**Part 1 – Mountain trips**

1. Shortest (least stages) routes from Darjeeling to Sandakphu that can be used in winter

MATCH (from:town { name:"Darjeeling" }), (to:peak { name: "Sandakphu"}) , path = (from)-[\*{winter:"true"}]->(to)

RETURN path AS shortestPath

LIMIT 1

1. All routes you can use to get from Darjeeling to Sandakphu, sorted by the distance you need to travel

MATCH (from:town { name:"Darjeeling" }), (to:peak { name: "Sandakphu"}) , path = (from)-[\*]->(to)

RETURN path AS shortestPath,

    reduce(distance = 0, r in relationships(path) | distance+r.distance) AS totalDistance

    ORDER BY totalDistance ASC

1. All places you can get from Darjeeling using a twowheeler in the summer

MATCH (from:town { name:"Darjeeling" }), path = (from)-[:twowheeler {summer:"true"}]->()

RETURN path AS shortestPath

* MATCH p=()-[r: twowheeler]->() RETURN p

**Part 2 – Airlines**

1. List of all airports sorted by number of flights starting in them

MATCH (f:Flight), (a:Airport) , path = (f)-[r:ORIGIN]->(a)

RETURN  a.name AS Airport, count(a.name) AS FlightCounter

1. List of all airports you can reach (directly or indirectly) from LAX spending less than 3000

MATCH (from:Airport {name:"LAX"}), path = (from)<-[:ORIGIN]-(flight)-[:DESTINATION\*]->(dest:Airport)

WHERE dest.name <> "LAX"

WITH  flight, path

MATCH (flight)-[:ASSIGN]-(t:Ticket)

WITH path AS shortestPath, reduce(price = 0, r in relationships(path) | price+t.price) AS totalPrice

WHERE totalPrice < 3000

RETURN shortestPath, totalPrice

ORDER BY totalPrice ASC

1. Sort connections you can use to go from LAX to DAY by connection price

NIE MA POŁĄCZENIA LAX-DAY –> PLIKI WYNIKOWE BYŁY PUSTE

WYNIKI DO ZADAŃ 6 -8 SĄ Z ZAPYTANIA O POŁĄCZENIA DO ABQ

MATCH (from:Airport {name:"LAX"}), (to:Airport { name:"DAY"}),

    path = (from)<-[:ORIGIN]-(flight:Flight)-[:DESTINATION\*]->(to)

WITH path, flight as f

MATCH (flight)-[:ASSIGN]-(t:Ticket)

RETURN path AS shortestPath,

    reduce(price = 0, r in relationships(path) | price+t.price) AS totalPrice

ORDER BY totalPrice ASC

1. Cheapest connection from LAX to DAY

MATCH path = (from:Airport {name:"LAX"})<-[:ORIGIN]-(flight:Flight)-[:DESTINATION\*]->(to:Airport { name:"ABQ"})

WITH path, flight as f

MATCH (flight)-[:ASSIGN]-(t:Ticket)

WITH path AS shortestPath,

    reduce(price = 0, r in relationships(path) | price+t.price) AS totalPrice

RETURN  min(totalPrice)

1. Cheapest connection from LAX to DAY in business class

MATCH path =  (a:Airport {name:"LAX"})-[:ORIGIN]-(f)-[:DESTINATION|:ORIGIN\*1..5]-(aofa)-[]-(f)-[]-(dest:Airport {name:"DAY"})

WITH path, f as flight

MATCH (flight)-[:ASSIGN]-(t:Ticket)

WITH path AS shortestPath,

    reduce(price = 0, r in relationships(path) | price+t.price) AS totalPrice

RETURN  min(totalPrice)

1. List of airlines sorted by the number of cities they offer connections to and from (unique cities taking part in :ORIGIN and :DESTINATION of Flight nodes handled by given airline

1. Cheapest connection between 3 different airports

MATCH (from:Airport), path = (from)<-[:ORIGIN]-(flight)-[:DESTINATION\*1..3]->(dest:Airport)

WITH flight, path

MATCH (flight)-[:ASSIGN]-(t:Ticket)

WITH path AS shortestPath, reduce(price = 0, r in relationships(path) | price+t.price) AS totalPrice

WHERE totalPrice>120

RETURN shortestPath, totalPrice

ORDER BY totalPrice ASC