

Capítulo 8 – Redis

Dia 3 – Playing with Other Databases

No terceiro dia com o **Redis**, vamos ver como ele pode atuar em conjunto com outros bancos de dados, aproveitando sua velocidade e versatilidade para complementar diferentes arquiteturas.

Redis como Cache para Bancos Relacionais

O uso mais comum do Redis é como **cache** em frente a bancos de dados relacionais (PostgreSQL, MySQL, Oracle).

- Exemplo: uma aplicação consulta um produto no PostgreSQL.
- Resultado é salvo no Redis:

```
SET produto:123 '{"nome": "Notebook", "preco": 4500}' EX 60
```

-  O **EX 60** define expiração em 60 segundos.
• Consultas subsequentes são respondidas em microssegundos pelo Redis.

 Vantagem: reduz a carga no banco relacional, melhora latência.

Redis + MongoDB

O Redis pode complementar o MongoDB: - MongoDB guarda documentos completos e persistentes. - Redis mantém dados de sessão, contadores e cache de consultas frequentes.

Exemplo: cache de ranking de artigos:

```
ZADD ranking_artigos 100 artigo:001  
ZADD ranking_artigos 250 artigo:002
```

Enquanto o MongoDB mantém os artigos, o Redis fornece o **ranking em tempo real**.

Redis + HBase ou Cassandra

Em sistemas distribuídos (HBase/Cassandra), o Redis pode atuar como **camada de acesso rápido**. - HBase → guarda bilhões de linhas (durabilidade e escala). - Redis → mantém resultados de consultas mais acessadas.

 Combinação útil em Big Data: persistência em HBase, agilidade em Redis.

Redis como Message Broker

Embora não seja especializado em filas como RabbitMQ ou Kafka, o Redis pode ser usado para **mensageria**.

- Usando **Lists**:

```
LPUSH fila_pedidos "pedido123"  
RPOP fila_pedidos
```

- Usando **Pub/Sub**:

```
PUBLISH notificacoes "Novo pedido"  
SUBSCRIBE notificacoes
```

 Bom para sistemas simples de fila e notificações em tempo real.

Redis em Arquiteturas Poliglotas

O Redis brilha quando usado em conjunto com outros bancos: - **PostgreSQL/MySQL** → dados relacionais e integridade. - **MongoDB** → documentos ricos. - **HBase/Cassandra** → Big Data distribuído. - **Redis** → cache, sessões, filas, rankings.

Cada banco cumpre seu papel específico, formando uma arquitetura poliglota.

Resumo do Dia 3

- Redis pode ser usado como **cache** para bancos relacionais e NoSQL.
 - Complementa MongoDB e HBase, oferecendo acesso em tempo real.
 - Atua como **message broker** simples com Pub/Sub ou listas.
 - Ideal em arquiteturas **poliglotas**, onde bancos diferentes cooperam.
-

Wrap-Up do Capítulo 8 – Redis

- **Dia 1** → CRUD e estruturas de dados (strings, listas, sets, hashes, TTL).
- **Dia 2** → Recursos avançados (transações, Lua, Pub/Sub, persistência, clusters).
- **Dia 3** → Integração com outros bancos e papel em arquiteturas poliglotas.

 Conclusão: o Redis não é apenas um cache, mas um **servidor de estruturas de dados** versátil, que acelera aplicações e coopera com outros bancos em sistemas modernos.