Database Technology– FSS 2018

Exercise 9: Query Processing

9.1. Sort-merge

Apply the sort-merge algorithm to the following relation (sort it based on the first attribute). Show the runs created on each pass assuming that only one tuple fits in a block and memory holds at most 3 blocks.

<table>
<thead>
<tr>
<th>Animal</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>kangaroo</td>
<td>17</td>
</tr>
<tr>
<td>wallaby</td>
<td>21</td>
</tr>
<tr>
<td>emu</td>
<td>1</td>
</tr>
<tr>
<td>wombat</td>
<td>13</td>
</tr>
<tr>
<td>platypus</td>
<td>3</td>
</tr>
<tr>
<td>lion</td>
<td>8</td>
</tr>
<tr>
<td>warthog</td>
<td>4</td>
</tr>
<tr>
<td>zebra</td>
<td>11</td>
</tr>
<tr>
<td>meerkat</td>
<td>6</td>
</tr>
<tr>
<td>hyena</td>
<td>9</td>
</tr>
<tr>
<td>hornbill</td>
<td>2</td>
</tr>
<tr>
<td>baboon</td>
<td>12</td>
</tr>
</tbody>
</table>

9.2 Optimized relational-algebra

Consider the following bank database:

- branch(branch_name, branch_city, assets)
- customer(customer_name, customer_stree, customer_city)
- loan(loan_number, branch_name, amount)
- borrower(customer_name, loan_number)
- account(account_number, branch_name, balance)
- depositor(customer_name, account_number)

Write an efficient relational-algebra expression that is equivalent to the following SQL query:

```
SELECT T.branch_name
FROM branch T, branch S
WHERE T.assets > S.assets AND S.branch_city="Brooklyn"
```

Justify your choice.
9.3. Query costs

Let relations \( r_1(A, B, C) \) and \( r_2(C, D, E) \) have the following properties: \( r_1 \) has 20,000 tuples, \( r_2 \) has 45,000 tuples, 25 tuples of \( r_1 \) fit on one block, and 30 tuples of \( r_2 \) fit on one block. Estimate the number of block transfers and seeks required, using each of the following join strategies for \( r_1 \Join r_2 \):

a. Nested-loop join.
b. Block nested-loop join.
c. Merge join.
d. Hash join.

9.4. Query costs 2

Let \( r \) and \( s \) be relations with no indices, and assume that the relations are not sorted. Assuming infinite memory, what is the lowest-cost way (in terms of I/O operations) to compute \( r \Join s \)? What is the amount of memory required for this algorithm?

9.5. Index usage

Consider again the bank database of from exercise 9.2. Suppose that a B+-tree index on branch city is available on relation branch, and that no other index is available. List different ways to handle the following selections that involve negation:

a. \( \sigma \neg(branch\_city<"Brooklyn") (branch) \)
b. \( \sigma \neg(branch\_city="Brooklyn") (branch) \)
c. \( \sigma \neg(branch\_city<"Brooklyn" \lor assets<5000) (branch) \)

9.6. Query plans

Execute the following query on the university database from the second exercise:

```
SELECT student.ID, takes.grade
FROM student INNER JOIN takes ON(student.ID=takes.ID)
WHERE student.name = "Zhang" AND takes.year=2009
```

To do that

1. Open your browser at [http://localhost/phpmyadmin/](http://localhost/phpmyadmin/)
2. Click on the university database and open the SQL tab and execute the query
3. Click on “explain SQL”

a. Find out what the columns contain
b. Formulate the query plan in text