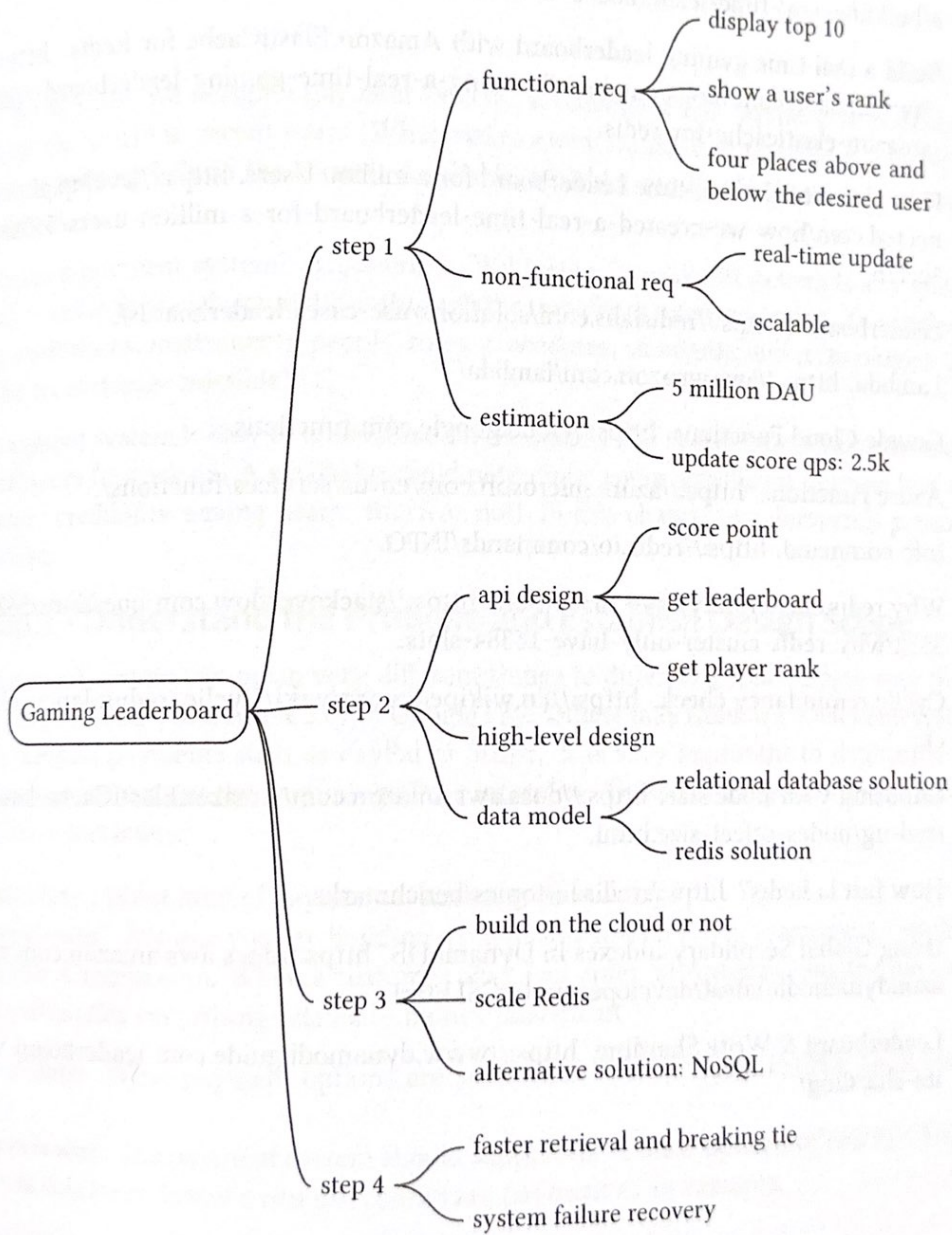## System failure recovery

The Redis cluster can potentially experience a large-scale failure. Given the design above, we could create a script that leverages the fact that the MySQL database records an entry with a timestamp each time a user won a game. We could iterate through all of the entries for each user, and call ZINCRBY once per entry, per user. This would allow us to recreate the leaderboard offline if necessary, in case of a large-scale outage.

Congratulations on getting this far! Now give yourself a pat on the back. Good job!

Chapter Summary

Gaming Leaderboard
- step 1
  - functional req
    - display top 10
    - show a user's rank
    - four places above and below the desired user
  - non-functional req
    - real-time update
    - scalable
  - estimation
    - 5 million DAU
    - update score qps: 2.5k
- step 2
  - api design
    - score point
    - get leaderboard
    - get player rank
  - high-level design
  - data model
    - relational database solution
    - redis solution
- step 3
  - build on the cloud or not
  - scale Redis
  - alternative solution: NoSQL
- step 4
  - faster retrieval and breaking tie
  - system failure recovery

Reference Material

[1] Man-in-the-middle attack. https://en.wikipedia.org/wiki/Man-in-the-middle_attack.

[2] Redis Sorted Set source code. https://github.com/redis/redis/blob/unstable/src/t_zset.c.

[3] Geekbang. https://static001.geekbang.org/resource/image/46/a9/46d283cd82c987153b3fe0c76dfba8a9.jpg.

[4] Building real-time Leaderboard with Redis. https://medium.com/@sandeep4.verma/building-real-time-leaderboard-with-redis-82c98aa47b9f.

[5] Build a real-time gaming leaderboard with Amazon ElastiCache for Redis. https://aws.amazon.com/blogs/database/building-a-real-time-gaming-leaderboard-with-amazon-elasticache-for-redis.

[6] How we created a real-time Leaderboard for a million Users. https://levelup.gitconnected.com/how-we-created-a-real-time-leaderboard-for-a-million-users-555aaa3ccf7b.

[7] Leaderboards. https://redislabs.com/solutions/use-cases/leaderboards/.

[8] Lambda. https://aws.amazon.com/lambda/.

[9] Google Cloud Functions. https://cloud.google.com/functions.

[10] Azure Functions. https://azure.microsoft.com/en-us/services/functions/.

[11] Info command. https://redis.io/commands/INFO.

[12] Why redis cluster only have 16384 slots. https://stackoverflow.com/questions/36203532/why-redis-cluster-only-have-16384-slots.

[13] Cyclic redundancy check. https://en.wikipedia.org/wiki/Cyclic_redundancy_check.

[14] Choosing your node size. https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/nodes-select-size.html.

[15] How fast is Redis? https://redis.io/topics/benchmarks.

[16] Using Global Secondary Indexes in DynamoDB. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html.

[17] Leaderboard & Write Sharding. https://www.dynamodbguide.com/leaderboard-write-sharding/.