

项目申请书

项目名称: Apache ShardingSphere: 增强 SQL Server SQL 解析引擎, 适配

更多 DDL、DML 语句

项目主导师: 张成

申请人: 陈喜盈

日期: 2025.06.03

邮箱: xiaosaxiaozai@qq.com

一、项目背景

- 项目基本需求
- 项目任务

二、项目实现

- 详细方案
 - 尽可能的收集 SQL Server 官方文档上不支持的语法 SQL Cases
 - 参考文档修复完善对应的语法解析
 - 添加解析的 E2E 测试用例
- 实现示例

三、规划

- 项目研发第一阶段 (07 月 01 日—08 月 30 日)
- 项目研发第二阶段 (08 月 31 日—09 月 30 日)
- 期望

一、项目背景

1. 项目基本需求

Apache ShardingSphere 是一款分布式数据库生态系统。目前，ShardingSphere 对 SQL Server 的 SQL 解析支持仍存在局限性，部分复杂 DDL（数据定义语言）和 DML（数据操作语言）语法无法被准确解析，导致用户在分片、读写分离、数据加密等场景中无法充分利用 SQL Server 的功能。此任务的目标是提升 ShardingSphere SQL 解析引擎对 SQL Server 的 SQL 支持度，提升对 SQL Server 的生态兼容性，满足用户需求。

2. 项目任务

Apache ShardingSphere 目前已经对 SQL Server 进行了初步适配，支持通过 JDBC 连接底层存储，实现数据加密、数据脱敏等增强功能。为了提升对 SQL Server 语句的支持度，需要增强 Apache ShardingSphere 内部的 SQL Server 解析引擎，适配更多的 DDL、DML 语句，并且增加 SQL Server 自动化测试程序，保证数据加密和数据脱敏功能可用。

参考 SQL Server 语法文档：

<https://learn.microsoft.com/zh-cn/sql/t-sql/language-reference>，
适配常用的 DDL 和 DML 语句，并补充对应的测试用例。相关任务如下：

- [SELECT](#)
- [INSERT](#)
- [UPDATE](#)
- [DELETE](#)
- [MERGE](#)
- [FUNCTIONS](#)

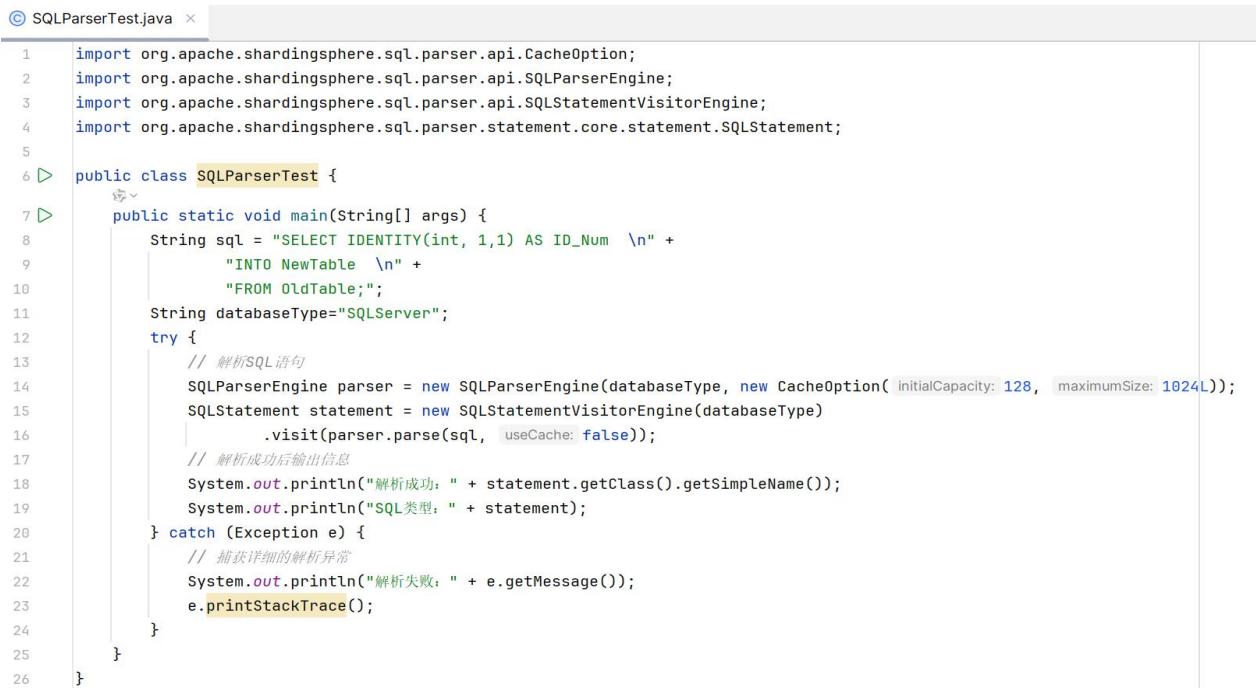
二、项目实现

项目成果仓库：<https://github.com/apache/shardingsphere>

1. 详细方案

i. 尽可能的收集 SQL Server 官方文档上暂不支持的语法 SQL Cases

- 关于寻找语法 SQL Cases，我阅读的是以下官方文档：
[Transact-SQL 参考（数据库引擎） – SQL Server | Microsoft Learn](#)，[SQL Server 故障排除文档 | Microsoft Learn](#)，以及技术文档下“参考”标签内所有内容。
- 通过一段编写的 Java 测试程序，调用 ShardingSphere 的 SQL parser 可以快速帮助判断 ShardingSphere 是否已支持以上文档中的语法 SQL Cases，已支持的语法便无需再处理，不支持的语法就记录收集下来，尽可能多的收集，后续根据不支持的 SQL Cases 进行修复。
- 测试程序如下图所示：



```
⑤ SQLParserTest.java ×
1 import org.apache.shardingsphere.sql.parser.api.CacheOption;
2 import org.apache.shardingsphere.sql.parser.api.SQLParserEngine;
3 import org.apache.shardingsphere.sql.parser.api.SQLStatementVisitorEngine;
4 import org.apache.shardingsphere.sql.parser.statement.core.Statement.SQLStatement;
5
6 public class SQLParserTest {
7     public static void main(String[] args) {
8         String sql = "SELECT IDENTITY(int, 1,1) AS ID_Num \n" +
9             "INTO NewTable \n" +
10            "FROM OldTable;";
11         String databaseType="SQLServer";
12         try {
13             // 解析SQL语句
14             SQLParserEngine parser = new SQLParserEngine(databaseType, new CacheOption(initialCapacity: 128, maximumSize: 1024L));
15             SQLStatement statement = new SQLStatementVisitorEngine(databaseType)
16                 .visit(parser.parse(sql, useCache: false));
17             // 解析成功后输出信息
18             System.out.println("解析成功: " + statement.getClass().getSimpleName());
19             System.out.println("SQL类型: " + statement);
20         } catch (Exception e) {
21             // 捕获详细的解析异常
22             System.out.println("解析失败: " + e.getMessage());
23             e.printStackTrace();
24         }
25     }
26 }
```

目前在空余时间利用该测试程序，我已收集一百五十余处不支持的 SQL Cases，大部分来自技术文档下“参考”标签内的内容，因篇幅有限无法一一列出。如果能有机会参与本项目我将一定会充分阅读一些还未阅读和测试的文档并尽可能多的收集目前 ShardingSphere 暂不支持的语法，完善和支持现有语法解析。以下为不支持的 SQL Cases 示例：

<https://learn.microsoft.com/zh-cn/sql/relational-databases/system-dynamic-management-views/sys-dm-exec-cached-plans-transact-sql?view=sql-server-ver17>

```
SELECT plan_handle, pvt.set_options, pvt.sql_handle
FROM (
    SELECT plan_handle, epa.attribute, epa.value
    FROM sys.dm_exec_cached_plans
    OUTER APPLY sys.dm_exec_plan_attributes(plan_handle) AS epa
    WHERE cacheobjtype = 'Compiled Plan'
) AS ecpa
PIVOT (MAX(epca.value) FOR ecpa.attribute IN ("set_options", "sql_handle")) AS pvt;
```

<https://learn.microsoft.com/zh-cn/sql/relational-databases/system-catalog-views/sys-partitions-transact-sql?view=sql-server-ver17>

```
SELECT object_name(object_id) AS ObjectName,
       total_pages / 128. AS SpaceUsed_MB,
       p.partition_id,
       p.object_id,
       p.index_id,
       p.partition_number,
       p.rows,
       p.data_compression_desc
  FROM sys.partitions AS p
 INNER JOIN sys.allocation_units AS au
    ON p.partition_id = au.container_id
 ORDER BY SpaceUsed_MB DESC;
```

<https://learn.microsoft.com/zh-cn/sql/relational-databases/system-dynamic-management-views/sys-dm-external-script-execution-stats?view=sql-server-ver17>

```
SELECT counter_name, counter_value
  FROM sys.dm_external_script_execution_stats
 WHERE language = 'Python';
```

<https://learn.microsoft.com/zh-cn/sql/relational-databases/system-dynamic-management-views/sys-dm-fts-active-catalogs-transact-sql?view=sql-server-ver17>

```
SELECT catalog.name, catalog.is_importing, catalog.auto_population_count,
OBJECT_NAME(population.table_id) AS table_name,
population.population_type_description, population.is_clustered_index_scan,
population.status_description, population.completion_type_description,
population.queued_population_type_description, population.start_time,
population.range_count
FROM sys.dm_fts_active_catalogs catalog
CROSS JOIN sys.dm_fts_index_population population
WHERE catalog.database_id = population.database_id
AND catalog.catalog_id = population.catalog_id
AND catalog.database_id = (SELECT dbid FROM sys.sysdatabases WHERE name = DB_NAME());
```

<https://learn.microsoft.com/zh-cn/sql/relational-databases/system-dynamic-management-views/sys-dm-tran-active-transactions-transact-sql?view=sql-server-ver17>

```
SELECT
GETDATE() as now,
DATEDIFF(SECOND, transaction_begin_time, GETDATE()) as
tran_elapsed_time_seconds,
st.session_id,
txt.text,
*
FROM
sys.dm_tran_active_transactions at
INNER JOIN sys.dm_tran_session_transactions st ON st.transaction_id =
at.transaction_id
LEFT OUTER JOIN sys.dm_exec_sessions sess ON st.session_id = sess.session_id
LEFT OUTER JOIN sys.dm_exec_connections conn ON conn.session_id =
sess.session_id
OUTER APPLY sys.dm_exec_sql_text(conn.most_recent_sql_handle) AS txt
ORDER BY
tran_elapsed_time_seconds DESC;
```

<https://learn.microsoft.com/zh-cn/sql/relational-databases/system-dynamic-management-views/sys-dm-tran-active-transactions-transact-sql?view=sql-server-ver17>

[em-functions/changetable-transact-sql?view=sql-server-ver17](https://learn.microsoft.com/zh-cn/sql/relational-databases/tables/functions-changetable-transact-sql?view=sql-server-ver17)

```
UPDATE
    SalesLT.Product
SET
    ListPrice = @new_listprice
FROM
    SalesLT.Product AS P
WHERE
    ProductID = @product_id AND
    @last_sync_version >= ISNULL (
        (SELECT CT.SYS_CHANGE_VERSION FROM
            CHANGETABLE(VERSION SalesLT.Product,
            (ProductID), (P.ProductID)) AS CT),
        0);
```

<https://learn.microsoft.com/zh-cn/sql/t-sql/xml/insert-xml-dml?view=sql-server-ver17>

```
UPDATE T
SET x.modify('insert <Material>Aluminum</Material> as first
    into    (/Root/ProductDescription/Features)[1]
');
'
```

<https://learn.microsoft.com/zh-cn/sql/t-sql/xml/nodes-method-xml-data-type?view=sql-server-ver17>

```
SELECT T2.Loc.query('.')
FROM T
CROSS APPLY Instructions.nodes('/root/Location') AS T2(Loc)
```

<https://learn.microsoft.com/zh-cn/sql/t-sql/functions/count-transact-sql?view=sql-server-ver17>

```
SELECT DISTINCT Name
    , MIN(Rate) OVER (PARTITION BY edh.DepartmentID) AS MinSalary
    , MAX(Rate) OVER (PARTITION BY edh.DepartmentID) AS MaxSalary
    , AVG(Rate) OVER (PARTITION BY edh.DepartmentID) AS AvgSalary
    , COUNT(edh.BusinessEntityID) OVER (PARTITION BY edh.DepartmentID) AS EmployeesPerDept
FROM HumanResources.EmployeePayHistory AS eph
```

```
JOIN HumanResources.EmployeeDepartmentHistory AS edh
    ON eph.BusinessEntityID = edh.BusinessEntityID
JOIN HumanResources.Department AS d
    ON d.DepartmentID = edh.DepartmentID
WHERE edh.EndDate IS NULL
ORDER BY Name;
```

ii. 参考文档以修复完善对应的语法解析

- 确认一个 SQL Cases 不被支持后，首先确认这是正确的 sql 语法，然后根据 SQL Server 官方文档，在 ShardingSphere 的 parser/sql/dialect/sqlserver/src/main/antlr4/imports/sqlserver 下相应的 g4 文件里添加支持，直到 ANTLR 可以正确解析 SQL。
- 然后编译 sqlserver 模块，运行 mvn -T 2C clean install -DskipTests
- 接着根据需要在 org/apache/shardingsphere/sql/parser/sqlserver/visitor/statement/SQLServerStatementVisitor 中添加或重写对应的 visit 方法。

iii. 添加解析的 E2E 测试用例

- 添加测试用例：根据语句的类型，在 test/it/parser/src/main/resources/sql/supported 下将待测语句依照格式加入到 .xml 文件中。
- 添加断言解析结果：根据语句类型，在 test/it/parser/src/main/resources/case 下将待测语句进行断言。
- 运行单元测试：运行 org/apache/shardingsphere/test/it/sql/parser/it/sqlserver

/internal/InternalSQLServerParserIT，运行成功则表示测试通过，完成了相应sql语法的修复完善。若运行异常，根据报错找出原因完善文件，直至没有异常测试通过。

2. 实现示例

示例根据详细方案的过程，解决了 IDENTITY 函数不被支持的问题：

```
https://learn.microsoft.com/zh-cn/sql/t-sql/functions/identity-function-transact-sql?view=sql-server-ver17
```

```
SELECT IDENTITY(int,1,1) AS ID_Num INTO NewTable FROM OldTable;
```

当在测试程序中尝试解析“SELECT IDENTITY(int, 1, 1) AS ID_Num INTO NewTable FROM OldTable;”时报错：

```
解析失败: You have an error in your SQL syntax: no viable alternative at input 'IDENTITY'  
org.apache.shardingsphere.sql.parser.exception.SQLParsingException Create breakpoint Lingma → : You have an error in your SQL syntax: no viable alternative at input 'IDENTITY'  
at org.apache.shardingsphere.sql.parser.core.database.parser.SQLParserErrorListener.syntaxError(SQLParserErrorListener.java:47)  
at org.antlr.v4.runtime.ProxyErrorListener.syntaxError(ProxyErrorListener.java:41)  
at org.antlr.v4.runtime.Parser.attemptParse(Parser.java:110)
```

在 ShardingSphere 中进行单元测试也如此：



于是参考文档修复对应的语法解析：

- src/main/antlr4/imports/sqlserver/DMLStatement.g4：在该语法文件的 projection 规则中添加对 IDENTITY 的解析。

```

144     projection
145         : qualifiedShorthand
146         | unqualifiedShorthand
147         | (alias EQ_)? (columnName | expr)
148         | (columnName | expr) (AS? alias)?
149         | identityProjection (AS? alias)?
150         ;
151
152     identityProjection
153         : IDENTITY LP_ dataTypeName (COMMA_ NUMBER_ (COMMA_ NUMBER_)? )? RP_
154         ;
155

```

2. parser/sql/dialect/sqlserver : 在该模块下执行: mvn -T 2C clean install -DskipTests, 生成 parser 代码。

3. org/apache/shardingsphere/sql/parser/sqlserver/visitor/stateme nt/SQLServerStatementVisitor.java : 在该 Visitor 中的 visitProjection 方法中, 判断并处理 identityProjection 分支:

```

± Thanoshan MV 1*
└─
1455 @Override
1456 public ASTNode visitProjection(final ProjectionContext ctx) {
1457     if (null != ctx.qualifiedShorthand()) {
1458         QualifiedShorthandContext shorthand = ctx.qualifiedShorthand();
1459         ShorthandProjectionSegment result = new ShorthandProjectionSegment(shorthand.getStart().getStartIndex(), shorthand.getStop().getStopIndex());
1460         IdentifierValue identifier = new IdentifierValue(shorthand.identifier().getText());
1461         result.setOwner(new OwnerSegment(shorthand.identifier().getStart().getStartIndex(), shorthand.identifier().getStop().getStopIndex(), identifier));
1462         return result;
1463     }
1464     if (null != ctx.unqualifiedShorthand()) {
1465         return new ShorthandProjectionSegment(ctx.unqualifiedShorthand().getStart().getStartIndex(), ctx.unqualifiedShorthand().getStop().getStopIndex());
1466     }
1467     AliasSegment alias = null == ctx.alias() ? null : (AliasSegment) visit(ctx.alias());
1468
1469     if (null != ctx.identityProjection()) {
1470         int startIndex = ctx.identityProjection().getStart().getStartIndex();
1471         int stopIndex = ctx.identityProjection().getStop().getStopIndex();
1472         String expr = ctx.identityProjection().getText();
1473         ExpressionProjectionSegment result = new ExpressionProjectionSegment(startIndex, stopIndex, expr);
1474         result.setAlias(alias);
1475         return result;
1476     }
1477
1478     if (null != ctx.columnName()) {
1479         ColumnSegment column = (ColumnSegment) visit(ctx.columnName());
1480         ColumnProjectionSegment result = new ColumnProjectionSegment(column);
1481         result.setAlias(alias);
1482         return result;
1483     }
1484     return createProjection(ctx, alias);
1485 }

```

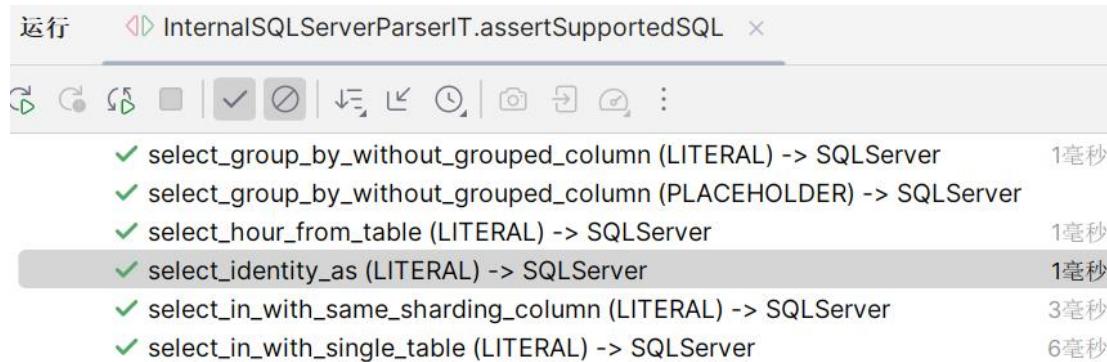
修 复 完 成 后 , 在
test/it/parser/src/main/resources/sql/supported/dml/select.xml 添加测
试用例:

```
<sql-case id="select_identity_as" value="SELECT IDENTITY(int,1,1) AS ID_Num INTO NewTable FROM OldTable;" db-types="SQLServer"/>
```

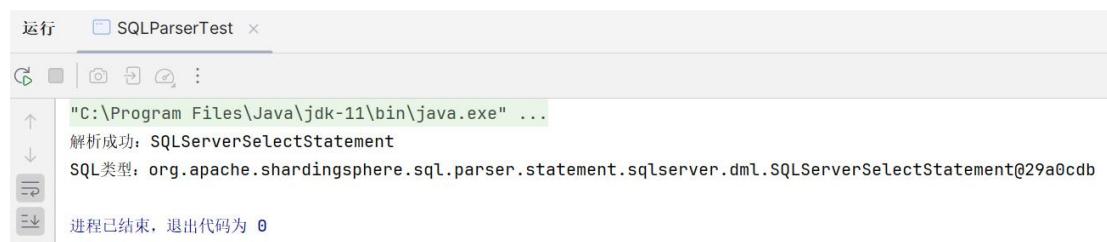
在 test/it/parser/src/main/resources/case/dml/select.xml 添加断言解析结果：

```
10220
10221     <select sql-case-id="select_identity_as">
10222         <projections start-index="7" stop-index="33">
10223             <expression-projection text="IDENTITY(int,1,1)" alias="ID_Num" start-index="7" stop-index="33"/>
10224     </projections>
10225     <into start-index="37" stop-index="49">
10226         <simple-table name="NewTable" start-index="40" stop-index="47"/>
10227     </into>
10228     <from start-index="50" stop-index="63">
10229         <simple-table name="OldTable" start-index="54" stop-index="61"/>
10230     </from>
10231 </select>
```

运行 InternalSQLServerParserIT， IDENTITY 成功被解析，没有异常：



在测试程序中也显示解析成功，完成修复；



三、规划

1. 项目研发第一阶段（07月01日—08月30日）
 - 完成收集 SQL Server 官方文档上不支持的语法 SQL Cases
 - 完成修复对应的语法解析
 - 完成添加 E2E 解析测试用例
2. 项目研发第二阶段（08月31日—09月30日）
 - 解决在中期验收阶段中遇到的问题
 - 对第一阶段完成的内容进行更详细的测试
 - 对第一阶段的完成内容进行总结，输出相关文档内容
 - 思考能够改进或者补充的地方

3. 期望

今年是我第一次参加开源之夏，我对 ShardingSphere 社区很感兴趣，阅读过官网文档和源码，也在学习过程中提交过 PR，参与了 good first issue，熟悉了社区规范与提交规范，为进行这次开源项目打下了基础，觉得这个项目是我有信心和把握可以完成的，非常希望我能有机会为 ShardingSphere 社区做出更多的贡献。

希望能够借助开源项目这个机会，收获开源相关的经验，为之后做出更多开源贡献提供一些帮助和借鉴。