CSC 244
HW 1
Description

Karan Mitra
Sac ID : 219032620

1. Description of Simulation of memory and disk storage -

- The relation structure consists of three related Java classes, the base of which is Tuple. This contains three integer attributes, namely, x,y and z. The class above this is the Block class which contains an array of 3 tuples which ensures data is stored and transported as a block.

- The memory structure is simulated using a class called MemoryBlock. This block uses the above mentioned relation structure to simulate the most granular / singular part of the memory structure. This memory block can store one block of tuples and an integer count, required by the intersection algorithm. The memory is simulated by creating an array of such blocks. This ensures that the blocks of the relations can be stored in the memory along with additional space to store the counts.

- Disk storage has been simulated by using an ArrayList of Blocks. This ensures that the disk is not limited and grows according to storage requirements. This is done so that extra disk storage is not reserved unless and until we do not require that space.

2. Example used to test the program -

The input, in this program, can be accepted from the user. The user needs to enter size of R and S and the attributes of the tuples when prompted.

Relation R (size 7)-

X = 1 Y = 1 Z = 1
X = 2 Y = 2 Z = 2
X = 2 Y = 2 Z = 2

X = 1 Y = 2 Z = 3
X = 2 Y = 3 Z = 4
X = 1 Y = 2 Z = 3

X = 3 Y = 4 Z = 5

Relation S (size 5)-

X = 2 Y = 2 Z = 2
X = 2 Y = 2 Z = 2
X = 2 Y = 3 Z = 4

X = 1 Y = 2 Z = 3
X = 1 Y = 2 Z = 3

S is split and stored as 2 blocks. It is then loaded into 2 memory blocks giving the following memory state -

Memory M -

Block 1
X = 2 Y = 2 Z = 2 Count = 2
X = 2 Y = 3 Z = 4 Count = 1

Block 2
X = 1 Y = 2 Z = 3 Count = 2

Using these distinct tuples and their counts, R is brought in block by block at the Mth block and compared, decrementing counts which are non-zero if the tuples match. While doing this, the tuple is put into the memory buffer.
The memory buffer is flushed to disk if it is full and once at the end of the program, giving the final disk state as follows -

Block 1
X = 2 Y = 2 Z = 2
X = 2 Y = 2 Z = 2
X = 1 Y = 2 Z = 3

Block 2
X = 2 Y = 3 Z = 4
X = 1 Y = 2 Z = 3

This is the output of the bag intersection of the two relations R & S.