**Assignment 8: Building a Family Tree in Prolog**

Student Name

University

Advanced Programming Languages - Full term

Professor Name

August 11, 2025

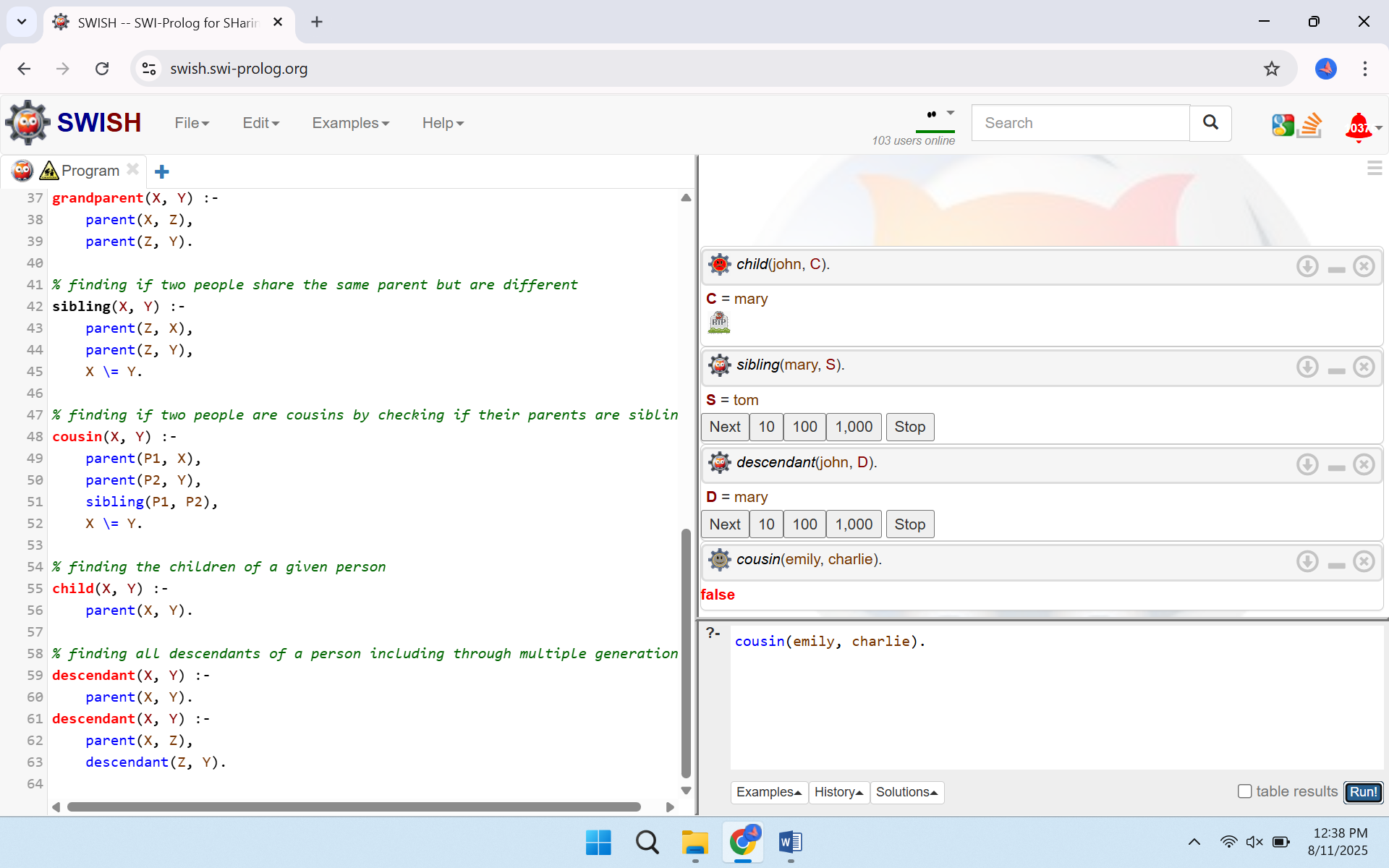
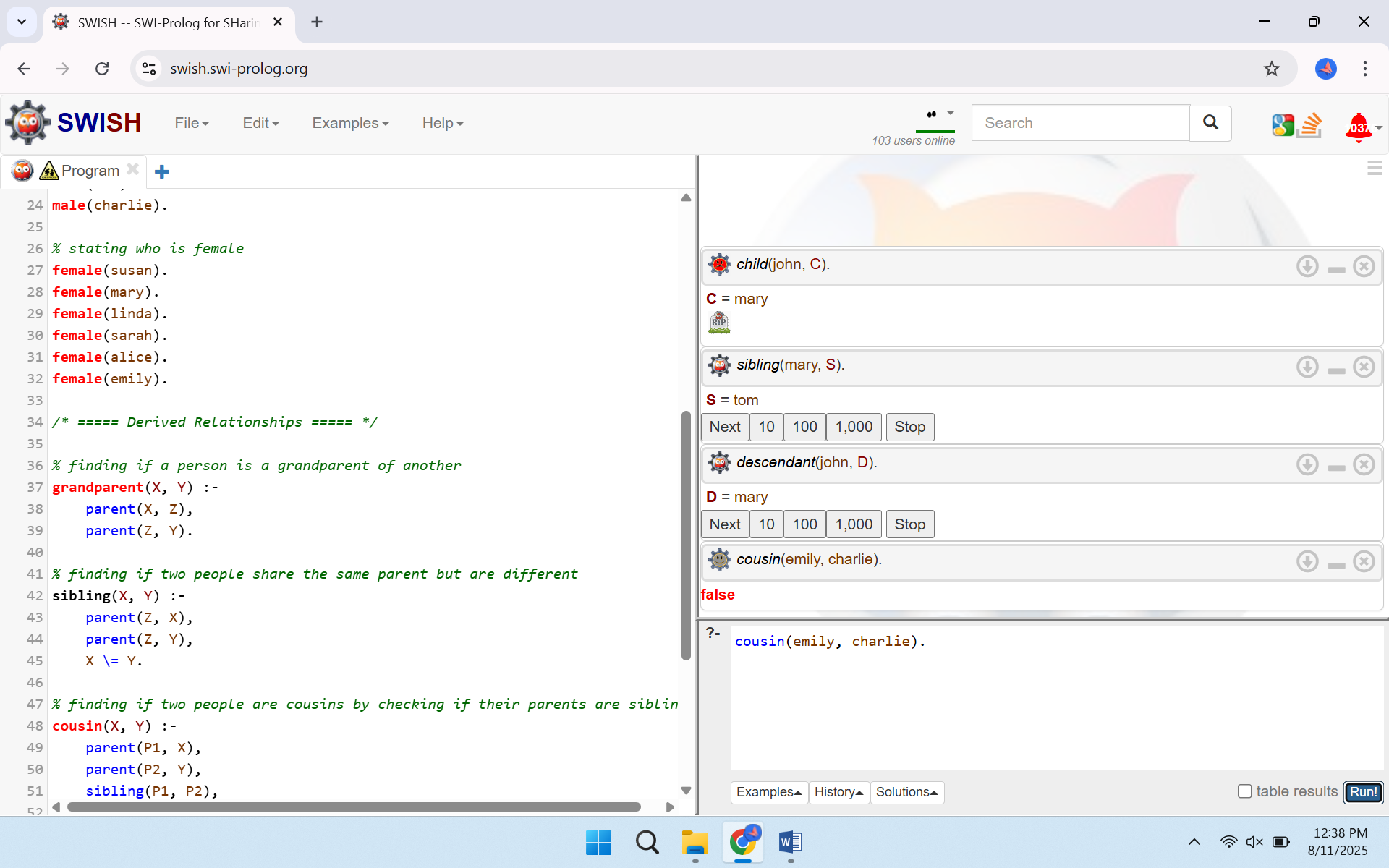
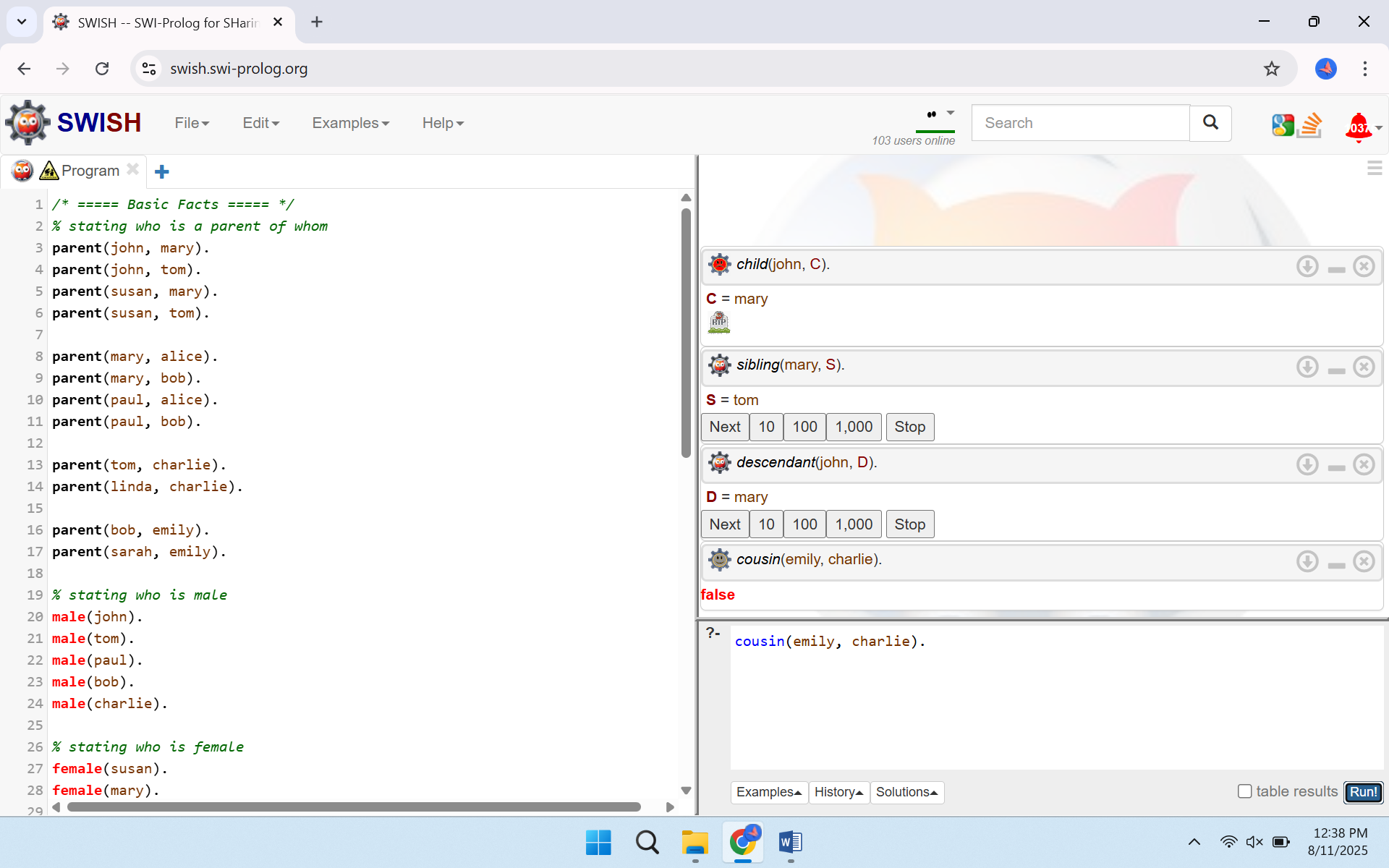
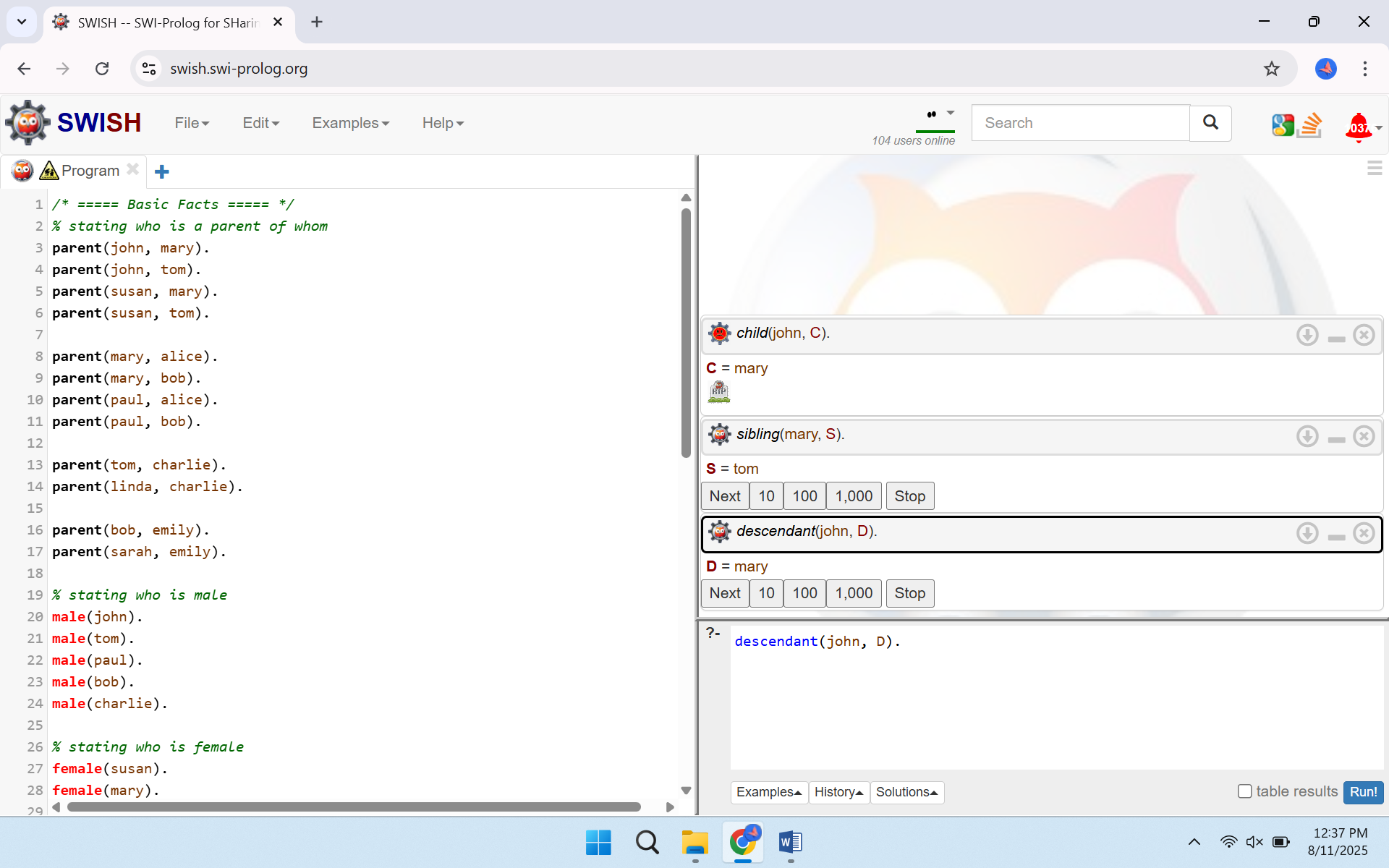
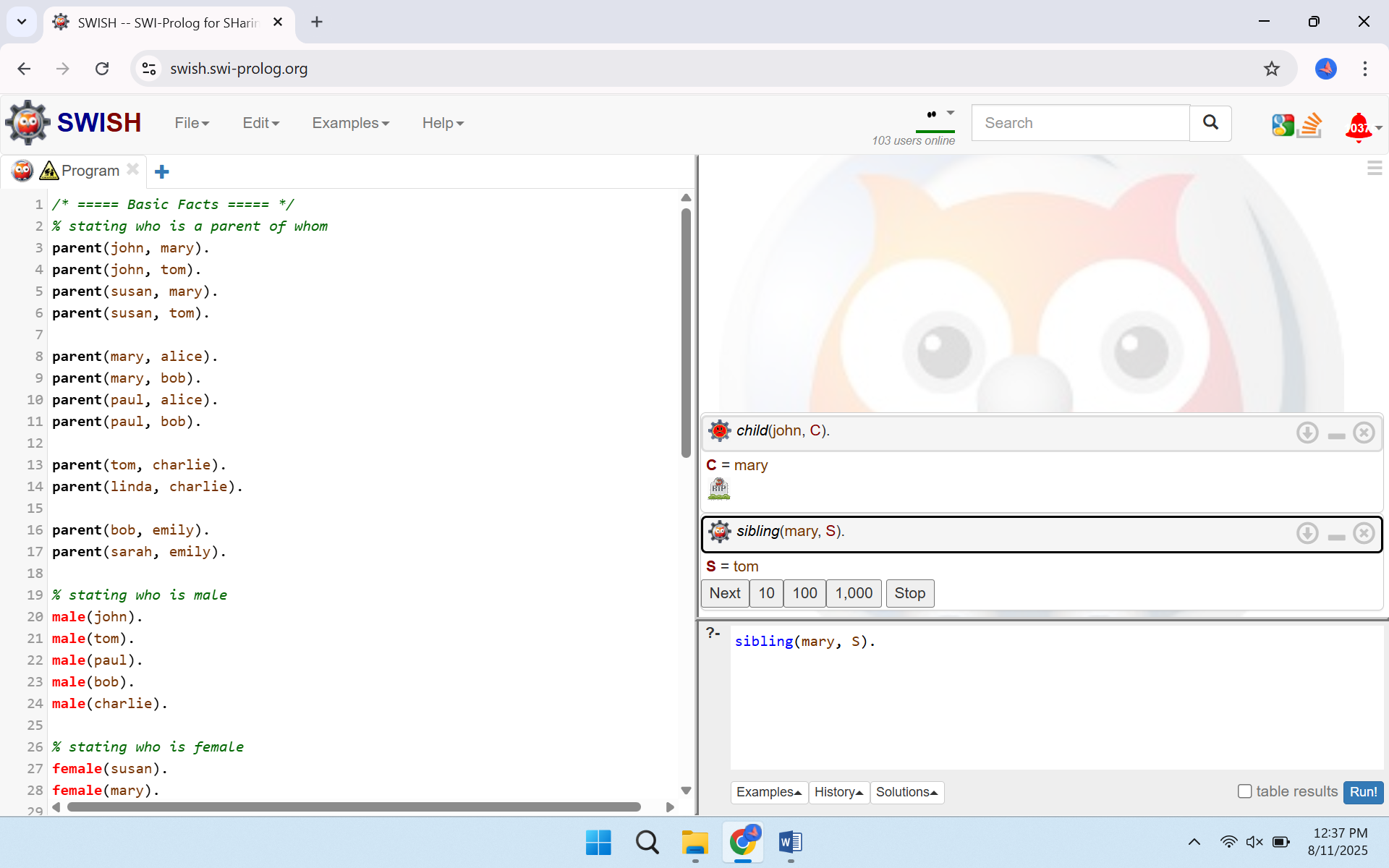
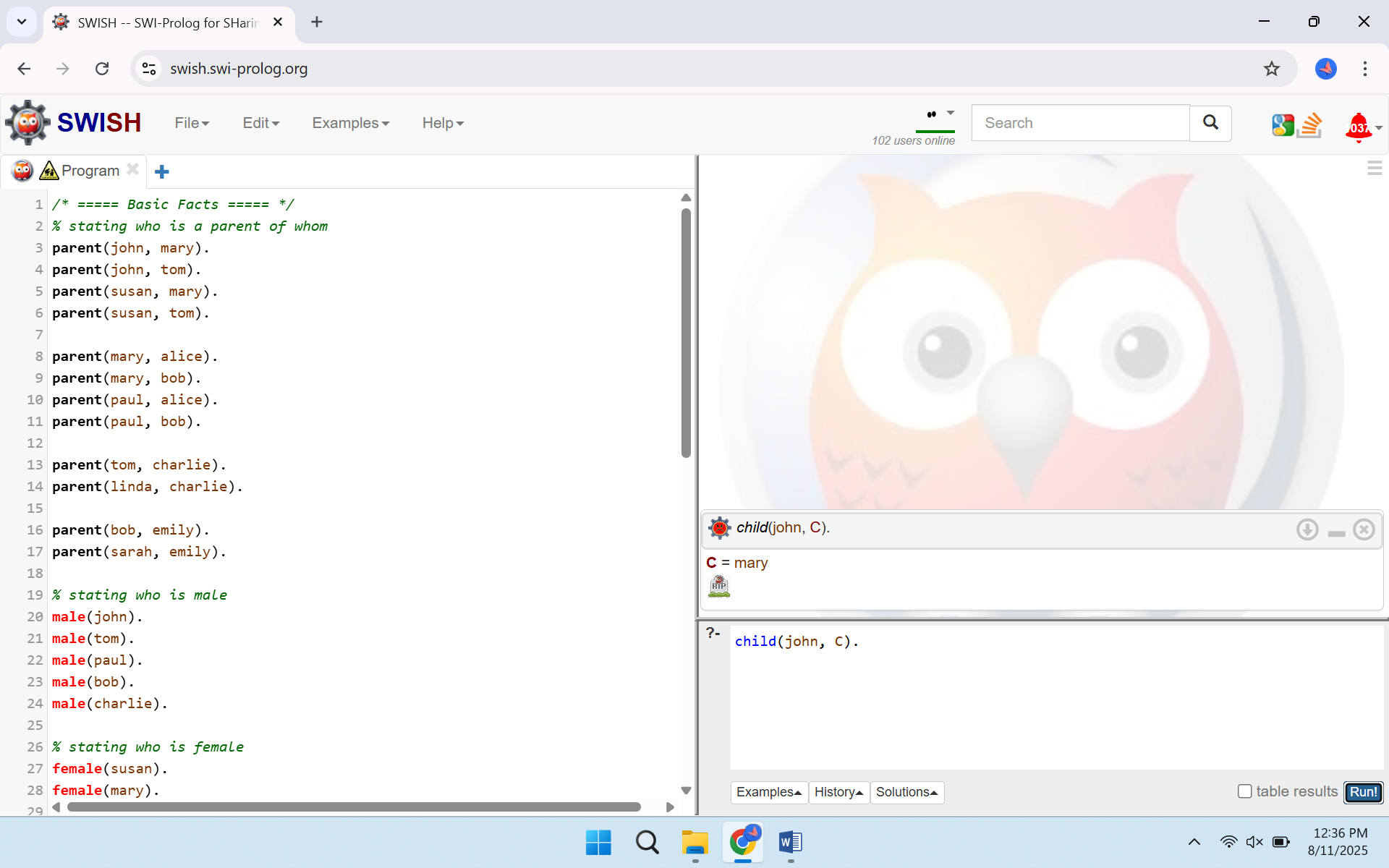
**Creating a Family Tree in Prolog  
Overview**

The project presented a basic family tree implemented in Prolog, and it was developed in the SWISH online environment. Its foremost objective was to express family relations at the lowest level and to infer higher orders of relationships using the process of logical inference. The use of declarative facts and inference rules to respond to queries about kinship was highlighted in the exercise, revealing how a logic-based language can take the form of hierarchical relationships across generations.

**Implementation process**

Beginning with objective statements, the program documented relationships between parents and the gender assignments of each person. These founding statements were the basis of inference. Derived relations were established in the following way. Where a grandparent relation was interpreted, an individual was interpreted as a parent of someone, and that individual was a parent to yet another individual. The relation of the siblings was assumed whenever two different individuals had at least one parent in common. When the fathers of two people were brothers, the parenthood was also known as a cousin. The recursion on the descendant relation had a base case that just identified a direct parent-child relationship, and a recursive case that recurred across intergenerational levels to identify more indirect descendants. Various substitutions were done with pattern matching and variable binding, as well as the ability to generate all legitimate responses to a particular query.

**Identification of Family Tree Implementation**

The implementation of the family tree starts with creating some initial information that determines the relationship between the parent and a child, and also the gender of a person. Such facts are the sources of the first data, on the basis of which other relations may be discussed logically. More rules are then provided to recognize extended relationships. The grandparent rule is used to define that a single individual is a parent of another who is also a parent. In the sibling rule, a check is made on two different persons, at least one parent in common. The cousin rule entails the connection between two people whose fathers and mothers are brothers/sisters. A child rule merely reverses any person who is registered as a direct ancestor of someone. The descendant rule is applied recursively, and it begins with direct parent-child relationships and then moves step-by-step to second, third, and subsequent levels to cover the broader descendants. The combination of these facts and rules permits queries to retrieve information on direct and indirect familial relationships, which shows off the recursion and logical inference in relation modeling.

**Description of Query Results**

Stringent query implementation proves the retrieval of designated family relations in the stipulated knowledge base. The first question names Mary, the child of John. The second question asks if Tom is a brother or a sister of Mary. The third question enumerates all of the descendants of John, beginning with Mary. The last and fourth query of verifying the relationship of Emily and Charlie being cousins returns false, which implies that they are not cousins, according to the facts provided.

**Inquiries and experimentation**

In SWISH, a group of representative questions was issued, and the responses were recorded to test the program. Possible queries ran the gamut of requesting the children of a named individual, the list of siblings of a named individual, asking whether two individuals were cousins, and retrieving all of the descendants of a named individual within several generations. Test runs generated some returned solutions where possible, representing nondeterministic search and backtracking. The share option was used to create a lasting link to the state of the program to submit results.

**Challenges**

To make sure that the recursive descendant relation was properly terminated, base and recursive cases had to be considered with care to prevent an infinite exploration. Self-relations had to be effectively prevented between siblings and cousins by express checks to ensure that two identical persons were not considered related to each other. Cases in which many parent facts would result in identical answers had to be met with a knowledge of how the inference engine would list solutions and how to make sense of repeated outputs during result gathering.

**Conclusion**

The Prolog approach offered an unambiguous and concise means of representation of family ties as well as query execution, where the family-related entities were executed as a result of logical inference and recursion. This exercise strengthened the study of declarative programming, the feasibility of recursion to transitive relations, and the significance of precise conditions of rules in order to uphold accurate and significant results of queries.