

The background is a solid blue color. It is decorated with various white and light blue icons related to system design. In the top left, there is a stack of three server units. In the top center, a network switch or router is shown. To its right, a cloud icon contains three server units, with five arrows pointing outwards in different directions. Further right, another cloud icon is connected to two database cylinder icons. On the far right, a tall, multi-bay server rack is depicted. In the bottom left, there are two stacks of server units, each with green indicator lights. Above them, a cloud icon is connected to a central node. In the bottom center, there is a desktop computer setup with a monitor and a tower. To the right of the center, there are three circular icons containing smiley, neutral, and frowny faces, respectively. On the far right, there is a large, detailed database cylinder icon. The text 'System Design' is written in a large, white, sans-serif font with a yellow outline, underlined. Below it, 'Concepts' is written in a similar style. At the bottom, 'part - 1' is written in the same style.

System Design

Concepts

part - 1

Scalability.

- Scale to handling more number of requests
- Horizontal scaling
 - Adding nodes or instances
- Vertical scaling
 - More hardware resources in current machines

Data Partitioning

- Dividing large datasets into small manageable subsets
- To improve query performance, scalability and storage requirements.
- **Sharding**
 - Managing large databases into manageable subsets and hosting in different servers
- **Partitioning**
 - Dividing data into separate partition based on host , list and range.

Database Design

- Organize the data and create the relationship between them.
- **Normalization** - To reduce redundancy.
- **Denormalization** - To improve performance.
- **CAP Theorem**
 - To deliver any two characteristics from these three, consistency, availability, and partition tolerance.
- **ACID** - To ensure the sanctity of the data.
- **BASE** - To handle large volumes of data and high concurrency

Caching

- Process of storing data in temporary storage and can be accessed in an easy way
- Consistency
 - Data should remain updated and consistent.
- Cache Eviction
 - Follow the policies or algorithm to manage data in cache.
 - LRU, LFU, MRU, FIFO
- Distributed cache
 - Store cache data across multiple servers

Consistency Models

- All nodes in a distributed system should have the same data at any time.
- Strong Consistency
 - All nodes have the same data at the same time.
- Eventual Consistency
 - To ensure the updates are reflected in all nodes.
- Causal Consistency
 - To ensure the sequence of events.