

team12_assignment3_report

106062137 徐郁閔、106033233 周聖諺、p123786579 王麒銘

Implement

1. 判斷 / 儲存 explain 關鍵字
 - a. 在 Lexer 的 `initKeywords()` 增加 explain 的 keyword。
 - b. 在 QueryData 的增加布林值 `isExplain`，儲存輸入的 query 是否有 explain 關鍵字。
 - c. 在 Parser 的 `queryCommand()` 首先判斷開頭是否為 explain 關鍵字，並更改 QueryData 的 `isExplain` 布林值。
2. 處理 `isExplain` 為真的 QueryData
 - a. 在 BasicQueryPlanner 中 `createPlan()` 的尾端 (Plan tree 的 root) 判斷 `isExplain`，依判斷結果加入 ExplainPlan 至當前的 Plan tree。
 - b. 新增 ExplainPlan 與 ExplainScan 兩個 class:
 - i. 在 ExplainPlan 的 constructor 中初始化自定義的 private String `explain`，並呼叫 child plan 中新增的 `getExplain()`。另外，在 ExplainPlan 的 constructor 增加 schema，依規定設定 field name "query-plan" 與 type "varchar(500)"。
 - ii. 在 ExplainScan 的 constructor 中傳入 ExplainPlan 的 `explain` String，並在 `getVal()` 中回傳 `explain` 值。
 - iii. 在 ExplainScan 的 `getVal` method 中，將 `explain` 的文字結果放到 VarcharConstant 並回傳。
3. `getExplain()` 實作
 - a. 在每個 *Plan 的 class 中新增 `getExplain()` (包含 ProductPlan、ProjectPlan、SelectPlan、TablePlan，與 materialize 的 GroupByPlan、MaterializePlan、MergeJoinPlan、SortPlan)，將當前 `className` 與 `blocksAccessed()`、`recordsOutput()` 等參數組成 String 回傳，同時遞迴地呼叫下一層 Plan 的 `getExplain()`，直到尾端 leaf plan 的 TablePlan。

Test cases

db21-assignment-3 - core-patch/src/test/java/org/vanilladb/core/query/algebra/materialize/MaterializeTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer JUnit

Finished after 1.193 seconds

Runs: 26/26 Errors: 0 Failures: 0

- org.vanilladb.core.query.algebra.materialize.MaterializeTest [Runner: JUnit 4] (0.205 s)
 - testSort (0.109 s)
 - testAggregationFn (0.012 s)
 - testGroupBy (0.084 s)
- org.vanilladb.core.query.parse.ParserTest [Runner: JUnit 4] (0.004 s)
 - testIndexCreationWithGivenType (0.004 s)
 - testMultiKeysIndexCreation (0.000 s)
 - testIndexCreation (0.000 s)
- org.vanilladb.core.query.algebra.BasicQueryTest [Runner: JUnit 4] (0.088 s)
 - testTable (0.002 s)
 - testProduct (0.079 s)
 - testProject (0.003 s)
 - testSelect (0.004 s)
- org.vanilladb.core.query.planner.BasicQueryPlannerTest [Runner: JUnit 4] (0.092 s)
 - testQuery (0.053 s)
 - testView (0.025 s)
 - testDelete (0.007 s)
 - testInsert (0.002 s)
 - testModify (0.005 s)
- org.vanilladb.core.query.planner.VerifierTest [Runner: JUnit 4] (0.019 s)
 - testQueryData (0.002 s)
 - testInsertData (0.003 s)
 - testModifyData (0.007 s)
 - testCreateTableData (0.001 s)
 - testCreateViewData (0.003 s)
 - testDeleteData (0.001 s)
 - testCreateIndexData (0.001 s)
- org.vanilladb.core.query.parse.ParseTest [Runner: JUnit 4] (0.001 s)
 - testParseInsert (0.000 s)
 - testParseSelect (0.001 s)
 - testCreateTable (0.000 s)
 - testParseConstant (0.000 s)

Experiements

Experiment 1: Multi-Table WHERE

SQL

```
EXPLAIN SELECT c_id, c_first, c_middle, c_last
FROM customer
WHERE c_id < 100;
```

Result

```
SQL> EXPLAIN SELECT c_id, c_first, c_middle, c_last FROM customer WHERE c_id < 100;

query-plan
-----
->ProjectPlan(#blks=15001,#recs=940)
  ->SelectPlan pred:(c_id<100.0) (#blks=15001, #recs=940)
    ->TablePlan on (customer) (#blks=15001,#recs=30000)

Actual #recs: 990

SQL> SELECT c_id, c_first, c_middle, c_last FROM customer WHERE c_id < 100;
|
|      c_first  c_id      c_lastc_middle
|-----|
| 1d0aR1m9gl    1    BARBARBAROE
| 1Gw5LOm       2    BARBAROUGHTOE
| iA6RfmXcIf     3    BARBARABLEOE
| nOHR9EATvZ     4    BARBARPRIOE
| 7dRA7k         5    BARBARPRESOE
| RlbvgzM        6    BARBARESEOE
```

Experiment 2: Single-Table WHERE

SQL

```
EXPLAIN SELECT c_id, c_first, c_middle, c_last
FROM orders, customer
WHERE o_c_id = c_id;
```

Result

```
SQL> EXPLAIN SELECT d_id FROM district, warehouse WHERE d_w_id = w_id
```

```
query-plan
```

```
-----
->ProjectPlan(#blks=22,#recs=10)
  ->SelectPlan pred:(d_w_id=w_id) (#blks=22, #recs=10)
    ->ProductPlan (#blks=22, #recs=10)
      ->TablePlan on (district) (#blks=2,#recs=10)
      ->TablePlan on (warehouse) (#blks=2,#recs=1)
```

```
Actual #recs: 10
```

```
SQL> SELECT d_id FROM district, warehouse WHERE d_w_id = w_id
```

```
|
d_id
-----
1
2
3
4
5
6
7
8
9
10
```

Experiment 3: ORDER BY

SQL

```
EXPLAIN SELECT c_id, c_first, c_middle, c_last
FROM customer
WHERE c_id < 100
ORDER BY c_id;
```

Result

```
SQL> EXPLAIN SELECT c_id, c_first, c_middle, c_last FROM customer WHERE c_id < 100 ORDER BY c_id;

query-plan
-----
->SortPlan(#blks=29,#recs=940)
  ->ProjectPlan(#blks=15001,#recs=940)
    ->SelectPlan pred:(c_id<100.0) (#blks=15001, #recs=940)
      ->TablePlan on (customer) (#blks=15001,#recs=30000)
```

Actual #recs: 990

```
SQL> SELECT c_id, c_first, c_middle, c_last FROM customer WHERE c_id < 100 ORDER BY c_id;
|
+-----+-----+-----+-----+
| c_id | c_first | c_middle | c_last |
+-----+-----+-----+-----+
| 1 | uINYDF5G | BARBAROE | BARBAROE |
| 1 | lViS6w | BARBAROE | BARBAROE |
| 1 | KoBbkpz8D | BARBAROE | BARBAROE |
| 1 | lxApntP | BARBAROE | BARBAROE |
| 1 | 1d0aR1m9gl | BARBAROE | BARBAROE |
| 1 | Df48wXwg | BARBAROE | BARBAROE |
| 1 | Oz4cs2eW | BARBAROE | BARBAROE |
| 1 | xArz5m | BARBAROE | BARBAROE |
| 1 | JQ3Hzttf | BARBAROE | BARBAROE |
| 1 | jSSF0zqIbJ | BARBAROE | BARBAROE |
| 2 | pU2Fsy40D | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | BmrPu57 | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | xohf1509 | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | KRoPWF | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | 1Gw5LOm | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | IJpVMzb | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | i1MKXit26 | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | UpjxrOE | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | ltZIXn | BARBAROUGHTOE | BARBAROUGHTOE |
| 2 | i0tKoR | BARBAROUGHTOE | BARBAROUGHTOE |
| 3 | YuAqr90y | BARBARABLEOE | BARBARABLEOE |
| 3 | skMJa2rA | BARBARABLEOE | BARBARABLEOE |
| 3 | I0dF31091 | BARBARABLEOE | BARBARABLEOE |
| 3 | PhMSkVUV1 | BARBARABLEOE | BARBARABLEOE |
| 3 | iA6RfmXcIf | BARBARABLEOE | BARBARABLEOE |
| 3 | F5Xy0u2 | BARBARABLEOE | BARBARABLEOE |
| 3 | sjiCbLa1 | BARBARABLEOE | BARBARABLEOE |
| 3 | ... | ... | ... |
```

Experiment 4: GROUP BY

SQL

```
EXPLAIN SELECT COUNT(d_id)
FROM district, warehouse
WHERE d_w_id = w_id
GROUP BY w_id;
```

Result

```
SQL> EXPLAIN SELECT COUNT(d_id) FROM district, warehouse WHERE d_w_id = w_id GROUP BY w_id
```

```
query-plan
```

```
-----
->ProjectPlan(#blks=2,#recs=1)
  ->GroupByPlan(#blks=2,#recs=1)
    ->SortPlan(#blks=2,#recs=10)
      ->SelectPlan pred:(d_w_id=w_id) (#blks=22, #recs=10)
        ->ProductPlan (#blks=22, #recs=10)
          ->TablePlan on (district) (#blks=2,#recs=10)
          ->TablePlan on (warehouse) (#blks=2,#recs=1)
```

```
Actual #recs: 1
```

```
SQL> SELECT COUNT(d_id) FROM district, warehouse WHERE d_w_id = w_id GROUP BY w_id
```

```
|
countofd_id
-----
          10
```

```
SQL>
```