



redis

# SQL? NoSQL? What's the difference?

SQL ist great for ...

- Relational data
- Transactions
- Durability

# Overkill for other things

- Caching? Don't do this
- No updates, no relations? There are better ways
- Speed. It's possible, but why bother?

# What's Redis?

- key-value swiss-army knife
- in-memory
- blazingly fast
- persistent

# In-memory

- Everything in memory
- Read/write is fast
- Process dies → the data is gone

# Persistence

- Append-only on each write, sync to disk
- Snapshot every few minutes

# Key-Value?!?!

> SET OTS rocks

OK

> GET OTS

"rocks"

> GET dortmund

(nil)

# More than just strings

```
> SET attendees 0
```

```
OK
```

```
> INCR attendees
```

```
(integer) 1
```

```
> INCRBY attendees 41
```

```
(integer) 42
```

```
> INCR OTS
```

```
(error) ERR value is not an integer or out of range
```

# Lists

```
> LPUSH list a b c  
(integer) 3
```

```
> LRANGE list 0 -1
```

```
1) "c"
```

```
2) "b"
```

```
3) "a"
```

# Lists

```
> RPUSH list foo
(integer) 4
> LRANGE list 0 -1
1) "c"
2) "b"
3) "a"
4) "foo"
```

# Hashes

```
> HSET hash name 'Grace Hopper'  
(integer) 1  
> HSET hash profession  
  'computer scientist'  
(integer) 1
```

# Hashes

> HGETALL hash

1) "name"

2) "Grace Hopper"

3) "profession"

4) "computer scientist"

# Sets

```
> SADD set foo bar baz
(integer) 3
> SMEMBERS set
1) "foo"
2) "baz"
3) "bar"
```

# Sets

```
> SADD set foo  
(integer) 0  
> SMEMBERS set  
1) "foo"  
2) "baz"  
3) "bar"
```

# Sorted Sets

```
> ZADD zset 1940 Alan  
    1953 Richard 1916 Claude  
    1957 Sophie 1969 Linus  
(integer) 5
```

# Sorted Sets

```
> ZRANGE zset 0 -1
```

```
1) "Claude"
```

```
2) "Alan"
```

```
3) "Richard"
```

```
4) "Sophie"
```

```
5) "Linus"
```

# Sorted Sets

```
> ZRANGEBYSCORE zset 1950 +inf
1) "Richard"
2) "Sophie"
3) "Linus"
> ZRANGEBYSCORE zset -inf 1950
1) "Claude"
2) "Alan"
```

# Bitmaps

```
> SETBIT bitmap 9 1  
(integer) 0  
> GET bitmap  
"\x00@"
```

# HyperLogLogs

```
> PFADD hll a b c d e f g
```

```
(integer) 1
```

```
> PFCOUNT hll
```

```
(integer) 7
```

```
> GET hll
```

```
"HYLL\x01\x00\x00\x00\a\x00\x00  
\x00\x00\x00\x00\x00Fm\x80V\f\x  
80D<\x848\x80P\xB1\x84I\x8C\x80  
Bm\x80BZ"
```

# Scripts

```
> EVAL 'return "Scripts are ato  
mic"' 0  
"Scripts are atomic"
```

# KEYS

> KEYS \*

(error) KEYS is an expensive command.

→ Never use KEYS.

*above error is not an actual response ;)*

```
> INFO memory
# Memory
used_memory:531344
used_memory_human:518.89K
used_memory_rss:9658368
used_memory_peak:531344
used_memory_peak_human:518.89K
used_memory_lua:44032
mem_fragmentation_ratio:18.18
mem_allocator:jemalloc-3.6.0
```

# High Availability with Sentinel

- Master-Slave Setup for redundancy / load-balancing
- Monitoring of Redis Instances
- Automatic Failover on failure

# Cluster Support

- *Real* Clustering
- Shard your data across multiple instances
- Nodes organize by themselves
- → 3.0 RC

# Use Cases

- Cache
- Task queues
- Real-time analytics
- Log processing
- *your-use-case-here*

# Real-world usage

- Instagram - User  $\leftrightarrow$  Photo mapping
- Twitter - Timeline with newest tweets
- Pinterest - Follower Graph
- Flickr - Real-time updates

# Thanks for listening!

[redis.io](https://redis.io)

[try.redis.io](https://try.redis.io)

P.S.: I got stickers

OpenTechSchool Dortmund 16.12.2014  
- Learners Meetup

Jan-Erik Rediger / @badboy\_  
fnordig.de