

CS411 MP7 Visualizing Query Execution

Yuetong Liu
Department of Statistics

In this MP I wrote a Java file called **VisualizationSimpleDB.java**, located in **SimpleDB_2.10/studentClient/simpledb/**, to hack into the SimpleDB system and display its query execution process. Besides, I also achieved some extension to SimpleDB to make up its limitation compared with MySQL. The purpose of this MP is to show how a query gets processed step by step in a real DBMS system. This document contain the following sections:

1. Visualizing and Functionality achieved
2. Implementation
3. Introduction to usage
4. Test cases

1 Visualizing and Functionality Achieved

The purpose of this MP is to visualize the query execution process in SimpleDB and output intermediate results on each node of a query tree. The original SimpleDB can only achieve basic functions in SQL query. For example, a basic SFW query (SELECT... FROM... WHERE...) may contain 4 kinds of query plan nodes in a query tree. First TablePlan creates plans for the tables in "FROM" clause and scans them; Then ProductPlan joins those tables if the number of tables is greater than 1; Next, SelectPlan filters the tuples that meet the conditions on "WHERE" clause; Last, ProjectPlan only selects the fields required in "SELECT" clause.

Besides, I also extended the functionality of SimpleDB and achieved some clauses in MySQL query. There will be corresponding query plan of these functionality in the query tree and in the right order. The summary of extended functionality is list below.

1. Group by – GroupByPlan
2. Order by – OrderPlan
3. Limit – LimitPlan
4. * (SELECT all)

2 Implementation

My solution to achieve visualizing query execution in SimpleDB is through a client code called **VisualizationSimpleDB.java**, located in SimpleDB_2.10/studentClient/simpledb/. It provides a SQL-like interface to interpret the query string, and gives corresponding query tree and intermediate results.

First, it creates a **Driver** object to connect the local database. Then read in the query string. When execute a query string, it creates a **Statement** object and use its **executeQuery()** function to execute the query string and returns a **ResultSet** object. From this ResultSet, the result tuples will be returned one at a time.

To display the intermediate results, I first pasered the query string and got a **QueryData** object. From this object, we can extract fields, tables, pred, groupFields, DescSortFields (sortFields with their desc situations), and Limit. Then according to their query order to reorganize the query string to return intermediate results.

3 Introduction to Usage

To use the VisualizationSimpleDB, first we need to conduct the database. Take the StudentDB as example, first we need to create StudentDB database, using the following command:

```
#!/bin/bash
jar cf simpledb.jar simpledb/*/*.class simpledb/*/*/*.class
cp simpledb.jar studentClient/simpledb/
cd studentClient/simpledb
java -cp simpledb.jar:. CreateStudentDB
```

Then to launch the VisualizationSimpleDB, you need to execute this command to start the MySQL-like interface:

```
java -cp simpledb.jar:. VisualizationSimpleDB
```

Next in the MySQL-like interface, you can enter your query string and get corresponding query tree and intermediate results. This advanced SimpleDB supports the clauses like **SELECT**, **FROM**, **WHERE**, **GROUP BY**, **ORDER BY**, **LIMIT** and the keywords and aggregation functions **COUNT()**, **AVG()**, **MIN()**, **MAX()**, **SUM()**, **DESC**, *****.

4 Test Cases

Test case 1:

```
SELECT SName, DName
FROM Student, Dept
WHERE MajorId = DID AND GradYear = 2004;
```

Test case 2:

```
SELECT COUNT(sid), majorid, gradyear
FROM Student
GROUP BY MajorId, GradYear
```

```
SQL> select sname, dname, gradyear from student, dept where majorid = did and gradyear=2004;

The Query Tree:
Node 1: TablePlan on table student
Node 2: TablePlan on table dept
Node 3: ProductPlan from Node 1,2,3
Node 4: SelectPlan from Node 3
Node 5: ProjectPlan from Node 4

Node 1 Outputs:
select * from student
majorid gradyear sname sid
-----
10 2004 joe 1
20 2004 amy 2
10 2005 max 3
20 2005 sue 4
30 2003 bob 5
20 2001 kim 6
30 2004 art 7
20 2001 pat 8
10 2004 lee 9

Node 2 Outputs:
select * from dept
did dname
-----
10 compsci
20 math
30 drama

Node 3 Outputs:
select * from student, dept
majorid gradyear sname dname did sid
-----
10 2004 joe compsci 10 1
10 2004 joe math 20 1
10 2004 joe drama 30 1
20 2004 amy compsci 10 2
20 2004 amy math 20 2
20 2004 amy drama 30 2
10 2005 max compsci 10 3
10 2005 max math 20 3
10 2005 max drama 30 3
20 2005 sue compsci 10 4
20 2005 sue math 20 4
20 2005 sue drama 30 4
30 2003 bob compsci 10 5
30 2003 bob math 20 5
30 2003 bob drama 30 5
20 2001 kim compsci 10 6
20 2001 kim math 20 6
20 2001 kim drama 30 6
30 2004 art compsci 10 7
30 2004 art math 20 7
30 2004 art drama 30 7
20 2001 pat compsci 10 8
20 2001 pat math 20 8
20 2001 pat drama 30 8
10 2004 lee compsci 10 9
10 2004 lee math 20 9
10 2004 lee drama 30 9

Node 4 Outputs:
select * from student, dept where majorid=did and gradyear=2004
majorid gradyear sname dname did sid
-----
10 2004 joe compsci 10 1
20 2004 amy math 20 2
30 2004 art drama 30 7
10 2004 lee compsci 10 9

Node 5 Outputs:
select sname, dname, gradyear from student, dept where majorid=did and gradyear=2004
gradyear sname dname
-----
2004 joe compsci
2004 amy math
2004 art drama
2004 lee compsci
```

(a) Test case 1

```
SQL> select count(sid), majorid, gradyear from student group by majorid,gradyear;

The Query Tree:
Node 1: TablePlan on table student
Node 2: GroupByPlan from Node 1
Node 3: ProjectPlan from Node 2

Node 1 Outputs:
select * from student
majorid gradyear sname sid
-----
10 2004 joe 1
20 2004 amy 2
10 2005 max 3
20 2005 sue 4
30 2003 bob 5
20 2001 kim 6
30 2004 art 7
20 2001 pat 8
10 2004 lee 9

Node 2 Outputs:
select * from student group by majorid, gradyear
gradyear majorid
-----
2004 10
2005 10
2001 20
2004 20
2005 20
2003 30
2004 30

Node 3 Outputs:
select countofsid, majorid, gradyear from student group by majorid, gradyear
countofsid majorid gradyear
-----
2 10 2004
1 10 2005
2 20 2001
1 20 2004
1 20 2005
1 30 2003
1 30 2004
```

(b) Test case 2

Test case 3:

```
SELECT *
FROM Student, Dept
WHERE MajorId = DID
ORDER BY gradyear DESC, dname
```

SQL> select * from student,dept where majorid=did order by gradyear desc, dname

The Query Tree:
Node 1: TablePlan on table student
Node 2: TablePlan on table dept
Node 3: ProductPlan from Node 1,2
Node 4: SelectPlan from Node 3
Node 5: OrderByPlan from Node 4
Node 6: ProjectPlan from Node 5

Node 1 Outputs:
select * from student
majorid gradyear sname sid

10 2004 joe 1
20 2004 amy 2
10 2005 max 3
20 2005 sue 4
30 2003 bob 5
20 2001 kim 6
30 2004 art 7
20 2001 pat 8
10 2004 lee 9

Node 2 Outputs:
select * from dept
did dname

10 compsci
20 math
30 drama

Node 3 Outputs:
select * from student, dept
majorid gradyear sname dname did sid

10 2004 joe compsci 10 1
10 2004 joe math 20 1
10 2004 joe drama 30 1
20 2004 amy compsci 10 2
20 2004 amy math 20 2
20 2004 amy drama 30 2
10 2005 max compsci 10 3
10 2005 max math 20 3
10 2005 max drama 30 3
20 2005 sue compsci 10 4
20 2005 sue math 20 4
20 2005 sue drama 30 4
30 2003 bob compsci 10 5
30 2003 bob math 20 5
30 2003 bob drama 30 5
20 2001 kim compsci 10 6
20 2001 kim math 20 6
20 2001 kim drama 30 6
30 2004 art compsci 10 7
30 2004 art math 20 7
30 2004 art drama 30 7
20 2001 pat compsci 10 8
20 2001 pat math 20 8
20 2001 pat drama 30 8
10 2004 lee compsci 10 9
10 2004 lee math 20 9
10 2004 lee drama 30 9

(a) Test case 3 (1/2)

Node 4 Outputs:
select * from student, dept where majorid=did
majorid gradyear sname dname did sid

10 2004 joe compsci 10 1
20 2004 amy math 20 2
10 2005 max compsci 10 3
20 2005 sue math 20 4
10 2004 lee compsci 10 5
30 2004 art drama 30 7
20 2001 kim math 20 6
30 2004 art drama 30 8
20 2001 pat math 20 9
10 2004 lee compsci 10 9

Node 5 Outputs:
select * from student, dept where majorid=did order by gradyear desc, dname
majorid gradyear sname dname did sid

10 2005 max compsci 10 3
20 2005 sue math 20 4
10 2004 lee compsci 10 5
10 2004 joe compsci 10 1
30 2004 art drama 30 7
20 2004 amy math 20 2
30 2003 bob drama 30 5
20 2001 pat math 20 8
20 2001 kim math 20 6

Node 6 Outputs:
select * from student, dept where majorid=did order by gradyear desc, dname
majorid gradyear sname dname did sid

10 2005 max compsci 10 3
20 2005 sue math 20 4
10 2004 lee compsci 10 5
10 2004 joe compsci 10 1
30 2004 art drama 30 7
20 2004 amy math 20 2
30 2003 bob drama 30 5
20 2001 pat math 20 8
20 2001 kim math 20 6

(b) Test case 3 (2/2)