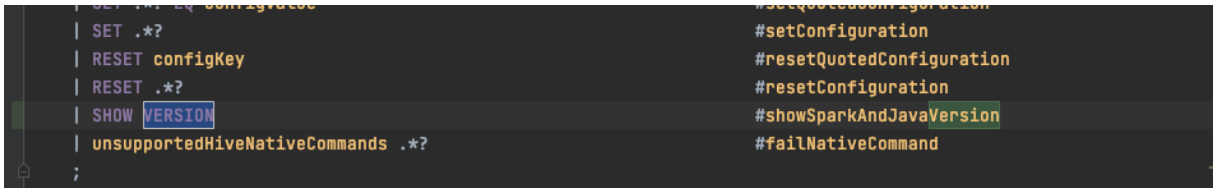
Spark SQL 作业实践

1.为Spark SQL添加一条自定义命令

- SHOW VERSION;
- 显示当前Spark版本和Java版本

在sqlbase.g4文件中增加新Keyword：SHOW VERSION，并编译antlr4。

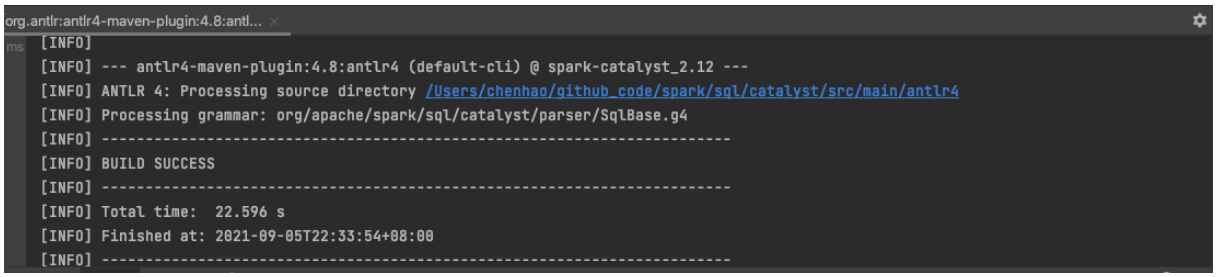
增加show version命令



增加version命令

在nonkeyword和ansinonkeyword中增加保留字

编译antlr4



在sparkSqlParse.scala文件中增加visitShowVersion方法

```
  //noinspection ScalaStyle
  override def visitShowVersion(ctx:
ShowSparkAndJavaVersionContext): LogicalPlan = withOrigin(ctx) {
      showVersionCommand()
  }
```

新增文件sql/core/src/main/scala/org/apache/spark/sql/execution/command/showVersionCommand.scala，实现showVersionCommand方法

```
package org.apache.spark.sql.execution.command

import org.apache.spark.sql.{Row, SparkSession}
import org.apache.spark.sql.catalyst.expressions.{Attribute,
AttributeReference}
import org.apache.spark.sql.types.StringType


case class showVersionCommand() extends LeafRunnableCommand {

    override val output: Seq[Attribute] =
        Seq(AttributeReference("user", StringType, nullable =
true)())

    //noinspection ScalaStyle
    override def run(sparkSession: SparkSession) : Seq[Row] ={
        val outputString = System.getenv("SPARK_VERSION")
        Seq(Row(outputString))
    }

}
```

2.构建SQL满足如下要求

通过set spark.sql.planChangeLog.level=WARN;查看

1. 构建一条SQL，同时apply下面三条优化规则：

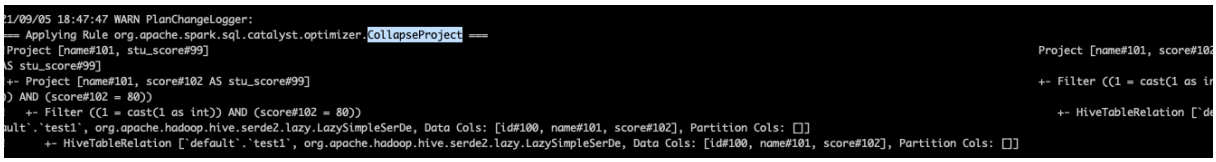
- CombineFilters
- CollapseProject
- BooleanSimplification

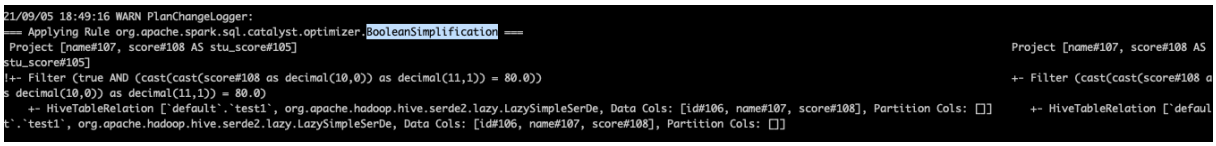
```
-- 建表
drop table if exists test1;
create table test1(
    id int,
    name string,
    score int
);
-- 记录初始化
insert overwrite table test1
select 1 as id, '张三' as name, 60 as score
union all
select 2 as id, '李四' as name, 70 as score
union all
select 3 as id, '王五' as name, 80 as score;
```

执行以下SQL语句，观察优化规则。

```
select a.name,a.stu_score
from ( select name, score as stu_score from test1
        where 1='1') a
where a.stu_score=80.0;
```

apply结果如图





2. 构建一条SQL，同时apply下面五条优化规则：

- ConstantFolding
- PushDownPredicates
- ReplaceDistinctWithAggregate
- ReplaceExceptWithAntiJoin
- FoldablePropagation

```
-- 建表
drop table if exists test2;
create table test2(
    id int,
    name string,
    sex string
);
drop table if exists test3;
create table test3(
    id int,
    course string,
    score int
);
-- 记录初始化
insert overwrite table test2
select 1 as id, '张三' as name, '男' as sex
union all
select 2 as id, '李四' as name, '男' as sex
union all
select 3 as id, '王五' as name, '男' as sex
union all
select 4 as id, '思思' as name, '女' as sex
union all
select 5 as id, '兰兰' as name, '女' as sex
union all
```

```
select 6 as id, null as name, null as sex;

insert overwrite table test3
select 1 as id, '语文' as course, 65 as score
union all
select 1 as id, '数学' as course, 70 as score
union all
select 1 as id, '英文' as course, 85 as score
union all
select 2 as id, '语文' as course, 65 as score
union all
select 2 as id, '数学' as course, 75 as score
union all
select 2 as id, '英文' as course, 90 as score
union all
select 3 as id, '语文' as course, 90 as score
union all
select 3 as id, '数学' as course, 80 as score
union all
select 3 as id, '英文' as course, 85 as score
union all
select 4 as id, '语文' as course, 80 as score
union all
select 4 as id, '数学' as course, 70 as score
union all
select 4 as id, '英文' as course, 85 as score;
```

执行以下SQL语句，观察优化规则。

```
select distinct t3.id
from (select id, sum(score) as allscore
        from test3
    group by id) as t3
join test2 as t2
  on t3.id = t2.id and t2.sex='男'
where t3.allscore >= 200
```

apply结果如图

```
21/09/05 21:54:58 WARN PlanChangeLogger:
=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ConstantFolding ===
 Aggregate [id#64], [id#64]                                                          Aggregate [id#64], [id#64]
 +- Project [id#64]                                                                  +- Project [id#64]
   +- Join Inner, ((id#64 = id#67) AND (sex#69 = 男))                                   +- Join Inner, ((id#64 = id#67) AND (sex#69 = 男))
      :- Project [id#64]                                                                  :- Project [id#64]
!     : +- Filter (allscore#63L >= cast(200 as bigint))                                    : +- Filter (allscore#63L >= 200)
      :    +- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]          :    +- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]
      :       +- Project [id#64, score#66]                                                :       +- Project [id#64, score#66]
      :          +- HiveTableRelation [`default`.`test3`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]       :          +- HiveTableRelation [`default`.`test3`,
org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]
      +- Project [id#67, sex#69]                                                          +- Project [id#67, sex#69]
         +- HiveTableRelation [`default`.`test2`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]                  +- HiveTableRelation [`default`.`test2`,
org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]
```

```
21/09/05 21:54:58 WARN PlanChangeLogger:
=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===
 Aggregate [id#64], [id#64]                                                          Aggregate [id#64], [id#64]
 +- Project [id#64]                                                                  +- Project [id#64]
   +- Join Inner, (id#64 = id#67)                                                       +- Join Inner, (id#64 = id#67)
!     :- Filter isnotnull(id#64)                                                           :- Project [id#64]
!     : +- Project [id#64]                                                                 : +- Filter ((isnotnull(allscore#63L) AND (allscore#63L >= 200)) AND isnotnull(id#64))
!     :    +- Filter (isnotnull(allscore#63L) AND (allscore#63L >= 200))                        :    +- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]
!     :       +- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]       :       +- Project [id#64, score#66]
!     :          +- Project [id#64, score#66]                                             :          +- HiveTableRelation [`default`.`test3`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64,
course#65, score#66], Partition Cols: []]
!     :             +- HiveTableRelation [`default`.`test3`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]       +- Project [id#67]
!     +- Filter isnotnull(id#67)                                                           +- Filter ((isnotnull(sex#69) AND (sex#69 = 男)) AND isnotnull(id#67))
!        +- Project [id#67]                                                                  +- HiveTableRelation [`default`.`test2`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68,
sex#69], Partition Cols: []]
!           +- Filter (isnotnull(sex#69) AND (sex#69 = 男))
!              +- HiveTableRelation [`default`.`test2`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]

21/09/05 21:54:58 WARN PlanChangeLogger:
=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===
 Aggregate [id#64], [id#64]                                                          Aggregate [id#64], [id#64]
 +- Project [id#64]                                                                  +- Project [id#64]
   +- Join Inner, (id#64 = id#67)                                                       +- Join Inner, (id#64 = id#67)
      :- Project [id#64]                                                                  :- Project [id#64]
!     : +- Filter ((isnotnull(allscore#63L) AND (allscore#63L >= 200)) AND isnotnull(id#64))      : +- Filter (isnotnull(allscore#63L) AND (allscore#63L >= 200))
      :    +- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]          :    +- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]
      :       +- Project [id#64, score#66]                                                :       +- Project [id#64, score#66]
!     :          +- HiveTableRelation [`default`.`test3`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]       :          +- Filter isnotnull(id#64)
!     +- Project [id#67]                                                                  :             +- HiveTableRelation [`default`.`test3`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65,
score#66], Partition Cols: []]
!        +- Filter ((isnotnull(sex#69) AND (sex#69 = 男)) AND isnotnull(id#67))                +- Project [id#67]
!           +- HiveTableRelation [`default`.`test2`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]          +- Filter ((isnotnull(sex#69) AND (sex#69 = 男)) AND isnotnull(id#67))
!                                                                                           +- HiveTableRelation [`default`.`test2`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69],
Partition Cols: []]
```

```
21/09/05 21:54:58 WARN PlanChangeLogger:
=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReplaceDistinctWithAggregate ===
!Distinct                                                                            Aggregate [id#64], [id#64]
 +- Project [id#64]                                                                     +- Project [id#64]
   +- Filter (allscore#63L >= cast(200 as bigint))                                         +- Filter (allscore#63L >= cast(200 as bigint))
      +- Join Inner, ((id#64 = id#67) AND (sex#69 = 男))                                      +- Join Inner, ((id#64 = id#67) AND (sex#69 = 男))
         :- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]            :- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]
         : +- HiveTableRelation [`default`.`test3`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]       : +- HiveTableRelation [`default`.`test3`,
org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]
         +- HiveTableRelation [`default`.`test2`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]          +- HiveTableRelation [`default`.`test2`,
org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]

21/09/05 21:54:58 WARN PlanChangeLogger:
=== Result of Batch Replace Operators ===
!Distinct                                                                            Aggregate [id#64], [id#64]
 +- Project [id#64]                                                                     +- Project [id#64]
   +- Filter (allscore#63L >= cast(200 as bigint))                                         +- Filter (allscore#63L >= cast(200 as bigint))
      +- Join Inner, ((id#64 = id#67) AND (sex#69 = 男))                                      +- Join Inner, ((id#64 = id#67) AND (sex#69 = 男))
         :- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]            :- Aggregate [id#64], [id#64, sum(cast(score#66 as bigint)) AS allscore#63L]
         : +- HiveTableRelation [`default`.`test3`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]       : +- HiveTableRelation [`default`.`test3`,
org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#64, course#65, score#66], Partition Cols: []]
         +- HiveTableRelation [`default`.`test2`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]          +- HiveTableRelation [`default`.`test2`,
org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [id#67, name#68, sex#69], Partition Cols: []]
```

3.实现自定义优化规则（静默规则）

待实现