Here in out orders table...if we want to find the avg of all sales...we just do avg(sales)
 from orders



2. if we have table like this

..then if calculate avg of sales..we

orde	r=,al	sales	
	1	1200	900
	2	600	

get 600..but in reality..it shud be

- 3. So here first we have to calculate the total sales of each order\_id..and then calculate avg of sales...
- 4. Here come the subqueries
- 5. Lets see this in SQL now...to find avg order sales from orders

6.

- Here first we will compute..sum of sales for each order\_id and we will name it as orders\_aggregated(this is our inner query)
- 8. Next from our orders\_aggregated table..we compute avg(order\_sales) from this table..see code and get intuition
- 9. Now we want to find order id with sales greater than avg sales

```
| select order_id
| from orders
| group by order_id
| having summ(sales) > (select avg(order_sales) as avg_order_value from
| (select order_id, sum(sales) as order_sales
| from orders
| group by order_id) as orders_aggregated)
```

10.

- 11. Here for our second subquery ..we didn't give any alias name..because in our 3rd query..we are just comparing the result with the second query..
- 12. And we have to give alias only if we using **from**...and here if we can see..there's no from keyword in our 3rd subquery

```
select * from employee
where dept_id not in (select dep_id from dept)
13.
```

choosing an employee dept

id..where the dep id is not in dept table

14. Here in this guery ..we used inner guery as condition..so there's no need of aliasing the inner query

```
select * from employee
    where dept_id not in (select dep_id from dept)
15.
                                                      try this query ..giving two columns in
   outer query and inner query(task for myself)
   select dept_id from employee
     except
    select dep_id from dept
16.
                                  try this..using * instead of dept_id & dep_id(task for
```

17. Subqueries with joins

mvself)

18.

```
select A.*,B.*
 (select order_id , sum(sales) as order_sales
from orders
group by order id ) A
inner join
(select avg(orders_aggregated.order_sales) as avg_order_value from
(select order_id, sum(sales) as order_sales
from orders
group by order_id) as orders_aggregated) B
on 1=1;
                                                                      here you can
```

from table a<sub>T</sub> inner join table b on col=col

select a.\*,b.

compare this query with this

```
select A.*,B.*
           from
           (select order_id , sum(sales) as order_sales
           from orders
           group by order_id ) A
           inner join
           (select avg(orders_aggregated.order_sales) as avg_order_value from
           (select order_id, sum(sales) as order_sales
           from orders
           group by order_id) as orders_aggregated) B
           on 1=1;
         ⊟select a.*,b.*
           from
      100 % - 4 |
      ⊞ Results ⊯ Messages
                      order sales avg order value
          order id
          CA-2018-100006 377.97 458-614665661807
          CA-2018-100090 699.192 458-014665661807
           CA-2018-100293 91.056
                                458.614665661807
          CA-2018-100328 3.928
                                 458.614865661807
           CA-2018-100363 21.376
                                458.614665661807
                                 458.614865681807
          CA-2018-100391 14.62
           CA-2018-100678 697.074
                                 458.614665661807
          CA 2018 100706 129 44
                                 458 614665681807
19. l
```

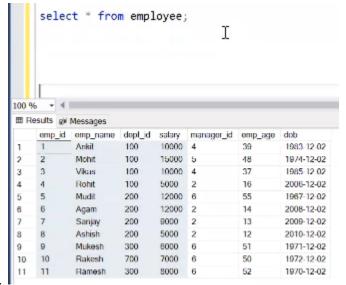
the output of

subqueries using inner join query

20. Order id with order sales greater than avg\_order\_sales

```
select A. , B. "
     from
     (select order_id , sum(sales) as order_sales
     from orders
     group by order_id ) A
     inner join
     (select avg(orders_aggregated.order_sales) as avg_order_value from
     (select order id, sum(sales) as order sales
     from orders
     group by order_id) as orders_aggregated) B
     on 1=1
     where order sales > avg order value;
                                                           Ι
   ⊟select a.*,b.*
100 % - 4
⊞ Results ⊯ Messages
                 order_sales avg_order_value
                           458.614665681807
    CA-2018-100090 699.192
     CA-2018-100678 697.074
                           458.614665661807
                           458.614665661807
    CA-2018-100762 508.62
    CA-2018-100895 605.47
                           458.614665661807
    CA-2018-100916 788.86
                           458 614665661807
     CA-2018-101560 542.34
                           458.614865681807
                           458.614665661807
     CA-2018-101602 803.96
    CA 2018 101931 1252 602 458 614665681807
```

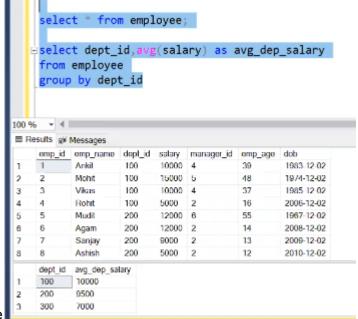
21. Here we have our employee table



23. Here we are trying to calculate the avg salaries for each department in a separate table..next we will join avg salaries table..with the employee table

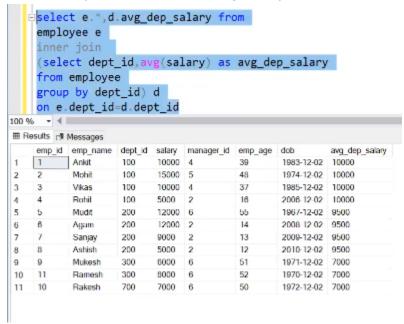
22.

24. Now here we have calculated avg dept salary..see pic..2nd guery is avg dep sal and first



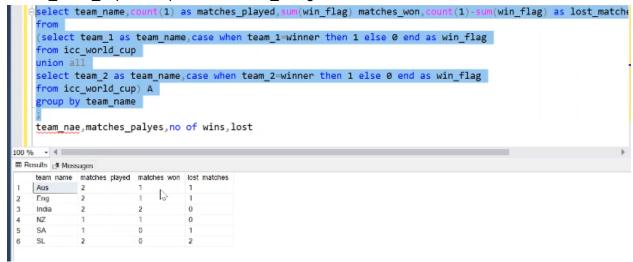
query is employee table

25. Now we will join this 2 queries using inner join

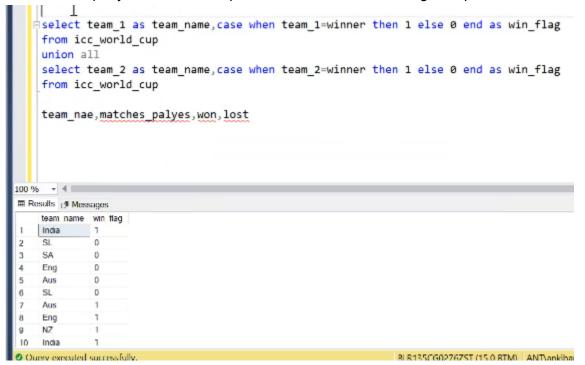


26. Basically imaging this subquery results as a table...here calculating avg salaries output is named as table d and employee table as table e...and we performed inner join on this two tables

27. For icc\_world\_cup table question KSQL7\_Assignment ..the answer is



28. In the above query ... first we have performed union all.. with using two queries



29. Now we will name this output table as A(sub query concept)..and from this table..we generate our required output...which is "team\_name,matches\_played,won,lost"

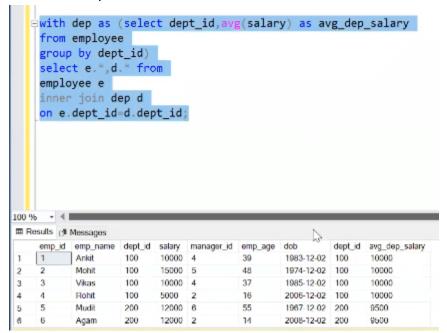
## CTE - common table expression

 The CTE's are similar to subqueries...but in CTE..we don't alias ..instead ..we use "with table\_name as" and perform the query...and store this result output table in table\_name

```
--cte common table expression
  with A as
  (select team_1 as team_name, case when team_1=winner then 1 else 0 end as win_flag
  from icc_world_cup
  union all
  select team_2 as team_name,case when team_2=winner then 1 else 0 end as win_flag
  from icc_world_cup)
  \mathsf{select} \mathsf{team\_name}, \mathsf{count}(*) as \mathsf{matches\_played}, \mathsf{sum}(\mathsf{win\_flag}) \mathsf{matches\_won}, \mathsf{count}(1) - \mathsf{sum}(\mathsf{win\_flag}) as \mathsf{lost\_matches}
  from A
                                                    Ι
  group by team_name
  - 4 |
Results of Messages
  team_name matches_played matches_won lost_matches
  Eng
  India
  NZ
  SA
```

- 3. After that we performed ..the last 3 lines of the SQL queries...using table\_name output
- 4. Another example of CTE

2.



- 5. Refer the previous example...and understand this
- 6. Structure wise..both the CTE and subqueries are same..but the way of writing is diff

```
with dep as (select dept_id,avg(salary) as avg_dep_salary
from employee
group by dept_id)
,total_salary as (select sum(salary) as ts from employee)
select e.*,d.* from
employee e
inner join dep d
on e.dept_id=d.dept_id;
```

- 7. Nested CTE example
- 8. Readability will be good in CTE compared to subqueries

10. This query with CTE would be

9.

```
Fwith order_wise_sales as (select order_id , sum(sales) as order_sales from orders group by order_id) select A.*,B.* from order_wise_sales A__ inner_join (select avg(orders_aggregated.order_sales) as avg_order_value from order_wise_sales as orders_aggregated) B on 1=1 where order_sales > avg_order_value;
```

11. Now in below highlighted query...we can use CTE there

```
dwith order_wise_sales as (select order_id , sum(sales) as order_sales
from orders
group by order_id)
,
diselect A.*,B.*
from
order wise sales A
inner join
[select avg(orders_aggregated.order_sales) as avg_order_value from
order_wise_sales as orders_aggregated) B
on 1=1
where order_sales > avg_order_value;
```

12.

13. It would look like this

- 14. It uses nested CTE's
- 15. Whatever can be done by subquery..can be done by CTE's as well
- 16. We have this query

```
select *,(select avg(salary) from employee) as avg_sal from employee
where dept_id in (select dep_id from dept);
```

18. Now this can also be