

JdbcExecutor

`JdbcExecutor` 是一个用于执行 SQL 查询的组件，特别是针对通过 JDBC 连接到数据库的查询。它实现了 `QueryExecutor` 接口，并提供了具体的执行逻辑。以下是对这个类的详细解析：

主要功能

1. 接受并执行查询 (`accept` 和 `execute` 方法):

- `accept` 方法表示当前的 `JdbcExecutor` 可以处理所有的 `QueryStatement` 类型的查询请求。这个方法简单返回 `true`，表示它接受所有查询。
- `execute` 方法是核心方法，用于执行实际的 SQL 查询。它首先尝试使用查询加速器来执行查询（如果可能），如果加速器失败，则直接通过 JDBC 执行 SQL 查询。

主要方法解析

1. `accept(QueryStatement queryStatement)` :

- 该方法用于判断当前 `JdbcExecutor` 是否可以处理传入的 `QueryStatement` 对象。在此实现中，始终返回 `true`，意味着它接受所有查询语句。

2. `execute(QueryStatement queryStatement)` :

- 这是执行 SQL 查询的核心方法。流程如下：
 1. **查询加速器**：首先尝试通过 `ComponentFactory.getQueryAccelerators()` 获取的查询加速器列表，依次检查是否有可以加速当前查询的加速器。每个加速器通过 `check(QueryStatement)` 方法来判断是否可以处理该查询，如果可以，则使用 `query(QueryStatement)` 方法执行查询。如果加速器返回有效结果，则直接返回查询结果，而不执行后续的 JDBC 查询。
 2. **直接执行 SQL**：如果所有加速器都无法处理该查询，或者加速器返回了空结果，则通过 JDBC 直接执行 SQL 查询。此过程使用了 `SqlUtils` 工具类来执行 SQL 并获取结果。`SqlUtils` 类需要根据查询语句中的数据库信息初始化连接（通过 `init(database)`），然后调用 `queryInternal` 方法执行实际的 SQL 查询。
 3. **结果处理**：查询结果封装在 `SemanticQueryResp` 对象中，并返回给调用者。结果中还包含了执行的 SQL 语句。

3. `SemanticQueryResp` :

- 这个类用于封装查询的响应结果，包含查询的结果集、执行的 SQL 语句等信息。

4. `SqlUtils` :

- 这是一个工具类，负责处理与 JDBC 相关的操作。`init(database)` 方法初始化数据库连接，`queryInternal` 方法用于实际执行 SQL 查询并填充结果。

代码流程图

1. 加速器检查:

- 遍历所有查询加速器 → 检查加速器是否可以处理当前查询 (`check`) → 如果可以，则执行加速查询 (`query`) → 如果加速查询成功，则返回结果。

2. 直接执行 SQL:

- 如果加速器都未成功处理查询 → 初始化数据库连接 (`SqlUtils.init`) → 直接执行 SQL 查询 (`queryInternal`) → 返回查询结果。

日志与异常处理

- 代码中包含了详细的日志记录，比如当查询通过加速器执行时，记录了使用的加速器类名；当直接执行 SQL 时，记录了具体的 SQL 语句。
- 如果查询语句中没有包含 `SourceId`，则会发出警告日志并返回 `null`，这表示在执行查询之前检查 SQL 语句的完整性非常重要。

总结

`JdbcExecutor` 是一个灵活的 SQL 执行器，优先尝试使用加速器优化查询速度，如果加速器不可用或无效，则回退到标准的 JDBC 执行方式。这种设计允许在查询性能和灵活性之间取得平衡，并确保查询请求能够可靠地执行。