Joins on same table

1. Here in our employee table….we have emp_id and manager_id..and they both are inter

```
select * from employee;
```

100 %

Results | Messages

	emp_id	emp_name	dept_id	salary	manager_id	emp_age
1	1	Ankit	100	10000	4	39
2	2	Mohit	100	15000	5	48
3	3	Vikas	100	10000	4	37
4	4	Rohit	100	5000	2	16
5	5	Mudit	200	12000	6	55
6	6	Agam	200	12000	2	14
7	7	Sanjay	200	9000	2	13
8	8	Ashish	200	5000	2	12
9	9	Mukesh	300	6000	6	51
10	10	Rakesh	500	7000	6	50
11	11	Ramesh	300	8000	6	52

related with each other

2. As we c in the pic..for emp_id = 1..the manager_id = 4..and manager_id = 4 is same as emp_id = 4
3. We can describe as…"who is the manager of emp_id = 1..it is emp_id = 4"...here emp_id 4 is the manager of emp_id = 1"..so ankits manager is rohit

```
select * from employee;
emp_name,manager_name
Ankit, rohit
mohit,mudit
```

4. For these kind of things..we use self join…here we are joining the same table on manager_id(from table1) and emp_id(from table2)..so after joining we get ..see pic col H

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	emp1										emp2					
2	emp_id	emp_name	dept_id	salary	manager_id	emp_age					emp_id	emp_nam	dept_id	salary	manager_id	emp_age
3	1	Ankit	100	10000	4	39	4	Rohit			1	Ankit	100	10000	4	39
4	2	Mohit	100	15000	5	48	5	Mudit			2	Mohit	100	15000	5	48
5	3	Vikas	100	10000	4	37	4	Rohit			3	Vikas	100	10000	4	37
6	4	Rohit	100	5000		16			(Ctrl)		4	Rohit	100	5000	2	16
7	5	Mudit	200	12000	6	55					5	Mudit	200	12000	6	55
8	6	Agam	200	12000	2	14					6	Agam	200	12000	2	14
9	7	Sanjay	200	9000	2	13					7	Sanjay	200	9000	2	13
10	8	Ashish	200	5000	2	12					8	Ashish	200	5000	2	12
11	9	Mukesh	300	6000	6	51					9	Mukesh	300	6000	6	51
12	10	Rakesh	500	7000	6	50					10	Rakesh	500	7000	6	50
13	11	Ramesh	300	8000	6	52					11	Ramesh	300	8000	6	52
14																
15					emp1.manager_id=emp2.emp_id											
16					emp_name	manager_name										
17					emp1.emp_name	emp.emp_name as manager_name										

```sql
;
select
e1.emp_id,e1.emp_name,e2.emp_name as manager_name
from
employee e1
inner join employee e2 on e1.manager_id=e2.emp_id
```

100 %

Results | Messages

	emp_id	emp_name	manager_name
1	1	Ankit	Rohit
2	2	Mohit	Mudit
3	3	Vikas	Rohit
4	4	Rohit	Mohit
5	5	Mudit	Agam
6	6	Agam	Mohit
7	7	Sanjay	Mohit
8	8	Ashish	Mohit
9	9	Mukesh	Agam
10	10	Rakesh	Agam
11	11	Ramesh	Agam

5. The above summary in sql
6. This types of questions are important for interview
7. Question can be like “tell the employees name whose salary is more than their manager salary”

```sql
select
e1.emp_id,e1.emp_name,e2.emp_name as manager_name
from
employee e1
inner join employee e2 on e1.manager_id=e2.emp_id
where e1.salary > e2.salary
```

Results | Messages

	emp_id	emp_name	manager_name
1	1	Ankit	Rohit
2	2	Mohit	Mudit
3	3	Vikas	Rohit

8.
9. The above query gives us the employee names whose salary is greater than their manager salary
10. Later while practicing..write ur own queries like..comparing age of employee and manager etc

Functions

1. We have seen aggregate functions like avg,sum,min,max,count to use on columns and then we can group by column

```
select dept_id,avg(salary) as avg_sal
from employee
group by dept_id
;
```

100 %

Results | Messages

	dept_id	avg_sal
1	100	10000
2	200	9500
3	300	7000
4	500	7000

2. Here we need all the employees names for each dept_id...like this

```
dept_id, list of emp
1, Ankit,monhit,vikas,rohit
```

this can be achieved using string aggregate functions..

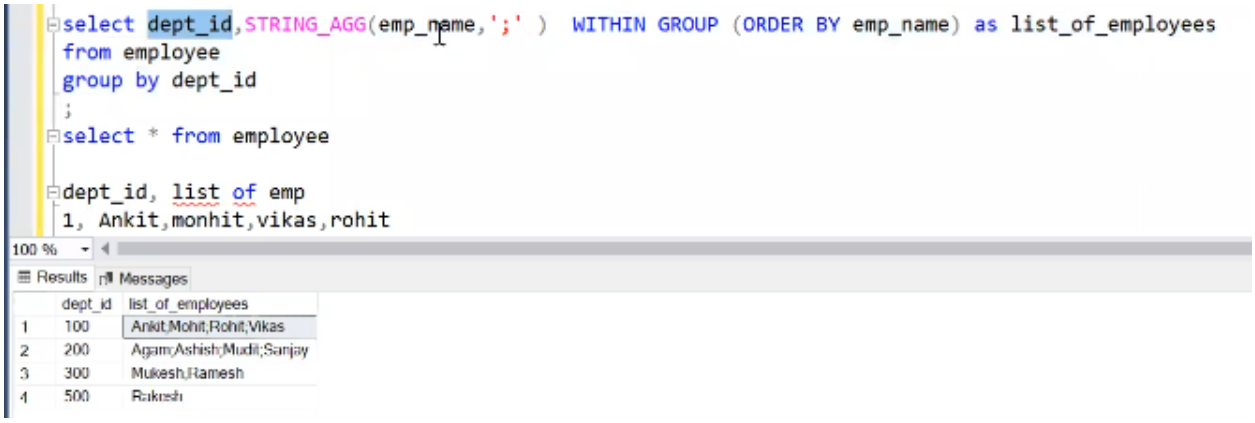
```
select dept_id,STRING_AGG(emp_name,';') as list_of_employees
from employee
group by dept_id
;
select * from employee

dept_id, list of emp
1, Ankit,monhit,vikas,rohit
```

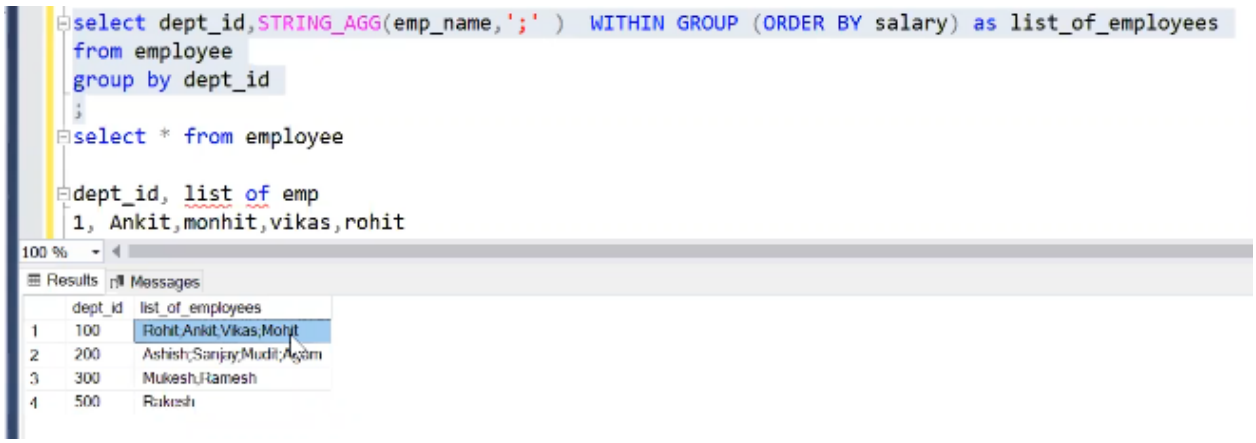
100 %

Results | Messages

	dept_id	list_of_employees
1	100	Ankit;Mohit;Vikas;Rohit
2	200	Mudit;Agam;Sanjay;Ashish
3	300	Mukesh;Ramesh
4	500	Rakesh

3. here we have used string agg function..to get employee names for each dept…and while using string_agg..we have to give separator…which separates our string values
4. List_agg and String_agg is same thing
5. We can also use order by while using String_agg

6. 

7. Here we have retrieved employees of each departments..and sorted by emp_name using order_by
8. 

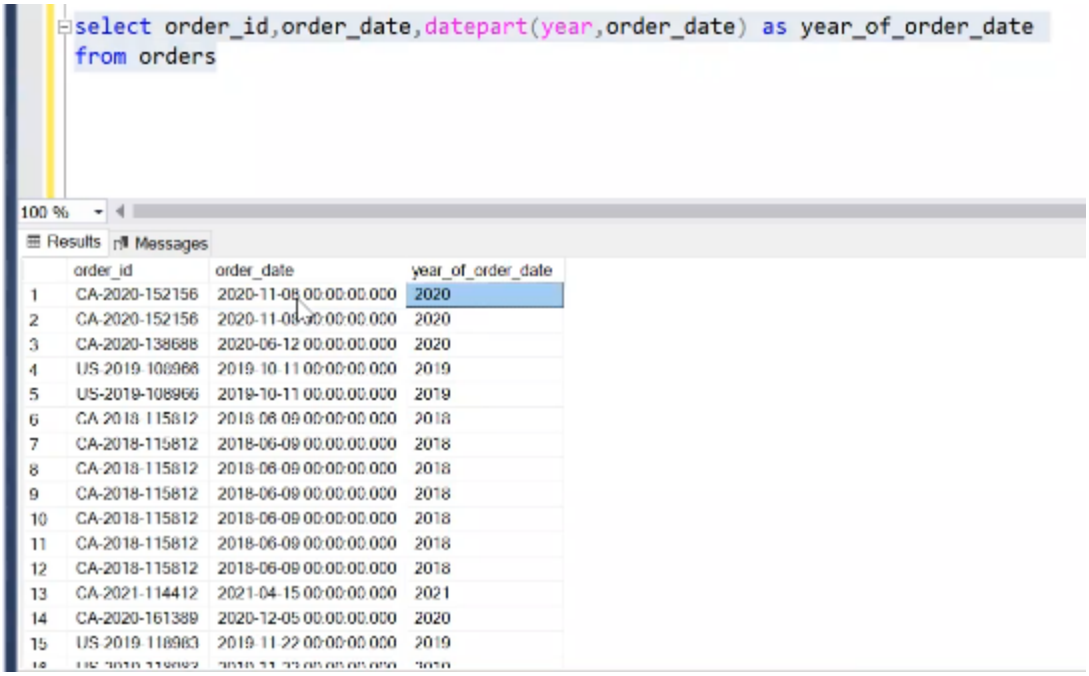
9. Here we have retrieved employees of each departments..and sorted by salaries using order_by …here we can also use DESC in order by

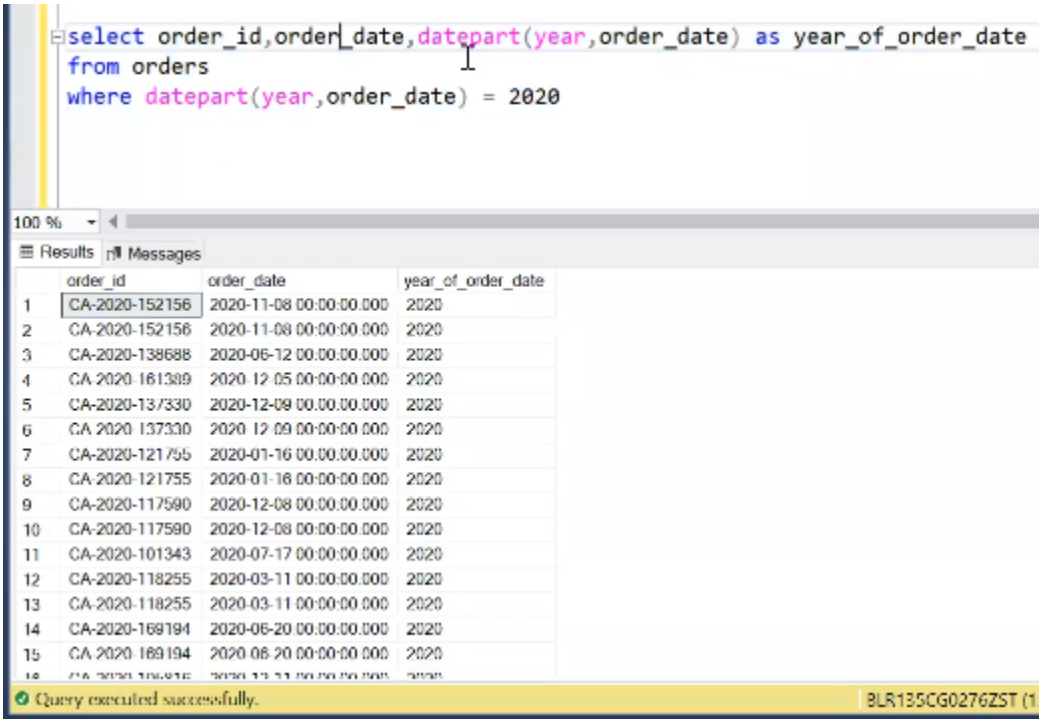
Date functions

1.
2. DatePart -

The DATEPART() function returns a specified part of a date.

This function returns the result as an integer value.

3. 

4. Here we have retrieved the year of order date using datepart…
5. As Datepart gives us a part of date..here we have part of date as year…and while using datepart we have specify column name too…see code
6. We can also filter the year using where
7. 

8. In the above SQL query..we have retrieved year of order date using datepart..where the year is 2020..see code and get intuition
9. Also refer : https://www.w3schools.com/sql/func_sqlserver_datepart.asp
10. Not only year we can also retrieve month part using datepart

11.

```sql
select order_id,order_date,datepart(year,order_date) as year_of_order_date
,datepart(month,order_date) as year_of_order_date
from orders
```

100 %

Results | Messages

	order_id	order_date	year_of_order_date	year_of_order_date
1	CA-2020-152156	2020-11-08 00:00:00.000	2020	11
2	CA-2020-152156	2020-11-08 00:00:00.000	2020	11
3	CA-2020-138688	2020-06-12 00:00:00.000	2020	6
4	US-2019-108966	2019-10-11 00:00:00.000	2019	10
5	US-2019-108966	2019-10-11 00:00:00.000	2019	10
6	CA-2018-115812	2018-06-09 00:00:00.000	2018	6
7	CA-2018-115812	2018-06-09 00:00:00.000	2018	6
8	CA-2018-115812	2018-06-09 00:00:00.000	2018	6
9	CA-2018-115812	2018-06-09 00:00:00.000	2018	6
10	CA-2018-115812	2018-06-09 00:00:00.000	2018	6
11	CA-2018-115812	2018-06-09 00:00:00.000	2018	6
12	CA-2018-115812	2018-06-09 00:00:00.000	2018	6
13	CA-2021-114412	2021-04-15 00:00:00.000	2021	4
14	CA-2020-161389	2020-12-05 00:00:00.000	2020	12
15	US-2019-118983	2019-11-22 00:00:00.000	2019	11

12. We can also get..the week of the year…check below code

13.

```sql
select order_id,order_date,datepart(yyyy,order_date) as year_of_order_date
,datepart(month,order_date) as month_of_order_date
,datepart(week,order_date) as week_of_order_date
from orders
```

100 %

Results | Messages

	order_id	order_date	year_of_order_date	month_of_order_date	week_of_order_date
1	CA-2020-152156	2020-11-08 00:00:00.000	2020	11	46
2	CA-2020-152156	2020-11-08 00:00:00.000	2020	11	46
3	CA-2020-138688	2020-06-12 00:00:00.000	2020	6	24
4	US-2019-108966	2019-10-11 00:00:00.000	2019	10	41
5	US-2019-108966	2019-10-11 00:00:00.000	2019	10	41
6	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23
7	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23
8	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23
9	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23
10	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23

14. And if we need the names of the months we can use DATENAME function..check below code

15.

```sql
select order_id,order_date,datepart(yy,order_date) as year_of_order_date
,datepart(month,order_date) as month_of_order_date
,datepart(week,order_date) as week_of_order_date
,DATENAME(month,order_date) as mont
from orders;
```

100 %

Results | Messages

	order_id	order_date	year_of_order_date	month_of_order_date	week_of_order_date	(No column name)
2	CA-2020-152156	2020-11-08 00:00:00.000	2020	11	46	November
3	CA-2020-138688	2020-06-12 00:00:00.000	2020	6	24	June
4	US-2019-108966	2019-10-11 00:00:00.000	2019	10	41	October
5	US-2019-108966	2019-10-11 00:00:00.000	2019	10	41	October
6	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23	June
7	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23	June
8	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23	June
9	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23	June
10	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23	June
11	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23	June
12	CA-2018-115812	2018-06-09 00:00:00.000	2018	6	23	June

16. We can also add more days or months to our dates which are present in our table(order_date,ship_date etc)

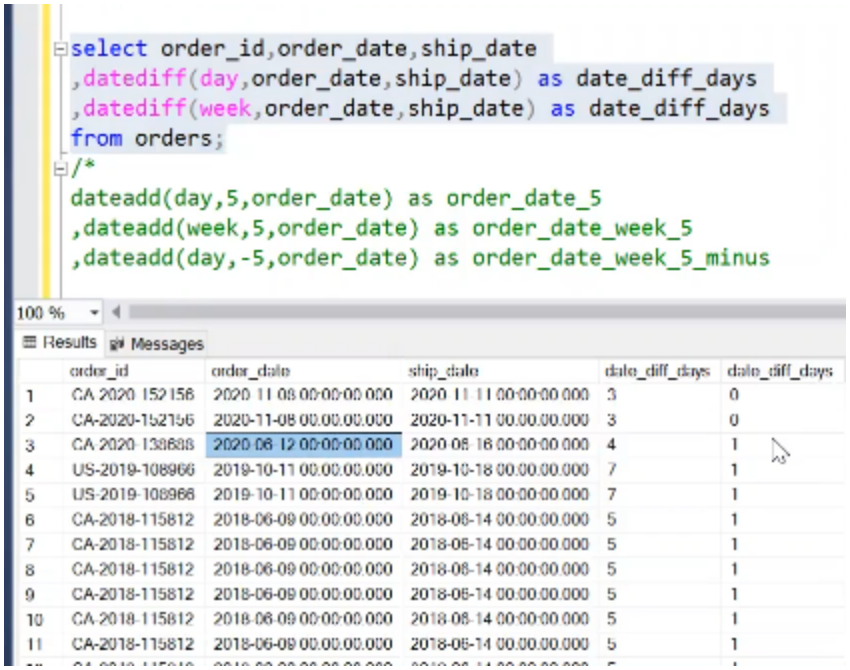
17.

```sql
select order_id,order_date,
dateadd(day,5,order_date) as order_date_5
,dateadd(week,5,order_date) as order_date_week_5
from orders;
```

18. In the above query ..we have added 5 days to our dates and 5 weeks to our weeks in order_date_column
19. We can use this query to go back 5 days..similarly we can use for weeks,months etc

```sql
,dateadd(day,-5,order_date) as order_date_week_5
```

20. We can also calculate difference of dates in days and weeks using datediff …check below code

21. 

```
select order_id,order_date,ship_date
,datediff(day,order_date,ship_date) as date_diff_days
,datediff(week,order_date,ship_date) as date_diff_days
from orders;
/*
dateadd(day,5,order_date) as order_date_5
,dateadd(week,5,order_date) as order_date_week_5
,dateadd(day,-5,order_date) as order_date_week_5_minus
```

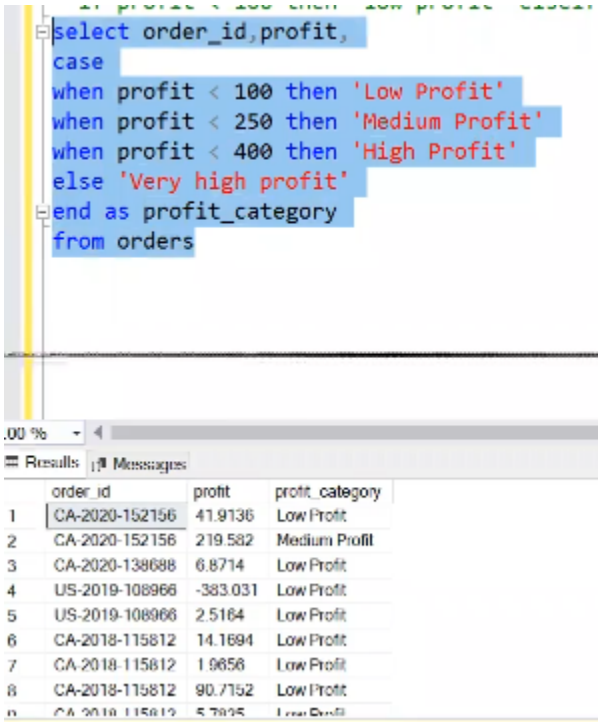
100 %

Results | Messages

	order_id	order_date	ship_date	date_diff_days	date_diff_days
1	CA-2020-152156	2020-11-08 00:00:00.000	2020-11-11 00:00:00.000	3	0
2	CA-2020-152156	2020-11-08 00:00:00.000	2020-11-11 00:00:00.000	3	0
3	CA-2020-138688	2020-06-12 00:00:00.000	2020-06-16 00:00:00.000	4	1
4	US-2019-108966	2019-10-11 00:00:00.000	2019-10-18 00:00:00.000	7	1
5	US-2019-108966	2019-10-11 00:00:00.000	2019-10-18 00:00:00.000	7	1
6	CA-2018-115812	2018-06-09 00:00:00.000	2018-06-14 00:00:00.000	5	1
7	CA-2018-115812	2018-06-09 00:00:00.000	2018-06-14 00:00:00.000	5	1
8	CA-2018-115812	2018-06-09 00:00:00.000	2018-06-14 00:00:00.000	5	1
9	CA-2018-115812	2018-06-09 00:00:00.000	2018-06-14 00:00:00.000	5	1
10	CA-2018-115812	2018-06-09 00:00:00.000	2018-06-14 00:00:00.000	5	1
11	CA-2018-115812	2018-06-09 00:00:00.000	2018-06-14 00:00:00.000	5	1

22. In the datediff..we have gave 2 columns which contains dates..see pic
23. These are our main 3 date function

Case when

1. Case when in SQL is similar to if conditions in programmings
2. See below code and understand
3. 

```
select order_id,profit,
case
when profit < 100 then 'Low Profit'
when profit < 250 then 'Medium Profit'
when profit < 400 then 'High Profit'
else 'Very high profit'
end as profit_category
from orders
```

.00 %

Results | Messages

	order_id	profit	profit_category
1	CA-2020-152156	41.9136	Low Profit
2	CA-2020-152156	219.582	Medium Profit
3	CA-2020-138688	6.8714	Low Profit
4	US-2019-108966	-383.031	Low Profit
5	US-2019-108966	2.5164	Low Profit
6	CA-2018-115812	14.1694	Low Profit
7	CA-2018-115812	1.9656	Low Profit
8	CA-2018-115812	90.7152	Low Profit
9	CA-2018-115812	5.7825	Low Profit

4. Here if observe at row 1 we have profit of 41.9136..which satisfies all the when conditions of profit…
5. So in SQL when conditions execute one after another(top to bottom) ..so if a value satisfies in one when condition..it stops and doesn't move to another when condition
6. If no when conditions satisfies ..it goes to else

```
select order_id,profit,
case
when profit < 250 then 'Medium Profit'
when profit < 100 then 'Low Profit'
when profit < 400 then 'High Profit'
else 'Very high profit'
end as profit_category
from orders
```

7.
8. If we execute the above command …there will be no low pr ofit cases…because everything goes to medium profit condition
9. We can also use filters and range in when conditions

```
when profit >= 100 and profit < 250 then 'Medium Profit'
```