CSIT 5300 - Spring 2020 - HKUST
Assignment 3

Deadline: Tuesday, May 19, 2020, 11:59pm
Submission: Please submit your solutions in .pdf format on Canvas

<table>
<thead>
<tr>
<th>Student Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID</td>
<td></td>
</tr>
<tr>
<td>Email Address</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td></td>
</tr>
</tbody>
</table>

For T.A. use only

<table>
<thead>
<tr>
<th>Problem</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/ 25</td>
</tr>
<tr>
<td>2</td>
<td>/ 25</td>
</tr>
<tr>
<td>3</td>
<td>/ 25</td>
</tr>
<tr>
<td>4</td>
<td>/ 25</td>
</tr>
<tr>
<td>Total</td>
<td>/ 100</td>
</tr>
</tbody>
</table>
Problem 1 [25 points] Functional Dependencies

Consider relation \( R(A, B, C, D, E, F) \) with functional dependencies \( F = \{ A \rightarrow BC, C \rightarrow AD, E \rightarrow ABC, F \rightarrow CD, CD \rightarrow BEF, AB \rightarrow D \} \).

(a) [5 points] Consider relation \( R \) with functional dependencies \( F \). Identify the candidate key(s). Explain your answer.

Answer:

(b) [8 points] Consider relation \( R \) with functional dependencies \( F \). Identify the strongest Normal form. Explain your answer.

Answer:
(c) [12 points] Consider relation $R$ with functional dependencies $F$. Give a lossless 3NF decomposition. Explain how you derived your decomposition.

Answer:
Problem 2 [25 points] B⁺-Trees
Consider a relation with the following schema (the primary key is underlined):

\[
\text{Movie}(id, \text{title}, \text{year}, \text{producerId})
\]

(a) [8 points] We construct a B⁺-Tree on \(id\). Suppose that this is a primary (i.e., clustering) index with 200,000 index entries, and the average fanout of the B⁺-Tree is 50. What is the cost for answering the following query (in terms of page accesses, without considering the cost of writing the results)?

\[
\text{SELECT title FROM Movie WHERE id = 101}
\]

Answer:
(b) [8 points] We construct a B$^+$-Tree on year. Suppose that this is a primary (i.e., clustering) index with 200 index entries, and the average fanout of the B$^+$-Tree is 50. Assume that 1,000 movies are produced per year, each page has size 2,000 bytes, and the size of each movie tuple is 50 bytes. What is the cost for answering the following query (in terms of page accesses, without considering the cost of writing the result)?

```
SELECT title FROM Movie WHERE year >= 2000 AND year <= 2004
```

Answer:
(c) [9 points] We construct a B$^+$-Tree on producerId. Suppose that this is a dense, secondary (i.e., non-clustering) index, and that it has 3 levels. What is the cost for answering the following query (in terms of page accesses, without considering the cost of writing the result)? Assume an 80% page utilization in the index, each index data entry is 8 bytes long, and each producer has produced 500 movies. Also assume that we use the multiple index entries option for storing entries with the same search key (see Option 2 in slide 12 of L07_indexing_intro.pdf).

SELECT id FROM Movie WHERE producerId = 5

Answer:
Problem 3 [25 points] Extendible Hashing

We consider a hash index using the extendible hashing approach. We assume that each bucket contains 2 entries. In the first four questions below, you are given a figure depicting an instance of the index. For each entry, we illustrate only its search key in both binary and decimal form, omitting the pointers for clarity. The hash function simply maps the records to the buckets, using the rightmost bits of their binary representation (similar to what we have seen in L08_bplus_dynam_hashing.pdf). You are required to draw a figure showing the resulting index contents (i.e., directory, bucket contents, local and global depths) after inserting the given new entries.

(a) [5 points] You are given the following index instance. What is the result after the insertion of a record with key 17 (10001)?

Answer:
(b) [5 points] You are given the following index instance. What is the result after the insertion of two records, one with key 17 (10001) and one with 23 (10111)?

Answer:

```
<table>
<thead>
<tr>
<th>key</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>4 (100)</td>
<td>5 (101)</td>
<td>6 (110)</td>
<td>7 (111)</td>
</tr>
<tr>
<td></td>
<td>8 (1000)</td>
<td>29 (1101)</td>
<td>14 (1110)</td>
<td>11 (1011)</td>
</tr>
</tbody>
</table>
```
(c) [5 points] You are given the following index instance. What is the result after the insertion of a record with key 11 (1011)?

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4 (100)</td>
<td>8 (1000)</td>
<td>1</td>
<td>5 (101)</td>
</tr>
</tbody>
</table>

Answer:
(d) [5 points] You are given the following index instance. What is the result after the insertion of two records, both with key 4 (100)? Denote the first new entry as 4*(100), and the second as 4**(100).

Answer:

(e) [5 points] Explain briefly why B⁺-Tree is preferred over multi-level tree (see slide 6 of L07_indexing_intro.pdf) in DBMS?

Answer:
Problem 4 [25 points] Sorting and Join Algorithms

Consider the following two relations, and the following SQL query:

- $R(A, B, C, D)$: 25,000 records
- $S(A, E)$: 40,000 records

```
SELECT R.A, B, C, D, E FROM R, S WHERE R.A = S.A
```

(a) [10 points]

The SQL query is processed with the Block Nested-Loop Join algorithm with $R$ as the outer relation. Each page is 2000 bytes long. We use a memory buffer of 10 pages. We also assume that all attribute values are 10 bytes long. What is the total processing cost for the Block Nested-Loop Join algorithm?

**Answer:**
(b) [15 points]

The SQL query is processed with the Merge-Join algorithm. Assume that $R$ and $S$ are not sorted on any attribute. Each page is 2000 bytes long. We use a memory buffer of 10 pages. We also assume that all attribute values are 10 bytes long. What is the total processing cost for the Merge-Join algorithm?

Answer: