Simulate the task of concurrency verification in DBMS
The system processes already executed transactions and only needs to efficiently check whether concurrent queries conflict with them.
- A list of insert and delete statements (transactions) is provided
- We need to figure out whether given predicates (queries) match the inserted or deleted data.

Step 1: Read and put transactions and queries into queues
Step 2: Concurrently process transactions
Step 3: Concurrently build indexes
Step 4: Concurrently validate queries and corresponding transactions
Step 5: Write results

OUR MISSION
To build a program that:
- Can handle a massive amount of transactions & queries
- Run as fast as possible

OUR APPROACH
- Minimize memory allocations / deallocations (avoid memory leaks)
- Use indexes as much as possible
- Parallelize every step

IMPROVEMENTS
- Use max-min index for low-cardinality columns
- Index tuples that are belonged to a range of small transactions instead of indexing tuples of each such transaction
- Try to find a good condition which leads to a very small number of matched tuples instead of combining query conditions using set intersection.

STATISTICS
1. Language: C++/make/std lib
2. Lines of code: 771
3. Time spent: 1 month
4. Third party: Intel® Threading Building Blocks (TBB) for multiple threading.