# 全局唯一id

分库分表之后

*全局唯一Id 四种算法*

*1 交替递增法*

*奇数、偶数*

***局限：2张表***

***2 uuid***

***缺点：字母、太长***

***3 redis incr***

***自增键唯一性***

***4 雪花算法***

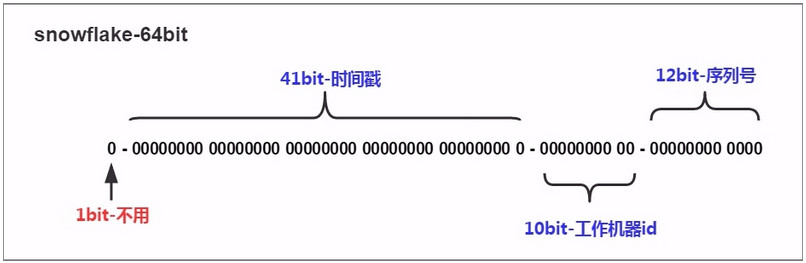
***Snowflake***

***雪花算法的原始版本是scala版，用于生成分布式ID（纯数字，时间顺序）,订单编号等。***

***自增ID：对于数据敏感场景不宜使用，且不适合于分布式场景。***

***GUID：采用无意义字符串，数据量增大时造成访问过慢，且不宜排序。***

***算法描述：***



***最高位是符号位，始终为0，不可用。***

***41位的时间序列，精确到毫秒级，41位的长度可以使用69年。时间位还有一个很重要的作用是可以根据时间进行排序。***

***10位的机器标识，10位的长度最多支持部署1024个节点。***

***12位的计数序列号，序列号即一系列的自增id，可以支持同一节点同一毫秒生成多个ID序号，12位的计数序列号支持每个节点每毫秒产生4096个ID序号。***

***Twitter发明的算法！！！！***

***我们就用的雪花算法。***

# 分库分表

**垂直拆分：**

**根据业务拆分：分库！！！！！！！！！！**

**水平拆分：**

**同一业务：同一张表拆成多张表！！！！！**