**INF3200 Assignment**

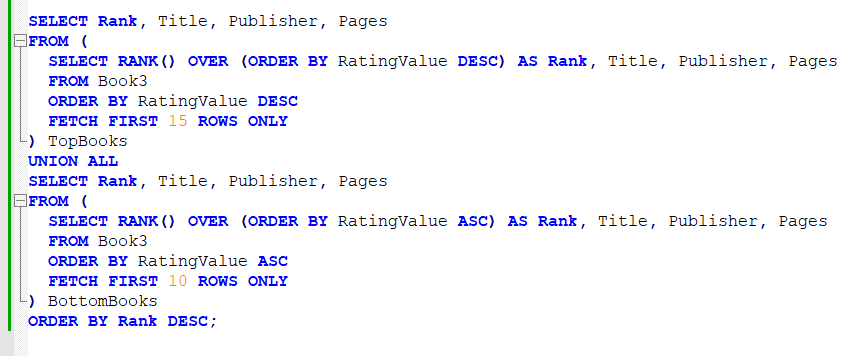
Name: Nguyen Van Quoc Chuong

Id: 47787810

**Part 1: Database schema and fragmentation**

Question 1:

a/



I take the top 15 highest ranked books and the 10 lowest ranked books from the book3 table, then I combine them together using the UNION operator.

b/ Book 3 is used because it contains the required values as described.

Question 2:

a/ Verticle fragmentation because each query is handled by a delicated local site. (no information needed from the other site)

Example schema:

Book3\_P1(ID, Title, Author1, Author2, Author3, Publisher, ISBN13)

Book3\_P2(ID, Date, Sales Rank, RatingsCount, RatingValue)

Book3\_P3(ID, Pages, ProductDimensions, PaperbackPrice, HardcoverPrice, EbookPrice, AudiobookPrice)

Explanation: Table 1 is used to store basic book information such as title and author, in fact these are the most frequently used attributes. Tables 2 and 3 contain related information about books, especially table 3 will serve the 2nd query well because the necessary attributes are all in the table, so we don't need to perform the join operation.

b/ It’s not valid because the books having number of pages from 601 to 800 disappeared. It violates the correctness and reconstructability.

Simple predicates:

P1: pages <= 200 -P1: pages > 200

P2: 200 < pages <= 600 -P2: pages <= 200; pages > 600

P3: pages > 800 -P3: pages <= 800

Using AND operator to get the result of the minterms.

M1: P1 ^ P2 ^ P3

200 600 800

--------]---------------------------------------------------------------

--------(------------------------]--------------------------------------

---------------------------------------------------(--------------------

0000000000000000000000000000000000000000000

M2: P1 ^ P2 ^ -P3

200 600 800

--------]---------------------------------------------------------------

--------(------------------------]--------------------------------------

---------------------------------------------------(--------------------

0000000000000000000000000000000000000000000

M3: P1 ^ -P2 ^ P3

200 600 800

--------]---------------------------------------------------------------

--------]------------------------(--------------------------------------

---------------------------------------------------(--------------------

0000000000000000000000000000000000000000000

M4: P1 ^ -P2 ^ -P3

200 600 800

--------]---------------------------------------------------------------

--------]------------------------(--------------------------------------

---------------------------------------------------]--------------------

11111100000000000000000000000000000000000000

M5: -P1 ^ P2 ^ P3

200 600 800

--------(---------------------------------------------------------------

--------(------------------------]--------------------------------------

---------------------------------------------------(--------------------

00000000000000000000000000000000000000000000

M6: -P1 ^ P2 ^ -P3

200 600 800

--------(---------------------------------------------------------------

--------(------------------------]--------------------------------------

---------------------------------------------------]--------------------

0000001111111111111111100000000000000000000000

M7: -P1 ^ -P2 ^ P3

--------(---------------------------------------------------------------

--------]------------------------(--------------------------------------

---------------------------------------------------(--------------------

000000000000000000000000000000011111111111111

M8: -P1 ^ -P2 ^ -P3

200 600 800

--------(---------------------------------------------------------------

--------]------------------------(--------------------------------------

---------------------------------------------------]--------------------

000000000000000000000111111111111000000000000

Elimination step and result: M4, M6, M7, M8

**Part 2: Data Warehouse Design**

Question 3:

Fact table:

Sales {Date\_id, Publisher\_id, Language\_id, Sales}

Dimension tables:

Date {Id, Day, Month, Year}

Publisher {Publisher\_key, Name}

Language {Language\_key, Name}

Explaination: The fact table would have the most records as it contains the sales data and can have multiple records for each day, publisher, and language combination. The Date Dimension table is expected to have the second-highest number of records as it includes all dates for which sales data is available. The Publisher and Language dimension tables are expected to have fewer records since they only include unique values for their respective attributes.

Question 4:

a/

Advantages:

1. Reduced response time for large classes of ad hoc queries
2. A substantial reduction of space usage compared to other indexing techniques.
3. Dramatic performance gains even on very low-end hardware
4. Very efficient parallel DML and loads

Which type of column is not suitable for bitmap index?

Columns that have a high number of unique values, known as high cardinality columns, are unsuitable for bitmap indexing because they require a larger number of bits in the index. This can result in slower query performance and increased storage needs.

b/

Publisher:

|  |  |  |  |
| --- | --- | --- | --- |
| AAAI Press | Springer International Publishing | Springer London | IEEE Computer Society Press |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

Language:

|  |  |
| --- | --- |
| English | Spanish |
| 1 | 0 |
| 1 | 0 |
| 1 | 0 |
| 1 | 0 |
| 0 | 1 |
| 0 | 1 |
| 0 | 1 |
| 0 | 1 |

c/ From the bitmap index tables above, we can find the total sales of “English” books published by “AAAI Press” following:

Step 1: Create bitmap indecies for Publisher and Language tables. The value of the corresponding column will be mark as 1 if the initial value is equal to the name of that column, otherwise, the value is marked as 0. In this case, “English” and “AAAI Press” are marked as 1 in the Language and Publisher tables respectively.

Step 2: Calculate the AND operation (bitwise operation) to two bitmap tables.

Step 3: The result is the index of corresponding rows which satisfy for both “English” and “AAAI Press”.

Step 4: Sumarize the sales value from the rows resulted in step 3 to get the total sales.

**Part 3: Data Integration**

Question 5:

a/

Global schema: Book (ID, Title, Authors, Publisher, Date, ISBN13, Pages)

Here, authors are stored as a list of authors, the “publication day” columns in all local schame are normalized to “Date”.

b and c/ Structural and semantic heterogeneity isses:

* “pubyear”, “pubmonth”, “pubday” (Book1) vs “publication\_year”, “publication\_month”, “publication\_day” (Book2) vs “Date” (Book3) vs “Publication\_Date” (Book4)
* “Author1”, “Author2”, “Author3” (Book3) vs a list of “Authors” (Book1 an Book2) vs a single “Author” (Book 4)
* “isbn13” vs “ISBN13”

This can be a challenge to integrate and execute queries because the names of the attributes are inconsistent across schemas.

Solution:

* Rename the attributes in all schemas to have consistent naming conventions. For example, standardize on "pub\_year," "pub\_month," and "pub\_day" or "publication\_date" for the publication date attributes across all schemas.
* Ensure that the data types of corresponding attributes are aligned. For example, if "Date" in Book3 and "Publication\_Date" in Book4 represent the same information, make sure they have the same data type (e.g., date or timestamp) in the global schema.
* Author Representation:
  + Normalize the representation of authors across schemas to ensure consistency.
  + In Book1 and Book2, where authors are listed as a single attribute, modify Book3 and Book4 to have a similar attribute named "Authors" that contains a list of authors.
  + If Book4 represents a single author, we may need to modify Book1, Book2, and Book3 to have a single "Author" attribute instead of a list.

**Part 4: Data Quality Issues**

Question 6:

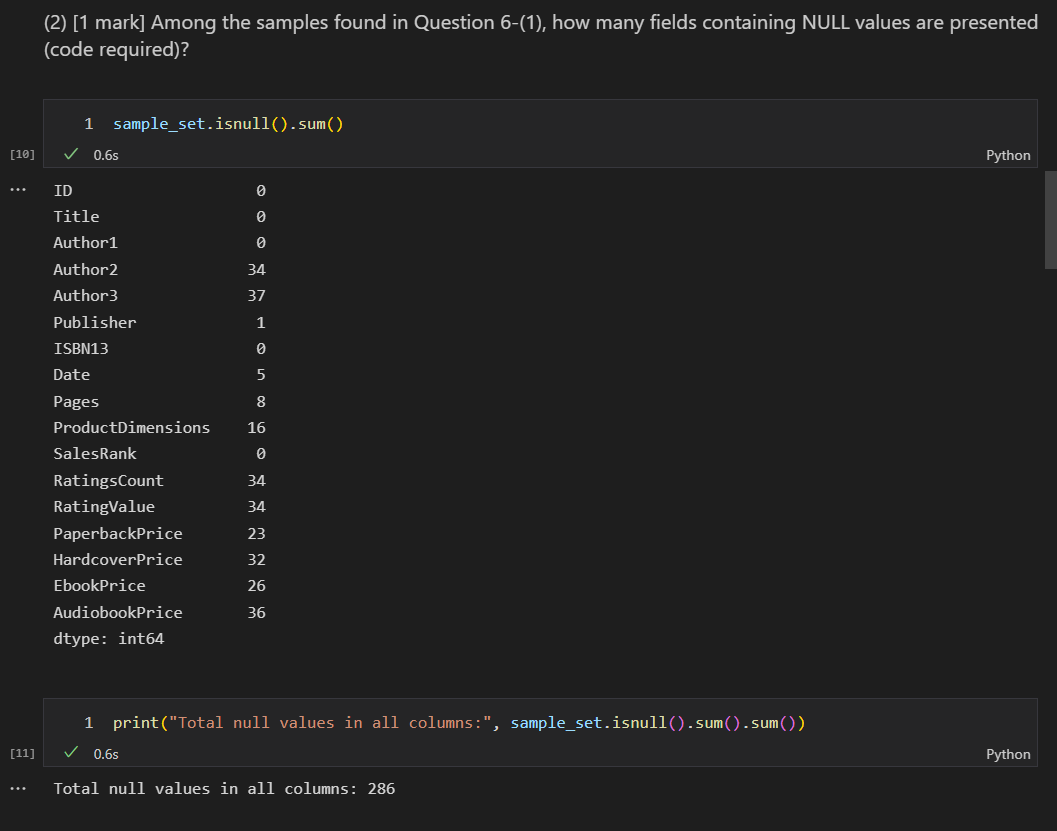
In this section, I use python pandas library to solve the problem.

a/ There’re 37 books whose id is the multiple of 100. The shape (37, 17) means that 37 rows (number of books) and 17 columns (features)

A screenshot of a computer

Description automatically generated with medium confidence

b/ There’re 286 null values in total.



c/

A screenshot of a computer

Description automatically generated with medium confidence

Question 7:

a/ The Jaccard distance is more likely to regard the two author strings as similar than the edit distance. This is because the Jaccard distance only considers the intersection of the two strings, in other words, the order of the names does not affect their similarity, while the edit distance considers all of the differences between the two strings.

b/

A screen shot of a computer program

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

Overall, the result is quite good. There’re total of 215 book pairs that exceeded the Jaccard similarity threshold. The precision is 0.1823, indicating that out of the retrieved book pairs, approximately 18.23% were truly similar. The recall is 0.9267, implying that out of the truly similar book pairs, approximately 92.67% were successfully retrieved.