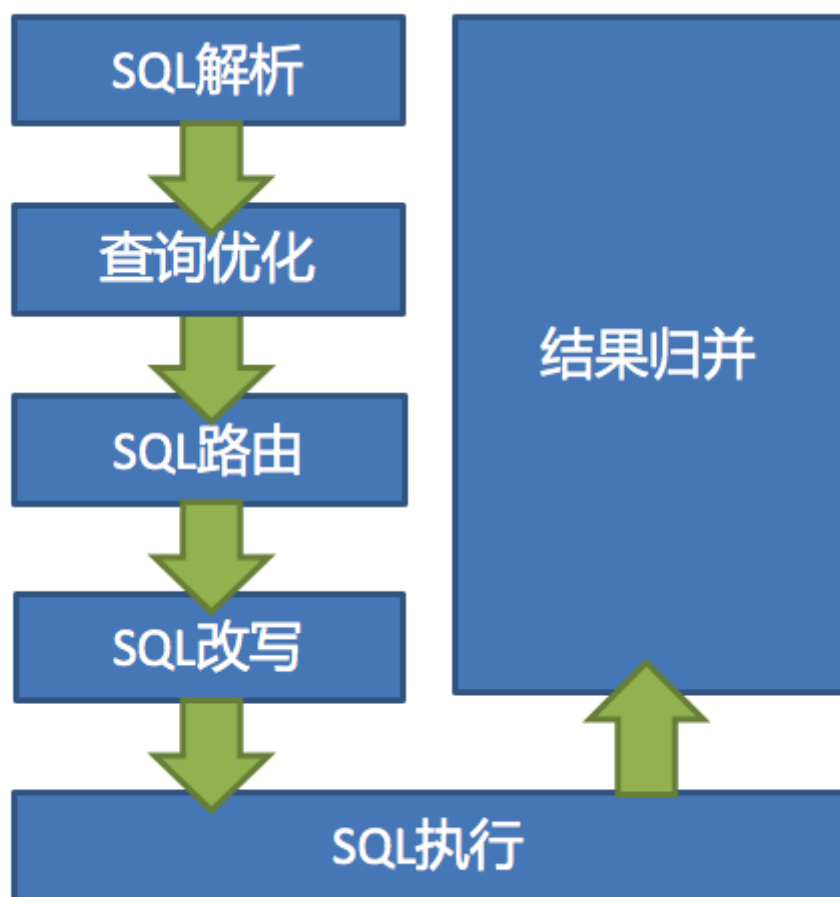


Apache孵化版-模块划分

| 名称 | 作用 |
|---------------------------|--|
| sharding-core | sharding内核模块；定义了核心api，SQL解析，SQL重写，SQL路由，spi，引擎等等核心功能 |
| sharding-distribution | 部署、运维相关zip包中的代码 |
| sharding-integration-test | 整合测试 |
| sharding-jdbc | app-分库分表jdbc增强 |
| sharding-opentracing | 应用性能监控 |
| sharding-orchestration | 数据库编排治理 |
| sharding-proxy | 服务端代理分库分表模块 |
| sharding-spring | 与spring集成 |
| sharding-sql-test | SQL测试用例 |
| sharding-transaction | 分布式事务 |

内核剖析

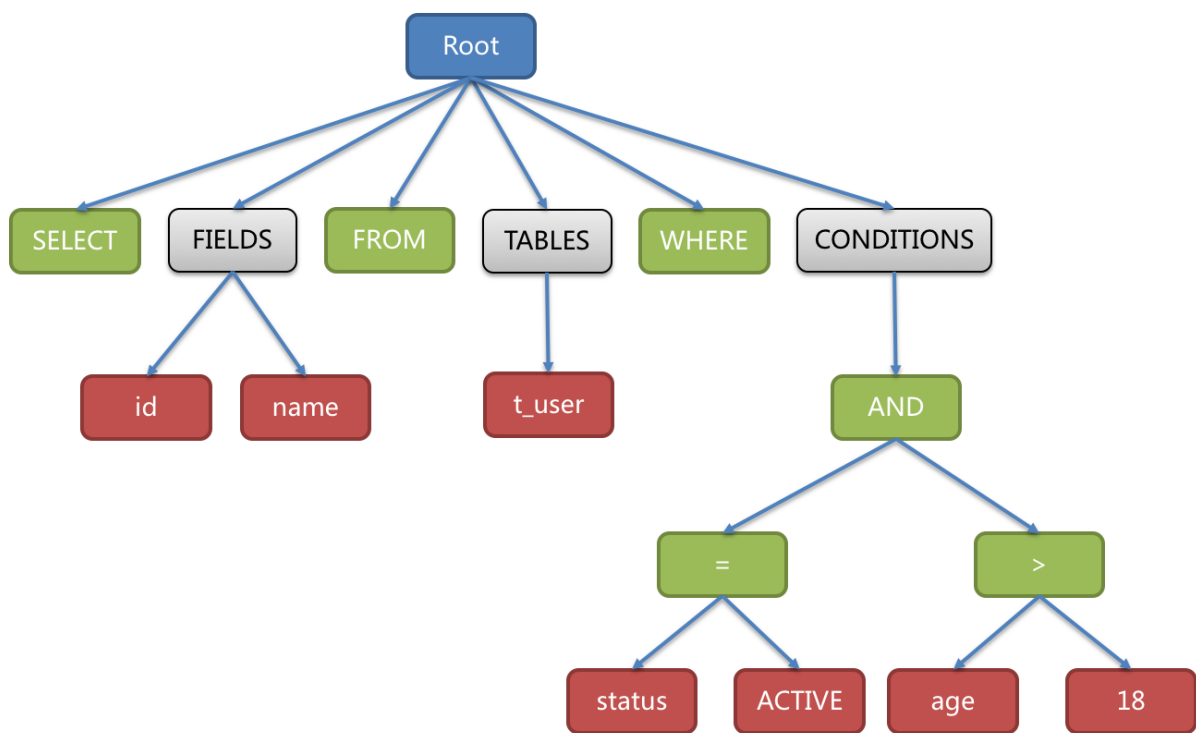
数据分片-执行过程



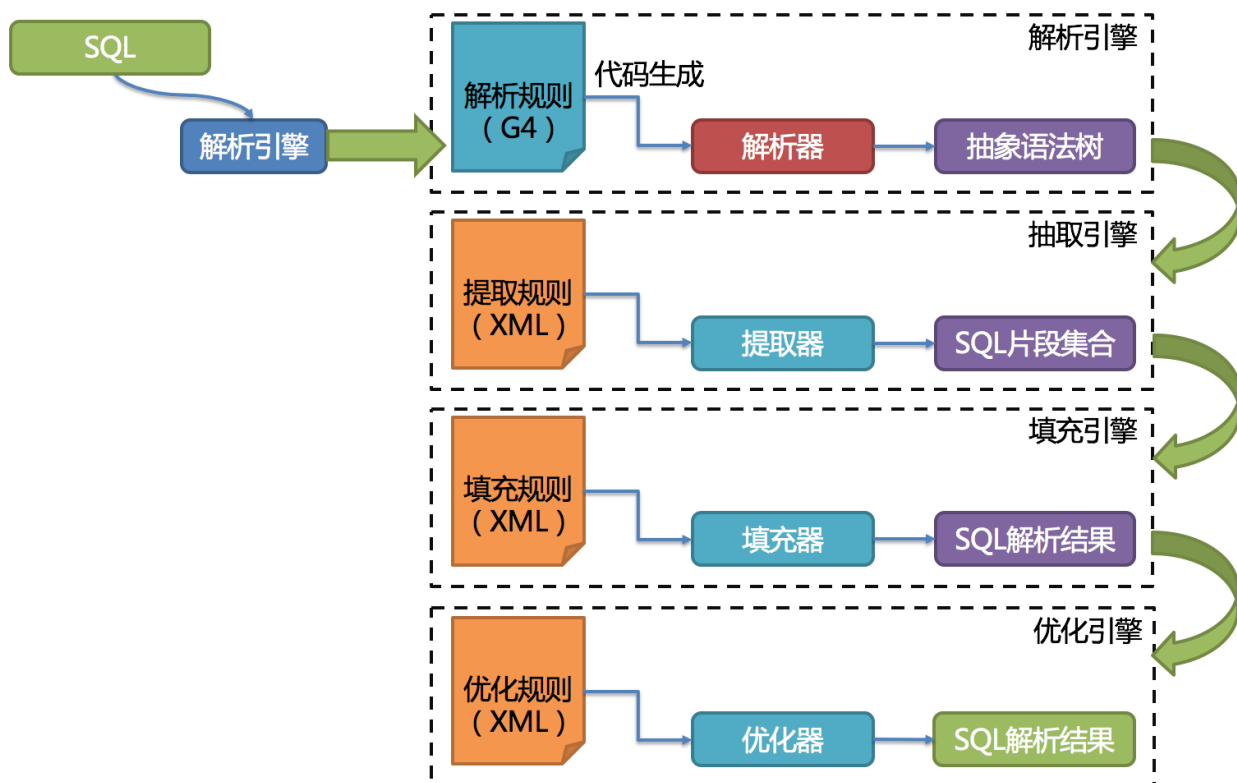
解析引擎

解析过程分为词法解析和语法解析。词法解析器用于将SQL拆解为不可再分的原子符号，称为Token。并根据不同数据库方言所提供的字典，将其归类为关键字，表达式，字面量和操作符。再使用语法解析器将SQL转换为抽象语法树。

```
SELECT id, name FROM t_user WHERE status = 'ACTIVE' AND age > 18
```

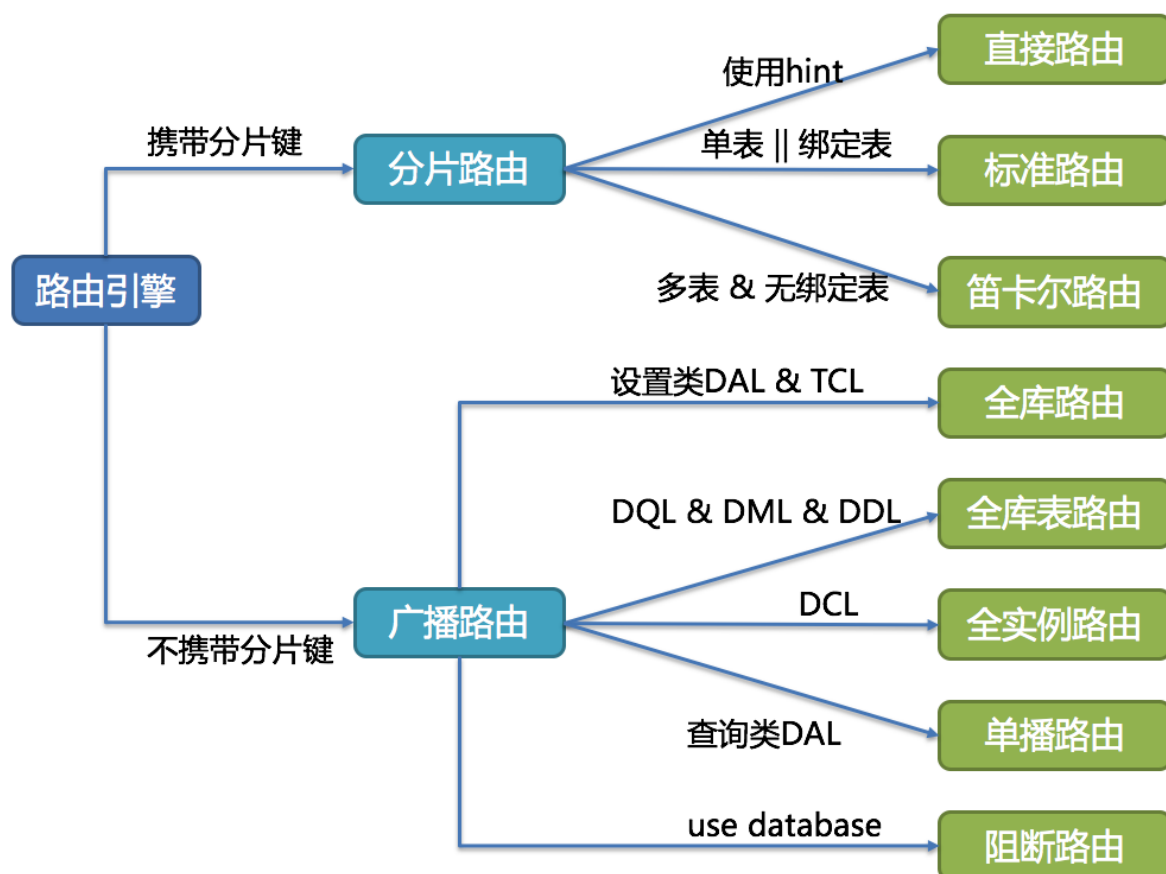


sql解析整体结构：



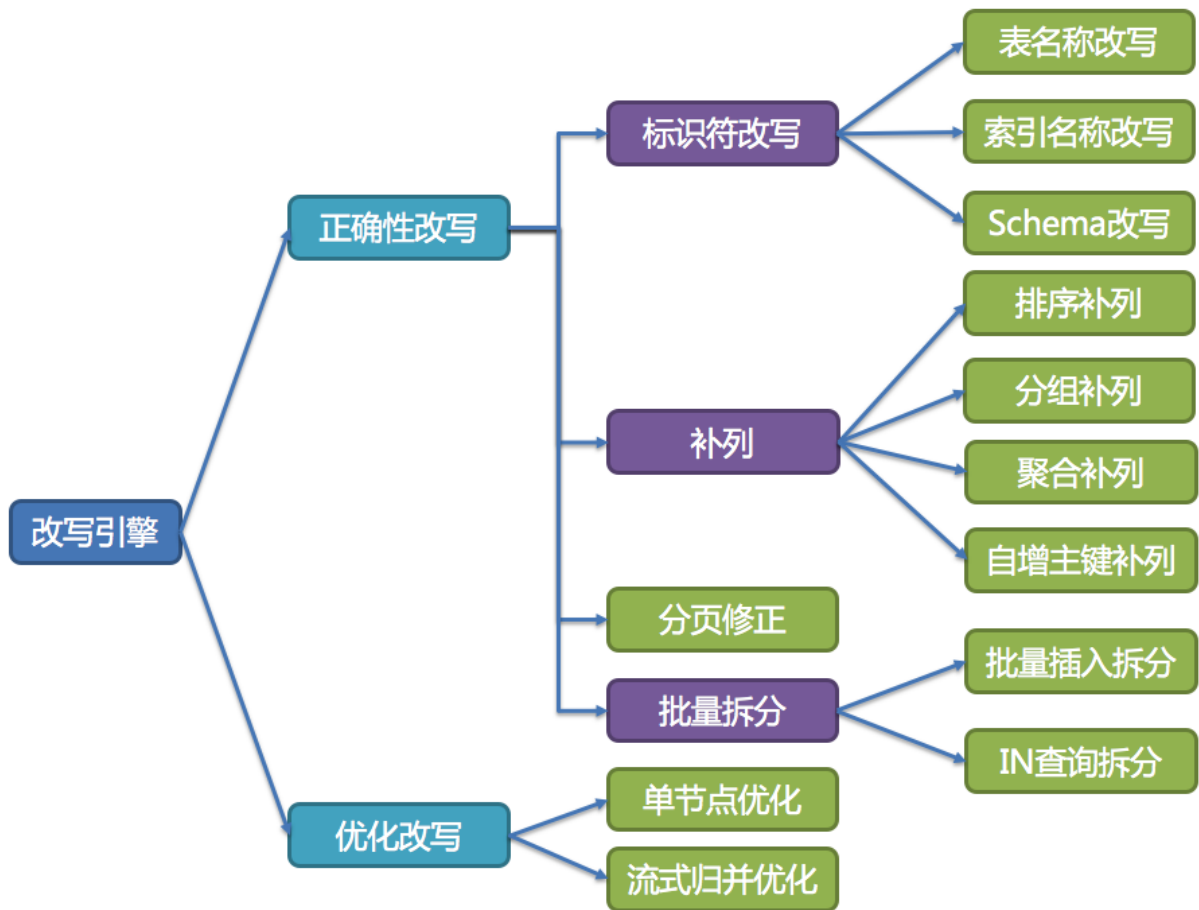
路由引擎

内置的分片策略大致可分为尾数取模、哈希、范围、标签、时间等。由用户方配置的分片策略则更加灵活，可以根据使用方需求定制复合分片策略。

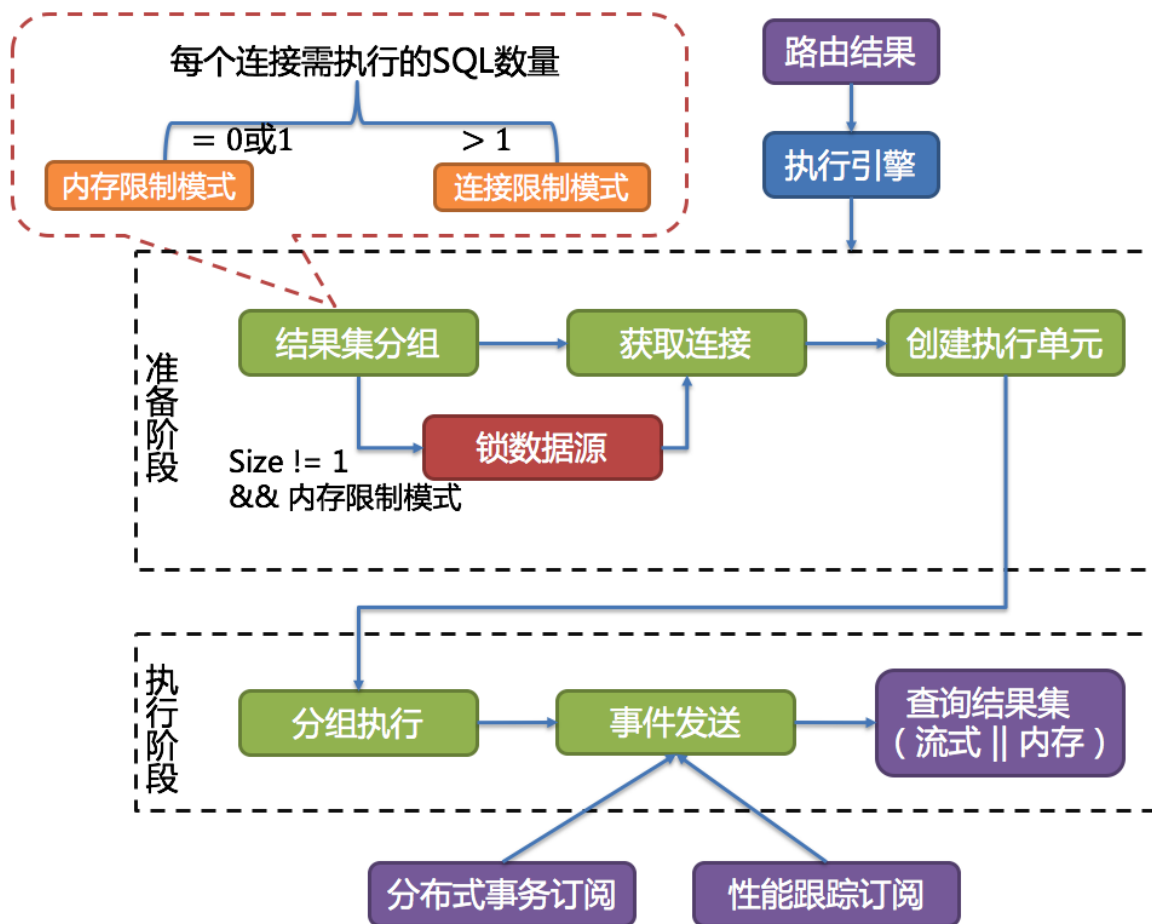


SQL改写

在分表的场景中，需要将分表配置中的逻辑表名称改写为路由之后所获取的真实表名称。

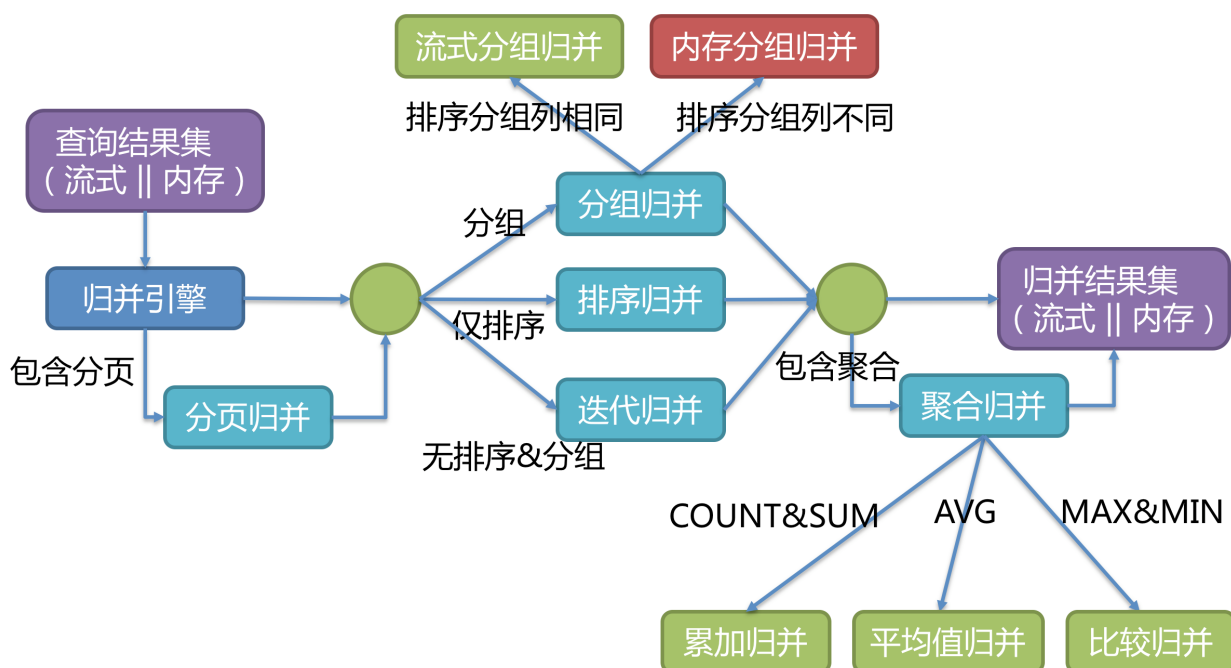


执行引擎



结果归并

将从各个数据节点获取的多数据结果集，组合成为一个结果集并正确的返回至请求客户端，称为结果归并。



分布式主键

内置生成器支持：UUID、SNOWFLAKE、LEAF(进行中)，并抽离出分布式主键生成器的接口，方便用户自行实现自定义的自增主键生成器。

SNOWFLAKE

雪花算法,能够保证不同进程主键的不重复性，相同进程主键的有序性。二进制形式包含4部分，从高位到低位分表为：1bit符号位、41bit时间戳位、10bit工作进程位以及12bit序列号位。

- 符号位(1bit)

预留的符号位，恒为零。

- 时间戳位(41bit)

41位的时间戳可以容纳的毫秒数是2的41次幂，一年所使用的毫秒数是： $365 * 24 * 60 * 60 * 1000$

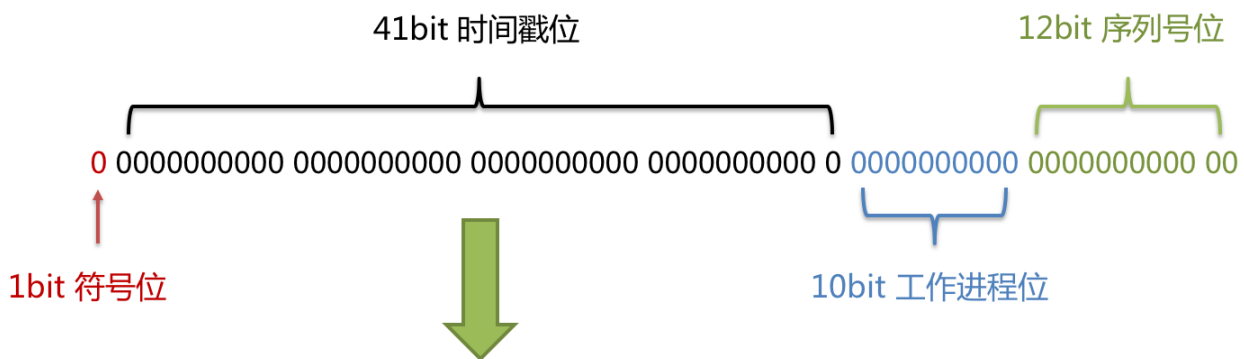
$\text{Math.pow}(2, 41) / (365 * 24 * 60 * 60 * 1000L) = 69.73$ 年不重复；

- 工作进程位(10bit)

该标志在Java进程内是唯一的，如果是分布式应用部署应保证每个工作进程的id是不同的。该值默认为0，可通过属性设置。

- 序列号位(12bit)

该序列是用来在同一个毫秒内生成不同的ID。如果在这个毫秒内生成的数量超过4096(2的12次幂)，那么生成器会等待到下个毫秒继续生成。

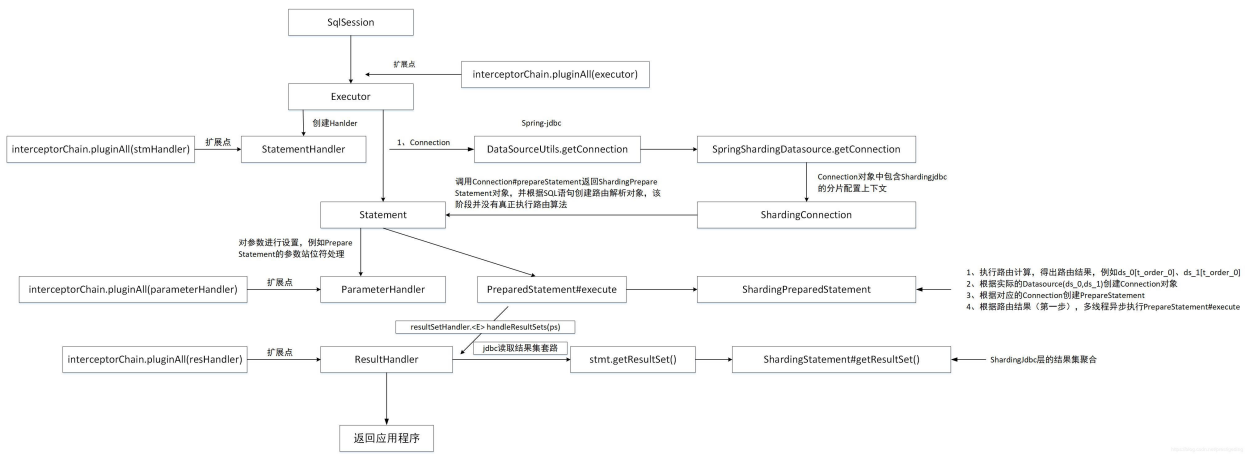


时间范围： $2^{41} / (365 * 24 * 60 * 60 * 1000L) = 69.73$ 年
工作进程数量： $2^{10} = 1024$
生成不碰撞序列的TPS： $2^{12} * 1000 = 409.6$ 万

如何自定义分布式主键？

(课堂问题)

Mybatis-SQL执行过程



Copyright © 2014-2015 Alibaba Group Holding Limited.