Warmup #1  (Textbook Problem 8.21 (modified))

Normalize the following schema, with given constraints, to BCNF.

books(accessionno, isbn, title, author, publisher)
users(userid, name, deptid, deptname)

accessionno → isbn
isbn → title
isbn → publisher
isbn → author
userid → name
userid → deptid
deptid → deptname

Warmup #2  (Textbook Problem 10.11)

How does the remapping of bad sectors by disk controllers affect data-retrieval rates?

Warmup #3  (Textbook Problem 10.15)

Explain why the allocation of records to blocks affects database-system performance significantly.

Warmup #4  (Textbook Problem 11.3)

Construct a B+-tree for the following set of key values:

(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

Assume that the tree is initially empty and values are added in ascending order. Construct B+-trees for the cases where the number of pointers that will fit in one node is as follows:

a. Four
b. Six
c. Eight

Problem 5 (To Be Graded)

Following is the schema for the database of music festivals:

• musician(mid, name, address)
• musician_in_band(mid, bid, instrument)
• band(bid, bandname, home_state, home_zip)

(1) Assume the following requirement for band: a state will contain multiple zip codes, but a zip code will always be in the same state. Different bands may have the same name, come from the same home state and have the same home zip code. Please answer the following questions:

(1.a) Is the band schema in BCNF? If it is not, decompose it into a schema that is in BCNF. Please ensure your decomposition to be lossless.

(1.b) Consider the following decomposition of band:
• band_name(bid, bandname)
• band_address(bandname, home_state, home_zip)
Is the above decomposition lossless? Is the result in BCNF? Please explain.

(2) Suppose a new entity set is added
• shows(sid, bid, ticket_price, qty_sold, total_profit)
Where:
• total_profit is the total amount of money that the show generated
• qty_sold refers to the number of tickets sold at the given ticket_price (each show only has a single ticket_price initially).
• Every show contains exactly one band, but bands may play in multiple shows.

Assume the following requirement: a show is capable of selling tickets for different seats at a venue, each with a different ticket_price. The qty_sold attribute represents the number of tickets sold of one specific ticket_price. A show will still only feature a single band. After adding these constraints, the shows table contains the following dependencies:

\[
\begin{align*}
\text{sid} & \rightarrow \text{bid} \\
\text{sid, ticket_price} & \rightarrow \text{qty_sold} \\
\text{sid} & \rightarrow \text{total_profit}
\end{align*}
\]

Please answer the following questions:

(2.a) The schema for the shows table is not in BCNF. Please explain why not.

(2.b) Decompose shows into a new schema that is in BCNF. List all foreign keys on the final schema.

**Problem 6 (To Be Graded)**

Imagine that you have the following table:
The table contains 1024 rows in total. Answer the following questions about this table:

1. Suppose the database administrator decides to store `bandId` as an 8-byte integer, `bandName` as a 48-byte character array, and `ranking` as a 8-byte integer. If an instance of `bandName` is shorter than 48 bytes, the empty space will be filled with null characters.

   - All attributes of a tuple are stored in contiguous space within the same disk block\(^1\).
   - A disk block size is 512 bytes.
   - The disk on average performs the sequential read at 1\(\text{ms}\) per disk block and the random read at 10\(\text{ms}\) per disk block.

   (a) What is the maximum number of tuples that can be stored in a disk block?
   (b) What is the minimum number of disk blocks that need to be allocated for all tuples in the table?
   (c) What is the minimum time to read all tuples (in no particular order), assuming that the minimum number of disk blocks are allocated?

2. Suppose that a secondary index of B+-tree is created on the ranking column (which you can assume is a candidate key with unique values). Assume the following

   - Each tree node has a size of 64 bytes.
   - The data in a tree node are stored in contiguous space within the same disk block.
   - The tree has a fanout of 4 and each leaf node is 60% loaded\(^2\).
   - A pointer is 8-byte long.

   (a) How many bytes of space does the secondary index require?
   (b) How many disk reads (including index search and tuple retrieval) in the worst case are required to find a tuple via a specific ranking?

\(^1\)The minimum storage unit on a disk is called a disk block or a disk page.
\(^2\)The ratio of the number entries over the maximum number of entries is at least 60%.