ORACLE SQL SESSIONS ASSIGNMENT -03

1.

select \*from(

select dense\_rank () over(order by dor.order\_total desc) as ranks,

dc.customer\_id,

dc.cust\_first\_name,

dc.cust\_last\_name,

dc.cust\_city,

ds.state\_name,

dc.phone\_number1,

dor.order\_total

from demo\_states ds

inner join demo\_customers dc on ds.st=dc.cust\_state

inner join demo\_orders dor on dc.customer\_id=dor.customer\_id

)

where ranks<=3;

2.

select \*from(

SELECT

row\_number() OVER (ORDER BY orders\_count DESC) AS row\_number,

customer\_id,

orders\_count,

cust\_first\_name,

cust\_last\_name,

cust\_city,

state\_name,

phone\_number1

FROM (

SELECT

DISTINCT dc.customer\_id,

COUNT(\*) OVER (PARTITION BY dc.customer\_id) AS orders\_count,

dc.cust\_first\_name,

dc.cust\_last\_name,

dc.cust\_city,

ds.state\_name,

dc.phone\_number1

FROM demo\_states ds

INNER JOIN demo\_customers dc ON ds.st = dc.cust\_state

INNER JOIN demo\_orders dor ON dc.customer\_id = dor.customer\_id

))

where row\_number<=3;

3.

select \*from(

select

row\_number() over(order by total\_quantity desc) as row\_number,

product\_id,

product\_name,

product\_description,

category,

total\_quantity

from(

select

distinct dp.product\_id,

dp.product\_name,

dp.product\_description,

dp.category,

count(\*)over(partition by doi.product\_id) as total\_quantity

from demo\_products dp

inner join demo\_order\_items doi on doi.product\_id=dp.product\_id

)

order by total\_quantity desc

)

where row\_number<=3;

4.

select

dc.cust\_last\_name,

dp.product\_name,

doi.quantity

from demo\_customers dc

inner join demo\_orders dor on dor.customer\_id=dc.customer\_id

inner join demo\_order\_items doi on dor.order\_id=doi.order\_id

inner join demo\_products dp on dp.product\_id=doi.product\_id

order by 1;

5.

select \*from(

select

dc.cust\_last\_name,

dp.product\_name,

doi.quantity

from demo\_customers dc

inner join demo\_orders dor on dor.customer\_id=dc.customer\_id

inner join demo\_order\_items doi on dor.order\_id=doi.order\_id

inner join demo\_products dp on dp.product\_id=doi.product\_id

order by 1

)

pivot(

max(quantity) for product\_name in(

'Business Shirt' as BUSINESS\_SHIRT,

'Trousers' AS TROUSERS,

'Jacket' AS JACKET,

'Blouse' AS BLOUSE,

'Skirt' AS SKIRT,

'Ladies Shoes' AS LADIES\_SHOES,

'Belt' AS BELT,

'Bag' AS BAG,

'Mens Shoes' AS MENS\_SHOES,

'Wallet' AS WALLET

)

);

6.

select \*from(

select

row\_number() over(order by year desc) as row\_number,

year,

orders\_count

from(

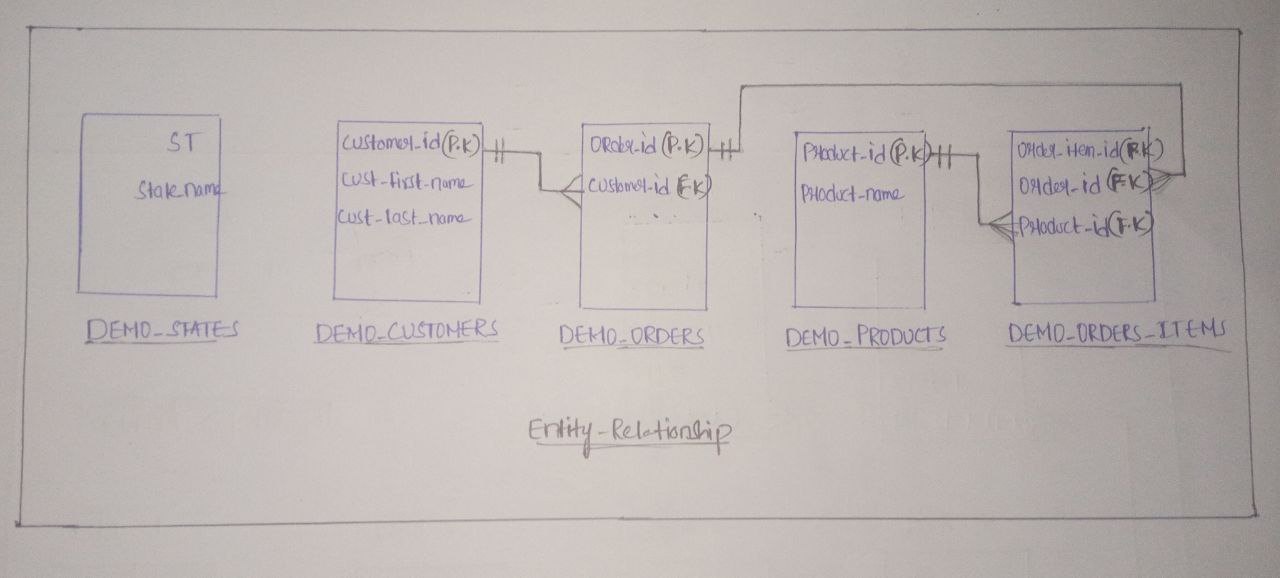
select distinct to\_char(order\_timestamp,'yyyy') as year,

count(\*) over(partition by to\_char(order\_timestamp,'yyyy')) as orders\_count

from demo\_orders do

))

where row\_number=1;

7.

8.

Select \*from(

select

row\_number() over(order by category\_count desc) as row\_number,

category,

category\_count

from(

select

distinct dp.category,

count(\*) over(partition by dp.category) as category\_count

from demo\_products dp

))

where row\_number=1;

9.

select \*from(

select

row\_number() over(order by quantity\_count desc) as row\_number,

category,

quantity\_count

from(

select

distinct dp.category,

sum(quantity) over(partition by dp.category) as quantity\_count

from demo\_products dp

inner join demo\_order\_items doi on dp.product\_id=doi.product\_id

))

where row\_number=2;

10.

select

distinct dc.customer\_id,

dp.product\_id,

sum(doi.quantity)

from demo\_customers dc

inner join demo\_orders dor on dor.customer\_id=dc.customer\_id

inner join demo\_order\_items doi on dor.order\_id=doi.order\_id

inner join demo\_products dp on dp.product\_id=doi.product\_id

group by rollup(dc.customer\_id,dp.product\_id);