Database Homework 3 Phase 1 Report

Members

- 109062302 郭品毅
- 109062325 張庭瑋
- 109062328 吳邦寧

Explain Operation Implemention

QueryPlanner

We first add the new keyword `explain` into the keywords list in `Lexer`.
Then we add a boolean attribute `explain` in `QueryData` class to record whether `explain` operation is enabled
Also we add the ability to parse `explain` in `queryCommand()` in `Parser` to check and eat `explain`.
Last, in `BasicQueryPlanner`, if `explain` in `Querydata` is enabled, then we wrap the ProjectPlan `p` with our new implemented `ExplainPlan`.

ExplainTree

To implement ExplainPlan, we created a new data class `ExplainTree` to save the name of the plan, estimated blks, recs accessed, and optional description. Most importantly, we save the underlying ExplainTrees in an ArrayList, in order to save the hierarchy of the tree.

Plans

In `ExplainPlan`, We use `schema.addField()` to add a field `query-plan` to store and allowing access to the explain result

For each `Plan`, we add a new method `explainTree()`, which warps the underlying explan trees, and return the explain tree data with the name of the plan, estimated number of blocks and records for each plan.

ExplainScan

In `generateExplainString()`, we implement `getRecursiveExplainString()` recursively to get all the plans' details
In `getRecursiveExplainString()`, we recursively get the name of the plan, estimated

blks, recs accessed in each plan layer, and formatted them as `"-> %s %s (#blks=%d, #recs=%d)\n"`. Additionally, we pass the depth into `indentHierarchy()` to add the tabs according to the depth.
In order to get actual record count, we implemented `getActualRecordCount()`, in which we use `next()` to actually run the query to count how many records the query actually returns, and place it at the end of the result with the format `"Actual #recs: %d\n"`.
Finally, we store the result in `explainString`, then output it when `getVal()` is called.

Query Results

- A query accessing single table with `WHERE`
  `select d_name from district where d_id > 8`

```
SQL> select d_name from district where d_id > 8

    d_name
----------
 xnmoHZyes
  bJvzPLq7

SQL> explain select d_name from district where d_id > 8

query-plan
------------------------------------------------------------
-> ProjectPlan  (#blks=2, #recs=0)
    -> SelectPlan pred:(d_id>8.0) (#blks=2, #recs=0)
        -> TablePlan on (district) (#blks=2, #recs=10)
Actual #recs: 2
```

- A query accessing multiple table with `WHERE`
  `select d_name, w_name from district, warehouse where d_id < 3`

```
SQL> select d_name, w_name from district, warehouse where d_id < 3

    d_name    w_name
--------------------
 7U3ETsOrZ  3drWaGjn
    U9MvUF  3drWaGjn

SQL> explain select d_name, w_name from district, warehouse where d_id < 3

query-plan
-------------------------------------------------------------------------------
-> ProjectPlan  (#blks=22, #recs=0)
    -> SelectPlan pred:(d_id<3.0) (#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: 2
```

- A query with `ORDER BY`
  `select d_id, d_name from district where d_id < 4 order by d_name DESC`

```
SQL> select d_id, d_name from district where d_id < 4 order by d_name DESC

  d_id    d_name
---------------
     3fYFCOyj8PM
     2    U9MvUF
     1 7U3ETsOrZ

SQL> explain select d_id, d_name from district where d_id < 4 order by d_name DESC

query-plan
-------------------------------------------------------------------------------
-> SortPlan  (#blks=0, #recs=0)
    -> ProjectPlan  (#blks=2, #recs=0)
        -> SelectPlan pred:(d_id<4.0) (#blks=2, #recs=0)
            -> TablePlan on (district) (#blks=2, #recs=10)
Actual #recs: 3
```

- A query with `GROUP BY` and at least one aggregation function (`MIN`, `MAX`, `COUNT`, `AVG`... etc.)
  `select count(d_id) from district, warehouse where d_w_id = w_id group by w_id`

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

countofd_id
------
    10

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
```