Part 1 – ArangoDB (4 points)

Solution

```
FOR v, e, p IN 1..2 OUTBOUND
'airports/SEA' flights
OPTIONS {
uniqueVertices: 'none',
uniqueEdges: 'path'
}
FILTER v.city == "Dallas-Fort Worth"
    AND p.edges[0].DepTime > 2150
    AND p.edges[0].DayOfWeek == 4
    AND p.edges[0].UniqueCarrier == "AS"
    AND sum(p.edges[*].Distance) < 2000
SORT sum(p.edges[*].Distance)
RETURN distinct (CONCAT_SEPARATOR('->', p.vertices[*].city,
            p.edges[*].UniqueCarrier,
            "Flight No",
            p.edges[*].FlightNum,
            "Departure Time",
            p.edges[*].DepTime,
            "Distance",
            p.edges[*].Distance,
            "Total",
            sum(p.edges[*].Distance) ))
```

Part 2 – Cosmos DB Gremlin API (4 points)

Solution

1)

Create Vertex

```
g.addV('Person').property('EmployeeID', 2).property('LastName', 'Fuller').property('FirstName',
'Andrew').Property('Department', 'Null')
```

```
g.addV('Person').property('EmployeeID', 3).property('LastName', 'Leverling').property('FirstName', 'Janet').Property('Department', 'IT');

g.addV('Person').property('EmployeeID', 5).property('LastName', 'Buchanan').property('FirstName', 'Steven').Property('Department', 'Finance')

g.addV('Person').property('EmployeeID', 7).property('LastName', 'King').property('FirstName', 'Robert').Property('Department', 'Finance')

g.addV('Person').property('EmployeeID', 12).property('LastName', 'Chang').property('FirstName', 'Leslie').Property('Department', 'Finance')

g.addV('Person').property('EmployeeID', 14).property('LastName', 'Ng').property('FirstName', 'Jordan').Property('Department', 'Finance')

g.addV('Person').property('EmployeeID', 15).property('LastName', 'Black').property('FirstName', 'Lela').Property('Department', 'IT')

g.addV('Person').property('EmployeeID', 21).property('LastName', 'Thompson').property('FirstName', 'Connie').Property('Department', 'IT')
```

Create Edge

```
g.V().hasLabel('Person').has('EmployeeID', 3).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 2))

g.V().hasLabel('Person').has('EmployeeID', 5).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 2))

g.V().hasLabel('Person').has('EmployeeID', 7).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 2))

g.V().hasLabel('Person').has('EmployeeID', 15).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 2))

g.V().hasLabel('Person').has('EmployeeID', 7).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 3))

g.V().hasLabel('Person').has('EmployeeID', 7).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 5))

g.V().hasLabel('Person').has('EmployeeID', 12).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 7))

g.V().hasLabel('Person').has('EmployeeID', 14).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 7))

g.V().hasLabel('Person').has('EmployeeID', 15).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 7))
```

g.V().hasLabel('Person').has('EmployeeID',
21).addE('ReportsTo').to(g.V().hasLabel('Person').has('EmployeeID', 15))

2)

g.V().hasLabel('Person').has('department','IT')

3)

g.V().hasLabel('Person').has('EmployeeID',2).repeat(bothE('ReportsTo').outV().dedup()).times(4).values('EmployeeID')

[OR]

g.V().hasLabel('Person').has('EmployeeID',2).union(inE('ReportsTo').outV(),inE('ReportsTo').outV().inE('ReportsTo').outV()).hasLabel('Person').values('EmployeeID')