Junior Professorship Data Stream Management and Analysis
RWTH AACHEN UNIVERSITY

Prof. Dr. Sandra Geisler
Soo-Yon Kim, Liam Tirpitz

November 28, 2022

Implementation of Databases (Winter Term 2022/2023)

Exercise 4

Due until Monday, December 12 at 23:59. Please submit your solutions as a Jupyter notebook to Moodle. Please do *not* submit handwritten solutions!
Please submit your solutions in groups of three. Solutions to this exercise will be presented on Friday, December 16.

Group members: [Name, matriculation number], [name, matriculation number], [name, matriculation number]

Insert all group members by double-clicking on this cell.

Exercise 4.1 (Cost-Based Optimization) (12 pts)

Given is a relational schema with two relations $R(A, B, C), S(C, D)$. The following information about the relations, selectivities, and indexes is given:

- $|R| = |S| =$2.000.000 tuples
- A page can hold 200 tuples of R, or 20 tuples of S.
- The following selectivities are given:
  $F(R.C = S.C) = 10^{-10}$ (for each value in $R.C$, there is at most one matching tuple in $S.C$)
  $F(R.B = 20) = 10^{-3}$
  $F(S.D < 60) = 10^{-1}$
- There are the following indexes:
  A clustered B+ tree index on $R.B$.
  An unclustered B+ tree index on $S.C$.
  An unclustered B+ tree index on $S.D$.

The cost for navigating to a leaf node in a B+ tree index is 3 I/O operations. Consider in the following questions only the costs for reading the relations or writing intermediate results, but not the costs for writing the final result.

1. Compute the number of required I/O operations and the size of the results in terms of tuples and pages for the following operations:

- $\sigma_{B=20}(R)$ using the clustered index on $R.B$.
- $\sigma_{B=20}(R)$ using a relational scan of R (note: the relation is sorted according to $R.B$).
- $\sigma_{D<30}(S)$ using the unclustered index on $S.D$.

[Please replace this text with your answer.]

2. Given is the following query:

```
SELECT DISTINCT R.* FROM R,S
WHERE R.C=S.C AND R.B=20 AND S.D<60
```

Suppose that you can only use an index nested loop join to evaluate the query, using the indexes described above. Taking into account the results for the previous subtask, what would be the optimal strategy to execute the query? Give the costs for this strategy.

[Please replace this text with your answer.]

Exercise 4.2 (Map Reduce) (6 pts)

Consider documents containing the discography of artists in the following way (“duration” indicates the length of a track in seconds):

```
{
  "albumTitle": "Greatest Hits",
  "songs": [
    {
      "trackName": "Can't Stop",
      "duration": 269,
      "artistName": "Red Hot Chili Peppers"
    }
  ]
}
```

Provide a sketch for the MapReduce implementation of the following query using some form of Java/- JavaScript pseudocode, similar to the syntax used in the lecture.

1. Number of occurrences for each song across all albums. Two tracks are considered the same song, if their 'artistName', 'trackName' and 'duration' are identical.
2. For each artist, the number of unique songs which are longer than 2 minutes. You may use the result from the previous subtask.

[Please replace this text with your answer.]

Exercise 4.3 (NoSQL) (7 pts)

Given is a database in MongoDB describing Pokémon. Express the following queries for MongoDB using the pymongo Interface. *Note: Your code must compile and return results.*

Start MongoDB

You can use the following code to start a MongoDB environment. *Note: If you use the offline Docker environment, you may need to update the Docker image to the latest version.*

In [ ]:

```
import multiprocessing, time
from pymongo import MongoClient
import bson

!mkdir -p assets/mongodb
def start_mongo():
    !mongod -dbpath assets/mongodb > /dev/null


multiprocessing.Process(target=start_mongo).start()
```

```
In [ ]: !mongorestore -d pokemon -c pokedex assets/samples_pokemon.bson
```

1. Add a new collection named `Trainers` and add a new `Trainer` with the following properties:

- Name: Misty
- Region: Kanto
- Pokemon: Sterndu, Starmie

```
In [1]: # Query 1
        # YOUR CODE HERE
```

1. Express the following queries and output their results: Return only the fields from the database required for the output.

- Output all Pokémon and their number
- Output the name of all Pokémon eventually evolving into Butterfree.
- Find all Pokémon with an id larger than 50 and smaller than 100 and sort the result by descending spawn chance. Output only the name, number and spawn chance of each Pokémon.
- How many Pokemon are both of type Water or Ice?
- For each Pokémon find all trainers who have that Pokémon and append their document to the output document. That is, return all Pokémon documents with an additional field `Trainers` , containing all documents from the `Trainers` collection, which represent a trainer with that Pokémon.

```
In [ ]: # Query 2.1
        # YOUR CODE HERE
```

```
In [ ]: # Query 2.2
        # YOUR CODE HERE
```

```
In [ ]: # Query 2.3
        # YOUR CODE HERE
```

```
In [ ]: # Query 2.4
        # YOUR CODE HERE
```

```
In [ ]: # Query 2.5
        # YOUR CODE HERE
```

[Please replace this text with your answer.]

Exercise 4.4 (Short Questions) (5 pts)

1. Explain how histograms are used for cost estimation by a query optimizer.
2. In your own words: Name and explain the three terms that CAP stands for. Name and explain what the CAP theorem states.

[Please replace this text with your answer.]