Assignment 3 Report

Team14 106070038 杜葳葳 106030019 吳岱容

- **Implement Explain**
  - In org.vanilladb.core.query.parse
    - Lexer：將 explain 加入 keywords
    - Parser：新增一個 boolean 值 explain，matchkeyword 來判斷是不是 EXPLAIN，如果是則吃掉 EXPLAIN 並將 explain 的 boolean 值更為 true，並在回傳新的 QueryData 時，一併加入成員一起回傳
    - QueryData：在 class 內新增 explain(型態為 boolean)成員，並加入 getExplain()方法(回傳 explain 的值)
  - In org.vanilladb.core.query.planner
    - BasicQueryPlanner：在程式碼最下面的部分新增條件式判斷 query 是不是需要 explain(實際是 tree 的最上層)，如果需要就新增一個 ExplainPlan
  - In org.vanilladb.core.query.algebra
    - Plan：在這個 interface 下新增 getPlanInfo()的方法，讓我們可以在每種類型的 Plan 下實作
    - ExplainPlan：在建構子內將成員的 schema 新增一個新 field(query-pan)，然後在 open 的方法中回傳 ExplanScan，同時已遞迴方式打開下層的 Plan 以及使用 getPlanInfo 的方式拿到每一層的資訊
    - ExplainScan：接收我們在 ExplanPlan 中 open 的 plan 以及 getPlanInfo 收到的資料，並用 while 迴圈再次紀錄 Actual Record 的總數目，另外實作 getVal()的方法，讓我們有辦法拿到 query-plan 這個 field 的資訊，每層呼叫 getPlanInfo 拿到的資料都集合在這裡，同時在這個類別我們也需要紀錄是不是第一次進到 ExplainScan
    - 其他各類型的 Plan：在個別 Plan 的類別下實作各類型的 getPlanInfo
- **Result and explanation**
  - Example query from the spec

```
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
```

- A query accessing single table with WHERE

```
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 tables with WHERE

```
EXPLAIN SELECT d_street_1 FROM district, warehouse WHERE w_id = d_id

query-plan
-----------------------------------------------------------------------
->ProjectPlan  (#blks=22, #recs=0)
       ->SelectPlan pred:(w_id=d_id) (#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: 1
```

- A query with ORDER BY

```
EXPLAIN SELECT c_id, c_first FROM customer WHERE c_id < 50 ORDER BY c_id DESC

query-plan
-----------------------------------------------------------------------------
->SortPlan  (#blks=8, #recs=482)
        ->ProjectPlan  (#blks=15001, #recs=482)
                ->SelectPlan pred:(c_id<50.0) (#blks=15001, #recs=482)
                        ->TablePlan on (customer) (#blks=15001, #recs=30000)
Actual #recs: 490
```

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

```
SQL> EXPLAIN SELECT COUNT(d_street_1) FROM district GROUP BY d_street_1

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


SQL> EXPLAIN SELECT AVG(d_tax) FROM district GROUP BY d_street_1

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

- **Conclusion**
  - 不管甚麼指令都會有 ProjectPlan、SelectPlan、TablePlan
  - Predicate 的內容就是判斷是否要 Select 的條件
  - ExplainPlan 要放加在 tree 最上面，實際呼叫順序為 ExplainPlan → ProjectPlan → (GroupByPlan → SortPlan →) SelectPlan → (ProductPlan →) TablePlan
  - Product 在 access 兩個以上的 table 才會出現