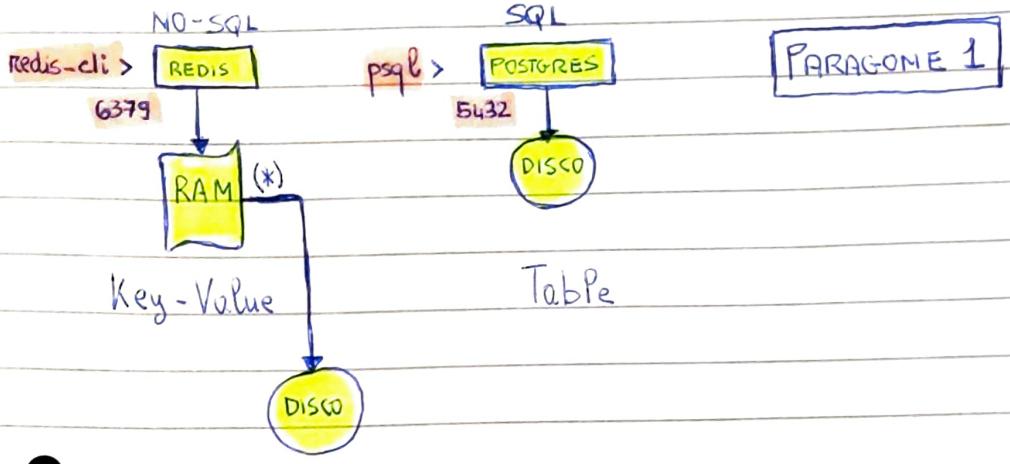
- **REDIS** (REmote DIctionary Server) Open Source



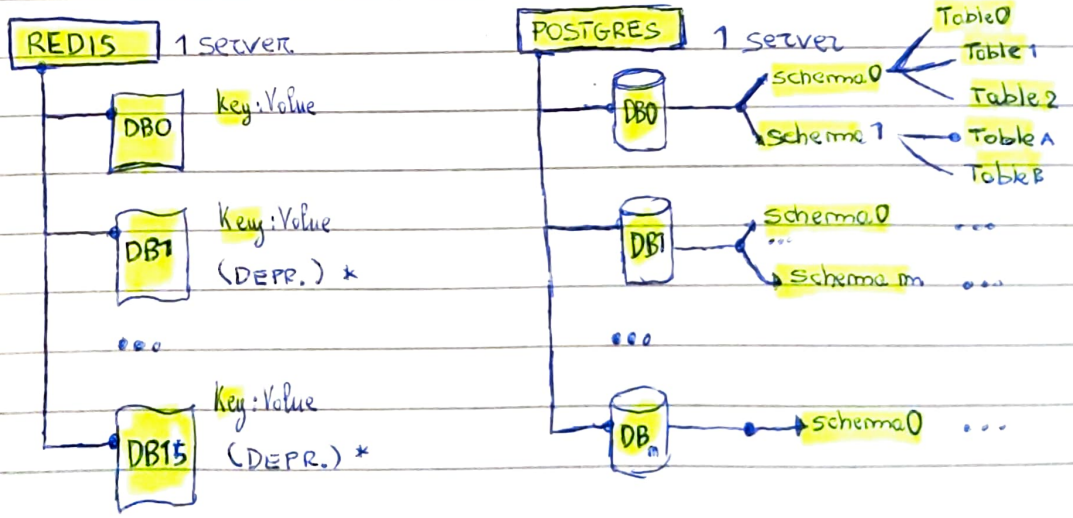
- (*): È possibile attivare il salvataggio dei dati su disco.

> redis-server : • STANDALONE (quello che interessa a me maggiormente)
• SENTINEL (replica dati)
• CLUSTER (sharding dati)

default: STANDALONE su porta 6379

- DATA-TYPES: STRING, LIST, HASH, SET, SORTEDSET, BITMAP, HYPERLOGLOG

- **PARAGONE 2**



- REDIS > DB0 > ~~Key~~ → Value

POSTGRES > DB0 > Schema1 > tableB > SQL → Value

- Dare dei buoni nomi alle chiavi è essenziale per poi poterle recuperare, magari anche con delle regexp.

Es. **namespase chiavi**

PG: musica : artisti : eventi

DB SCHEMA TABELLA

ID	LUOGO	...
1	SAN SIRO	
2	VERONA	...

RD: musica : artista : evento : id

key

musica : ligabue : concerto : 1 → SanSiro

musica : ligabue : concerto : 2 → Verona

NB: Non è che ci deve essere un mapping 1:1 tra postgres e redis ma è per dire che i nomi delle chiavi devono essere lungimiranti. Inoltre la scelta dipende anche dal tipo di dato che ho intenzione di usare

Es. **Tagging chiavi**

Posso anche "taggare" le chiavi, ma alla fine il concetto è sempre quello di avere un buon namespace.

Key-name : {tag} ⇝ users : 1 {users}

users : 2 {users}

...

(DEPR) * → Avere più di 1 DB in REDIS è deprecato. Se voglio 4 DB in REDIS è meglio lanciare 4 istanze di REDIS. Perché REDIS è single thread, se uso 1 REDIS per 4 DB rischio di creare colli di bottiglia

CPU [ ] — R1
CPU [ ] — R2
CPU [ ] — R3
CPU [ ] — R4

meglio che CPU — R1,2,3,4