```sql
--Which product has the highest price? Only return a single row

select top  1 *
from products
order by price desc ;
```

	product_id	product_name	price
1	13	Product M	70

```sql
 --Which customer has made the most orders?
 select c.customer_id , c.first_name , c.last_name ,
 count (order_id) as total_order
 from customers c
 join orders s
 on c.customer_id = s.customer_id
 group by c.customer_id , c.first_name , c.last_name
 order by total_order desc;
```

	customer_id	first_name	last_name	total_order
1	1	John	Doe	2
2	2	Jane	Smith	2
3	3	Bob	Johnson	2
4	4	Alice	Brown	1

```sql
--What's the total revenue per product?
select  p.product_id,p.product_name , sum (price * quantity) as total
from products p
join order_items o
on p.product_id = o.product_id
group by p.product_id , p.product_name
order by total desc ;
```

	product_id	product_name	total
1	13	Product M	420
2	10	Product J	330
3	6	Product F	210
4	12	Product L	195
5	11	Product K	180
6	3	Product C	160
7	9	Product I	150
8	2	Product B	135
9	8	Product H	135

```sql
--Find the day with the highest revenue
select  order_date ,sum (price * quantity) as total
from products p
join order_items o
on p.product_id = o.product_id
join orders s
on s.order_id = o.order_id
group by  order_date
order by total desc ;
```

	order_date	total
1	2023-05-16	340
2	2023-05-10	285
3	2023-05-11	275
4	2023-05-15	225
5	2023-05-13	185
6	2023-05-14	145
7	2023-05-08	145
8	2023-05-09	140
9	2023-05-07	85
10	2023-05-12	80

```sql
--Find the first order (by date) for each customer
select  s.customer_id , min (order_date ) as frist_order
from customers s
join orders d
on d.customer_id = s.customer_id
group by s.customer_id;
```

	customer_id	frist_order
1	1	2023-05-01
2	2	2023-05-02
3	3	2023-05-03
4	4	2023-05-07
5	5	2023-05-08
6	6	2023-05-09
7	7	2023-05-10
8	8	2023-05-11
9	9	2023-05-12
10	10	2023-05-13

```sql
--Find the top 3 customers who have ordered the most distinct products
select top 3 c.customer_id , c.first_name , c.last_name ,
count (distinct(t.product_id)) as distinct_products
from products p join  order_items  t
on p.product_id = t.product_id
join orders s
on s.order_id = t.order_id
join customers c
on c.customer_id = s.customer_id
group by c.customer_id ,c.first_name , c.last_name
order by distinct_products desc , c.customer_id
;
```

	customer_id	first_name	last_name	distinct_products
1	1	John	Doe	3
2	2	Jane	Smith	3
3	3	Bob	Johnson	3

```sql
--Which product has been bought the least in terms of quantity?
select product_name , sum (quantity) as total
from products p
join order_items t
on p.product_id = t.product_id
group by product_name
order by total asc ;
```

	product_name	total
1	Product D	3
2	Product E	3
3	Product G	3
4	Product H	3
5	Product I	3
6	Product K	3
7	Product L	3
8	Product A	5
9	Product F	6
10	Product J	6

```sql
--What is the median order total?
with t_order as (
select order_id , sum (price * quantity) as total ,
ROW_NUMBER() over (order by  sum(price * quantity) asc ) as asc_rank ,
ROW_NUMBER() over (order by sum(price * quantity) desc ) as desc_rank
from order_items d join products p
on d.product_id = p.product_id
group by order_id
)
select avg(total) as median_total
from t_order
where asc_rank in (desc_rank ,desc_rank -1 ,desc_rank +1 ) ;
```

	median_total
1	112.500000

```sql
--For each order, determine if it was 'Expensive' (total over 300), 'Affordable' (total over 100), or 'Cheap'.
select s.order_id ,
case
when  sum(price*quantity) > 300 then   'Expensive'
when  sum(price * quantity) > 100 then  'Affordable'
else 'cheap'
end as stutes
from orders s
join order_items t
on s.order_id = t.order_id
join products p
on p.product_id = t.product_id
group by s.order_id ;
```

	order_id	stutes
1	1	cheap
2	2	cheap
3	3	cheap
4	4	cheap
5	5	cheap
6	6	cheap
7	7	cheap
8	8	Affordable
9	9	Affordable
10	10	Affordable
11	11	Affordable
12	12	cheap
13	13	Affordable
14	14	Affordable
15	15	Affordable
16	16	Expensive

```sql
--Find customers who have ordered the product with the highest price.
with cte as
(
select c.customer_id , c.first_name,c.last_name, price ,
dense_rank () over(order by price desc) as ra_k
from customers c
join orders s
on c.customer_id = s.customer_id
join order_items t
on t.order_id = s.order_id
join products p
on p.product_id = t.product_id
 )
 select * from cte
 where ra_k =1 ;
```

	customer_id	first_name	last_name	price	ra_k
1	8	Ivy	Jones	70	1
2	13	Sophia	Thomas	70	1