**Important Queries**

--find the name of the emps that contribute the top 40% of the salary of all the emps

--Method 1

select distinct Employee\_Salary, Employee\_Name

from Emptab e1

where 4>=

(select COUNT( distinct Employee\_Salary)

from Emptab e2

where e1.Employee\_Salary<= e2.Employee\_Salary)

order by Employee\_Salary desc;

Employee\_Salary Employee\_Name

100 A

80 B

60 S

50 R

--Method 2

select top 4 Employee\_Id, Employee\_Name ,Employee\_Salary , sum(Employee\_Salary)

over(order by Employee\_Id) AS running\_total,

CAST(Employee\_Salary as decimal(18,2)) \* 100 /SUM(Employee\_Salary) over() as pct

from Emptab order by pct desc;

Employee\_Id Employee\_Name Employee\_Salary running\_total pct

4 A 100 210 25.00000

6 B 80 350 20.00000

5 S 60 270 15.00000

3 R 50 110 12.50000

--Method 3

select top 40 PERCENT [Employee\_Salary], Employee\_Name

from Emptab order by Employee\_Salary desc;

Employee\_Salary Employee\_Name

100 A

80 B

60 S

50 R

---Name of the emp that contribute the top 40% of the salary of the emps

select distinct Employee\_Salary, Employee\_Name

from Emptab e1

where 2>=

(select COUNT( distinct Employee\_Salary)

from Emptab e2

where e1.Employee\_Salary<= e2.Employee\_Salary)

order by Employee\_Salary desc;

Employee\_Salary Employee\_Name

100 A

80 B

A & B together contributes to 45% of the total salary

--Interview Q2

--Top 5 categories on basis of the amount for each city

select top 5 r.category, city, r.amount from User1 u

inner join Rides r

on

u.id = r.id group by amount, u.city, category order by amount desc;

category city amount

6 Delhi 676

5 Bangalore 454

4 Kolkata 410

-- Find User ID who have not taken ride

select u.id, u.City from User1 u

FULL outer join Rides r

on

u.id = r.id where rides IS NULL;

id City

1 Bangalore

2 Delhi

--Interview Q3

--3a) PCt of conversations in last 14days

select Sender\_id, receiver\_id, sum(has\_reaction) as Total\_Reaction, COUNT(has\_reaction) as Reaction\_count,

ROUND(sum(Has\_reaction)\*100.0/COUNT(Has\_reaction), 1) as Pct\_of\_conversation

from Reaction

where DATEDIFF(day, '2021-03-19', reaction.Date) <=14

Group by sender\_id, receiver\_id;

Sender\_id receiver\_id Total\_Reaction Reaction\_count Pct\_of\_conversation

1234 2314 3 7 42.900000000000

1234 2315 7 15 46.700000000000

1234 2316 1 3 33.300000000000

1234 2317 2 4 50.000000000000

--3b) Average number of days from conversation start to time of message when first reaction is used

select Sender\_id, receiver\_id, sum(has\_reaction) as Total\_Reaction, COUNT(has\_reaction) as Reaction\_count,

ROUND(sum(Has\_reaction)\*100.0/COUNT(Has\_reaction), 1) as Pct\_of\_conversation

from Reaction

where DATEDIFF(day, '2021-03-19', reaction.Date) <=14

Group by sender\_id, receiver\_id;

select \* into Conv\_Start from(

select Sender\_id, receiver\_id, MIN(date) as conv\_start\_date, COUNT(sender\_id) as num\_of\_msgs from Reaction

Group by sender\_id, receiver\_id)b

select \* into Fst\_reaction from (

Select Sender\_id, receiver\_id, MIN(date) as Fst\_rctn\_date

from Reaction

where has\_reaction = 1

Group by sender\_id, receiver\_id)b

select \* into Dates from(

select Conv\_Start.sender\_id, Conv\_Start.receiver\_id, num\_of\_msgs, conv\_start\_date,

Fst\_reaction.Fst\_rctn\_date from Conv\_Start

INNER JOIN Fst\_reaction

ON Conv\_Start.receiver\_id = Fst\_reaction.receiver\_id)b

select AVG(DIFF) as Avg\_num\_of\_days from(

select \*, (DATEDIFF(day, left(conv\_start\_date,10), left(Fst\_rctn\_date,10))) as DIFF

from Dates)b

Solution:

Avg\_num\_of\_days

5

----3c) Hypothesis Test

Conversation with message reaction is more active than the conversations without reactions, how will you validate the above hypothesis using the above data

This can be validated by taking difference in days from one message to next message and then we can consider the average of days when has reaction is zero and average number of days when has reaction is one. When the average number of days is less indicates there is active conversation.

--Interview Q4

--generate the output

-- Calculate avg(sal) & Group by Manager\_ID

--Rename Emp\_name as Manager\_name

select \* into Test from(

select Manager\_id ,avg(salary) as Average\_sal\_Under\_Manager

from Empman

where Manager\_Id>1

Group by Manager\_ID)b

select t.Manager\_id, Emp\_name as Manager\_name, t.Average\_sal\_Under\_Manager from Empman e

INNER JOIN Test t

on e.Emp\_id = t.Manager\_id

Manager\_id Manager\_name Average\_sal\_Under\_Manager

16 Rajesh 65000

17 Raman 62500

18 Santosh 53750

-----------------------------------Interview MongoDB----------------------------------------------------------------------------------------------------

---Q1

import pymongo

client = pymongo.MongoClient()

db = client["mydb"]

col = db["Transaction"]

result1 = col.aggregate([

{"$lookup":

{

"from": "Empnew",

"localField": "id",

"foreignField": "id",

"as": "Employee\_transaction"

}},

{"$unwind": "$Employee\_transaction"},

{"$match":

{"status":{"$eq": "Pending"}}

},

{"$group":

{"\_id": {"empid":"$Employee\_transaction.empid","status": "$status"},

"Number of Pending Transactions": {"$sum": 1}}

}])

for i in result1:

print(i)

Solution:

{'\_id': {'empid': '66', 'status': 'Pending'}, 'Number of Pending Transactions': 1}

{'\_id': {'empid': '44', 'status': 'Pending'}, 'Number of Pending Transactions': 2}

---------------------------------------------------------------------------------------------------------------------------------------------------------------

----Q2

import pymongo

client = pymongo.MongoClient()

db = client["mydb"]

col = db["Empnew"]

result1 = col.aggregate([{"$group":

{"\_id":"$empid",

"Average\_salary" : {"$avg":"$Salary"}}}])

for i in result1:

print(i)

Solution:

{'\_id': '22', 'Average\_salary': 53333.333333333336}

{'\_id': '66', 'Average\_salary': 10000.0}

{'\_id': '77', 'Average\_salary': 2000.0}

{'\_id': '44', 'Average\_salary': 45000.0}

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

--Q3

import pymongo

client = pymongo.MongoClient()

db = client["mydb"]

col = db["Transaction"]

result1 = col.aggregate([

{"$lookup":

{

"from": "Empnew",

"localField": "id",

"foreignField": "id",

"as": "Employee\_transaction"

}},

{"$unwind": "$Employee\_transaction"},

{"$group":

{"\_id": {"empid":"$Employee\_transaction.empid"},

"Average\_Transaction\_Amount": {"$avg": "$amount"}}

}

])

for i in result1:

print(i)

Solution:

{'\_id': {'empid': '44'}, 'Average\_Transaction\_Amount': 850.0}

{'\_id': {'empid': '66'}, 'Average\_Transaction\_Amount': 1000.0}

{'\_id': {'empid': '22'}, 'Average\_Transaction\_Amount': 600.0}

{'\_id': {'empid': '77'}, 'Average\_Transaction\_Amount': 1100.0}

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

--- Q4

import pymongo

client = pymongo.MongoClient()

db = client["mydb"]

col = db["Transaction"]

result1 = col.aggregate([

{"$lookup":

{

"from": "Empnew",

"localField": "id",

"foreignField": "id",

"as": "Employee\_transaction"

}},

{"$unwind": "$Employee\_transaction"},

{"$match":

{"status":{"$eq": "Transferred"}}

},

{"$group":

{"\_id": {"empid":"$Employee\_transaction.empid","amount": "$amount"},

"Count": {"$sum": 1}}

},

{"$sort":

{"amount": -1}},

{"$limit": 5}

])

for i in result1:

print(i)

Solution:

{'\_id': {'empid': '22', 'amount': 700}, 'Count': 1}

{'\_id': {'empid': '22', 'amount': 600}, 'Count': 1}

{'\_id': {'empid': '22', 'amount': 500}, 'Count': 1}