Hash-based sharding

🧠 Sharding কী?

👉 **Sharding = database কে horizontally ভাগ করা**
যাতে:

- বেশি data handle করা যায়
- বেশি traffic handle করা যায়

📌 প্রতিটা shard = independent database

❌ Sharding কেন দরকার?

Single DB হলে:

- Storage limit
- CPU bottleneck
- Single point of failure

✅ Sharded DB

Shard-1   Shard-2   Shard-3

✓ Parallel reads/writes
✓ Horizontal scaling

🔑 Hash-Based Sharding (Core Topic)

1. Definition

👉 **Shard selection করা হয় hash(key) দিয়ে**

shard = hash(shard_key) % number_of_shards

2. Example (Very Important)

ধরি:

- 4 shards
- shard_key = userId

hash(101) % 4 = 1 → Shard-1

hash(202) % 4 = 2 → Shard-2

hash(303) % 4 = 3 → Shard-3

📌 Same key → always same shard

3. Architecture Diagram (Mental)

Client

↓

App Server

↓

Hash Router

↓

Shard-0 | Shard-1 | Shard-2 | Shard-3

4. Why Hash-Based Sharding?

✅ **Advantages**

- Uniform data distribution

- Avoids hot shards
- Simple logic

📌 Interview line:

"Hash-based sharding evenly distributes data across shards."

❌ **Disadvantages (VERY IMPORTANT)**

- **Re-sharding problem**
- Hard to add/remove shard
- Range queries inefficient

📌 Re-sharding pain:

Change shards from 4 → 5

→ almost all data must move

5. Query Pattern Impact

🔹 **Point Query (Good)**

Get user by userId

→ Direct shard lookup

🔹 **Range Query (Bad)**

Users between id 100–500

→ All shards scan

6. Hash-Based vs Range-Based (Quick Contrast)

Aspect	**Hash-Based**	**Range-Based**
Distribution	Even	Can hotspot
Range query	❌	✅
Re-sharding	Hard	Easier
Popular for	User data	Time-series

7. Consistent Hashing (IMPORTANT EXTENSION)

To fix re-sharding issue:
👉 **Consistent Hashing**

- Shards on hash ring
- Minimal data movement
- Used by DynamoDB, Cassandra

📌 Interview line:

“Consistent hashing minimizes re-sharding impact.”

8. Real-World Usage

Instagram

- Users sharded by userId (hash)
- Feeds cached in Redis
- Media via CDN

Amazon

- Customer data → hash-based shards

- Orders → strong consistency shards

9. Common Interview Mistakes ❌

- “Hash-based sharding is always best”
- “Sharding solves all problems”

✓ Correct:

“Shard strategy depends on access pattern.”

🏆 FAANG Answer Template 💎

Use this 👇

“Hash-based sharding distributes data using a hash function on the shard key, providing even load distribution. However, it makes range queries inefficient and re-sharding expensive, which is why systems often combine it with consistent hashing.”

🧠 One-Line Memory Hook

Hash sharding = even load, poor range queries