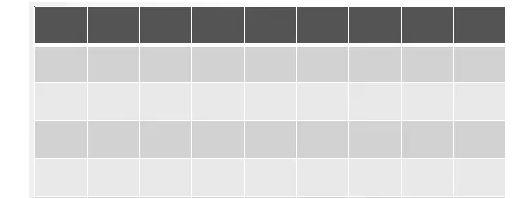
HBase：分布式的nosql数据库，是key-value存储方式，非常适合存储海量数据。

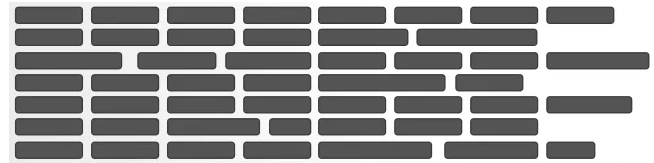
HBase比Redis更像mysql数据库。Redis的数据在内存中，所以它的访问速度比HBase快，但存储大量数据时会贵得多，因为内存贵。HBase的数据存储在HDFS上，HDFS是一个分布式文件系统，HDFS比mysql快，mysql的数据直接存在硬盘上。可以把HBase看成mysql，HDFS就是硬盘。

HBase是一个列式存储。

传统关系型数据库如mysql都是行式存储，类似一个表格，每行是一条数据，列则代表属性，如果有数据没有某个属性，对应的位置就空缺(前提是允许属性为null)，这样会造成一定的空间浪费。



HBase则类似于下面这样：

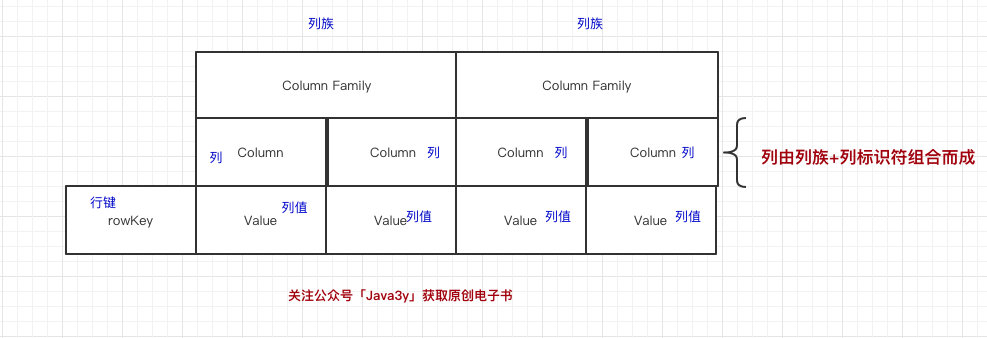


类似一个稀疏矩阵，每个黑块就是一条数据。

HBase的数据结构：

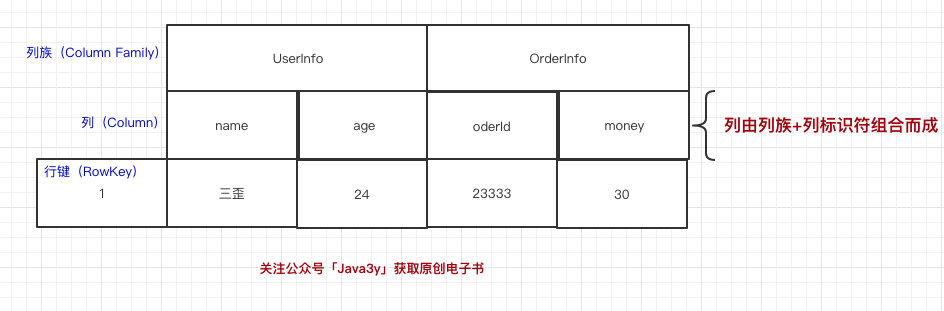
HBase是key-value结构，其中充当传统意义上key的称为RowKey，它也类似mysql中的Primary Key，最好不要有重复的。而value则存储在列中。HBase中每一行中，列是非常灵活的，行与行并不遵循相同的列定义。

HBase有列族的概念，每隔列都必须有列族。列族可以理解为列的属性类别。



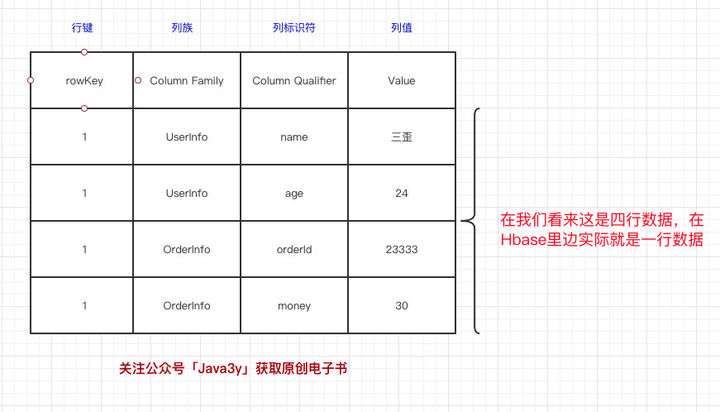
列标识符就是列的名称。

实际的数据：

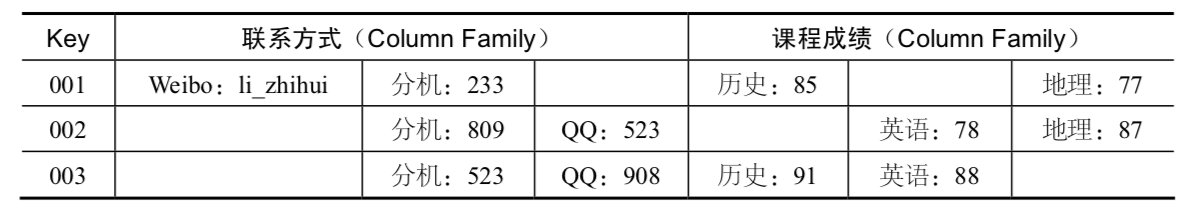


其中有两个列族，UserInfo和OrderInfo。

上面的数据对应到mysql中类似下面这种情况：



行与行之间不一定是相同的列：



HBase在创建表时必须指定列族，可以有多个。HBase官方建议列族要小于或等于3。在HBase中不同的表可以有相同的列族，但这些列族不会集中存储，因为表和表时隔离的。

HBase的数据修改和删除：

HBase对数据进行修改和删除时，不会对原数据进行操作，这样太慢，而是插入一条新的数据，数据插入到HBase时会记录一个时间戳，相当于一个版本号。例，原始数据：

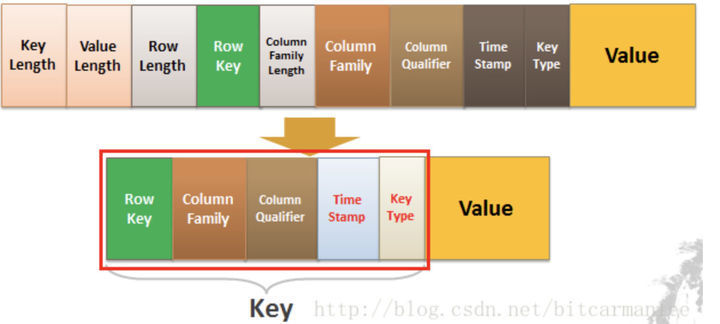


修改列值：



HBase中的Key-Value：

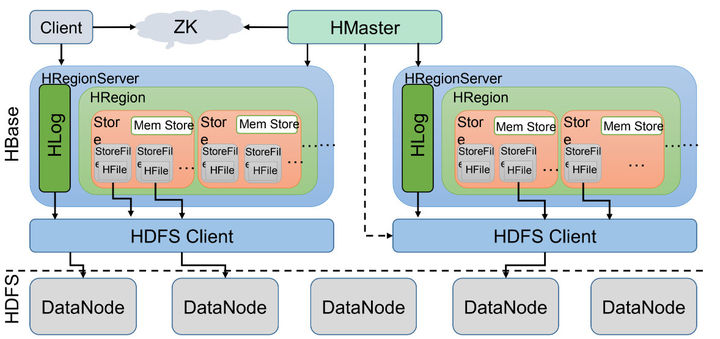
HBase中如果映射到类似dict那样通过key直接访问到最终的值，Key就不是只包含Row Key了，而是还要包含一些其它的内容：



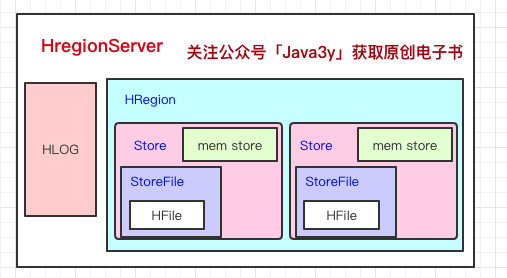
可以看出Key包含Row Key，Column Family，Column名称，Time Stamp，Key Type。

其中Key Type是标识put，delete这样的操作类型。

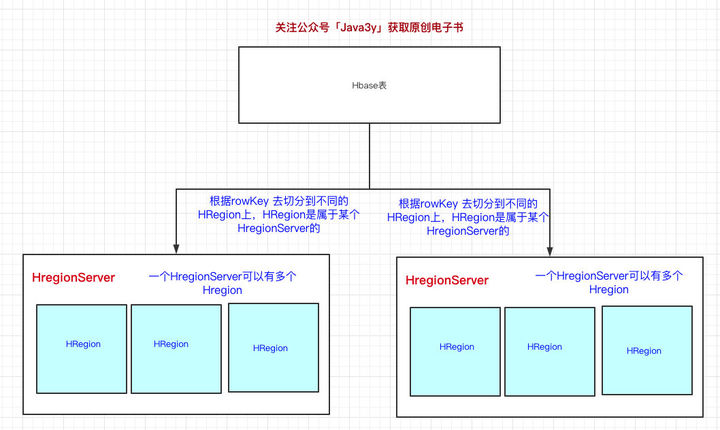
HBase的架构：



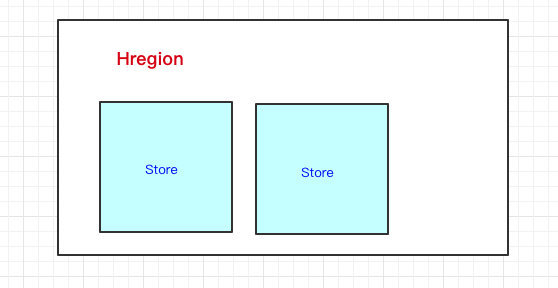
1. HRegionServer：处理客户端的读写请求，与HDFS底层交互，是真正干活的结点



HBase是分布式数据库，用来存储海量的数据，所以一张表肯定不会只存储在一台服务器上，那就需要对表进行切割，HBase就是按照Row Key进行切割表的。

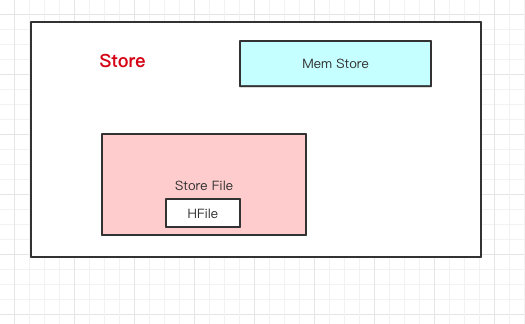


从上图看出，一个HRegion就存储了HBase表的一部分。



HRegion下面还有Store，它是用来存储列族的。HBase表必须指定列族，可以有多个，列族之间数据的差异性可能会比较大，所以要分开存储，一个列族就存储一个Store中。所以HBase其实是基于列族存储的。

Store里面有Mem Store，Store File，HFile。



HBase写数据时，并不会直接写到最终的文件里，这样效率太低，而是会先写到内存Mem Store中，当Mem Store超过一定的阈值时，就会将内存中的数据形成Store File，而Store File的底层就HFile，所以HFile就是HBase中Key-Value的存储格式。HFile是HDFS系统的二进制格式文件。

HRegionServer中的HLog就是WAL的模块，写数据到Mem Store时先写一条log到硬盘上，这条Log是append方式，速度非常快，防止数据丢失，一台HRegionServer只有一个WAL文件，所有的Region以及他们包含的Store File都共享这个WAL文件。

HBase和BigTable类似，内存Mem Store也是一个有序跳跃表，采用的是LSM(Log-Structured-Merge)的三层存储方式，先存储在内存中并添加Log日志防止内存挂掉后导致数据丢失，然后再存入到结构化的文件到对应的文件系统，满足一定条件后，把小的文件合并成大的文件。

1. ZK(ZooKeeper)：分布式锁服务器，
2. 提供加锁服务，客户端可以向它请求锁并执行某些操作以避免形成资源竞争，
3. 存储meta元数据，告诉客户端去哪台机器上进行读写操作。
4. 如果HMaster宕机，要从后备的Master结点选一台出来充当HMaster。

HMaster：Master结点，分配HRegion到HRegionServer中，如果某些HRegion失效(例如HRegionServer挂了)，会重新分配这些失效的HRegion到其它的HRegionServer中。

上述的内容是完全分布式的HBase，项目中的HBase是独立结点，没有分布式。如果希望部署完全分布式的HBase，要先部署完全分布式的Hadoop，然后配置HMaster，ZooKeeper，HRegionServer等结点。

HBase的缺点：

对于传统 RDBMS 来说，支持 ACID 事务是数据库的基本能力，而 HBase 则使用行级锁来保证写操作的原子性，但是不支持多行写操作的事务性，这主要是从灵活性和扩展性上做出的权衡。

ACID：原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）以及持久性（Durability）

所以HBase一般不适用OLTP(联机事务处理应用)系统，就是类似银行转账等业务的，这类应用对事务要求比较高，而适合OLAP(联机分析处理应用)系统，比如推荐系统，是在收集了大量用户行为后进行分析，再得出结论的应用，主要侧重分析，对事务要求非常低。