UNIVERSITY OF SCIENCE

VIETNAM NATIONAL UNIVERSITY

PROJECT 2

LOGIC

Lecturer

Bui Tien Len

Nguyen Ngoc Duc

Ho Chi Minh City, April 23th, 2022

Contents

1. INFORMATION

Lecturer

- Bui Tien Len
- Nguyen Ngoc Duc

Group of students

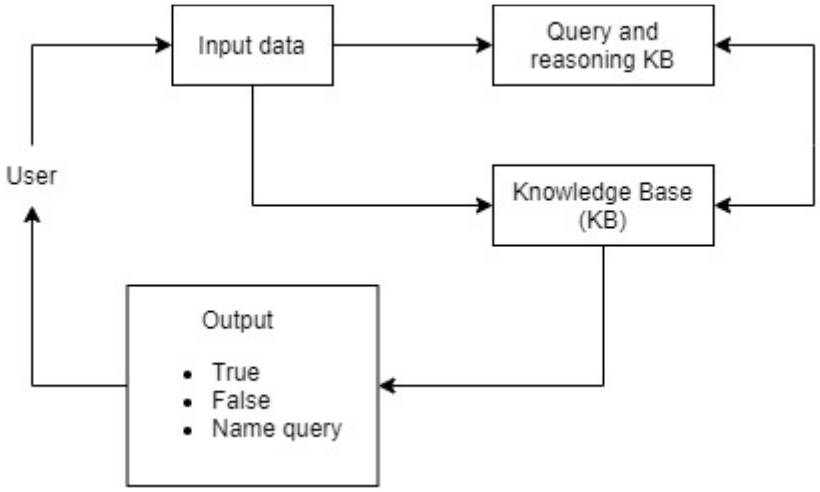
Class	Student ID	Name	Contribution
20CLC08	20127039	Tran Dam Gia Huy	33.3%
20CLC08	20127043	Nguyen Thoai Dang Khoa	33.3%
20CLC08	20127050	Nguyen Duc Minh	33.3%

The percentage of completion

Task		Accomplish	Total
Construct database	**parent**(*Parent*,*Child*), **married**(*Person*, *Person*), **divorced**(*Person*, *Person*).	X	100%
	male(*Person*), **female**(*Person*),	X	
Define predicates	**father**(*Parent*,*Child*), **mother**(*Parent*,*Child*),	X	
	child(*Child*,*Parent*), **son**(*Child*,*Parent*), **daughter**(*Child*,*Parent*),	X	
	grandparent(*GP*,*GC*), **grandmother**(*GM*,*GC*), **grandfather**(*GF*,GC),	X	
	grandchild(*GC*,*GP*), **grandson**(*GS*,*GP*), **granddaughter**(*GD*,*GP*),	X	
	spouse(*Husband*,*Wife*), **husband**(*Person*,*Wife*), **wife**(*Person*,*Husband*),	X	
	sibling(*Person1*,*Person2*), **brother**(*Person*,*Sibling*), **sister**(*Person*,*Sibling*).	X	
	aunt, **uncle**, **nephew, niece**, and **firstCousin**	X	
Handle	Avoid to duplicate result	X	

2. CONTENT

2.1.Diagram



2.2.Construct database

- **parent**(*Parent*,*Child*)
  - *Parent* is parent of *Child*, maybe father of mother.
  - Two parameter needs to be in order
  - Example implement (Queen – Prince Philip have 4 children namely Charles, Andrew, Anne, Edward), etc:

```
/* Queen - Prince Philip */

parent(queen,prince_charles).
parent(prince_philip,prince_charles).

parent(queen,prince_andrew).
parent(prince_philip,prince_andrew).

parent(queen,princess_anne).
parent(prince_philip,princess_anne).

parent(queen,prince_edward).
parent(prince_philip,prince_edward).
```

…etc

- **male**(*Person*)
  - *Person* is male
  - Implement:

```
/*------------ MALE -------------*/

male(prince_philip).
male(prince_charles).
male(prince_william).
male(prince_harry).
male(prince_george).
male(prince_louis).
male(archie_harrison).
male(prince_andrew).
male(mark_phillips).
male(timothy_laurence).
male(peter_phillips).
male(mike_tindall).
male(prince_edward).
male(james_viscount_severn).
```

- **female**(*Person)*
  - *Person* is female
  - Implement:

```
/*------------ FEMALE -------------*/
female(queen).
female(diana).
female(camilla_parker_bowles).
female(kate_middleton).
female(meghan_markle).
female(princess_charlotte).
female(sarah_ferguson).
female(princess_eugenie).
female(princess_beatrice).
female(princess_beatrice).
female(princess_anne).
female(autumn_phillips).
female(zara_tindall).
female(sophie_rhys_jones).
female(lady_louise_windsor).
```

- **married**(*Person*, *Person*)
  - *Person* as argument 1 and *Person* as argument 2 was get married
  - Implement:

```
/*------------ MARRIED -------------*/

married(queen,prince_philip).
married(prince_philip,queen).

married(diana,prince_charles).
married(prince_charles,diana).

married(prince_charles,camilla_parker_bowles).
married(camilla_parker_bowles,prince_charles).

married(kate_middleton,prince_william).
married(prince_william,kate_middleton).

married(prince_harry,meghan_markle).
married(meghan_markle,prince_harry).

married(prince_andrew,sarah_ferguson).
married(sarah_ferguson,prince_andrew).

married(mark_phillips,princess_anne).
married(princess_anne,mark_phillips).

married(princess_anne,timothy_laurence).
married(timothy_laurence,princess_anne).

married(peter_phillips,autumn_phillips).
married(autumn_phillips,peter_phillips).

married(zara_tindall,mike_tindall).
married(mike_tindall,zara_tindall).

married(prince_edward,sophie_rhys_jones).
married(sophie_rhys_jones,prince_edward).
```

- **divorced**(*Person*, *Person*)
  - *Person* as argument 1 and *Person* as argument 2 was divorced.
  - Implement:

```
/*------------ DIVORCED ------------*/

divorced(diana,prince_charles).
divorced(prince_charles,diana).

divorced(mark_phillips,princess_anne).
divorced(princess_anne,mark_phillips).
```

2.3.Define predicates

Predicates	Meaning
father(*Parent*,*Child*):- **male**(*Parent*), **parent**(*Parent*,*Child*).	*Parent* is **father** of *Child* if predicate **parent**(*Parent*,*Child*) is true and *Parent* is male. • **mother**(*Parent*,*Child*) is similarly while *Parent* is female
child(*Child*,*Parent*):- **parent**(*Parent*,*Child*).	*Child* is **child** of *Parent* when **parent**(*Parent*,*Child*) is true • **son**(*Child*,*Parent*) is a child with checking *Child* is male • **daugther**(*Child*,*Parent*) is similarly while *Child* is female
grandparent(*GP*,*GC*):- **parent**(*GP*,*Parent*), **parent**(*Parent*,*GC*).	*GP* is **grandparent** of *GC* when **child** of *GP* is **parent** (Father or Mother) of *GC* • **grandmother**(*GM*,*GC*) is grandparent with checking *GM* is female • **grandfather**(*GF*,*GC*) is similarly while *GF* is male
grandchild(*GC*,*GP*):- **child**(*GC*,*Parent*), **child**(*Parent*,*GP*).	*GC* is **grandchild** of *GP* when **parent** (Father or Mother) of *GC* is **child** of *GP* • **grandson**(*GS*,*GP*) is grandchild with check *GS* is male • **granddaughter**(*GD*,*GP*) is similarly while *GD* is female
spouse(*Husband*,*Wife*):- **male**(*Husband*), **female**(*Wife*), **married**(*Husband*,*Wife*), not(**divorced**(*Husband*,*Wife*)).	*Husband* and *Wife* are spoused when *Husband* is male, Wife is female, **married**(*Husband*,*Wife*) is true and *Husband* and *Wife* are not divorced.
husband(*Person*,*Wife*):- **female**(*Wife*), **married**(*Person*,*Wife*), not(**divorced**(*Person*,Wife)).	*Person* is husband of *Wife* when *Wife* is female, *Person* and *Wife* was married and not divorced.
wife(*Person*,*Husband*):- **male**(*Husband*), **married**(*Person*,*Husband*), not(**divorced**(*Person*,*Husband*)).	*Person* is wife of *Husband* when *Husband* is male, *Person* and *Husband* was married and not divorced.

sibling(*Person1*,*Person2*):- **father**(*Parent*,*Person1*), **father**(*Parent*,*Person2*), *Person1*\=*Person2*.	*Person1* and *Person2* are sibling when both them have the same father and *Person1* and *Person2* are different when query to find.
brother(*Person*,*Sibling*):- **male**(*Person*), **sibling**(*Person*,*Sibling*).	*Person* is brother of *Sibling* when *Person* is male and both of them are sibling.
sister(*Person*,*Sibling*):- **female**(*Person*), **sibling**(*Person*,*Sibling*).	*Person* is sister of *Sibling* when *Person* is female and both of them are sibling.
aunt(*Aunt*,*Child*):- **female**(*Aunt*), **parent**(*X*,*Child*), **sibling**(*X*,*Aunt*).	*Aunt* is aunt of *Child* when *Aunt* is female and the parent of *Child* and *Aunt* are sibling • **uncle**(*Uncle*,*Child*) is similarly with **aunt**(*Aunt*,*Child*) while *Unce* is male
nephew(*Child*,*Sibling*):- **male**(*Child*), **parent**(*Parent*,*Child*), **sibling**(*Parent*,*Sibling*).	*Child* is nephew of *Sibling* when *Child* is male, the parent of *Child* and Silbing are sibling • **niece**(*Child*,*Sibling*) is similarly with **nephew**(*Child*,*Sibling*) while *Child* is female
firstCousin(Person,*Child*):- **parent**(*Parent_Person*,*Person*), **parent**(Parent_*Child*,Child), **sibling**(*Parent*_Person,*Parent_Child*).	*Person* and *Child* is firstCousin when the parent of *Person* and *Child* is sibling

2.4. Test case

1. Check if Prince Philip is father of Prince Edward

?- father(prince_philip,prince_edward).

```
?- father(prince_philip,prince_edward).
true.
```

2. Check if Kate MiddleTon is mother of Archie Harrison

?- mother(kate_middleton,archie_harrison).

```
?- mother(kate_middleton,archie_harrison).
false.
```

3. Princess Eugenie is child of whom?

?- child(princess_eugenie,X).

```
?- child(princess_eugenie,X).
X = prince_andrew ;
X = sarah_ferguson.
```

4. Who is son of Prince William?

?- son(X,prince_william).

```
?- son(X,prince_william).
X = prince_george ;
X = prince_louis.
```

5. Check if Prince Charles is daughter of Queen

?- daughter(prince_charles,queen).

```
?- daughter(prince_charles,queen).
false.
```

6. Prince Philip is grand parent of whom?

```
?- grandparent(prince_philip,X).
X = prince_william ;
X = prince_harry ;
X = princess_eugenie ;
X = princess_beatrice ;
X = peter_phillips ;
X = zara_tindall ;
X = lady_louise_windsor ;
X = james_viscount_severn.
```

7. Who is grand mother of Archie Harrison?

?- grandmother(X,archie_harrison).

```
?- grandmother(X,archie_harrison).
X = diana ;
false.
```

8. Check if Prince Edward is grand father of Lady Louise Windsor

?- grandfather(prince_edward,lady_louise_windsor).

```
?- grandfather(prince_edward,lady_louise_windsor).
false.
```

9. Check if Princess Beatrice is grand child of Queen

?- grandchild(princess_beatrice,queen).

```
?- grandchild(princess_beatrice,queen).
true .
```

10. Who is grand son of Prince Philip?

?- grandson(X,prince_philip).

```
?- grandson(X,prince_philip).
X = prince_william ;
X = prince_harry ;
X = peter_phillips ;
X = james_viscount_severn ;
false.
```

11. Zara Tindall is grand daughter of whom?

?- granddaughter(zara_tindall,X).

```
?- granddaughter(zara_tindall,X).
X = queen ;
X = prince_philip.
```

12. Prince Andrew is spouse of whom?

?- spouse(prince_andrew,X).

```
?- spouse(prince_andrew,X).
X = sarah_ferguson ;
false.
```

13. Check if Prince Charles is husband of Diana?

?- husband(prince_charles,diana).

```
?- husband(prince_charles,diana).
false.
```

14. Check if Camilla Parker is wife of Prince Charles?

?- wife(camilla_parker_bowles,prince_charles).

```
?- wife(camilla_parker_bowles,prince_charles).
true.
```

15. Prince Andrew is sibling of whom?

?- sibling(prince_andrew,X).

```
?- sibling(prince_andrew,X).
X = prince_charles ;
X = princess_anne ;
X = prince_edward ;
false.
```

16. who is brother of prince andrew? (charles, edward)

?- brother(X,prince_andrew).

```
?- brother(X,prince_andrew).
X = prince_charles ;
X = prince_edward ;
false.
```

17.check if prince charles is sister of princess anne (false)

?- sister(prince_charles,princess_anne).

```
?- sister(prince_charles,princess_anne).
false.
```

18. princess anne is aunt of whom? (william, harry, eugenie, beatrice,lady, james)

?- aunt(princess_anne,X).

```
?- aunt(princess_anne,X).
X = prince_william ;
X = prince_harry ;
X = princess_eugenie ;
X = princess_beatrice ;
X = lady_louise_windsor ;
X = james_viscount_severn ;
false.
```

19. who is uncle of james viscount? (andrew, charles)

?- uncle(X,james_viscount_severn).

```
?- uncle(X,james_viscount_severn).
X = prince_charles ;
X = prince_andrew ;
false.
```

20. check if harry is newphew of edward? Yes

?- nephew(prince_harry,prince_edward).

```
?- nephew(prince_harry,prince_edward).
true ;
false.
```

21. check if harry is niece of edward? No

?- niece(prince_harry,prince_edward).

```
?- niece(prince_harry,prince_edward).
false.
```

22. find X is neice of Y (use niece(x,y) to query)

?- niece(X,Y).

```
?- niece(X,Y).
X = princess_charlotte,
Y = prince_harry ;
X = princess_eugenie,
Y = prince_charles ;
X = princess_eugenie,
Y = princess_anne ;
X = princess_eugenie,
Y = prince_edward ;
X = princess_beatrice,
Y = prince_charles ;
X = princess_beatrice,
Y = princess_anne ;
X = princess_beatrice,
Y = prince_edward ;
X = princess_beatrice,
Y = prince_charles ;
X = princess_beatrice,
Y = princess_anne ;
X = princess_beatrice,
Y = prince_edward ;
X = zara_tindall,
Y = prince_charles ;
X = zara_tindall,
Y = prince_andrew ;
X = zara_tindall,
Y = prince_edward ;
X = lady_louise_windsor
Y = prince_charles ;
X = lady_louise_windsor
Y = prince_andrew ;
X = lady_louise_windsor
Y = princess_anne ;
false.
```

23. check if beatric is first cousin with george? No

?- firstCousin(princess_beatrice,prince_george).

```
?- firstCousin(princess_beatrice,prince_george).
false.
```

24. who is the first cousin with beatric? (william, harry, peter, zara, lady, james)

?- firstCousin(X,princess_beatrice).

```
?- firstCousin(X,princess_beatrice).
X = prince_william ;
X = prince_harry ;
X = peter_phillips ;
X = zara_tindall ;
X = lady_louise_windsor ;
X = james_viscount_severn ;
false.
```

25. find all X is the first cousin of Y (firstCousin(X,Y))

?- firstCousin(X,Y).

```
?- firstCousin(X,Y).
X = prince_william,
Y = princess_eugenie ;
X = prince_william,
Y = princess_beatrice ;
X = prince_william,
Y = peter_phillips ;
X = prince_william,
Y = zara_tindall ;
X = prince_william,
Y = lady_louise_windsor ;
X = prince_william,
Y = james_viscount_severn ;
X = prince_harry,
Y = princess_eugenie ;
X = prince_harry,
Y = princess_beatrice ;
X = prince_harry,
Y = peter_phillips ;
X = prince_harry,
Y = zara_tindall ;
X = prince_harry,
Y = lady_louise_windsor ;
X = prince_harry,
Y = james_viscount_severn ;
X = prince_george,
Y = archie_harrison ;
X = princess_charlotte,
Y = archie_harrison ;
X = prince_louis,
Y = archie_harrison ;
X = archie_harrison,
Y = prince_george ;
X = archie_harrison,
Y = princess_charlotte ;
X = archie_harrison,
Y = prince_louis ;
X = princess_eugenie,
Y = prince_william ;
X = princess_eugenie,
Y = prince_harry ;
X = princess_eugenie,
Y = peter_phillips ;
X = princess_eugenie,
Y = zara_tindall ;
X = princess_eugenie,
Y = lady_louise_windsor ;
X = princess_eugenie,
Y = james_viscount_severn ;
X = princess_beatrice,
Y = prince_william ;
X = princess_beatrice,
Y = prince_harry ;
X = princess_beatrice,
Y = peter_phillips ;
X = princess_beatrice,
Y = zara_tindall ;
X = princess_beatrice,
Y = lady_louise_windsor ;
X = princess_beatrice,
Y = james_viscount_severn ;
X = peter_phillips,
Y = prince_william ;
X = peter_phillips,
Y = prince_harry ;
X = peter_phillips,
Y = princess_eugenie ;
X = peter_phillips,
Y = princess_beatrice ;
X = peter_phillips,
Y = lady_louise_windsor ;
X = peter_phillips,
Y = james_viscount_severn ;
X = zara_tindall,
Y = prince_william ;
X = zara_tindall,
Y = prince_harry ;
X = zara_tindall,
Y = princess_eugenie ;
X = zara_tindall,
Y = princess_beatrice ;
X = zara_tindall,
Y = lady_louise_windsor ;
X = zara_tindall,
Y = james_viscount_severn ;
X = lady_louise_windsor,
Y = prince_william ;
X = lady_louise_windsor,
Y = prince_harry ;
X = lady_louise_windsor,
Y = princess_eugenie ;
X = lady_louise_windsor,
Y = princess_beatrice ;
X = lady_louise_windsor,
Y = peter_phillips ;
X = lady_louise_windsor,
Y = zara_tindall ;
X = james_viscount_severn,
Y = prince_william ;
X = james_viscount_severn,
Y = prince_harry ;
X = james_viscount_severn,
Y = princess_eugenie ;
X = james_viscount_severn,
Y = princess_beatrice ;
X = james_viscount_severn,
Y = peter_phillips ;
X = james_viscount_severn,
Y = zara_tindall ;
false.
```

3. REFERENCE

- Reference in moodle by lecture
- Prolog Tutorial: https://www.youtube.com/watch?v=SykxWpFwMGs