

CS 3743
Introduction to Database Systems
Midterm 2 Solutions

<table>
<thead>
<tr>
<th>Question</th>
<th>Mark</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NAME:_________________________

Instructions
1. Do all of the 5 problems
2. You have 60 minutes for the exam
4. Show all your work
5. Do not separate exam papers

<table>
<thead>
<tr>
<th>Easy</th>
<th>Difficulty Level</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
1. **Indexing:** (20 pts, 4 pts each) Consider a disk with block size \( B = 1024 \) bytes. A block pointer is 8 bytes long and a record pointer is 10 bytes long. A file has 20000 customer records of fixed length. \( \text{CustomerId} \) is a key and is 6 bytes long. Each record is 100 bytes long. Assume that the file is not ordered on \( \text{CustomerId} \) and we want to construct a secondary index on \( \text{CustomerId} \). Answer the following questions.

(a) What is the index blocking factor?

\[
\begin{align*}
\text{Index Record Size} & = 6 + 10 = 16 \text{ bytes} \\
\text{Index Blocking Factor} & = \left\lfloor \frac{1024}{16} \right\rfloor = 64
\end{align*}
\]

(b) What is the number of index entries?

\[
\text{Number of Index Entries} = \text{Number of Records} = 20000
\]

(c) What is the total number of index blocks?

\[
\text{Number of Index Blocks} = \left\lceil \frac{20000}{64} \right\rceil = 313
\]

(d) What is the number of block accesses needed to search for and retrieve a record from file given its \( \text{CustomerId} \) using the index?

- **Binary Search on index requires** \( \lceil \log_{2} 313 \rceil = 9 \) block accesses
- **An extra block access is needed to read the disk block**
- **Total number of block accesses is 10.**
2. **Normalization:** (20 pts) Consider the universal relation \( R = \{A, B, C, D, E, F\} \) and the following functional dependencies

\[
\begin{align*}
EF & \rightarrow D \\
F & \rightarrow BC \\
D & \rightarrow A \\
\end{align*}
\]

EF is a key of this relation. Answer the following questions based on above information.

(a) Decompose \( R \) into 2NF relations.

\[
\begin{align*}
R1(A, D, E, F) \\
R2(B, C, F) \\
\end{align*}
\]

(b) Decompose \( R \) into 3NF relations.

\[
\begin{align*}
R1A(D, E, F) \\
R1B(A, D) \\
R2(B, C, F) \\
\end{align*}
\]
3. *Disk Storage*: (20 pts) Consider a disk with the following characteristics: block size B=1024 bytes; interblock gap size G=64 bytes; number of blocks per track = 32; number of tracks per surface = 200. A disk pack consists of 50 double-sided disks. Answer the following based on above information.

(a) What are the total and useful capacity (excluding interblock gaps) of a track?

Total Track Size = 32*(1024+64) = 34816 bytes
Useful Capacity = 32*1024 = 32768 bytes

(b) What are the total capacity and useful capacity of a cylinder?

Total Cylinder Capacity = 50*2*32*(1024+64)=3481600 bytes
Useful Cylinder Capacity = 50*2*32*1024 = 3276800 bytes

(c) What are the total capacity and useful capacity of a disk pack?

Total Capacity of a Disk Pack = 50*2*200*32*(1024+64) = 696320000 bytes
Useful Capacity of a Disk Pack = 50*2*200*32*1024 = 655360000 bytes

(d) If the drive rotates at a speed of 15000 revolutions per minute, what is the block transfer time?

Transfer Rate = 34816/(60*1000/15000) = 8704 bytes/msec
Block Transfer Time = 1024/8704 = 0.1176 msec
4. **ER-to-relational Mapping:** (20 pts) Map the following ER scheme into a set of relations.

![ER Diagram](image)

**Figure 1: ER Diagram**

**Solution:**

![ER Diagram Relations](image)

**Figure 2: ER Diagram Relations**
5. Indexing Structures: (20 pts) Given a point $p$, nearestneighbor query $nn(p)$ returns the item in the database that is closest to $p$ according to some metric such as absolute value. For example, consider the set $\{1,4,7\}$. The query $nn(6)$ will return 7 since $|7-6|=1$, $|4-6|=2$ and $|1-6|=5$ and 1 is the smallest distance. Your are given 1000 records. Each record is 100 bytes long. Each record includes a point plus some additional information. Block size is 1000 bytes. All the point coordinates are distinct and are integers in $\{1,2,3,...5000\}$. You have full control over how the file is structured, stored on the disk. Assume each point is 4 bytes and block pointer is 6 bytes. Suggest an index structure that speeds up nearest neighbor queries and show how the file is organized and stored to use your index. To show the speedup, compare the number of blocks retrieved without index and number of blocks retrieved with index.

Solution: Sort the file according to the point value. Given a point $p$ find where it would be if it were in the file. Lets say largest point less than $p$ is $p_1$ and smallest point larger than $p$ is $p_2$. Therefore, we have $p_1 < p < p_2$. In this case, either $p_1$ or $p_2$ is nearest neighbor of $p$. To do this you can use a primary index on point value and keep one index entry per file block. File has 100 file blocks. So, index has 100*(4+6)=1000 bytes and fits in a block. To retrieve the file you need to access the index and the file block resulting in retrieval of 2 file blocks. Without the index binary search over the data needs to be done resulting in $log_2100 \approx 7$ disk accesses.