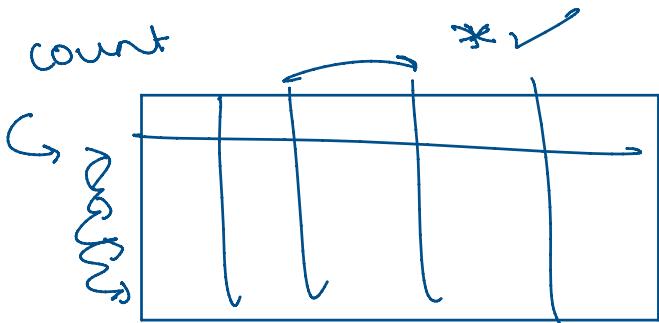


## # Aggregate function

# count



distinct

Krishna  
looks

Krishna  
1 time

## # min / max

min (released\_year)

order by released\_year  
asc  
limit 1

max (stock-quantity)

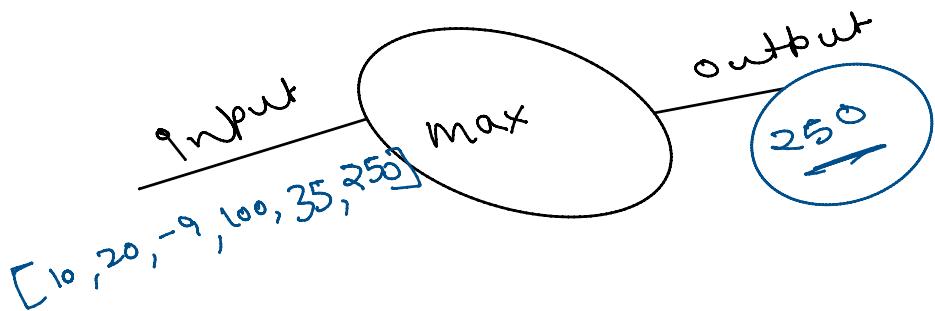
order by stock-quantity desc  
limit 1

input  
~ 9)

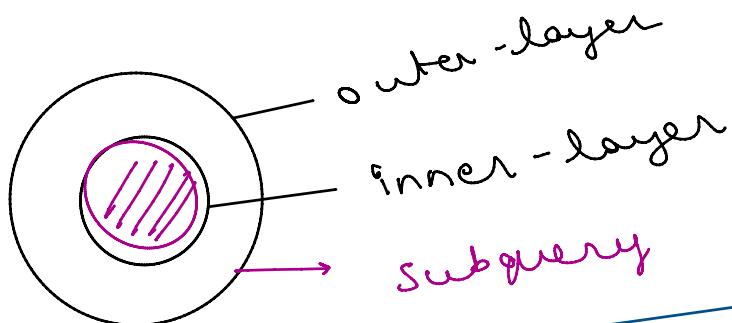
min

output

-2



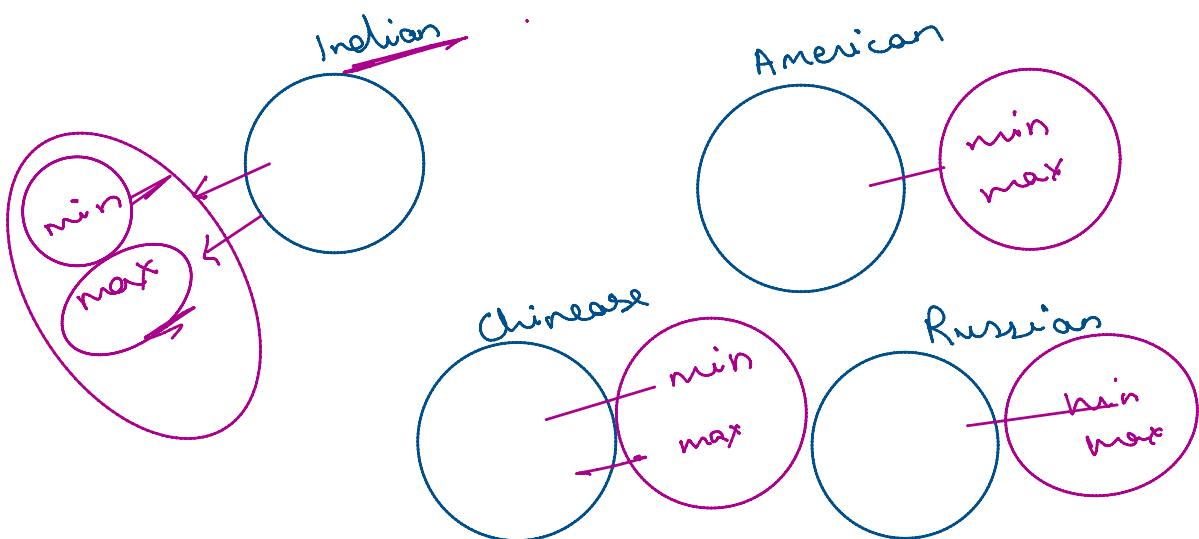
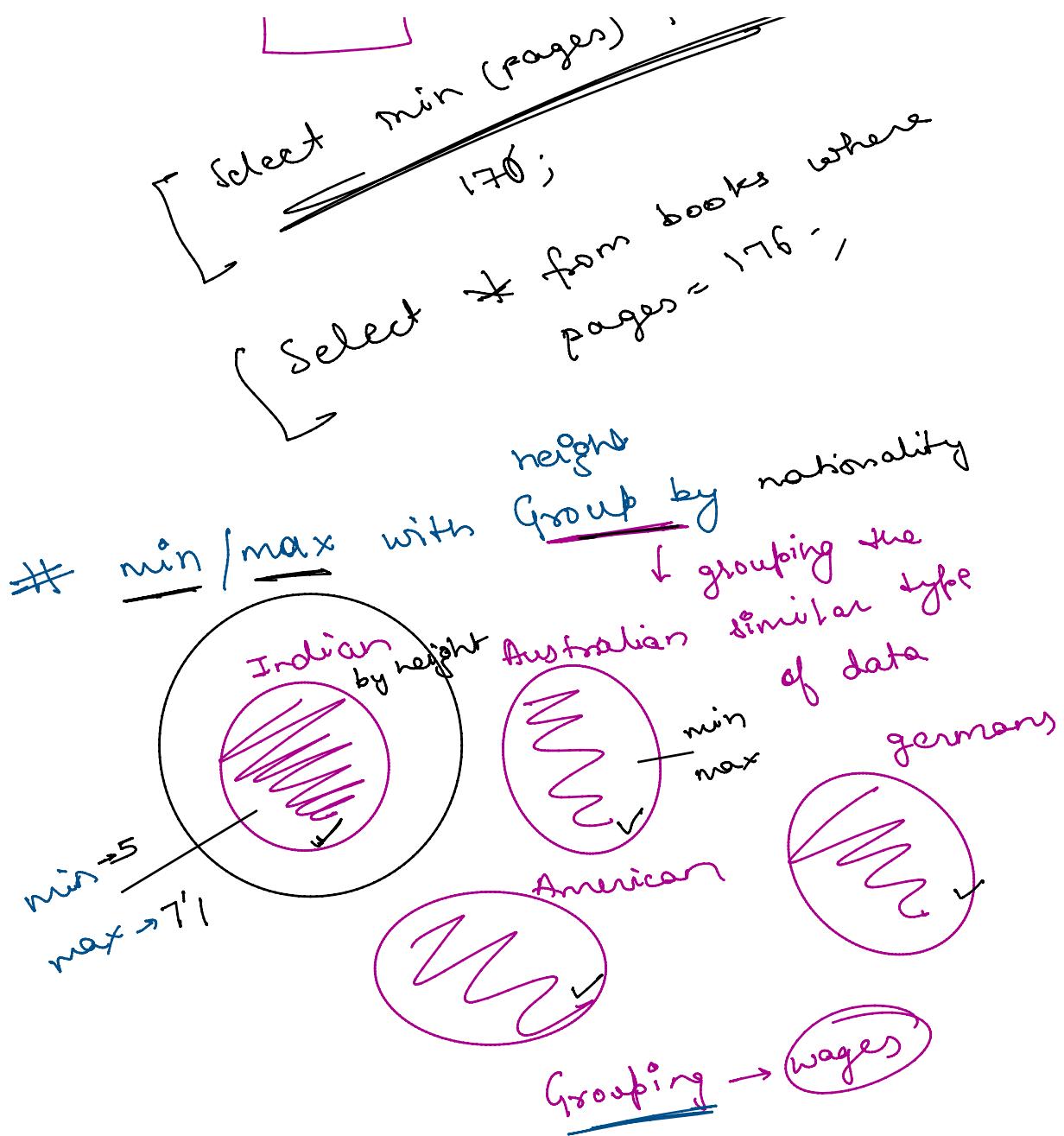
Subquery → query inside query.

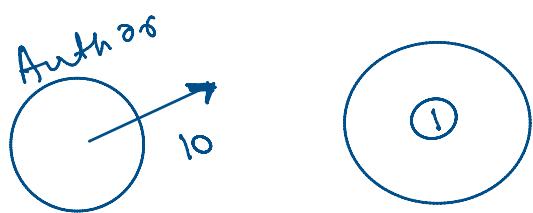
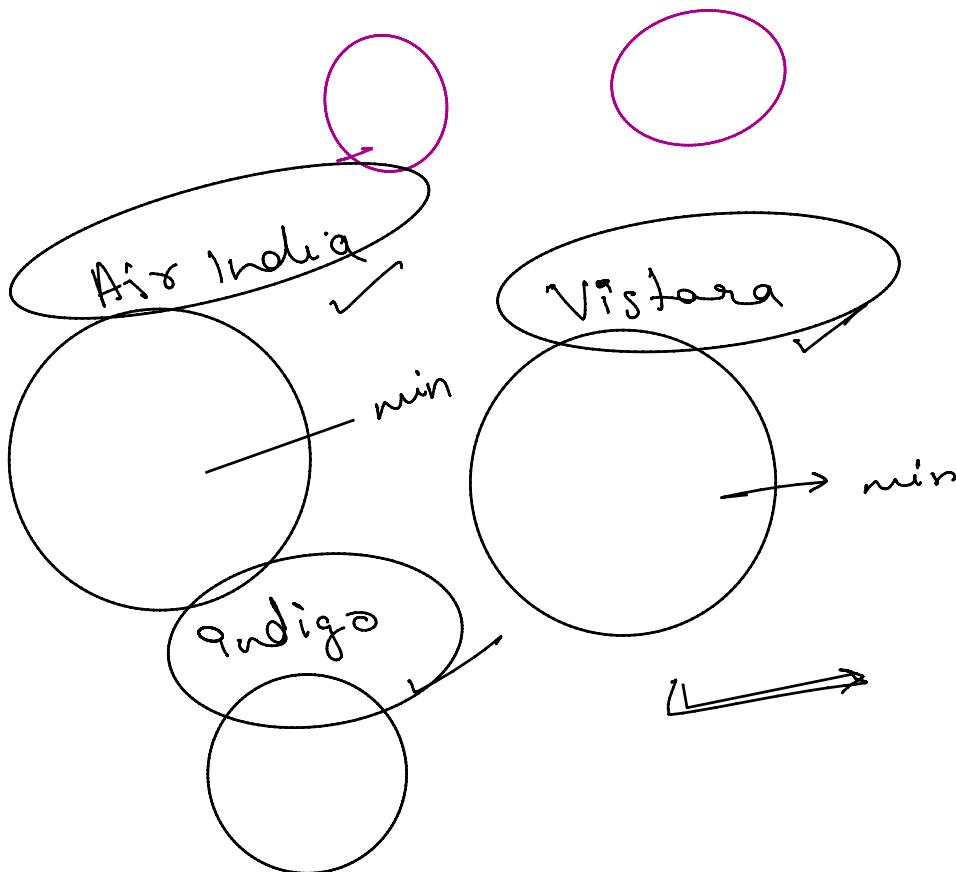
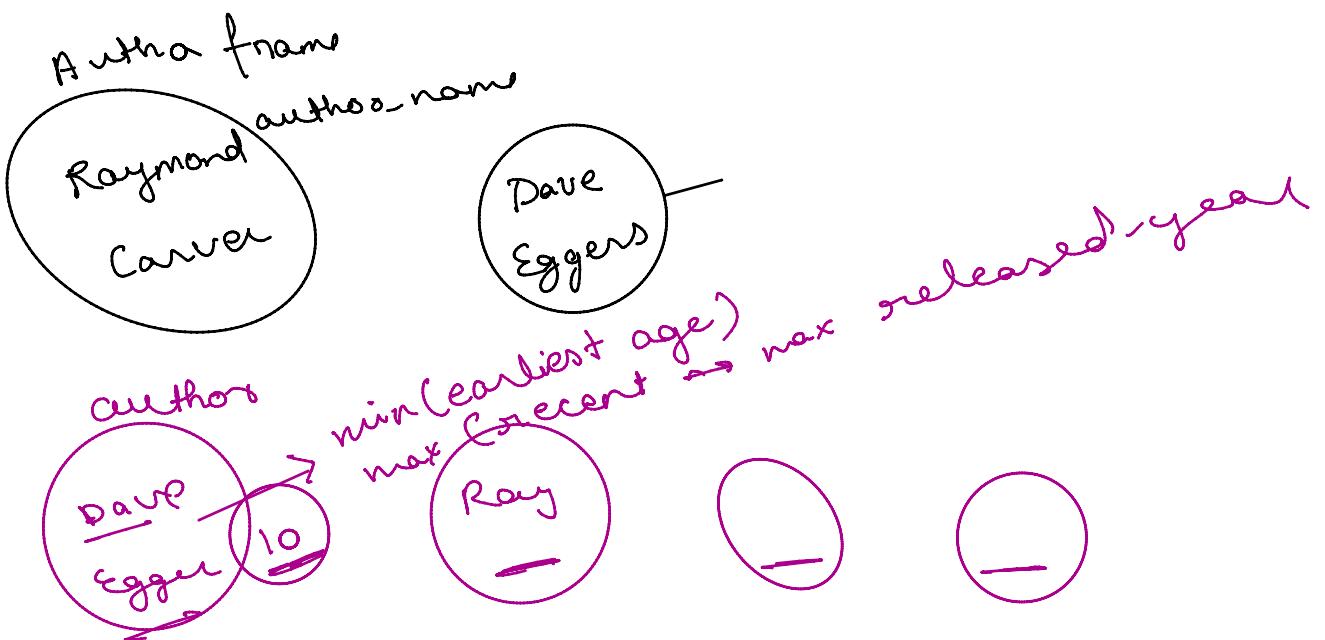


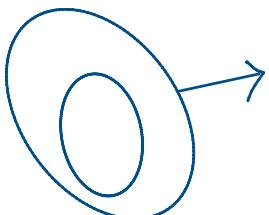
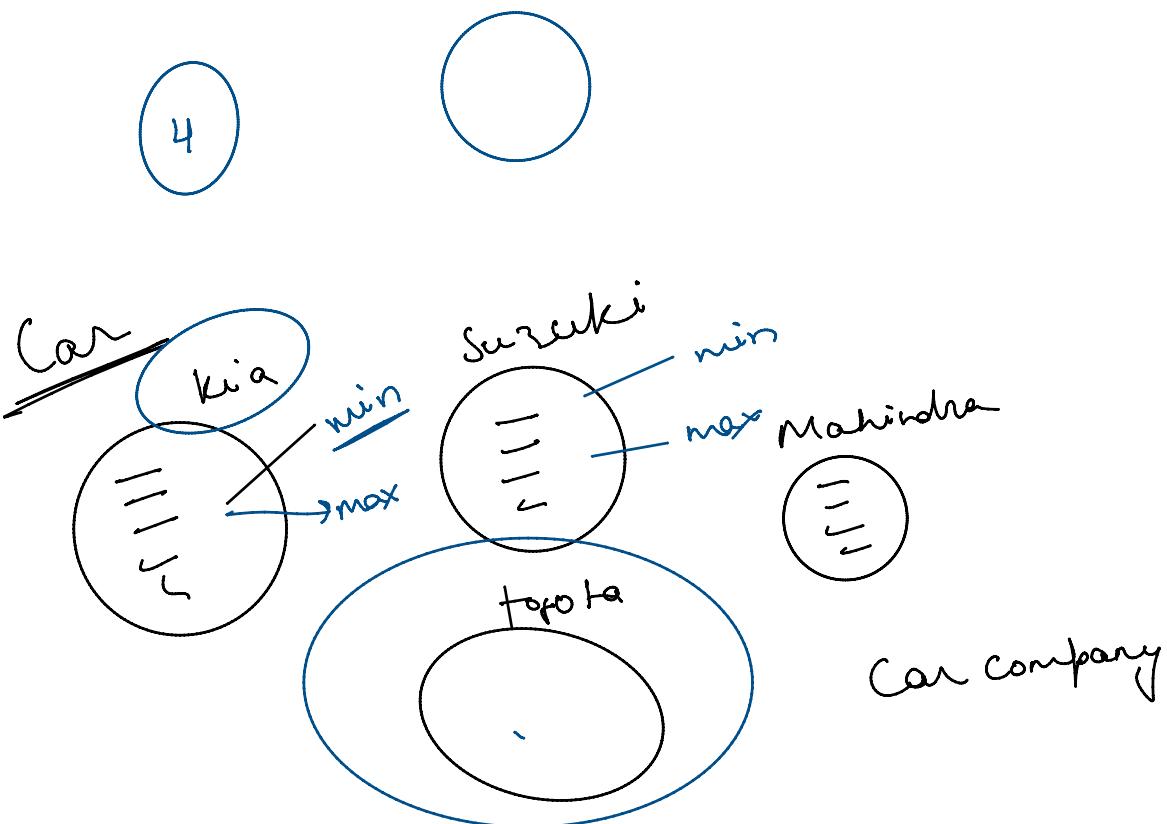
Eg: (Select \* from books where pages) = (select min(pages) from books) ↗ 176

select \* from books where pages = 111 ↗

millions  
in (pages) from books)

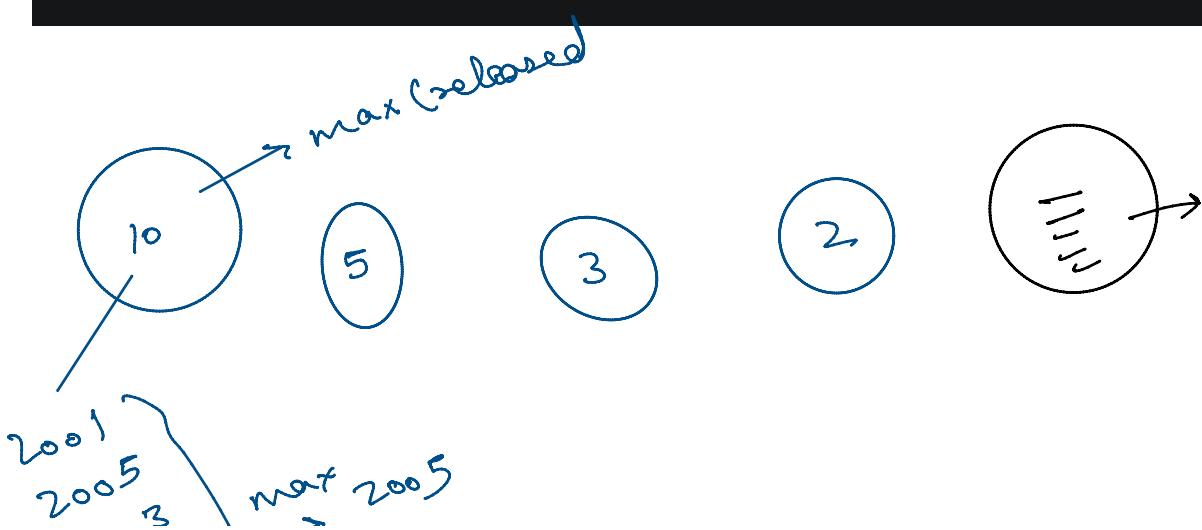


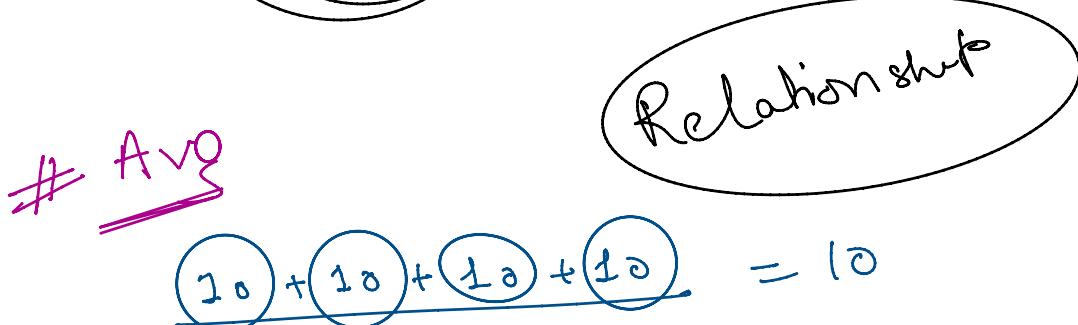
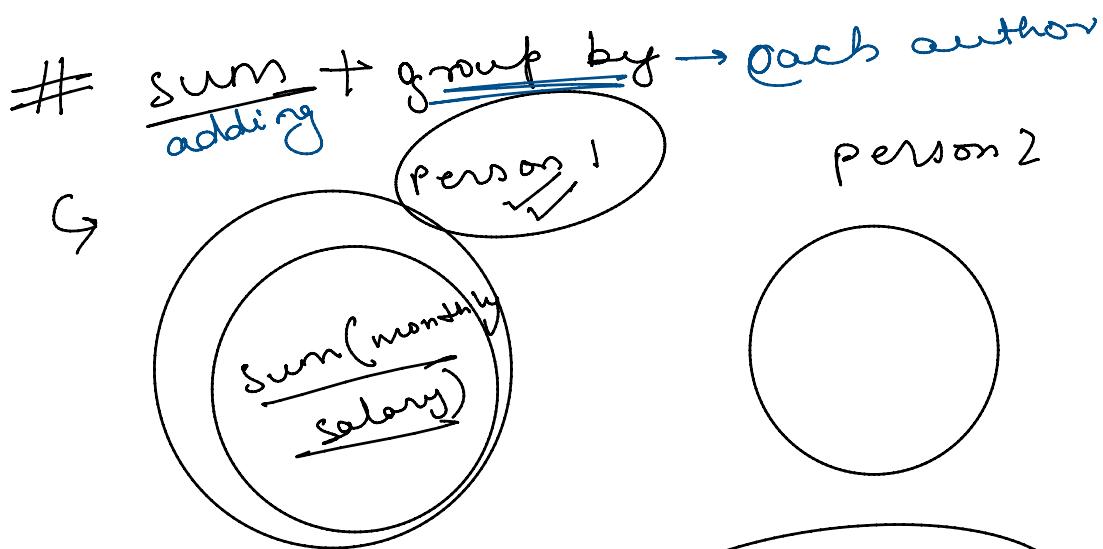
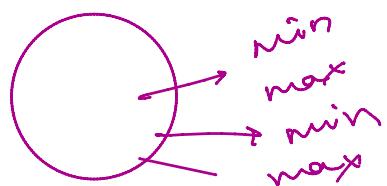
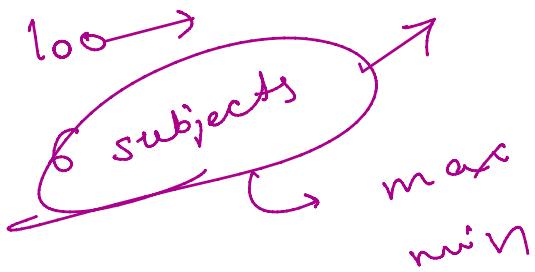
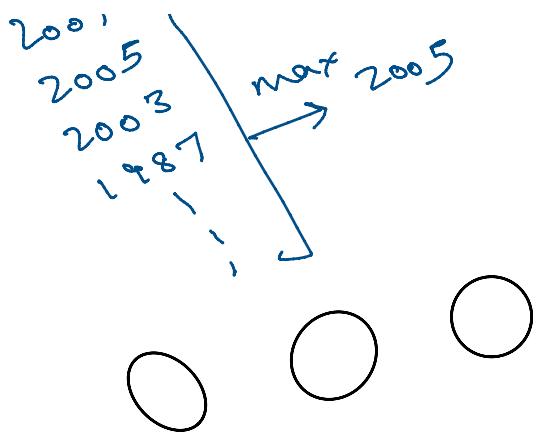




```
#Q2 find the year each author published their last book
SELECT author_fname , author_lname , max(released_year) from books
GROUP BY author_lname , author_fname;
```

grouping



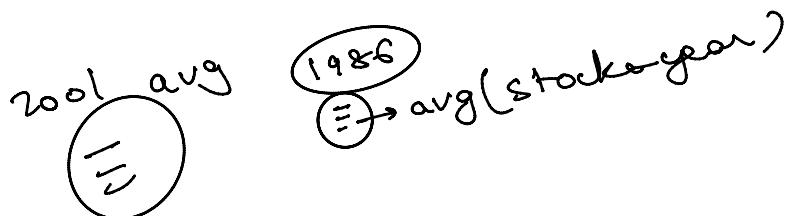
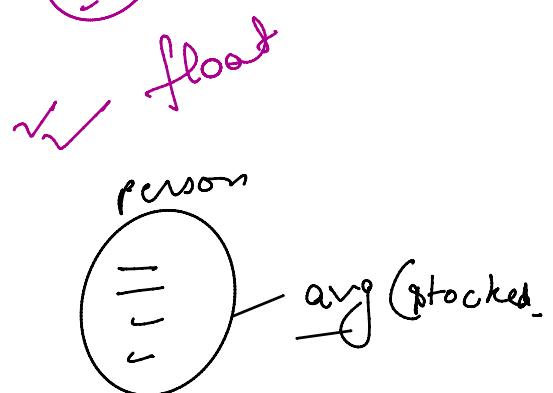
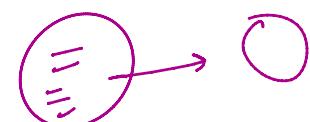
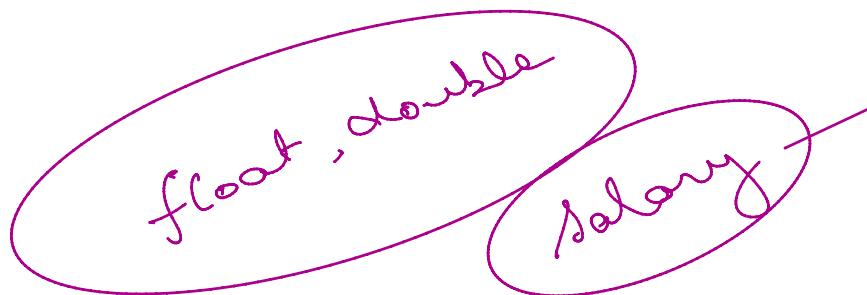
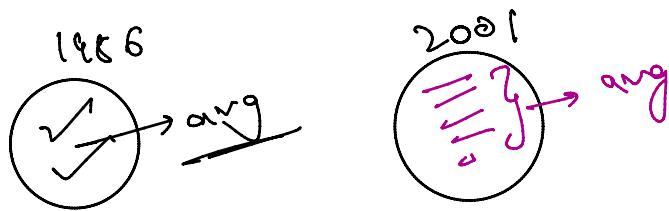


$$(10 + 10 + 10 + 10) = 10$$

4

$$\frac{\text{sum(10 values)}}{10} = -$$

group by  
released-year



## # Aggregate function challenges

① Print the number of books in the database

② Print out how many books were released in each year

③ Print out the total number

each year

Java + DSA

Print out the total number  
of books in stock ✓

Find the full name of the  
author who wrote the  
longest book

Find the average  
released\_year for each  
author

6 Make This Happen

year	# books	avg pages
1945	1	181.0000
1981	1	176.0000
1985	1	320.0000
1989	1	526.0000
1996	1	198.0000
2000	1	634.0000
2001	3	443.3333
2003	2	249.5000
2004	1	329.0000
2005	1	343.0000
2010	1	304.0000
2012	1	352.0000
2013	1	504.0000
2014	1	256.0000
2016	1	304.0000
2017	1	367.0000