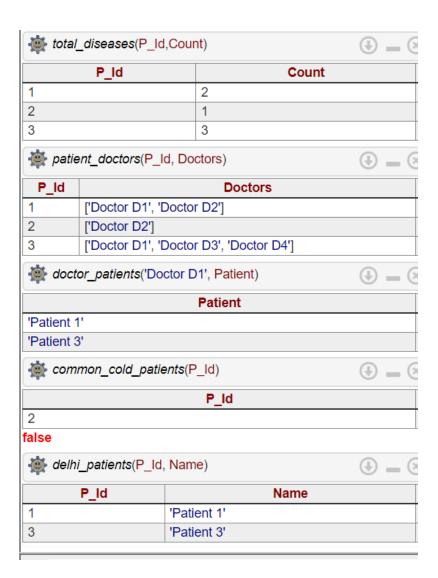
PRACTICAL EXAM

U20CS005 BANSI MARAKANA

```
QUESTION 1:
indication(fever).
indication(rash).
indication(headache).
indication(runny nose).
indication(conjunctivitis).
indication(cough).
indication(body ache).
indication(chills).
indication(sore_throat).
indication(sneezing).
name('Abc').
name('Def').
name('Ghi').
disease('Flu', fever, headache, body ache, conjunctivitis, chills, sore throat, runny nose,
cough]).
disease('Common cold', [headache, sneezing, sore throat, runny nose, chills]).
disease('Chicken pox',[fever, chills, body ache, rash]).
disease('Measles',[cough, sneezing, runny_nose]).
patient(1, 'Patient 1', 'ABC Apartment, Delhi - 110001', [treatment('Doctor D1', 'Flu'),
treatment('Doctor D2', 'Measles')]).
patient(2, 'Patient 2', 'XYZ Building, Gujarat - 400069', [treatment('Doctor D2', 'Common cold')]).
patient(3, 'Patient 3', 'PQR Society, Delhi - 110029', [treatment('Doctor D1', 'Flu'),
treatment('Doctor D3', 'Chicken pox'), treatment('Doctor D4', 'Measles')]).
symptom(Name, Indication) :- write('Does'), write(Name), write(' have a '),
  write(Indication), write('? (y/n)'), nl,
  response(Reply), Reply='y'.
hypothesis(Name, Disease):- disease(Disease, Indications),
  forall(member(I, Indications), symptom(Name, I)).
%1
total_diseases(P_Id, Count) :-
  patient(P_Id, _, _, Treatments),
  findall(Disease, member(treatment( , Disease), Treatments), Diseases),
  length(Diseases, Count).
```

```
%2
name and zipcode(P Id, Name, Zipcode):-
  patient(P Id, Name, Address, ),
  atom codes(Address, Codes),
  reverse(Codes, RCodes),
  phrase(postal(Zip), RCodes),
  reverse(Zip, RZip),
  atom_codes(Zipcode, RZip).
%3
delhi_patients(P_Id, Name):- patient(P_Id, Name, Address, _),sub_atom(Address, _, _, _,
'Delhi').
%4.
doctor_patients('Doctor D1', Patient) :-
  patient(_, Patient, _, Treatments),
  member(treatment('Doctor D1', ), Treatments).
%5
common cold patients(P Id):-
  patient(P_Id, _, _, Treatments),
  member(treatment(_, 'Common cold'), Treatments).
%6
patient_details(P_Id, Details) :-
  patient(P_Id, _, Address, _),
  sub_atom(Address, B, _, A, ','),
  sub_atom(Address, _, A, 0, CityZip),
  atom_number(CityZip, Zip),
  atom_codes(Building, Address, 0, B),
  tuple(Building, Zip, Details).
%7
patient_doctors(P_Id, Doctors) :-
  patient(P_Id, _, _, Treatments),
  findall(Doc, member(treatment(Doc, _), Treatments), Doctors).
```



QUESTION 2:

```
#include <bits/stdc++.h>
using namespace std;
const int N = 20;
int n, tsp[N][N], seen[1 << N][N], ans = INT MAX;
struct node
};
void bfs()
    q.push({1, 0, 0});
    seen[1][0] = 1;
    while (!q.empty())
       node nd = q.front();
        q.pop();
        if (nd.removed == (1 << n) - 1 && tsp[nd.current][0])</pre>
            ans = min(ans, nd.cost + tsp[nd.current][0]);
            if (!seen[nd.removed | (1 << i)][i] && tsp[nd.current][i])</pre>
                 seen[nd.removed | (1 << i)][i] = 1;</pre>
                q.push({nd.removed | (1 << i), i, nd.cost +</pre>
tsp[nd.current][i]});
int main()
    cout << "\nTSP using BFS\n";</pre>
```

```
TSP using BFS
Enter the number of nodes: 3
Enter adjacency matrix:
1 0 1
1 1 1
1 1 1
Minimum cost is: 3
```