

## 2 重要概念

influxdb里面有一些重要概念: database, timestamp, field key, field value, field set, tag key, tag value, tag set, measurement, retention policy, series, point。结合下面的例子数据来说明这几个概念:

```
1  name: census
2  -----
3  time                butterflies    honeybees    location    scientist
4  2015-08-18T00:00:00Z    12              23           1      langstroth
5  2015-08-18T00:00:00Z    1              30           1      perpetua
6  2015-08-18T00:06:00Z    11             28           1      langstroth
7  2015-08-18T00:06:00Z    3              28           1      perpetua
8  2015-08-18T05:54:00Z    2              11           2      langstroth
9  2015-08-18T06:00:00Z    1              10           2      langstroth
10 2015-08-18T06:06:00Z    8              23           2      perpetua
11 2015-08-18T06:12:00Z    7              22           2      perpetua
```

### timestamp

既然是时间序列数据库, influxdb的数据都有一列名为time的列, 里面存储UTC时间戳。

### field key, field value, field set

butterflies和honeybees两列数据称为字段(fields), influxdb的字段由field key和field value组成。其中butterflies和honeybees为field key, 它们为string类型, 用于存储元数据。

而butterflies这一列的数据12-7为butterflies的field value, 同理, honeybees这一列的23-22为honeybees的field value。field value可以为string, float, integer或boolean类型。field value通常都是与时间关联的。

field key和field value对组成的集合称之为field set。如下:

```
1  butterflies = 12 honeybees = 23
2  butterflies = 1 honeybees = 30
3  butterflies = 11 honeybees = 28
4  butterflies = 3 honeybees = 28
5  butterflies = 2 honeybees = 11
6  butterflies = 1 honeybees = 10
7  butterflies = 8 honeybees = 23
8  butterflies = 7 honeybees = 22
```

在influxdb中，字段必须存在。注意，字段是没有索引的。如果使用字段作为查询条件，会扫描符合查询条件的所有字段值，性能不及tag。类比一下，fields相当于SQL的没有索引的列。

## tag key, tag value, tag set

location和scientist这两列称为标签(tags)，标签由tag key和tag value组成。location这个tag key有两个tag value：1和2，scientist有两个tag value：langstroth和perpetua。tag key和tag value对组成了tag set，示例中的tag set如下：

```
1 | location = 1, scientist = langstroth
2 | location = 2, scientist = langstroth
3 | location = 1, scientist = perpetua
4 | location = 2, scientist = perpetua
```

tags是可选的，但是强烈建议你用上它，因为tag是有索引的，tags相当于SQL中的有索引的列。tag value只能是string类型 如果你的常用场景是根据butterflies和honeybees来查询，那么你可以将这两个列设置为tag，而其他两列设置为field，tag和field依据具体查询需求来定。

## measurement

measurement是fields，tags以及time列的容器，measurement的名字用于描述存储在其中的字段数据，类似mysql的表名。如上面例子中的measurement为census。measurement相当于SQL中的表，本文中我在部分地方会用表来指代measurement。

## retention policy

retention policy指数据保留策略，示例数据中的retention policy为默认的autogen。它表示数据一直保留永不过期，副本数量为1。你也可以指定数据的保留时间，如30天。

## series

series是共享同一个retention policy，measurement以及tag set的数据集合。示例中数据有4个series，如下：

| Arbitrary series number | Retention policy | Measurement | Tag set                             |
|-------------------------|------------------|-------------|-------------------------------------|
| series 1                | autogen          | census      | location = 1,scientist = langstroth |
|                         |                  |             | location = 2 scientist =            |

|          |         |        |                                     |
|----------|---------|--------|-------------------------------------|
| series 2 | autogen | census | location = 2,scientist = langstroth |
| series 3 | autogen | census | location = 1,scientist = perpetua   |
| series 4 | autogen | census | location = 2,scientist = perpetua   |

## point

point则是同一个series中具有相同时间的field set, points相当于SQL中的数据行。如下面就是一个point:

```

1 | name: census
2 | -----
3 | time           butterflies  honeybees  location  scientist
4 | 2015-08-18T00:00:00Z      1         30         1        perpetua

```

## database

上面提到的结构都存储在数据库中, 示例的数据库为my\_database。一个数据库可以有多个measurement, retention policy, continuous queries以及user。influxdb是一个无模式的数据库, 可以很容易的添加新的measurement, tags, fields等。而它的操作却和传统的数据库一样, 可以使用类SQL语言查询和修改数据。

influxdb不是一个完整的CRUD数据库, 它更像是一个CR-ud数据库。它优先考虑的是增加和读取数据而不是更新和删除数据的性能, 而且它阻止了某些更新和删除行为使得创建和读取数据更加高效。