CSE 510 - Database Management System Implementation

Spring 2022

Phase III

Due Date: Monday, April 25th

1 Goal

The version of the MiniBase I have distributed to you implements various modules of a relational database management system. Our goal this semester is to use these modules of MiniBase as building blocks for implementing an *RDF DBMS*.

2 Project Description

The following is a list of tasks that you need to perform for this final phase of the project:

- Modify Minibase's `Tuple` class into a `BasicPatternClass`. The `BasicPatternClass` will store one or more nodeIDs. Unlike the `Tuple` class, the `BasicPatternClass` will have a default field `Confidence`.
- Modify Minibase's `Iterator` class into a `BPIterator` class that operates over basic patterns.
- Implement a `BP_Triple_Join` class with the following constructor
  - `BP_Triple_Join(int amt_of_mem, int num_left_nodes, BPIterator left_itr, int BPJoinNodePosition, int JoinOnSubjectorObject, java.lang.String RightSubjectFilter, java.lang.String RightPredicateFilter, java.lang.String RightObjectFilter, double RightConfidenceFilter, int [] LeftOutNodePositions; int OutputRightSubject, int OutputRightObject)`, where

```
amt_of_mem - available pages for theoperation
num_left_nodes - the number of node IDs in the left basic pattern stream
BPIterator left_itr -  the left basic pattern stream
BPJoinNodePosition - the position of the join node in the basic pattern
JoinOnSubjectorObject - 0: join on subject; 1: join on object
RightSubjectFilter - subject filter for the right source
RightPredicateFilter - predicate filter for the right source
RightObjectFilter - object filter for the right source
RightConfidenceFilter - confidence filter for the right source
int[] LeftOutNodePositions - positions of the projected nodes from the
                             left source
int OutputRightSubject - 0/1 project subject node from the right source?
int OutputRightObject - 0/1 project object node from the right source?
```

The basic patterns in the given `left_itr` are joined with the RDF triples in the data store based on subject or object node ID. The confidence of a resulting basic pattern is the *minimum* of the confidences of the contributing (left) basic pattern and (right) triple. The class should provide `get_next()` and `close()` methods (see NestedLoopJoins).

- Implement an external `BPSort` operator that sorts the basic patterns in a given `BPIterator` according to the given criterion,

```
BPSort( BPIterator input_itr, BPOrder sort_order,
        int SortNodeIDPos,
        int n_pages)
```

  Here `BPOrder` is similar to *TupleOrder* of minibase. If `SortNodeIDPos` is equal to "-1", then the basic patterns will be sorted based on the confidence field. Otherwise, the basic patterns will be sorted based on the labels of the nodes specified in the corresponding position in the basic pattern. Like its Minibase counterpart, `BPSort` needs to support a `getNext()` method that returns the resulting basic patterns in the specified order.

- Implement a command-line program `query`. Given the command line invocation

```
query RDFDBNAME QUERYFILE NUMBUF
```

  the program will process the query specified in the `QUERYFILE` against the rdf database specified in `RDFDBNAME`, using 3 different strategies. Minibase will use at most `NUMBUF` buffer pages to run the query for each of the strategies (see the Class `BufMgr`).

  The format of the query specification file will be follows:

```
S( J(
     J([SF1,PF1,OF1,CF1],
       JNP,JONO,RSF,RPF,ROF,RCF,LONP,ORS,ORO
      ),
     JNP,JONO,RSF,RPF,ROF,RCF,LONP,ORS,ORO
     )
   SO, SNP, NP
 )
```

- Implement a command line program `report` which outputs the various statistics for the RDF database.

- You will evaluate and model the performances (in terms of page accesses) of the query execution strategies. The variables of interest could include
  - RDF database statistics,
  - page size,
  - buffer size,
  - join selectivities, etc.

Your report should include graphs depicting the performance of the operators as a function of the relevant parameters. Make sure that you have at least 5 experiment sets. Whenever possible, try to develop a formula that matches/explains the graphs. Explain in detail the performance behavior of these operators.

At the end of each operation, the program should also output the number of disk pages that were read and written (separately).

3 Deliverables

You will be provided with a sample data set. You have to return the following before the deadline:

- Your source code properly **commented**, `tar`ed and `zip`ed.
- The output of your program with the provided test data.
- A report. The report should contain a short description of the operators. The report should analytically model and compare the performances of various operators, implemented throughout the semester, through experiments. The report should also explain the observed behaviors.
- The report should clearly state *who did what*. This will be taken very seriously! So, be honest. Be prepared to explain on demand (not only your part) but the entire set of modifications. See the report specifications.
- A confidential document (individually submitted by each group member) which rates group members' contributions out of 10 (10 best; 0 worst). Please provide a brief explanation for each group member.