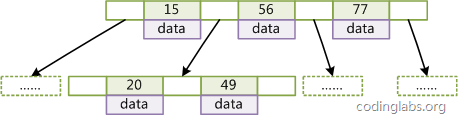
1. **关系型数据库**

本科所学即为关系型数据库，具体代表MySQL。

**B-Tree和B+Tree**

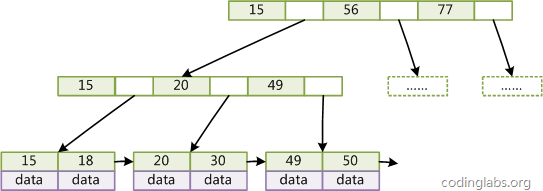
* B-Tree：平衡树



* B+Tree：内节点不存储data，只存储key；叶子节点不存储指针

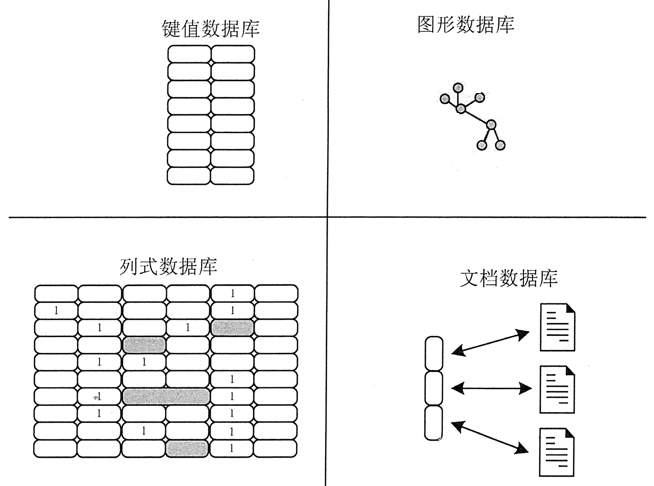


* 带有顺序访问指针的B+Tree：一般在数据库系统或文件系统中使用的B+Tree结构都在经典B+Tree的基础上进行了优化，增加了顺序访问指针(称为B\* tree)，即在B+Tree的每个叶子节点增加一个指向相邻叶子节点的指针，就形成了带有顺序访问指针的B+Tree。



* B+树的分裂：当一个结点满时，分配一个新的结点，并将原结点中1/2的数据复制到新结点，最后在父结点中增加新结点的指针；B+树的分裂只影响原结点和父结点，而不会影响兄弟结点，所以它不需要指向兄弟的指针。
* B\*树的分裂：当一个结点满时，如果它的下一个兄弟结点未满，那么将一部分数据移到兄弟结点中，再在原结点插入关键字，最后修改父结点中兄弟结点的关键字（因为兄弟结点的关键字范围改变了）；如果兄弟也满了，则在原结点与兄弟结点之间增加新结点，并各复制1/3的数据到新结点，最后在父结点增加新结点的指针。

1. **文档型数据库**



**基本概念：**

* MongoDB:文档型数据库，按照文档的形式存储。如XML文档、HTML文档和JSON 文档。
* 文档（数据模型）
* 用<key/attribute,value>存放，json的格式
* 一个文档里还可以包括子文档
* 每个文档都有个\_id，如果在一个文档中没有说\_id是什么，数据库会自动分配一个，可以通过\_id找到数据库的任何一个文档
* 一个文档可以是做一个对象（json）
* 文档集(collection)
* 文档的集合（一个文档集可以视为一类对象）
* 数据库(database)
* 若干个文档集构成一个数据库
* 部署在一个服务器上，【一般】一个数据库对应一个应用

| **Oracle** | **MongoDB** |
| --- | --- |
| 数据库实例（database instance） | MongoDB实例（MongoDB instance） |
| 模式（schema） | 数据库（database） |
| 表（table） | 集合（collection） |
| 行（row） | 文档（document） |
| 伪列（rowid） | \_id |
| join | DBRef |

**MongoDB优点：**

性能优越：快速！在适量级的内存的MongoDB的性能是非常迅速的，它将热数据存储在物理内存中，使得热数据的读写变得十分快。

高扩展：第三方支持丰富（这是与其他的No SQL相比，MongoDB也具有的优势）

自身的Failover机制！

弱一致性（最终一致），更能保证用户的访问速度

文档结构的存储方式，能够更便捷的获取数据：json的存储格式

支持大容量的存储，内置GridFS

内置Sharding

**MongoDB 缺点：**

主要是无事务机制

① MongoDB 不支持事务操作(最主要的缺点)

② MongoDB 占用空间过大

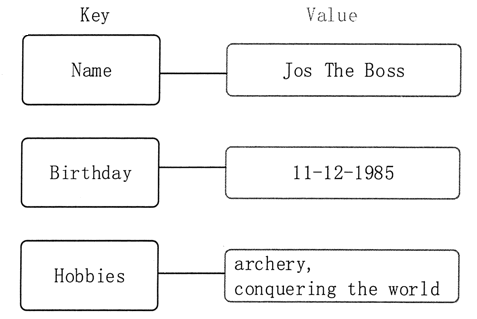
③ MongoDB 没有如 MySQL 那样成熟的维护工具，这对于开发和IT运营都是个值得注意的地方

**适用案例：**

* 事件记录，应用程序对事件记录各有需求。
* 内容管理系统及博客平台，用来管理用户评论、用户注册、用户配置和面向Web文档（web-facing document）。
* 网站分析与实时分析，用来存储“页面浏览量”（page view）或“独立访客数”（unique visitor）会非常方便，而且可以无需改变模式即可新增度量标准。
* 电子商务应用程序，以存储产品和订单。

MongoDB最常见的用例包括单视图，物联网，移动，实时分析，个性化，目录和内容管理。

1. **键值（KV）数据库**



**几个概念：**

* 关联数组：和普通数组一样的结构，区别在于没有普通数组一样的约束或者说规范

（1）key（下标）不限于整数，可以是字符串

（2）value可以是实数、字符串、列表及整数等类型

（2）key和value都不要求是同一类型

* 命名空间：由键值对构成的集合，相当于数据库或桶
* 分区：根据键名，把数据分割成不同的单元，存储在集群中的不同服务器上，实现负载均衡

**键和值介绍：**

* 键的定义：

键最好以命名空间为前缀，这样确保了它的唯一性；可以采用实体+标识符+属性的方式，以冒号隔开。如 customer:1:name

分区：可以按键名分区；也可以按hash分区

* 值的介绍：

值可以是任何类型，对于值的约束取决于具体使用的kv数据库

* 注意：

kv数据库只支持根据键查找值，一般不支持查询语言，有的kv数据库提供文本搜索功能，如Riak

**数据库特点：**

1.简洁

用到的只是增加和删除，不需要设计复杂的数据模型、纲要，也不需要为每个属性指定数据类型。动态添加时不需要修改原有数据库的定义

2.高速

把数据不保存在内存中，在RAM中读取和写入速度要快很多，当然也可以选择持久化

因为是存在内存中，有时需要释放来存储新的数据，最常用的算法：LRU(Least Recently Used，最久未使用算法)

3.易于缩放

可缩放性：根据系统负载量，随时添加或删除服务器

缩放的两种方式：

（1）主从式复制

主服务器处理写入请求和读取请求，并把珠吉路的数据复制到集群中的其他服务器里。从服务器只响应读取请求，遵从树形结构

应用情况：希望服务器能快速相应查询请求，而对写入请求响应能力没有太多要求

优点：简洁，其他服务器只需与主服务器通信，而且不用协调写入冲突

缺点：主服务器作为独木桥，一旦故障，整个集群无法处理写入请求（解决：从服务器遵守协议，若主服务器故障，一台从服务器升为主服务器）

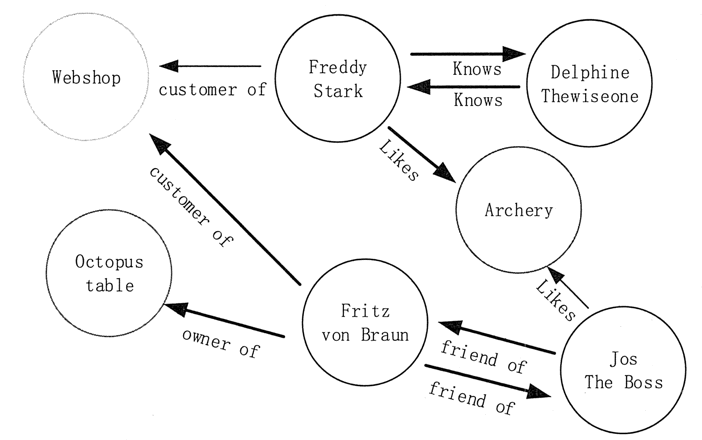
（2）无主式复制

对于诸如网上购票这种需要大量读取和写入请求的情况，主从不适合，采用无主式复制，它遵从一种环形或网状结构

这种情况下每台服务器都可以把自己活得新数据复制到其他服务器里面，可以指定哪几台作为它的副本存储服务器。

1. **图形数据库**

图形或网络数据主要由结点和边两部分组成。结点是实体本身。边代表两个实体之间的关系，用线来表示，并具有自己的属性。另外，边还可以有方向，如果箭头指向谁，谁就是该关系的主导方，如图所示。



**各大开源图数据库优势**

