Eunice Worifah

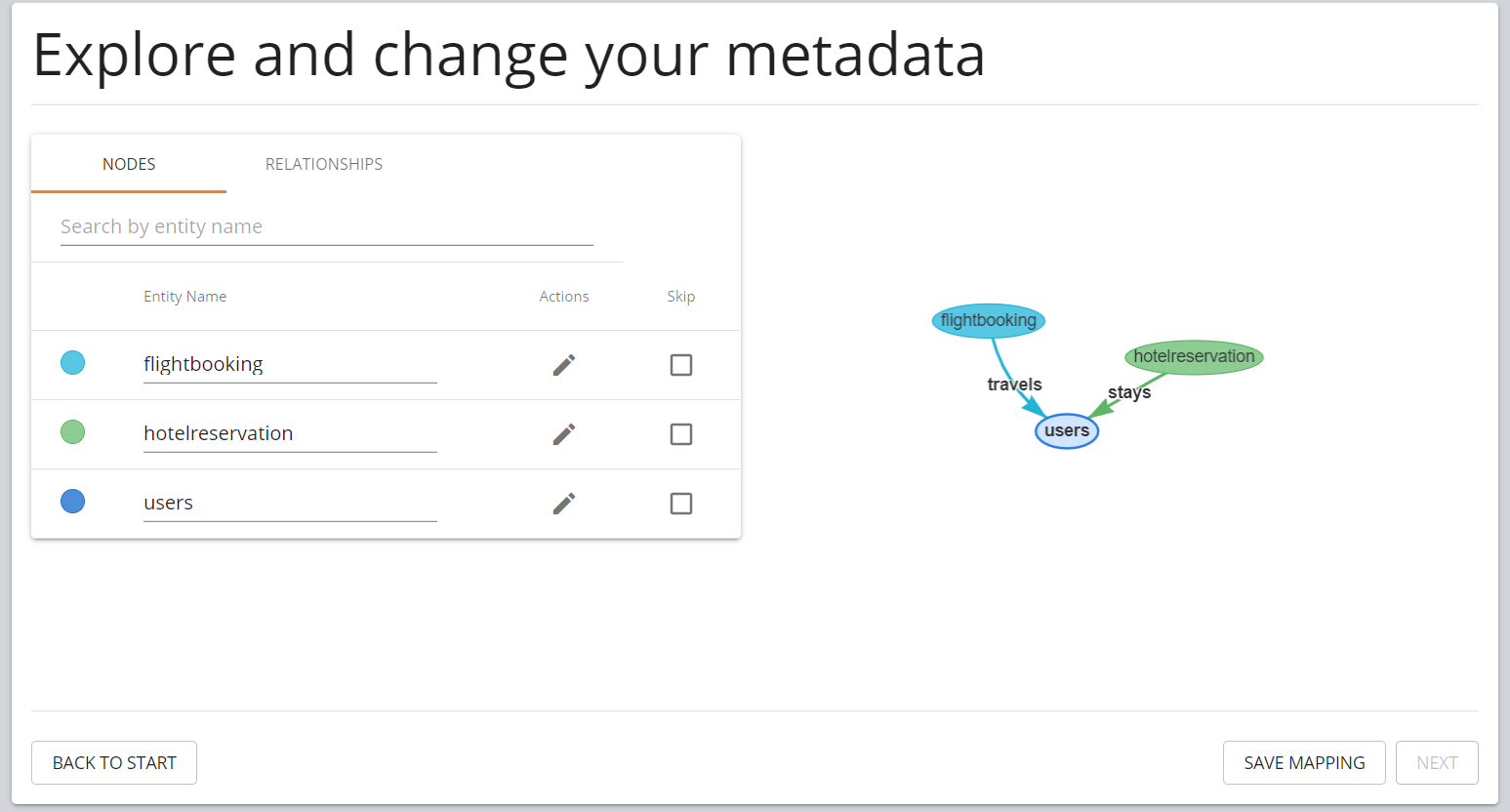
student id: 260560384

MCGILL DESAUTELS

**INSY 661: Individual Project – PART 2 (neo4j)**

**AN ANALYSIS OF CORPORATE TRAVEL IN BRAZIL**

**2019 and 2020**



1. **.**
   1. **Total amount of money spent on business travel over the last 2 years, and the amount of money spent in 2019 vs 2020.**

MATCH (n:flightbooking)

RETURN '2019 & 2020' AS type, sum(n.flightprice)

UNION

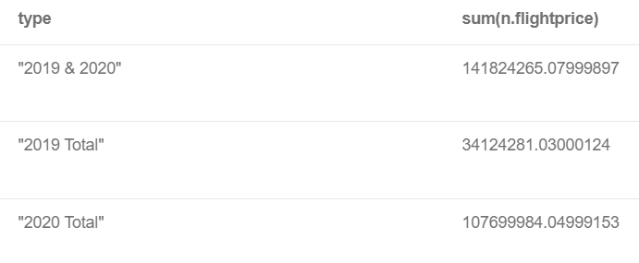
MATCH (n:flightbooking) WHERE n.flightdate CONTAINS '/2019'

RETURN '2019 Total' AS type, sum(n.flightprice)

UNION

MATCH (n:flightbooking) WHERE n.flightdate CONTAINS '/2020'

RETURN '2020 Total' AS type, sum(n.flightprice)



**b. Total amount of money spent on hotel accommodation over the last 2 years, and the amount of money spent in 2019 vs 2020.**

MATCH (n:hotelreservation)

RETURN '2019 & 2020' AS Year, sum(n.totalpaid) AS GrandTotal, sum(n.days) as TotalDays

UNION

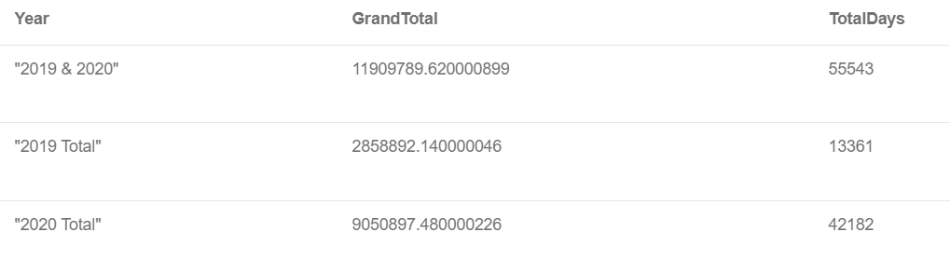
MATCH (n:hotelreservation) WHERE n.checkindate CONTAINS '/2019'

RETURN '2019 Total' AS Year, sum(n.totalpaid) AS GrandTotal, sum(n.days) AS TotalDays

UNION

MATCH (n:hotelreservation) WHERE n.checkindate CONTAINS '/2020'

RETURN '2020 Total' AS Year, sum(n.totalpaid) AS GrandTotal, sum(n.days) AS TotalDays

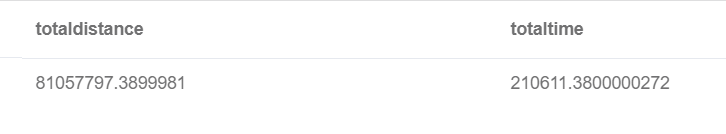


**c.**

**Total distance travelled by flights and the total time spent travelling over the last 2 years. Additionally, find the breakdown of distance and time travelled by each company over the last 2 years.**

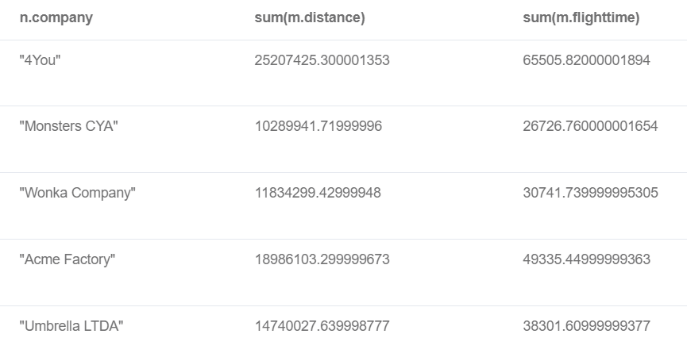
MATCH (n:flightbooking)

RETURN sum(n.distance) AS totaldistance, sum(n.flighttime) AS totaltime



MATCH (n:users) - [:travels]- (m:flightbooking)

RETURN n.company, sum(m.distance), sum(m.flighttime)



**d. How many employees travelled on business in 2019 vs in 2020**

MATCH (m:users) - [:travels] - (n:flightbooking) WHERE n.flightdate CONTAINS '/2019' RETURN '2019' AS Year, count(distinct m.usercode)

UNION

MATCH (m:users) - [:travels] - (n:flightbooking) WHERE n.flightdate CONTAINS '/2020' RETURN '2020' AS Year, count(distinct m.usercode)

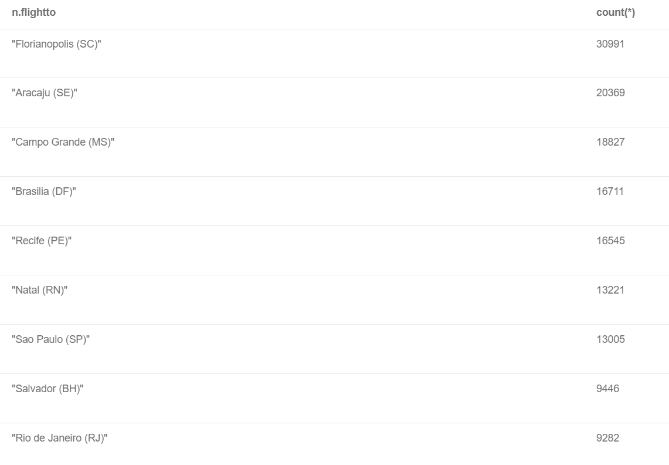


**2) What is the destination that most employees (users) travelled to for business?**

MATCH (n:flightbooking)

RETURN n.flightto, count(\*)

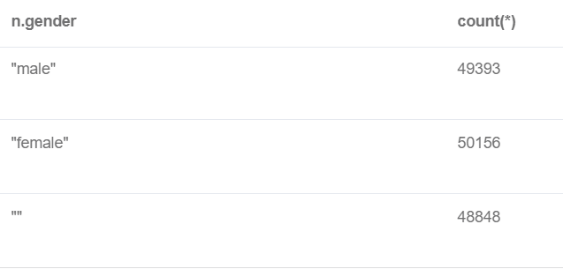
ORDER BY count(\*) DESC



1. **What is the gender distribution between men and women? Does one gender group take more business trips than the other?**

MATCH (n:users) - [:travels] - (m:flightbooking)

RETURN n.gender, count(\*)

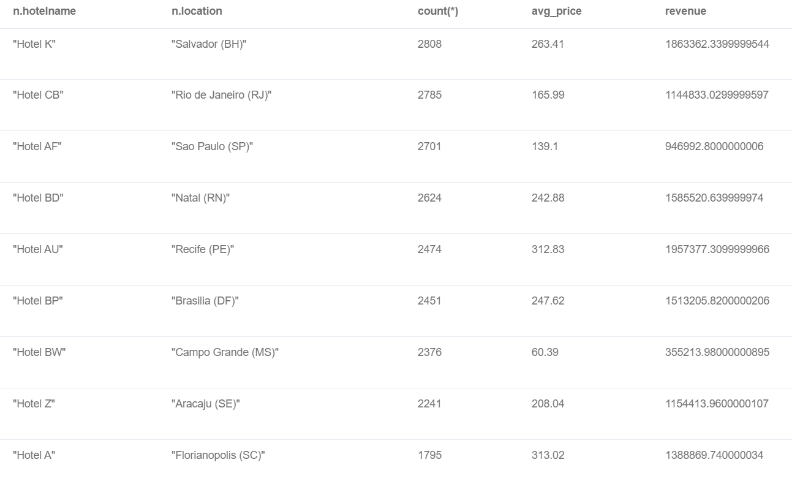


1. **Which hotel received the most reservations and how much did each hotel make in revenue from these reservations?**

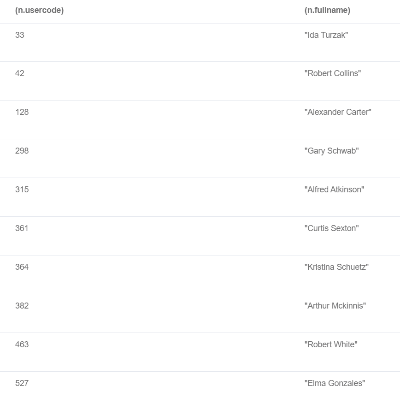
MATCH (n:hotelreservation)

RETURN n.hotelname, n.location, count(\*), avg(n.hotelprice) AS avg\_price, sum(n.totalpaid) AS revenue

ORDER BY count(\*) DESC



1. **Which users did not stay in hotels during their business travels?**



MATCH (n:users)

WHERE NOT EXISTS ((n) - [:stays] - ())

RETURN (n.usercode), (n.fullname)

1. **Which hotel offers the cheapest nightly rate, and which hotel offers the most expensive nightly rate?**

MATCH (n:hotelreservation)

CALL {MATCH (n:hotelreservation) RETURN max(n.hotelprice)}

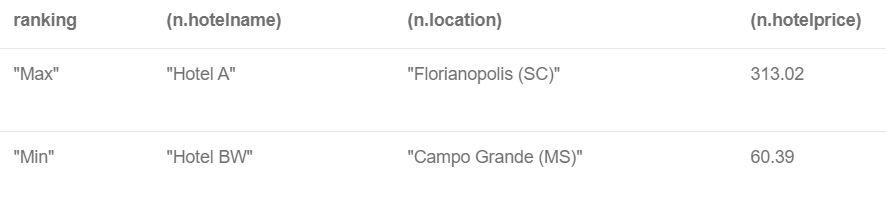
RETURN 'Max' AS ranking, (n.hotelname), (n.location), (n.hotelprice) LIMIT 1

UNION

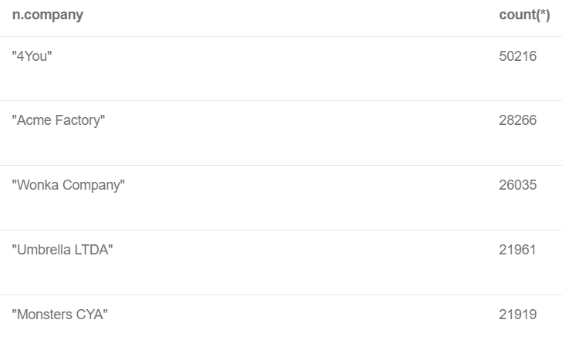
MATCH (n:hotelreservation)

CALL {MATCH (n:hotelreservation) RETURN min(n.hotelprice)}

RETURN 'Min' AS ranking, (n.hotelname), (n.location), (n.hotelprice) ORDER BY (n.hotelprice) ASC LIMIT 1



1. ***\*\*THIS IS A THREE-PART QUERY\*\*:***
   1. **Which company has the most business flight activity?**



MATCH (n:users) - [:travels] - (m:flightbooking)

RETURN n.company, count(\*)

ORDER BY count(\*) DESC

* 1. **Which users did the most travel? Which company do these users work for? Did the top traveller come from the top company (see 4a)?**

MATCH (n:users) - [:travels] - (m:flightbooking)

RETURN n.usercode, n.fullname, n.company, count(\*)

ORDER BY count(\*) DESC

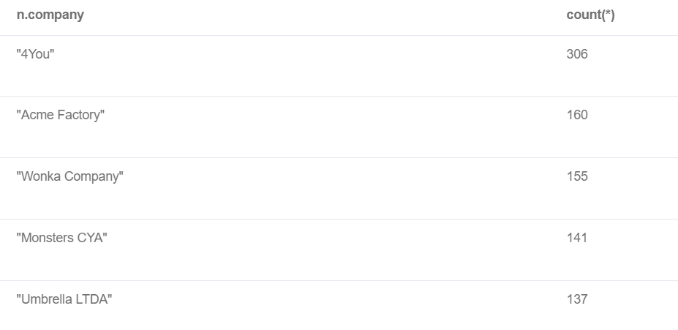


*It looks like there are multiple employees who took 133 flights (which is the top # of flights) – it is necessary to write another query to determine which company has the most top travellers:*

MATCH (n:users) - [:travels] - (m:flightbooking)

WITH n, count(m.travelcode) as cnt

WHERE cnt = 133

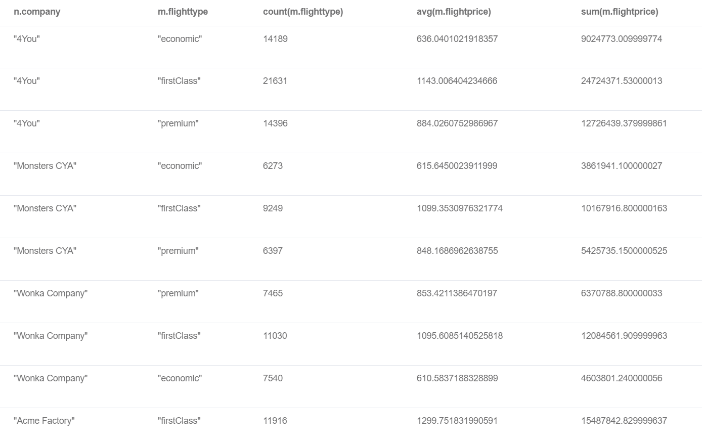
RETURN n.company, count(\*)

ORDER BY count(\*) DESC

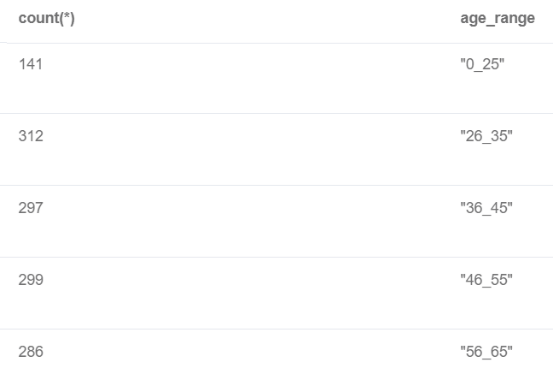
1. **Let’s take a look at the different classes of flights (economic, first class, premium) that each company purchased.**

MATCH (n:users) - [:travels] - (m:flightbooking)

RETURN n.company, m.flighttype, count(m.flighttype), avg(m.flightprice), sum(m.flightprice)



1. **How many employees are in each age range?**



MATCH (n:users)

RETURN count(\*), (case

when (n.age>=0 AND n.age <=25) THEN '0\_25'

when (n.age>=26 AND n.age<=35) THEN '26\_35'

when (n.age>=36 AND n.age<=45) THEN '36\_45'

when (n.age>=46 AND n.age<=55) THEN '46\_55'

when (n.age>=56 AND n.age<=65) THEN '56\_65'

when n.age>=66 THEN 'over\_65'

END) AS age\_range

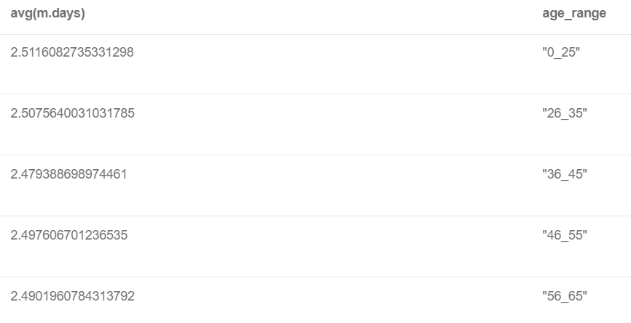
ORDER BY age\_range

1. **Average length of stay per age group**

MATCH (n:users) - [:stays] - (m:hotelreservation)

RETURN avg(m.days), (case

when (n.age>=0 AND n.age <=25) THEN '0\_25'

when (n.age>=26 AND n.age<=35) THEN '26\_35'

when (n.age>=36 AND n.age<=45) THEN '36\_45'

when (n.age>=46 AND n.age<=55) THEN '46\_55'

when (n.age>=56 AND n.age<=65) THEN '56\_65'

when n.age>=66 THEN 'over\_65'

END) AS age\_range

ORDER BY age\_range

1. **In which city are youngest employees most likely to be located?**

MATCH (n:users) - [:travels] - (m:flightbooking)

WHERE n.ageRange = '0\_25'

RETURN n.ageRange, m.flightfrom, count(\*)

ORDER BY count(\*) DESC



1. **On what months are most flights done? And are flight prices more expensive during those months compared to other months?**

match (n:flightbooking)

RETURN count(\*), (case

when (n.flightdate STARTS WITH '01/') then 'january'

when (n.flightdate STARTS WITH '02/') then 'february'

when (n.flightdate STARTS WITH '03/') then 'march'

when (n.flightdate STARTS WITH '04/') then 'april'

when (n.flightdate STARTS WITH '05/') then 'may'

when (n.flightdate STARTS WITH '06/') then 'june'

when (n.flightdate STARTS WITH '07/') then 'july'

when (n.flightdate STARTS WITH '08/') then 'august'

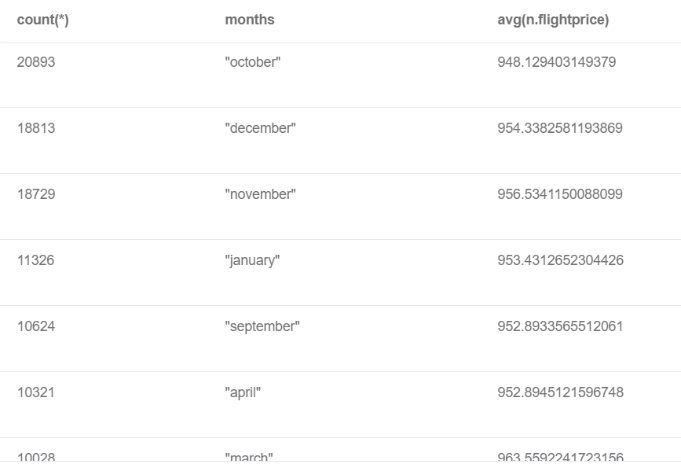
when (n.flightdate STARTS WITH '09/') then 'september'

when (n.flightdate STARTS WITH '10/') then 'october'

when (n.flightdate STARTS WITH '11/') then 'november'

when (n.flightdate STARTS WITH '12/') then 'december'

END) AS months, avg(n.flightprice)

ORDER BY count(\*) DESC

1. **Is there a correlation between distance and days. We are trying to find out if people tend to stay in hotels for more days if they are travelling long distances, to avoid doing 2 long trips back-to-back).**

MATCH (n:flightbooking) - [:travels] -> (m:users) <- [:stays] - (p:hotelreservation)

RETURN (case

when (n.distance >= 0 AND n.distance <= 200) THEN '0\_200'

when (n.distance >= 201 AND n.distance <= 400) THEN '200\_400'

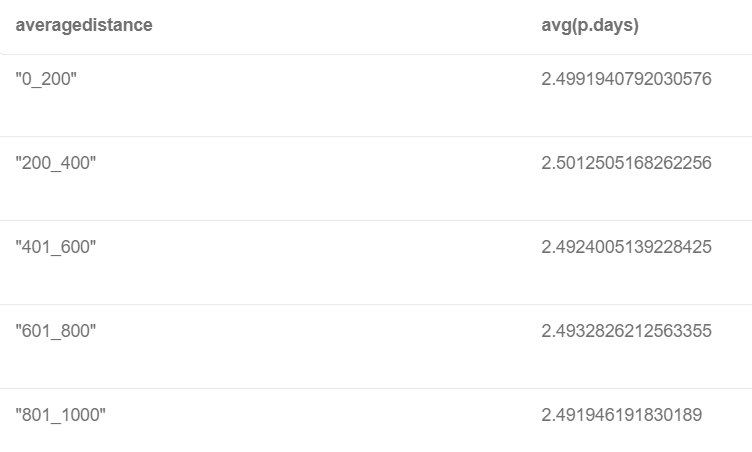
when (n.distance >= 401 AND n.distance <= 600) THEN '401\_600'

when (n.distance >= 601 AND n.distance <= 800) THEN '601\_800'

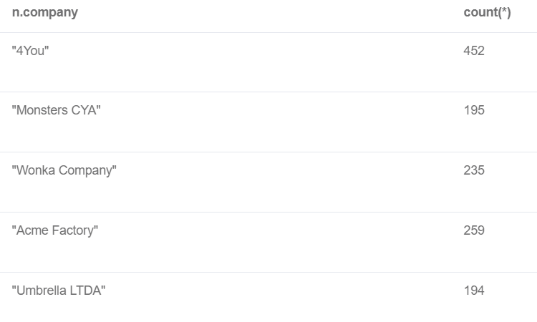
when (n.distance >= 801 AND n.distance <= 1000) THEN '801\_1000'

END) as averagedistance, avg(p.days)

ORDER BY averagedistance

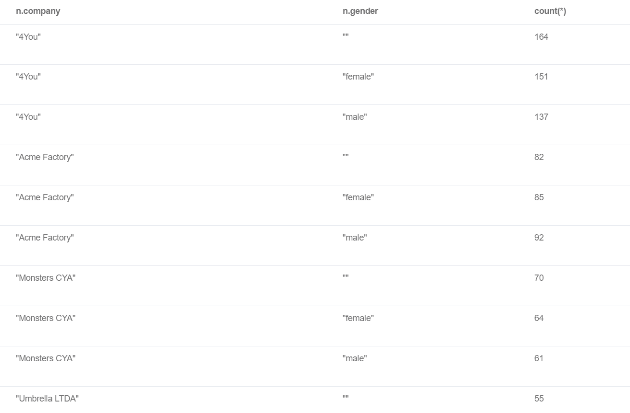


1. **How many employees are there per company?**

MATCH (n:users)

RETURN n.company, count(\*)

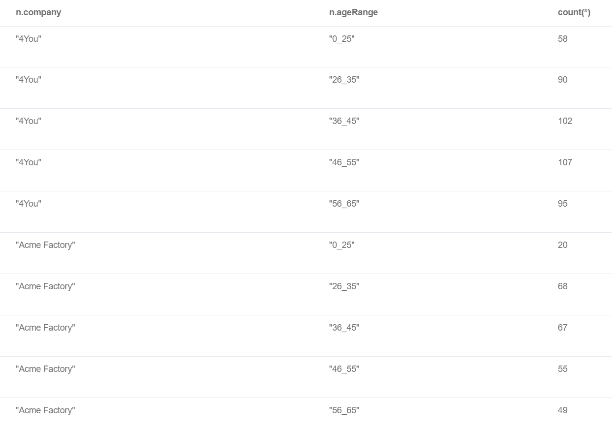
1. **What is the gender distribution in each company**

MATCH (n:users)

RETURN n.company, n.gender, count(\*)

ORDER BY n.company, n.gender

1. **Age distribution at the company**



MATCH (n:users)

RETURN n.company, n.ageRange, count(\*)

ORDER BY n.company, n.ageRange

1. **Let’s have a breakdown of the Total amount of money that each company spent on flights and on hotels separately, as well as the Grand Sum they spent on both flights and hotels over the last 2 years.**

MATCH (n:flightbooking) - [:travels] -> (m:users) <- [:stays] - (p:hotelreservation)

RETURN m.company, sum(n.flightprice) as TotalFlights, sum(p.totalpaid) as TotalHotel, sum(n.flightprice + p.totalpaid) AS TotalSpend

ORDER BY TotalSpend



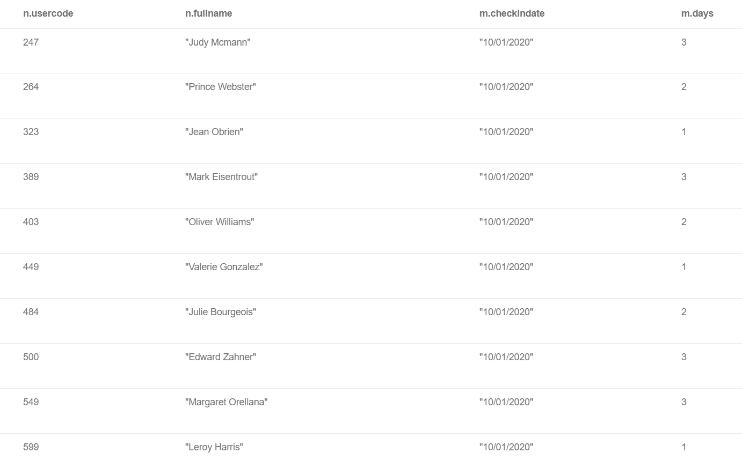
1. **SCENARIO: Since October, November and December are usually the busiest travel month (as determined by Query #12), Hotel AU wants to temporarily VIEW all of its upcoming reservations in those months to ensure that they are ready for the volume of people checking in.**

MATCH (n:users)-[:stays] - (m:hotelreservation)

WHERE m.hotelname = 'Hotel AU' AND m.checkindate ENDS WITH '/2020' AND (m.checkindate STARTS WITH '10/' OR m.checkindate STARTS WITH '11/' OR m.checkindate STARTS WITH '12/')

RETURN n.usercode, n.fullname, m.checkindate, m.days

ORDER BY m.checkindate



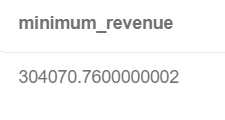
1. **Hotel AU knows from historical experience that business travellers who only book for 1 night are more likely to cancel. They would like to calculate how much minimum revenue they can expect from their busy months (Oct, Nov, Dec) even if the guests who booked for 1 night cancel.**

MATCH (n:users) - [:stays] - (m:hotelreservation)

WITH n,m

WHERE m.hotelname = 'Hotel AU' AND m.checkindate ENDS WITH '/2020' AND (m.checkindate STARTS WITH '10/' OR m.checkindate STARTS WITH '11/' OR m.checkindate STARTS WITH '12/') AND m.days>1

RETURN sum(m.days \* m.hotelprice) as minimum\_revenue



1. **SCENARIO: There has been a closure of the airport in Natal (RN). Travel agency ‘FlyingDrops’ wants to see all the upcoming flights that it has booked where employees are either departing from or landing at the airport in Natal (RN). With this information, they will be able to rebook their flights via another airport.**

MATCH (n:users)-[:travels] - (m:flightbooking)

WHERE m.flightto = 'Natal (RN)' AND m.agency = 'FlyingDrops' AND (m.flightdate CONTAINS '12/24/2020' OR m.flightdate CONTAINS '12/25/2020')

RETURN n.usercode, n.fullname, m.flighttype, m.flightdate

