Users:

INSERT INTO public."User"(

username, email, passwod, "firstName", "lastName", country, "registrationDat")

VALUES

('Khaled47', 'Khaled47@gmail.com', '5e884898da28047151d0e56f8dc6292773603d0d6b8b4b9b883883a1541d3343', 'Khaled', 'Charaf', 'Lebanon', '2023-01-15'),

('AliFarhat', 'AliFarhat@gmail.com', '2d3e1312b4f10a8a7c0ebc7dc5ea8cf2cbca99e7c8bd6a57f188e7e1e5571d7', 'Ali', 'Farhat', 'Lebanon', '2022-10-21'),

('ay\_charaf', 'ay\_charaf@gmail.com', '3a63c03f43710ad8eae4c0d6de315df0d73f0b3b5f5ae8e09d4da598b968c66', 'Ayman', 'Charaf', 'Lebanon', '2023-06-08'),

('HussKiller', 'HussKiller@gmail.com', 'ad8b3b7d8755165555a80f8b0b5f196ea44d6c0a54f491b7a2a66e8dfdece0b', 'Hussein', 'Bakri', 'Lebanon', '2022-07-30'),

('ImanDestroyer', 'ImanDestroyer@gmail.com', 'ea4109f401e9aabbab4c5fb17c25f31e5c64bfa5e1efcdfcf4c9ec9910b896d', 'Iman', 'Ghalayini', 'Lebanon', '2023-03-04'),

('Danosour', 'Danosour@gmail.com', '1f548a2d8992b4d8a6cb424f3c6aeef19e719e5d158c6fb39e9ea2ae23f4570', 'Dana', 'Kossaybeyati', 'Lebanon', '2022-08-16'),

('HashTrash', 'HashTrash@gmail.com', 'f5297c5e46c2f95e5c700f946e77dd715583cfb41895fcb52e4d85b7eb38127', 'Hashem', 'Khodor', 'Lebanon', '2023-02-14'),

('TheJoker', 'TheJoker@gmail.com', 'efbc56e0d798f9372831ceab35f8c3d812ead0c4e1c1ac21f96c0f8bb4249f7', 'Omar', 'Ramadan', 'Lebanon', '2023-02-04'),

('MommyBoy', 'MommyBoy@gmail.com', 'fd1fc301decbde0a5ad5c4c27ae398d07bb9ec75c7e2c1b11a76d0068d15cd', 'Abdelatif', 'Saleh', 'Lebanon', '2023-01-29'),

('cutiepie', 'cutiepie@gmail.com', 'e2bf8da1bde8ed5e6ccf3f936da984ae11a2c0403c54f64eb69a1e7b1d3d62c', 'Lama', 'Hasbini', 'America', '2023-02-21'),

('Tourist', 'Tourist@gmail.com', '5e884898da28047151d0e56f8dc6292773603d0d6b8b4b9b883883a1541d3343', 'Gennady', 'Korotkevich', 'Belarus', '2022-08-16');

Befriends:

INSERT INTO public."Befriends"(

"friend1Username", "friend2Username", "timestamp")

VALUES ('Khaled47', 'AliFarhat', '2023-01-15'),

('Khaled47', 'ay\_charaf', '2022-10-21'),

('Khaled47', 'HussKiller', '2023-06-08'),

('Khaled47', 'ImanDestroyer', '2022-07-30'),

('AliFarhat', 'ay\_charaf', '2023-03-04'),

('AliFarhat', 'HussKiller', '2022-08-16'),

('AliFarhat', 'ImanDestroyer', '2023-02-14'),

('AliFarhat', 'TheJoker', '2023-02-04'),

('ay\_charaf', 'HussKiller', '2023-01-29'),

('ay\_charaf', 'ImanDestroyer', '2023-02-21'),

('HussKiller', 'ImanDestroyer', '2022-08-16'),

('HussKiller', 'HashTrash', '2023-01-15'),

('ImanDestroyer', 'Danosour', '2022-10-21'),

('ImanDestroyer', 'TheJoker', '2023-06-08'),

('Danosour', 'HashTrash', '2022-07-30'),

('Danosour', 'TheJoker', '2023-03-04'),

('HashTrash', 'TheJoker', '2022-08-16'),

('HashTrash', 'MommyBoy', '2023-02-14'),

('TheJoker', 'MommyBoy', '2023-02-04'),

('TheJoker', 'cutiepie', '2023-01-29');

Contests:

INSERT INTO public."Contest"(

"roundNumber", name, divsion, "startTimestamp", length, description)

VALUES

-- Contest 1

(1, 'HusseinBakri Championship', 1, '2023-01-15 08:00:00', '03:00:00', 'The ultimate coding challenge for elite programmers.'),

-- Contest 2

(2, 'AUB Showdown', 2, '2023-02-05 10:30:00', '02:30:00', 'Test your algorithmic skills in this exciting contest.'),

-- Contest 3

(3, 'LAU Titans Challenge', 3, '2023-03-20 09:15:00', '03:00:00', 'An opportunity for newcomers to prove their tech prowess.'),

-- Contest 4

(4, 'CodeCrafters Duel', 1, '2023-04-10 14:00:00', '04:00:00', 'A battle of wits among the coding elite.'),

-- Contest 5

(5, 'AlgorithmShip', 2, '2023-05-02 11:45:00', '02:30:00', 'Marvel at the magic of algorithms in this contest and get a Scholarship.'),

-- Contest 6

(6, 'Rookie Coding Challenge', 3, '2023-06-15 10:00:00', '03:00:00', 'An ideal platform for budding programmers to shine.'),

-- Contest 7

(7, 'CodeFusion Challenge', 1, '2023-07-08 13:30:00', '04:00:00', 'Merge your coding skills in this fusion of challenges.'),

-- Contest 8

(8, 'Algorithm Explorers', 2, '2023-08-19 11:15:00', '02:30:00', 'Explore the depths of algorithms in this exciting contest.'),

-- Contest 9

(9, 'Dakowdas Enthusiasts', 3, '2023-09-25 09:45:00', '03:00:00', 'A showdown for tech enthusiasts to conquer.'),

-- Contest 10

(10, 'Bakri Finale', 1, '2023-10-15 14:15:00', '04:00:00', 'Place top 1 to get a 100 on Bakris database course at AUB!');

ProgrammingProblem:

INSERT INTO public."ProgrammingProblem"("problemID", "difficultyLevel", title, description, "timeLimit", "memoryLimit", "roundNumber")

VALUES

-- Round 1

(1, 'A', 'Two Sum Problem', 'Given an array of integers and a target sum, find two numbers that add up to the target sum. Return the indices of the two elements.', 2, 256, 1),

(2, 'B', 'Palindrome Check', 'Write a program to check if a given string is a palindrome or not. Consider alphanumeric characters and ignore case.', 3, 512, 1),

(3, 'C', 'Maximum Subarray', 'Find the contiguous subarray with the largest sum in an array of integers. Return the starting and ending indices of the subarray.', 4, 128, 1),

(4, 'D', 'Factorial Calculation', 'Write a program to calculate the factorial of a given non-negative integer. Handle large factorials efficiently.', 5, 256, 1),

(5, 'E', 'Graph Traversal', 'Traverse a graph and perform depth-first or breadth-first search. Implement both DFS and BFS algorithms for a given graph.', 1, 512, 1),

(6, 'F', '0/1 Knapsack Problem', 'Solve the 0/1 Knapsack problem using dynamic programming. Given a set of items with weights and values, maximize the value within a limited weight capacity.', 4, 256, 1),

-- Sortings , BruteForce, Greedy , DP , Graphs , DP

-- Round 2

(7, 'A', 'Greatest Common Divisor', 'Find the greatest common divisor (GCD) of two positive integers using an efficient algorithm.', 1, 128, 2),

(8, 'B', 'Binary Tree Height', 'Calculate the height of a binary tree, which is the length of the longest path from the root to a leaf node.', 2, 512, 2),

(9, 'C', 'Longest Common Substring', 'Find the longest common substring between two given strings. Return the substring itself.', 3, 256, 2),

(10, 'D', 'Sparse Matrix Multiplication', 'Perform multiplication of two sparse matrices efficiently, taking advantage of their sparsity.', 4, 128, 2),

(11, 'E', 'Shortest Path in Maze', 'Find the shortest path from the start to the end in a maze using breadth-first search.', 5, 512, 2),

(12, 'F', 'Matrix Chain Multiplication', 'Solve the matrix chain multiplication problem using dynamic programming to minimize multiplication cost.', 1, 256, 2),

-- NumberTheory , Graphs , DP ,Matrices,Graphs,DiveAndConquer

-- Round 3

(13, 'A', 'Rotated Sorted Array Search', 'Search for a target element in a rotated sorted array efficiently using a modified binary search.', 2, 256, 3),

(14, 'B', 'Binary Tree Traversal', 'Perform various traversals (inorder, preorder, postorder) on a binary tree and print the node values.', 3, 512, 3),

(15, 'C', 'Regular Expression Matcher', 'Implement a regular expression matcher that can match strings against regular expressions.', 4, 128, 3),

(16, 'D', 'Sudoku Solver', 'Solve a Sudoku puzzle by filling in the empty cells with the correct numbers while adhering to the rules.', 5, 256, 3),

(17, 'E', 'Chess Game Logic', 'Implement the logic for a basic chess game, including moving pieces and checking for checkmate conditions.', 1, 512, 3),

(18, 'F', 'Longest Common Subsequence', 'Find the longest common subsequence of two given strings using dynamic programming.', 4, 256, 3),

-- Binary Search , Graphs, Graphs , Brute Force, Graphs, DP

-- Round 4

(25, 'A', 'Longest Increasing Subsequence', 'Find the length of the longest increasing subsequence in an array of integers using dynamic programming.', 1, 128, 4),

(26, 'B', 'Shortest Path', 'Implement Dijkstras algorithm to find the shortest path between two nodes in a weighted graph.', 2, 512, 4),

(27, 'C', 'Knuth-Morris-Pratt', 'Write an algorithm to search for a pattern in a text using the Knuth-Morris-Pratt string matching algorithm.', 3, 256, 4),

(28, 'D', 'Prime Factorization', 'Factorize a given integer into its prime factors and their exponents.', 4, 128, 4),

(29, 'E', 'Maximum Subarray Sum', 'Find the maximum sum of a contiguous subarray within an array of integers using dynamic programming.', 5, 512, 4),

(30, 'F', 'Traveling Salesman', 'Solve the Traveling Salesman Problem using dynamic programming to find the shortest tour that visits a set of cities.', 1, 256, 4),

-- DP , Graphs , String Matching , NumberTheory , DP , DP

-- Round 5

(31, 'A', 'Longest Palindromic Substring', 'Find the longest palindromic substring in a given string using dynamic programming.', 1, 128, 5),

(32, 'B', 'Minimum Spanning Tree', 'Implement Kruskals algorithm to find the minimum spanning tree of a connected graph.', 2, 512, 5),

(33, 'C', 'Boyer-Moore', 'Write an algorithm to search for a pattern in a text using the Boyer-Moore string matching algorithm.', 3, 256, 5),

(34, 'D', 'Modular Exponentiation', 'Compute large powers modulo a prime number efficiently using modular exponentiation.', 4, 128, 5),

(35, 'E', 'Coin Change', 'Find the number of ways to make change for a given amount using a set of coin denominations.', 5, 512, 5),

(36, 'F', 'Longest Common Subarray', 'Find the longest common subarray between two arrays using dynamic programming.', 1, 256, 5),

-- DP , Graphs , String Matching , Number Theory , DP , DP

-- Round 6

(37, 'A', 'Maximum Subarray Sum (Non-Contiguous)', 'Find the maximum sum of a non-contiguous subarray within an array of integers using dynamic programming.', 2, 256, 6),

(38, 'B', 'Topological Sorting', 'Implement topological sorting for directed acyclic graphs to find a linear order of vertices.', 3, 512, 6),

(39, 'C', 'Rabin-Karp', 'Write an algorithm to search for a pattern in a text using the Rabin-Karp string matching algorithm.', 4, 256, 6),

(40, 'D', 'Greatest Common Divisor', 'Find the greatest common divisor (GCD) of two integers using the Euclidean algorithm.', 5, 128, 6),

(41, 'E', 'Longest Increasing Subsequence', 'Find the length of the longest increasing subsequence in an array of integers using dynamic programming.', 1, 512, 6),

(42, 'F', 'Bipartite Graph Check', 'Determine if a given graph is bipartite or not using graph coloring algorithms.', 2, 256, 6),

-- DP , Graphs , String Matching , NumberTheory, DP, Graphs

-- Round 7

(43, 'A', 'Matrix Chain Multiplication', 'Solve the matrix chain multiplication problem using dynamic programming to minimize multiplication cost.', 3, 128, 7),

(44, 'B', 'Strongly Connected Components', 'Implement Tarjans algorithm to find strongly connected components in a directed graph.', 4, 512, 7),

(45, 'C', 'Aho-Corasick', 'Write an algorithm to search for multiple patterns in a text using the Aho-Corasick string matching algorithm.', 5, 256, 7),

(46, 'D', 'Sieve of Eratosthenes', 'Generate prime numbers up to a given limit using the Sieve of Eratosthenes algorithm.', 1, 128, 7),

(47, 'E', 'Longest Common Subsequence', 'Find the longest common subsequence of two sequences using dynamic programming.', 2, 512, 7),

(48, 'F', 'Shortest Path (Negative Weight)', 'Implement the Bellman-Ford algorithm to find the shortest path in a graph with negative weight edges.', 3, 256, 7),

-- DP , Graphs, String Matching , NumberTheory , DP , Graphs

-- Round 8

(49, 'A', 'Coin Change (Minimum Coins)', 'Find the minimum number of coins needed to make change for a given amount using dynamic programming.', 4, 128, 8),

(50, 'B', 'Articulation Points and Bridges', 'Find articulation points and bridges in an undirected graph using depth-first search.', 5, 512, 8),

(51, 'C', 'Longest Common Prefix Array', 'Compute the longest common prefix array for a set of strings to optimize string matching.', 1, 256, 8),

(52, 'D', 'Modular Multiplicative Inverse', 'Calculate the modular multiplicative inverse of an integer modulo a prime using the extended Euclidean algorithm.', 2, 128, 8),

(53, 'E', 'Rod Cutting', 'Solve the rod cutting problem to maximize profit using dynamic programming.', 3, 512, 8),

(54, 'F', 'Maximum Flow', 'Find the maximum flow in a flow network using the Ford-Fulkerson algorithm.', 4, 256, 8),

-- DP , Graphs , String Matching, NumberTheory, DP , Graphs

-- Round 9

(55, 'A', 'Longest Palindromic Subsequence', 'Find the length of the longest palindromic subsequence in a given string using dynamic programming.', 5, 128, 9),

(56, 'B', 'Minimum Spanning Tree (Prims Algorithm)', 'Implement Prims algorithm to find the minimum spanning tree of a connected graph.', 1, 512, 9),

(57, 'C', 'Z Algorithm', 'Compute the Z array to efficiently search for a pattern in a text using the Z algorithm.', 2, 256, 9),

(58, 'D', 'Chinese Remainder Theorem', 'Solve simultaneous modular congruences using the Chinese Remainder Theorem.', 3, 128, 9),

(59, 'E', 'Edit Distance', 'Find the edit distance (Levenshtein distance) between two strings using dynamic programming.', 4, 512, 9),

(60, 'F', 'Hamiltonian Path', 'Find a Hamiltonian path in a directed or undirected graph using backtracking or dynamic programming.', 5, 256, 9),

-- DP, Graphs, String Matching, NumberTheory, DP, Graphs

-- Round 10

(61, 'A', 'Longest Zigzag Subsequence', 'Find the length of the longest zigzag subsequence in an array of integers using dynamic programming.', 1, 128, 10),

(62, 'B', 'Maximum Bipartite Matching', 'Find the maximum cardinality matching in a bipartite graph using augmenting paths.', 2, 512, 10),

(63, 'C', 'Suffix Array', 'Construct a suffix array for a given string to enable efficient substring searches and pattern matching.', 3, 256, 10),

(64, 'D', 'Lucas Theorem', 'Apply Lucass Theorem to compute binomial coefficients modulo a prime number.', 4, 128, 10),

(65, 'E', 'Longest Increasing Subarray', 'Find the longest increasing subarray within an array of integers using dynamic programming.', 5, 512, 10),

(66, 'F', 'Maximum Planar Subgraph', 'Find the maximum planar subgraph of a planar graph using planarity testing and Kuratowskis theorem.', 1, 256, 10);

-- DP , Graph , String Matching , NumberTheory , DP, Graph

Solutions:

INSERT INTO public."Solution"("problemID", "solutionID", "sourceCode")

VALUES

(1, 1, '#include <iostream> using namespace std ; int main(){//solution for Problem 1 return 0 ;}'),

(2, 2, '#include <iostream> using namespace std ; int main(){//solution for Problem 2 return 0 ;}'),

(3, 3, '#include <iostream> using namespace std ; int main(){//solution for Problem 3 return 0 ;}'),

(4, 4, '#include <iostream> using namespace std ; int main(){//solution for Problem 4 return 0 ;}'),

(5, 5, '#include <iostream> using namespace std ; int main(){//solution for Problem 5 return 0 ;}'),

(6, 6, '#include <iostream> using namespace std ; int main(){//solution for Problem 6 return 0 ;}'),

(7, 7, '#include <iostream> using namespace std ; int main(){//solution for Problem 7 return 0 ;}'),

(8, 8, '#include <iostream> using namespace std ; int main(){//solution for Problem 8 return 0 ;}'),

(9, 9, '#include <iostream> using namespace std ; int main(){//solution for Problem 9 return 0 ;}'),

(10, 10, '#include <iostream> using namespace std ; int main(){//solution for Problem 10 return 0 ;}'),

(11, 11, '#include <iostream> using namespace std ; int main(){//solution for Problem 11 return 0 ;}'),

(12, 12, '#include <iostream> using namespace std ; int main(){//solution for Problem 12 return 0 ;}'),

(13, 13, '#include <iostream> using namespace std ; int main(){//solution for Problem 13 return 0 ;}'),

(14, 14, '#include <iostream> using namespace std ; int main(){//solution for Problem 14 return 0 ;}'),

(15, 15, '#include <iostream> using namespace std ; int main(){//solution for Problem 15 return 0 ;}'),

(16, 16, '#include <iostream> using namespace std ; int main(){//solution for Problem 16 return 0 ;}'),

(17, 17, '#include <iostream> using namespace std ; int main(){//solution for Problem 17 return 0 ;}'),

(18, 18, '#include <iostream> using namespace std ; int main(){//solution for Problem 18 return 0 ;}'),

(25, 25, '#include <iostream> using namespace std ; int main(){//solution for Problem 25 return 0 ;}'),

(26, 26, '#include <iostream> using namespace std ; int main(){//solution for Problem 26 return 0 ;}'),

(27, 27, '#include <iostream> using namespace std ; int main(){//solution for Problem 27 return 0 ;}'),

(28, 28, '#include <iostream> using namespace std ; int main(){//solution for Problem 28 return 0 ;}'),

(29, 29, '#include <iostream> using namespace std ; int main(){//solution for Problem 29 return 0 ;}'),

(30, 30, '#include <iostream> using namespace std ; int main(){//solution for Problem 30 return 0 ;}'),

(31, 31, '#include <iostream> using namespace std ; int main(){//solution for Problem 31 return 0 ;}'),

(32, 32, '#include <iostream> using namespace std ; int main(){//solution for Problem 32 return 0 ;}'),

(33, 33, '#include <iostream> using namespace std ; int main(){//solution for Problem 33 return 0 ;}'),

(34, 34, '#include <iostream> using namespace std ; int main(){//solution for Problem 34 return 0 ;}'),

(35, 35, '#include <iostream> using namespace std ; int main(){//solution for Problem 35 return 0 ;}'),

(36, 36, '#include <iostream> using namespace std ; int main(){//solution for Problem 36 return 0 ;}'),

(37, 37, '#include <iostream> using namespace std ; int main(){//solution for Problem 37 return 0 ;}'),

(38, 38, '#include <iostream> using namespace std ; int main(){//solution for Problem 38 return 0 ;}'),

(39, 39, '#include <iostream> using namespace std ; int main(){//solution for Problem 39 return 0 ;}'),

(40, 40, '#include <iostream> using namespace std ; int main(){//solution for Problem 40 return 0 ;}'),

(41, 41, '#include <iostream> using namespace std ; int main(){//solution for Problem 41 return 0 ;}'),

(42, 42, '#include <iostream> using namespace std ; int main(){//solution for Problem 42 return 0 ;}'),

(43, 43, '#include <iostream> using namespace std ; int main(){//solution for Problem 43 return 0 ;}'),

(44, 44, '#include <iostream> using namespace std ; int main(){//solution for Problem 44 return 0 ;}'),

(45, 45, '#include <iostream> using namespace std ; int main(){//solution for Problem 45 return 0 ;}'),

(46, 46, '#include <iostream> using namespace std ; int main(){//solution for Problem 46 return 0 ;}'),

(47, 47, '#include <iostream> using namespace std ; int main(){//solution for Problem 47 return 0 ;}'),

(48, 48, '#include <iostream> using namespace std ; int main(){//solution for Problem 48 return 0 ;}'),

(49, 49, '#include <iostream> using namespace std ; int main(){//solution for Problem 49 return 0 ;}'),

(50, 50, '#include <iostream> using namespace std ; int main(){//solution for Problem 50 return 0 ;}'),

(51, 51, '#include <iostream> using namespace std ; int main(){//solution for Problem 51 return 0 ;}'),

(52, 52, '#include <iostream> using namespace std ; int main(){//solution for Problem 52 return 0 ;}'),

(53, 53, '#include <iostream> using namespace std ; int main(){//solution for Problem 53 return 0 ;}'),

(54, 54, '#include <iostream> using namespace std ; int main(){//solution for Problem 54 return 0 ;}'),

(55, 55, '#include <iostream> using namespace std ; int main(){//solution for Problem 55 return 0 ;}'),

(56, 56, '#include <iostream> using namespace std ; int main(){//solution for Problem 56 return 0 ;}'),

(57, 57, '#include <iostream> using namespace std ; int main(){//solution for Problem 57 return 0 ;}'),

(58, 58, '#include <iostream> using namespace std ; int main(){//solution for Problem 58 return 0 ;}'),

(59, 59, '#include <iostream> using namespace std ; int main(){//solution for Problem 59 return 0 ;}'),

(60, 60, '#include <iostream> using namespace std ; int main(){//solution for Problem 60 return 0 ;}'),

(61, 61, '#include <iostream> using namespace std ; int main(){//solution for Problem 61 return 0 ;}'),

(62, 62, '#include <iostream> using namespace std ; int main(){//solution for Problem 62 return 0 ;}'),

(63, 63, '#include <iostream> using namespace std ; int main(){//solution for Problem 63 return 0 ;}'),

(64, 64, '#include <iostream> using namespace std ; int main(){//solution for Problem 64 return 0 ;}'),

(65, 65, '#include <iostream> using namespace std ; int main(){//solution for Problem 65 return 0 ;}'),

(66, 66, '#include <iostream> using namespace std ; int main(){//solution for Problem 66 return 0 ;}');

TestCase:

INSERT INTO public."TestCase"("problemID", "testCaseNumber", "input", "expectedOutput")

VALUES

(1, 1, '2 7', '9'),

(2, 1, 'racecar', '1'),

(3, 1, '-2 1 3 -4 5', '3'),

(4, 1, '6', '720'),

(5, 1, '0 1 2 3 4', '0 1 3 2 4'),

(6, 1, '4 7', '11'),

(7, 1, '10 20 30', '10'),

(8, 1, '5', '1'),

(9, 1, 'programming contest', '0'),

(10, 1, '1 2 3 4 5', '15'),

(11, 1, '8 5', '40'),

(12, 1, '10 20 30', '10 20 30'),

(13, 1, 'programming', 'false'),

(14, 1, 'algorithm is fun', 'true'),

(15, 1, '4 7 3 2 8', '9 2 6 4 10'),

(16, 1, '5 2 1 8 6 3', '5 4 5 9 10 11'),

(17, 1, '15 10 5', '5'),

(18, 1, '4 5 6 7', '4 3 2 1'),

(25, 1, '3', '4'),

(26, 1, '4 6 7', '9'),

(27, 1, 'programming', 'false'),

(28, 1, 'algorithm is fun', 'true'),

(29, 1, '5 7 9', '5'),

(30, 1, '10 20 30', '10'),

(31, 1, 'racecar', '1'),

(32, 1, 'hello', '0'),

(33, 1, 'level', '1'),

(34, 1, '10 20 30', '30'),

(35, 1, '-2 1 3 -4 5', '3'),

(36, 1, '5', '15'),

(37, 1, '0 1 2 3 4', '5'),

(38, 1, '4 7', '8'),

(39, 1, '10 20 30', '20'),

(40, 1, '6', '15'),

(41, 1, '5 10', '6'),

(42, 1, '8 16', '6'),

(43, 1, '7 11', '1'),

(44, 1, '12 18', '9'),

(45, 1, '6', '20'),

(46, 1, '0', '1'),

(47, 1, '20', '683'),

(48, 1, 'programming', 'false'),

(49, 1, 'coding is fun', 'true'),

(50, 1, '5 4 3 2 1', '5 4 3 2 1'),

(51, 1, '1 2 3 4 5', '120'),

(52, 1, '0 0 0 0 0', '1'),

(53, 1, '8 5', '40'),

(54, 1, '10 20 30', '30'),

(55, 1, 'programming contest', '0'),

(56, 1, 'racecar', '1'),

(57, 1, 'hello', '0'),

(58, 1, 'level', '1'),

(59, 1, '10 20 30', '30'),

(60, 1, 'algorithm is fun', '1'),

(61, 1, '12 18', '36'),

(62, 1, '7 21', '7'),

(63, 1, '9 12', '3'),

(64, 1, '5 2 1 8 6 3', '11'),

(65, 1, '15 10 5', '10'),

(66, 1, '4 5 6 7', '4');

Tags:

INSERT INTO public."Tag"("problemID", "tag")

VALUES

-- Tags for Problems 1 to 18 and 25 to 66

(1, 'Sortings'),

(2, 'BruteForce'),

(3, 'Greedy'),

(4, 'DP'),

(5, 'Graphs'),

(6, 'DP'),

(7, 'NumberTheory'),

(8, 'Graphs'),

(9, 'DP'),

(10, 'Matrices'),

(11, 'Graphs'),

(12, 'DivideAndConquer'),

(13, 'BinarySearch'),

(14, 'Graphs'),

(15, 'Graphs'),

(16, 'BruteForce'),

(17, 'Graphs'),

(18, 'DP'),

(25, 'DP'),

(26, 'NumberTheory'),

(27, 'DP'),

(28, 'DP'),

(29, 'Graphs'),

(30, 'StringMatching'),

(31, 'NumberTheory'),

(32, 'DP'),

(33, 'DP'),

(34, 'DP'),

(35, 'Graphs'),

(36, 'StringMatching'),

(37, 'NumberTheory'),

(38, 'DP'),

(39, 'Graphs'),

(40, 'DP'),

(41, 'Graphs'),

(42, 'StringMatching'),

(43, 'NumberTheory'),

(44, 'DP'),

(45, 'Graphs'),

(46, 'DP'),

(47, 'Graphs'),

(48, 'StringMatching'),

(49, 'NumberTheory'),

(50, 'DP'),

(51, 'Graphs'),

(52, 'DP'),

(53, 'Graphs'),

(54, 'StringMatching'),

(55, 'NumberTheory'),

(56, 'DP'),

(57, 'Graphs'),

(58, 'DP'),

(59, 'Graphs'),

(60, 'StringMatching'),

(61, 'NumberTheory'),

(62, 'DP'),

(63, 'Graphs'),

(64, 'DP'),

(65, 'Graphs'),

(66, 'StringMatching');

Contest Creator:

INSERT INTO public."ContestCreator"(

"creatorUsername", "assessmentScore")

VALUES ('HussKiller', 90),

('ImanDestroyer', 90);

Organizes:

NSERT INTO public."Organizes"("creatorUsername", "roundNumber")

VALUES

('HussKiller', 1),

('HussKiller', 2),

('HussKiller', 3),

('HussKiller', 4),

('HussKiller', 5),

('ImanDestroyer', 6),

('ImanDestroyer', 7),

('ImanDestroyer', 8),

('ImanDestroyer', 9),

('ImanDestroyer', 10);

Battles:

INSERT INTO public."Battle"("battleID", "startTimestamp", "endTimestamp")

VALUES

(1, '2023-01-15 14:00:00', '2023-01-15 16:00:00'),

(2, '2023-02-05 13:30:00', '2023-02-05 15:30:00'),

(3, '2023-03-20 11:45:00', '2023-03-20 13:45:00'),

(4, '2023-04-10 09:15:00', '2023-04-10 11:15:00'),

(5, '2023-05-02 10:00:00', '2023-05-02 12:00:00'),

(6, '2023-06-15 08:30:00', '2023-06-15 10:30:00'),

(7, '2023-07-08 14:45:00', '2023-07-08 16:45:00'),

(8, '2023-08-19 12:15:00', '2023-08-19 14:15:00'),

(9, '2023-09-25 13:00:00', '2023-09-25 15:00:00'),

(10, '2023-10-15 11:30:00', '2023-10-15 13:30:00');

Joins:

INSERT INTO public."Joins"(username, "battleID", "isWinner")

VALUES

('Khaled47', 1, true),

('AliFarhat', 1, false),

('ay\_charaf', 2, true),

('HussKiller', 2, false),

('ImanDestroyer', 3, true),

('Danosour', 3, false),

('HashTrash', 4, true),

('TheJoker', 4, false),

('MommyBoy', 5, true),

('cutiepie', 5, false),

('Tourist', 6, true),

('Khaled47', 6, false),

('MommyBoy', 7, true),

('HashTrash', 7, false),

('HashTrash', 8, true),

('Danosour', 8, false),

('ImanDestroyer', 9, true),

('HussKiller', 9, false),

('Tourist', 10, true),

('HashTrash', 10, false);

contestants :

INSERT INTO public."Contestant"("contestantUsername", rating)

VALUES

('Khaled47', 0),

('AliFarhat', 0),

('ay\_charaf', 0),

('Danosour', 0),

('HashTrash', 0),

('TheJoker', 0),

('MommyBoy', 0),

('cutiepie', 0),

('Tourist', 0);

ConsistsOf:

NSERT INTO public."ConsistsOf"("battleID", "problemID")

VALUES

(1, 1),

(1, 2),

(1, 3),

(2, 7),

(2, 8),

(2, 9),

(3, 13),

(3, 14),

(3, 15),

(4, 25),

(4, 26),

(4, 27),

(5, 31),

(5, 32),

(5, 33),

(6, 37),

(6, 38),

(6, 39),

(7, 43),

(7, 44),

(7, 45),

(8, 49),

(8, 50),

(8, 51),

(9, 55),

(9, 56),

(9, 57),

(10, 61),

(10, 62),

(10, 63);

competesIn :

-- For user Khaled47

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('Khaled47', 1, 400, 1, '2023-01-10 08:30:00'), -- d

('Khaled47', 2, -100, 2, '2023-01-15 10:45:00'),-- d

('Khaled47', 3, 500, 3, '2023-01-22 13:15:00'), -- d

('Khaled47', 4, 500, 3, '2023-01-28 16:20:00'); -- d

-- For user AliFarhat

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('AliFarhat', 5, 500, 1, '2023-02-05 09:00:00'), --d

('AliFarhat', 6, 750, 1, '2023-02-10 11:45:00'), -- d

('AliFarhat', 7, -250, 3, '2023-02-16 14:30:00'), -- d

('AliFarhat', 8, 500, 1, '2023-02-22 16:45:00'); -- d

-- For user ay\_charaf

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('ay\_charaf', 9, 600, 1, '2023-03-05 08:15:00'), -- d

('ay\_charaf', 10, 400, 2, '2023-03-10 09:30:00'), -- d

('ay\_charaf', 1, 400, 2, '2023-03-16 12:45:00'), -- d

('ay\_charaf', 2, -100, 3, '2023-03-22 14:00:00'); -- d

-- For user Danosour

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('Danosour', 3, 600, 2, '2023-04-05 09:30:00'), -- d

('Danosour', 4, 600, 2, '2023-04-10 11:15:00'), -- d

('Danosour', 5, 100, 4, '2023-04-16 14:30:00'), -- d

('Danosour', 6, -1, 4, '2023-04-22 16:45:00'); -- d

-- For user HashTrash

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('HashTrash', 7, 601, 1, '2023-05-05 08:45:00'), -- d

('HashTrash', 8, 300, 2, '2023-05-10 10:30:00'), -- d

('HashTrash', 9, 200, 3, '2023-05-16 13:15:00'), -- d

('HashTrash', 10, 200, 3, '2023-05-22 15:30:00'); -- d

-- For user TheJoker

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('TheJoker', 1, 200, 4, '2023-06-05 09:00:00'), -- d

('TheJoker', 2, 500, 1, '2023-06-10 11:15:00'), -- d

('TheJoker', 3, -100, 4, '2023-06-16 13:30:00'), -- d

('TheJoker', 4, 400, 4, '2023-06-22 15:45:00'); -- d

-- For user MommyBoy

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('MommyBoy', 5, 250, 3, '2023-07-05 08:30:00'), -- d

('MommyBoy', 6, 250, 3, '2023-07-10 10:45:00'), -- d

('MommyBoy', 7, -50, 2, '2023-07-16 13:00:00'), -- d

('MommyBoy', 8, 200, 3, '2023-07-22 15:15:00'); -- d

-- For user cutiepie

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('cutiepie', 9, 350, 2, '2023-08-05 08:45:00'), -- d

('cutiepie', 10, 400, 1, '2023-08-10 11:00:00'), -- d

('cutiepie', 1, 400, 3, '2023-08-16 13:15:00'), -- d

('cutiepie', 2, -100, 4, '2023-08-22 15:30:00'); -- d

-- For user Tourist

INSERT INTO public."CompetesIn"(

"contestantUsername", "roundNumber", score, rank, "timestampOfRegistration")

VALUES

('Tourist', 3, 1000, 1, '2023-09-05 09:00:00'), -- d

('Tourist', 4, 1000, 1, '2023-09-10 11:15:00'), -- d

('Tourist', 5, 350, 2, '2023-09-16 13:30:00'), -- d

('Tourist', 6, 400, 2, '2023-09-22 15:45:00'); -- d

Submission:

INSERT INTO public."Submission"(

"submissionID", "programmingLanguge", "sourceCode", verdict, "timestamp")

VALUES (1, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 1 A

(2, 'C++', 'Code', 'Accepted', '2023-01-14 09:33:00'), -- 2 B

(3, 'C++', 'Code', 'Accepted', '2023-01-14 09:36:00'), -- 7 A

(4, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 8 B

(5, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 13 A

(6, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 14 B

(7, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 15 C

(8, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 16 D

(9, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 25 A

(10, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 26 B

(11, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 27 C

(12, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 28 D

(13, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 31 A

(14, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 32 B

(15, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 33 C

(16, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 37 A

(17, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 38 B

(18, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 39 C

(19, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 40 D

(20, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 43 A

(21, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 44 B

(22, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 45 C

(23, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 49 A

(24, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 50 B

(25, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 51 C

(26, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 52 D

(27, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 55 A

(28, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 56 B

(29, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 57 C

(30, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 61 A

(31, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 62 B

(32, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 63 C

(33, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 1 A

(34, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 2 B

(35, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 5 E

(36, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 7 A

(37, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 12 F

(38, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 13 A

(39, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 14 B

(40, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 15 C

(41, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 25 A

(42, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 26 B

(43, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 27 C

(44, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 28 D

(45, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 31 A

(46, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 32 B

(47, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 37 A

(48, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 38 B

(49, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 43 A

(50, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 44 B

(51, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 45 C

(52, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 49 A

(53, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 52 D

(54, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 57 C

(55, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 61 A

(56, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 62 B

(57, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 63 C

(58, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 1 A

(59, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 7 A

(60, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 8 B

(61, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 13 A

(62, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 14 B

(63, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 25 A

(64, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 26 B

(65, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 27 C

(66, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 28 D

(67, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 31 A

(68, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 32 B

(69, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 37 A

(70, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 38 B

(71, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 43 A

(72, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 49 A

(73, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 50 B

(74, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 51 C

(75, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 55 A

(76, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 61 A

(77, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 62 B

(78, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 1 A

(79, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 2 B

(80, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 3 C

(81, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 7 A

(82, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 8 B

(83, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 9 C

(84, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 13 A

(85, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 14 B

(86, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 15 C

(87, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 17 E

(88, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 18 F

(89, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 25 A

(90, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 26 B

(91, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 27 C

(92, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 28 D

(93, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 29 E

(94, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 31 A

(95, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 32 B

(96, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 33 C

(97, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 34 D

(98, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'), -- 35 E

(99, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 37 A

(100, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 38 B

(101, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 39 C

(102, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 40 D

(103, 'C++', 'Code', 'Accepted', '2023-01-14 09:30:00'), -- 41 E

(104, 'C++', 'Code', 'Wrong Answer', '2023-01-14 09:30:00'); -- 42 F

Submits:

INSERT INTO public."Submits"(

username, "submissionID", "problemNumber")

VALUES ('Khaled47', 1, 1),

('Khaled47', 2, 2),

('Khaled47', 3, 7),

('Khaled47', 4, 8),

('Khaled47', 5, 13),

('Khaled47', 6, 14),

('Khaled47', 7, 15),

('Khaled47', 8, 16),

('Khaled47', 9, 25),

('Khaled47', 10, 26),

('Khaled47', 11, 27),

('Khaled47', 12, 28),

('AliFarhat', 13, 31),

('AliFarhat', 14, 32),

('AliFarhat', 15, 33),

('AliFarhat', 16, 37),

('AliFarhat', 17, 38),

('AliFarhat', 18, 39),

('AliFarhat', 19, 40),

('AliFarhat', 20, 43),

('AliFarhat', 21, 44),

('AliFarhat', 22, 45),

('AliFarhat', 23, 49),

('AliFarhat', 24, 50),

('AliFarhat', 25, 51),

('AliFarhat', 26, 52),

('ay\_charaf', 27, 55),

('ay\_charaf', 28, 56),

('ay\_charaf', 29, 57),

('ay\_charaf', 30, 61),

('ay\_charaf', 31, 62),

('ay\_charaf', 32, 63),

('ay\_charaf', 33, 1),

('ay\_charaf', 34, 2),

('ay\_charaf', 35, 5),

('ay\_charaf', 36, 7),

('ay\_charaf', 37, 12),

('Danosour', 38, 13),

('Danosour', 39, 14),

('Danosour', 40, 15),

('Danosour', 41, 25),

('Danosour', 42, 26),

('Danosour', 43, 27),

('Danosour', 44, 28),

('Danosour', 45, 31),

('Danosour', 46, 32),

('Danosour', 47, 37),

('Danosour', 48, 38),

('HashTrash', 49, 43),

('HashTrash', 50, 44),

('HashTrash', 51, 45),

('HashTrash', 52, 49),

('HashTrash', 53, 52),

('HashTrash', 54, 57),

('HashTrash', 55, 61),

('HashTrash', 56, 62),

('HashTrash', 57, 63),

('TheJoker', 58, 1),

('TheJoker', 59, 7),

('TheJoker', 60, 8),

('TheJoker', 61, 13),

('TheJoker', 62, 14),

('TheJoker', 63, 25),

('TheJoker', 64, 26),

('TheJoker', 65, 27),

('TheJoker', 66, 28),

('MommyBoy', 67, 31),

('MommyBoy', 68, 32),

('MommyBoy', 69, 37),

('MommyBoy', 70, 38),

('MommyBoy', 71, 43),

('MommyBoy', 72, 49),

('MommyBoy', 73, 50),

('MommyBoy', 74, 51),

('cutiepie', 75, 55),

('cutiepie', 76, 61),

('cutiepie', 77, 62),

('cutiepie', 78, 1),

('cutiepie', 79, 2),

('cutiepie', 80, 3),

('cutiepie', 81, 7),

('cutiepie', 82, 8),

('cutiepie', 83, 9),

('Tourist', 84, 13),

('Tourist', 85, 14),

('Tourist', 86, 15),

('Tourist', 87, 17),

('Tourist', 88, 18),

('Tourist', 89, 25),

('Tourist', 90, 26),

('Tourist', 91, 27),

('Tourist', 92, 28),

('Tourist', 93, 29),

('Tourist', 94, 31),

('Tourist', 95, 32),

('Tourist', 96, 33),

('Tourist', 97, 34),

('Tourist', 98, 35),

('Tourist', 99, 37),

('Tourist', 100, 38),

('Tourist', 101, 39),

('Tourist', 102, 40),

('Tourist', 103, 41),

('Tourist', 104, 42);

Messages:

INSERT INTO public."Messages"(

"senderUsername", "receiverUsername", content, "timestamp")

VALUES

('AliFarhat', 'ay\_charaf', 'Hey, lets see who can solve problem A first!', '2023-11-04 10:15:00'),

('ay\_charaf', 'AliFarhat', 'Challenge accepted! Im already working on it!', '2023-11-05 14:22:00'),

('Khaled47', 'Danosour', 'Im the coding king! You dont stand a chance!', '2023-11-06 18:30:00'),

('Danosour', 'Khaled47', 'Well see about that. Prepare for the coding battle!', '2023-11-08 09:45:00'),

('HashTrash', 'TheJoker', 'You think youre a joker? Ill show you real coding humor!', '2023-11-10 16:50:00'),

('TheJoker', 'HashTrash', 'I always have a trick up my sleeve! Get ready!', '2023-11-12 22:35:00'),

('MommyBoy', 'cutiepie', 'Coding is childs play for me. Watch and learn!', '2023-11-15 11:20:00'),

('cutiepie', 'MommyBoy', 'A piece of cake, huh? Lets code our way through this!', '2023-11-17 13:45:00'),

('Tourist', 'Khaled47', 'Khaled47, youre just a tourist here. I rule!', '2023-11-19 17:25:00'),

('Khaled47', 'Tourist', 'Tourist, you may be the king, but Im the challenge!', '2023-11-21 08:30:00'),

('AliFarhat', 'ay\_charaf', 'Ive solved problem A. It was easy!', '2023-11-23 09:55:00'),

('ay\_charaf', 'AliFarhat', 'Well done! Now, lets tackle problem B!', '2023-11-26 14:40:00'),

('Danosour', 'HashTrash', 'The real showdown begins! Lets solve problem A.', '2023-11-28 15:05:00'),

('HashTrash', 'Danosour', 'Danosour, Im ready to take the challenge!', '2023-11-30 19:30:00'),

('TheJoker', 'MommyBoy', 'Ready for a coding joke? Here comes problem A.', '2023-12-01 22:55:00'),

('MommyBoy', 'TheJoker', 'I can handle your jokes! Lets tackle this challenge.', '2023-12-03 09:10:00'),

('cutiepie', 'Tourist', 'Tourist, Im right behind you. Problem A awaits!', '2023-12-05 12:20:00'),

('Tourist', 'cutiepie', 'Cutiepie, its a race! Lets solve it first.', '2023-12-07 15:40:00'),

('AliFarhat', 'Khaled47', 'Khaled47, you thought you could beat me? Problem A is done!', '2023-12-09 18:55:00'),

('Khaled47', 'AliFarhat', 'AliFarhat, Im always up for a challenge. Now, problem B awaits!', '2023-12-11 22:00:00');

Announcement:

INSERT INTO public."Announcement"(

"announcementID", "username", "language", "content", "timestamp")

VALUES

(1, 'HussKiller', 'English', 'Get ready for HusseinBakri Championship!', '2023-11-01 08:00:00'),

(2, 'HussKiller', 'English', 'Announcing AUB Showdown - Prepare for a coding battle!', '2023-11-03 10:30:00'),

(3, 'HussKiller', 'English', 'LAU Titans Challenge is coming up. Dont miss it!', '2023-11-05 12:15:00'),

(4, 'HussKiller', 'English', 'Its time for CodeCrafters Duel! Join the competition!', '2023-11-07 14:40:00'),

(5, 'HussKiller', 'English', 'Announcement: AlgorithmShip - Challenge your coding skills!', '2023-11-09 17:00:00'),

(6, 'ImanDestroyer', 'English', 'Rookie Coding Challenge is around the corner. Be ready!', '2023-11-11 19:25:00'),

(7, 'ImanDestroyer', 'English', 'Join CodeFusion Challenge and prove your coding expertise!', '2023-11-13 21:45:00'),

(8, 'ImanDestroyer', 'English', 'Get set for Algorithm Explorers - A coding competition like no other!', '2023-11-15 23:55:00'),

(9, 'ImanDestroyer', 'English', 'Announcing Dakowdas Enthusiasts - Show your coding skills!', '2023-11-17 08:10:00'),

(10, 'ImanDestroyer', 'English', 'Bakri Finale is coming soon. Get ready for the challenge!', '2023-11-19 10:30:00');

Users : (Admins)

INSERT INTO public."User"(

username, email, passwod, "firstName", "lastName", country, "registrationDat")

VALUES ('MK', 'MK@gmail.com','6600bf6331c98f1619619fc8e405d5ec3dd97e614b9d241f14275168f130c916', 'Mohamad', 'Kreidieh', 'Lebanon', current\_timestamp),

('TamTam', 'TamTam@gmail.com','6c3bf86f56a3e67d88f6d6bea7e03dd5ca3c31b07a51fc1ce28571e049dce5ea' ,'Tamara', 'Sadek', 'Lebanon', current\_timestamp),

('JohnSmith', 'JohnSmith@example.com', 'd19edeb7198f1e9ee2ef8d1876f49b3ec3c4c3fc126c6419734397741f048d318', 'John', 'Smith', 'USA', current\_timestamp),

('EmmaJohnson', 'EmmaJohnson@example.com', 'f01c86dbff968e105264b2dab1b71f0631a36047aa4b0f66a4b51b69cb68ff12', 'Emma', 'Johnson', 'USA', current\_timestamp),

('MichaelWilson', 'MichaelWilson@example.com', 'bfc47af6e83f5a3f98f3c5ec42022100b453bd5d6d17c79d05d7482d309558d7', 'Michael', 'Wilson', 'Canada', current\_timestamp),

('SophiaJones', 'SophiaJones@example.com', '8e11a7fbf6d88abf9c20758bb8c005855542f6c2edc283f8d8d5d9eb61f4c5e6', 'Sophia', 'Jones', 'Canada', current\_timestamp),

('DavidBrown', 'DavidBrown@example.com', '0bf10ed62bf8762c753ab8c1b9d9326b53c7cd0491deea8db3f3a6b675400bda', 'David', 'Brown', 'UK', current\_timestamp),

('OliviaDavis', 'OliviaDavis@example.com', 'e6ccf9d13c1167c3a6b61c0176473cd977640999ef974ea83a75be88918046ec', 'Olivia', 'Davis', 'UK', current\_timestamp),

('JamesMiller', 'JamesMiller@example.com', '2a0a3842a5046e301f3e4d24cfb2d5fb924932f6c3c72318d5f1cfa216e68549', 'James', 'Miller', 'Australia', current\_timestamp),

('SophiaHarris', 'SophiaHarris@example.com', 'cfdad2c74ab40a22fc08c67183ad02ce431826b26d69c650ff8c877ed489b950', 'Sophia', 'Harris', 'Australia', current\_timestamp);

Admin:

INSERT INTO public."Admin"(

username, role, "contributionScore")

VALUES ('MK', 'Super Admin', 5),

('TamTam', 'Moderator', 10),

('JohnSmith', 'Moderator', 15),

('EmmaJohnson', 'Super Admin', 8),

('MichaelWilson', 'Moderator', 12),

('SophiaJones', 'Moderator', 5),

('DavidBrown', 'Super Admin', 20),

('OliviaDavis', 'Moderator', 7),

('JamesMiller', 'Super Admin', 13),

('SophiaHarris', 'Moderator', 18);

Responsibilities:

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('JohnSmith', 'Manage User Accounts'),

('JohnSmith', 'Create Contests');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('EmmaJohnson', 'Review Contest Entries'),

('EmmaJohnson', 'Manage Announcements');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('MichaelWilson', 'Create Contests'),

('MichaelWilson', 'Monitor Chat Rooms');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('SophiaJones', 'Manage User Accounts'),

('SophiaJones', 'Manage Announcements');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('DavidBrown', 'Create Contests'),

('DavidBrown', 'Review Contest Entries');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('OliviaDavis', 'Monitor Chat Rooms'),

('OliviaDavis', 'Manage User Accounts');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('JamesMiller', 'Manage Announcements'),

('JamesMiller', 'Review Contest Entries');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('SophiaHarris', 'Create Contests'),

('SophiaHarris', 'Manage User Accounts');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('MK', 'Manage User Accounts'),

('MK', 'Create Contests'),

('MK', 'Moderate Forum');

INSERT INTO public."Responsibilities"("adminUsername", responsibility)

VALUES ('TamTam', 'Review Contest Entries'),

('TamTam', 'Monitor Chat Rooms'),

('TamTam', 'Manage Announcements');

BlogEntry , Comments, Hashtag, Hack, Hacks, Notifies