Department of Computer Science and Engineering

Course Code: CSE404

Implement a Basic Knowledgebase using Prolog

Title: Knowledgebase on World Wars using Prolog

Submitted By:

Name: Mahabub Uddin Azmi

ID: 22101180

Section: D

Submitted to:

Bidita Sarkar Diba

Department of Computer Science and Engineering, University of Asia Pacific

Date : 11/08/25

Problem Description:

This project aims to develop a knowledgebase in Prolog that stores and retrieves detailed information about significant battles from both World War I and World War II. The knowledgebase contains facts about wars, battles, participating countries, troop numbers, and alliances.
Using Prolog's logic programming capabilities, the system allows users to perform complex queries such as finding all battles in a given war, listing countries involved in a specific battle, retrieving wars fought by a particular country, and checking troop strengths.
Recursion and rule-based querying are implemented to allow flexible and intelligent information retrieval.

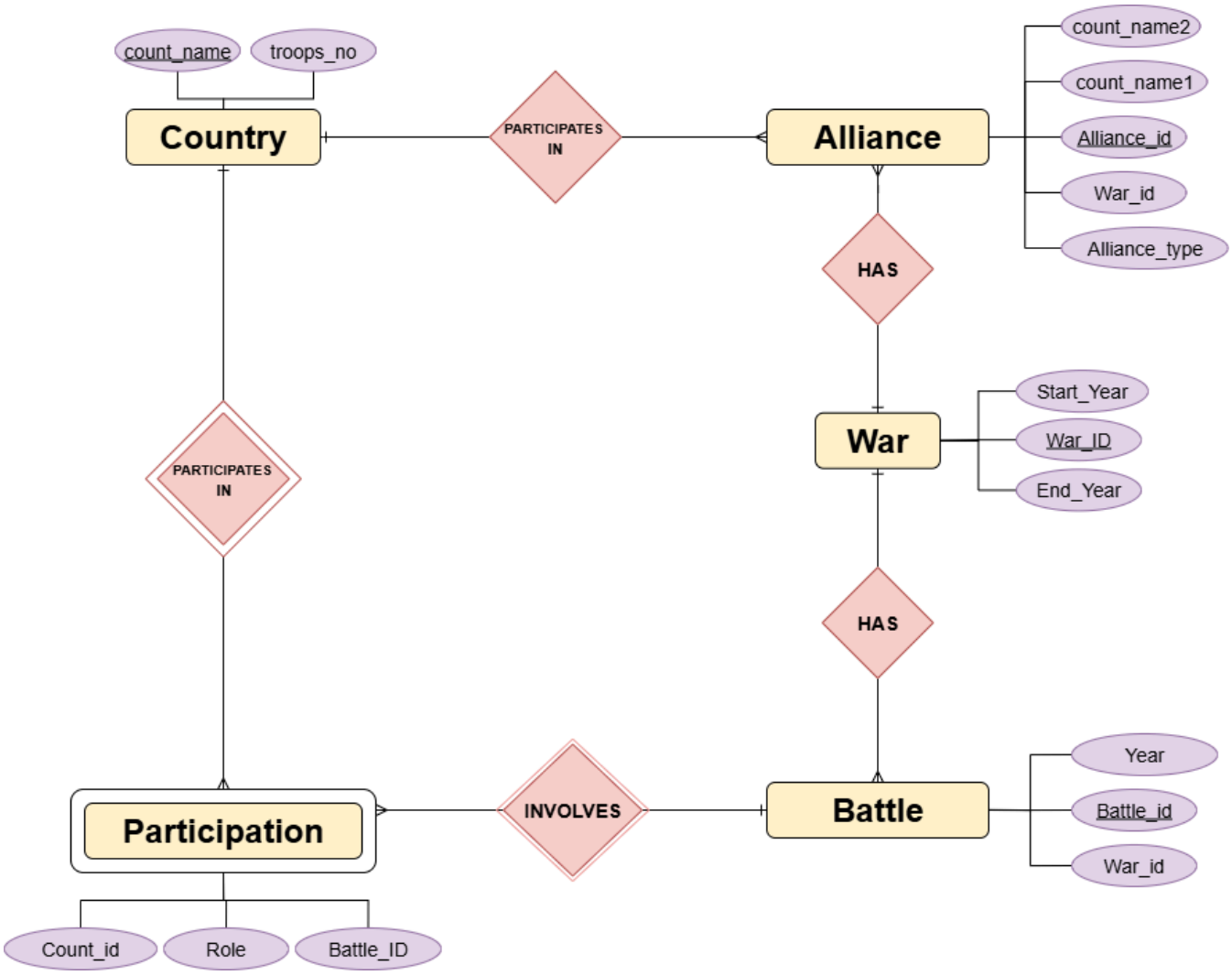
Tools and Languages Used:

Programming Language: Prolog 

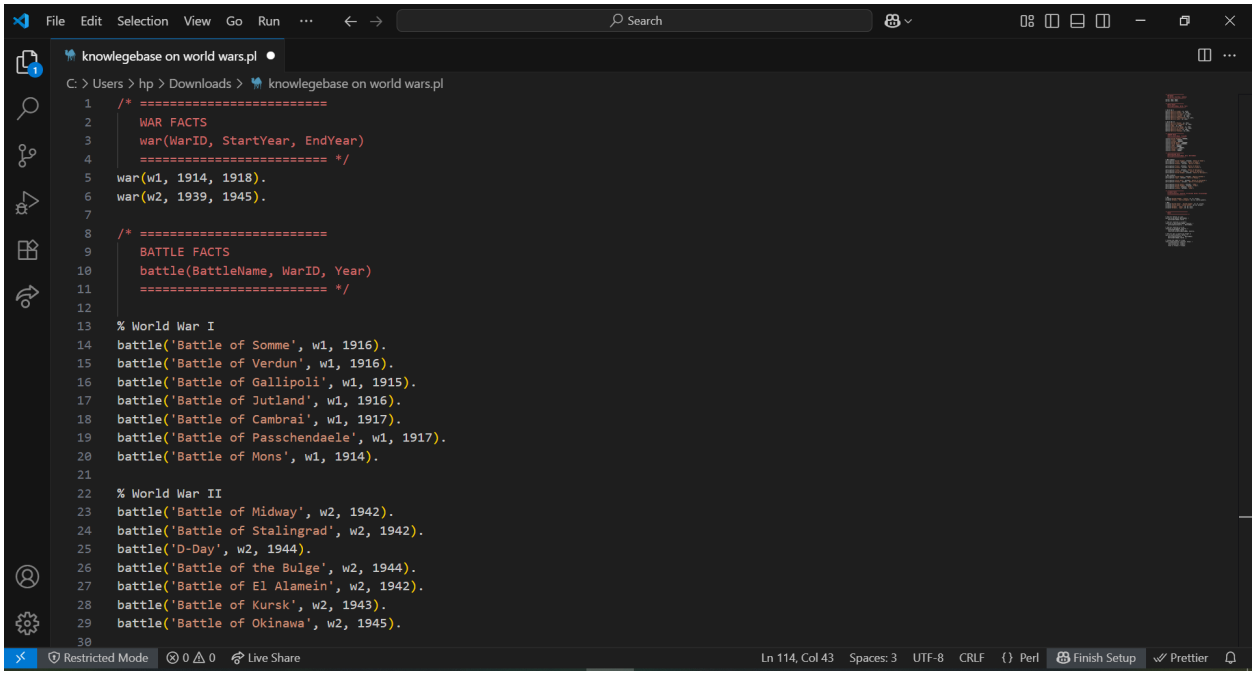
Tool: SWI-Prolog IDE and VS code for writing and testing the knowledgebase 

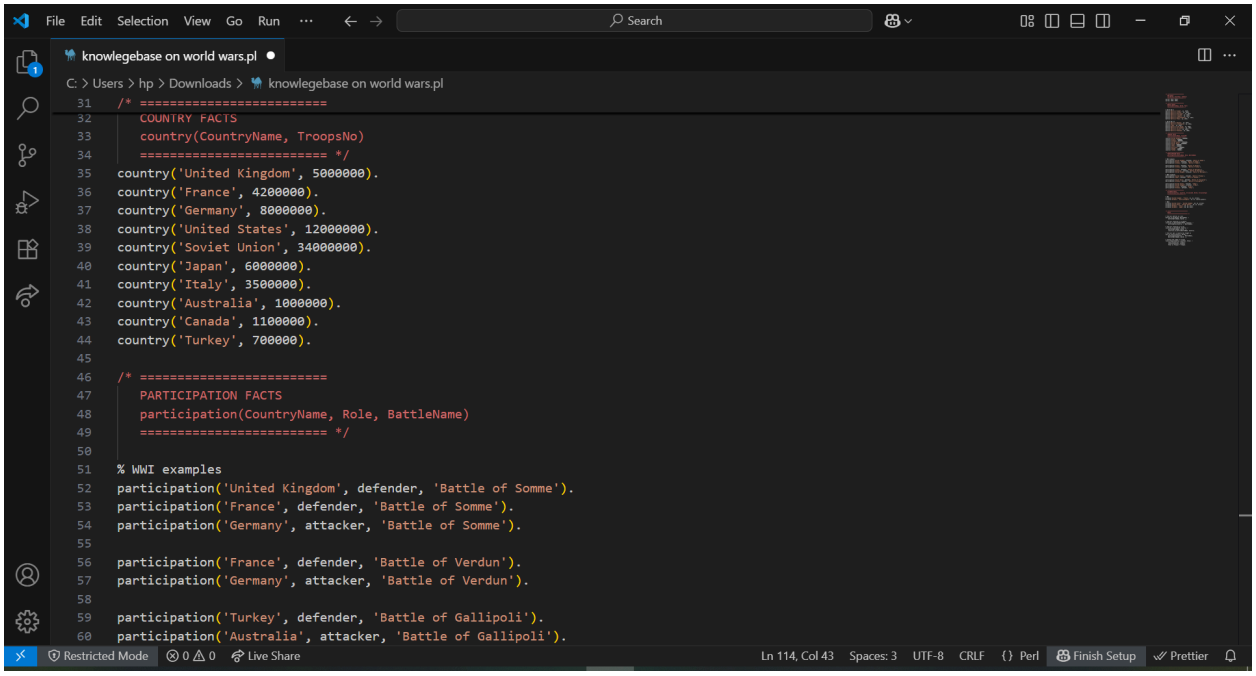
Diagram Tool: draw.io for ER diagram creation 

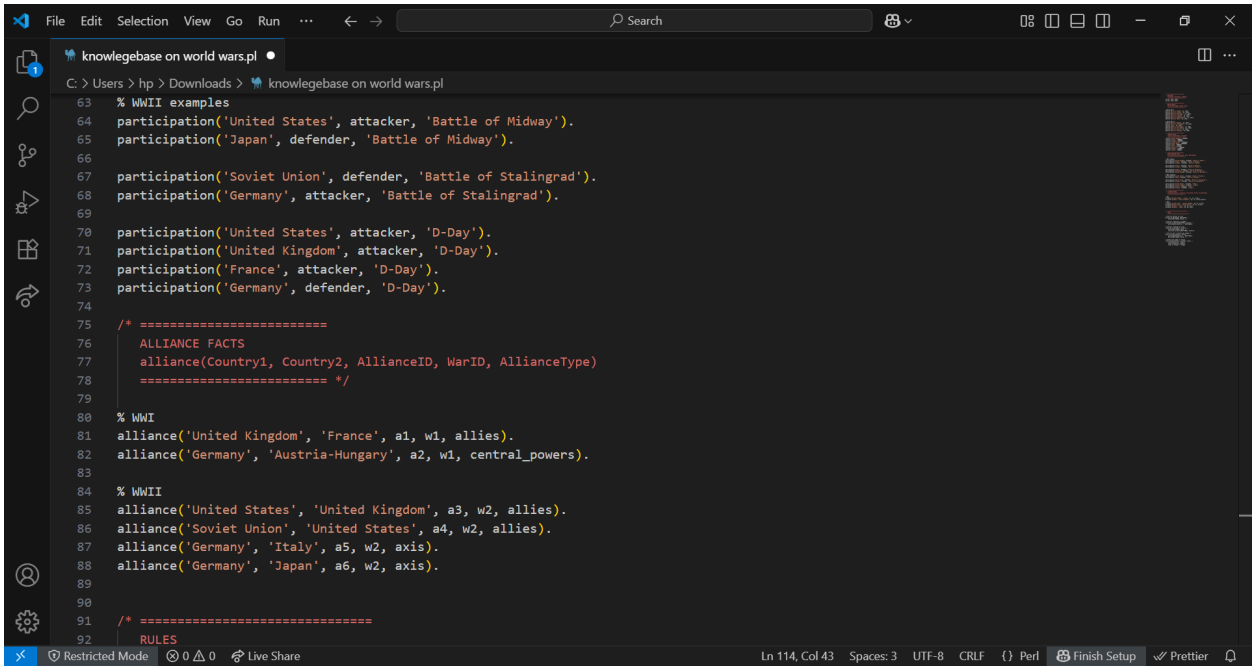
Diagram/Figure:



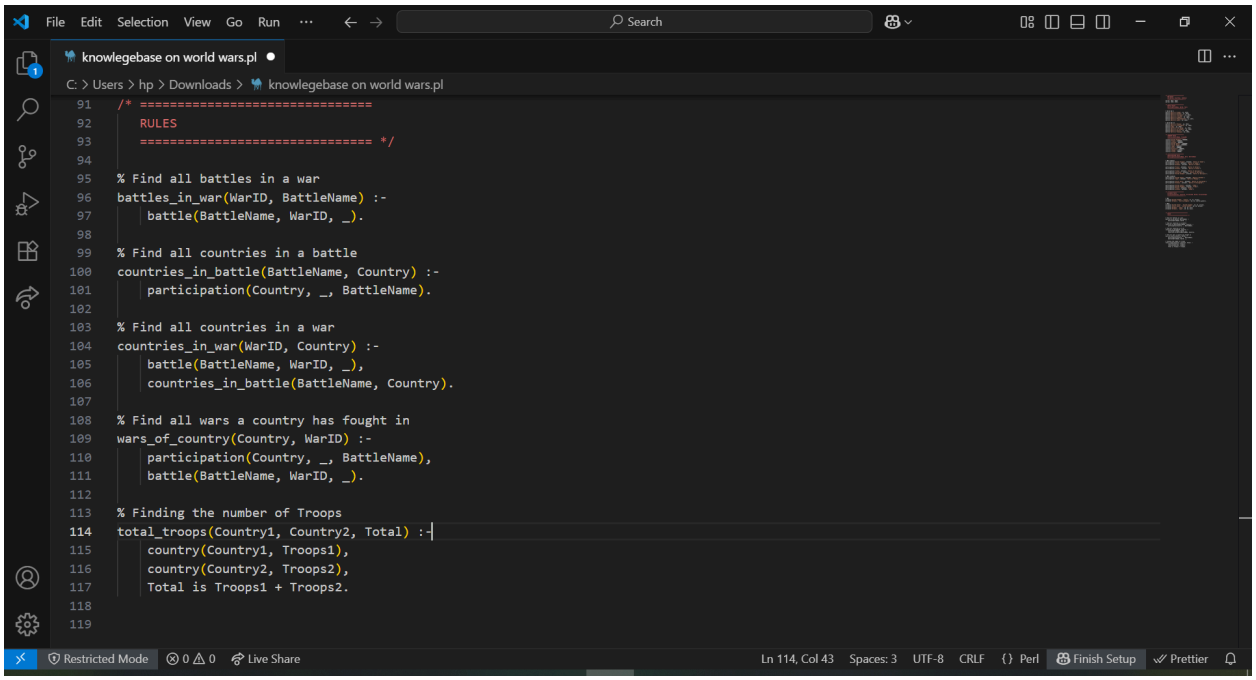
Sample Input/output:



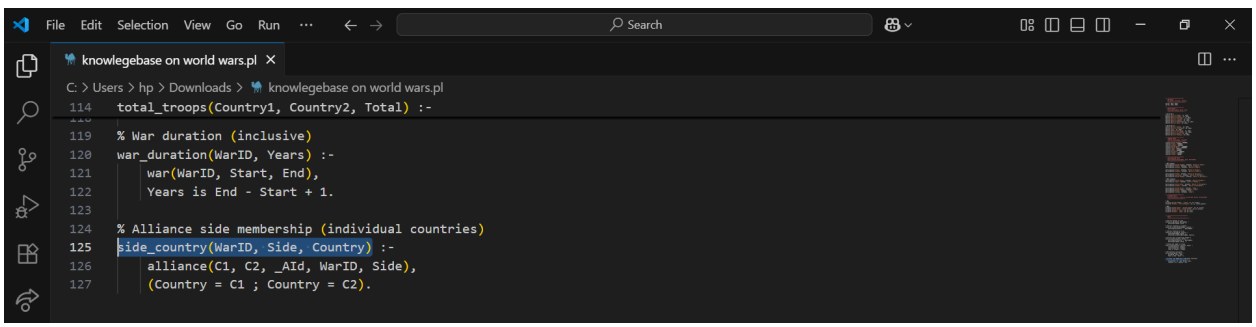




```
% WWII examples
participation('United States', attacker, 'Battle of Midway').
participation('Japan', defender, 'Battle of Midway').

participation('Soviet Union', defender, 'Battle of Stalingrad').
participation('Germany', attacker, 'Battle of Stalingrad').

participation('United States', attacker, 'D-Day').
participation('United Kingdom', attacker, 'D-Day').
participation('France', attacker, 'D-Day').
participation('Germany', defender, 'D-Day').

/* ========================
   ALLIANCE FACTS
   alliance(Country1, Country2, AllianceID, WarID, AllianceType)
   ======================== */

% WWI
alliance('United Kingdom', 'France', a1, w1, allies).
alliance('Germany', 'Austria-Hungary', a2, w1, central_powers).

% WWII
alliance('United States', 'United Kingdom', a3, w2, allies).
alliance('Soviet Union', 'United States', a4, w2, allies).
alliance('Germany', 'Italy', a5, w2, axis).
alliance('Germany', 'Japan', a6, w2, axis).


/* ==============================
   RULES
```

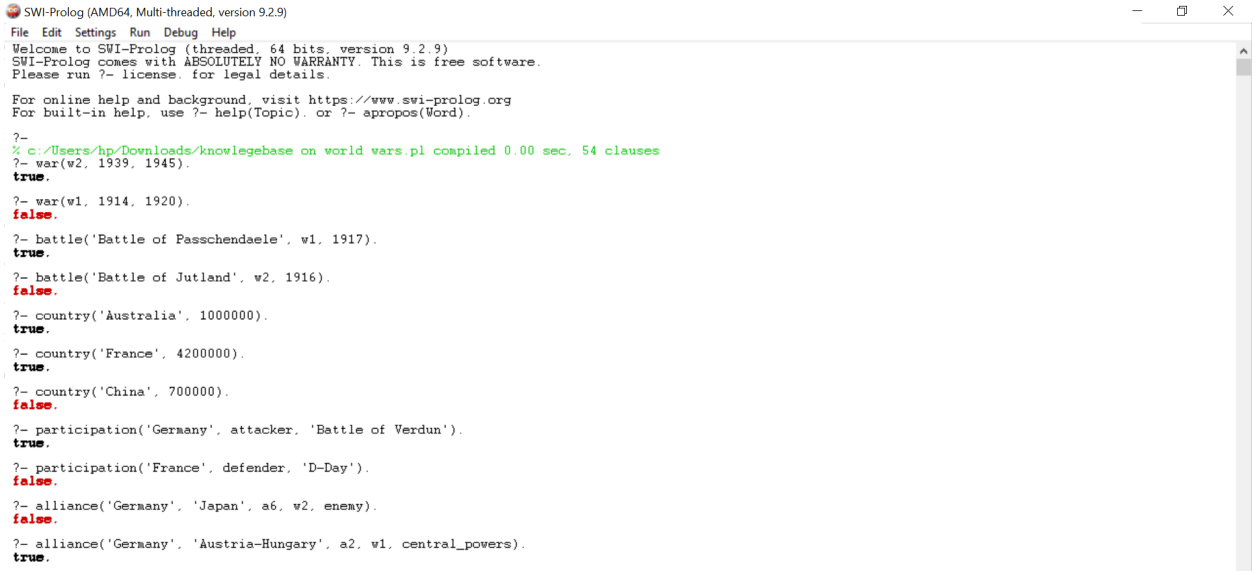


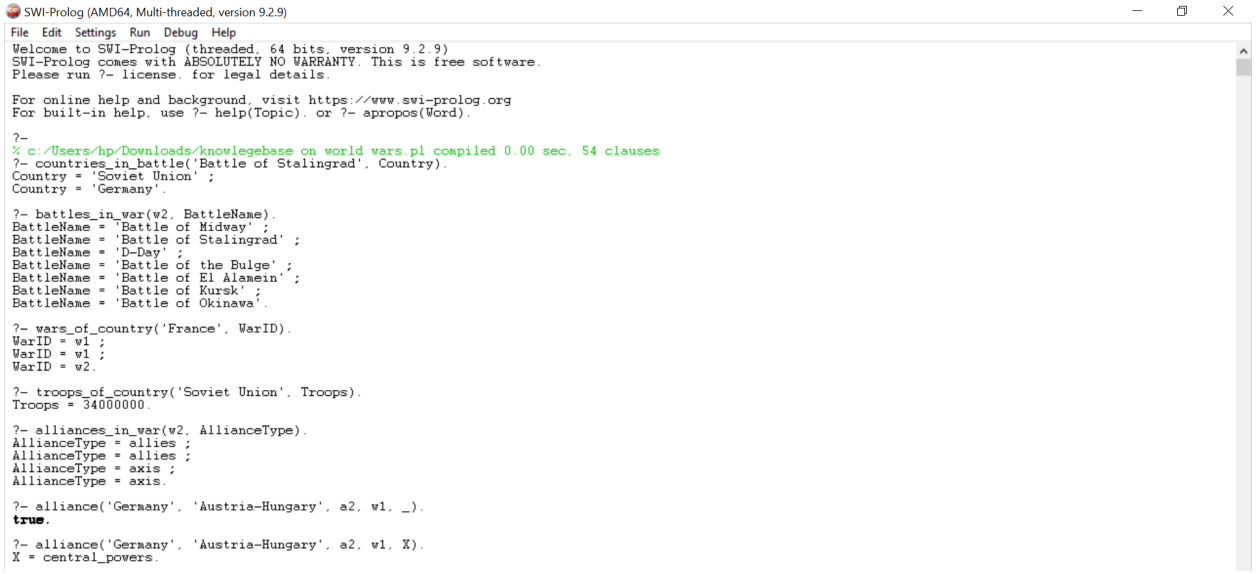
```
/* ==============================
   RULES
   ============================== */

% Find all battles in a war
battles_in_war(WarID, BattleName) :-
    battle(BattleName, WarID, _).

% Find all countries in a battle
countries_in_battle(BattleName, Country) :-
    participation(Country, _, BattleName).

% Find all countries in a war
countries_in_war(WarID, Country) :-
    battle(BattleName, WarID, _),
    countries_in_battle(BattleName, Country).

% Find all wars a country has fought in
wars_of_country(Country, WarID) :-
    participation(Country, _, BattleName),
    battle(BattleName, WarID, _).

% Finding the number of Troops
total_troops(Country1, Country2, Total) :-
    country(Country1, Troops1),
    country(Country2, Troops2),
    Total is Troops1 + Troops2.
```

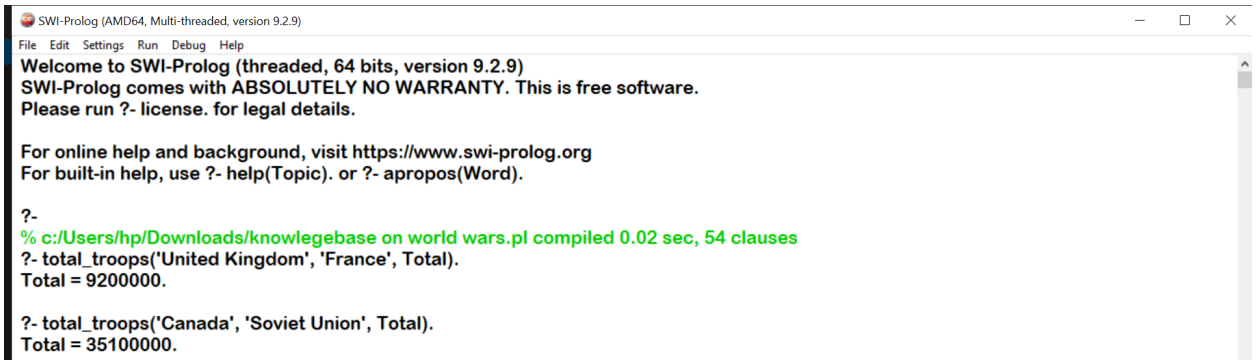


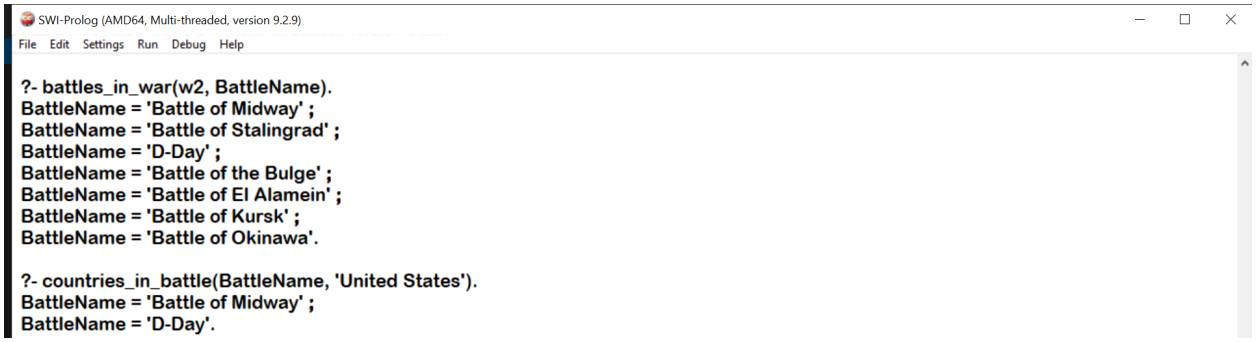
```
total_troops(Country1, Country2, Total) :-

% War duration (inclusive)
war_duration(WarID, Years) :-
    war(WarID, Start, End),
    Years is End - Start + 1.

% Alliance side membership (individual countries)
side_country(WarID, Side, Country) :-
    alliance(C1, C2, _AId, WarID, Side),
    (Country = C1 ; Country = C2).
```

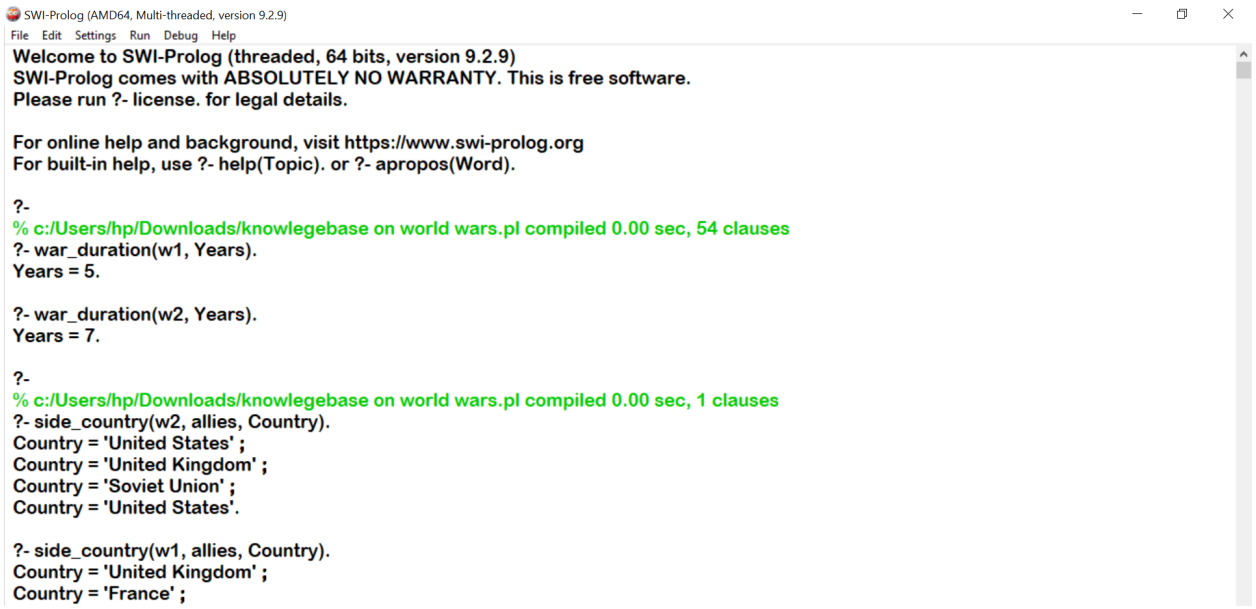
Output:


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?-
% c:/Users/hp/Downloads/knowlegebase on world wars.pl compiled 0.00 sec, 54 clauses
?- war(w2, 1939, 1945).
true.
?- war(w1, 1914, 1920).
false.
?- battle('Battle of Passchendaele', w1, 1917).
true.
?- battle('Battle of Jutland', w2, 1916).
false.
?- country('Australia', 1000000).
true.
?- country('France', 4200000).
true.
?- country('China', 700000).
false.
?- participation('Germany', attacker, 'Battle of Verdun').
true.
?- participation('France', defender, 'D-Day').
false.
?- alliance('Germany', 'Japan', a6, w2, enemy).
false.
?- alliance('Germany', 'Austria-Hungary', a2, w1, central_powers).
true.


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?-
% c:/Users/hp/Downloads/knowlegebase on world wars.pl compiled 0.00 sec, 54 clauses
?- countries_in_battle('Battle of Stalingrad', Country).
Country = 'Soviet Union' ;
Country = 'Germany'.
?- battles_in_war(w2, BattleName).
BattleName = 'Battle of Midway' ;
BattleName = 'Battle of Stalingrad' ;
BattleName = 'D-Day' ;
BattleName = 'Battle of the Bulge' ;
BattleName = 'Battle of El Alamein' ;
BattleName = 'Battle of Kursk' ;
BattleName = 'Battle of Okinawa'.
?- wars_of_country('France', WarID).
WarID = w1 ;
WarID = w1 ;
WarID = w2.
?- troops_of_country('Soviet Union', Troops).
Troops = 34000000.
?- alliances_in_war(w2, AllianceType).
AllianceType = allies ;
AllianceType = allies ;
AllianceType = axis ;
AllianceType = axis.
?- alliance('Germany', 'Austria-Hungary', a2, w1, _).
true.
?- alliance('Germany', 'Austria-Hungary', a2, w1, X).
X = central_powers.


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?-
% c:/Users/hp/Downloads/knowlegebase on world wars.pl compiled 0.02 sec, 54 clauses
?- total_troops('United Kingdom', 'France', Total).
Total = 9200000.
?- total_troops('Canada', 'Soviet Union', Total).
Total = 35100000.


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
?- battles_in_war(w2, BattleName).
BattleName = 'Battle of Midway' ;
BattleName = 'Battle of Stalingrad' ;
BattleName = 'D-Day' ;
BattleName = 'Battle of the Bulge' ;
BattleName = 'Battle of El Alamein' ;
BattleName = 'Battle of Kursk' ;
BattleName = 'Battle of Okinawa'.
?- countries_in_battle(BattleName, 'United States').
BattleName = 'Battle of Midway' ;
BattleName = 'D-Day'.


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?-
% c:/Users/hp/Downloads/knowlegebase on world wars.pl compiled 0.00 sec, 54 clauses
?- war_duration(w1, Years).
Years = 5.
?- war_duration(w2, Years).
Years = 7.
?-
% c:/Users/hp/Downloads/knowlegebase on world wars.pl compiled 0.00 sec, 1 clauses
?- side_country(w2, allies, Country).
Country = 'United States' ;
Country = 'United Kingdom' ;
Country = 'Soviet Union' ;
Country = 'United States'.
?- side_country(w1, allies, Country).
Country = 'United Kingdom' ;
Country = 'France' ;

Conclusion and Challenges:

The project successfully demonstrates the use of Prolog for building a structured and queryable knowledgebase. By modeling World War I and World War II battles, the system showcases Prolog's strength in handling relationships and recursive queries for knowledge representation. The final implementation allows efficient retrieval of historical war data based on user queries.

1. Designing a clear ER model to represent wars, battles, countries, and their relationships without redundancy.

2. Ensuring consistency in data entry while managing multiple attributes like troop numbers, war IDs, and alliances.

3. Implementing rules with recursion to allow flexible queries without making the code overly complex.

4. Deciding the granularity of data — whether to focus only on major battles or include smaller conflicts as well.