ShardingSphere内置提供了多种分片算法，按照类型可以划分为自动分片算法、标准分片算法、复合分片算法和Hint分片算法，能够满足用户绝大多数业务场景的需要。此外，考虑到业务场景的复杂性，内置算法也提供了自定义分片算法的方式，用户可以通过编写Java代码来完成复杂的分片逻辑。

1. 标准分片算法
2. 行表达式分片算法

使用Groovy的表达式，提供对SQL语句中的=和IN的分片操作支持，只支持单分片键。对于简单的分片算法，可以通过简单的配置使用，从而避免繁琐的Java代码开发。

1. 时间范围分片算法

此算法主动忽视了datetime-pattern的时区信息。这意味着当datetime-lower, datetime-upper和传入的分片键含有时区信息时，不会因为时区不一致而发生时区转换。当传入的分片键为java.time.Instant时存在特例处理，其会携带上系统的时区信息后转化为datetime-pattern的字符串格式，再进行下一步分片。

1. 基于CosId的固定时间范围的分片算法

基于me.ahoo.cosid:cosid-core的工具类实现的固定时间范围的分片算法。 当分片键为JSR-310的所含类或与时间相关的类，将转换为java.time.LocalDateTime后再做下一步分片。

1. 基于CosId的雪花ID固定时间范围的分片算法

基于me.ahoo.cosid:cosid-core的工具类实现的雪花ID固定时间范围的分片算法。当分片键为JSR-310的所含类或与时间相关的类，将转换为java.time.LocalDateTime后再做下一步分片。

1. 基于CosId的取模分片算法

基于me.ahoo.cosid:cosid-core的工具类实现的取模分片算法。

1. 复合分片算法
2. 复合行表达式分片算法

类似行表达式分片算法，后续会进行详细讲解。

1. Hint分片算法
2. Hint行表达式分片算法

类似行表达式分片算法，后续会进行详细讲解。

1. 自定义类分片算法

通过配置分片策略类型和算法类名，实现自定义扩展。CLASS\_BASED允许向算法类内传入额外的自定义属性，传入的属性可以通过属性名为props的java.util.Properties类实例取出。