

HBase

- 以region为最小单位，由若干个行组成。由于行是排序的，所以可以包含一个范围的行。
- 数据可以按列进行压缩。
- Schema不严格，一行可以没有某些属性。看起来是个稀疏的表格，但是在存储的是不会存储那些不存在的属性。
- 表格的行根据主键排序（字典序）。所以行的主键的设计很重要，比如网站的url，应该逆序存储。保证尽可能把相近的内容存储在同一块区域。
- 列族，按列存，对于动态加入列族非常友好。
- 内嵌zookeeper，无需自己起一个。
- 如果一行太大，可以存到列族里。列族按顺序存储若干行，但是列族间是独立的。所以列族不能太多，太多的话索引会比较多，影响效率。
- 关系型数据库的表存的是数据结构相似的，而HBase的表存的是语义上相关的数据，对结构要求不严格。

Column Qualifier（列的限定符）

列的限定符是列族中数据的索引。例如给定了一个列族 `content`，那么限定符可能是 `content:html`，也可以是 `content:pdf`。列族在创建表格时是确定的了，但是列的限定符是动态地并且行与行之间的差别也可能是非常大的。

HBase 表中的每个列都归属于某个列族，列族必须作为表模式(schema)定义的一部分预先给出。如 `create 'test', 'course'`;

列名以列族作为前缀，每个“列族”都可以有多个列成员(column); 如 `course:math`, `course:english`，新的列族成员（列）可以随后按需、动态加入；

权限控制、存储以及调优都是在列族层面进行的；

<https://blog.csdn.net/wyqwilliam>

- Cell是有版本的，默认返回最新的数据。

Cell（单元）

由行和列的坐标交叉决定；

单元格是有版本的；

单元格的内容是未解析的字节数组；

单元格是由行、列族、列限定符、值和代表值版本的时间戳组成的（{row key, column(=<family> +<qualifier>), version}）唯一确定单元格。cell中的数据是没有类型的，全部是字节码形式存储。

<https://blog.csdn.net/wyqwilliam>

Timestamp（时间戳）

时间戳是写在值旁边的一个用于区分值的版本的数据。默认情况下，时间戳表示的是当数据写入时 `RegionServer` 的时间点，但你也可以在写入数据时指定一个不同的时间戳。

在 HBase 每个 cell 存储单元对同一份数据有多个版本，根据唯一的时间戳来区分每个版本之间的差异，不同版本的数据按照时间倒序排序，最新的数据版本排在最前面。

时间戳的类型是 64 位整型。时间戳可以由 HBase(在数据写入时自动)赋值，此时时间戳是精确到毫秒的当前系统时间。时间戳也可以由客户显式赋值，如果应用程序要避免数据版本冲突，就必须自己生成具有唯一性的时间戳。

<https://blog.csdn.net/wyqwilliam>

- 版本理论和逻辑上可以无限叠加。

- 时间戳可以用户自定义，（但是要记住默认返回的时间戳最大的版本，所以如果是自定义的话可能返回的不是最新的版本），以此来节省空间（时间戳是长整型）。而且自定义逻辑时间戳可以更好发现两个版本之间存在一个版本的情况，而默认时间戳无法做到这一点。

Schema rules

- Region应该比较大。
- Cell不应该太大，如果太大可以考虑分离存储。（存一个pointer）
- 一个table不应该有太多的列族。
- 一个有一两个列族的表应该有50-100个region（注意到列族是按列存储的）
- 行的主键不应该太大。
- 列族在不同的地方存储，使用一个列族的时候不用加载其他列族。