

Министерство науки и Высшего образования Российской Федерации
Севастопольский государственный университет
Кафедра ИС

Отчет
по лабораторной работе №6
«Создание динамических баз данных»
по дисциплине
«МЕТОДЫ И СРЕДСТВА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

Выполнил студент группы ИС/б-17-2-о
Горбенко К. Н.
Проверил
Забаштанский А.К.

Севастополь
2021

1 ЦЕЛЬ РАБОТЫ

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

2 ПОСТАНОВКА ЗАДАЧИ

Корректировка данных в базе по номеру поезда; вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени; если таких поездов нет – выдать на дисплей соответствующее сообщение.

3 ХОД РАБОТЫ

Текст программы:

```
1 :-dynamic %информирует интерпретатор о том, что определения предикатов
2 %могут изменяться в ходе выполнения программы
3 время/2,
4 поезд/3, % формат: <имя предиката>/<кол-во аргументов>
5 сотрудник_ф/8.
6 % первоначальная база, загружаемая при запуске программы
7 % Пункт назначения, № поезда, врем отправления.
8 поезд('Москва', 101, время(14, 00)).
9 поезд('Казань', 102, время(12, 00)).
10 поезд('Казань', 103, время(13, 00)).
11 поезд('Москва', 103, время(23, 00)).
12 поезд('Симферополь', 104, время(11, 30)).
13 start:- меню. %предикат для запуска программы
14
15 %0===== отображение меню =====
16 меню:-
17 repeat, nl,
18 write('*****'),nl,
19 write('* 1. Добавление записи в БД *'),nl,
20 write('* 2. Удаление записи из БД *'),nl,
21 write('* 3. Выборка записей из БД *'),nl,
22 write('* 4. Просмотр БД *'),nl,
23 write('* 5. Загрузка БД из файла *'),nl,
24 write('* 6. Сохранение БД в файле *'),nl,
25 write('* 7. Реляционные операции *'),nl,
26 write('* 8. Выход *'),nl,
27 write('*****'), nl ,nl,
28 write('Введите номер пункта меню с точкой в конце!!!'),nl,
29 read(C),nl, %Ввод пункта меню
```

```

30 proc(C), %Запуск процедуры с номером C
31 C=8, %Если C не равно 8, то авт. возврат к repeat
32 !. %Иначе успешное завершение
33 %0-----
34
35 %1===== добавление записи в базу данных =====
36 proc(1):-
37 write('Ввод завершайте точкой!!! :'),nl,
38 write('Введите пункт назначения:'),nl, read(Пункт),
39 write('Введите номер поезда:'),nl, read(N),
40 write('Введите час отправления:'),nl, read(Час),
41 write('Введите минуту отправления:'),nl, read(Мин),
42 assertz(поезд(Пункт,N,время(Час,Мин))), %добавление факта в БД
43 write('Поезд №'),write(N),write(' был добавлен в БД'),nl,
44 write('Введите любой символ'),nl, %ожидание ввода литеры
45 get0(_).
46 %1-----
47
48 %2===== удаление записи из базы данных =====
49 proc(2):-
50 write('Введите номер поезда для удаления'), nl,
51 read(N),
52 retract(поезд(_,N,_)), %удаление записи
53 write('Поезд №'),
54 write(N), tab(2), %вывод сообщения об успешном удалении
55 write('был успешно удален из БД'),nl,
56 write('Введите любой символ'),nl,
57 get0(_), %ожидание ввода символа
58 !; %отсечение альтернативы и завершение
59 write('Такого поезда:'),tab(2), %вывод сообщения о безуспешном удалении
60 write('в базе данных нет'),nl,
61 write('Введите любой символ'),nl,
62 get0(_). %ожидание ввода символа
63 %2-----
64
65 %3===== выборка записи из базы данных по критерию =====
66 %----- выбирают поезда, отправляющиеся после времени
        -----
67 proc(3):-
68 write('Введите час отправления:'), nl,
69 read(Ч2), %ввод часа
70 write('Введите час отправления:'), nl,
71 read(М2), %ввод минуты
72 retractall(flag(_)), %удаление фактов - flag(_)
73 nl,write('Поезда:'),nl,
74 поезд(Пункт,N,Время), %выбор записи о сотруднике
75 Время = время(Ч1,М1),
76 отправляется_после(Время,время(Ч2,М2)), %проверка критерия

```

```

77 assert(flag(1)), %запомнить флаг- запись найдена
78 write('Номер: '), write(N),nl,
79 write('Пункт назначения: '), write(Пункт), nl,
80 write('Время отправления: '), write(Ч1), write(':'), write(М2), nl,nl,
81 fail; %возврат для выбора след. записи
82 flag(1),!. %если записи были найдены, то завершить успешно
83
84 proc(3):- %сообщение, если записи не найдены
85 write('В базе нет таких поездов'),nl,
86 write('Введите любой символ'),nl,
87 get0(_),get0(_).
88
89 %проверка времени после
90 отправляется_после(Время,После):-
91 Время = время(Ч1,М1),
92 После = время(Ч2,М2),
93 время_больше(Ч1,М1,Ч2,М2).
94
95 время_больше(Ч1,М1,Ч2,М2):-
96 Ч1>Ч2;
97 Ч1:=Ч2,
98 М1>=М2.
99 %3-----
100
101 %4===== просмотр базы данных =====
102 proc(4):-
103 write('Поезда:'),nl,
104 поезд(Пункт,Н,Время), %извлечение записи из БД
105 Время = время(Час,Мин),
106 write('Номер поезда: '), write(N),nl, %отображение на дисплее
107 write('Пункт назначения: '), write(Пункт), nl, %элементов записи
108 write('Время: '), write(Час), write(':'), write(Мин),nl,nl,
109 fail; %возврат к выбору записи
110 write('Введите любой символ'),nl,
111 get0(_),get0(_), %ожидание ввода символ
112 true. %завершение - записей больше нет
113 %4-----
114
115 %5===== загрузка базы данных из файла =====
116 proc(5):-
117 see('lab1.dat'), %текущий входной поток - lab1.dat
118 retractall(поезд(_,_,_)),%очистка БД от фактов "поезд"
119 db_load, %загрузка БД термами из файла
120 seen, %заккрытие потока
121 write('БД загружена из файла'),nl.
122 %загрузка термов в БД из открытого вх. потока
123 db_load:-
124 read(Term), %чтение термина

```

```

125 (Term == end_of_file,!; %если конец файла, то завершение
126 assertz(Term), %иначе добавить терм в конец БД
127 db_load). %рекурсивный вызов для чтения след. терма
128 %5-----
129
130 %6===== сохранение БД в файле =====
131 proc(6):-
132 tell('lab1.dat'), %открытие вых. потока
133 save_db(поезд(Пункт,N,Время)), %сохранение терма
134 told, %заккрытие вых. потока
135 write('БД скопирована в файл lab1.dat'),nl.
136 %сохранение терма в открытом файле
137 save_db(Term):- %сохранение терма (факта!) Term в БД
138 Term, %отождествление терма с термом в БД
139 write_term(Term, [quoted(true)]), %запись терма
140 write(' '),nl, %запись точки в конце терма
141 fail; %неудача с целью поиска след. варианта
142 true. %завершение, если вариантов отождествления нет
143 %6-----
144
145 %7===== реализация операций реляционной алгебры =====
146 proc(7):-
147 write('Формирование отношения r1: поезда в Москву'), nl,
148 подмножество_поездов('Москва',R1), %R1 - список поездов в Москву
149 список_в_бд(R1), %добавление элементов из R1 в базу данных
150 вывод_списка(R1),nl, %вывод списка R1 на экран
151 write('Формирование отношения r2: поезда в Казань'), nl,
152 подмножество_поездов('Казань',R2), %R2 - список поездов в Казань
153 список_в_бд(R2), %добавление элементов из R2 в базу данных
154 вывод_списка(R2),nl, %вывод списка R2 на экран
155 write('Объединенное отношение r1_или_r2: '), nl,
156 объединение('Москва','Казань',Rez1), %Rez1 - список поездов в Москву и Казань
157 вывод_списка(Rez1),nl,
158 write('Пересечение отношений r1_и_r2: '), nl,
159 пересечение('Москва','Казань',Rez2), %Rez2 - список поездов в оба города
160 вывод_списка(Rez2),nl,
161 write('Разность отношений r1-r2: '), nl,
162 разность('Москва','Казань',Rez3), %Rez3-список поездов в Москву, но не в Казань
163 вывод_списка(Rez3),nl,
164 write('Введите любой символ'),nl,
165 get0(_),get0(_).
166 %-----
167
168 подмножество_поездов(Пункт,R):-
169 bagof(поезд_п(Пункт,N,Время),
170 поезд(Пункт,N,Время), R).
171
172 объединение_r1_r2(П1,П2,Пункт,N,Время):-

```

```

173 поезд_п(П1, N, Время), Пункт=П1;
174 поезд_п(П2, N, Время), Пункт=П2.
175
176 объединение(П1, П2, Rez) :-
177 bagof(поезд_п1_или_п2(Пункт, N, Время),
178 объединение_r1_r2(П1, П2, Пункт, N, Время), %условие вкл. в список
179 Rez).
180
181 пересечение_r1_r2(П1, П2, N, В1, В2) :-
182 поезд_п(П1, N, В1),
183 поезд_п(П2, N, В2).
184
185 пересечение(П1, П2, Rez) :-
186 bagof(поезд_п1_и_п2(П1, N, В1, П2, N, В2),
187 пересечение_r1_r2(П1, П2, N, В1, В2),
188 Rez).
189
190 разность_r1_r2(П1, П2, N, В1, В2) :-
191     поезд_п(П1, N, В1),
192     not(поезд_п(П2, N, В2)).
193
194 разность(П1, П2, Rez) :-
195 bagof(поезд_п1_и_не_п2(П1, N, В1),
196 разность_r1_r2(П1, П2, N, В1, В2),
197 Rez).
198
199
200 список_в_бд([]).
201 список_в_бд([H|T]) :-
202 H=поезд_п(Пункт, N, Время),
203 assertz(поезд_п(Пункт, N, Время)),
204 список_в_бд(T).
205
206 вывод_списка([]).
207 вывод_списка([H|T]) :- write(H), nl, вывод_списка(T).
208 %7-----
209 %8=====выход=====
210 proc(8) :- write('Ну и ладно...'), nl.
211 %8-----

```

ВЫВОДЫ

В ходе работы были изучены технологии подготовки и выполнения Пролог-программ в интегрированной среде, исследование способов организации динамических баз данных (БД) средствами языка Пролог.