CSLR 51 : DBMS LAB-7

Roll no. : **106119100**

Name : **Rajneesh Pandey**

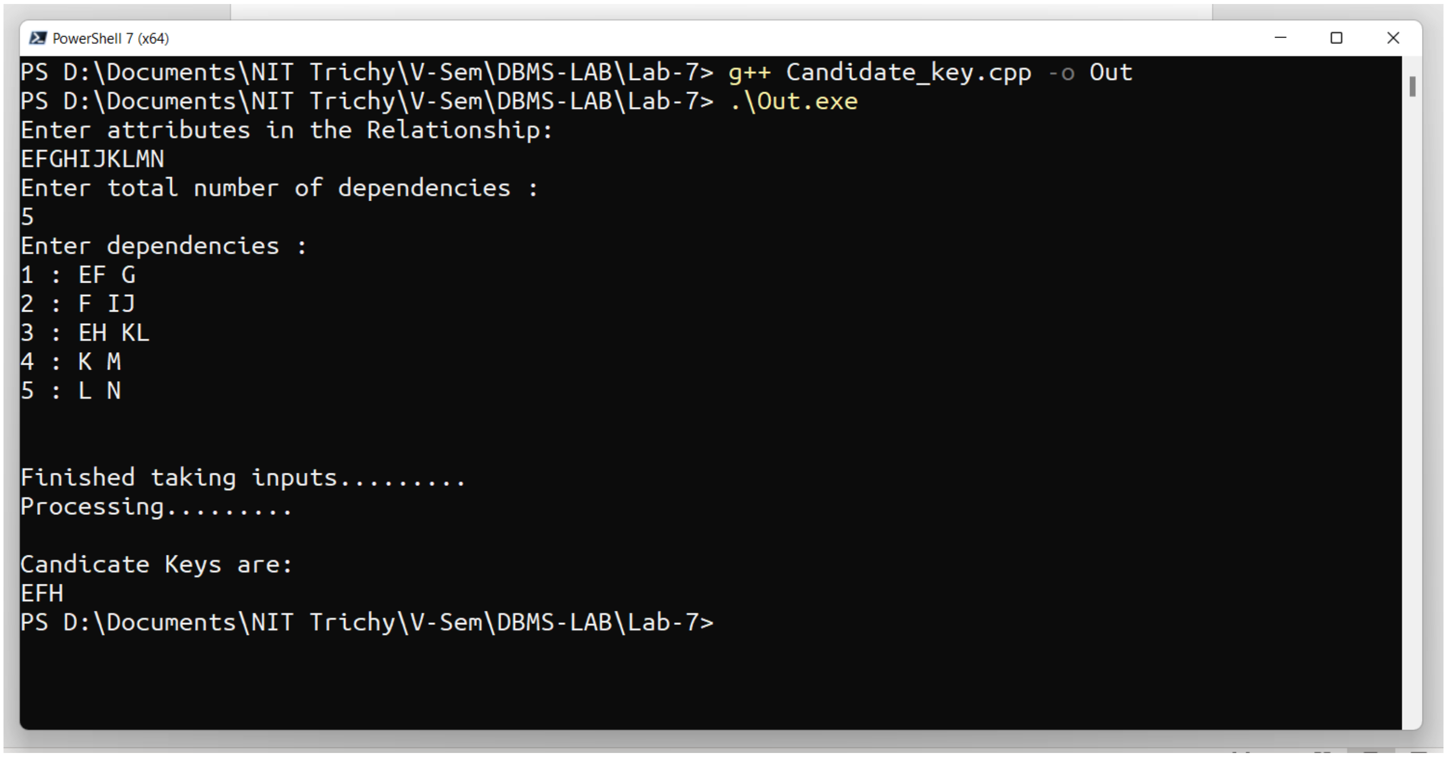
Section : **CSE-B**

CODE

```
/*
CSLR-51 : DBMS LAB-7

Roll no. : 106119100
Name : Rajneesh Pandey
Section : CSE-B
*/

#include <bits/stdc++.h>
using namespace std;

vector<string> ans;
vector<int> mapps;
vector<int> nodes;
string relationship_alpha;
int relationship_alpha_len;
int dependency;

unordered_map<int, int> depends;
unordered_map<char, int> alpha_to_int;
unordered_map<int, char> int_to_alpha;

string mask_to_string(const int &mask)
{
    string str = "";
    for (int i = 0; i < relationship_alpha_len; i++)
    {
        if ((mask >> i) & 1)
            str += int_to_alpha[i];
    }
    return str;
}

int string_to_mask(const string &s)
{
    int mask = 0;
    for (auto &ch : s)
    {
        mask |= (1 << alpha_to_int[ch]);
    }

    return mask;
}

void init()
{

    cout << "Enter attributes in the Relationship: \n";
    cin >> relationship_alpha;

    relationship_alpha_len = relationship_alpha.length();

    mapps.assign(1 << relationship_alpha_len, 0);

    for (int i = 0; i < (1 << relationship_alpha_len); i++)
        mapps[i] = i;

    for (int i = 0; i < relationship_alpha_len; i++)
    {
        alpha_to_int[relationship_alpha[i]] = i;
        int_to_alpha[i] = relationship_alpha[i];
    }
    cout << "Enter total number of dependencies :\n";
    cin >> dependency;
    cout << "Enter dependencies : \n";
    for (int i = 0; i < dependency; i++)
    {
        cout<< i + 1 << " : ";
        string lhs, rhs;
        cin >> lhs >> rhs;
        depends[string_to_mask(lhs)] = depends[string_to_mask(lh
s)] | string_to_mask(rhs);
    }
    cout << "\n\nFinished taking inputs.........\n";
    cout << "Processing.........\n";
}
/* C+ = C; while (there is changes to C+)
    { do (for each functional dependency X-
>Y in F) { if (X⊆C+) then C+= C+∪Y } }
*/
int get_closures(int mask)
{
    int c = mask;
    int prevc = 0;
    while (c != prevc)
    {
        prevc = c;
        for (auto &ele : depends)
        {
            if ((c & ele.first) == ele.first)
                c |= ele.second;
        }
    }
    return c;
}
void get_closures_all()
{
    for (int i = 0; i < (1 << relationship_alpha_len); i++)
    {
        mapps[i] = get_closures(i);
    }
}
void get_keys()
{
    bool found = false;
    int total = (1 << relationship_alpha_len) - 1;
    for (int len = 1; len <= relationship_alpha_len; len++)
    {
        vector<int> perm(relationship_alpha_len, 0);
        for (int t = relationship_alpha_len - len; t < relations
hip_alpha_len; perm[t++] = 1)
            ;
        do
        {
            int mask = 0;
            for (int i = 0; i < relationship_alpha_len; i++)
                if (perm[i])
                    mask |= (1 << i);
            if (mapps[mask] == total)
            {
                found = true;
                ans.push_back(mask_to_string(mask));
            }
        } while (next_permutation(perm.begin(), perm.end()));
        if (found)
            break;
    }
    cout << "\nCandicate Keys are: \n";
    for (auto &cands : ans)
        cout << cands << " ";
}
int main()
{
    init();
    get_closures_all();
    get_keys();
    return 0;
}
```

Problem 1

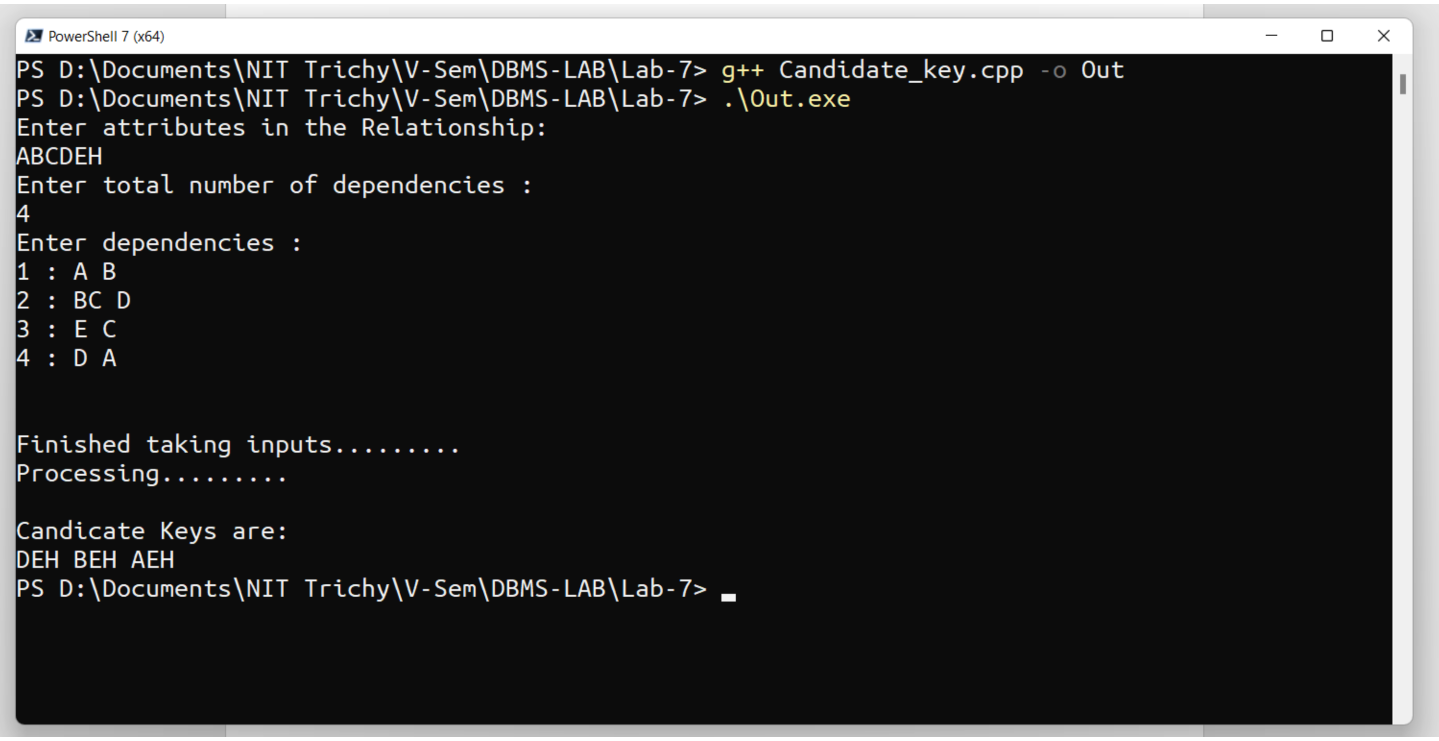
1. Consider the relation scheme R = {E, F, G, H, I, J, K, L, M, N} and the set of functional dependencies {{E, F} -> {G}, {F} -> {I, J}, {E, H} -> {K, L}, K -> {M}, L -> {N}} on R. What is the key for R?



```
PS D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-7> g++ Candidate_key.cpp -o Out
PS D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-7> .\Out.exe
Enter attributes in the Relationship:
EFGHIJKLMN
Enter total number of dependencies :
5
Enter dependencies :
1 : EF G
2 : F IJ
3 : EH KL
4 : K M
5 : L N


Finished taking inputs.........
Processing.........

Candicate Keys are:
EFH
PS D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-7>
```

Problem 2

Consider a relation scheme R = (A, B, C, D, E, H) on which the following functional dependencies hold: {A–>B, BC–> D, E–>C, D–>A}. What are the candidate keys of R? write a c program to find it.



```
PS D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-7> g++ Candidate_key.cpp -o Out
PS D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-7> .\Out.exe
Enter attributes in the Relationship:
ABCDEH
Enter total number of dependencies :
4
Enter dependencies :
1 : A B
2 : BC D
3 : E C
4 : D A


Finished taking inputs.........
Processing.........

Candicate Keys are:
DEH BEH AEH
PS D:\Documents\NIT Trichy\V-Sem\DBMS-LAB\Lab-7>
```