Step 4: Database Design

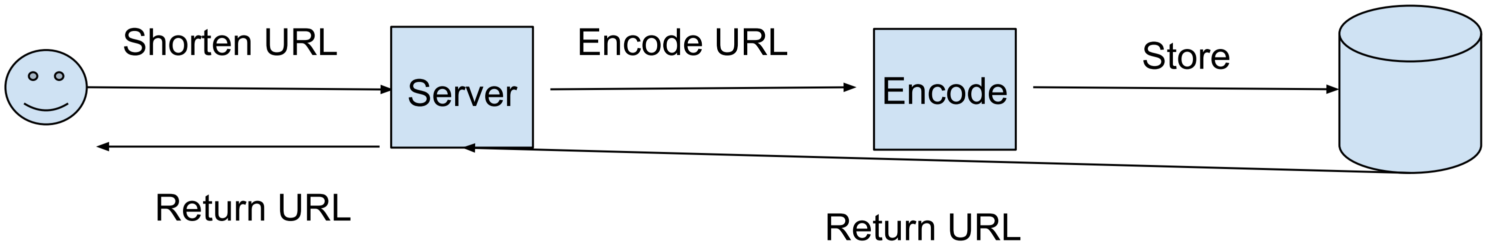
Considerations
- Need to store billions of small objects
- No relationship between records
- Read-heavy service

Two tables:
- URL with primary key Hash:varchar(16)
    - Original URL, creationDate, expirationDate, UserID
- UserID with primary key UserID:int
    - Name, Email, CreationDate, LastLogin

NoSQL e.g., DynamoDB/Cassandra for storing billions of rows with no relationships, easy to scale

Step 5: Basic System Design

To encode URL:
- Append userID (unique) to input URL, perform MD5 hash



Step 6: Detailed System Design

Data partitioning and replication
- Range based partitioning: store URLs in separate partitions based on first hash letter
- Hash-based partition: calculate partition based on hash

Cache
- Previously calculated 256GB easily fit on one machine
- Use LRU
- Cache sits between Server and Database. If URL not found, go to DB and update cache.

Load balancer
- Between client and server, server and DB, server and cache