

U18CO018
Shekhaliya Shubham
PPL
Lab assignment 5

Format of structure:

student(rollno, name, address(building_name, city,
zipcode),[sub(teacher_code,subject_name)]).

Following data are in represented according to the format given:

student(u223, ram, address(shlimar_park, surat, 395001), [sub(t1,algebra),
sub(t2,physics), sub(t3,english), sub(t5,hindi)]).

student(u226, lakshman, address(honey_park, delhi, 110002), [sub(t3, history),
sub(t4,science), sub(t1,geometry), sub(t5, hindi)]).

student(u227, bharat, address(shally_tower, mumbai,400004), [sub(t1,geometry),
sub(t2, chemistry), sub(t3,english_grammer)]).

Find the results for following questions using PROLOG program:

1. Find total number of subjects chosen by each student.
2. Find name and zipcode of each student.
3. Write roll no. and name of all students staying in delhi.
4. List name of all subjects taken by teacher t1.
5. List roll no. of all students learning "hindi" subject
6. List building_name and city_code for all students in the given format (format: [(building_name, citycode)]).
7. List all teachers (teaching codes) for each given student.
8. Find the subject, which is maximum chosen.
9. List all subjects taken by each teacher.

Code:

```
student(u223, ram, address(shlimar_park, surat, 395001), [subj(t1,algebra),  
subj(t2,physics), subj(t3,english), subj(t5,hindi)]).
```

```
student(u226, lakshman, address(honey_park, delhi, 110002),[subj(t3, history),  
subj(t4,science), subj(t1,geometry), subj(t5, hindi)]).
```

```
student(u227, bharat, address(shally_tower, mumbai,400004 ), [subj(t1,geometry),  
subj(t2, chemistry), subj(t3,english_grammer)]).
```

```
% flattens list of lists into lists
```

```
flatten1([],[]).
```

```
flatten1([H|T],W):-
```

```
    flatten1(T,W1),
```

```
    append(W1,H,W).
```

```
% remove duplicates from List1 and form List2
```

```
remove_duplicates([],[]).
```

```
remove_duplicates([H|T], [H|T2]):- not(member(H,T)), remove_duplicates(T,T2).
```

```
remove_duplicates([H|T], L2):- member(H,T), remove_duplicates(T,L2).
```

```
% get all the subjects
```

```
get_all_subjects(Subjects):-
```

```
    findall(X,student(_,_ ,X),L),
```

```
    flatten1(L,S),
```

```
    remove_duplicates(S,Subjects).
```

```
total_subjects:-student(_X,_Y), length(Y,L), write(X), write(": "), write(L).
```

```
name_and_zip:-student(_,X,address(_,_,Zip),_),write(X), write(" has zip code: "),
write(Zip).
```

```
delhi_students:-student(Roll,Name,address(_,delhi,_,_), write("Name: "),
write(Name), write(" Roll No: "), write(Roll).
```

```
teaches(Teacher):- get_all_subjects(Subjects), teaches(Teacher, Subjects,
TeacherSubjects), write("Teacher "), write(Teacher), write(" teaches: "),
write(TeacherSubjects), nl, !.
```

```
%all students learning hindi subject
contains_hindi([]):-fail.
contains_hindi([subj(_,hindi)|_]):-!.
contains_hindi([_|T]):-contains_hindi(T).
```

```
hindi_students:-student(Roll,_,_,X), contains_hindi(X), write(Roll), nl.
%building name and city code of all students
```

```
cityaddress:- student(_,_,address(Building,_, Code),_), write("("), write(Building),
write(", "), write(Code), write(")"), nl.
```

```
%list teachers for student
```

```
extract_teachers([],[]).
extract_teachers([subj(T,_)|Rest], [T|Tail]):-extract_teachers(Rest,Tail).
```

```
teacher_for_students:- student(_,Name,_,Y),extract_teachers(Y,Z), write(Name),
write(": "), write(Z), nl.
```

```
% max choosen subject (subj, Count)
```

```
get_subject_freq([],_,0).
```

```
get_subject_freq([subj(_,S)|T],S,C):-get_subject_freq(T,S,C1), C is C1 + 1, !.
```

```
get_subject_freq([_|T],S,C):-get_subject_freq(T,S,C).
```

```
max_occurring_subject(Subjects,Subject):- get_subject_freq(Subjects,Subject,C),
```

```
forall(member(subj(_,OtherSubject),
```

```
Subjects),(get_subject_freq(Subjects,OtherSubject,C1), C >= C1)).
```

```
max_chosen_subject(Subject):-
```

```
    findall(X,student(_,_,X),L),
```

```
    flatten1(L,Subjects),
```

```
    max_occurring_subject(Subjects,Subject).
```

```
% subject by each teacher
```

```
get_all_teachers([],[]).
```

```
get_all_teachers([subj(H,_)|T],[H|T1]):-forall(member(subj(K,_)|T), K \= H),get_all_teachers(T,T1),!.
```

```
get_all_teachers([_|T],T1):-get_all_teachers(T,T1).
```

```
teaches(_,[],[]).
```

```
teaches(T,[subj(T,Subj)|Tail], Y):-member(subj(T,Subj), Tail), teaches(T,Tail,Y).
```

```
teaches(T,[subj(T,Subj)|Tail], [Subj|Rest]):-teaches(T,Tail,Rest).
```

```
teaches(T,[_|Tail], Y):-teaches(T,Tail,Y).
```

```
subjects:- get_all_subjects(Subjects), get_all_teachers(Subjects,Teachers),
```

```
forall(member(Teacher,Teachers), (teaches(Teacher, Subjects, TeacherSubjects),
```

```
write(Teacher), write(" teaches: "), write(TeacherSubjects), nl)).
```

1.

```
?-  
|   total_subjects.  
ram: 4  
true ;  
lakshman: 4  
true ;  
bharat: 3  
true.
```

2.

```
?- name_and_zip.  
ram has zip code: 395001  
true ;  
lakshman has zip code: 110002  
true ;  
bharat has zip code: 400004  
true.
```

3.

```
?- delhi_students.  
Name: lakshman Roll No: u226  
true.
```

4.

```
?- teaches(t1).  
Teacher t1 teaches: [algebra,geometry]  
true.
```

5.

```
?- hindi_students.  
u223  
true ;  
u226  
true ;  
false.
```

6.

```
?- cityaddress.  
(shlimar_park, 395001)  
true ;  
(honey_park, 110002)  
true ;  
(shally_tower, 400004)  
true.
```

7.

```
?- teacher_for_students.  
ram: [t1,t2,t3,t5]  
true ;  
lakshman: [t3,t4,t1,t5]  
true ;  
bharat: [t1,t2,t3]  
true.
```

8.

```
?- max_chosen_subject(X).  
X = geometry.
```

9.

```
?- subjects.  
t4 teaches: [science]  
t5 teaches: [hindi]  
t1 teaches: [algebra,geometry]  
t2 teaches: [physics,chemistry]  
t3 teaches: [english,history,english_grammar]  
true ;
```