Assignment 3

Student Name: Mert Gurkan
Student ID: 260716883

Ex 1 - Using MapReduce in Social Media

Datasets

Relation(Person, List of Friends)

R(Joe, [Abe, Jane, Ali, Zack])
R(Ali, [Joe, Jane, Zack, Mary])

a) Design MapReduce workflow

Assume R(Person, [List of Friends]) relation (no duplicates)

Map

For each tuple of R, output ((Person, Friend), [List of Friends]) where Person and Friend are the new key sorted

(Abe, Joe) -> [Abe, Jane, Ali, Zack]
(Ali, Joe) -> [Abe, Jane, Ali, Zack]
(Jane, Joe) -> [Abe, Jane, Ali, Zack]
(Joe, Zack) -> [Abe, Jane, Ali, Zack]

(Ali, Joe) -> [Joe, Jane, Zack, Mary]
(Ali, Jane) -> [Joe, Jane, Zack, Mary]
(Ali, Zack) -> [Joe, Jane, Zack, Mary]
(Ali, Mary) -> [Joe, Jane, Zack, Mary]

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

(Abe, Joe) -> [[Abe, Jane, Ali, Zack], []]
(Ali, Joe) -> [[Abe, Jane, Ali, Zack], [Joe, Jane, Zack, Mary]]
(Jane, Joe) -> [[Abe, Jane, Ali, Zack], []]
(Joe, Zack) -> [[Abe, Jane, Ali, Zack], []]
(Ali, Jane) -> [[Joe, Jane, Zack, Mary], []]
(Ali, Zack) -> [[Joe, Jane, Zack, Mary], []]
(Ali, Mary) -> [[Joe, Jane, Zack, Mary], []]

Reduce

For each tuple intersect the lists of values, output ((Person 1, Person 2)), [Intersect List]) if intersect list is not empty

(Ali, Joe) -> [Jane, Zack]

b) After the MapReduce execution when Ali visit Joe and vice versa, we can quickly lookup by the key (Ali, Joe) to find the friends in common

c) No, it doesn't store practically duplicated data because we kept only the intersect list for each pair of persons.

d) Samples can be found along the workflow.

Ex 2 - Using MapReduce and Relational Operations

Datasets

Hospital(Hname, Province)

H(Toronto General Hospital, ON)
H(Toronto City Hospital, ON)
H(Jewish General Hospital, QC)

Patient(HInsurNum, age, Hname) Hname references Hospital

P(1, 62, Toronto General Hospital)
P(2, 63, Toronto City Hospital)
P(3, 61, Toronto General Hospital)
P(4, 59, Toronto General Hospital)
P(5, 29, Jewish General Hospital)
P(6, 34, Toronto City Hospital)
P(7, 65, Jewish General Hospital)

--

Assume P(HInsurNum, age, Hname) relation (no duplicates)

Map

For each tuple of P for which condition "age > 60", output (Hname, HInsurNum)

(Toronto General Hospital, 1)
(Toronto City Hospital, 2)
(Toronto General Hospital, 3)
(Jewish General Hospital, 7)

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

Reduce

For each tuple (Hname, HInsurNum) group by Hname, output (Hname, Patients)

(Toronto General Hospital, 2)
(Toronto City Hospital, 1)
(Jewish General Hospital, 1)

--

Assume R(Hname, Patients)
Assume H(Hname, Province)

Natural Join R(Hname, Patients) with H(Hname, Province)

Map

For each tuple (Hname, Patients) of R, output (Hname, Patients)

(Toronto General Hospital, 2)
(Toronto City Hospital, 1)
(Jewish General Hospital, 1)

For each tuple (Hname, Province) of H, output (Hname, Province)

(Toronto General Hospital, ON)
(Toronto City Hospital, ON)
(Jewish General Hospital, QC)

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

Reduce

For each tuple (key, value-list)
Rt = Ht = empty
for each v = (rel, tuple) in value-list
if v.rel = R insert tuple into Rt else insert tuple into Ht
for v1 in Rt, for v2 in Ht, output (key, v1, v2)

It produces all combinations of Hospital, Patients and Provinces based on Hospital name, which is the key

(Toronto General Hospital, 2, ON)

(Toronto City Hospital, 1, ON)
(Jewish General Hospital, 1, QC)

--

Assume O(Hname, Patients, Province)

Map

For each tuple (Hname, Patients, Province) of O, output (Province, Patients)

(ON, 2)
(ON, 1)
(QC, 1)

Reduce

For each tuple (Province, Patients) group by Province, output (Province, SumPatients)

(ON, 3)
(QC, 1)

--

Assume X(Province, SumPatients)

Map

For each tuple of X for which condition "SumPatients > 100", output (Province, SumPatients)

Reduce

Identity, that is, output (Province, SumPatients)

Ex 4 - PigLatin - Covid Recovery Info

```
2021-04-07 16:30:50,068 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(2020-03-26,66,186,2.8181818181818183)
(2020-04-05,408,1219,2.9877450980392157)
(2020-04-18,165,1162,7.042424242424242)
(2020-04-18,723,3315,4.5850622406639)
(2020-04-18,485,4875,10.051546391752577)
(2020-05-04,70,229,3.2714285714285714)
(2020-05-05,57,277,4.859649122807017)
(2020-05-06,70,333,4.757142857142857)
(2020-05-07,54,257,4.7592592592592595)
(2020-05-08,81,211,2.6049382716049383)
(2020-05-09,59,184,3.1186440677966103)
(2020-05-13,62,210,3.3870967741935485)
(2020-05-14,50,129,2.58)
(2020-05-21,52,126,2.423076923076923)
(2020-06-06,226,648,2.8672566371681416)
(2020-06-09,138,367,2.6594202898550723)
(2020-06-10,156,480,3.076923076923077)
(2020-06-10,251,551,2.195219123505976)
(2020-06-11,203,505,2.4876847290640396)
(2020-06-11,144,458,3.1805555555555554)
(2020-06-12,181,524,2.8950276243093924)
(2020-06-13,158,458,2.8987341772151898)
(2020-06-14,128,461,3.6015625)
(2020-06-14,197,423,2.1472081218274113)
(2020-06-15,102,471,4.617647058823529)
(2020-06-22,69,184,2.6666666666666665)
(2020-06-26,111,226,2.036036036036036)
(2020-06-29,311,816,2.6237942122186495)
(2020-06-30,68,196,2.8823529411764706)
(2020-07-11,130,267,2.0538461538461537)
(2020-07-17,141,23686,167.98581560283688)
(2020-07-29,76,174,2.289473684210526)
(2020-08-10,156,2155,13.814102564102564)
(2020-08-16,67,138,2.0597014925373136)
(2020-08-19,64,155,2.421875)
(2020-09-09,98,204,2.0816326530612246)
(2020-09-22,96,797,8.302083333333334)
(2020-11-03,570,1315,2.307017543859649)
(2020-11-23,594,1592,2.68013468013468)
(2020-11-30,596,1807,3.0318791946308723)
(2020-12-07,647,1629,2.517774343122102)
(2020-12-07,325,4067,12.513846153846155)

(2020-12-07,325,4067,12.513846153846155)
(2020-12-12,274,713,2.602189781021898)
(2020-12-14,759,1609,2.1198945981554678)
(2020-12-21,529,1866,3.5274102079395084)
(2020-12-22,155,1491,9.619354838709677)
(2020-12-24,154,654,4.246753246753247)
(2020-12-27,2291,5962,2.6023570493234396)
(2020-12-27,198,500,2.525252525252525)
(2020-12-28,917,6710,7.317339149400218)
(2020-12-29,114,405,3.5526315789473686)
(2020-12-29,382,3391,8.87696335078534)
(2020-12-30,138,377,2.7318840579710146)
(2020-12-31,190,439,2.3105263157894735)
(2021-01-02,210,479,2.280952380952381)
(2021-01-03,2869,7409,2.58243290345068)
(2021-01-04,539,3102,5.755102040816326)
(2021-01-04,1128,5727,5.077127659574468)
(2021-01-11,430,2336,5.432558139534883)
(2021-01-11,131,1445,11.030534351145038)
(2021-01-12,652,1311,2.0107361963190185)
(2021-01-12,89,397,4.46067415730337)
(2021-01-13,156,322,2.0641025641025643)
(2021-01-18,301,1541,5.119601328903655)
(2021-01-19,456,1267,2.7785087719298245)
(2021-01-20,241,694,2.879668049792531)
(2021-01-21,226,816,3.6106194690265485)
(2021-01-25,346,1376,3.976878612716763)
(2021-01-26,230,839,3.6478260869565218)
(2021-01-26,366,1040,2.841530054644809)
(2021-02-01,147,310,2.108843537414966)
(2021-02-01,277,1566,5.653429602888087)
(2021-02-02,268,732,2.7313432835820897)
(2021-02-02,745,2297,3.083221476510067)
(2021-02-03,259,561,2.166023166023166)
(2021-02-08,343,3038,8.857142857142858)
(2021-02-08,52,1682,32.34615384615385)
(2021-02-09,195,548,2.81025641025641)
(2021-02-09,80,253,3.1625)
(2021-02-16,136,529,3.889705882352941)
(2021-02-17,427,2159,5.056206088992974)
(2021-02-18,137,538,3.927007299270073)
(2021-02-22,449,1343,2.9910913140311806)
(2021-02-24,57,159,2.789473684210526)
(2021-03-01,475,1588,3.343157894736842)
(2021-03-08,278,592,2.129496402877698)
(2021-03-08,385,1485,3.857142857142857)
(2021-03-12,96,429,4.46875)
(2021-03-15,460,1565,3.402173913043478)
2021-04-07 16:30:50,202 [main] INFO  org.apache.pig.Main - Pig script completed in 1 minute, 9 seconds and 690 milliseconds (69690 ms)
```

Ex 5 - PigLatin - Covid Mortality

c - Schema following GROUP Operation

```
datacluster: {
   group: chararray,
   deathperprovince: {
      (
         prname: chararray,
         newdeaths: int
      )
   }
}
```

d -

```
2021-04-07 16:36:29,291 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(Quebec,10558)
(Ontario,7175)
(Alberta,1956)
(British Columbia,1407)
(Manitoba,917)
(Saskatchewan,409)
2021-04-07 16:36:29,420 [main] INFO  org.apache.pig.Main - Pig script completed in 1 minute, 25 seconds and 565 milliseconds (85565 ms)
```

Ex 6 - PigLatin - Covid Mortality Rate in Quebec

c - Schema following JOIN Operation

DESCRIBE combineddata;

```
combineddata: {
   dataQuebec::prname: chararray,
   dataQuebec::idate: chararray,
   dataQuebec::newdeaths: int,
   dataagg::prname: chararray,
   dataagg::total_deaths: long
}
```

d -

```
2021-04-07 16:39:20,437 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(2020-05-02,114,0.010797499526425459)
(2020-05-06,112,0.01060806971017238)
(2020-05-14,131,0.012407652964576625)
(2020-05-07,121,0.011460503883311234)
(2020-05-10,142,0.013449516953968555)
(2020-05-31,202,0.0191324114415609)
(2020-04-16,143,0.013544231862095094)
(2020-04-23,109,0.010323924985792763)
(2020-05-05,118,0.011176359158931616)
(2020-04-25,106,0.010039780261413146)
(2020-04-18,117,0.011081644250805076)
(2020-05-12,118,0.011176359158931616)
(2020-05-01,163,0.015438530024625877)
2021-04-07 16:39:20,526 [main] INFO  org.apache.pig.Main - Pig script completed in 58 seconds and 319 milliseconds (58319 ms)
```