

DataSourceNode

`DataSourceNode` 是一个用于构建 SQL 语句的类，特别是与数据源相关的 SQL 片段。这个类主要负责将数据源的信息转换为适当的 SQL 表达式，确保查询能够正确地映射到底层数据库的表和字段上。以下是对该类的详细解析：

主要功能

1. 构建数据源 SQL 节点 (`build`):

- 该方法根据数据源信息构建 SQL 表达式。如果数据源包含 `sqlQuery`，直接使用该查询语句。如果包含 `tableQuery`，则将其转换为一个简单的 `SELECT * FROM ...` 语句。
- 构建后的 SQL 表达式会被进一步处理为 `SqlNode`，并添加到 SQL 验证范围 (`SqlValidatorScope`) 中。

2. 添加模式信息 (`addSchema`):

- 该方法从 SQL 查询中提取表和字段信息，并将这些信息添加到 SQL 验证范围中。这确保了在解析过程中能够正确识别这些表和字段。

3. 扩展 SQL 节点 (`buildExtend`):

- 该方法允许在构建 SQL 表达式时添加扩展的字段，例如使用 `Lateral View` `Explode` 这种 Hive 风格的操作符。这个方法通过将扩展表达式和基础数据源结合起来生成最终的 SQL 节点。

4. 匹配数据源 (`getMatchDataSources`):

- 这个方法用于根据查询中的度和维度信息，从语义模式中找到匹配的数据源。它首先根据度量信息找到最合适的基础数据源，然后通过检查维度和度量的匹配程度来选择其他相关的数据源。

5. 检查匹配程度 (`checkMatch`):

- 该方法用于验证基础数据源是否能够完全满足查询需求。它会检查查询中的度和维度是否在基础数据源中全部存在。

6. 通过连接关系获取数据源 (`getLinkDataSourcesByJoinRelation`):

- 这个方法通过检查数据源之间的连接关系，找到所有与基础数据源有相同标识符的关联数据源。

主要方法解析

1. `build` :
 - 构建基本的数据源 SQL 节点，主要用于处理简单的表查询或 SQL 查询，并将其转换为 `SqlNode`。
2. `addSchema` 和 `addSchemaTable` :
 - 这些方法负责将从 SQL 查询中提取的表和字段信息添加到 SQL 验证范围中，确保在后续的查询解析中能够正确识别这些元素。
3. `buildExtend` :
 - 允许在构建 SQL 表达式时添加额外的字段或表达式（如 `Lateral View` `Explode`），用于更复杂的查询需求。
4. `getMatchDataSources` :
 - 根据查询需求，从语义模式中找到最匹配的数据源。这个方法是整个查询生成过程中的核心部分，确保最终生成的 SQL 查询能够正确映射到底层的数据源。
5. `checkMatch` :
 - 验证基础数据源是否能够完全满足查询需求，检查查询中的度和维度是否都能在数据源中找到。
6. `getLinkDataSourcesByJoinRelation` :
 - 通过数据源之间的连接关系，找到所有可能与基础数据源相关联的其他数据源，确保能够生成正确的多表查询。

总结

`DataSourceNode` 类在 SQL 查询生成过程中起到了关键作用。它确保数据源的 SQL 表达式能够正确映射到底层数据库中的表和字段，并通过扩展和匹配数据源的功能支持更复杂的查询需求。通过解析语义模式中的信息，这个类可以生成精确的 SQL 查询，使得数据查询过程更加智能和高效。