**ЛАБОРАТОРНАЯ РАБОТА №1**

**«СОЗДАНИЕ ДИНАМИЧЕСКИХ БАЗ ДАННЫХ»**

Цель работы.

Изучение технологии подготовки и выполнения Пролог-программ в интегрированной среде, исследование способов организации динамических баз данных (БД) средствами языка Пролог.

Вариант задания.

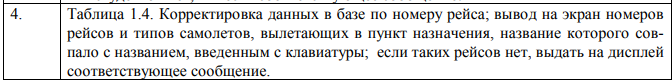


Рисунок 1 – Вариант задания

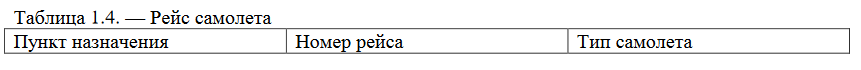


Рисунок 2 – Таблица для базы данных

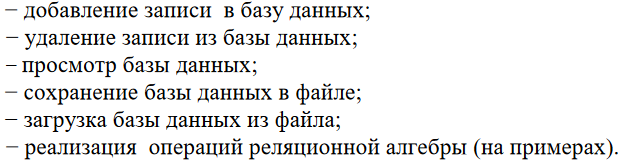


Рисунок 3 – Общее задание

Разработка программы.

Листинг 1 – Код lab1.pl.

:- dynamic(flight/3).

flight(moscow,123,boeing).

flight(paris,456,airbus).

flight(london,789,boeing).

flight(beijing,234,boeing).

flight(newyork,567,airbus).

flight(tokyo,890,boeing).

flight(berlin,345,airbus).

flight(dubai,678,boeing).

flight(sydnei,901,boeing).

flight(rome,123,airbus).

flights(Destination):-

flight(Destination,FlightNumber,Type),

write('Number '), write(FlightNumber), write(' on '), write(Type), nl, fail.

flights(Destination):-

\+ flight(Destination,\_,\_),

write('No flights to this Destination').

changeflightdetails(Destination,FlightNumber,Type,NewNumber) :-

(

flight(\_,FlightNumber,\_)->

retract(flight(Destination,FlightNumber,Type)),

assert(flight(Destination,NewNumber,Type)), nl;

write('No flights with this Number'), nl

).

allfacts() :-

listing(flight).

add\_flight(Destination, FlightNumber, Type) :-

assert(flight(Destination, FlightNumber, Type)).

remove\_flight(Destination, FlightNumber, Type) :-

retract(flight(Destination, FlightNumber, Type)).

read\_from\_file() :-

see('c:/users/belik/onedrive/8\_semestr/misii/misii/test.txt'),

read\_data,

seen.

read\_data :-

repeat,

read(Term),

( Term == end\_of\_file -> true ;

assert(Term), fail

).

write\_to\_file() :-

tell('c:/users/belik/onedrive/8\_semestr/misii/misii/test.txt'),

listing(flight),

told.

union\_flight(Type1, Type2) :-

(flight(Destination, FlightNumber, Type1); flight(Destination, FlightNumber, Type2)),

write('Flight to '), write(Destination), write(' with number '), write(FlightNumber), nl,

fail.

intersection\_flight\_types(Type1, Type2) :-

flight(Destination, FlightNumber, Type1),

flight(Destination, \_, Type2),

write('Flight to '), write(Destination), write(' with number '), write(FlightNumber), nl,

fail.

Для работы программы были созданы предикаты flights(Type) в котором описана логика вывода всех фактов с данным типом, также если введенного типа не существует, то выводится соответствующее сообщение.

Также был написан предикат changeflightdetails(Type, Destination, FlightNumber, NewType), который проверяет есть ли в базе самолеты с типом Type, и если он есть то заменяет все записи на новую с полями (Destination, FlightNumber, NewType), если нет то выводит соответствующее сообщение.

Добавлены предикаты add\_flight(Destination, FlightNumber, Type) и remove\_flight(Destination, FlightNumber, Type), которые добавляют и удаляют записи соответственно.

Добавлены предикаты read\_from\_file() и write\_to\_file(), которые читают или записывают базу знаний в файл соответственно.

Для реализации операций реляционной алгебры были добавлены предикаты union\_flight(Type1, Type2) и intersection\_flight\_types(Type1, Type2), которые отражают операцию объединения и пересечения соответственно.

Выводы программы.

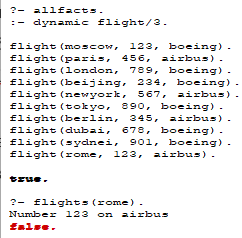


Рисунок 4 – Вывод всех рейсов по типу

На рисунке 4 представлен вывод всех рейсов по типу, который осуществляется путем использования предиката flights(Type).

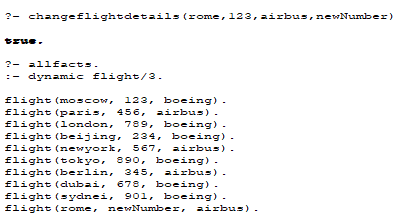


Рисунок 5 – Изменение данных по типу

На рисунке 5 представлено изменение записей с типом airbus на newType. Затем производится проверка изменились ли данные в базе.

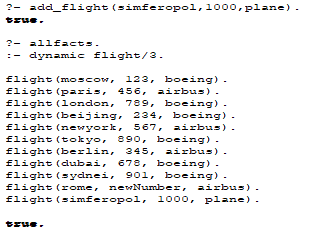


Рисунок 6 – Добавление записи в базу

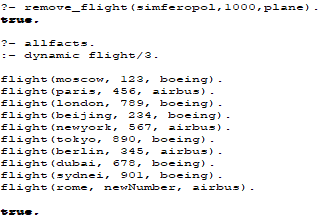


Рисунок 7 – Удаление записи из базы

На рисунках 6 и 7 продемонстрировано добавление и удаление записей в базу.

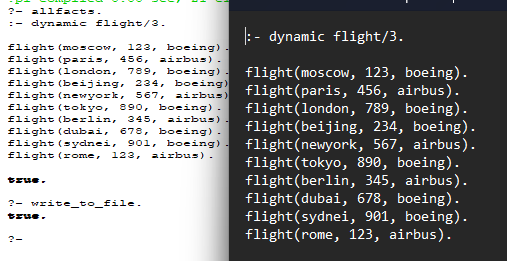


Рисунок 8 – Запись в файл

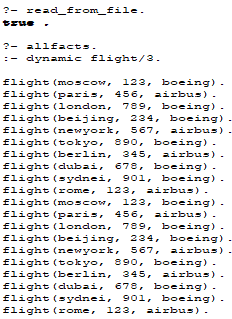


Рисунок 9 – Чтение из файла

На рисунках 8 и 9 представлена запись и чтение из файла.

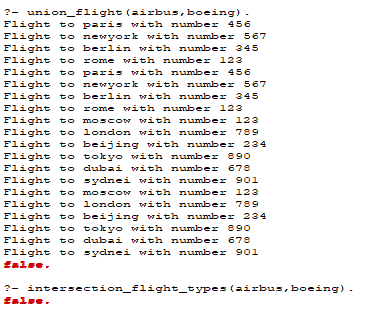


Рисунок 10 – Объединение и пересечение

На рисунке 10 представлена операция объединения и пересечения по типу самолёта.

**ВЫВОДЫ**

В ходе выполнения данной лабораторной работы была изучена технология подготовки и выполнения Пролог-программ в интегрированной среде, исследованы способы организации динамических баз данных (БД) средствами языка Пролог.

Также была создана динамическая база данных с полями: пункт назначения, номер рейса, тип самолёта. Для этой базы были реализованы разные операции такие как: добавление, удаление, изменение, объединение и пересечения.