A PERFORMANCE STUDY OF JOIN OPERATION USING OPENMP ON SHARCNET

CONTENTS

- Join Operation
- Sequential Join Operation
- Parallel Join Operation Using OpenMP
- Graphical Representation of Timing Results
- Challenges
- Conclusion

JOIN OPERATION

- One of the most investigated research area in the context of relational data model.
- Join operation is a performance critical as large amount of data is involved.
- Join operation have attracted most of the attention as:
    a. Most frequently used
    b. Expensive operation
    c. Data intensive relational operation
- The join I chose to study is **INNER JOIN OPERATION**.

SEQUENTIAL JOIN OPERATION

- Execution time of any join operation depends on:
  1. Size of the tables
  2. Number of tables and join attributes
  3. Number of matching tuples in a relation.

- Time & Effort has been spend by many researchers to use the joins efficiently.

- Best practice is to parallelize the join operation efficiently using multiple number of processors as execution time is inversely proportional to the number of processors.

PROGRAMMING PLATFORM, TECHNOLOGIES & DATA USED

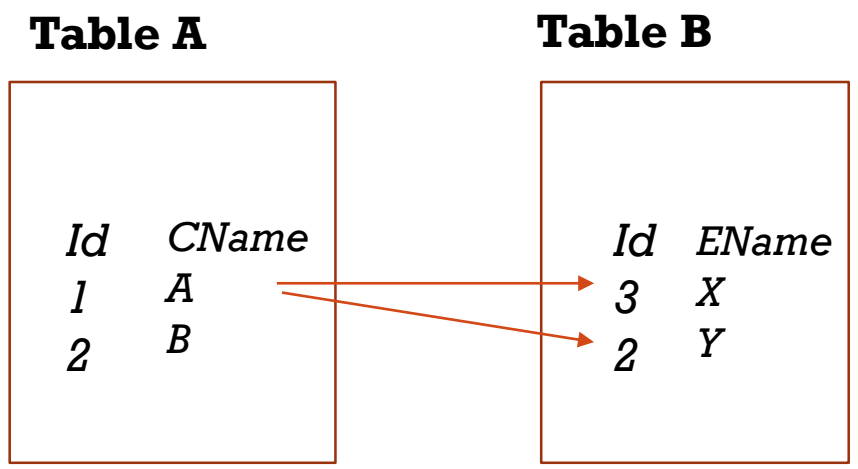
- C - Programming language
- OpenMP - for parallelization of the code
- SharcNet – A high performance computing platform to run the jobs
- Data - Random Data is generated using standard C functions.

PARALLEL INNER JOIN OPERATION USING OPENMP

For t1 in A:
For t2 in B:
if(t1,t2 have matched records):
Add(t1,t2) to the result

Parallel this part using OpenMP pragma directives

Time Consuming Process



Scans outer table

Reads inner table once for each row in the outer table

- Like Nested Loop Join
- Spilt the data over several threads.
- Each thread computes part of the join.
- Then, the results are merged.

GRAPHICAL RESULTS

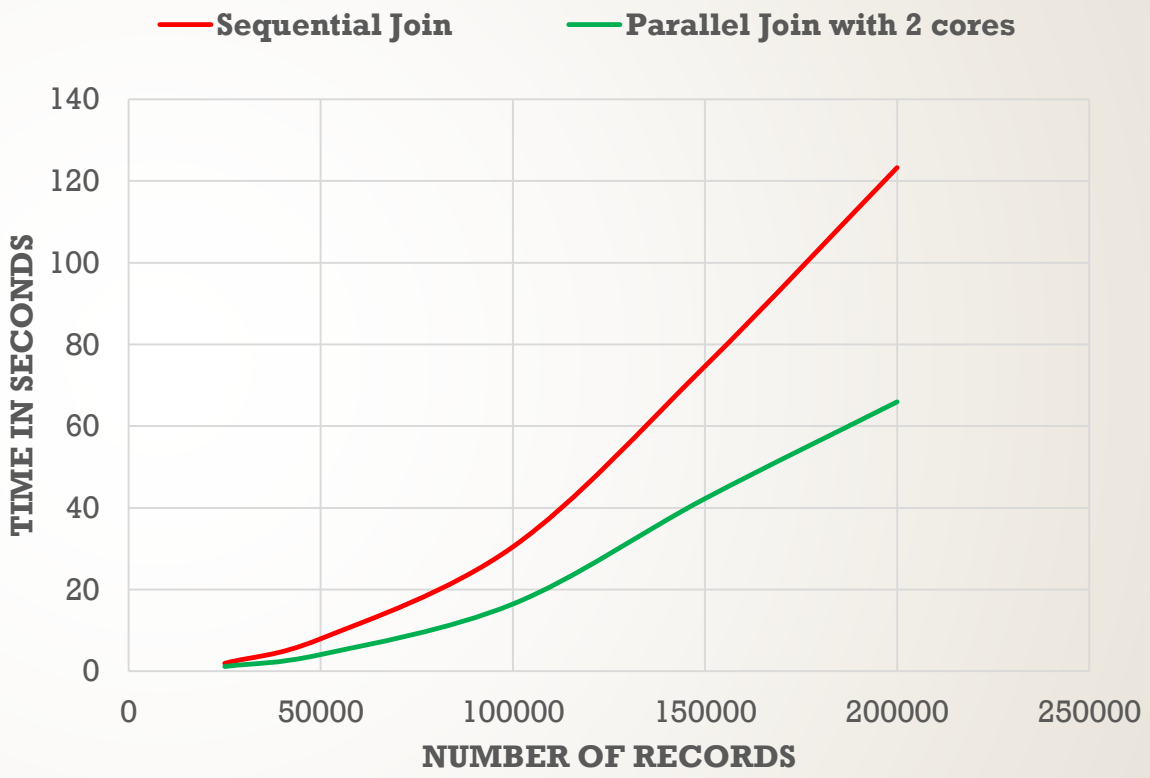
- Execution Time of Sequential and Parallel Inner Join Operation is presented:
    1. When the number of records are increased
    2. When the number of attributes are increased
    3. When the number of tables to be joined are increased

- Speed-Up and Efficiency Curves

EXECUTION TIME OF SEQUENTIAL & PARALLEL INNER JOIN OPERATION VS NUMBER OF RECORDS

- Inner Join Between Two tables
- Execution time is reduced to half of the sequential time when parallelised using openmp.

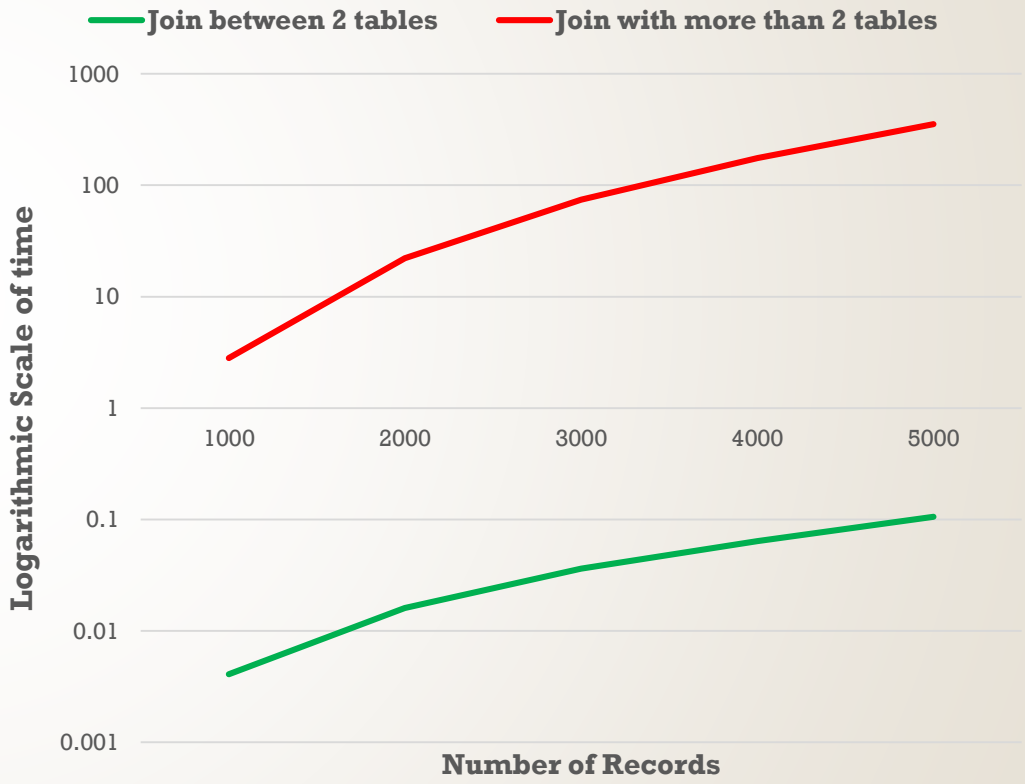
.



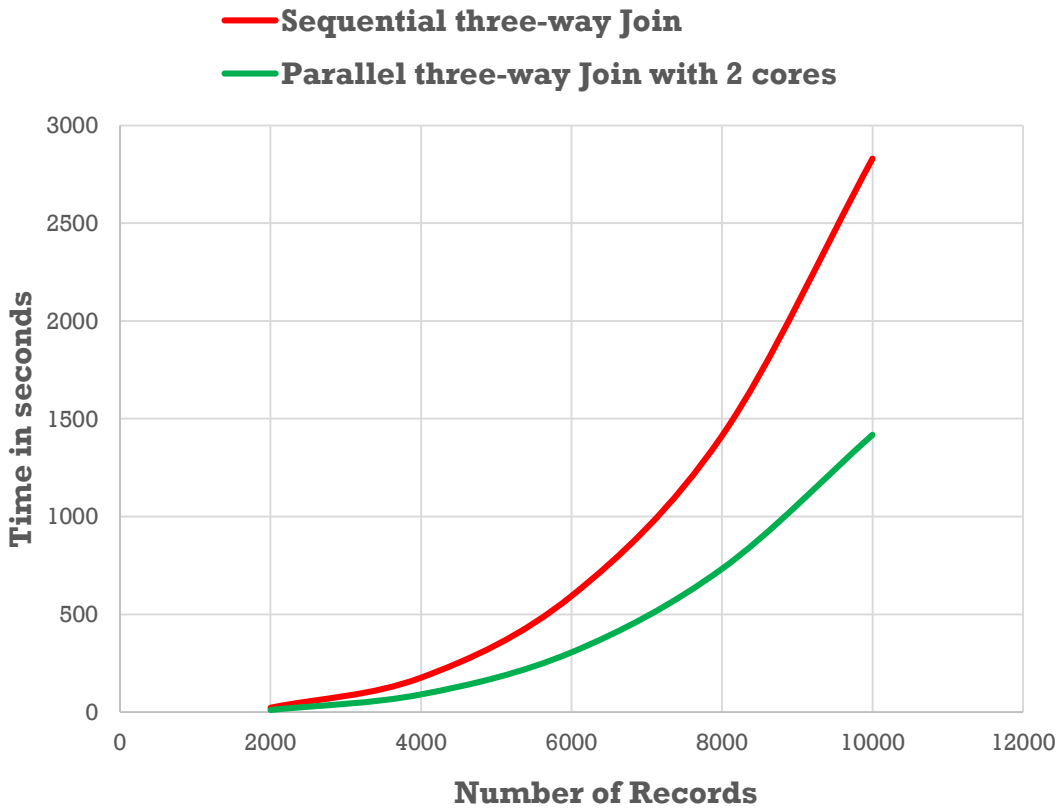
SEQUENTIAL EXECUTION TIME WHEN THE TABLES ARE INCREASED

- Complexity is increased when the tables are increased because of:
  - More nested loops
  - More Join attributes

- Sequential Time has increased a lot when more tables are joined.

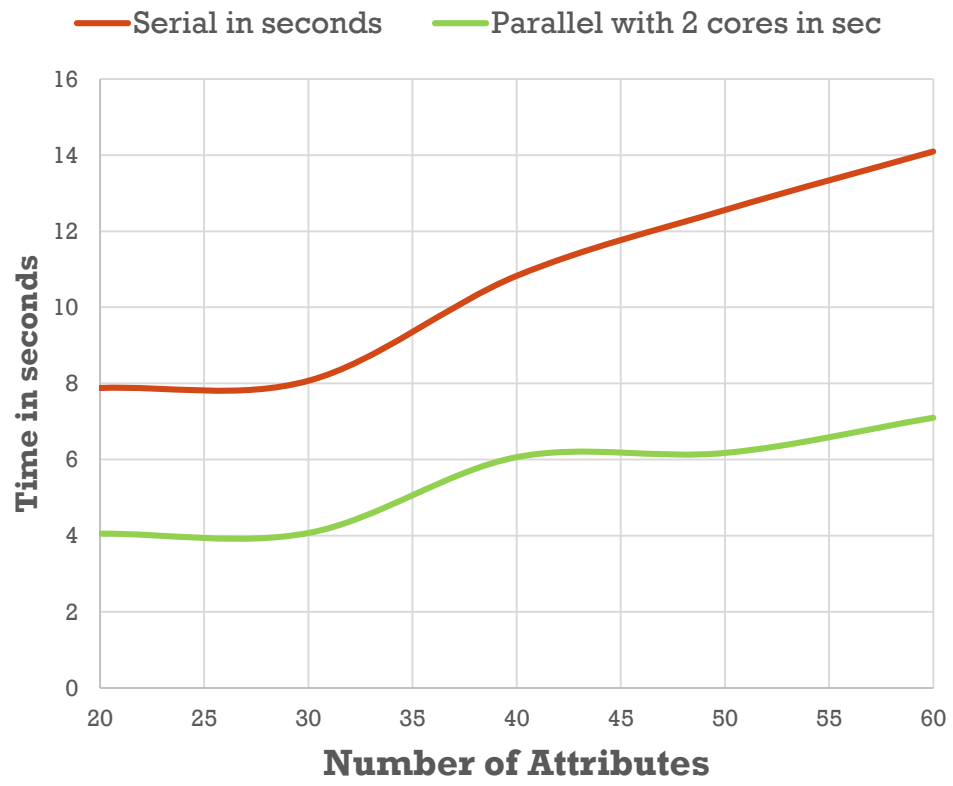


EXECUTION TIME OF SEQUENTIAL & PARALLEL JOIN OPERATION VS NUMBER OF RECORDS



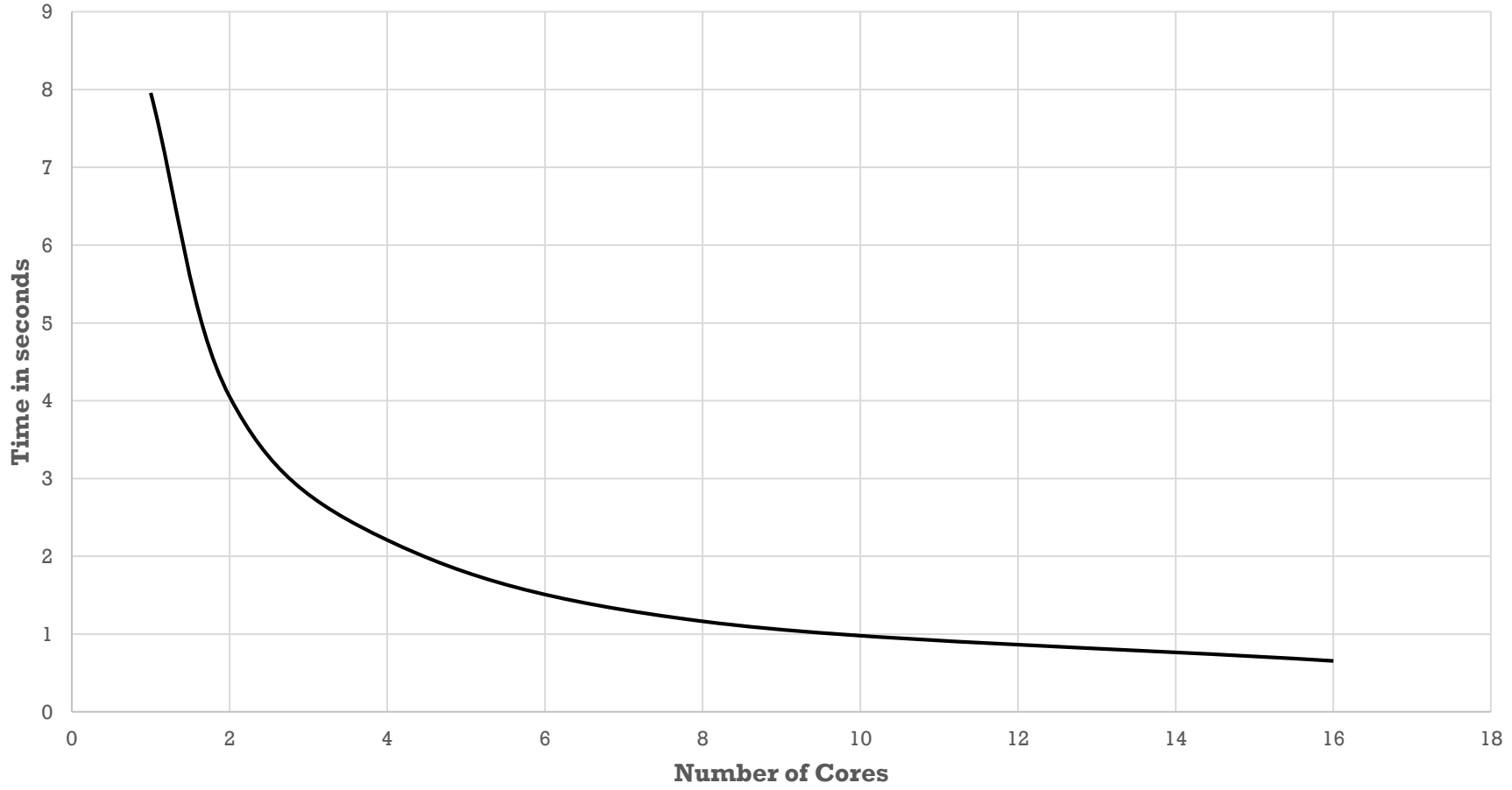
- Inner Join between three tables
- Execution is reduced to half of sequential time.
- Good performance when parallelized using OpenMP

EXECUTION TIME VS NUMBER OF ATTRIBUTES



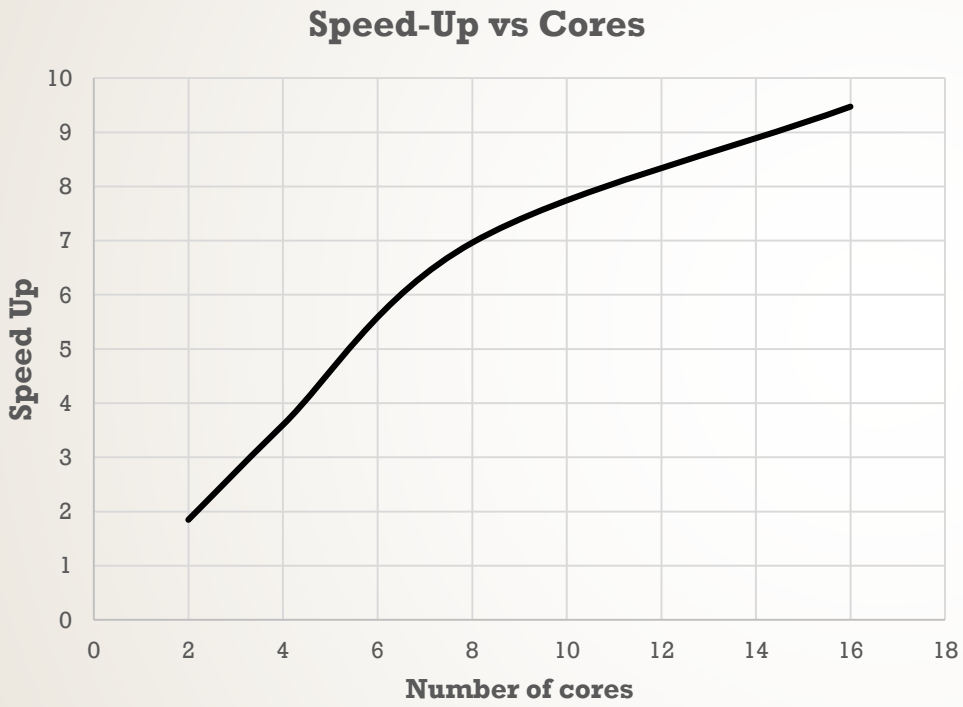
- Performance of join operation is also affected by number of attributes in each table.
- Increase in the number of attributes means increasing size of the tables
- Execution time scales linearly with number of attributes.

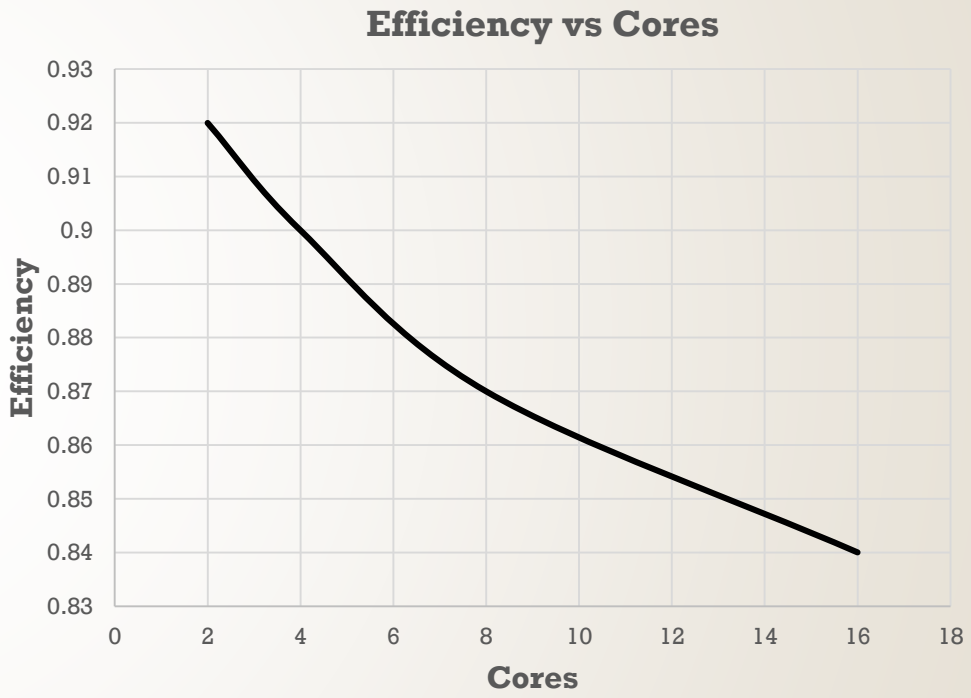
SCALABILITY OF A INNER JOIN BETWEEN TWO TABLES



Execution Time decreases as the number of cores are increased.

SPEED UP & EFFICIENCY USING OPENMP





Records=100000	Execution Time	Speed-Up	Efficiency
Sequential Time	30.456162		
Parallel with 2 cores	16.417028	1.85	0.92
Parallel with 4 cores	8.43908	3.61	0.9
Parallel with 8 cores	4.375433	6.96	0.87
Parallel with 16 cores	2.263359	13.46	0.84

CHALLENGES FACED

- Managing and processing exponentially growing volume of data
- Memory Management due to increase in the structure of tables and size of tables
- Timing Calculations of the Inner Join.

CONCLUSION

- Parallelisation of Join Operation is a better way to reduce the cost of join operation.
- Execution time of parallel inner join using OpenMP is very less as compared to sequential inner join.
- As more tables are joined, complexity is increased that demands more parallelisation.

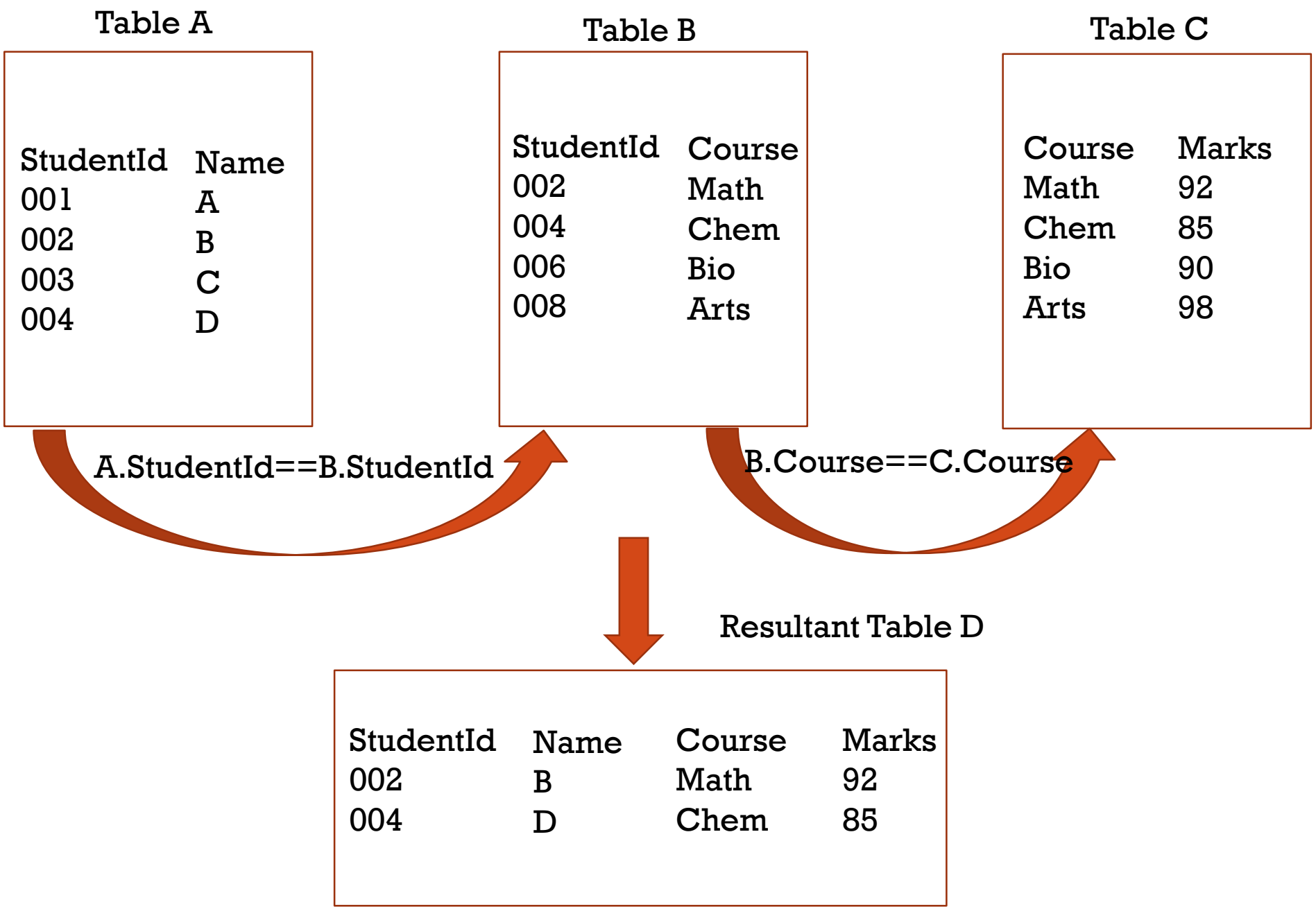


Table A
StudentId Name
001 A
002 B
003 C
004 D
Table B
StudentId Course
002 Math
004 Chem
006 Bio
008 Arts
Table C
Course Marks
Math 92
Chem 85
Bio 90
Arts 98
A.StudentId==B.StudentId
B.Course==C.Course
Resultant Table D
StudentId Name Course Marks
002 B Math 92
004 D Chem 85

THANK YOU!!

QUESTIONS?