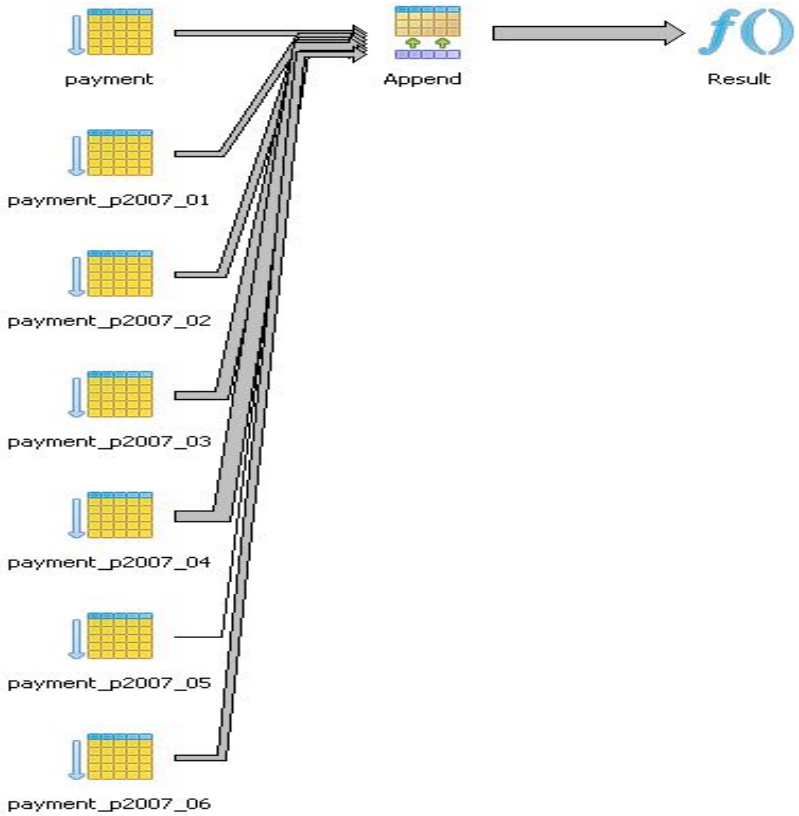
DB Visualizer

Vikas Adiwal
17201

DB Visualizer:

Problem Statement:

Recreate the functionality of Explain() found in PGADMIN for MongoDB



A Sample Explain Result in PGADMIN

Result (cost=0.00..420.63 rows=54 width=31) (actual time=0.262..5.193 rows=37 loops=1)

-> Append (cost=0.00..420.63 rows=54 width=31) (actual time=0.261..5.173 rows=37 loops=1)

-> Seq Scan on payment (cost=0.00..29.95 rows=7 width=31) (actual time=0.001..0.001 rows=0 loops=1)

Filter: ((payment_date >= '2007-02-01 00:00:00'::timestamp without time zone)

AND (payment_date <= '2007-02-15 00:00:00'::timestamp without time zone))

-> Seq Scan on payment_p2007_01 payment (cost=0.00..26.36 rows=1 width=28) (actual time=0.244..0.244 rows=0 loops=1)

Filter: ((payment_date >= '2007-02-01 00:00:00'::timestamp without time zone)

AND (payment_date <= '2007-02-15 00:00:00'::timestamp without time zone))

Structure of result returned by explain in MongoDB

```
{
  "queryPlanner" : {
    "plannerVersion" : <int>,
    "namespace" : <string>,
    "indexFilterSet" : <boolean>,
    "parsedQuery" : {
      ...
    },
    "winningPlan" : {
      "stage" : <STAGE1>,
      ...
      "inputStage" : {
        "stage" : <STAGE2>,
        ...
        "inputStage" : {
          ...
        }
      }
    },
    "rejectedPlans" : [
      <candidate plan 1>,
      ...
    ]
  }
```

```
"executionStats" : {
  "executionSuccess" : <boolean>,
  "nReturned" : <int>,
  "executionTimeMillis" : <int>,
  "totalKeysExamined" : <int>,
  "totalDocsExamined" : <int>,
  "executionStages" : {
    "stage" : <STAGE1>
    "nReturned" : <int>,
    "executionTimeMillisEstimate" : <int>,
    "works" : <int>,
    "advanced" : <int>,
    "needTime" : <int>,
    "needYield" : <int>,
    "saveState" : <int>,
    "restoreState" : <int>,
    "isEOF" : <boolean>,
    ...
    "inputStage" : {
      "stage" : <STAGE2>,
      ...
```

```
      "nReturned" : <int>,
      "executionTimeMillisEstimate" :
<int>,
      "keysExamined" : <int>,
      "docsExamined" : <int>,
      "saveState" : <int>,
      "restoreState" : <int>,
      ...
    }
  },
  "allPlansExecution" : [
    { <partial executionStats1> },
    { <partial executionStats2> },
    ...
  ]
}
"serverInfo" : {
  "host" : <string>,
  "port" : <int>,
  "version" : <string>,
  "gitVersion" : <string>
}
```

mongoDB explain() Explained...

Explain Results

The explain results present the query plans as a tree of stages. Each stage passes its results (i.e. documents or index keys) to the parent node. The leaf nodes access the collection or the indices. The internal nodes manipulate the documents or the index keys that result from the child nodes. The root node is the final stage from which MongoDB derives the result set.

Stages are descriptive of the operation eg:

- COLLSCAN for a collection scan
- IXSCAN for scanning index keys
- FETCH for retrieving documents

Major Fields returned by explain:

- **Query Planner**
- **Execution Stats**
- **Server Info**

Consider a collection `inventory` with the following documents:

copy

```
{ "_id" : 1, "item" : "f1", type: "food", quantity: 500 }
{ "_id" : 2, "item" : "f2", type: "food", quantity: 100 }
{ "_id" : 3, "item" : "p1", type: "paper", quantity: 200 }
{ "_id" : 4, "item" : "p2", type: "paper", quantity: 150 }
{ "_id" : 5, "item" : "f3", type: "food", quantity: 300 }
{ "_id" : 6, "item" : "t1", type: "toys", quantity: 500 }
{ "_id" : 7, "item" : "a1", type: "apparel", quantity: 250 }
{ "_id" : 8, "item" : "a2", type: "apparel", quantity: 400 }
{ "_id" : 9, "item" : "t2", type: "toys", quantity: 50 }
{ "_id" : 10, "item" : "f4", type: "food", quantity: 75 }
```

Query with No Index

The following query retrieves documents where the `quantity` field has a value between `100` and `200`, inclusive:

```
db.inventory.find( { quantity: { $gte: 100, $lte: 200 } } )
```

copy

The query returns the following documents:

```
{ "_id" : 2, "item" : "f2", "type" : "food", "quantity" : 100 }
{ "_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }
{ "_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }
```

copy

To view the query plan selected, chain the `cursor.explain("executionStats")` cursor method to the end of the **find** command:

copy

```
db.inventory.find(
   { quantity: { $gte: 100, $lte: 200 } }
).explain("executionStats")
```

`explain()` returns the following results:

copy

```
{
   "queryPlanner" : {
         "plannerVersion" : 1,
         ...
         "winningPlan" : {
            "stage" : "COLLSCAN",
            ...
         }
   },
   "executionStats" : {
      "executionSuccess" : true,
      "nReturned" : 3,
      "executionTimeMillis" : 0,
      "totalKeysExamined" : 0,
      "totalDocsExamined" : 10,
      "executionStages" : {
         "stage" : "COLLSCAN",
         ...
      },
      ...
   },
   ...
}
```

Query with Index

To support the query on the `quantity` field, add an index on the `quantity` field:

```
db.inventory.createIndex( { quantity: 1 } )
```

copy

To view the query plan statistics, use the `explain("executionStats")` method:

```
db.inventory.find(
   { quantity: { $gte: 100, $lte: 200 } }
).explain("executionStats")
```

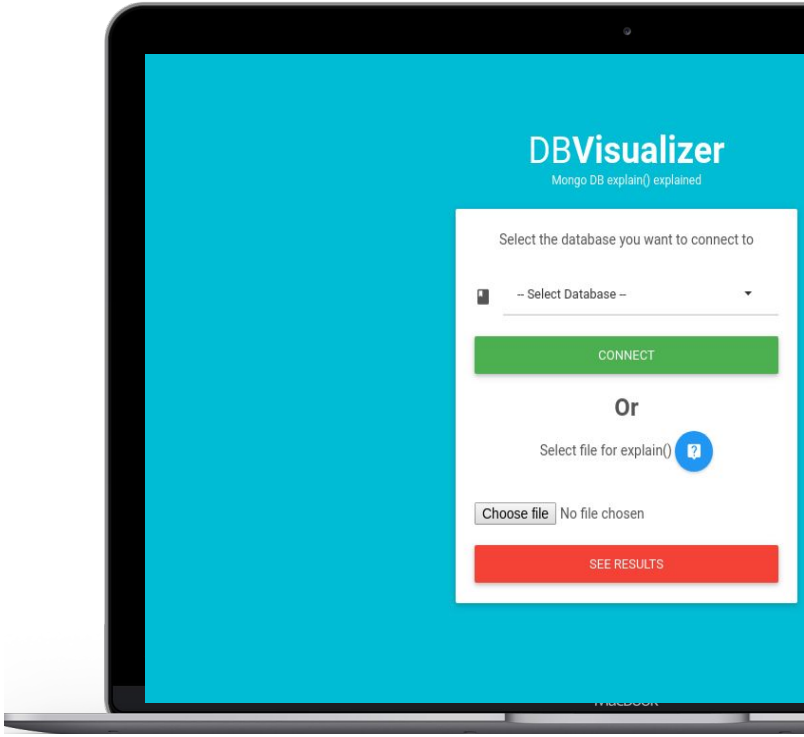
copy

The `explain()` method returns the following results:

copy

```
{
   "queryPlanner" : {
         "plannerVersion" : 1,
         ...
         "winningPlan" : {
               "stage" : "FETCH",
               "inputStage" : {
                  "stage" : "IXSCAN",
                  "keyPattern" : {
                     "quantity" : 1
                  },
                  ...
               }
         },
         "rejectedPlans" : [ ]
   },
   "executionStats" : {
         "executionSuccess" : true,
         "nReturned" : 3,
         "executionTimeMillis" : 0,
         "totalKeysExamined" : 3,
         "totalDocsExamined" : 3,
         "executionStages" : {
            ...
         },
         ...
   },
   ...
}
```

Demonstration Time



THANKS!