Exercise 6

Released: May 25, 2020 · Discussion: June 8, 2020

1 Outer Joins

The lecture presented the merge join and hash join algorithms to perform equijoins. But the algorithms can be also used to compute left outer joins or full outer joins if they are modified accordingly.

1. How does the merge join algorithm has to be altered to compute left outer joins? Sketch your algorithm and describe any new data structures needed.
2. Modify the hash join algorithm given in the lecture to produce a full outer join. Again describe any new data structures you introduce and sketch your algorithm.

2 Query Optimization — Dynamic Programming

Consider the following SQL query on a hypothetical flight reservation database.

```
SELECT A.name, F.airline, C.name
  FROM Airport AS A, Flight AS F, Crew AS C
 WHERE F.to = A.code
   AND F.flightNo = C.flightNo
```

Assume there is a B$^+$tree index on the attribute *code* of the relation *Airport*. Further assume that the database is running an optimizer utilizing the *Dynamic Programming* algorithm as presented in the lecture. Assume that the optimizer keeps plans with interesting orders. The database provides the join algorithms *Block Nested Loops Join* and *Index Nested Loops Join* of which the latter is considered to be the cheaper one. Additionally the system avoids cross-products.

For each pass of the optimizer show which plans are retained and which are discarded. Explain why. It is not necessary for you to show which plan is the best overall in the last pass.

3 Concurrency Control

Given the following schedule:

$$S = \langle r_3(x), r_1(x), r_2(x), w_1(y), r_2(x), w_3(x) \rangle \ .$$

Answer the following questions:

1. When are two schedules conflict equivalent?
2. What is a conflict serializable schedule?
3. Draw the **serialization graph** for S.
4. Is S **conflict serializable**? Give an explanation for your answer.

4 Two-Phase Locking

Locking protocols are typically used to ensure that only serializable schedules are allowed in a DBMS. A widely used variant of a locking protocol is the two-phase locking protocol, that was discussed in the lecture.

1. Describe the basic idea of the two-phase locking protocol.
2. What techniques do you know to deal with deadlocks in two-phase locking?
3. What is the difference between preclaiming/conservative and strict two-phase locking?

5 Join Operator Implementation

As we have seen the join operator has many implementations, which have their up and down sites. In this exercise we will implement the nested loops join and the hash join operator.

Implementation instructions:

1. Implement the methods `open()`, `next()` and `close()` of the nested loops join operator `src/execution/nested_loops_join_operator.cpp`
2. Implement the methods `build_hash_table()` and `probe_hash_table()` of the hash join operator `src/execution/hash_join_operator.cpp`
3. Run `SELECT g.name, m.title FROM movie m JOIN genre genre_movie on m.id = gm.movie_id JOIN genre g on gm.genre_id = g.id`.
4. Consider to limit the result to 1000 by adding `LIMIT 1000` to the query.
5. Enable hash join. (set `enable-hash-join` to 1 in `beedb.ini` or use the `--enable-hash-join` flag)
6. Using the hash join, running the query without limit should be fast enough.