COMP - 6521
Advanced Database Technology and Applications

Project Report
On

Lab Assignment 2: Bitmap Indexing

Professor
Dr. Nematollaah Shiri

Team Members

Student Name	Student Id	Email Id
Yash Pandya	40119272	y_pandy@encs.concordia.ca
Himen Sidhpura	40091993	h_sidhpu@encs.concordia.ca
Sucheta Sudhakumari	40080543	s_ijaya@encs.concordia.ca

Index

1. Bitmap Indexing

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure. Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.

Indexes are primarily built on specific keys /fields of the data base even though some databases extend the power of indexing by letting developers create indexes on functions or expressions. A bitmap index is built on table attributes such that bitmap index for a field F is a collection of bit-vectors of length n, one for each possible value that may appear in the field F. The vector for value u has 1 in position i if the ith record has v in field F, and it ha5 0 there if not.

The number of bit vectors created for a field will be equal to the number of distinct values for that field and the length of the bit vector will be equal to the number of rows/tuples in a table.

2. Program Description

Phase 1 –Program Initialization: Program execution is triggered by the ProgramController class. Constant values including block size and file I/O paths are initialized. The input files are ready block by block into the program, starting with T1 followed by T2.

Phase 2 – Creation of bitmap index and sorting index: A 2D array of data type: long and size : (number of tuples)*(number of tuples +1) is created to store the bit vectors .The first column holds the distinct values of the field while the remaining columns signify the bits corresponding to each tuple. The row size of the array is set to be equal to the number of tuples, assuming the maximum row numbers possible in case no field values are repeated across the block. Every time a unique value is read, the 1st column in the most recent row is set to that value after which the corresponding bit vector is set to 1. The key values are then sorted using quicksort and the bitmap index for each block is then written into a sublist.

Phase 3 – Merging index blocks: The block wise bitmap index sublists, corresponding to each file are merged recursively into a single file using TPMMS technique. At the end of this point, we get 6 separate index files corresponding to the 3 fields and 2 files. For each distinct value of the index, the columns with bit vector 1 indicate duplicates tuples.

Phase 4 – Compression of Index: Each of the index files are then compressed using run length encoding algorithm.

Phase 5 – Merging the files: The bitmap indices of employee file are read into the memory using TPMMS algorithm. For each unique employee number, corresponding tuples are located from data file and checked for duplicates. In case of duplicates, the data is sorted using date and the latest tuple is then written into the output file.

3. Steps to run the program

1. Import the project Bitmap-Indexing to the IDE.
2. Set constant values including the block size, memory allocation size, I/O file paths in the constants.java class.
3. Run the program from ProgramController.java class.
4. The execution will be performed phase by phase and results will be printed on the console.

4. Algorithms

PHASE - 1: Program initialization:

- Start.
- Allocate main memory.
- Allocate block size, file I/O paths and other utility constants.

PHASE - 2: Bitmap Index Creation:

- Start
- Read the file block by block.
- For each block, create a 2D array of size equal to the number of tuples in a block, to store the bitmap indices for the corresponding field.
- For each unique field value read, add a row to the array.
- For each field value read (new or existing in the array), set the corresponding bit vector to 1 in the array.
- Add the key value into a list of key values already read.
- Sort the key list.
- Write the bit vectors for each block into a file, fetching tuples in sorted order (This creates a sorted index).
- Continue till all blocks are read.
- Repeat the steps for indexes on other fields on the same file as well as on file T2.
- Stop

PHASE – 3: Merging Bitmap Index blocks:

- Start.
- Read bitmap index sublists of the same type, file by file.
- Taking 2 indices at a time, merge them into a single file .
- Repeat the process until a single index file remains for each index type.
- Identify duplicates using the index and eliminate them from the data file.
- Repeat the steps for indexes on other fields as well as for file T2.
- Stop

PHASE – 4: Compression of Bitmap indices:

- Start.
- Read bitmap index blocks one by one into the program.
- Using Run Length Encoding technique perform index compression.
- Repeat the steps for indexes on other fields as well as for file T2.
- Stop

PHASE – 5: Merging the data files:

- Start.
- Using TPMMS technique, read the index for employee field into the program.
- For each employee IDs read, identify the indices with bit vector 1 which correspond to duplicate tuples.
- Find the corresponding block of data from the data blocks.
- Scan the data block and locate all tuples with the same employee ID, adding them to a list.
- Sort the list according to date and write the latest one to the output file.
- Repeat the steps until the end of both index files.
- Stop

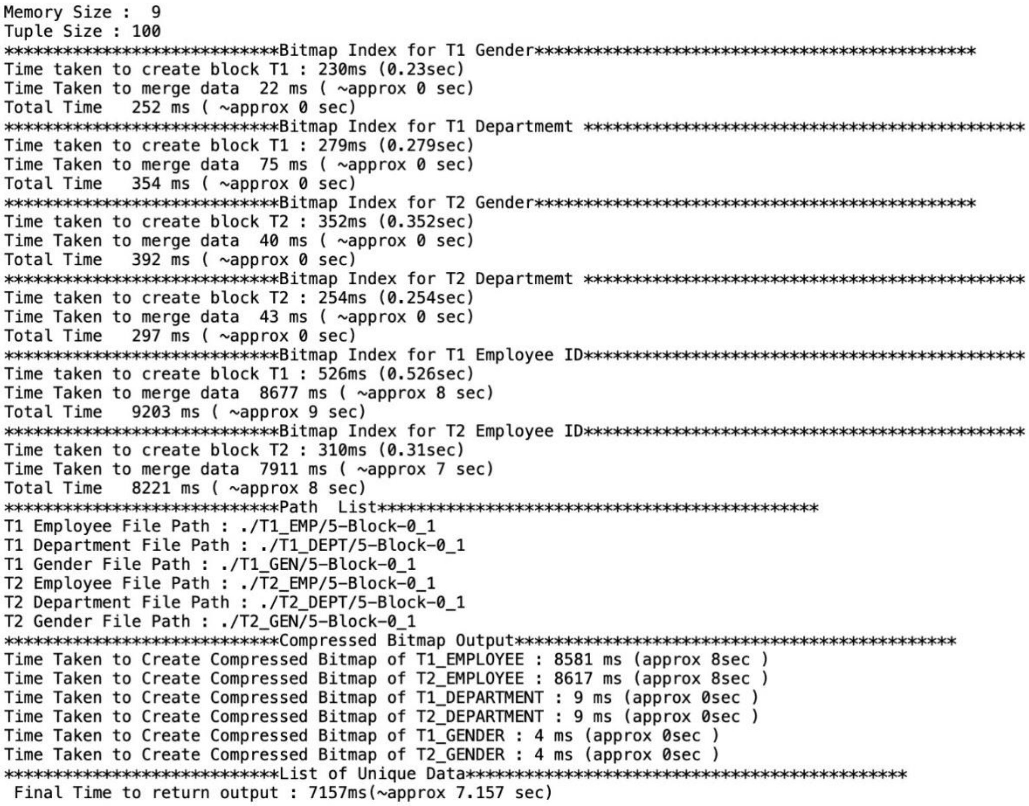
5. Experiment Results:

The experiment was conducted by changing the input file size, the result of which has been consolidated below:

Memory Size = 10mb Block Size = 40						
Tuple Count	**20,000**	**50,000**	**1,00,000**	**20,000**	**50,000**	**1,00,000**
T1 File				**T2 File**		
Gender Bitmap Index						
Time taken to generate sublists	230 ms	317 ms	806 ms	352 ms	421 ms	498 ms
Time taken for Merge Data	22 ms	36 ms	96 ms	40 ms	36 ms	70 ms
Read Count	248	608	1210	248	608	1210
Write Count	86	206	408	86	206	408
Total Time for Bitmap index	252 ms	407 ms	902 ms	392 ms	457 ms	568 ms
Time taken for compression	4 ms	8 ms	16 ms	4 ms	10 ms	16 ms
Department Bitmap Index						
Time taken to generate sublists	279 ms	654 ms	1062 ms	254 ms	349 ms	568 ms
Time taken for Merge Data	75 ms	87 ms	172 ms	43 ms	61 ms	183 ms
Read Count	920	2240	4450	920	2240	4450
Write Count	430	1030	2040	430	1030	2040
Total Time for Bitmap index	354 ms	741 ms	1234 ms	297 ms	410 ms	751 ms
Time taken for compression	9 ms	19 ms	46 ms	9 ms	19 ms	50 ms
Employee Bitmap Index						
Time taken to generate sublists	526 ms	915 ms	1617 ms	310 ms	986 ms	1670 ms
Time taken for Merge Data	8677 ms	11	206229 ms	7911 ms	46749 ms	170764 ms
Read Count	119469	346709	785848	119755	348511	793391
Write Count	99257	296324	685118	99597	298131	692592
Total Time for Bitmap index	9203 ms	48094 ms	207846 ms	8221 ms	47735 ms	172434 ms
Time taken for compression	8581 ms	52762 ms	215179 ms	8617 ms	53650 ms	217501 ms

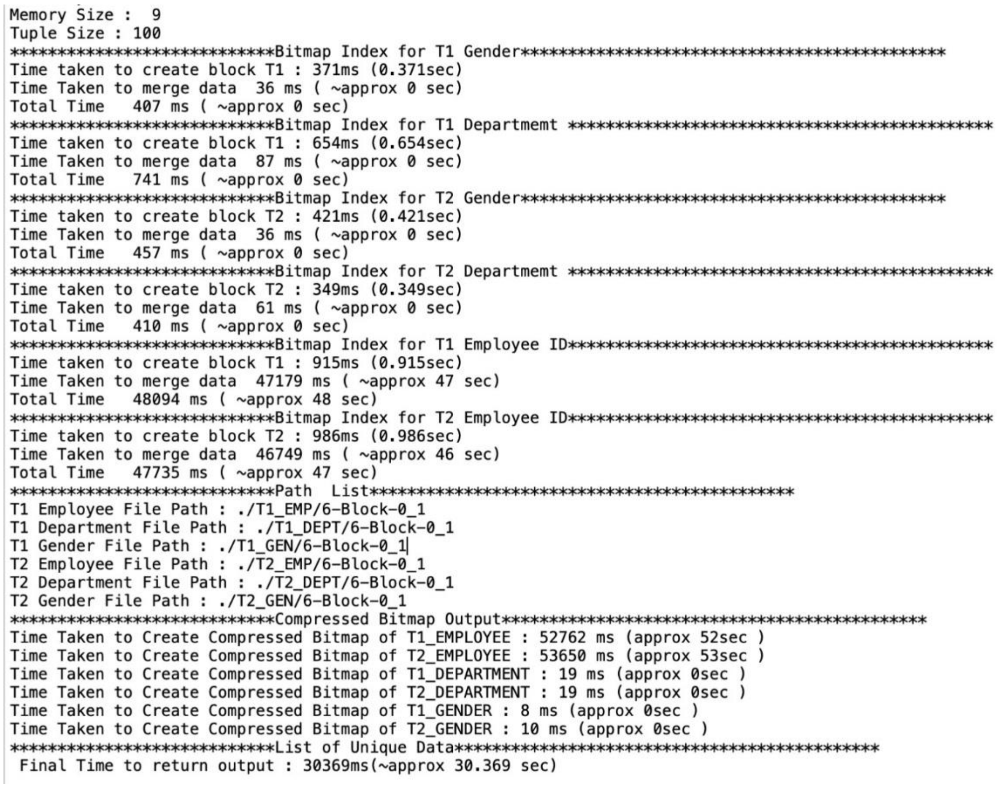
Time taken to merge T1 and T2:

- For 20,000: 7157 ms (~approx. 7.157 sec)
- For 50,000: 30369 ms (~approx. 30.369 sec)
- For 1,00,000: 138799 ms (~approx. 138.799 sec)



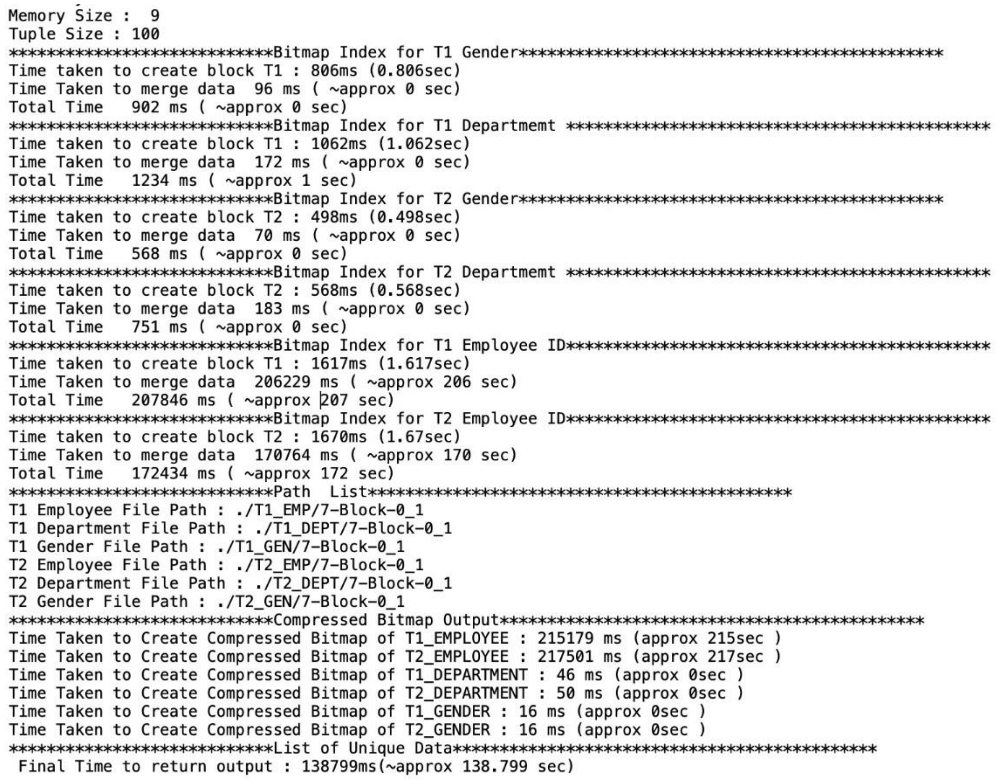
```
Memory Size :  9
Tuple Size : 100
****************************Bitmap Index for T1 Gender*********************************************
Time taken to create block T1 : 230ms (0.23sec)
Time Taken to merge data  22 ms ( ~approx 0 sec)
Total Time   252 ms ( ~approx 0 sec)
****************************Bitmap Index for T1 Departmemt *********************************************
Time taken to create block T1 : 279ms (0.279sec)
Time Taken to merge data  75 ms ( ~approx 0 sec)
Total Time   354 ms ( ~approx 0 sec)
****************************Bitmap Index for T2 Gender*********************************************
Time taken to create block T2 : 352ms (0.352sec)
Time Taken to merge data  40 ms ( ~approx 0 sec)
Total Time   392 ms ( ~approx 0 sec)
****************************Bitmap Index for T2 Departmemt *********************************************
Time taken to create block T2 : 254ms (0.254sec)
Time Taken to merge data  43 ms ( ~approx 0 sec)
Total Time   297 ms ( ~approx 0 sec)
****************************Bitmap Index for T1 Employee ID*********************************************
Time taken to create block T1 : 526ms (0.526sec)
Time Taken to merge data  8677 ms ( ~approx 8 sec)
Total Time   9203 ms ( ~approx 9 sec)
****************************Bitmap Index for T2 Employee ID*********************************************
Time taken to create block T2 : 310ms (0.31sec)
Time Taken to merge data  7911 ms ( ~approx 7 sec)
Total Time   8221 ms ( ~approx 8 sec)
****************************Path  List*********************************************
T1 Employee File Path : ./T1_EMP/5-Block-0_1
T1 Department File Path : ./T1_DEPT/5-Block-0_1
T1 Gender File Path : ./T1_GEN/5-Block-0_1
T2 Employee File Path : ./T2_EMP/5-Block-0_1
T2 Department File Path : ./T2_DEPT/5-Block-0_1
T2 Gender File Path : ./T2_GEN/5-Block-0_1
****************************Compressed Bitmap Output*********************************************
Time Taken to Create Compressed Bitmap of T1_EMPLOYEE : 8581 ms (approx 8sec )
Time Taken to Create Compressed Bitmap of T2_EMPLOYEE : 8617 ms (approx 8sec )
Time Taken to Create Compressed Bitmap of T1_DEPARTMENT : 9 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T2_DEPARTMENT : 9 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T1_GENDER : 4 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T2_GENDER : 4 ms (approx 0sec )
****************************List of Unique Data*********************************************
 Final Time to return output : 7157ms(~approx 7.157 sec)
```

Figure 1: 10MB – 20,000 records Bitmap Index



```
Memory Size :  9
Tuple Size : 100
****************************Bitmap Index for T1 Gender*********************************************
Time taken to create block T1 : 371ms (0.371sec)
Time Taken to merge data  36 ms ( ~approx 0 sec)
Total Time   407 ms ( ~approx 0 sec)
****************************Bitmap Index for T1 Departmemt *********************************************
Time taken to create block T1 : 654ms (0.654sec)
Time Taken to merge data  87 ms ( ~approx 0 sec)
Total Time   741 ms ( ~approx 0 sec)
****************************Bitmap Index for T2 Gender*********************************************
Time taken to create block T2 : 421ms (0.421sec)
Time Taken to merge data  36 ms ( ~approx 0 sec)
Total Time   457 ms ( ~approx 0 sec)
****************************Bitmap Index for T2 Departmemt *********************************************
Time taken to create block T2 : 349ms (0.349sec)
Time Taken to merge data  61 ms ( ~approx 0 sec)
Total Time   410 ms ( ~approx 0 sec)
****************************Bitmap Index for T1 Employee ID*********************************************
Time taken to create block T1 : 915ms (0.915sec)
Time Taken to merge data  47179 ms ( ~approx 47 sec)
Total Time   48094 ms ( ~approx 48 sec)
****************************Bitmap Index for T2 Employee ID*********************************************
Time taken to create block T2 : 986ms (0.986sec)
Time Taken to merge data  46749 ms ( ~approx 46 sec)
Total Time   47735 ms ( ~approx 47 sec)
****************************Path  List*********************************************
T1 Employee File Path : ./T1_EMP/6-Block-0_1
T1 Department File Path : ./T1_DEPT/6-Block-0_1
T1 Gender File Path : ./T1_GEN/6-Block-0_1
T2 Employee File Path : ./T2_EMP/6-Block-0_1
T2 Department File Path : ./T2_DEPT/6-Block-0_1
T2 Gender File Path : ./T2_GEN/6-Block-0_1
****************************Compressed Bitmap Output*********************************************
Time Taken to Create Compressed Bitmap of T1_EMPLOYEE : 52762 ms (approx 52sec )
Time Taken to Create Compressed Bitmap of T2_EMPLOYEE : 53650 ms (approx 53sec )
Time Taken to Create Compressed Bitmap of T1_DEPARTMENT : 19 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T2_DEPARTMENT : 19 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T1_GENDER : 8 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T2_GENDER : 10 ms (approx 0sec )
****************************List of Unique Data*********************************************
 Final Time to return output : 30369ms(~approx 30.369 sec)
```

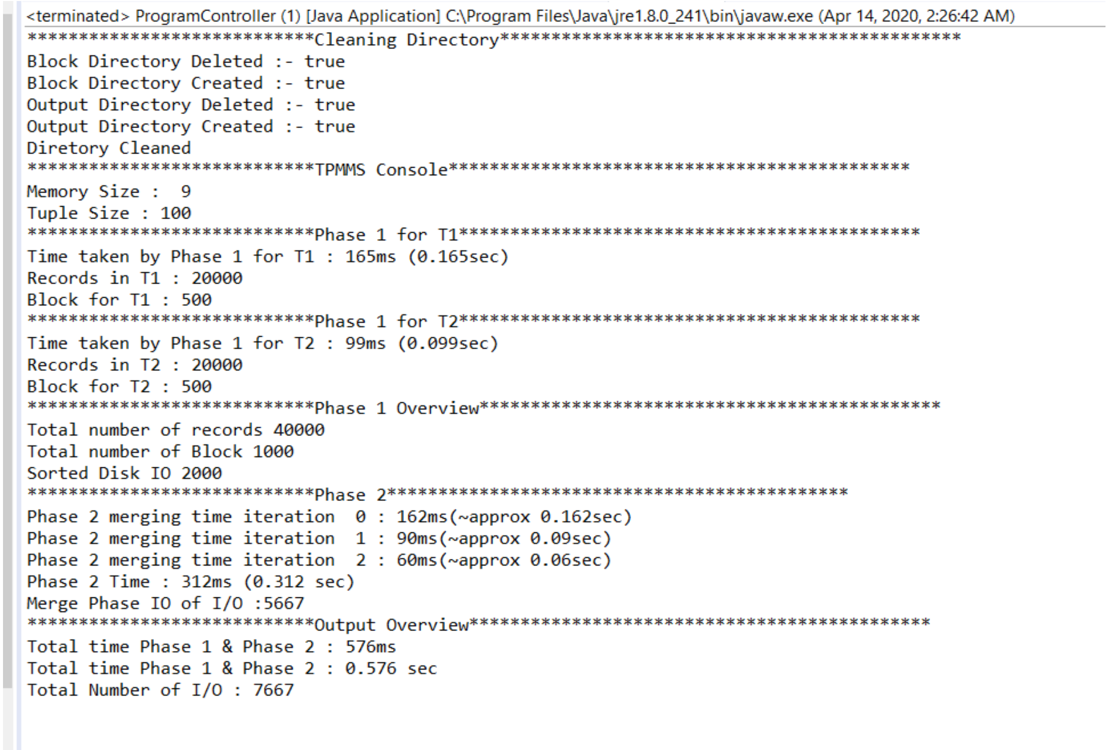
Figure 2: 10MB – 50,000 records Bitmap Index



```
Memory Size :  9
Tuple Size : 100
****************************Bitmap Index for T1 Gender*********************************************
Time taken to create block T1 : 806ms (0.806sec)
Time Taken to merge data  96 ms ( ~approx 0 sec)
Total Time   902 ms ( ~approx 0 sec)
****************************Bitmap Index for T1 Departmemt *********************************************
Time taken to create block T1 : 1062ms (1.062sec)
Time Taken to merge data  172 ms ( ~approx 0 sec)
Total Time   1234 ms ( ~approx 1 sec)
****************************Bitmap Index for T2 Gender*********************************************
Time taken to create block T2 : 498ms (0.498sec)
Time Taken to merge data  70 ms ( ~approx 0 sec)
Total Time   568 ms ( ~approx 0 sec)
****************************Bitmap Index for T2 Departmemt *********************************************
Time taken to create block T2 : 568ms (0.568sec)
Time Taken to merge data  183 ms ( ~approx 0 sec)
Total Time   751 ms ( ~approx 0 sec)
****************************Bitmap Index for T1 Employee ID*********************************************
Time taken to create block T1 : 1617ms (1.617sec)
Time Taken to merge data  206229 ms ( ~approx 206 sec)
Total Time   207846 ms ( ~approx 207 sec)
****************************Bitmap Index for T2 Employee ID*********************************************
Time taken to create block T2 : 1670ms (1.67sec)
Time Taken to merge data  170764 ms ( ~approx 170 sec)
Total Time   172434 ms ( ~approx 172 sec)
****************************Path  List*********************************************
T1 Employee File Path : ./T1_EMP/7-Block-0_1
T1 Department File Path : ./T1_DEPT/7-Block-0_1
T1 Gender File Path : ./T1_GEN/7-Block-0_1
T2 Employee File Path : ./T2_EMP/7-Block-0_1
T2 Department File Path : ./T2_DEPT/7-Block-0_1
T2 Gender File Path : ./T2_GEN/7-Block-0_1
****************************Compressed Bitmap Output*********************************************
Time Taken to Create Compressed Bitmap of T1_EMPLOYEE : 215179 ms (approx 215sec )
Time Taken to Create Compressed Bitmap of T2_EMPLOYEE : 217501 ms (approx 217sec )
Time Taken to Create Compressed Bitmap of T1_DEPARTMENT : 46 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T2_DEPARTMENT : 50 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T1_GENDER : 16 ms (approx 0sec )
Time Taken to Create Compressed Bitmap of T2_GENDER : 16 ms (approx 0sec )
****************************List of Unique Data*********************************************
 Final Time to return output : 138799ms(~approx 138.799 sec)
```

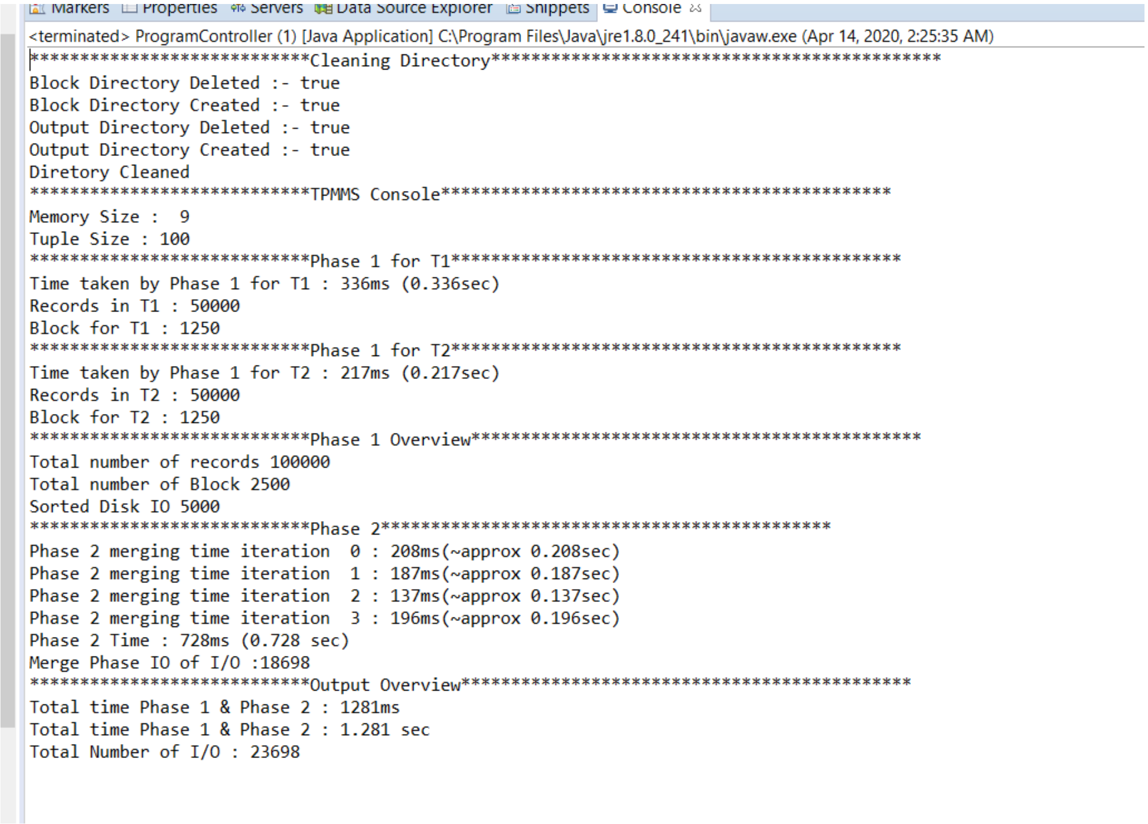
Figure 3: 10MB – 1,00,000 records Bitmap Index

6. Results TPMMS



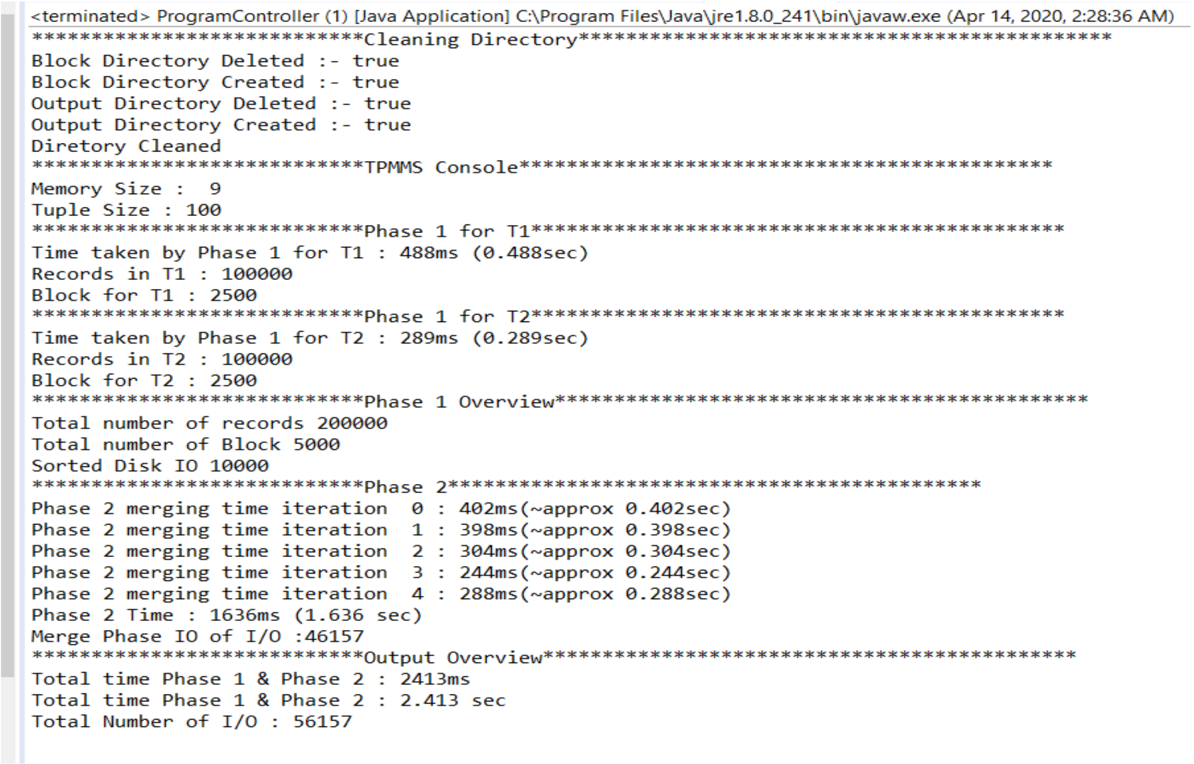
```
<terminated> ProgramController (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Apr 14, 2020, 2:26:42 AM)
****************************Cleaning Directory*********************************************
Block Directory Deleted :- true
Block Directory Created :- true
Output Directory Deleted :- true
Output Directory Created :- true
Diretory Cleaned
****************************TPMMS Console*********************************************
Memory Size :  9
Tuple Size : 100
****************************Phase 1 for T1*********************************************
Time taken by Phase 1 for T1 : 165ms (0.165sec)
Records in T1 : 20000
Block for T1 : 500
****************************Phase 1 for T2*********************************************
Time taken by Phase 1 for T2 : 99ms (0.099sec)
Records in T2 : 20000
Block for T2 : 500
****************************Phase 1 Overview*********************************************
Total number of records 40000
Total number of Block 1000
Sorted Disk IO 2000
****************************Phase 2*********************************************
Phase 2 merging time iteration  0 : 162ms(~approx 0.162sec)
Phase 2 merging time iteration  1 : 90ms(~approx 0.09sec)
Phase 2 merging time iteration  2 : 60ms(~approx 0.06sec)
Phase 2 Time : 312ms (0.312 sec)
Merge Phase IO of I/O :5667
****************************Output Overview*********************************************
Total time Phase 1 & Phase 2 : 576ms
Total time Phase 1 & Phase 2 : 0.576 sec
Total Number of I/O : 7667
```

Figure 4: 10MB – 20,000 records TPMMS



```
<terminated> ProgramController (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Apr 14, 2020, 2:25:35 AM)
****************************Cleaning Directory*********************************************
Block Directory Deleted :- true
Block Directory Created :- true
Output Directory Deleted :- true
Output Directory Created :- true
Diretory Cleaned
****************************TPMMS Console*********************************************
Memory Size :  9
Tuple Size : 100
****************************Phase 1 for T1*********************************************
Time taken by Phase 1 for T1 : 336ms (0.336sec)
Records in T1 : 50000
Block for T1 : 1250
****************************Phase 1 for T2*********************************************
Time taken by Phase 1 for T2 : 217ms (0.217sec)
Records in T2 : 50000
Block for T2 : 1250
****************************Phase 1 Overview*********************************************
Total number of records 100000
Total number of Block 2500
Sorted Disk IO 5000
****************************Phase 2*********************************************
Phase 2 merging time iteration  0 : 208ms(~approx 0.208sec)
Phase 2 merging time iteration  1 : 187ms(~approx 0.187sec)
Phase 2 merging time iteration  2 : 137ms(~approx 0.137sec)
Phase 2 merging time iteration  3 : 196ms(~approx 0.196sec)
Phase 2 Time : 728ms (0.728 sec)
Merge Phase IO of I/O :18698
****************************Output Overview*********************************************
Total time Phase 1 & Phase 2 : 1281ms
Total time Phase 1 & Phase 2 : 1.281 sec
Total Number of I/O : 23698
```

Figure 5: 10MB – 50,000 Records TPMMS



```
<terminated> ProgramController (1) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Apr 14, 2020, 2:28:36 AM)
****************************Cleaning Directory*********************************************
Block Directory Deleted :- true
Block Directory Created :- true
Output Directory Deleted :- true
Output Directory Created :- true
Diretory Cleaned
****************************TPMMS Console*********************************************
Memory Size :  9
Tuple Size : 100
****************************Phase 1 for T1*********************************************
Time taken by Phase 1 for T1 : 488ms (0.488sec)
Records in T1 : 100000
Block for T1 : 2500
****************************Phase 1 for T2*********************************************
Time taken by Phase 1 for T2 : 289ms (0.289sec)
Records in T2 : 100000
Block for T2 : 2500
****************************Phase 1 Overview*********************************************
Total number of records 200000
Total number of Block 5000
Sorted Disk IO 10000
****************************Phase 2*********************************************
Phase 2 merging time iteration  0 : 402ms(~approx 0.402sec)
Phase 2 merging time iteration  1 : 398ms(~approx 0.398sec)
Phase 2 merging time iteration  2 : 304ms(~approx 0.304sec)
Phase 2 merging time iteration  3 : 244ms(~approx 0.244sec)
Phase 2 merging time iteration  4 : 288ms(~approx 0.288sec)
Phase 2 Time : 1636ms (1.636 sec)
Merge Phase IO of I/O :46157
****************************Output Overview*********************************************
Total time Phase 1 & Phase 2 : 2413ms
Total time Phase 1 & Phase 2 : 2.413 sec
Total Number of I/O : 56157
```

Figure 6: 10MB – 1,00,000 Records TPMMS

7. Comparison to TPMMS

Compare to TPMMS, Bitmap index takes more time to generate output file containing recently updated records and no duplicates. One of the major problems occurs while using Bitmap index approach, size of the file is continuously increasing with increase in number of Tuple. For example, when file with 5,00,000 Tuples uses more than 137 GB space for only Bitmap index and output file as well as it might takes days to generate Bitmap Index only. While in TPMMS, it just takes few mb to generate output in few minutes. While Comparing Disk I\O, both Bitmap Index and TPMMS have approximately same number of disk I/O. Since, both approaches have advantages and disadvantages. It was also observed that if file contain

8. Coding Standards

The most general coding conventions were followed while the codes were developed as follows,

- The class name begins with an uppercase word.
- E.g.: ProgramController.java
- Constants are called with characters in the upper case
- The variable name is descriptive and is rendered in lower case including a capital letter to separate words.
- The procedure name begins with a lowercase character and uses the uppercase characters to separate words.

9. Class Description:

Constants.java: This class stores the constant values required for program execution like file IO paths, block size etc.

ProgramController.java: This class has the methods for reading tuples into main memory, sorting them and creating the sublists. It also handles reading the indices and producing final merged output.

BuildIndex.java: This class creates bit vectors and bitmap index subsequently from the input files.

CompressedBitmap.java: This creates the compressed bitmaps.

QuickSort.java: This class performs the quicksort algorithm on the bitmap index for blocks read into memory.

MergeData.java: This class merges the bitmap for blocks into a single bitmap index.

10. Group Member Contribution:

Member participation was uniform across all stages of the project development. We had meetings once a week to discuss on design changes and individual progress. This ensured that everyone is on the same page. We also adopted pair programming strategy which let us help each other with our areas of expertise, thereby developing efficient code. The documentation part was split into sections and assigned to each teammate as a part of even work distribution.

References

1. https://www.geeksforgeeks.org/java-program-for-quicksort/
2. https://oracle-base.com/articles/9i/bitmap-join-indexes
3. https://logicalread.com/2013/06/03/oracle-11g-bitmap-join-indexes-mc02/
4. https://www.youtube.com/watch?v=sMbQW7XNUZs
5. http://www.dba-oracle.com/art_builder_bitmap_join_idx.htm
6. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7030998/
7. https://www.geeksforgeeks.org/bitmap-indexing-in-dbms/
8. https://www.geeksforgeeks.org/indexing-in-databases-set-1/
9. https://en.wikipedia.org/wiki/Bitmap_index
10. https://roaringbitmap.org/about/
11. https://sdm.lbl.gov/~kewu/ps/LBNL-49627.pdf