|  |  |
| --- | --- |
| **Files to change** | **Changes** |
| simpledb/test/CreateStudentDB.java | 1. Added code to create student index on MajorId, in between the creation and insertion of/into the STUDENT table.  * Similar approach to creating a new table, just that we substitute the string with "create index idx\_MajorId on STUDENT(MajorId);"  1. Added code to create enroll index on StudentId, in between the creation and insertion of/into the ENROLL table.  * Same idea as above.  1. After implementing the support for multiple index structures, the code in Point 1 is then modified such that it uses either btree/hash index  * By adding “using btree/hash” at the end of the original code. |
| simpledb/server/SimpleDB.java | 1. Switched to the appropriate pair of (QueryPlanner, UpdatePlanner), as specified in Question 2. |
| simpledb/index/planner/IndexUpdatePlanner.java | 1. Added an additional string argument in mdm.createIndex().  * This field represents the type of index. |
| simpledb/metadata/IndexInfo.java | 1. Added an additional string field in the class as well as the constructor  * This field represents the type of index.  1. Modified open() to take into account the type of index when creating either a Btree index or a Hash index.  * By checking using type.equals(“btree”) * Or type.equals(“hash”) * Prints the type of index as well to allow us to verify that the multiple index structures are working. |
| simpledb/metadata/IndexMgr.java | 1. Added an additional string field to the schema in the constructor  * This field represents the type of index.  1. Added an additional string argument to the createIndex() function  * This argument represents the type of index.  1. Added an additional string argument when creating a new IndexInfo in getIndexInfo()  * This argument represents the type of index. |
| simpledb/metadata/MetadataMgr.java | 1. Added an additional string argument to the createIndex() function  * This field represents the type of index. |
| simpledb/parse/CreateIndexData.java | 1. Added an additional string field in the class as well as the constructor  * This field represents the type of index.  1. Created a new getter function (getIndexType()) that allows us to retrieve the type of index in a String. |
| simpledb/parse/Lexer.java | 1. Created a new eatType() function that allows us to check if the user has specified a valid index type  * Works similarly to eatId() * Throws BadSyntaxException() if the value of token does not match with either keywords (hash/btree).  1. Include new keywords in the initKeywords() function [“hash”, “btree”, “using”] |
| simpledb/parse/Parser.java | 1. Extended the createIndex() function to allow support for multiple index structures.  * “eat” the “using” keyword after eatDelim(‘)’) * Then, proceed to extract the type of index that the user specified, using our newly created eatType() function as explained above. * Lastly, we return a new instance of CreateIndexData with the following fields:   + Idxname   + Tblname   + Fldname   + Type (type of index) |
| simpledb/plan/BasicUpdatePlanner.java | 1. Added an additional string argument in mdm.createIndex(). 2. This field represents the type of index. |

Default setting for original program:

**Hash** (according to ~line 50 of simpledb/metadata/IndexInfo.java)