Graphical user interface, application

Description automatically generated  
CS3223 Database Systems Implementation

AY21/22 Semester 2

Project Report Lab 2

Team 37

Prof Tan Kian-Lee

|  |  |  |
| --- | --- | --- |
| Team Members | Student No. | Email |
| Kelvin Wong | A0201706U | [e0415515@u.nus.edu](mailto:e0415515@u.nus.edu) |
| Moon Geonsik | A0210908L | [e0484312@u.nus.edu](mailto:e0484312@u.nus.edu) |
| Wincent Tjoi | A0201480W | [e0412905@u.nus.edu](mailto:e0412905@u.nus.edu) |

Lab 2

Parser Related Changes

|  |  |
| --- | --- |
| **File Changes** | **Changes** |
| SimpleDBEngine/src/simpledb/parse  /**Lexer**.java | * added keywords “order” and “by” to the keywords list (in routine **initKeywords()**) * created a list of sort type (“asc,” “desc”) (in routine **initSortType()**) * added a new routine **eatSortType()** to take in the sort type and return a boolean value (“asc”=>true; “desc”=>false;) * added a new routine **matchSortType()** to check if the current token is a legal sort type (“asc,” “desc”) |
| SimpleDBEngine/src/simpledb/parse  /**Parser**.java | * modified routine **query()** to detect the “order by” keyword and call **sortList()** * added a new routine **sortList()** to parse the “order by” clause and to initialize and return a **LinkedHashMap<String, Boolean>** containing <key, value> pairs of the field to be sorted and whether it should be sorted in an ascending or descending order (order is set to ascending by default) |

Planner Related Changes

|  |  |
| --- | --- |
| **File Changes** | **Changes** |
| SimpleDBEngine/src/simpledb/  opt/**HeuristicQueryPlanner**.java | * modified the routine **createPlan()** to initialize and return a **SortPlan** object, if the query contains an “order by” clause. otherwise, return a **ProjectPlan** object |

Sorting Related Changes

|  |  |
| --- | --- |
| **File Changes** | **Changes** |
| SimpleDBEngine/src/simpledb/  materialize/**SortPlan**.java | * modified the constructor of **SortPlan** object that iterates over **sortfields** to assign a true Boolean value to each field as a key to create a **Map<String, Boolean>** object and initialize a **RecordComparator** object with the Map object * created a new constructor for **SortPlan** object that takes in a **Map<String, Boolean>** object, instead of a **List<String>** object, and initialize a **RecordComparator** object with the Map object |
| SimpleDBEngine/src/simpledb/  materialize/**RecordComparator**.java | * modified the constructor of **RecordComparator** object to take in a **Map<String, Boolean>** object and initialize the Map object as **sortMap**. Also, extract the keyset of the Map object as a **List<String>** and initialize the List object as **fields** * modified the routine **compare()** to check if the current field that the routine is iterating over should be sorted in an “ascending” or “descending” order using the **sortMap** and return the integer result. (if the order is descending, change the sign of the integer result and return) |