

Phase 1 Report

107062337 陽晴卉 107062324 廖淳鈺 107062222 張瑋芸

1. Implement the Explain operation

In assignment3, we have added 2 new files and modified 8 files.

- **Additional files**

ExplainPlan

參考 algebra 下的 Plan 架構設計 ExplainPlan，並增加 toString function，主要分成三個改動點。

- new ExplainPlan (Constructor)

將傳入的 Plan p 參數存在 variable p 中，用來取得下層 plan 的資訊。
接著在 schema 新增 "query-plan" field (type=VARCHAR)，jdbc client 可以藉由選擇該 field 取得 explain result。

- ExplainPlan.open()

不同於其他 plan 會層層呼叫 TablePlan 的 open function 來 access table file，explain 不需印出 record contents，所以不會呼叫 p.open()。相對的 explain 會將 plan tree 架構以字串的形式傳入 new ExplainScan。

- ExplainPlan.toString()

此次功課中新增的函數，目的是紀錄到本層 plan 為止的 plan tree 架構，並以字串形式回傳，在其他 plan class 中也有新增此函數，會在下面提及。實做細節為：

- 1) append 分隔線至 StringBuilder 中

- 2) 呼叫 p.toString 取得目前為止的 plan tree 架構，append 至 StringBuilder 中。

- 3) append 該層 plan 的 access blocks 和 records 到 StringBuilder 中。
step2 利用遞迴的方式，層層呼叫至底層 tablePlan，且回傳的過程中每個 plan 都會不斷合併下層結果至 StringBuilder 中，最終可取得完整的 plan tree 架構，傳入 ExplainScan 中。

- blocksAccessed、histogram、recordsOutput

因為 explain 並不會改變 records 和 fields 的數量，不涉及 cost time 的計算，所以直接回傳下層的 plan 資料即可，不須做改動。

ExplainScan

參考 algebra 下的 Scan 架構設計 ExplainScan，另外由於 explain 不需 access table file (前面有提及)，所以不需實作 scan 相關函數，只處理 plan tree result 的 output。主要有三個改動點：

➤ New ExplainScan()

將傳入的 plan tree info 參數存入 private 變數-info 中，記錄要 output 的所有資訊。

➤ ExplainScan.next()

確認是否還有內容要 output，若 info 為空字串，回傳 false。

➤ ExplainScan.getVal()

回傳要 output 的 explain result，並把 info 清空，代表 explain 已 output 過，這樣下一輪 next 就會回傳 false，結束 ExplainScan。

➤ beforeFirst()、close()、hasFiled()

不需改動，所以為空或回傳 false。

- **Modified files**

TablePlan、ProductPlan、SelectPlan、ProjectPlan

參考 GroupByPlan 的寫法，新增 toString function，實作方式與前面敘述的 ExplainPlan.toString()相同，並根據 plan 的種類修改對應的說明文字。

TablePlan

增加 private 變數-tblName，在 init 時紀錄 table 的名稱，因為 toString 需要 append table 的名字。

ProductPlan

在 toString 中需要額外處理 plans 之間的先後順序，先 append plan1 的 explain info，再 append plan2 的 explain info。

Lexer

initKeywords()中的 keyword array 儲存 Lexer 需要判斷的所有 keyword，因此在此 array 中增加 keyword "explain"。

Parser

在 queryCommand function 中，新增 boolean isExplain 變數，default 為 false，當 lexer 有 match 到"explain" 這個 keyword 時，lexer 便會讀入這個

keyword，並將 isExplain 設為 true 後作為參數傳入 QueryData。

QueryData

- 增加 private 變數-isExplain
表示此 query 是否為 explain 指令。
- 修改 toString()
增加對 isExplain 的判斷，若 isExplain 為 true，則在 result 中加入 "explain "的字串。
- 新增 QueryData.isExplain()
回傳 isExplain 參數值。

BasicQueryPlan

在 createPlan function 的最後加入對 Querydata 的 isExplain 判斷，如果 isExplain 為 true，則新增一個 ExplainPlan。

2. The Explain results for different queries.

- A query accessing **single table** with **WHERE**

```
SQL> explain select d_id from district where d_id < 5

query-plan
-----
->ProjectPlan: (#blks=2, #recs=4)
    ->SelectPlan pred:(d_id<5.0) (#blks=2, #recs=0)
        ->TablePlan: on (district) (#blks=2, #recs=10)
Actual #recs: 4
```

- A query accessing **multiple table** with **WHERE**

```
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
```

- A query with **ORDER BY**

```
SQL> explain select d_name from district order by d_name

query-plan
-----
->SortPlan (#blks=1, #recs=10)
    ->ProjectPlan: (#blks=2, #recs=10)
        ->SelectPlan pred:() (#blks=2, #recs=10)
            ->TablePlan: on (district) (#blks=2, #recs=10)
Actual #recs: 10
```

- A query with **GROUP BY** and **at least one aggregation function**

```
SQL> explain select count(d_id) from district

query-plan
-----
->ProjectPlan: (#blks=2, #recs=1)
    ->GroupByPlan: (#blks=2, #recs=1)
        ->SelectPlan pred:() (#blks=2, #recs=10)
            ->TablePlan: on (district) (#blks=2, #recs=10)
Actual #recs: 1
```