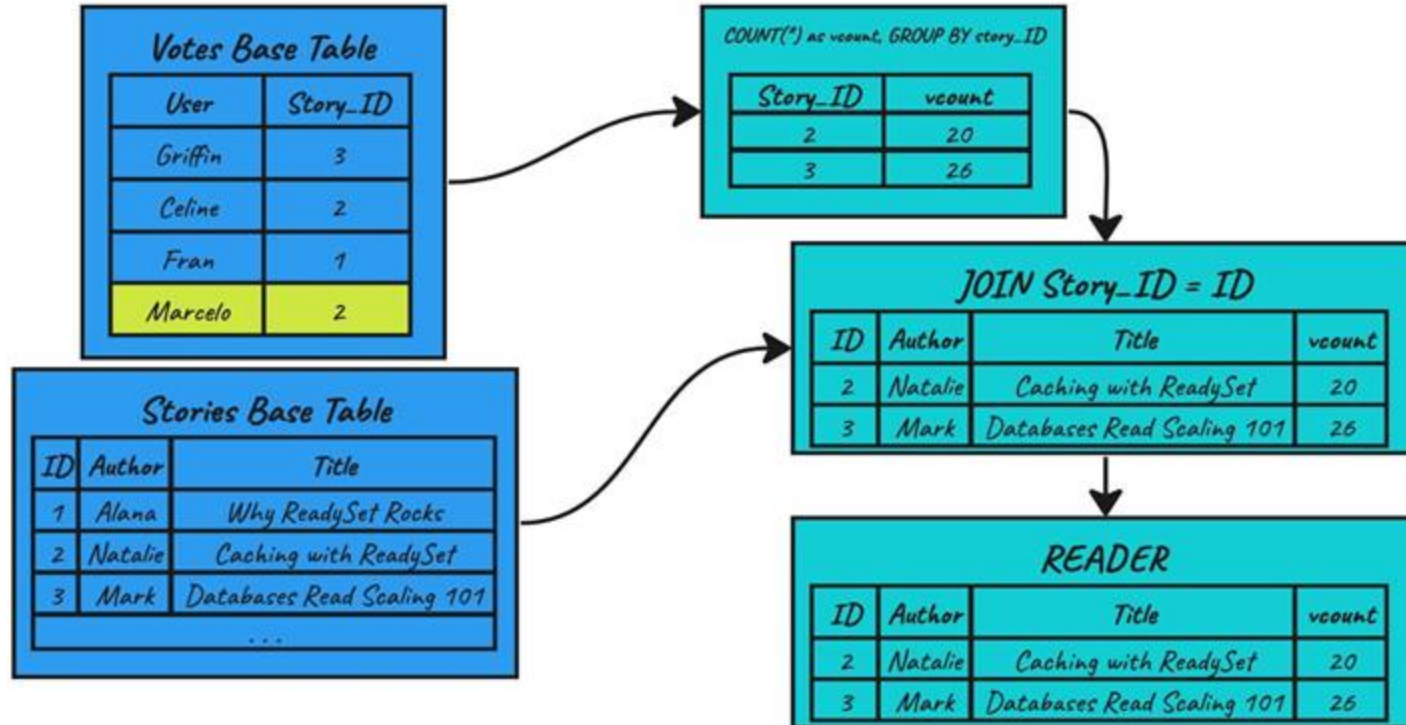


# How Readysset Works

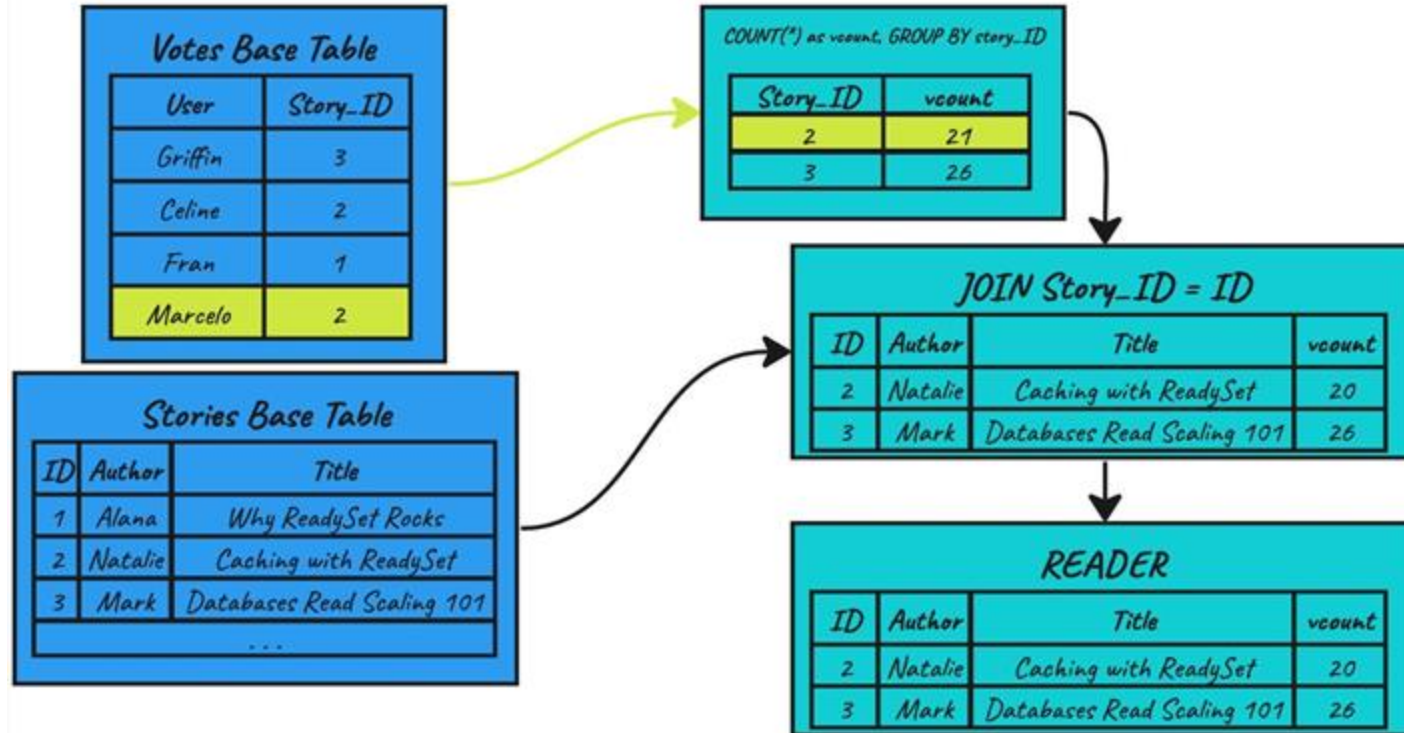
---

- **Snapshotting & CDC (Change Data Capture) / Replication**
- **Dataflow Graph**
- **Partial Materialization**
- **Cache Miss**
- **Cache Maintenance**

# How Readysset Works - Cache Maintenance

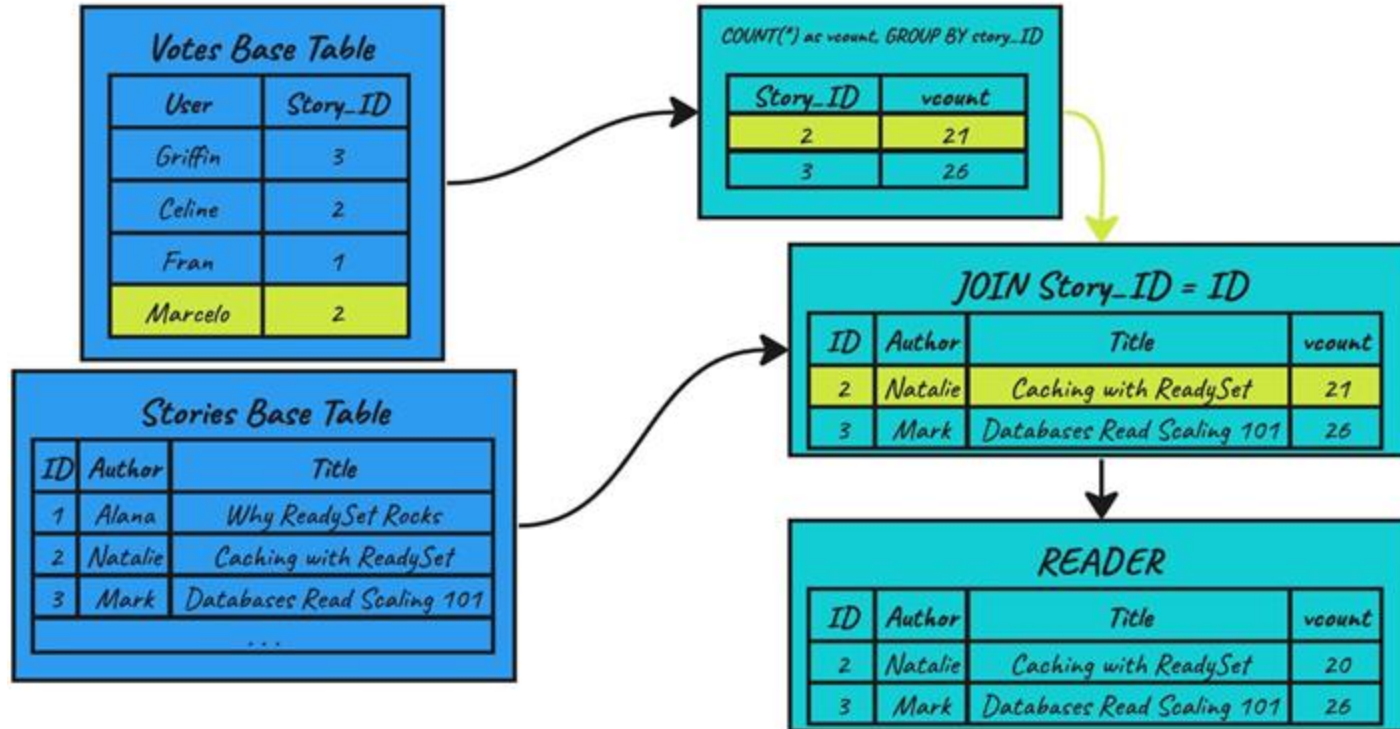


# How Readysset Works - Cache Maintenance

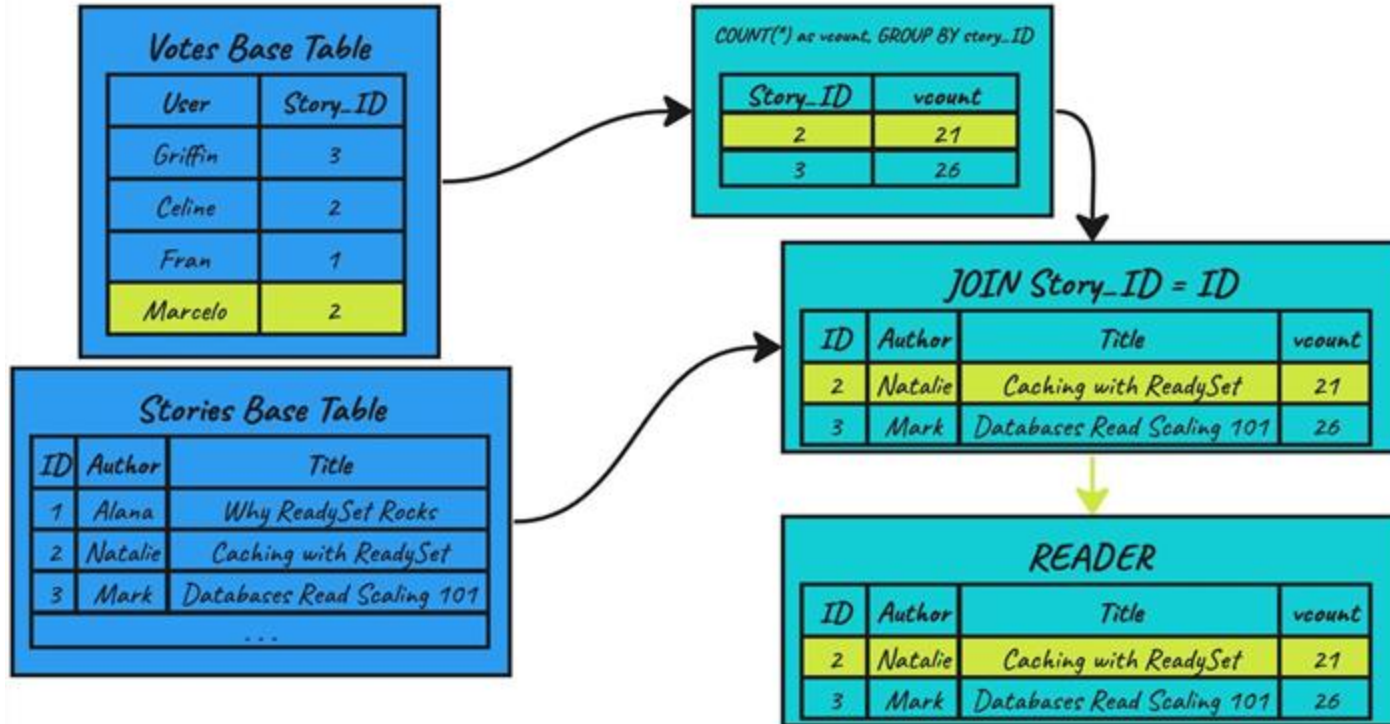




# How Readysset Works - Cache Maintenance



# How Readysset Works - Cache Maintenance



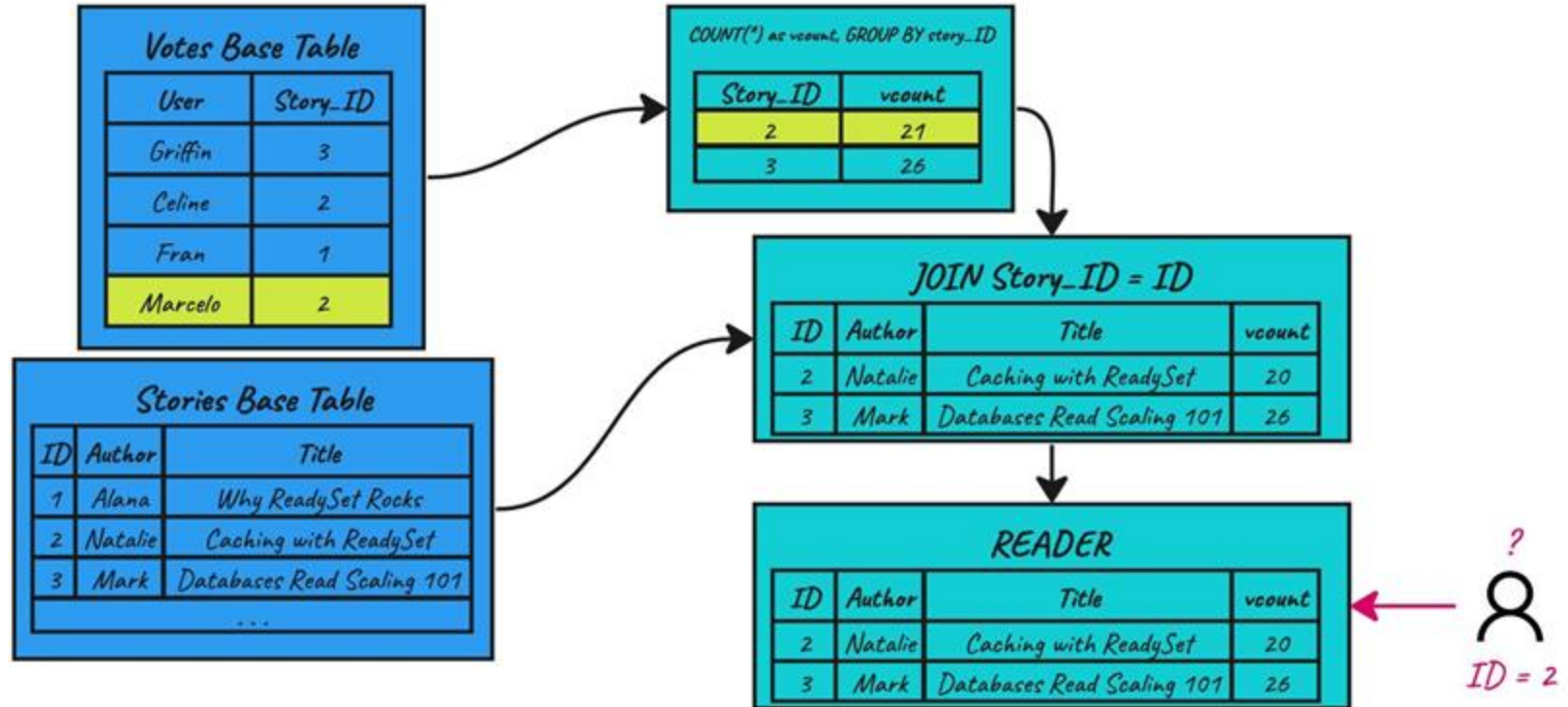


# How Readysset Works

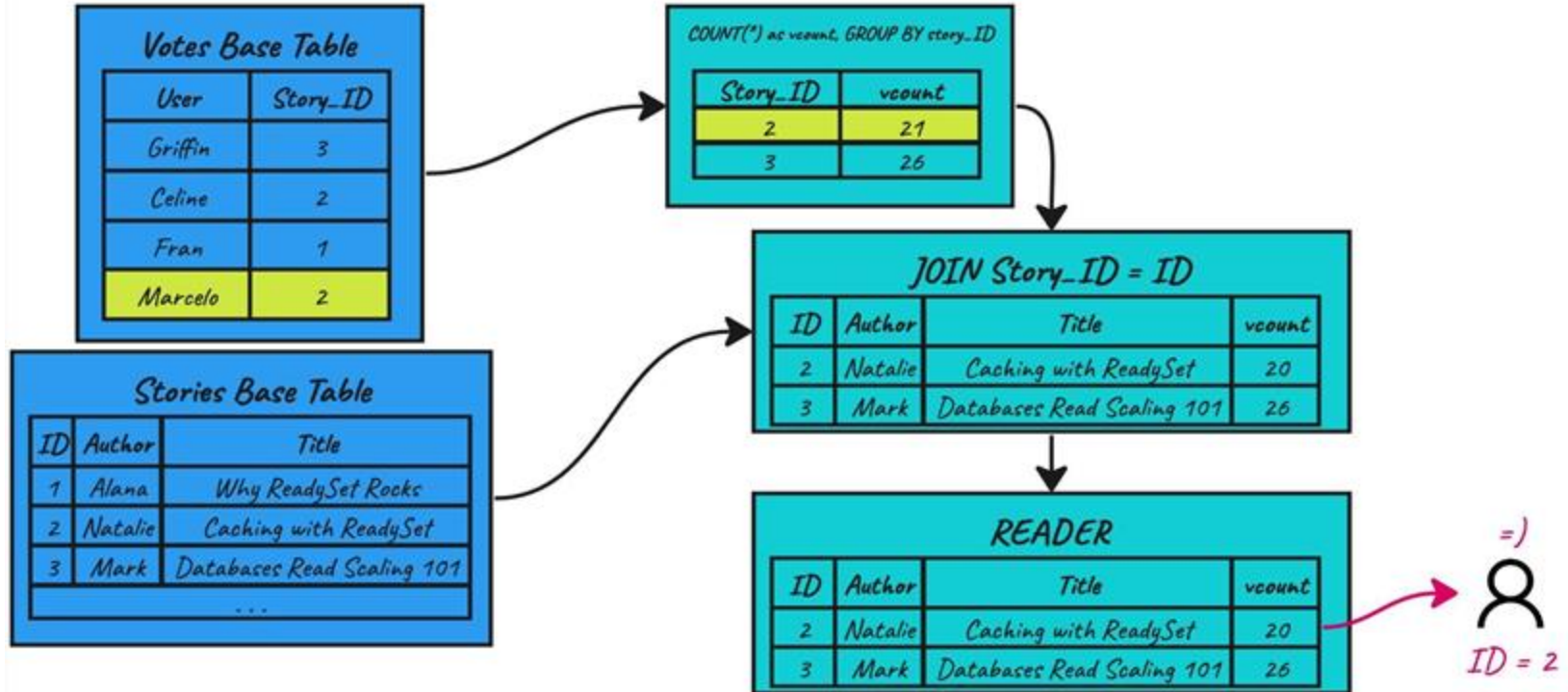
---

- **Snapshotting & CDC (Change Data Capture) / Replication**
- **Dataflow Graph**
- **Partial Materialization**
- **Cache Miss**
- **Cache Maintenance**
- **Stale Reads**

# How Readysset Works - Cache Maintenance - Stale Reads



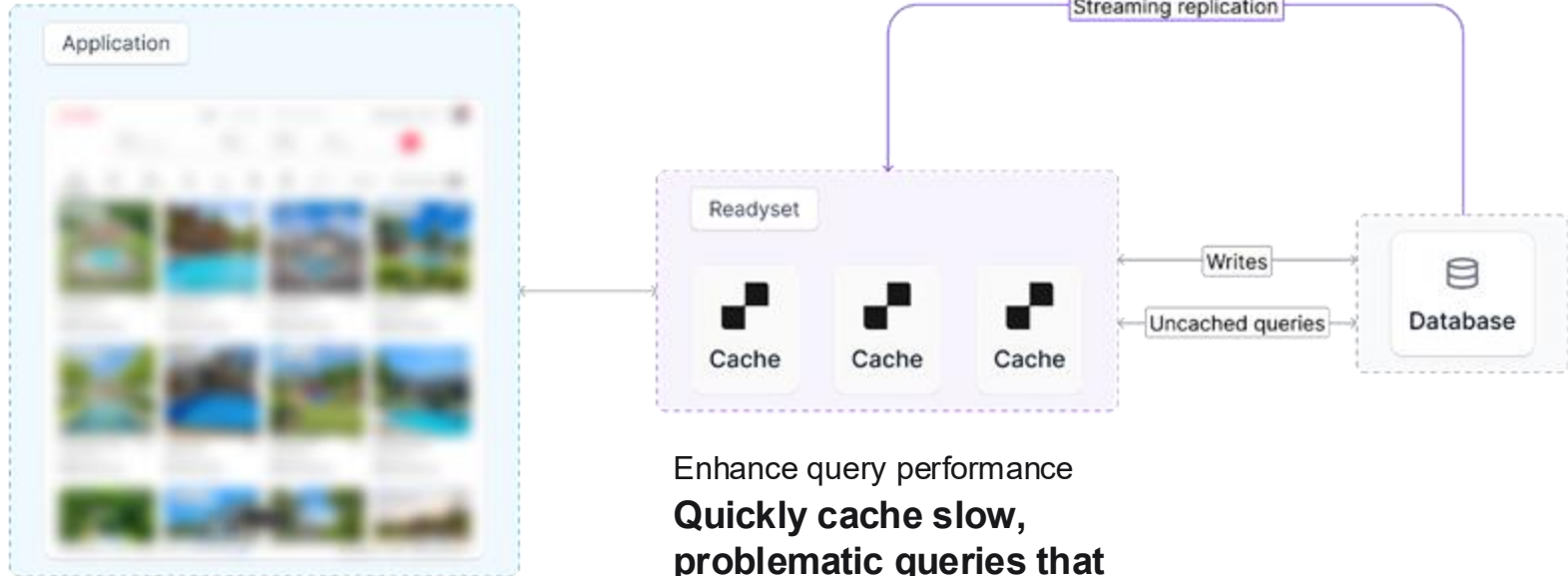
# How Readysset Works - Cache Maintenance - Stale Reads



# Using Readysset in your environment



## Readysset as a proxy

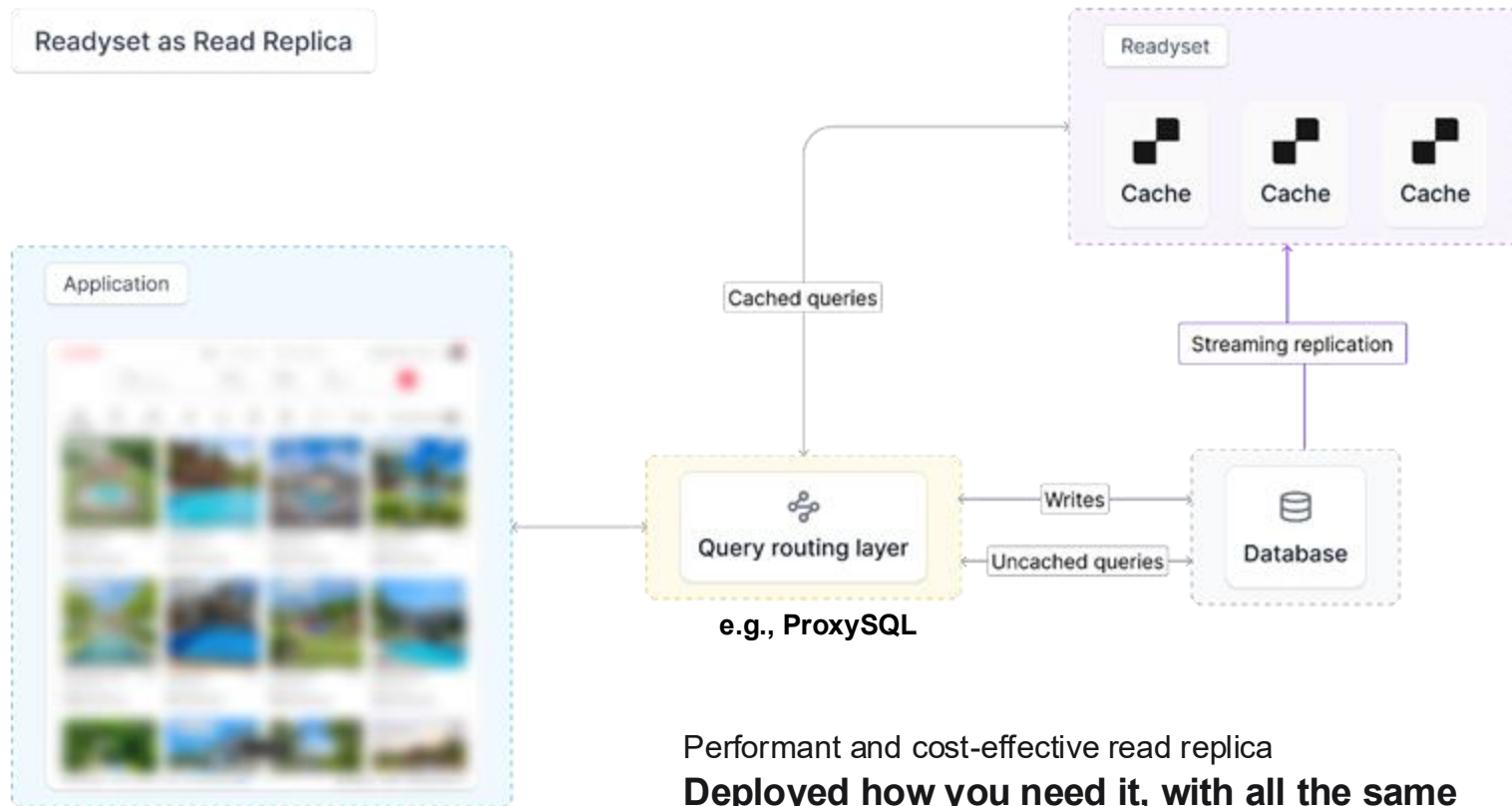


Enhance query performance

**Quickly cache slow,  
problematic queries that  
cause performance  
degradation.**



## Readysset as Read Replica



Performant and cost-effective read replica

**Deployed how you need it, with all the same performance benefits, no replication lag, and substantial cost improvements.**



# Demo



# Readysset Cloud







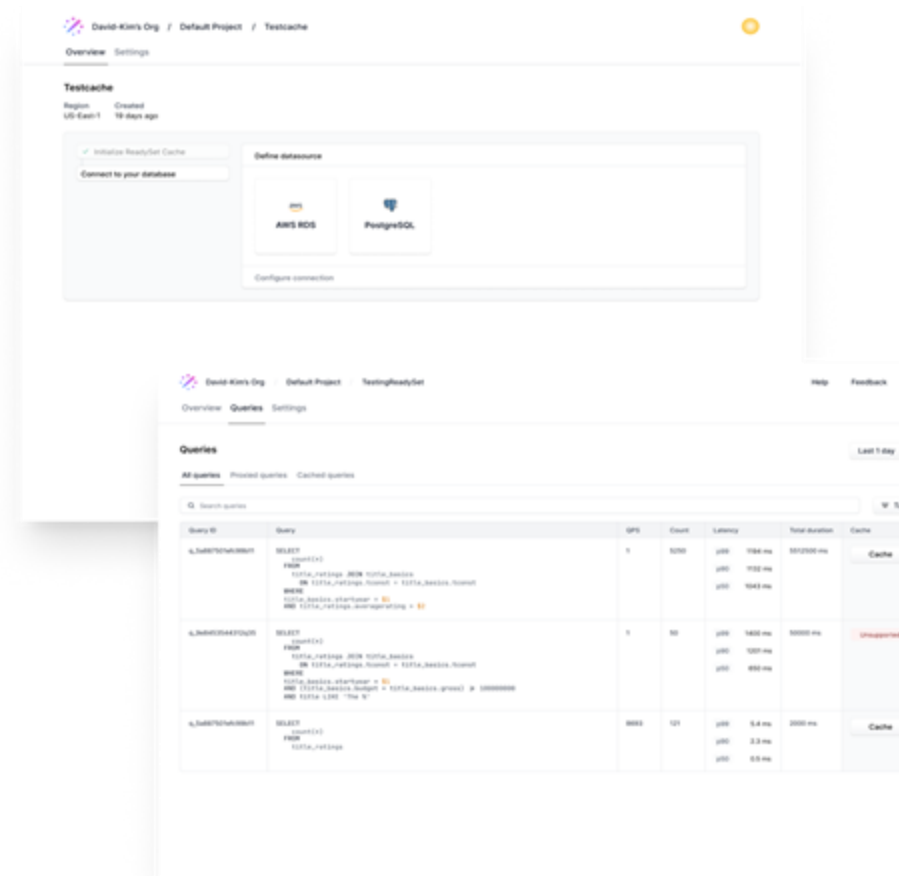
Fully-managed modern database scaling platform and SQL query caching system for PostgreSQL & MySQL.

Eliminate consistency issues

**Cache query results are automatically updated without eviction or slowdown.**

No added technical debt

**Easy to set up and maintain. Just swap a connection string and start caching.**



Now available on marketplace

# Summary



## **Simplify your stack.**

Replace or reduce the need for tech like:

- Redis
- Memcached
- In-memory key value caches
- Stream-processing systems
- Additional read replicas

SQL query cache with no  
complex migrations  
costly replicas  
code changes  
eviction logic  
TTLs



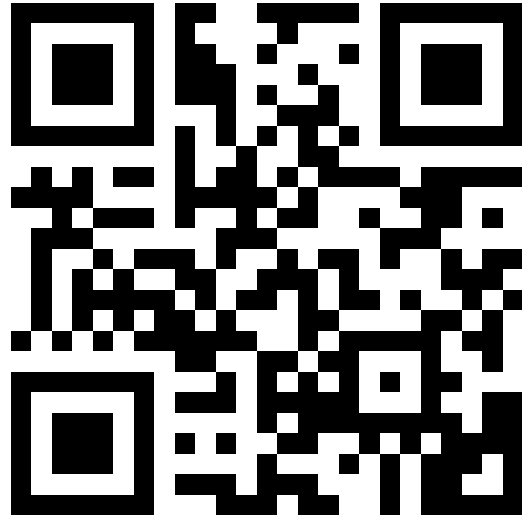
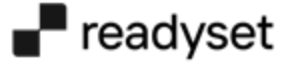
# Connect with us!

---

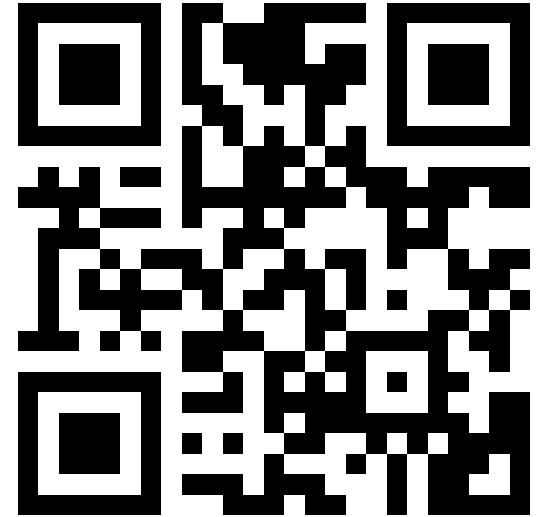
- Give it a try
  - <https://github.com/readyssettech/readysset/releases/>
  - If you like, star-us, blog / tweet about your experience.
- Slack Community - > <https://bit.ly/readysset-slack>
- X / Twitter @readyssetio -> <https://bit.ly/readysset-x>
- Linkedin @readyssettech -> <https://bit.ly/readysset-linkedin>
- Search Readysset on [\*\*https://aws.amazon.com/marketplace\*\*](https://aws.amazon.com/marketplace)

# Thank you!

---



**FREE 30-day trial – Readysset Cloud!**



[tanmay@readysset.io](mailto:tanmay@readysset.io)