**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение**

высшего образования

**«Севастопольский государственный университет»**

**Кафедра ИС**

## РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА

**по дисциплине «Методы и системы искусственного интеллекта»**

**вариант № 3**

**Выполнил: ст. гр. ИС/б-20-1-о**

**Галенин А.К.**

**Проверил: Сметанина Т.И.**

**Забаштанский А.К.**

**Севастополь**

**2024**

**Вариант № 3**

Разработать на языке Пролог реляционную базу данных (не меньше 5 фактов), соответствующую приведенной таблице, составить 5 правил, удовлетворяющих условиям:

1) проверить, верно ли, что в базе данных есть хотя бы один студент, размер стипендии у которого больше 2 000, но меньше 4 000;

2) проверить, верно ли, что в базе данных есть хотя бы один студент, размер стипендии у которого больше 4 000 или меньше 2 000;

3) проверить, верно ли, что в базе данных есть хотя бы один студент, у которого порядковый номер и номер курса совпадают (если есть, то вывести его фамилию);

4) вывести фамилии всех студентов, обучающихся на курсе, заданном пользователем;

5) проверить, есть ли в базе данных два студента, обучающихся в одной группе и имеющих одинаковый размер стипендии.

К каждому правилу реализовать характерные запросы (не менее двух).

Таблица 1 – Сведения о стипендиях студентов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | ФИО студента | Номер группы | Курс | Средний балл | Размер стипендии |

**ХОД РАБОТЫ**

1. В соответствии с таблицей сведений о стипендиях студентов, были составлены необходимые факты и правила, удовлетворяющие условию задания. Полный код программы на языке SWI Prolog представлен в приложении А.

2. После чего были протестированы правила получившейся пролог-программы:

Сперва было протестировано правило more\_and\_less, при помощи которого можно проверить, есть ли в базе данных хотя бы один студент, у которого размер стипендии больше 2000, но меньше 4000. Результаты тестирования представлены на рисунках 1 и 2.

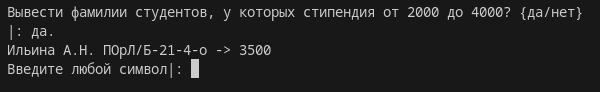


Рисунок 1 – Проверка корректности работы правила more\_and\_less

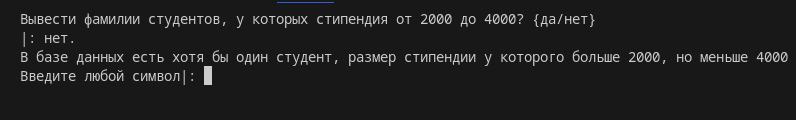


Рисунок 2 – Проверка корректности работы правила more\_and\_less

Затем было протестировано правило more\_or\_less, которое позволяет определить, есть ли в базе данных хотя бы один студент, размер стипендии у которого больше 4000 или меньше 2000. Результаты тестирования представлены на рисунках 3 и 4.

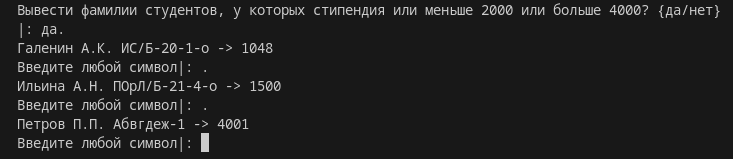


Рисунок 3 – Проверка корректности работы правила more\_or\_less

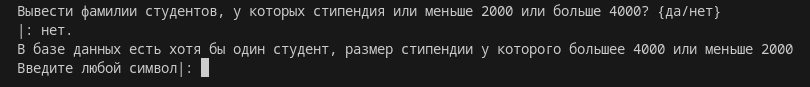


Рисунок 4 – Проверка корректности работы правила more\_or\_less

После чего была проверена корректность работы правила student\_with\_same\_course\_and\_id, которое позволяет проверить, есть ли в базе данных студент у которого порядковый номер совпадает с номером курса. Результаты тестирования представлены на рисунках 5 и 6

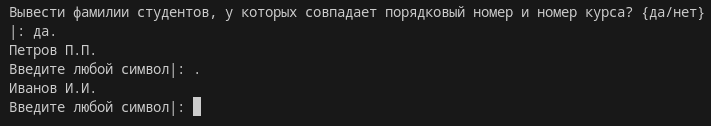


Рисунок 5 – Проверка корректности работы правила student\_with\_same\_course\_and\_id

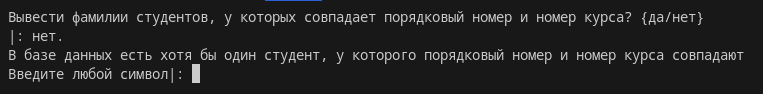


Рисунок 6 – Проверка корректности работы правила student\_with\_same\_course\_and\_id

Затем было протестировано правило find\_students\_by\_course, позволяющее вывести всех студентов одного курса, введенного пользователем системы. Результаты тестирования представлены на рисунках 7 и 8.

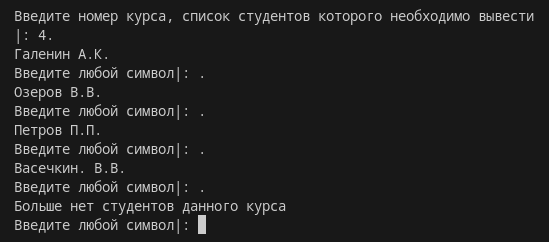


Рисунок 7 – Проверка корректности работы правила find\_students\_by\_course

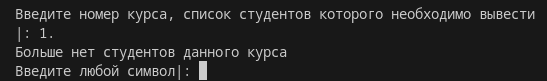


Рисунок 8 – Проверка корректности работы правила find\_students\_by\_course

После чего было протестировано правило matching\_students\_of\_same\_group\_with\_same\_studentship, которое необходимо, чтобы проверить, есть ли в базе данных 2 студента, которые обучаются в одной группе и имеют одинаковый размер стипендии. Результаты тестирования представлены на рисунках 9 и 10.

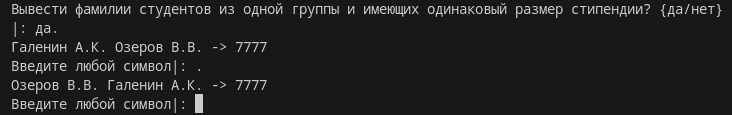


Рисунок 9 – Проверка корректности работы правила matching\_students\_of\_same\_group\_with\_same\_studentship

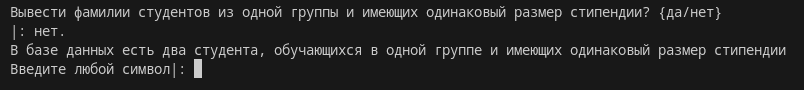


Рисунок 10 – Проверка корректности работы правила matching\_students\_of\_same\_group\_with\_same\_studentship

**ВЫВОДЫ**

В ходе выполнения расчётно-графической работы были закреплены навыки организации динамических баз данных средствами языка Пролог. Была разработана средствами языка Пролог база данных, включающая в себя сведения о стипендиях студентов, также были разработаны факты и пять правил, которые позволили выполнить задание по варианту.

Все правила потом были протестированы, с использованием различных фактов, и в результате тестирования ошибок обнаружено не было.

**ПРИЛОЖЕНИЕ А**

**Код программы**

Файл students.pl

:- dynamic student/6.

student(1, "Галенин А.К.", "ИС/Б-20-1-о", 4, 3, 7777).

student(2, "Ильина А.Н.", "ПОрЛ/Б-21-4-о", 3, 5, 1500).

student(3, "Озеров В.В.", "ИС/Б-20-1-о", 4, 3, 7777).

student(4, "Петров П.П.", "Абвгдеж-1", 4, 4, 4001).

student(5, "Иванов И.И.", "Абвгдеж-2", 5, 5, 3999).

student(6, "Васечкин. В.В.","ИС/Б-20-1-о", 4, 2, 2489).

menu:-

tty\_clear,

write('#############################################################################'),nl,

write('1. Студенты со стипендией > 2000 и < 4000 '),nl,

write('2. Студенты со стипендией > 4000 или < 2000 '),nl,

write('3. Студенты, у которых совпадает номер курса и порядковый номер '),nl,

write('4. Студенты, обучающиеся на заданном курсе '),nl,

write('5. Студенты, обучающиеся в одной группе и имеющие одинаковый размер стипендии'),nl,

write('6. Выход '),nl,

write('{Для выбора пункта меню, необходимо написать соответствующую цифру пункта, С ТОЧКОЙ в конце}'),nl,

write('#############################################################################'),nl,

read(Item), menu\_item(Item).

menu\_item(1):-

more\_and\_less,

menu,!.

menu\_item(2):-

more\_or\_less,

menu,!.

menu\_item(3):-

student\_with\_same\_course\_and\_id,

menu,!.

menu\_item(4):-

find\_students\_by\_course,

menu,!.

menu\_item(5):-

matching\_students\_of\_same\_group\_with\_same\_studentship,

menu,!.

menu\_item(6):-!.

menu\_item(\_Item):-

tty\_clear,

write('Нет такого пункта меню, выберите пункт в пределах от 1 до 6'),nl,

menu.

more\_and\_less:-

tty\_clear,

write('Вывести фамилии студентов, у которых стипендия от 2000 до 4000? {да/нет}'),nl,

read(Answer),

(Answer = да ->

student(\_,FIO,Group,\_,\_,Studentship),

Studentship > 2000, Studentship < 4000,

write(FIO),write(' '),write(Group),write(' -> '),write(Studentship),nl,

write('Введите любой символ'),get(temp),

fail;

Answer = нет ->

student(\_,FIO,Group,\_,\_,Studentship),

(Studentship > 2000, Studentship < 4000),!,

write('В базе данных есть хотя бы один студент, размер стипендии у которого больше 2000, но меньше 4000'),nl,

write('Введите любой символ'),get(temp)

).

more\_or\_less:-

tty\_clear,

write('Вывести фамилии студентов, у которых стипендия или меньше 2000 или больше 4000? {да/нет}'),nl,

read(Answer),

(Answer = да ->

student(\_,FIO,Group,\_,\_,Studentship),

(Studentship > 4000; Studentship < 2000),

write(FIO),write(' '),write(Group),write(' -> '),write(Studentship),nl,

write('Введите любой символ'),get(temp),

fail;

Answer = нет ->

student(\_,FIO,Group,\_,\_,Studentship),

(Studentship > 4000; Studentship < 2000),!,

write('В базе данных есть хотя бы один студент, размер стипендии у которого большее 4000 или меньше 2000'),nl,

write('Введите любой символ'),get(temp)

).

student\_with\_same\_course\_and\_id:-

tty\_clear,

write('Вывести фамилии студентов, у которых совпадает порядковый номер и номер курса? {да/нет}'),nl,

read(Answer),

(Answer = да ->

student(N,FIO,\_,Course,\_,\_),

N = Course,

write(FIO),nl,

write('Введите любой символ'),get(temp),

fail;

Answer = нет ->

student(N,FIO,\_,Course,\_,\_),

N = Course,!,

write('В базе данных есть хотя бы один студент, у которого порядковый номер и номер курса совпадают'),nl,

write('Введите любой символ'),get(temp)

).

find\_students\_by\_course:-

tty\_clear,

write('Введите номер курса, список студентов которого необходимо вывести'),nl,read(User\_Course),

student(N, FIO, Group, Course, Average, Studentship),

member(User\_Course, [Course]),

write(FIO),nl,write('Введите любой символ'),get(temp),fail,!;

write('Больше нет студентов данного курса'),nl,write('Введите любой символ'),get(temp).

matching\_students\_of\_same\_group\_with\_same\_studentship:-

tty\_clear,

write('Вывести фамилии студентов из одной группы и имеющих одинаковый размер стипендии? {да/нет}'),nl,

read(Answer),

(Answer = да ->

student(\_,FIO,Group,\_,\_,Studentship),

student(\_,FIO\_,Group,\_,\_,Studentship),

FIO \= FIO\_,

write(FIO),write(' '),write(FIO\_),write(' -> '),write(Studentship),nl,

write('Введите любой символ'),get(temp);

Answer = нет ->

student(\_,FIO,Group,\_,\_,Studentship),

student(\_,FIO\_,Group,\_,\_,Studentship),

FIO \= FIO\_,!,

write('В базе данных есть два студента, обучающихся в одной группе и имеющих одинаковый размер стипендии'),nl,

write('Введите любой символ'),get(temp)

).

:- menu.