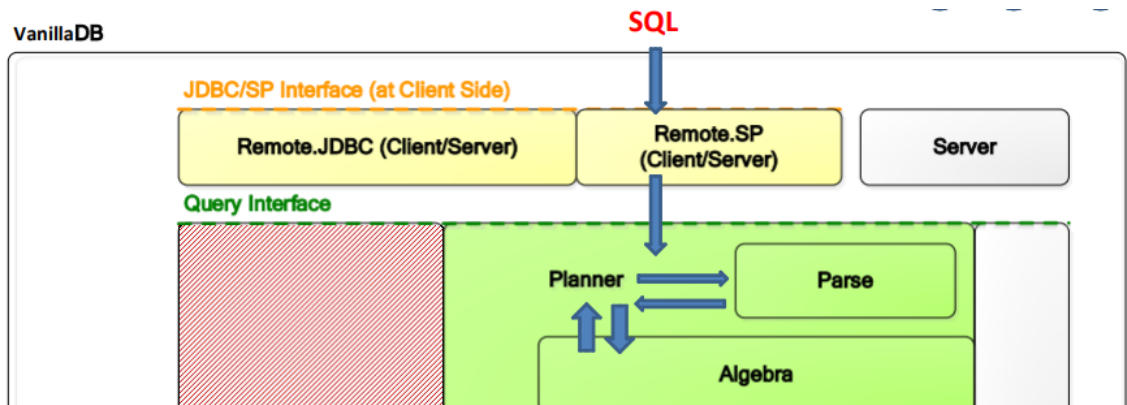


# Team16 Assignment 2 Phase 1 Report

109062123 曹瀚文 109062323 吳聲宏 109062330 黃鈺臻

## 1. Implementation



根據 VanillaDB 的架構, 我們如果要實作 EXPLAIN 的指令需要修改的幾個地方有

1. Parse: 讓 VanillaDB 可以解讀 EXPLAIN 指令
2. Planner: 創建 EXPLAIN 的 plan
3. Scan: 掃描所有經過的 plan

### 1.1 Parse

主要修改地方有兩個, 一個是 queryCommand 在這裏用 Lexer 的 matchKeyword 和 eatKeyword 去讀取使用者是否有輸入 EXPLAIN 的關鍵字, 同時要記得將 EXPLAIN 加入 Lexer 的 keywords list 裏面。如果使用者真的有下 EXPLAIN 則在 QueryData 裏面加入標記, 並在 createQueryPlan 檢查, 並將原本產生的 plan 最上層在加入 EXPLAIN 的 plan。

### 1.2 Planner

首先新增一個 ExplainTree 的資料結構, 目的是為了儲存 EXPLAIN 語法中要輸出的 plan 的資訊。

這裏為了能支援每個 EXPLAIN, 所以在 plan 的 interface 中增加一個新的 explainTree 的 function, 希望每個 plan 會在這個 function 中增加自己的 explainTree, 並且遞迴的呼叫接續的 plan 去創建 explainTree。

新增一個名為 ExplainPlan 的 class 並繼承 Plan, 接着實作 Plan 的 interface。constructor, blocksAccessed 和 open 都跟其他的 planner 實作方式類似, schema 部分則是回傳 長度500的 varchar, recordsOutput 回傳 1, histogram 回傳子 plan 的 histogram 結果, explainTree 則是創立一個 ExplainTree 的結構, 並將目前 plan 的資訊記錄起來, 並呼叫子 plan。

由於在 Plan 的 interface 新增了新的 explainTree function, 所以要將所有繼承 plan 的 class 都實作 explainTree function。

### 1.3 Scan

首先新增一個 class 名為 ExplainScan 並繼承 Scan, 接着實作應該要實作的 interface。hasField, getVal 和 close 則是跟其他 Scan 的實作方式類似。在 beforeFirst 和 next 的做法是用一個 flag 處理每次呼叫 beforeFirst 後 next 會回傳 true 否則皆回傳 false。

而 constructor 部分則是先初始化基本 scan 的參數, 並記錄子 plan 回傳的資料數量, 同時遞迴去讀取子 plan 中的 ExplainTree 的資訊, 並將其生成一個符合作業格式的字串, 最後將其記錄起來。

## 2. Result

- A query accessing single table with WHERE

```
SQL> EXPLAIN SELECT d_id FROM district WHERE d_id > 1;

query-plan
-----
->ProjectPlan (#blks=2, #recs=9)
  ->SelectPlan pred:(d_id>1.0) (#blks=2, #recs=0)
    ->TablePlan on (district) (#blks=2, #recs=10)

Actual #recs: 9
```

- A query accessing multiple tables with WHERE

```
SQL> EXPLAIN SELECT d_id, w_id FROM district, warehouse WHERE d_id > 1;

query-plan
-----
->ProjectPlan (#blks=22, #recs=9)
  ->SelectPlan pred:(d_id>1.0) (#blks=22, #recs=0)
    ->ProductPlan (#blks=22, #recs=10)
      ->TablePlan on (district) (#blks=2, #recs=10)
      ->TablePlan on (warehouse) (#blks=2, #recs=1)

Actual #recs: 9
```

- A query with ORDER BY

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

```
query-plan
```

```
-----  
->ProjectPlan (#blks=2, #recs=0)  
  ->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
```

- A query with GROUP BY and at least one aggregation function (MIN, MAX, COUNT, AVG... etc.)

```
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
```

### 3. Anything worth mentioning

在 EXPLAIN 的輸入必須要在一開始加一個換行，不然第一行的資料會接在橫線後，整個輸出會跑掉。