Part2:

Question 3:

Star schema:

|  |
| --- |
|  |

Dimension = 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DATE\_DIMENSION   |  |  |  | | --- | --- | --- | | Date | Month | Year |   PUBLISHER\_DIMENSION   |  | | --- | | Publisher |   LANGUAGE   |  | | --- | | Language | |

Fact table:

|  |
| --- |
| SALE |
| DAY |
| PUBLISHER |
| LANGUAGE |

Among all dimension tables and fact table, the table that has the most record is SALE fact table. It is because Fact table can contain foreign keys with reference to all the primary keys of Dimension tables.

Question4:

1. The advantages of building bitmap index are:

* It can reduce query time by converting normal comparison between numbers and chars (size depend on data type) to bit operation.
* Reduce storage space by reducing the size of table when storing in memory.
* Increase I/O processing by keeping the fixed part (dictionary part, can be negligible when table size is large) and only change the dynamic part (number of records)

“Day” column is not suitable for using bitmap indexing because “Day” column have too many distinct values which significantly affect the size of bitmap index.

1. There are 8 records in the example -> 8 column. Moreover, “Publisher” column only contains four distinct values and “Language” only contains two -> 6 rows (4+2). Therefore, the bitmap can be built as:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | AAAI Press | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | Springer International Publishing | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | Springer London | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | | IEEE Computer Society Press | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | | English | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | Spanish | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |   Green cells are for “Publisher” and blue cells are for “Language” |

1. To find the total sale of “English” books published by “AAAI Press”, we need:

* Scan though “AAAI Press” vector and get records that AAAI Press = 1.
* Scan though “English” vector and get records that English = 1.
* Using AND operation to merge both results.
* Summarize their SALE.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | RECORD | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | | AAAI Press | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | Springer International Publishing | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | Springer London | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | | IEEE Computer Society Press | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | | English | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | Spanish | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | | Result | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   Green cells are for “Publisher”, blue cells are for “Language”, orange cells are cell that satisfied the condition while scanning the vector |

Therefore, the result will be R1 (the first record).

Part 3:

Question 5:

1. The global schema is:

Book(id, tile, authors, pub\_year, pub\_month, pub\_day, Publisher, isbn13, pages)

1. Structural heterogeneity issues:

* Type conflicts such as authors maybe contain key to relation of all of the book author but in Book3 or 4, author can be a single or multiple relations of type characters.

1. Semantic heterogeneity issues:

* Synonyms, homonyms, and hypernyms such as Publisher, publisher or publisher\_name.
* Different ontology such as “publication\_day” in Book2 only contain day but that of Book 4 may contain all day, month, and year together.

Part 4: Data quality Issues

