Horizontal Partioning - partitioning in which some kind of key is used for allocating broken pieces to servers is now known as Horizontal Partioning.
Depends on one key
An example of a Key is a user ID

Vertical Partitioning uses columns to partition.

Read this -
https://stackoverflow.com/questions/18302773/what-are-horizontal-and-vertical-partitions-in-database-and-what-is-the-differen

Database Sharding - Technique that partitions the data across multiple servers, allowing for improved scalability and performance.
Consistency is important in Sharding.- changes in db should be consistent.
Availability - The database should not crash.

Sharding can be zoned on various Key -
User ID, Location, etc.

(Shards - Parts)
Problems with sharding -

1. Joins are highly expensive when Query requires joining across shards.
2. Fixed number of Shards - no flexibility. - (fixed by using hierarchical sharding - pieces are broken down into more pieces)

Best Practice is to use indexing in a particular shard for better performance.

If a shard fails,
let's say we have master-slave architecture (there are multiple slaves and a master - slaves are copying the master, wherever there is a written request, it is always on the master, the master is the most updated copy while slaves continuously pull the master, and read from it if a read request is from slaves,
If if master fails, slaves choose one master among themselves (good single point of failure tolerance here.)