Database Sharding vs **Database Partition**

Scenario

- **Handling large datasets:** Sharding
- **Scaling read operations:** Replication
- **Ensuring high availability:** Replication
- **Reducing query response time:** Sharding
- **Supporting global users:** Sharding
- **Failover protection:** Replication

Key Differences

1) **Feature**
   - **Sharding:** Splits data into **smaller, separate databases** (shards).
   - **Replication:** Copies data from a **primary database** to multiple replicas.
2) **Scaling**
   - **Sharding:** Supports **horizontal scaling**
     - by distributing data across multiple database instances.
   - **Replication:** Improves **read scalability**
     - by allowing multiple replicas to handle read queries.
3) **Storage**
   - **Sharding:** Each shard stores **only a portion** of the data.
   - **Replication:** Each replica maintains a **full copy** of the database.
4) **Performance Improvement**
   - **Sharding:** Increases both **write** and **read** performance by distributing load.
   - **Replication:** Improves only **read performance**,
     - but **write operations are still handled by the primary database**.
5) **Fault Tolerance**
   - **Sharding:** Failure of a **single shard affects only a portion of data**,
     - may impact availability.
   - **Replication:** Failure of a **replica does not affect availability**,
     - another replica can take over.