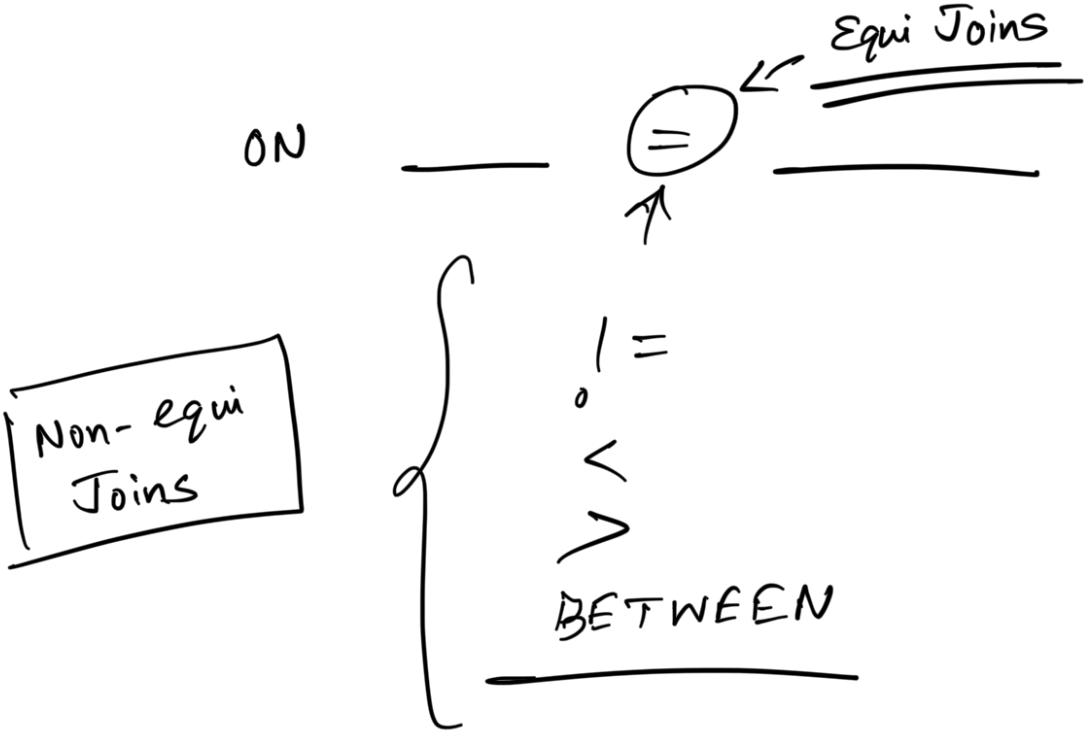


Q.

Renters

id	Pref. distt	min_bed rooms	budget

List all the possible pairs of renters along with their preferred district to see which all renters can rent the appt. together.

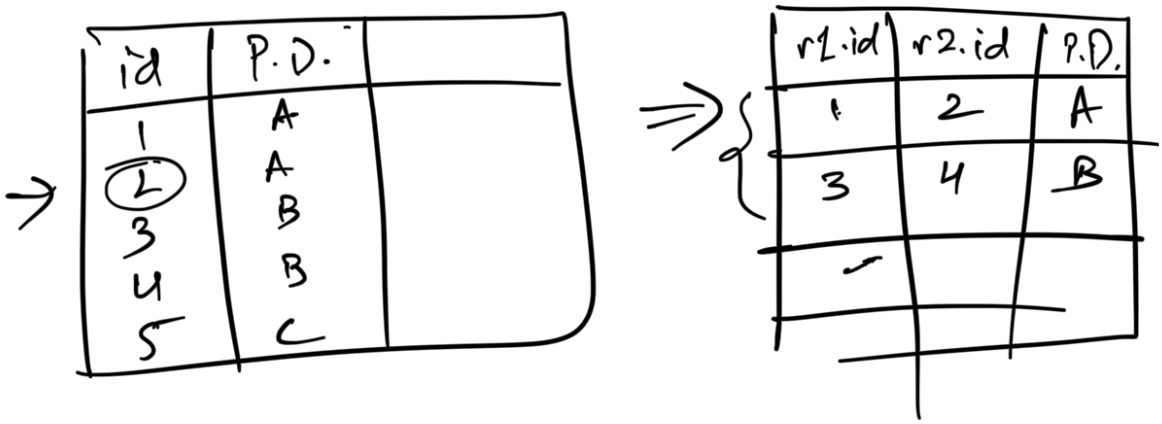
```
SELECT r1.id, r2.id, r1.pref.distt.

FROM renter r1
```

JOIN ratings r2

ON r1.preferred_diff = r2.pref_diff

AND

r1.id < r2.id



d. Find all employees who have a salary higher than their dept. avg. salary.

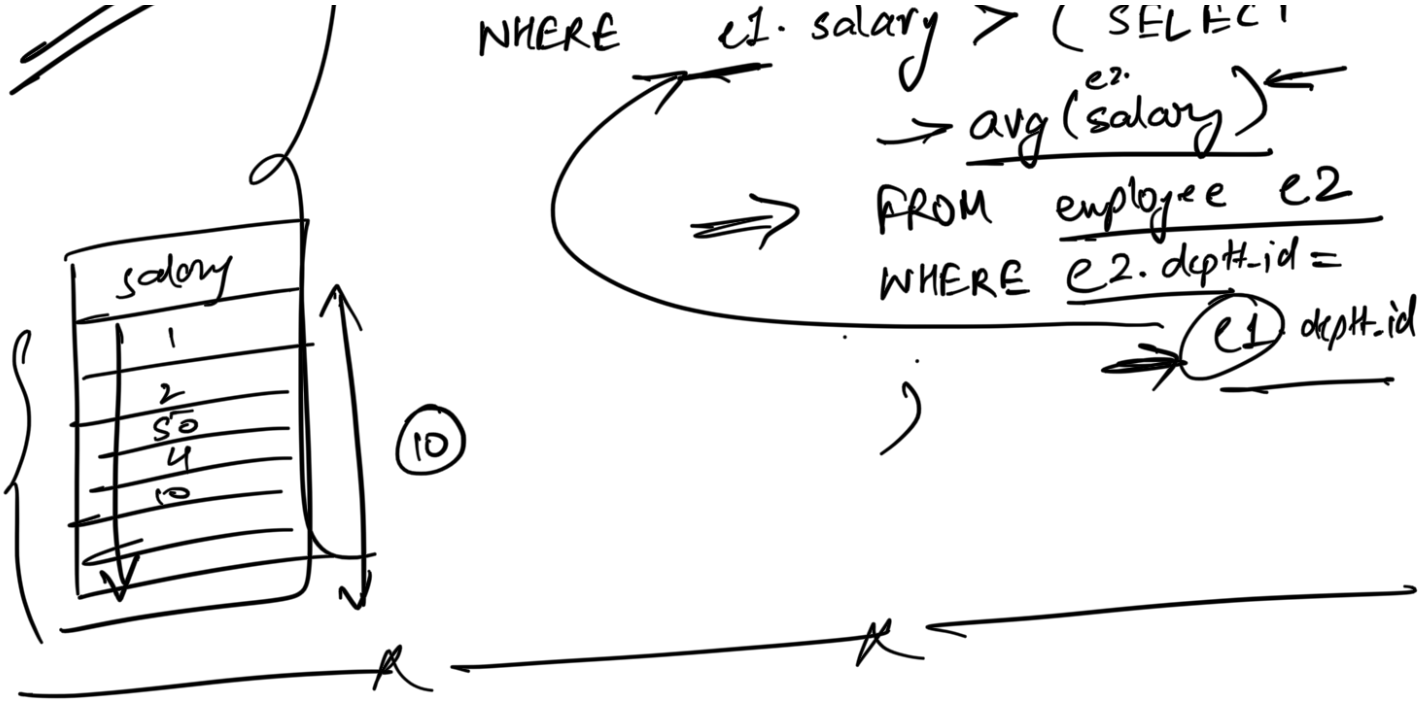
AVG()

e.id	dept_id	salary	

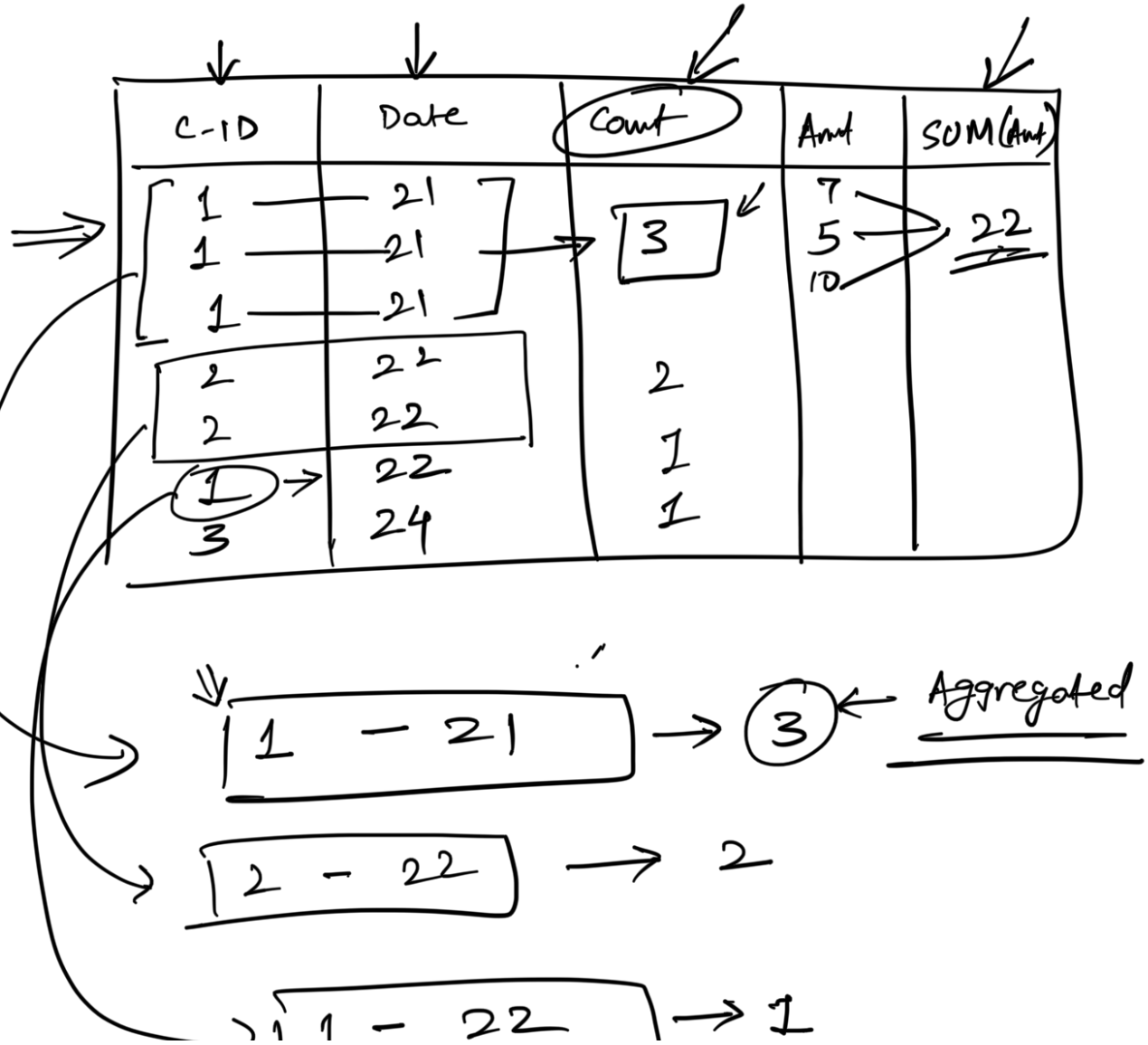
AVG(salary)

Correlated subquery

```
SELECT e.id
FROM employee e1
```


WHERE e1. salary > (SELECT
avg (e2. salary)
FROM employee e2
WHERE e2. dept_id =
e1. dept_id
salary
1
2
50
4
10
10

GROUP BY


C-ID
Date
Count
Amt
SUM(Amt)
1
1
1
21
21
21
3
7
5
10
22
2
2
22
22
2
1
1
1
3
22
24
1 - 21
3
Aggregated
2 - 22
2
1 - 22
1

Syntax:

SELECT ____ COUNT(C) ← AS ____
SUM(C) ← AS ____

FROM [table]

WHERE ____

GROUP BY ____, ____, ____

HAVING ____ ←

ORDER BY

LIMIT

Q. Get a list of unique customers who made purchase on each market date.

Q. Number of purchases each customer has made on each market date.

COUNT (*) ←

Q. Total quantity purchased by each customer on each market date.

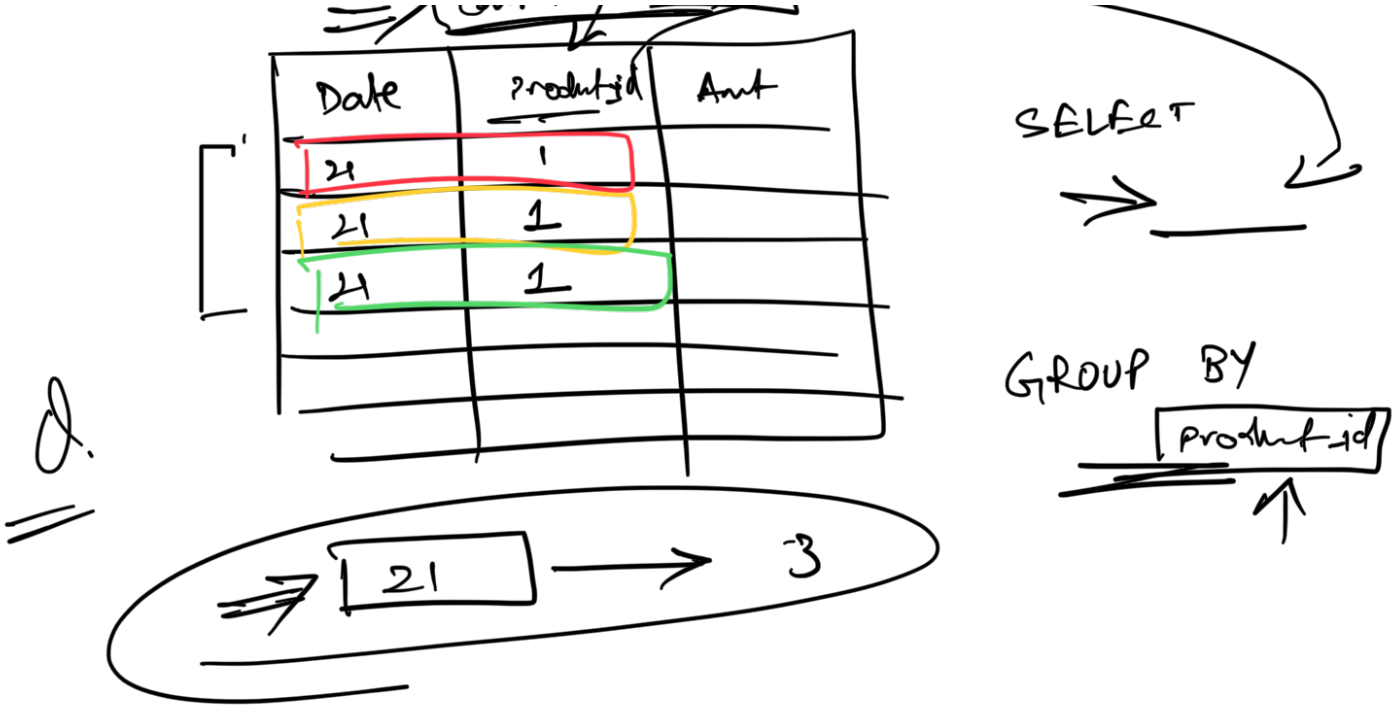
SUM (quantity)

Q. How many different kinds of products each customer purchased on each mkt date?

COUNT (*)	COUNT (1)	COUNT (col_name)	COUNT (DISTINCT col_name)
Total no. of records	Same as count (*)	Counts duplicate entries but ignores NULL	Neither duplicates Nor NULLs
	BUT Prefer to use COUNT (*) to avoid confusion		

Q. Calculate the total price paid by customer_id 3 per market_date.

COUNT(prod_id)



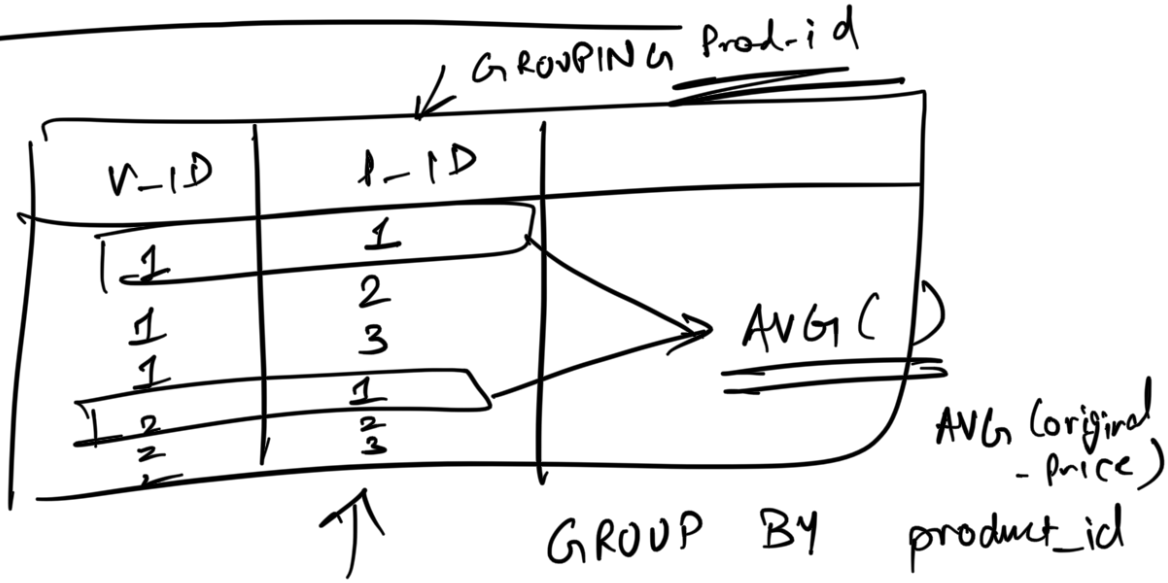
Q. How much each customer has spent at each vendor? Regardless of date.

Q. Find the most & least expensive product per prod. category.

Q. Avg original price of a product

- COUNT()
- SUM()
- MIN()
- MAX()
- AVG()

across vendor ?



Q Find the average amount spent on each market day ?

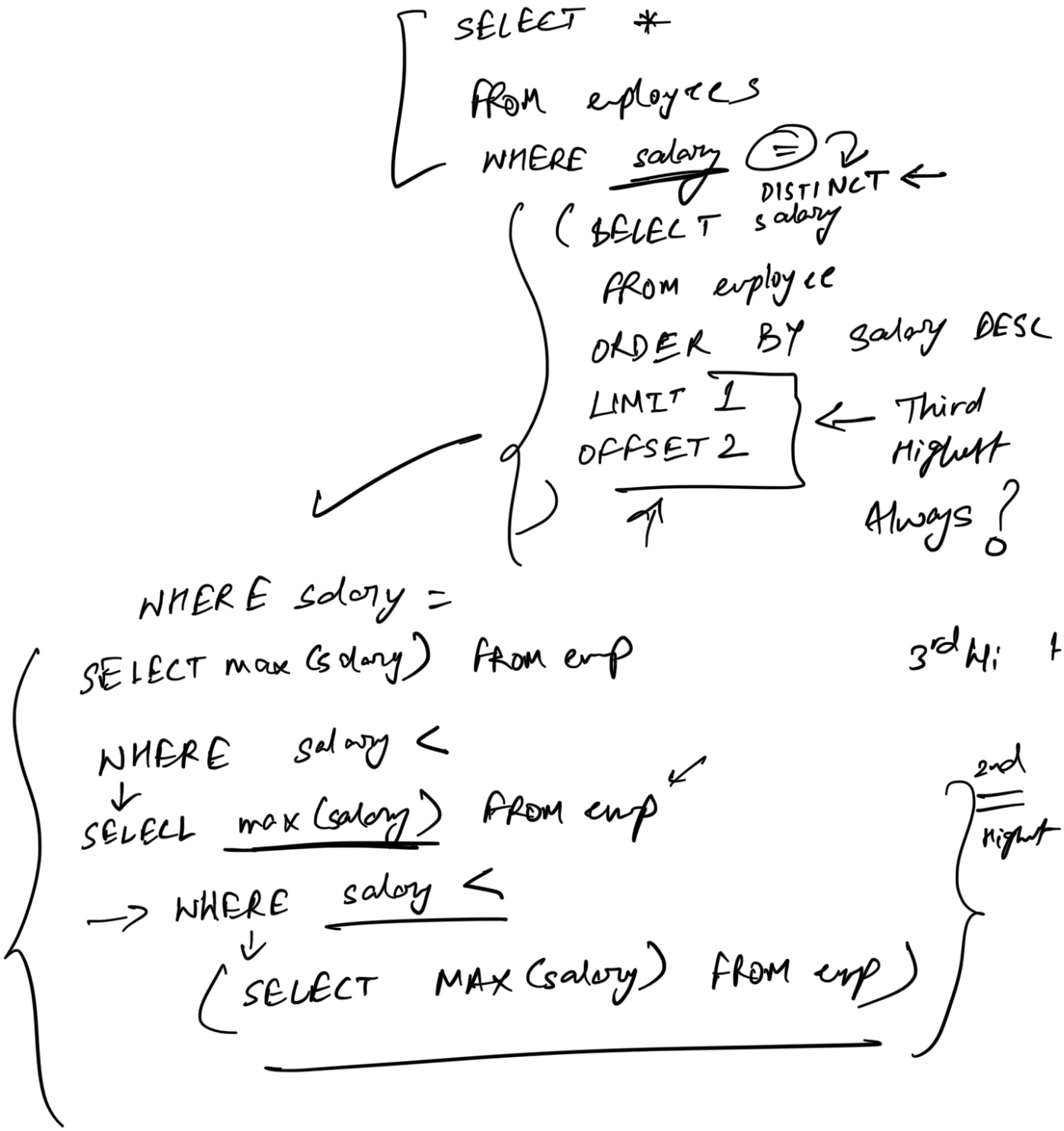
COUNT()

[1- where no of purchases were more than 3.

WHERE [2. Transaction amount was greater than 5] (WHERE)

HAVING (qty * cost) > 5 .

Q: Get all the employees who are earning the third-highest salary.


SELECT *
FROM employees
WHERE salary =
DISTINCT
(SELECT salary
FROM employee
ORDER BY salary DESC
LIMIT 1
OFFSET 2
Third Highest
Always ?
WHERE salary =
SELECT max(salary) FROM emp
3rd Hi
WHERE salary <
SELECT max(salary) FROM emp
-> WHERE salary <
(SELECT MAX(salary) FROM emp)
2nd Highest