assignment3

Q1:

Map Step:

Plain Text

```
For each input key-value pair (person, (friend1, ..., friendn)):
    For each friend in the friend list:
        output ((person, friend), (friend1, ..., friendn)
/*
For example, if the input is:
(A, (B, C))
(B, (A, C))
(C, (A, B))

Then after mapping, the output would be:
((A, B), (B, C))
((A, C), (B, C))
((B, A), (A, C))
((B, C), (A, C))
((C, A), (A, B))
((C, B), (A, B))
*/
```

Group and Shuffle Step:

Transform the output from the last step into key-value-sequence pairs.

Plain Text

```
/*
(suppose each pair isn't ordered (i.e. (A, B)=(B, A)))

For example, if the input is the output from the last step, then the output in this step would become:
((A, B), ((B, C), (A, C)))
((A, C), ((B, C), (A, B)))
((B, C), ((A, C), (A, B)))
*/
```

Reduce Step:

Plain Text

```
For each key-value-sequence pair:
    result=[];
    For each item in the value sequence:
        For each person in this item:
            if (!result.contains(person) && !key-sequence.contains(person)):
                result.add(person)
output (key-sequence, result)

/*
For example, if the input is the output from the last step, then the output in this step would become:
((A, B), (C))
((A, C), (B))
((B, C), (A))
*/
```

Q2:

Select Step:

Map Step:

Plain Text

```
For each tuple(Hname, Province) of Hospital:
    output (Hname, Province)
For each tuple(HInsurNum, age, Hname) of Patient:
    if age>60:
        output (HInsurNum, age, Hname)

/*
For example, if the input is:
(H1, Quebec)
(H2, BC)
(HIN001, 61, H1)
(HIN002, 20, H1)
(HIN003, 70, H2)
(HIN004, 30, H2)
(HIN005, 69, H1)

Then after mapping, the output would be:
(H1, Quebec)
(H2, BC)
(HIN001, 61, H1)
(HIN003, 70, H2)
(HIN005, 69, H1)
*/
```

Reduce Step:

Plain Text

```
For each input:
    output (input)
```

Join step:

Map Step:

Plain Text

```
For each tuple(Hname, Province) of Hospital:
    output (Hname, ("Hospotal", (Province)))
For each tuple(HInsurNum, age, Hname) of Patient:
    output (Hname, ("Patient", (HInsurNum, age)))

/*
For example, if the input is the output from the last step, then the output in this step would become:
(H1, ("Hospital", (Quebec)))
(H2, ("Hospital", (BC)))
(H1, ("Patient", (HIN001, 61)))
(H2, ("Patient", (HIN003, 70)))
(H1, ("Patient", (HIN005, 69)))
*/

```

Group and Shuffle Step:

Group and shuffle will aggregate all key/value pairs with the same Hname.

Plain Text

```
/*
For example, if the input is the output from the last step, then the output in this step would become:
(H1, (("Hospital", (Quebec)), ("Patient", (HIN001, 61)), ("Patient", (HIN005, 69))))
(H2, (("Hospital", (BC)), ("Patient", (HIN003, 70))))
*/
```

Reduce Step:

Plain Text

```
For each key-value-sequence pair (key, value-sequence):
    HospitalTable=[];
    PatientTable=[];
    For each v=(rel, tuple) in the value sequence:
        if v.rel == "Hospital":
            HospitalTable.add(tuple)
        else:
            PatientTable.add(tuple)
    For tuple1 in HospitalTable:
        For tuple2 in PatientTable:
            output (key, (tuple1[0], tuple2[0], tuple2[1]))

/*
For example, if the input is the output from the last step, then the output in this step would become:
(H1, (Quebec, HIN001, 61))
(H1, (Quebec, HIN005, 69))
(H2, (BC, HIN003, 70))
*/
```

Projection Step:

Map Step:

Plain Text

```
For each key-value-sequence pair (Hname, (Province, HInsurNum, age)):
    output (Province, (HInsurNum))

/*
For example, if the input is the output from the last step, then the output in this step would become:
(Quebec, (HIN001))
(Quebec, (HIN005))
(BC, (HIN003))
*/
```

Reduce Step:

Plain Text

```
For each input:
    output (input)
```

Aggregation Step:

Map Step:

Plain Text

```
For each tuple(Province, (HInsurNum)):
    output (Province, HInsurNum)

/*
For example, if the input is the output from the last step, then the output in this step would become:
(Quebec, HIN001)
(Quebec, HIN005)
(BC, HIN003)
*/
```

Group and Shuffle Step:

Group and shuffle will aggregate all key/value pairs with the same Province.

Plain Text

```
/*
For example, if the input is the output from the last step, then the output in this step would become:
(Quebec, (HIN001, HIN005))
(BC, (HIN003))
*/
```

Reduce Step:

Plain Text

```
For each tuple (Province, (HInsurNum)):
    i=0
    For each item in HInsurNum:
        i+=1
    output (Province, i)

/*
For example, if the input is the output from the last step, then the output in this step would become:
(Quebec, 2)
(BC, 1)
*/
```

Q5:

Plain Text

```
grpd: {
    group: chararray,
    coviddata: {
        (
            prname: chararray,
            idate: chararray,
            newcases: int,
            newdeaths: int,
            tests: int,
            recoveries: int
        )
    }
}
```

Q6:

Plain Text

```
joined: {
    QuebecData::prname: chararray,
    QuebecData::idate: chararray,
    QuebecData::newcases: int,
    QuebecData::newdeaths: int,
    QuebecData::tests: int,
    QuebecData::recoveries: int,
    QuebecDeaths::prname: chararray,
    QuebecDeaths::totaldeaths: long
}
```