USE-CASE 1

Sam who works in the financial & Operation services department, based on the user preferences like CarType (Petrol,Diesel,Electric) TripType(Flexible,Planned) and No.Of co- passengers , he needs to sent recommendation on no. of available passengers who you can join along the trip and expected expense and total distance .

Step -1 Loads the data from CSV file

LOAD CSV WITH HEADERS FROM

'file:///MOCK\_DATA.csv' AS line

MERGE (h:HomeCity{CityName:line.HomeCity,State:line.State,Country:line.Country,latitude:line.latitude,longitude:line.longitude})

MERGE (d:DestinationCity{CityName:line.DestinationCity,State:line.DestinationState,

Country:line.DestinationState,latitude:line.DestinationLat,longitude:line.DestinationLong})

MERGE (r:Rate\_Slab {TripType:line.TripType,CarType:line.CarType})

MERGE (u:User {username:line.Username, first\_name: line.first\_name,lastname: line.last\_name,gender: line.Gender,

Homecity: line.HomeCity,TripType:line.TripType,NumOfPasngrs:line.NumOfSeats,

Destin\_City:line.DestinationCity,CarType:line.CarType})

MERGE (c:Car {CarModel: line.CarModel,CarMake: line.CarMake,ModelYear: line.ModelYear,CarType: line.CarType})

Step-2 Create relationship between each user and other nodes

MERGE (u)-[rd:TRAVELS\_TO]->(d)

MERGE (u)<-[rh:TRAVELS\_FROM]-(h)

MERGE (h)-[rt:TRAVEL\_ROUTE {travelDistance:0}]->(d)

MERGE  (r)-[a:Rate{Expected\_Expense:0}]->(u)

MERGE (c)-[o:OWN{is\_owner:'N'}]->(u)

Step-3 Created Indexes

CREATE INDEX ON :User(username);

CREATE INDEX ON :HomeCity(CityName);

CREATE INDEX ON :DestinationCity(CityName);

CREATE INDEX ON :Car(CarModel)

Graphical user interface, text, application, email

Description automatically generated

Step-4 update the base rate for each trip type ,cartype in RATE node

MATCH (n {TripType:"Planned",CarType:"Diesel"})

SET n.BaseRate = 1

RETURN n.CarType, n.TripType,n.BaseRate;

MATCH (n {TripType:"Flexible",CarType:"Petrol"})

SET n.BaseRate = 2

RETURN n.CarType, n.TripType,n.BaseRate;

MATCH (n {TripType:"Flexible",CarType:"Diesel"})

SET n.BaseRate = 3

RETURN n.CarType, n.TripType,n.BaseRate;

MATCH (n {TripType:"Planned",CarType:"Petrol"})

SET n.BaseRate = 4

RETURN n.CarType, n.TripType,n.BaseRate;

MATCH (n {TripType:"Planned",CarType:"Electric"})

SET n.BaseRate = 5

RETURN n.CarType, n.TripType,n.BaseRate;

MATCH (n {TripType:"Flexible",CarType:"Electric"})

SET n.BaseRate = 6

RETURN n.CarType, n.TripType,n.BaseRate;

STEP-5 Update distance between each Homcity and DestinationCity node with the distance

MATCH (n:HomeCity)-[way:TRAVEL\_ROUTE]->(m:DestinationCity)

WITH

point({ longitude: toInteger(n.longitude), latitude: toInteger(n.latitude) }) AS p1,

point({ longitude: toInteger(m.longitude), latitude: toInteger(m.latitude)}) AS p2,

way

SET way.travelDistance = round(distance(p1,p2), 2) // two decimals

RETURN way

STEP-6 update the expected expense for each travel route between homcity and destination based on the distance travel and base\_rate from payment details on each node assigned to users

MATCH (n:HomeCity)-[way:TRAVEL\_ROUTE]->(m:DestinationCity)

OPTIONAL MATCH

 (p:Rate\_Slab)-[t:Rate]->(u:User)

SET t.Expected\_Expense = round((way.travelDistance\*p.BaseRate)/(1000\*toInteger(u.NumOfPasngrs)), 2)

RETURN way.travelDistance,p.BaseRate,t.Expected\_Expense

STEP-7

Fetch no of usesr available in the route with mentioned CarType,tripType and No.of Co-passengers

MATCH (h:HomeCity)-[way:TRAVEL\_ROUTE]->(m:DestinationCity)

OPTIONAL MATCH

 (p:Rate\_Slab)-[t:Rate]->(u:User)

RETURN h.CityName as Homecity,m.CityName as DestinationCity,toInteger(u.NumOfPasngrs) as NumOfPasngrs,

u.TripType as TripType,u.CarType as CarType,round(way.travelDistance/1000,2) as DistanceKM,t.Expected\_Expense as Expected\_Expense,count(\*) as NumOfUsers

Table

Description automatically generated

STEP-8

Fetch no. of carowners available in the given route with mentioned conditions

CarType,tripType and No.of Co-passengers

MATCH (h:User)-[:OWN]-(c:Car)

WHERE h.IS\_OWNER='Y'

RETURN h.Homecity as Homecity,h.Destin\_City as DestinationCity,toInteger(h.NumOfPasngrs) as NumOfPasngrs,

h.TripType as TripType,h.CarType as CarType,count(\*) as NumOfUsers

Table

Description automatically generated