

Приложение А (справочное) Создание динамической базы данных

```

:-dynamic                %информирует интерпретатор о том, что определения предикатов
                        %могут изменяться в ходе выполнения программы
сотрудник/8,            % формат: <имя предиката>/<кол-во аргументов>
сотрудник_ф/8.

% первоначальная база, загружаемая при запуске программы
%      N      Фам.      Имя      Отч.      Отд. Должн.      Филиал Тел.
сотрудник(101, петренко, сергей, иванович, 4, инженер, керчь, 68-23-45).
сотрудник(102, бенюк, иван, николаевич, 2, оператор, ялта, 11-25-32).
сотрудник(103, григорьев, григорий, васильевич, 2, оператор, форос, 20-25-32).
сотрудник(104, михайлов, григорий, андреевич, 2, оператор, алупка, 68-23-42).
сотрудник(105, бенюк, иван, николаевич, 3, оператор, керчь, 68-23-42).

start:- menu.           %предикат для запуска программы

%0===== отображение меню =====
menu:-
    repeat, nl,
    write('*****'), nl,
    write('* 1. Добавление записи в БД *'), nl,
    write('* 2. Удаление записи из БД *'), nl,
    write('* 3. Выборка записей из БД *'), nl,
    write('* 4. Просмотр БД *'), nl,
    write('* 5. Загрузка БД из файла *'), nl,
    write('* 6. Сохранение БД в файле *'), nl,
    write('* 7. Реляционные операции *'), nl,
    write('* 8. Выход *'), nl,
    write('*****'), nl, nl,
    write('Введите номер пункта меню с точкой в конце!!!'), nl,
    read(C), nl,                %Ввод пункта меню
    proc(C),                   %Запуск процедуры с номером C
    C=8,                       %Если C не равно 8, то авт. возврат к repeat
    !.                         %Иначе успешное завершение
%0-----

%1===== добавление записи в базу данных =====
proc(1):-
    write('Ввод завершайте точкой!!! :'), nl,
    write('Введите номер:'), nl, read(N),
    write('Введите фамилию:'), nl, read(Фам),
    write('Введите имя:'), nl, read(Имя),
    write('Введите отчество:'), nl, read(Отч),
    write('Введите отдел:'), nl, read(Отд),
    write('Введите должность:'), nl, read(Должн),
    write('Введите филиал:'), nl, read(Филиал),
    write('Введите телефон:'), nl, read(Тел),
    assertz(сотрудник(N, Фам, Имя, Отч, Отд, Должн, Филиал, Тел)), %добавление факта в БД
    write(Фам), write(' был добавлен в БД'), nl,
    write('Введите любой символ'), nl,                                %ожидание ввода литеры
    get0(C).
%1-----

%2===== удаление записи из базы данных =====
proc(2):-
    write('Введите номер для удаления сотрудника'), nl,
    read(N),                %ввод номера сотрудника
    retract(сотрудник(N,_,_,_,_,_,_)), %удаление записи о сотруднике
    write('Сотрудник:'), tab(2),
    write(N), tab(2),        %вывод сообщения об успешном удалении
    write('был успешно удален из БД'), nl,
    write('Введите любой символ'), nl,

```

```

get0(C),                                %ожидание ввода символа
!;                                     %отсечение альтернативы и завершение
write('Такого сотрудника:'), tab(2),   %вывод сообщения о безуспешном удалении
write('в базе данных нет'), nl,
write('Введите любой символ'), nl,
get0(C).                                %ожидание ввода символа
%2-----

%3===== выборка записи из базы данных по критерию =====
%----- выбираются сотрудники с фамилией на Б или Г -----
proc(3):-
    retractall(flag(_)),                %удаление фактов - flag(_)
    сотрудник(N, Фам, Имя, Отч, Отд, Должн, Филиал, Тел), %выбор записи о сотруднике
    фамилия_с_буквы(Фам, 'ББГГ'),      %проверка критерия
    assert(flag(1)),                    %запомнить флаг - запись найдена
    nl,
    write('Номер: '), write(N), nl,
    write('Фамилия: '), write(Фам), tab(2),
    write('Имя: '), write(Имя), tab(2),
    write('Отчество: '), write(Отч), nl,
    write('Отдел: '), write(Отд), tab(2),
    write('Должность: '), write(Должн), nl,
    write('Филиал: '), write(Филиал), tab(2),
    write('Телефон: '), write(Тел), nl, nl,
    write('Введите любой символ'), nl,
    get0(C1), get0(C2),
    fail;                               %возврат для выбора след. записи
    flag(1), !.                         %если записи были найдены, то завершить успешно

proc(3):-                               %сообщение, если записи не найдены
    write('В базе нет сотрудников с фамилиями на Б или Г'), nl,
    write('Введите любой символ'), nl,
    get0(C1), get0(C2).

%проверка принадлежности первой буквы фамилии атому из букв: 'ББГГ'
фамилия_с_буквы(Фам, Буквы):-
    name(Фам, [H|T]),                  %преобразование фамилии в список кодов букв
    name(Буквы, СписокКодов),          %преобразование атома из букв в список их кодов
    принадлежит(H, СписокКодов). %принадлежит ли первая буква фамилии списку кодов

%проверка принадлежности элемента списку
принадлежит(H, [H|T]).
принадлежит(H, [_|T]) :- принадлежит(H, T).
%3-----

%4===== просмотр базы данных =====
proc(4):-
    сотрудник(N, Фам, Имя, Отч, Отд, Должн, Филиал, Тел), %извлечение записи из БД
    nl,
    write('Номер: '), write(N), nl,      %отображение на дисплее
    write('Фамилия: '), write(Фам), tab(2), %элементов записи
    write('Имя: '), write(Имя), tab(2),
    write('Отчество: '), write(Отч), nl,
    write('Отдел: '), write(Отд), tab(2),
    write('Должность: '), write(Должн), nl,
    write('Филиал: '), write(Филиал), tab(2),
    write('Телефон: '), write(Тел), nl, nl,
    write('Введите любой символ'), nl,
    get0(C1), get0(C2),                  %ожидание ввода символа
    fail;                               %возврат к выбору записи
    true.                               %завершение - записей больше нет
%4-----

%5===== загрузка базы данных из файла =====
proc(5):-
    see('d:/db_sotrudnik.dat'),         текущий входной поток - d:/db_sotrudnik.dat

```

```

    retractall(сотрудник( _,_,_,_,_,_,_)), %очистка БД от фактов "сотрудник"
    db_load, %загрузка БД термами из файла
    seen, %закрытие потока
    write('БД загружена из файла'),nl.

%загрузка термов в БД из открытого вх. потока
db_load:-
    read(Term), %чтение термина
    (Term == end_of_file, !; %если конец файла, то завершение
    assertz(Term), %иначе добавить терм в конец БД
    db_load). %рекурсивный вызов для чтения след. термина
%5-----

%6===== сохранение БД в файле =====
proc(6):-
    tell('d:/db_sotrudnik.dat'), %открытие вых. потока
    save_db(сотрудник(N,Фам,Имя,Отч,Отд,Должн,Филиал,Тел)), %сохранение термина
    told, %закрытие вых. потока
    write('БД скопирована в файл d:/db_sotrudnik.dat'),nl.

%сохранение термина в открытом файле
save_db(Term):-
    Term, %сохранение термина (факта!) Терм в БД
    %отождествление термина с термом в БД
    write(Term), %запись термина
    write('.'),nl, %запись точки в конце термина
    fail; %неудача с целью поиска след. варианта
    true. %завершение, если вариантов отождествления нет
%6-----

%7===== реализация операций реляционной алгебры =====
proc(7):-
    nl,
    write('Формирование отношения r1: сотрудники филиала "Ялта" '), nl,
    подмножество_сотрудников(ялта,R1), %R1 - список сотрудников филиала 1
    список_в_бд(R1), %добавление элементов из R1 в базу данных
    вывод_списка(R1),nl, %вывод списка R1 на экран

    write('Формирование отношения r2: сотрудники филиала "Керчь" '), nl,
    подмножество_сотрудников(керчь,R2), %R2 - список сотрудников филиала 2
    список_в_бд(R2), %добавление элементов из R2 в базу данных
    вывод_списка(R2),nl, %вывод списка R2 на экран

    write('Объединенное отношение r1_или_r2: '), nl,
    объединение(Rez1), %Rez1 - список сотрудников филиала 1 или 2
    вывод_списка(Rez1),nl,

    write('Пересечение отношений r1_и_r2: '), nl,
    пересечение(Rez2), %Rez2 - список сотрудников 2-х филиалов
    вывод_списка(Rez2),nl,

    write('Разность отношений r1-r2: '), nl,
    разность(Rez3), %Rez3-список сотрудников филиала 1 без фил.2
    вывод_списка(Rez3),nl,

    write('Введите любой символ'),nl,
    get0(C). %Ожидание ввода символа
%-----

%формирование подмножества сотрудников R заданного Филиала
%подмножество R представляется в виде списка термов "сотрудник_ф(...)"
подмножество_сотрудников(Филиал,R):-
    bagof(сотрудник_ф(N,Фам,Имя,Отч,Отд,Должн,Филиал,Тел),
    сотрудник(N,Фам,Имя,Отч,Отд,Должн,Филиал,Тел), R) .

%правило объединения отношений - r1 или r2
%объединяются отношения сотрудник_ф(ялта) и сотрудник_ф(керчь)
объединение_r1_r2(X1,X2,X3,X4,X5,X6,X7,X8):-
    сотрудник_ф(X1,X2,X3,X4,X5,X6,ялта,X8), X7=ялта;

```

сотрудник_ф(X1,X2,X3,X4,X5,X6,керчь,X8),X7=керчь.

%формирование списка Rez из фактов "сотрудник_ф1_или_ф2"

объединение (Rez) :-

bagof(сотрудник_ф1_или_ф2(X1,X2,X3,X4,X5,X6,X7,X8),
объединение_r1_r2(X1,X2,X3,X4,X5,X6,X7,X8), %условие вкл. в список
Rez).

%правило **пересечения отношений - r1 и r2**

%строится пересечение отношений сотрудник_ф(ялта) и сотрудник_ф(керчь)

%пересечение **строится только по совпадению ф.и.о.** сотрудников

%обозначения аргументов: первая цифра - номер филиала, вторая - номер аргумента

% например: X11 - филиал 1, аргумент 1 (т.е. номер)

% X27 - филиал 2, аргумент 7 (т.е. название филиала)

%в конец списка аргументов отношения пересечения дописываются:

%филиал2 (X27),номер2(X21),отд2(X25),долж2 (X26)

пересечение_r1_r2(X11,X12,X13,X14,X15,X16,X17,X18,X27,X21,X25,X26) :-

сотрудник_ф(X11,X12,X13,X14,X15,X16,ялта,X18),X17=ялта,
сотрудник_ф(X21,X12,X13,X14,X25,X26,керчь,X28),X27=керчь.

%формирование списка Rez из фактов "сотрудник_ф1_и_ф2"

пересечение (Rez) :-

bagof(сотрудник_ф1_и_ф2(X11,X12,X13,X14,X15,X16,X17,X18,X27,X21,X25,X26),
пересечение_r1_r2(X11,X12,X13,X14,X15,X16,X17,X18,X27,X21,X25,X26),
Rez).

%правило **построения разности отношений: r1-r2**

%находится разность сотрудники_ф(ялта)- сотрудники_ф(керчь)

%учитываются только сотрудники с совпадающими ф.и.о.

разность_r1_r2(X11,X12,X13,X14,X15,X16,X17,X18) :-

сотрудник_ф(X11,X12,X13,X14,X15,X16,ялта,X18),X17=ялта,
not(сотрудник_ф(X21,X12,X13,X14,X25,X26,керчь,X28)),X27=керчь.

%построение списка Rez из фактов "сотрудник_ф1_и_не_ф2"

разность (Rez) :-

bagof(сотрудник_ф1_и_не_ф2(X11,X12,X13,X14,X15,X16,X17,X18),
разность_r1_r2(X11,X12,X13,X14,X15,X16,X17,X18), %условие вкл. в список
Rez).

%добавление термов из списка [H|T] в БД

список_в_бд([]).

список_в_бд([H|T]) :-

H=сотрудник_ф(N,Фам,Имя,Отч,Отд,Должн,Филиал,Тел),

assertz(сотрудник_ф(N,Фам,Имя,Отч,Отд,Должн,Филиал,Тел)),

список_в_бд(T). %Рекурсивный вызов для след. терма

%**Вывод элементов списка** [H|T] в каждой строке

вывод_списка([]).

вывод_списка([H|T]) :-**write**(H),**nl**,вывод_списка(T).

%7-----

%8=====ВЫХОД=====

proc(8) :-**write**('Досвидания'),**nl**.

%8-----