Database Systems WS 2023/24
Exercise 6: Distributed 27.11.2023, Due 04.12.2023 12:00 MEZ
Submitted by Erik Schwede, Suma Keerthi

1 Denormalization and rewriting (1 P.)

For this question, we will take another look at the uni_db schema from sheet 2:

1. How can you modify/denormalize this schema, to answer the following query efficiently? Which downsides will this have?
   Describe the normal forms of the original and changed schema in your answer. Also note which additional steps have to be taken to keep the data consistent.

   Required submission: Denormalized table schema; Explanation and downsides of denormalization; Query after denormalization; Explanation on how to keep data consistent; Normal form of original table schema; Normal form of new table schema.

   - Q1: A list of all student names and the lecture title and SWS points of all the lectures they attend.

2. Rewrite the following queries to more efficient queries, returning the same result set. Consider the conditions noted for each of the queries.

   (a)
   ```
   SELECT DISTINCT *
   FROM Professors
   ```

   (b)
   ```
   SELECT COUNT(DISTINCT l.LID)
   FROM Prerequisites p, Lectures l
   WHERE p.Required = l.LID
   ```

   (c)
   ```
   SELECT COUNT(HeldBy)
   FROM Lectures
   GROUP BY LID
   HAVING LID = 5001
   ```

3. Can the view be used to execute the query? Explain in detail why the view can be used or why it cannot be used.

   Query
   ```
   SELECT MatrNr
   FROM Exam
   WHERE Grade < 3
   ```

   View
   ```
   SELECT MatrNr, PID
   FROM Exam
   WHERE Grade = 1
   ```

Database Systems WS 2023/24
Exercise 6: Distributed 27.11.2023, Due 04.12.2023 12:00 MEZ
Submitted by Erik Schwede, Suma Keerthi

2 Min-Hashing (1 P.)

Implement Min-Hashing in a language of your choice. Your program has to take two text files as input and parse them into sets of words. The program has to calculate and display:

- Jaccard coefficient between the two texts.
- The similarity estimated by Min-Hashing with k different hash functions (for each $k \in [1, 6]$)
- The similarity estimated by Min-Hashing with k different min values of one hash function (for each $k \in [1, 6]$)

You can use the template in OLAT, which already parses the files and provides 6 hash functions. Submit the code and the output of your program when executed with the two data files provided in OLAT. If you do not use the template, also submit instructions on how to compile and execute your program. Do not submit the data files.

3 Quadtrees (1 P.)

1. Give 8 points, with coordinates where each dimension has values in the range $[0, 100]$, and sort them into two insertion orders. The first should have less empty leaf nodes if inserted into a quadtree than if they were inserted into a PR quadtree. The second one the other way around.

   Submit the points, the two insertion orders and draw the (PR) quadtrees. You can either draw them as tree or as the grid visualization.

2. Assuming we have the whole dataset present upon insertion, describe one possible way that optimizes the creation of the quadtree, such that the height of the resulting quadtree is minimized. Note that before the bulk-loading, preprocessing of the data is allowed.

3. Assuming we are given uniformly distributed point data, what is the probability that a node at depth k contains a particular point in a PR Quad tree? Additionally, for a collection of v points, calculate the probability that none of the points lies in a given cell at depth k?

4 kd tree (1 P.)

1. Insert the following values, in provided order, into an empty kd tree, with $k = 3$:
   1. (19, 31, 83)
   2. (64, 27, 97)
   3. (66, 8, 101)
   4. (99, 86, 34)
   5. (12, 92, 55)
   6. (76, 39, 85)
   7. (85, 68, 4)
   8. (98, 60, 59)
   9. (13, 52, 9)
   10. (44, 39, 91)
   11. (29, 35, 57)
   12. (43, 15, 98)

   Draw the result as a tree, like shown in the lecture. Note which values you used for the split.

2. Adaptive kd tree

   Provide **in detail** an algorithm, in pseudo code, that takes a list of coordinates in k dimensions and builds a kd tree. This tree may have a maximum of ten data points in its leaves. The dimension used for the split should be the one with the largest variance. The value used for the split, should be the average of the split dimension of all relevant coordinates.