题目一: 分析一条 TPCDS SQL

分析一条 TPCDS SQL（请基于 Spark 3.1.1 版本解答）

* 运行该 SQL，如 q38，并截图该 SQL 的 SQL 执行图
* 该 SQL 用到了哪些优化规则（optimizer rules）
* 请各用不少于 200 字描述其中的两条优化规则
* SQL 从中任意选择一条：  
  <https://github.com/apache/spark/tree/master/sql/core/src/test/resources/tpcds>

答案

我选用的是 q38 这条 SQL。 运行命令：

(base) chchang@chchang-a01 spark-tpcds-datagen % ./spark-3.1.1-bin-hadoop2.7/bin/spark-submit --conf spark.sql.planChangeLog.level=WARN --class org.apache.spark.sql.execution.benchmark.TPCDSQueryBenchmark --jars spark-core\_2.12-3.1.1-tests.jar,spark-catalyst\_2.12-3.1.1-tests.jar spark-sql\_2.12-3.1.1-tests.jar --data-location tpcds-data-1g --query-filter "q38" > /Users/chchang/Downloads/temp/spark.rules.log 2>&1

该 SQL 用到了这些优化规则：

(base) chchang@chchang-a01 temp % grep "org.apache.spark.sql.catalyst.optimizer" ./spark.rules.log

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReplaceIntersectWithSemiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReplaceDistinctWithAggregate ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReorderJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.CollapseProject ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.EliminateLimits ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ConstantFolding ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.InferFiltersFromConstraints ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RewritePredicateSubquery ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReplaceIntersectWithSemiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReplaceDistinctWithAggregate ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReorderJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.CollapseProject ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.EliminateLimits ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ConstantFolding ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.InferFiltersFromConstraints ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RewritePredicateSubquery ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReplaceIntersectWithSemiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReplaceDistinctWithAggregate ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ReorderJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.CollapseProject ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.EliminateLimits ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ConstantFolding ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.InferFiltersFromConstraints ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownPredicates ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.PushDownLeftSemiAntiJoin ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.ColumnPruning ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RemoveNoopOperators ===

=== Applying Rule org.apache.spark.sql.catalyst.optimizer.RewritePredicateSubquery ===

PushDownPredicates

谓词下推（Predicate PushDown），即将查询的过滤条件尽可能下沉到数据源。目的是为了减少非必须数据的读取。谓词下推能将过滤条件下推到JOIN之前进行，这样在扫描数据的时候就对数据进行了过滤，参与JOIN的数据量将会得到显著地减少，JOIN耗时必然也会降低。从执行计划来看，就是将Filter下推到Join之前先执行。

### ColumnPruning

列裁剪在 Spark SQL 是由 ColumnPruning  
实现的。因为我们查询的表可能有很多个字段，但是每次查询我们很大可能不需要扫描出所有的字段，这个时候利用列裁剪可以把那些查询不需要的字段过滤掉，使得扫描的数据量减少。

这个优化一方面大幅度减少了网络、内存数据量消耗，另一方面对于列存格式（Parquet）来说大大提高了扫描效率。