```sql
--select * from retails;
--Q1. Display all columns for all transactions.
select * from retails;
--Display only the Transaction ID, Date, and Customer ID for all records
select `Transaction ID`,
  `Date`,
  `Customer ID`
 from
  retails;
 --Display all the distinct product categories in the dataset.
 select distinct `Product Category`
 from
 retails;
 -- Display all the distinct gender values in the dataset.
 select distinct
  `Gender`
 from
  retails;
  -- Display all transactions where the Age is greater than 40.
  select *
  from
   retails
  where
   `Age` > 40;
   -- Display all transactions where the Price per Unit is between 100 and 500.
   select *
   from
    retails
   where
    `Price per Unit` between 100 and 500;
    --Q7. Display all transactions where the Product Category is either 'Beauty' or
'Electronics'.
select *
from
retails
where `Product Category` in ('Beauty' , 'Electronics');
--Q8. Display all transactions where the Product Category is not 'Clothing'.
select *
from
retails
where `Product Category` != 'Clothing';
--Q9. Display all transactions where the Quantity is greater than or equal to 3.
select *
from
retails
where `Quantity` >= 3;
--Count the total number of transactions.
select count(*) as total_transactions
from
retails;
--Find the average Age of customers.
select avg(`Age`) as avg_age
from
retails;
```

```sql
--Q12. Find the total quantity of products sold.
select sum(`Quantity`) as total_quantity
from
retails;
--Q13. Find the maximum Total Amount spent in a single transaction.
select max(`Total Amount`) as max_total_amount
from
retails;
--Q14. Find the minimum Price per Unit in the dataset.
select min(`Price per Unit`) as min_price_per_unit
from
retails;
--Q15. Find the number of transactions per Product Category.
select `Product Category`, count(*) as num_transactions
from
retails
group by `Product Category`;
--Q16. Find the total revenue (Total Amount) per gender.
select `Gender`, sum(`Total Amount`) as total_revenue
from
retails
group by `Gender`;
--Q17. Find the average Price per Unit per product category.
select `Product Category`, avg(`Price per Unit`) as avg_price_per_unit
from
retails
group by `Product Category`;
--Q18. Find the total revenue per product category where total revenue is greater than 10,000.
select `Product Category`, sum(`Total Amount`) as total_revenue
from
retails
group by `Product Category`
having total_revenue > 10000;
--Q19. Find the average quantity per product category where the average is more than 2.
select `Product Category`,avg(`Quantity`) as average_quantity
 from retails
 group by `Product Category`
 having avg(`Quantity`) >2

--Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000,
otherwise 'Low'.
select *,
case
when `Total Amount` > 1000 then 'High'
else 'Low'
end as Spending_Level
from
retails;
--Q21. Display a new column called Age_Group that labels customers as: 'Youth' if Age < 30
'Adult' if Age is between 30 and 59 'Senior' if Age >= 60
select *,
case
when `Age`<30 then 'Youth'
when `Age` between 30 and 59 then 'Adult'
```

```
else 'Senior'
end as Age_Group
 from
 retails;
```