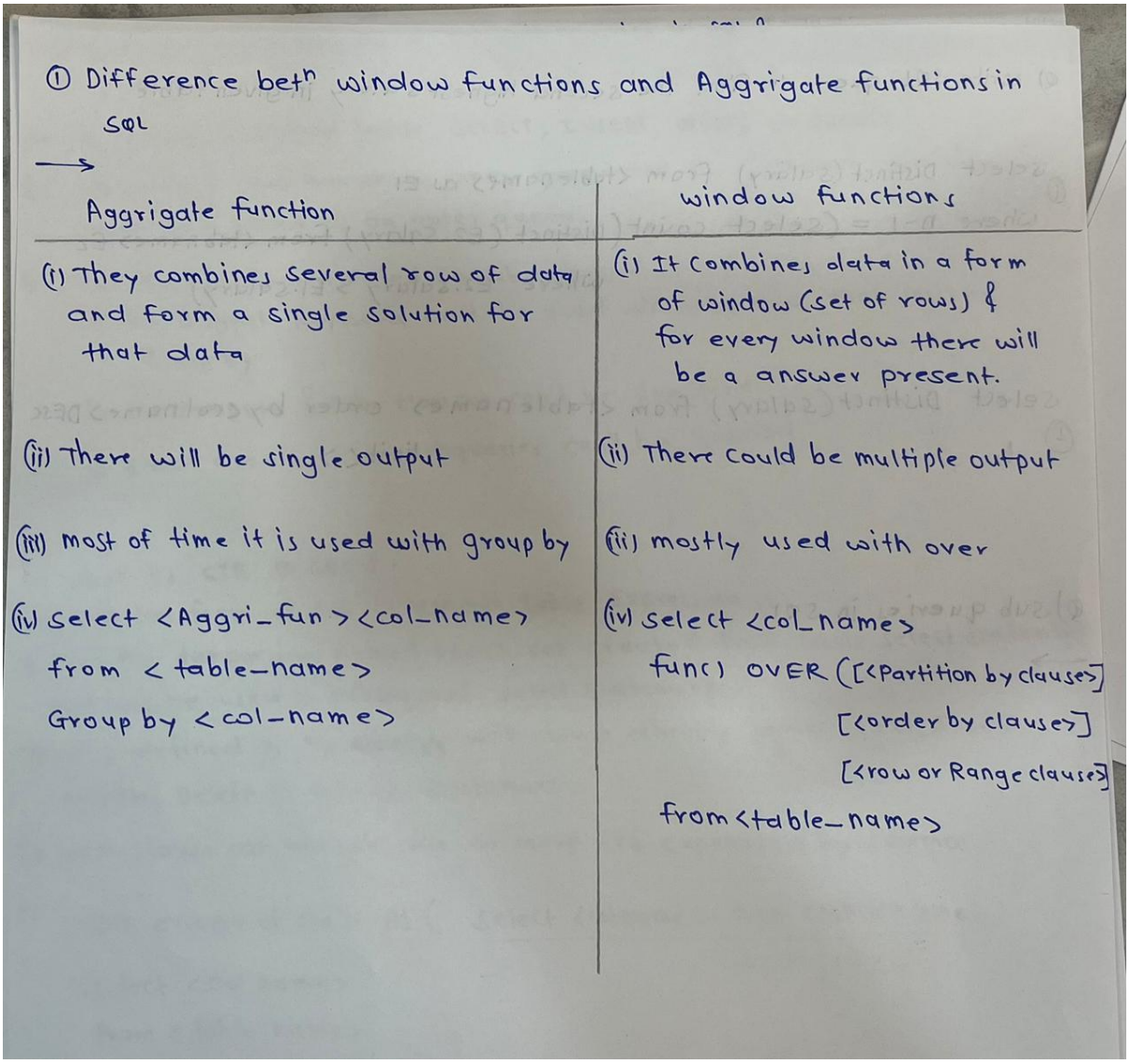
1. Describe the Difference Between Window Functions and Aggregate Functions in SQL



① Difference betⁿ window functions and Aggrigate functions in SQL

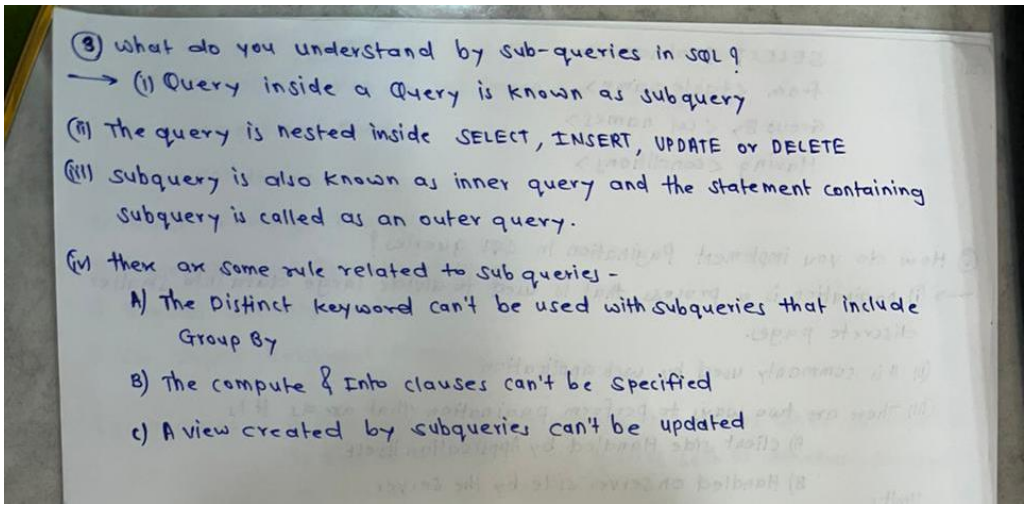
→

Aggrigate function	window functions
(i) They combines several row of data and form a single solution for that data	(i) It combines data in a form of window (set of rows) & for every window there will be a answer present.
(ii) There will be single output	(ii) There could be multiple output
(iii) most of time it is used with group by	(iii) mostly used with over
(iv) select <Aggri_fun><col_name> from <table_name> Group by <col_name>	(iv) select <col_name> func() OVER ([<Partition by clauses>] [<order by clauses>] [<row or Range clauses>] from <table_name>

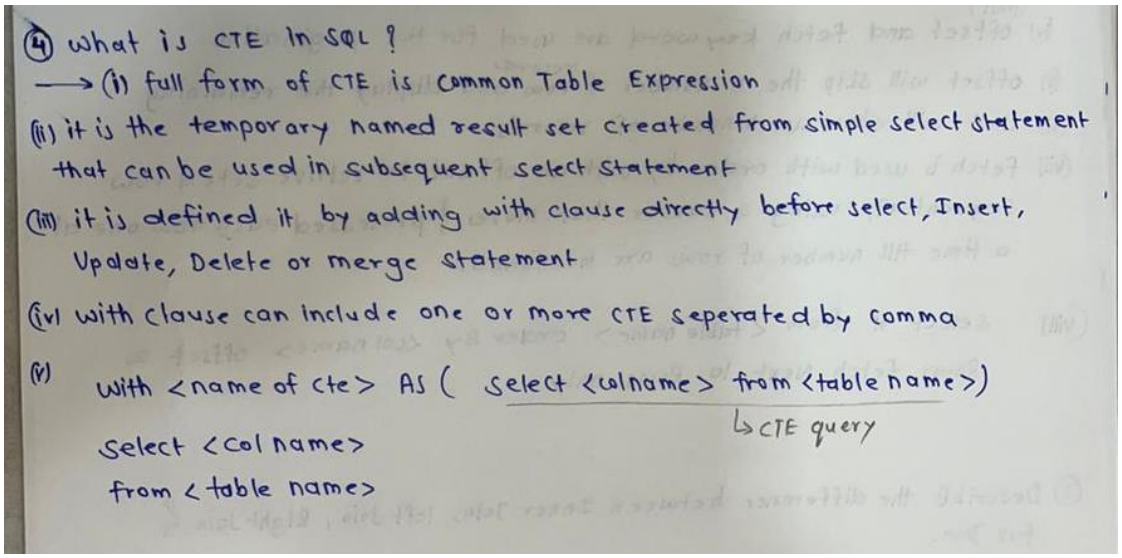
2. Write a SQL query to find the top three products with the highest revenue in the last quarter from a given database.

```
SELECT product_id, SUM(revenue) AS total_revenue
FROM sales
WHERE sales_date >= DATEADD(quarter, DATEDIFF(quarter, 0, GETDATE()) - 1, 0) -- > first day of last quarter
  AND sales_date < DATEADD(quarter, DATEDIFF(quarter, 0, GETDATE()), 0) -- First day of currect quarter
GROUP BY product_id
ORDER BY total_revenue DESC
LIMIT 3;
```

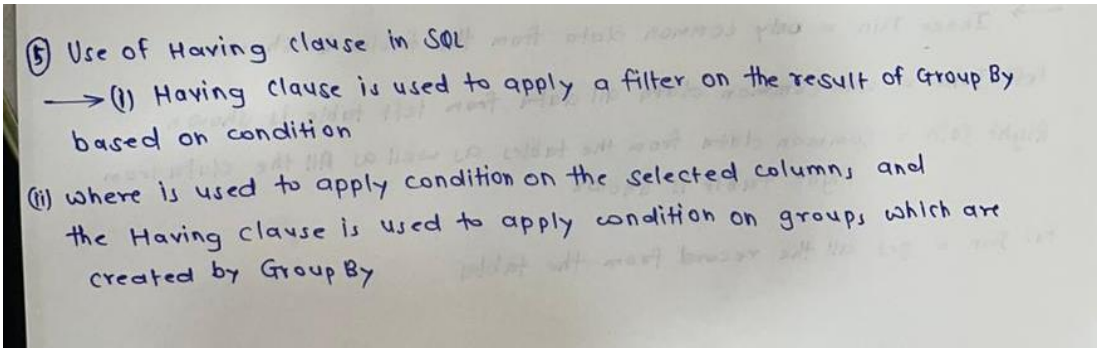
3. What do you understand by sub-queries in SQL?

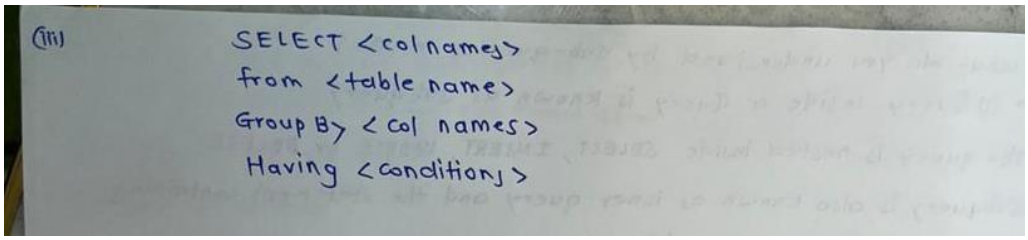


4. What is CTE in SQL?

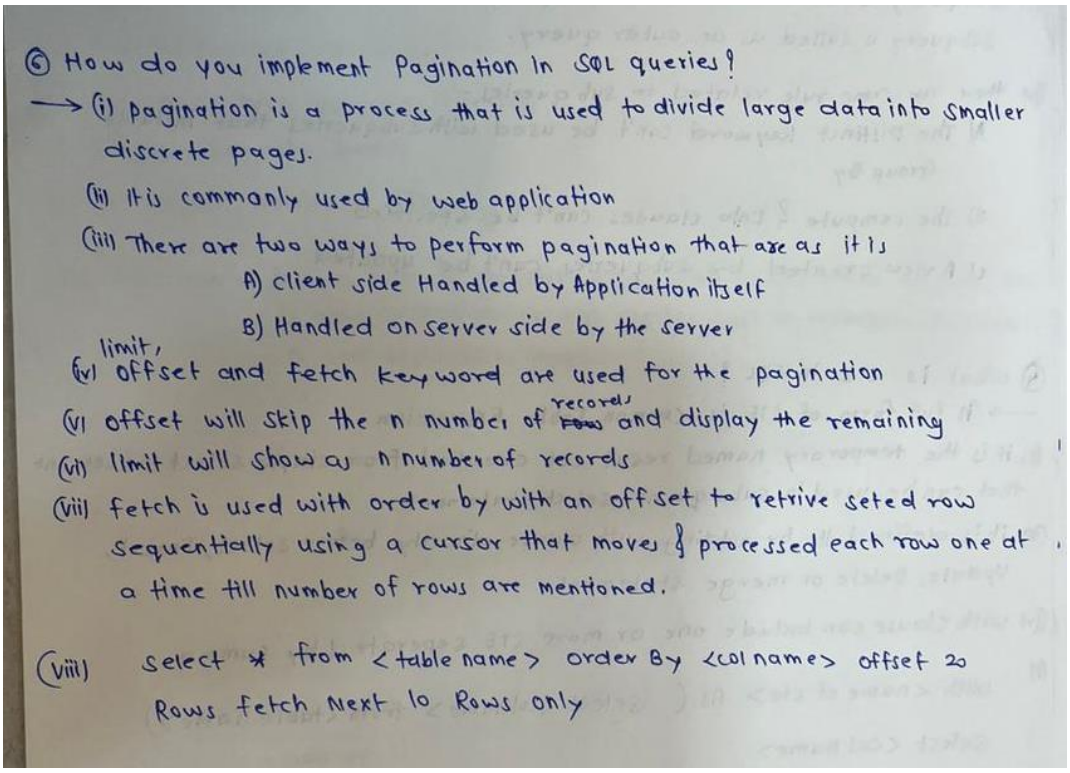


5. Explain the use of the HAVING clause in SQL.





6. How do you implement pagination in SQL queries?



⑥ How do you implement Pagination in SQL queries?

→ (i) pagination is a process that is used to divide large data into smaller discrete pages.

(ii) It is commonly used by web application

(iii) There are two ways to perform pagination that are as it is

A) client side Handled by Application itself

B) Handled on server side by the server

(iv) limit, offset and fetch keyword are used for the pagination

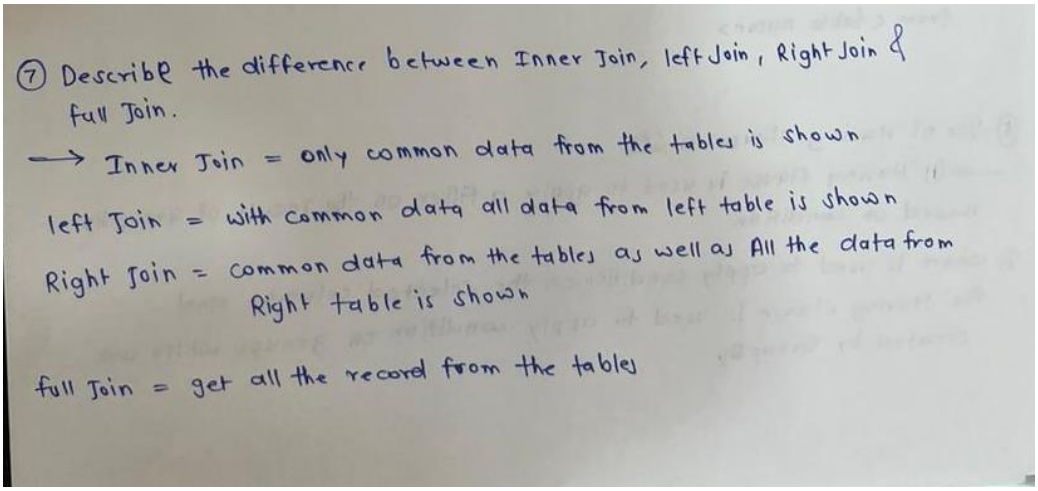
(v) offset will skip the n number of ~~row~~ records and display the remaining

(vi) limit will show as n number of records

(vii) fetch is used with order by with an offset to retrive seted row sequentially using a cursor that moves & processed each row one at a time till number of rows are mentioned.

(viii) Select * from <table name> order By <col name> offset 20 Rows fetch Next 10 Rows only

7. Describe the differences between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.



⑦ Describe the difference between Inner Join, left Join, Right Join & full Join.

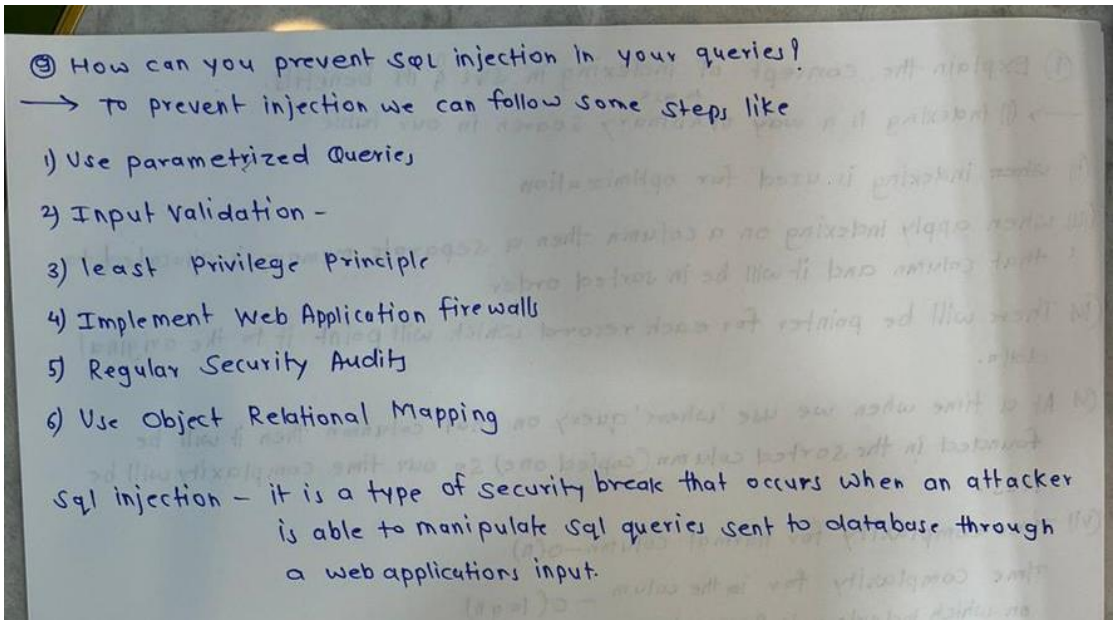
→ Inner Join = only common data from the tables is shown

left Join = with common data all data from left table is shown

Right Join = common data from the tables as well as All the data from Right table is shown

full Join = get all the record from the tables

9. How can you prevent SQL injection in your queries?



⑨ How can you prevent SQL injection in your queries?

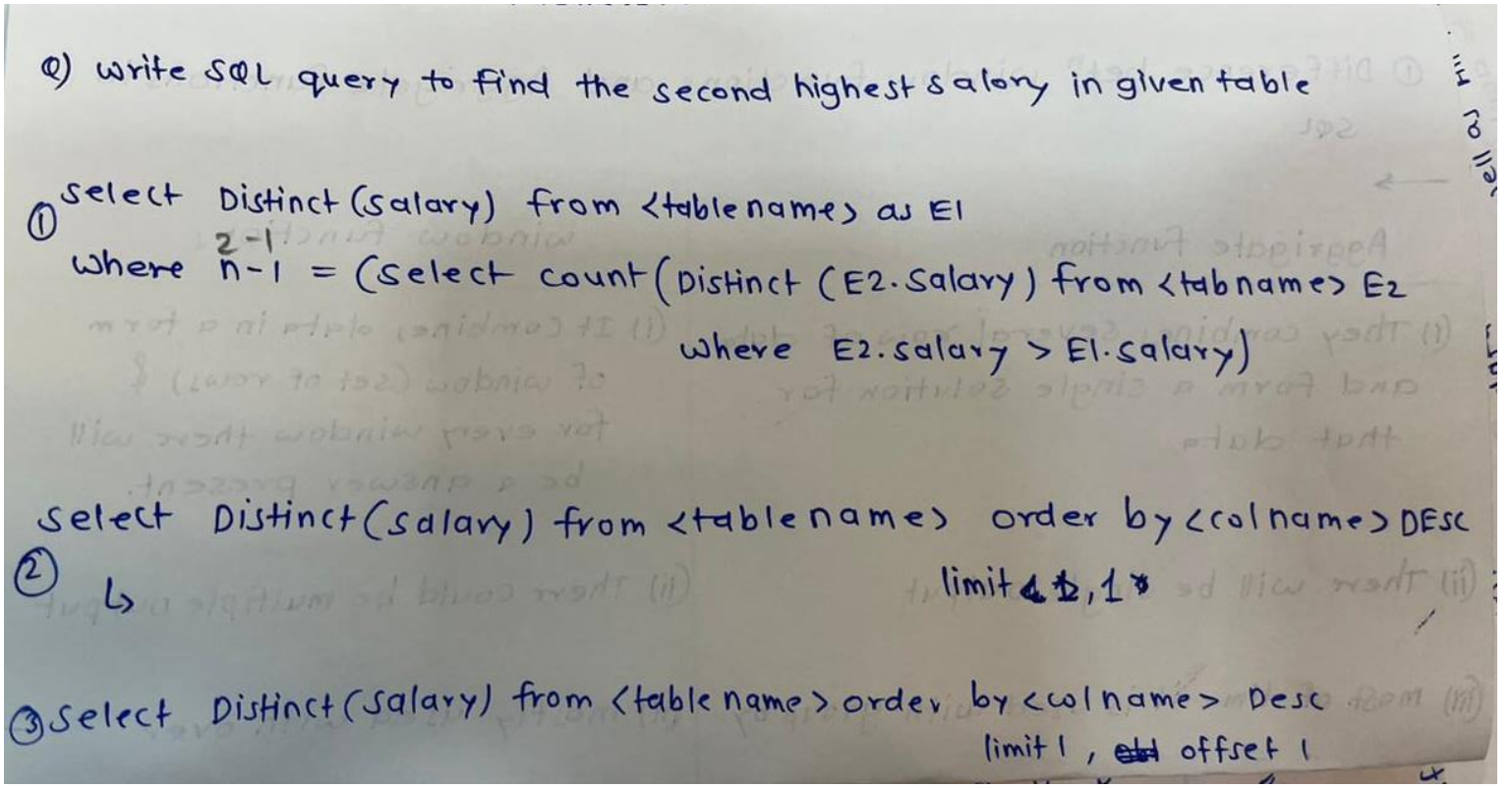
→ To prevent injection we can follow some steps like

1) Use parametrized Queries
2) Input validation -
3) least Privilege Principle
4) Implement Web Application firewalls
5) Regular Security Audits
6) Use Object Relational Mapping

Sql injection - it is a type of security break that occurs when an attacker is able to manipulate sql queries sent to database through a web applications input.

10 . Write a SQL query to find the second highest salary in

(done by three methods)



⑩ write SQL query to find the second highest salary in given table

① select Distinct (salary) from <tablename> as E1
where n-1 = (select count(Distinct (E2.Salary) from <tabname> E2
where E2.salary > E1.salary)

② select Distinct(salary) from <tablename> order by <colname> DESC
↳ limit 1,1

③ select Distinct(salary) from <table name> order by <colname> Desc
limit 1 , offset 1

8. Explain the concept of indexing in SQL and its benefits.

(8) Explain the concept of indexing in SQL & its benefits.

→ (i) indexing is a way of to use Binary Search in our table

(ii) ~~when~~ indexing is used for opHmization

(iii) when apply indexing on a column then a separate memory is allocated to that column and it will be in sorted order

(iv) There will be pointer for each record which will point it to the original data.

(v) At a time when we use 'where' query on that colunmn then it will be founded in the sorted column (copied one) so our time complaxity will be less.

(vi) Time complaxity for normal column - O(n)

Time complaxity for ~~is~~ the colum on which indexing is Applied — O(log n)

(vii) At final we can say

Indexing creates a lookup table with the column and the pointer to the memory location of the row, containing this column

(viii) B trees data structure is used in indexing

(ix) most of the time indexing is used in Read intensive Applications

~~(9) How can you prevent sql injection in your queries!~~

→ (x) Benifits of indexing -

A) faster Data Retrival

B) Improve Performance of WHERE clause

c) orderd Retrival

d) Opptimized Aggrigate function