**当前数据库分为关系型数据库和非关系型数据库**

**关系型数据库**

**关系型数据库：指采用了关系模型来组织数据的数据库。  
关系模型指的就是二维表格模型，而一个关系型数据库就是由二维表及其之间的联系所组成的一个数据组织。**

关系模型中常用的概念：  
关系：一张二维表，每个关系都具有一个关系名，也就是表名  
元组：二维表中的一行，在数据库中被称为记录  
属性：二维表中的一列，在数据库中被称为字段  
域：属性的取值范围，也就是数据库中某一列的取值限制  
关键字：一组可以唯一标识元组的属性，数据库中常称为主键，由一个或多个列组成  
关系模式：指对关系的描述。其格式为：关系名(属性1，属性2， ... ... ，属性N)，在数据库中成为表结构

**关系型数据库的优点：**

**1.容易理解：二维表结构是非常贴近逻辑世界的一个概念，关系模型相对网状、层次等其他模型来说更容易理解  
2.使用方便：通用的SQL语言使得操作关系型数据库非常方便  
3.易于维护：丰富的完整性(实体完整性、参照完整性和用户定义的完整性)大大减低了数据冗余和数据不一致的概率**

**关系型数据库存在的问题**

**1.网站的用户并发性非常高，往往达到每秒上万次读写请求，对于传统关系型数据库来说，硬盘I/O是一个很大的瓶颈  
2.网站每天产生的数据量是巨大的，对于关系型数据库来说，在一张包含海量数据的表中查询，效率是非常低的  
3.在基于web的结构当中，数据库是最难进行横向扩展的，当一个应用系统的用户量和访问量与日俱增的时候，数据库却没有办法像web server和app server那样简单的通过添加更多的硬件和服务节点来扩展性能和负载能力。当需要对数据库系统进行升级和扩展时，往往需要停机维护和数据迁移。  
4.性能欠佳：在关系型数据库中，导致性能欠佳的最主要原因是多表的关联查询，以及复杂的数据分析类型的复杂SQL报表查询。为了保证数据库的ACID特性，必须尽量按照其要求的范式进行设计，关系型数据库中的表都是存储一个格式化的数据结构。**

**数据库事务必须具备ACID特性，ACID分别是Atomic原子性，Consistency一致性，  
Isolation隔离性，Durability持久性。**

**当今十大主流的关系型数据库**

[**Oracle**](https://www.oracle.com/database/index.html)**，[Microsoft SQL Server](https://www.microsoft.com/en-us/sql-server/" \t "_blank)，[MySQL](https://www.mysql.com/" \t "_blank)，[PostgreSQL](https://www.postgresql.org/" \t "_blank)，[DB2](https://www.ibm.com/analytics/us/en/db2/" \t "_blank)，**[**Microsoft Access**](https://products.office.com/zh-cn/access)**，** [**SQLite**](https://www.sqlite.org/)**，[Teradata](https://www.teradata.com.cn/" \t "_blank)，[MariaDB](https://mariadb.org/" \t "_blank)(MySQL的一个分支)，[SAP](https://www.sap.com/" \t "_blank)**

**非关系型数据库**

**非关系型数据库：指非关系型的，分布式的，且一般不保证遵循ACID原则的数据存储系统。**

**非关系型数据库结构**

**非关系型数据库以键值对存储，且结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，不局限于固定的结构，可以减少一些时间和空间的开销。**

**优点**

**1.用户可以根据需要去添加自己需要的字段，为了获取用户的不同信息，不像关系型数据库中，要对多表进行关联查询。仅需要根据id取出相应的value就可以完成查询。  
2.适用于SNS(Social Networking Services)中，例如facebook，微博。系统的升级，功能的增加，往往意味着数据结构巨大变动，这一点关系型数据库难以应付，需要新的结构化数据存储。由于不可能用一种数据结构化存储应付所有的新的需求，因此，非关系型数据库严格上不是一种数据库，应该是一种数据结构化存储方法的集合。**

**不足：**

**只适合存储一些较为简单的数据，对于需要进行较复杂查询的数据，关系型数据库显的更为合适。不适合持久存储海量数据**

**非关系型数据库的分类**

**非关系型数据库都是针对某些特定的应用需求出现的，因此，对于该类应用，具有极高的性能。依据结构化方法以及应用场合的不同，主要分为以下几类：**

**面向高性能并发读写的key-value数据库：**

**key-value数据库的主要特点是具有极高的并发读写性能**  
Key-value数据库是一种以键值对存储数据的一种数据库，类似Java中的map。可以将整个数据库理解为一个大的map，每个键都会对应一个唯一的值。  
**主流代表为[Redis](https://redis.io/" \t "_blank)，** [**Amazon DynamoDB**](https://aws.amazon.com/cn/dynamodb/)**，** [**Memcached**](https://memcached.org/)**，**[**Microsoft Azure Cosmos DB**](https://azure.microsoft.com/en-us/services/cosmos-db/)**和[Hazelcast](https://hazelcast.com/" \t "_blank)**

**面向海量数据访问的面向文档数据库：**

**这类数据库的主要特点是在海量的数据中可以快速的查询数据**  
文档存储通常使用内部表示法，可以直接在应用程序中处理，主要是JSON。JSON文档也可以作为纯文本存储在键值存储或关系数据库系统中。  
**主流代表为[MongoDB](https://www.mongodb.com/" \t "_blank)，[Amazon DynamoDB](https://aws.amazon.com/cn/dynamodb/" \t "_blank)，[Couchbase](https://www.couchbase.com/" \t "_blank)，**[**Microsoft Azure Cosmos DB**](https://azure.microsoft.com/en-us/services/cosmos-db/)**和[CouchDB](https://couchdb.apache.org/" \t "_blank)**

**面向搜索数据内容的搜索引擎：**

**搜索引擎是专门用于搜索数据内容的NoSQL数据库管理系统。**  
主要是用于对海量数据进行近实时的处理和分析处理，可用于机器学习和数据挖掘  
**主流代表为**[**Elasticsearch**](https://www.elastic.co/products/elasticsearch)**，[Splunk](https://www.splunk.com/zh-hans_cn" \t "_blank)，[Solr](https://lucene.apache.org/solr/" \t "_blank)，[MarkLogic](https://www.marklogic.com/" \t "_blank)和[Sphinx](http://sphinxsearch.com/" \t "_blank)**

**面向可扩展性的分布式数据库：**

**这类数据库的主要特点是具有很强的可拓展性**  
普通的关系型数据库都是以行为单位来存储数据的，擅长以行为单位的读入处理，比如特定条件数据的获取。因此，关系型数据库也被成为面向行的数据库。相反，面向列的数据库是以列为单位来存储数据的，擅长以列为单位读入数据。  
**这类数据库想解决的问题就是传统数据库存在可扩展性上的缺陷，这类数据库可以适应数据量的增加以及数据结构的变化，将数据存储在记录中，能够容纳大量动态列。由于列名和记录键不是固定的，并且由于记录可能有数十亿列，因此可扩展性存储可以看作是二维键值存储。**  
**主流代表为[Cassandra](https://cassandra.apache.org/" \t "_blank)，[HBase](https://hbase.apache.org/" \t "_blank)，[Microsoft Azure Cosmos DB](https://azure.microsoft.com/en-us/services/cosmos-db/" \t "_blank)，**[**Datastax Enterprise**](https://www.datastax.com/products/datastax-enterprise)**和[Accumulo](https://accumulo.apache.org/" \t "_blank)**



**CAP理论**

**NoSQL的基本需求就是支持分布式存储，严格一致性与可用性需要互相取舍**  
CAP理论：**一个分布式系统不可能同时满足C(一致性)、A(可用性)、P(分区容错性)**三个基本需求，并且最多只能满足其中的两项。对于一个分布式系统来说，分区容错是基本需求，否则不能称之为分布式系统，因此需要在C和A之间寻求平衡  
**C(Consistency)一致性  
一致性是指更新操作成功并返回客户端完成后，所有节点在同一时间的数据完全一致。与ACID的C完全不同  
A(Availability)可用性  
可用性是指服务一直可用，而且是正常响应时间。  
P(Partition tolerance)分区容错性  
分区容错性是指分布式系统在遇到某节点或网络分区故障的时候，仍然能够对外提供满足一致性和可用性的服务。**

**关系型与非关系型数据库的比较**

**1.成本：Nosql数据库简单易部署，基本都是开源软件，不需要像使用Oracle那样花费大量成本购买使用，相比关系型数据库价格便宜。  
2.查询速度：Nosql数据库将数据存储于缓存之中，而且不需要经过SQL层的解析，关系型数据库将数据存储在硬盘中，自然查询速度远不及Nosql数据库。  
3.存储数据的格式：Nosql的存储格式是key,value形式、文档形式、图片形式等等，所以可以存储基础类型以及对象或者是集合等各种格式，而数据库则只支持基础类型。  
4.扩展性：关系型数据库有类似join这样的多表查询机制的限制导致扩展很艰难。Nosql基于键值对，数据之间没有耦合性，所以非常容易水平扩展。  
5.持久存储：Nosql不使用于持久存储，海量数据的持久存储，还是需要关系型数据库  
6.数据一致性：非关系型数据库一般强调的是数据最终一致性，不像关系型数据库一样强调数据的强一致性，从非关系型数据库中读到的有可能还是处于一个中间态的数据，  
Nosql不提供对事务的处理。**

作者：意识流丶  
链接：https://www.jianshu.com/p/fd7b422d5f93  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。