```sql
create database walmartt;

use walmartt;

select * from wm;

-- add time for the day to get clarity
alter table wm add time_of_day varchar(40);

--updateing time_of_day column
update wm set time_of_day =(
case
    when time between '00:00:00' and '12:00:00' then 'Morning'
    when time between '12:01:00' and '16:00:00' then 'afternoon'
    else 'evening'
end
);

--add day column
alter table wm add dayn varchar(30);

--updating values in dayn

update wm set dayn
= datename(weekday,date);


--add month column

alter table wm add months varchar(30);

--updating values

update wm set months=format(date,'MMMM');

-- How many unique cities does the data have?

SELECT COUNT(DISTINCT(CITY)) from wm;

-- In which city is each branch?

select distinct(City),branch from wm;

-- How many unique product lines does the data have?
select count(distinct(product_line)) from wm;

-- What is the most selling product line?
select sum(Quantity)as total_sold,
product_line from wm group by
```

```
product_line order by total_sold desc;

-- What is the total revenue by month?
select months,round(sum(total),2) as Revenue from wm
group by months order by revenue;

-- What month had the largest COGS?
select months ,Round(sum(cogs),2) as Total_sold_cogs
from wm group by months order by total_sold_cogs desc
;

-- What product line had the largest revenue?

select product_line,round(sum(total),2) as Total  from wm
group by product_line order by total desc;

-- What is the city with the largest revenue?
select branch,City,round(sum(total),2) as Total  from wm
group by city ,branch order by total desc;

-- What product line had the largest VAT?
select Product_line,round(sum(Tax_5),2) as Total  from wm
group by product_line order by total desc;

select * from wm;
-- Fetch each product line and add a column to those product
-- line showing "Good", "Bad". Good if its greater than average sales

select avg() from wm;

select product_line,
case
     when round(avg(quantity),2) >=6 then 'good'
     else 'bad'
     end as 'status'
     from wm
group by product_line
;

-- Which branch sold more products than average product sold?

select round(avg(quantity),2) from wm;

select branch,round(sum(quantity),2) as productsold from wm
 group by branch
 having round(sum(quantity),2) >(select round(avg(quantity),2));

 -- What is the most common product line by gender?
```

```sql
 SELECT
    gender,
    product_line,
    COUNT(gender) AS total_cnt
FROM wm
GROUP BY gender, product_line
ORDER BY total_cnt DESC;


-- What is the average rating of each product line
SELECT
    ROUND(AVG(rating), 2) as avg_rating,
    product_line
FROM wm
GROUP BY product_line
ORDER BY avg_rating DESC;

-- How many unique customer types does the data have?
SELECT customer_type,
    count(DISTINCT (customer_type))
FROM wm;

-- How many unique payment methods does the data have?
SELECT
    DISTINCT payment
FROM wm;


-- What is the most common customer type?
SELECT
    customer_type,
    count(*) as count
FROM wm
GROUP BY customer_type
ORDER BY count DESC;



-- Which customer type buys the most?
SELECT
    customer_type,
    COUNT(*)
FROM wm
GROUP BY customer_type;
-- What is the gender of most of the customers?
SELECT
    gender,
    COUNT(*) as gender_cnt
FROM wm
```

```sql
GROUP BY gender
ORDER BY gender_cnt DESC;

-- What is the gender distribution per branch?
SELECT branch,
    gender,
    COUNT(gender) as gender_cnt
FROM wm
GROUP BY gender,branch
ORDER BY gender_cnt DESC;


-- Which time of the day do customers give most ratings?
SELECT
    time_of_day,
    AVG(rating) AS avg_rating
FROM wm
GROUP BY time_of_day
ORDER BY avg_rating DESC;

-- Looks like time of the day does not really affect the rating, its
-- more or less the same rating each time of the day.alter
-- Which day fo the week has the best avg ratings?
SELECT
    dayn,
    AVG(rating) AS avg_rating
FROM wm
GROUP BY dayn
ORDER BY avg_rating DESC;
select * from wm;
-- Which of the customer types brings the most revenue?
SELECT
    customer_type,
    SUM(total) AS total_revenue
FROM wm
GROUP BY customer_type
ORDER BY total_revenue;
```