Name:- Aniruddh Bajaj

Roll no.:- B20CS005

Assignment 6 & 7
Query optimization.

Q.1

Table Set (Normalized form)

~~author_id name_id~~
author_name_id
author_ki_id (Author_ID)
author_ka_naam (Author_Name)

book_name_id
book_ki_id (Book_ID)
book_ka_naam (Book)

author_book_id
author_ki_id (Author_ID)
book_ki_id (Book_ID)

purchase_book
book_ki_id (Boo_ID)
purchase_date (Purchase_Date)
quantity (Quantity).

Indexes

(a) key, Author_ID (author_name_id ~~author_ki_id~~)

(b) key, Book_ID (book_name_id, author_book_id, purchase_book)

Hash functions.

(a) (key[0] +32) %5

(b) (key[0] +32) %5

We have done grouping on the basis of first character (the english alphabet system → not chosed to see different at bucket breaking).

a.

The given table structure is normalized to 2nd normal form as no multivalued attributes exist and primary key is a single column and not a composite value.

BOOK → Book_ID

~~Author~~ AUTHOR → Author_ID

BOOK PURCHASE → Book_ID

It is better to normalize it upto BCNF, to reduce data redundancy and make searching and hash indexing easier. This will also remove transitive dependency.
(Normalized form is at the previous page)

b.

Author_ID and Book_ID happen to be unique among the tables and have the most probability of being used for searching queries. Also both of them seem to have a mixed datatype with some numerical part and also following a pattern, i.e., utilizable on our end.

Q.2. and Q.3

(a) Relational Algebra to retrieve names of books written by 'Carl Safina'.

$$T_1 = \sigma_{author\_ka\_naam = \text{"Carl Safina"}} \{author\_name\_id\}$$

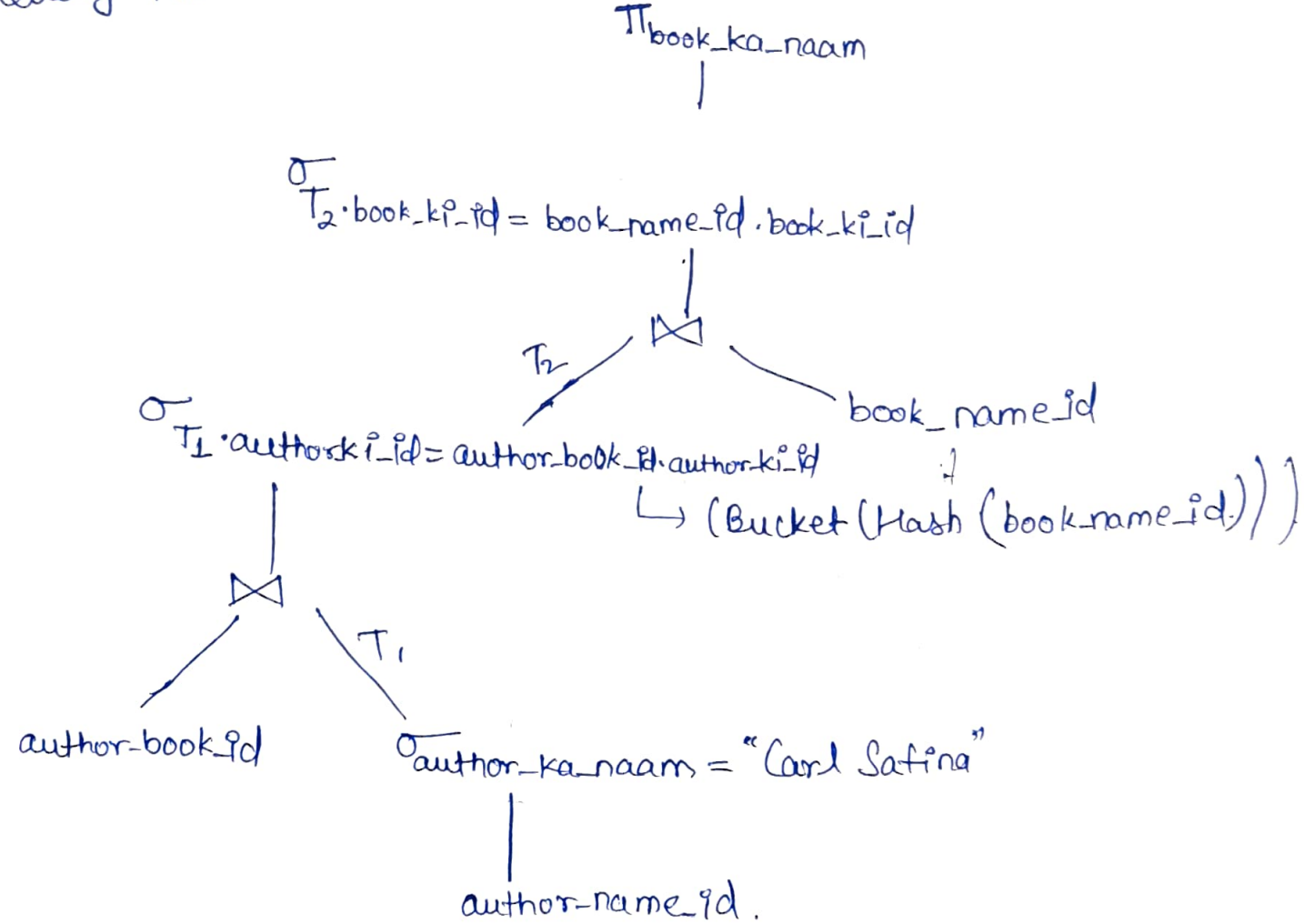
$$T_2 = \sigma_{T_1.author\_ki\_id = author\_book\_id.author\_ki\_id} \{T_1 \bowtie author\_book\_id\}$$

↓

Bucket(Hash(T_1))

Final relational algebra = $\Pi_{book\_ka\_naam} \{\sigma_{T_2.book\_ki\_id = book\_name\_id.book\_ki\_id} \{book\_name\_id \bowtie T_2\}\}$

Query Tree.



(b.)

Relational algebra to retrieve book names and author details of all books written by authors with names beginning with 'A' or 'P'

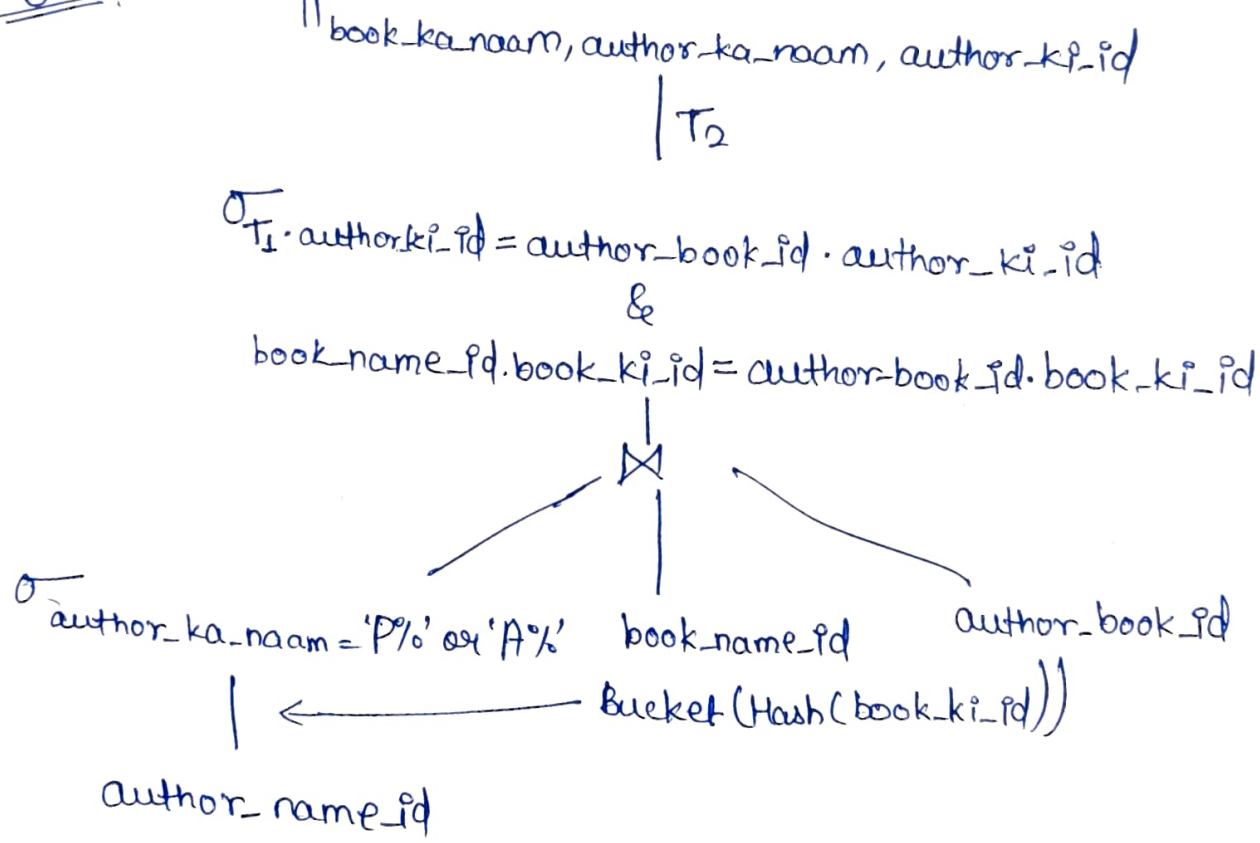
$$T_1 = \sigma_{\text{author\_ka\_naam} = \text{'P\%' or 'A\%'}} (\text{author\_name\_id})$$

$$T_2 = \sigma_{T_1.\text{author\_ki\_id} = \text{author\_book\_id}.\text{author\_ki\_id} \;\&\; \text{book\_name\_id}.\text{book\_ki\_id} = \text{author\_book\_id}.\text{book\_ki\_id}} (T_1 \bowtie \text{book\_name\_id} \bowtie \text{author\_book\_id})$$

Final Relational Algebra:-

$$= \Pi_{\text{book\_ka\_naam}, \text{author\_ka\_naam}, \text{author\_ki\_id}} (T_2)$$

Query Tree



(c).

Relational Algebra to retrieve all books with >= 5 copies.

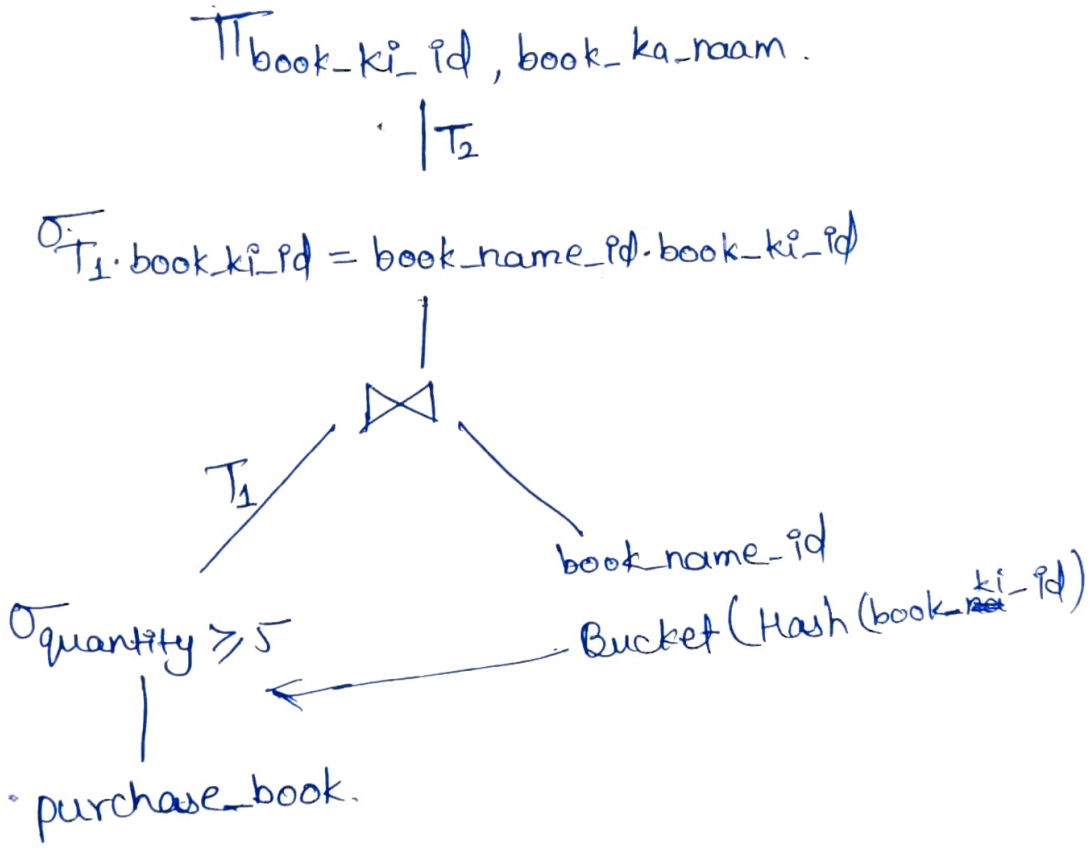
$$T_1 = \sigma_{quantity \geq 5}(purchase\_book)$$

$$T_2 = \sigma_{T_1.book\_ki\_id = book\_name\_id.book\_ki\_id}(T_1 \bowtie book\_name\_id)$$

Final Relational Algebra =

$$= \Pi_{book\_ki\_id, book\_ka\_naam}(T_2)$$

Query Tree



(d)

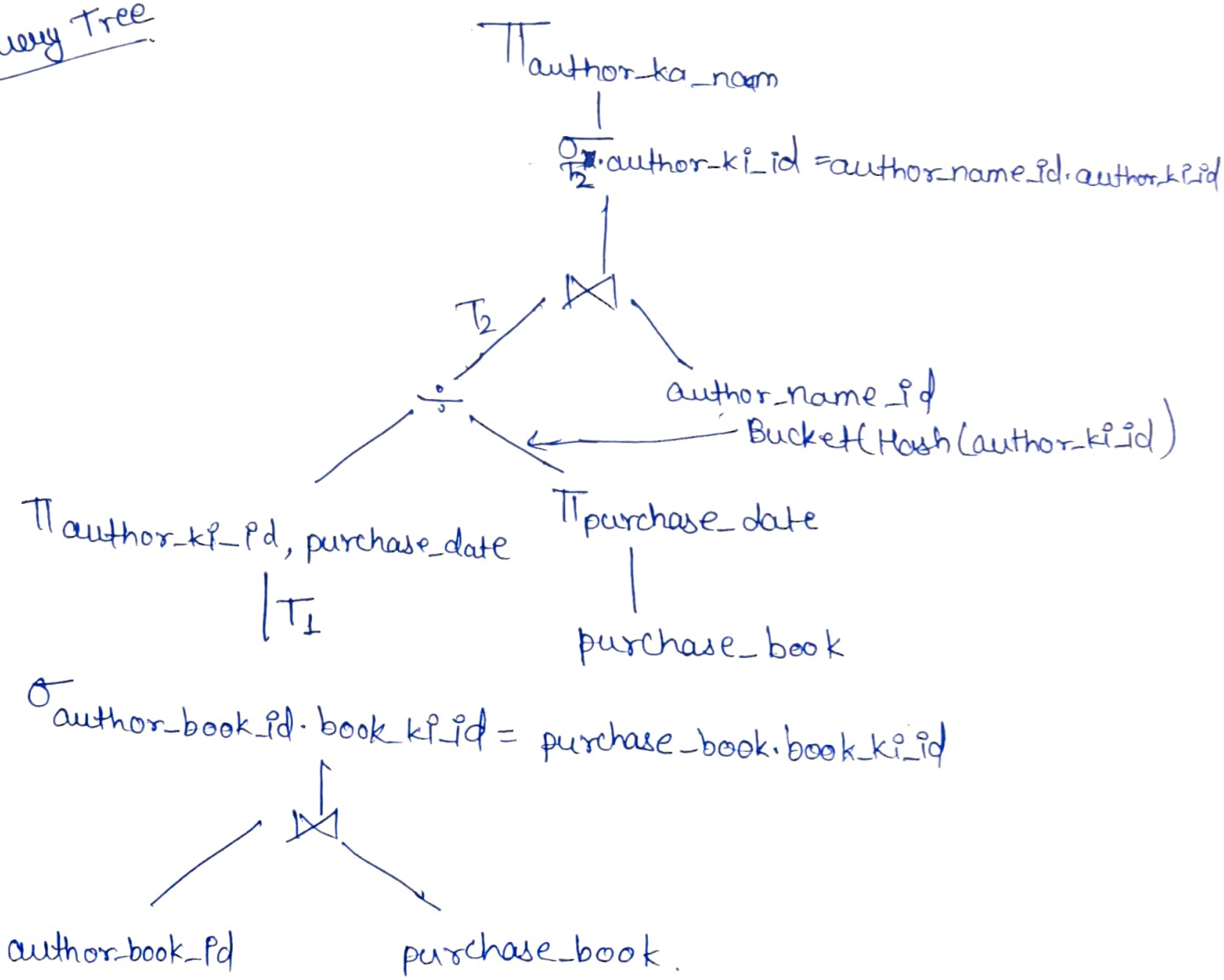
Relational algebra to retrieve author_names whose books have been purchased across all dates available on the purchase_table.

$$T_1 = \sigma_{\text{author\_book\_id.book\_ki\_id} = \text{purchase\_book.book\_ki\_id}} \{\text{author\_book\_id} \bowtie \text{purchase\_book}\}$$

$$T_2 = \Pi_{\text{author\_ki\_id, purchase\_date}}(T_1) \,/\, \Pi_{\text{purchase\_date}}(\text{purchase\_book})$$

$$\text{Final relational algebra} = \Pi_{\text{author\_ka\_naam}}(\sigma_{T_2.\text{author\_ki\_id} = \text{author\_name\_id.author\_ki\_id}}(T_2 \bowtie \text{author\_name\_id}))$$

Query Tree



(e).

Relational algebra for the books that have just a single copy in the database, to retrieve the author names.

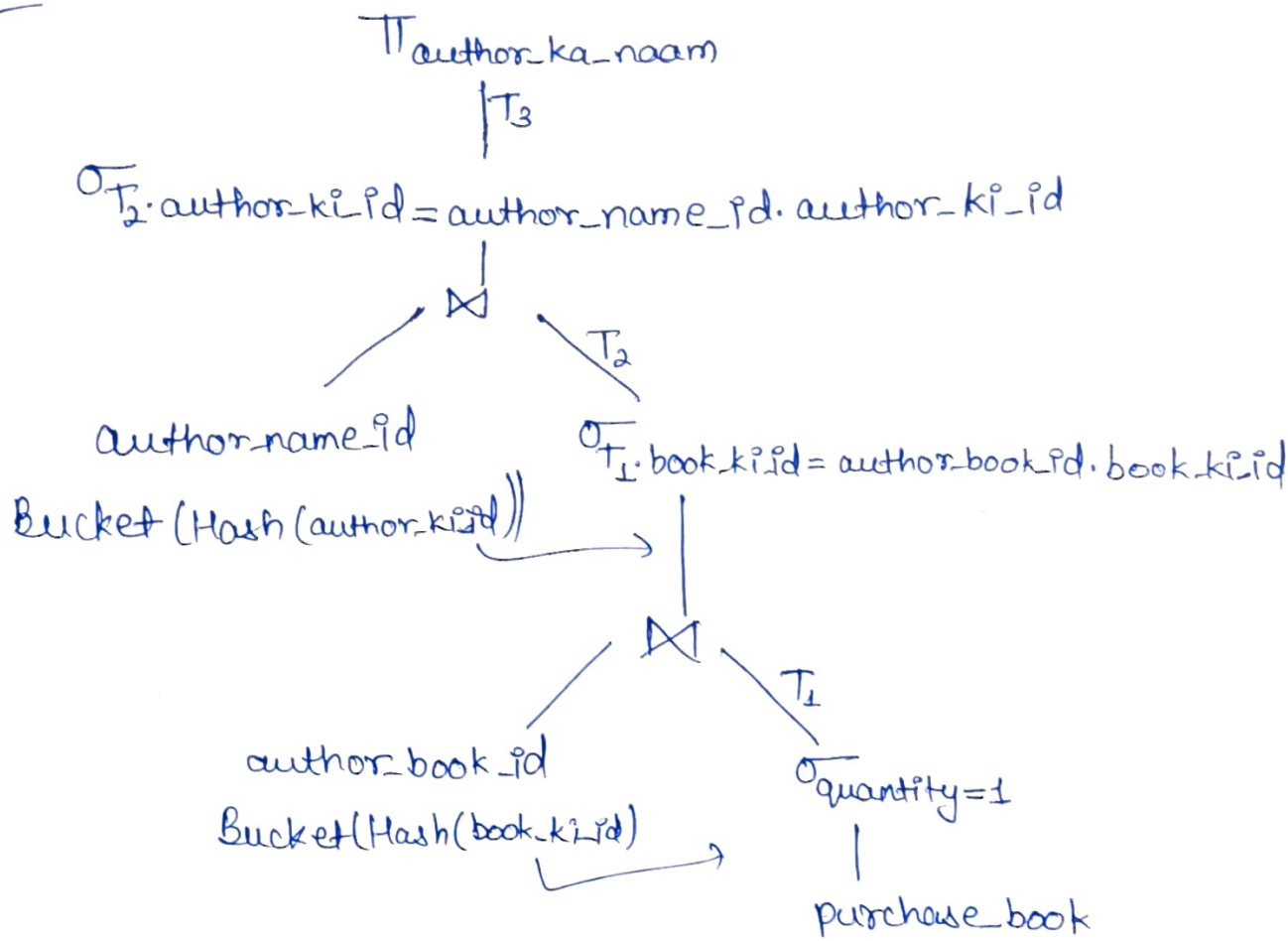
$$T_1 = \sigma_{quantity = 1} (purchase\_book)$$

$$T_2 = \sigma_{T_1.book\_ki\_id = author\_book\_id.book\_ki\_id} (T_1 \bowtie author\_book\_id)$$

$$T_3 = \sigma_{T_2.author\_ki\_id = author\_name\_id.author\_ki\_id} (author\_name\_id \bowtie T_2)$$

Final relational algebra = $\Pi_{author\_ka\_naam} (T_3)$

Query Tree



Output for Question 2 and 3:

```
Q2 Part A: Retrieve names of books written by 'Carl Safina'
1. Beyond Words: What Animals Think and Feel
2. Song for the Blue Ocean
Execution time: 989 microseconds.

Q2 Part B: Retrieve book names and author details of all books written by authors with names beginning with 'A' or 'P'
1. Author ID: An_Ch_0103, Author Name: Anjan Chatterjee, Book Name: The Aesthetic Brain
2. Author ID: An_Da_0104, Author Name: Antonio Damasio, Book Name: Self Comes to Mind
3. Author ID: Pe_Wo_1623, Author Name: Pelham G. Wodehouse, Book Name: Aunts ArenΓÇÖt Gentlemen
4. Author ID: Pe_Wo_1623, Author Name: Pelham G. Wodehouse, Book Name: Wodehouse at the Wicket
Execution time: 3008 microseconds.

Q2 Part C: Retrieve all books with >= 5 copies
1. Book ID: Deat_JR_1018, Book Name: Deathly Hallows_Harry Potter
2. Book ID: Fant_JR_1018, Book Name: Fantastic Beasts and Where to  Find Them
3. Book ID: Gobl_JR_1018, Book Name: Goblet of Fire_Harry Potter
4. Book ID: Phil_JR_1018, Book Name: PhilosopherΓÇÖs Stone_Harry Potter
5. Book ID: Pris_JR_1018, Book Name: Prisoner of Azkaban_Harry Potter
Execution time: 4986 microseconds.

Q2 Part D: Retrieve author_names whose books have been purchased across all dates available on the purchase table
1. Joanne K. Rowling
Execution time: 1006 microseconds.

Q2 Part E: For the books that have just a single copy in the database, retrieve the author_names
1. Anjan Chatterjee
2. Antonio Damasio
3. Marvin Minsky
4. Vilayanur Ramachandran
Execution time: 4004 microseconds.
```

```
Q3 Part A: Retrieve names of books written by 'Carl Safina'
1. Beyond Words: What Animals Think and Feel
2. Song for the Blue Ocean
Execution time: 3000 microseconds.

Q3 Part B: Retrieve book names and author details of all books written by authors with names beginning with 'A' or 'P'
1. Author ID: An_Ch_0103, Author Name: Anjan Chatterjee, Book Name: The Aesthetic Brain
2. Author ID: An_Da_0104, Author Name: Antonio Damasio, Book Name: Self Comes to Mind
3. Author ID: Pe_Wo_1623, Author Name: Pelham G. Wodehouse, Book Name: Aunts ArenΓÇÖt Gentlemen
4. Author ID: Pe_Wo_1623, Author Name: Pelham G. Wodehouse, Book Name: Wodehouse at the Wicket
Execution time: 9001 microseconds.

Q3 Part C: Retrieve all books with >= 5 copies
1. Book ID: Deat_JR_1018, Book Name: Deathly Hallows_Harry Potter
2. Book ID: Fant_JR_1018, Book Name: Fantastic Beasts and Where to  Find Them
3. Book ID: Gobl_JR_1018, Book Name: Goblet of Fire_Harry Potter
4. Book ID: Phil_JR_1018, Book Name: PhilosopherΓÇÖs Stone_Harry Potter
5. Book ID: Pris_JR_1018, Book Name: Prisoner of Azkaban_Harry Potter
Execution time: 6999 microseconds.

Q3 Part D: Retrieve author_names whose books have been purchased across all dates available on the purchase table
1. Joanne K. Rowling
Execution time: 1000 microseconds.

Q3 Part E: For the books that have just a single copy in the database, retrieve the author_names
1. Anjan Chatterjee
2. Antonio Damasio
3. Marvin Minsky
4. Vilayanur Ramachandran
Execution time: 6004 microseconds.
```

Question 4:

For part A, B, C and E, buckets of hash table of book_ki_id is used and buckets are doubled in extendible hashing of book_ki_id once. One index's overflow bucket is also used once in linear hashing of book_ki_id. Therefore, there are 6 buckets in the hash table by linear hashing process and 10 buckets in the hash table by extendible hashing process for book_ki_id. Therefore, the number of elements in each bucket in the hash table by extendible hashing process will be comparatively lesser than that in the buckets of hash table by linear hashing process. That's why execution time for extendible hashing will be comparatively lesser than that for linear hashing, which is clearly depicted in the output of code.

For part D, buckets of hash table of author_ki_id is used. None of the overflow buckets is used for linear hashing of author_ki_id and the buckets are never extended for the extendible hashing process of author_ki_id. Hence, there are 5 buckets in both the hash tables by linear hashing process and by extendible hashing process. Therefore, the number of elements in each bucket of both hash tables will not differ much which is clearly depicted in the output of code.