1. 数据库的分类
2. 关系型数据库和非关系型数据库

关系型数据库：Oracle、MySQL、MariaDB、SqlServer、Access、PostgreSQl…

非关系型数据库：NoSQL

1. 键值存储数据库：Redis、Memcached、MemcacheDB

（简单、易部署、高并发；一个哈希表）

1. 文档存储数据库：MongoDB、CouchDB

（数据以文档形式存储；每个数据项都有一个名词与对应值，值既可以是简单的数据类型，如字符串、数字和日期等；也可以是复杂的类型，如有序列表和关联对象。数据存储的最小单位是文档，同一个表中存储的文档属性可以是不同的，数据可以使用XML、JSON或JSONB等多种形式存储）

1. 列存储数据库：HBase、Cassandra

（分布式存储海量数据，将数据存储在列族中，一个列族中经常被一起查询的相关数据）

1. 图数据库：Neo4J、InforGrid
2. 分布式数据库：https://blog.csdn.net/xuheng8600/article/details/79976043
3. Greenplum: 基于Hadoop，处理海量数据
4. Vertica：基于列存储
5. Sybase IQ：大量并发用户的即席查询
6. Teradata Aster Data

分布式是以缩短单个任务的执行时间来提升效率的，而集群则是通过提高单位时间内执行的任务数来提升效率。

1. 数据库的架构设计方案：底层是MySQL、Redis、MongoDB，架构设计上都是相通的
2. 数据库集群方案
3. 单数据中心的数据集群：根据集群中的完整业务数据是否集中在同一台机器上，区分为数据集中和数据分散模式

a.数据集中：类似于一主一从式，主机读和写，从机只读

（关注从机的一致性处理：ZooKeeper）

（业务数据量不大、读数据远远大于写操作、集群规模小的业务场景）

b.数据分散：完整业务数据由多台主机共同分担、分散存储。

（数据均衡分散到各个机上，保证资源的均衡使用和性能的最佳，需要相互的数据备份）

（大数据量、集群规模较大的场景）

1. 多中心的数据分区：容灾考虑，在多个地区部署多套数据集群，=》数据分区架构

分区，关注分区规则和备份规则

分区：根据一定的分区规则，每一个分区都存储一部分数据

冗余备份：每个区的数据进行备份

1. 集中备份：可以随意拓展分区（节点）
2. 独立备份：分区数据互不影响，拓展性好
3. 相互备份：多个节点相互、循环备份
4. 数据库相关
5. 事务的ACID

1)原子性(Atomic)：事务中各项操作，要么全做要么全不做，任何一项操作的失败都会导致整个事务的失败；

2)一致性(Consistent)：事务结束后系统状态是一致的；

3)隔离性(Isolated)：并发执行的事务彼此无法看到对方的中间状态；

4)持久性(Durable)：事务完成后所做的改动都会被持久化，即使发生灾难性的失败。通过日志和同步备份可以在故障发生后重建数据

1. 当前数据库
2. SQLite

一个数据库就是一个文件，在这个文件中可建立多张表，可以建立索引、触发器等。备份这个文件就备份了整个数据库；

**备份**SQLite**数据库**有两种方法。如果数据库没有处于使用状态，则可以直接将数据库文件复制到安全位置；如果数据库正在使用中，则应从命令行界面使用.dump命令(.dump命令也可以用于备份数据库表)，创建一个包含必要命令和数据的文件，从而重新创建数据库：

SQLite通常情况下用作一个网站的后台数据库可以很好的工作。但是如果你的网站的访问量大到你开始考虑采取分布式的数据库部署，那么你应当毫不犹豫的考虑用一个企业级的客户端/服务器数据库来替代SQLite。