## Write a query to find out third highest salary. (<https://www.youtube.com/watch?v=aU-sYZEdX1Y>)

**Ans:**

select \*,dense\_rank() over(order by salary desc) rnk from employees

### Write query for the given customer name has to be segregated into first name lastname and the middlename(<https://www.youtube.com/watch?v=tVQUsozKkyI>)

|  |
| --- |
| Source: |
| **NAME** |
| Neeru Konda Babu |
| Anand Murugan |
| Anand\_Swathi\_Aadarsh |

|  |  |  |  |
| --- | --- | --- | --- |
| Target: |  |  |  |
| **NAME** | **First\_Name** | **Middle\_Name** | **Last\_Name** |
| Neeru Konda Babu | Neeru | Konda | Babu |
| Anand Murugan | Anand |  | Murugan |
| Anand\_Swathi\_Aadarsh | Anand | Swathi | Aadarsh |

Ans:

with clean\_delimiter as

(

select name, replace(name,'\_',' ') as cleaned\_name from cust\_name

),

name\_part\_cnt as (

select cleaned\_name as name,

length(cleaned\_name)-length(replace(cleaned\_name,' ',''))+1 as name\_part\_cnt

from clean\_delimiter

)

select \*,

case

when name\_part\_cnt=1 then name

else split\_part(name,' ',1) end AS FIRST\_NAME,

case

when name\_part\_cnt= 3 THEN split\_part(name,' ',2)

else NULL

end MIDDLE\_NAME,

case

when name\_part\_cnt=3 then split\_part(name,' ',3)

when name\_part\_cnt=2 then split\_part(name,' ',2)

else NULL

end as LAST\_NAME

from name\_part\_cnt

### 3.

### <https://www.youtube.com/watch?v=qeQtvG1v_WM>

source:

|  |  |  |
| --- | --- | --- |
| ID | STUDENT\_ID | DEPT\_ID |
| 1 | 1 | 105 |
| 1 | 2 | NULL |
| 1 | 3 | NULL |
| 1 | 4 | NULL |
| 1 | 11 | 110 |
| 1 | 12 | NULL |
| 1 | 13 | NULL |
| 1 | 14 | NULL |

Target:

|  |  |  |
| --- | --- | --- |
| ID | STUDENT\_ID | DEPT\_ID |
| 1 | 1 | 105 |
| 1 | 2 | 105 |
| 1 | 3 | 105 |
| 1 | 4 | 105 |
| 1 | 11 | 110 |
| 1 | 12 | 110 |
| 1 | 13 | 110 |
| 1 | 14 | 110 |

Ans:

with cte as

(

select \*,count(DEPT\_ID) over(order by student\_id) as rnk from students)

select \*,max(DEPT\_ID) over(partition by rnk) AS NEW\_DEPT\_ID from cte

### 4. Need to find out continuous MIN and MAX value. (<https://www.youtube.com/watch?v=uz_Bln-CJR4>)

|  |  |
| --- | --- |
| **Source:** |  |
| ID | VALUE |
| P | 1 |
| P | 2 |
| P | 3 |
| P | 5 |
| P | 6 |
| P | 8 |
| P | 9 |
| Q | 11 |
| R | 1 |
| R | 2 |
| R | 3 |

|  |  |  |
| --- | --- | --- |
| Target: |  |  |
| ID | MIN\_SEQ | MAX\_SEQ |
| P | 1 | 3 |
| P | 5 | 6 |
| P | 8 | 9 |
| R | 1 | 3 |
| Q | 11 | 11 |

Ans :

with cte as (

select \*,row\_number() over(partition by ID order by value) as RN,

value-(row\_number() over(partition by ID order by value)) as FLAG from Amazon\_Emp

), cte1 as (

select ID,min(value) over(partition by flag) AS MIN\_SEQ,

max(value) over(partition by flag ) as MAX\_SEQ from cte)

select ID,MIN\_SEQ,MAX\_SEQ from cte1 group by ID,MIN\_SEQ,MAX\_SEQ

### 15. find origin and final destination details.

( <https://www.youtube.com/watch?v=CigAAYkmpjk> )

Source:

|  |  |  |  |
| --- | --- | --- | --- |
| CUST\_ID | FLIGHT\_ID | ORIGIN | DESTINATION |
| 1 | SG1234 | Delhi | Hyderabad |
| 1 | SG3476 | Kochi | Mangalore |
| 1 | 69876 | Hyderabad | Kochi |
| 2 | 68749 | Mumbai | Varanasi |
| 2 | SG5723 | Varanasi | Delhi |

Expected Target:

|  |  |  |
| --- | --- | --- |
| CUST\_ID | ORIGIN | DESTINATION |
| 1 | Delhi | Mangalore |
| 2 | Mumbai | Delhi |

with get\_origin as (

select f1.cust\_id,f1.origin from flights f1 left join flights f2 on f1.cust\_id=f2.cust\_id and f1.origin=f2.destination where f2.cust\_id IS NULL ),

get\_destination as (

select f1.cust\_id,f1.destination from flights f1 left join flights f2 on f1.cust\_id =f2.cust\_id and

f1.destination=f2.origin where f2.cust\_id is null

)

select a.cust\_id,a.origin,b.destination from get\_origin a inner join get\_destination b on a.cust\_id=b.cust\_id

### 16. Find the missing weeks in a table.

( <https://www.youtube.com/watch?v=rQSQpsYMVU0> )

Source :

|  |  |  |
| --- | --- | --- |
| PID | SLS\_DT | SLS\_AMT |
| 201 | 11-07-2024 | 140 |
| 201 | 18-07-2024 | 160 |
| 201 | 25-07-2024 | 150 |
| 201 | 01-08-2024 | 180 |
| 201 | 15-08-2024 | 170 |
| 201 | 29-08-2024 | 130 |

Target:

|  |
| --- |
| WEEK\_START\_DATE |
| 08-08-2024 |
| 22-08-2024 |

Ans:

with weeks as (

select min(sls\_dt) as week\_start\_date from sls\_tbl

union ALL

select DATEADD(week,1,week\_start\_date) from weeks where week\_start\_date < (select max(sls\_dt) from sls\_tbl)

)

select wks.week\_start\_date from sls\_tbl sls Right join weeks wks on sls.sls\_dt=wks.week\_start\_date

where sls\_DT is null

### 17. Finding best delivery partner

### (https://www.youtube.com/watch?v=p-ql-bwuMTg)

Source:

|  |  |  |  |
| --- | --- | --- | --- |
| BRAND\_1 | BRAND\_2 | BRAND\_3 | WINNER |
| A | B | C | B |
| B | C | E | E |
| C | A | D | D |
| D | E | A | A |
| F | B | C | F |

Expected Output:

|  |  |  |  |
| --- | --- | --- | --- |
| BRAND\_NAME | NO\_OF\_RIDES | WIN\_RIDES | LOSSES |
| A | 3 | 1 | 2 |
| B | 3 | 1 | 2 |
| C | 4 | 0 | 4 |
| D | 2 | 1 | 1 |
| E | 2 | 1 | 1 |
| F | 1 | 1 | 0 |

Ans:

with cte as (

select Brand\_1 as brand\_name,Winner from Delievry\_Partner

union all

select Brand\_2 as brand\_name,Winner from Delievry\_Partner

union all

select Brand\_3 as brand\_name,Winner from Delievry\_Partner

)

select brand\_name,

count(BRAND\_NAME) as NO\_OF\_RIDES,

count(case when BRAND\_NAME=WINNER THEN 1 END) as WIN\_CNT,

count(BRAND\_NAME)-count(case when BRAND\_NAME=WINNER THEN 1 END) as losses

from cte group by brand\_name

### 18. find running total on the transactions table.

### ( <https://www.youtube.com/watch?v=PPsDrLRYSx0> )

Source:

|  |  |  |  |
| --- | --- | --- | --- |
| TRANSACTION\_ID | TYPE | AMOUNT | TRANSACTION\_DATE |
| 53151 | deposit | 178 | 08-07-2022 |
| 29776 | withdrawal | 25 | 08-07-2022 |
| 16461 | withdrawal | 45 | 08-07-2022 |
| 19153 | deposit | 65 | 10-07-2022 |
| 77134 | deposit | 32 | 10-07-2022 |

Target:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TRANSACTION\_ID | TYPE | AMOUNT | TRANSACTION\_DATE | Running\_total |
| 53151 | deposit | 178 | 08-07-2022 | 178 |
| 29776 | withdrawal | 25 | 08-07-2022 | 153 |
| 16461 | withdrawal | 45 | 08-07-2022 | 108 |
| 19153 | deposit | 65 | 10-07-2022 | 173 |
| 77134 | deposit | 32 | 10-07-2022 | 205 |

ANS:

with cte as (

select \*,row\_number() over(order by TRANSACTION\_DATE) as rno from transactions\_1308

)

select \*,

sum(case when type='deposit' then AMOUNT ELSE AMOUNT\*-1 END) over(order by rno ) as running\_tot from CTE

### 19. Find events with 3 or more consecutive years for each pid

### ( <https://www.youtube.com/watch?v=xb912q7CTm0> )

Source :

|  |  |
| --- | --- |
| PID | YEAR |
| 1 | 2019 |
| 1 | 2020 |
| 1 | 2021 |
| 2 | 2021 |
| 2 | 2022 |
| 3 | 2019 |
| 3 | 2021 |
| 3 | 2022 |

Expected Target:

|  |
| --- |
| PID |
| 1 |

Ans:

with cte as(

select \*, row\_number() over(order by pid,year) rno,

year-row\_number() over(order by pid,year) grp from events

)

select PID,count(GRP) from cte group by PID,grp having count(\*)>=3

### 20. Write a SQL query to retrieve first and last for each customer from the orders table.

### (https://www.youtube.com/watch?v=ABtxyLWdavI)

Source:

|  |  |  |
| --- | --- | --- |
| ORDER\_ID | CUSTOMER\_ID | ORDER\_DATE |
| 1 | 101 | 05-01-2024 |
| 2 | 101 | 15-03-2024 |
| 3 | 101 | 20-05-2024 |
| 4 | 102 | 10-02-2024 |
| 5 | 102 | 25-04-2024 |
| 6 | 102 | 30-06-2024 |
| 7 | 103 | 01-01-2024 |
| 8 | 103 | 18-02-2024 |
| 9 | 103 | 25-03-2024 |

Expected Target :

|  |  |  |  |
| --- | --- | --- | --- |
| ORDER\_ID | CUSTOMER\_ID | ORDER\_DATE | ORDER\_TYPE |
| 1 | 101 | 05-01-2024 | FIRST ORDER |
| 3 | 101 | 20-05-2024 | LAST ORDER |
| 4 | 102 | 10-02-2024 | FIRST ORDER |
| 6 | 102 | 30-06-2024 | LAST ORDER |
| 7 | 103 | 01-01-2024 | FIRST ORDER |
| 9 | 103 | 25-03-2024 | LAST ORDER |

**Ans:**

with cte as (

select \*,

--row\_number() over(partition by customer\_id order by order\_date) as rno

min(order\_id) over(partition by CUSTOMER\_ID order by order\_date) as First\_order,

max(order\_id) over(partition by CUSTOMER\_ID order by order\_date desc) as last\_order

from orders),

cte2 as (

select a.order\_id,a.customer\_id,a.order\_date, 'FIRST ORDER' as order\_type from orders a inner join cte b on a.order\_id=b.first\_order group by a.order\_id,a.customer\_id,a.order\_date

union

select a.order\_id,a.customer\_id,a.order\_date, 'LAST ORDER' as order\_type from orders a inner join cte b on a.order\_id=b.last\_order group by a.order\_id,a.customer\_id,a.order\_date

)

select \* from cte2 order by customer\_id,order\_date