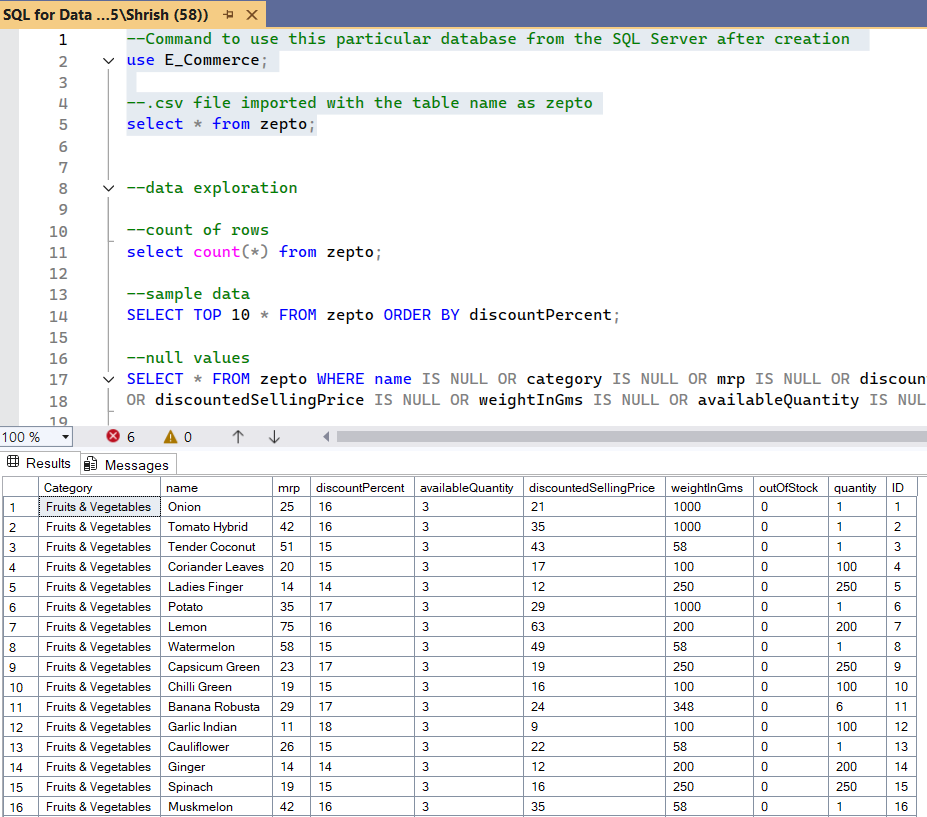
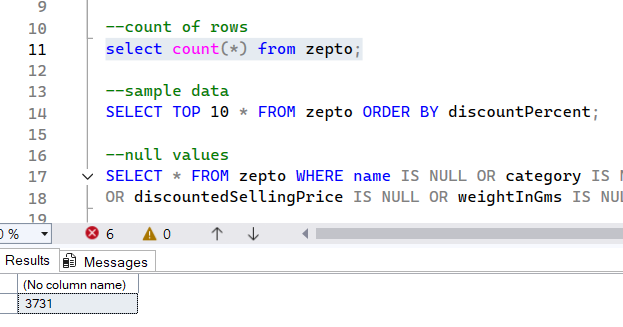
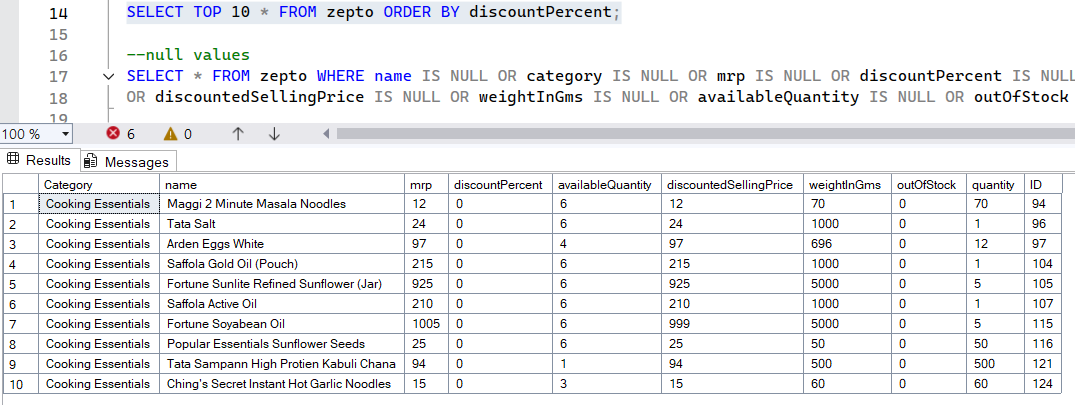
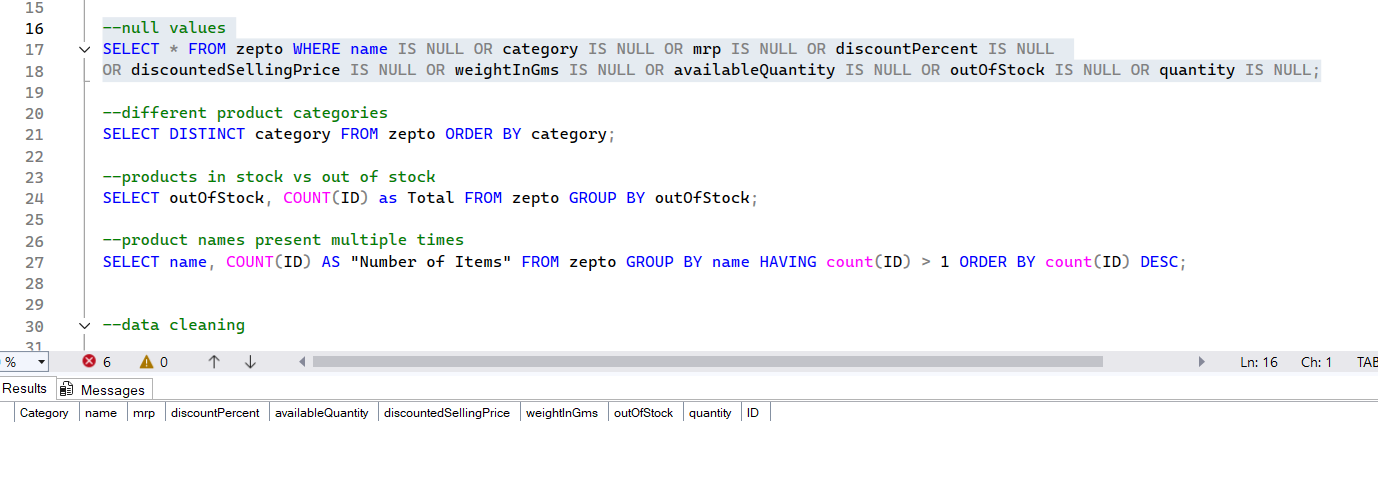
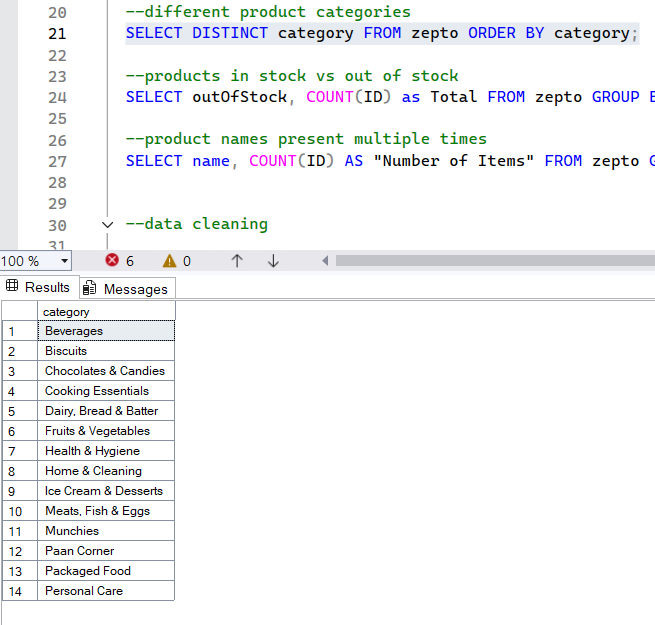
use E\_Commerce;  
select \* from zepto;  
  


select count(\*) from zepto;



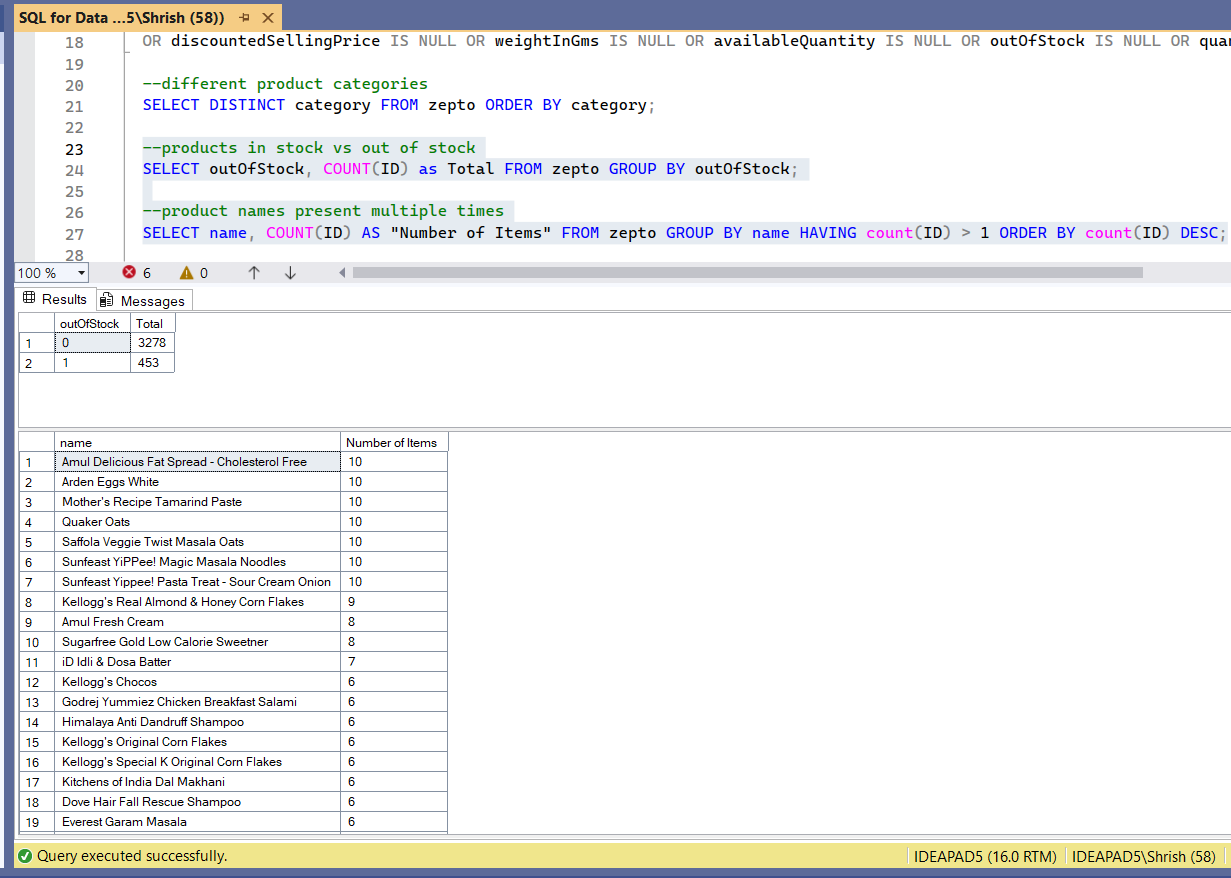
SELECT TOP 10 \* FROM zepto ORDER BY discountPercent;  


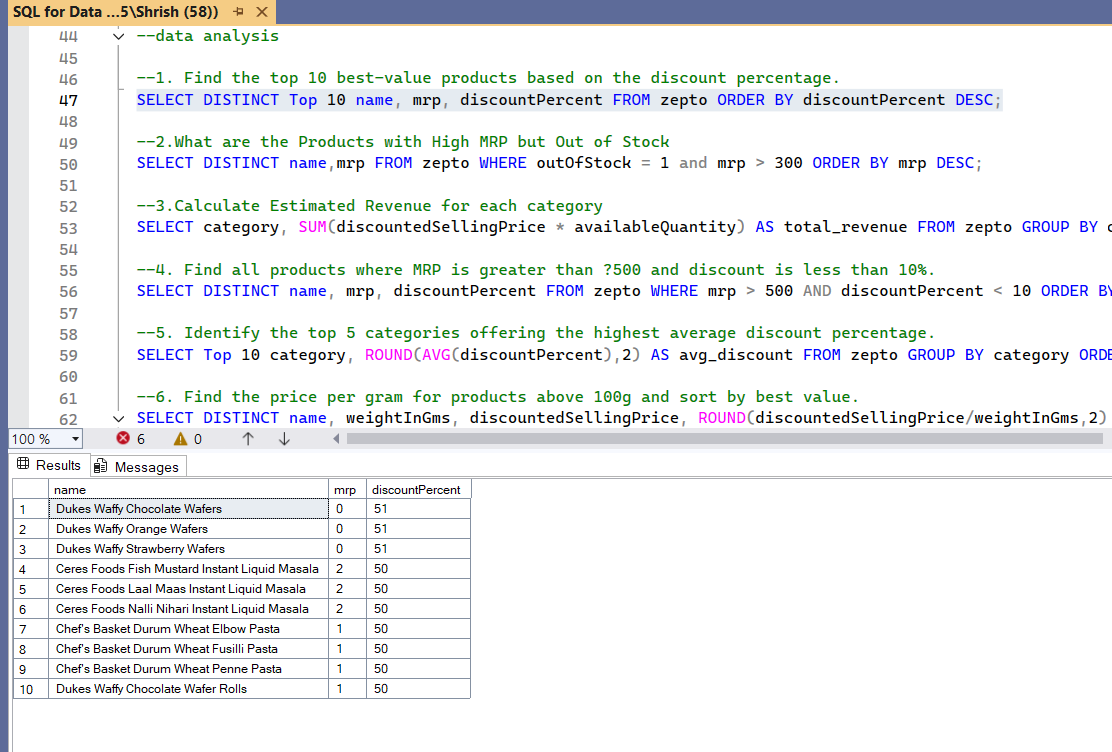
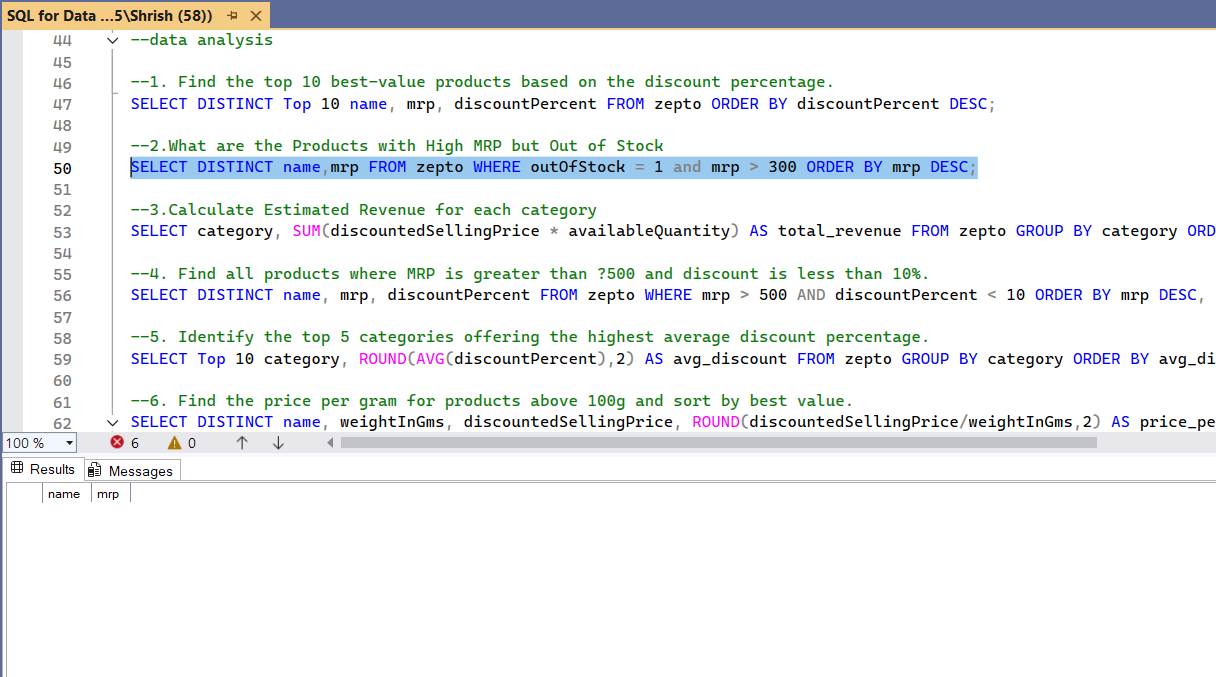
SELECT \* FROM zepto WHERE name IS NULL OR category IS NULL OR mrp IS NULL OR discountPercent IS NULL OR discountedSellingPrice IS NULL OR weightInGms IS NULL OR availableQuantity IS NULL OR outOfStock IS NULL OR quantity IS NULL;  
  
  
  
  
SELECT DISTINCT category FROM zepto ORDER BY category;

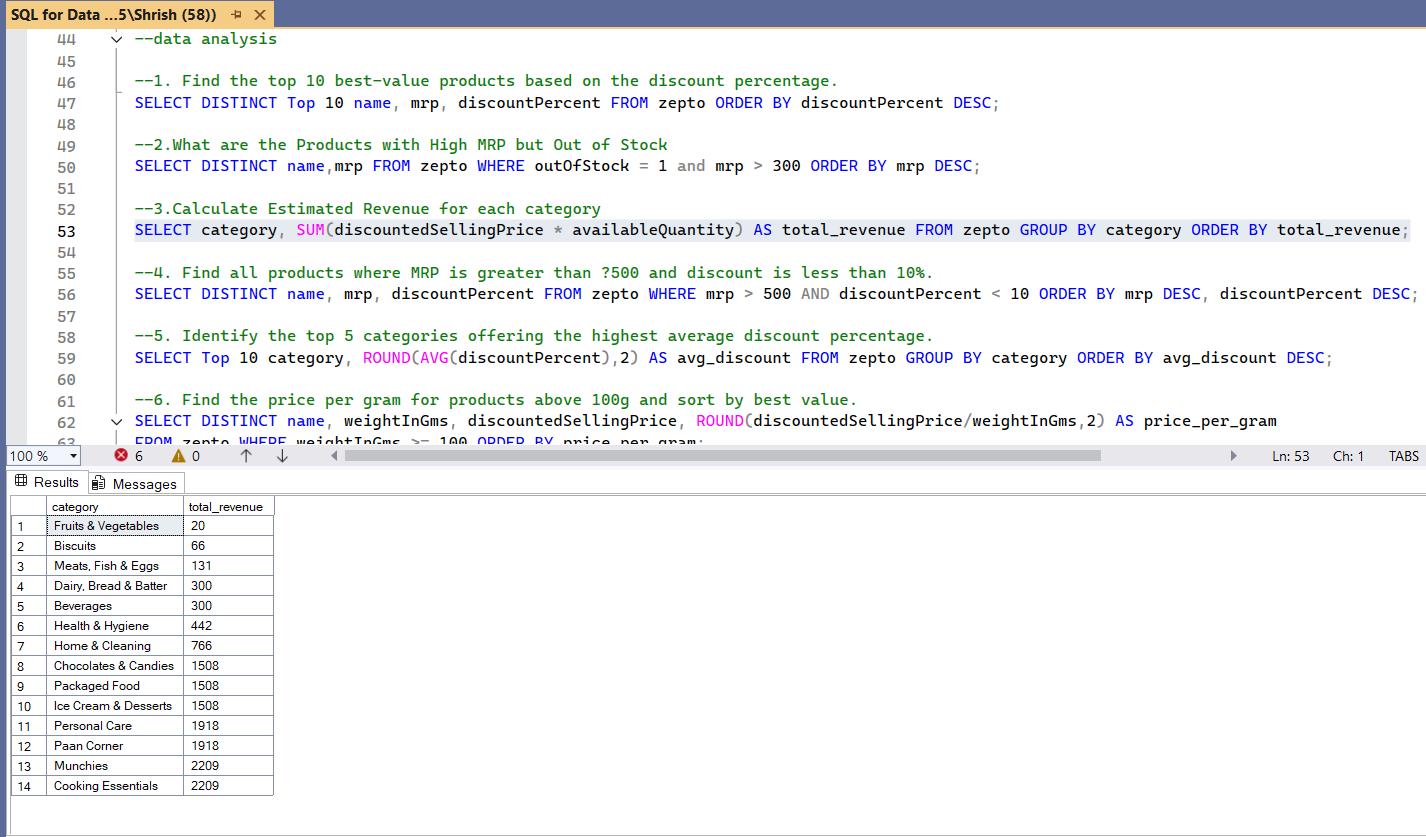
--products in stock vs out of stock

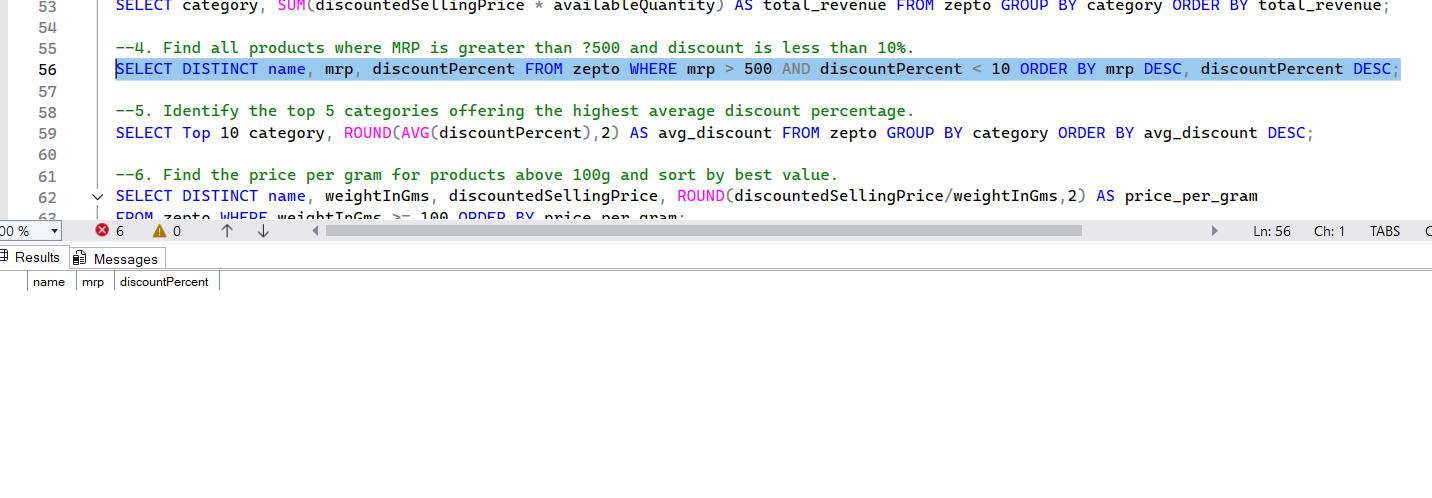
SELECT outOfStock, COUNT(ID) as Total FROM zepto GROUP BY outOfStock;

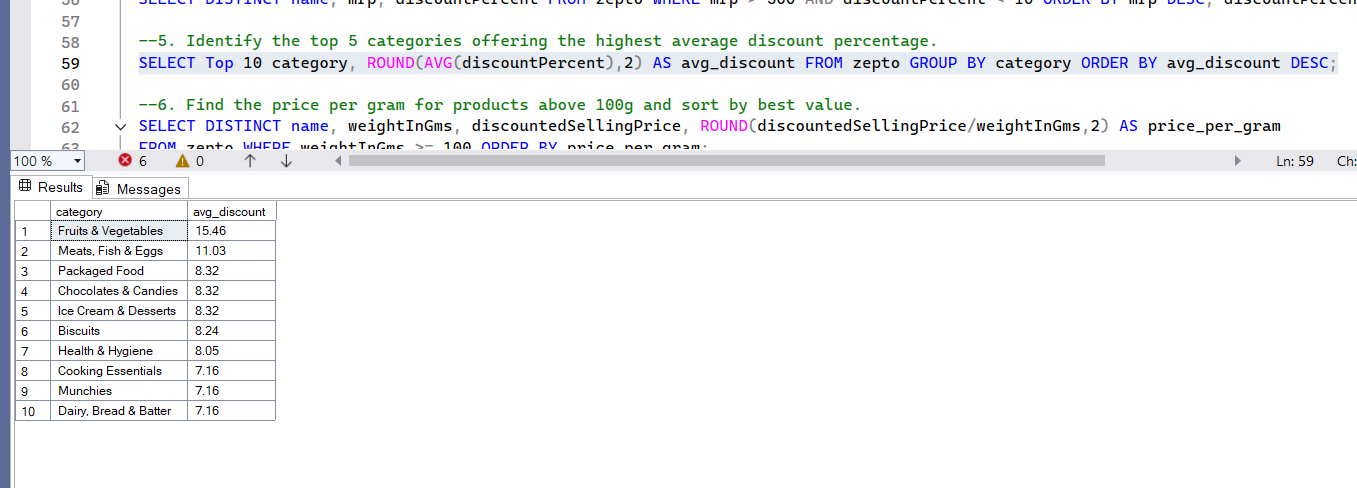
--product names present multiple times

SELECT name, COUNT(ID) AS "Number of Items" FROM zepto GROUP BY name HAVING count(ID) > 1 ORDER BY count(ID) DESC;  
  


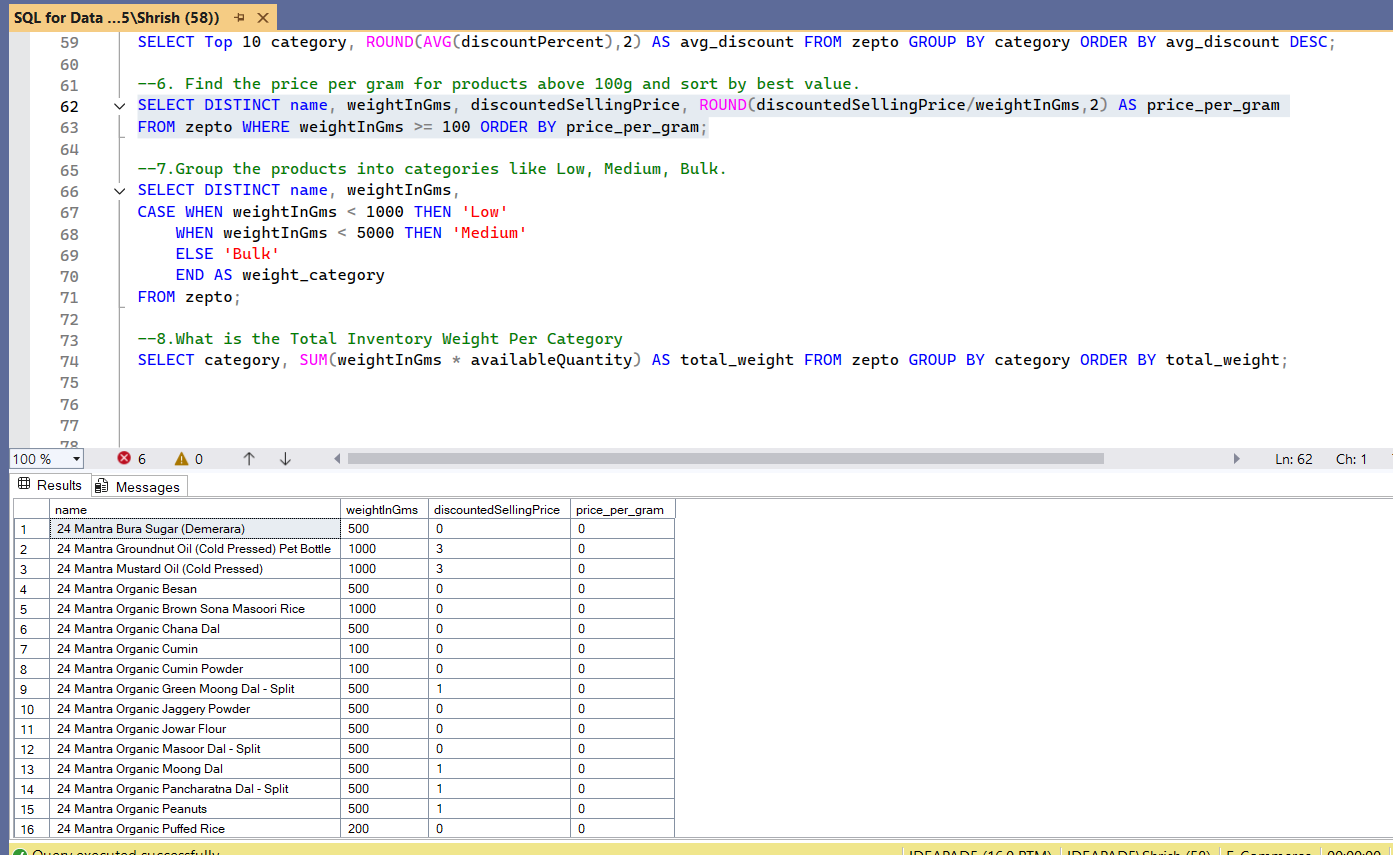
SELECT DISTINCT Top 10 name, mrp, discountPercent FROM zepto ORDER BY discountPercent DESC;  
  
  
  
  
  
SELECT DISTINCT name,mrp FROM zepto WHERE outOfStock = 1 and mrp > 300 ORDER BY mrp DESC;  


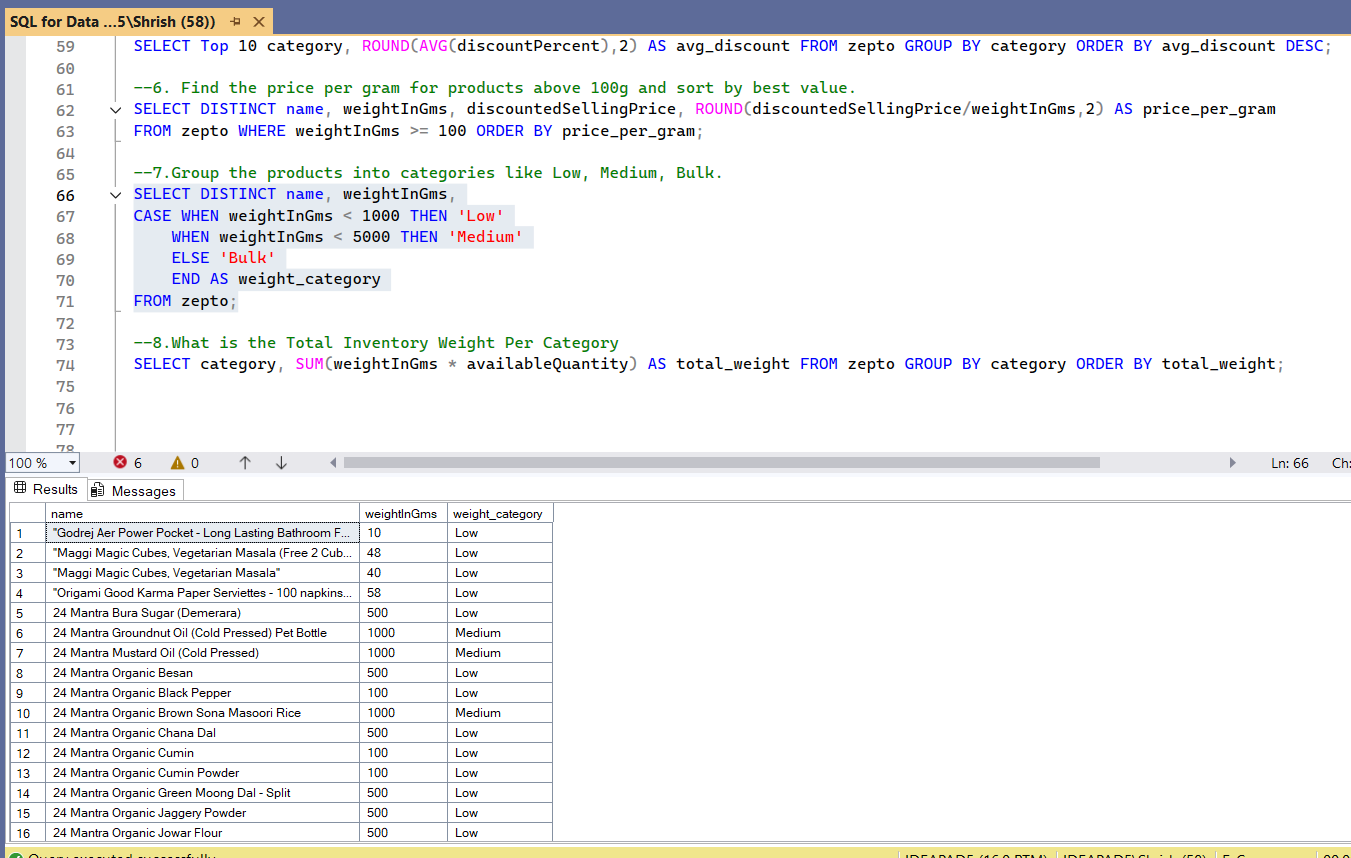
SELECT category, SUM(discountedSellingPrice \* availableQuantity) AS total\_revenue FROM zepto GROUP BY category ORDER BY total\_revenue;  


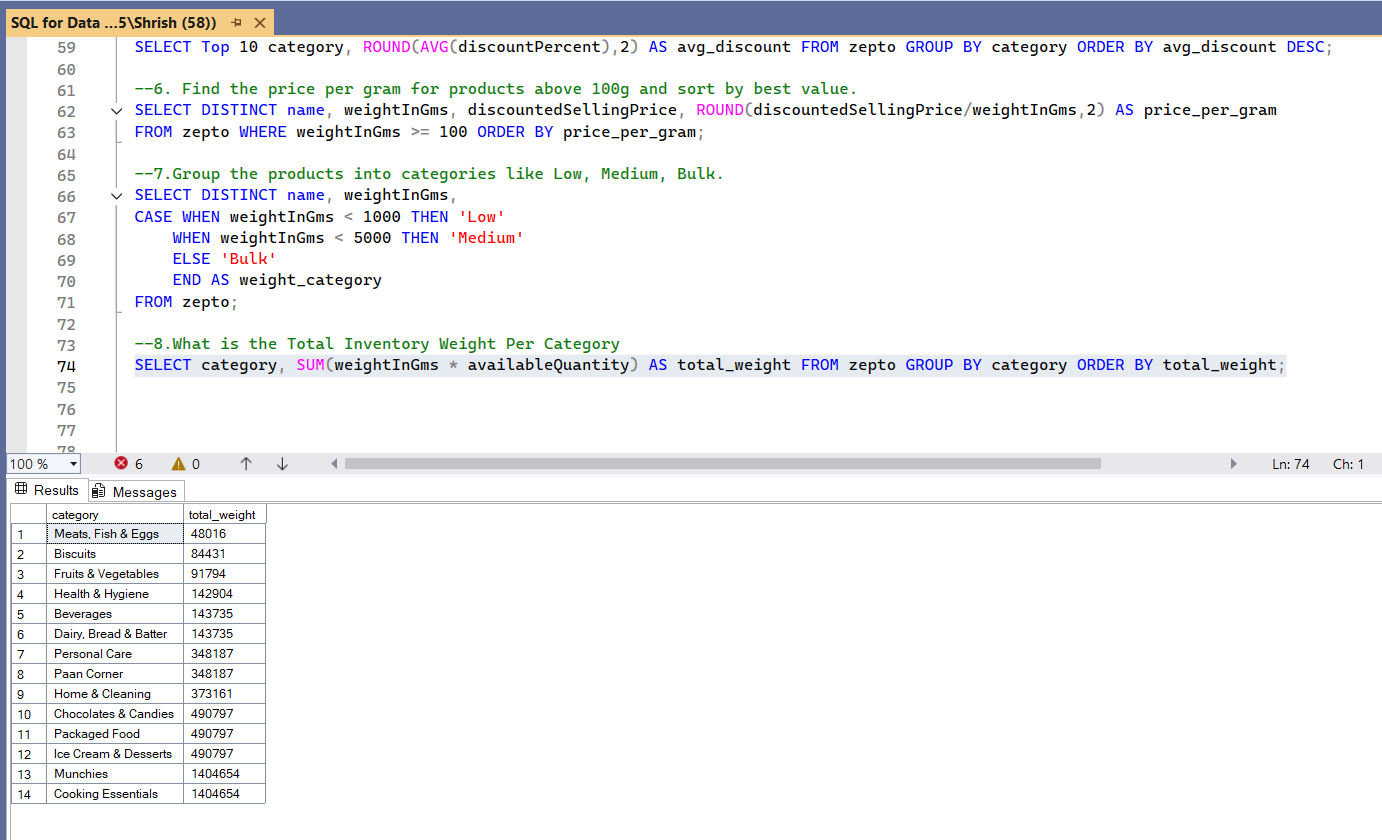
SELECT DISTINCT name, mrp, discountPercent FROM zepto WHERE mrp > 500 AND discountPercent < 10 ORDER BY mrp DESC, discountPercent DESC;  


SELECT Top 10 category, ROUND(AVG(discountPercent),2) AS avg\_discount FROM zepto GROUP BY category ORDER BY avg\_discount DESC;  


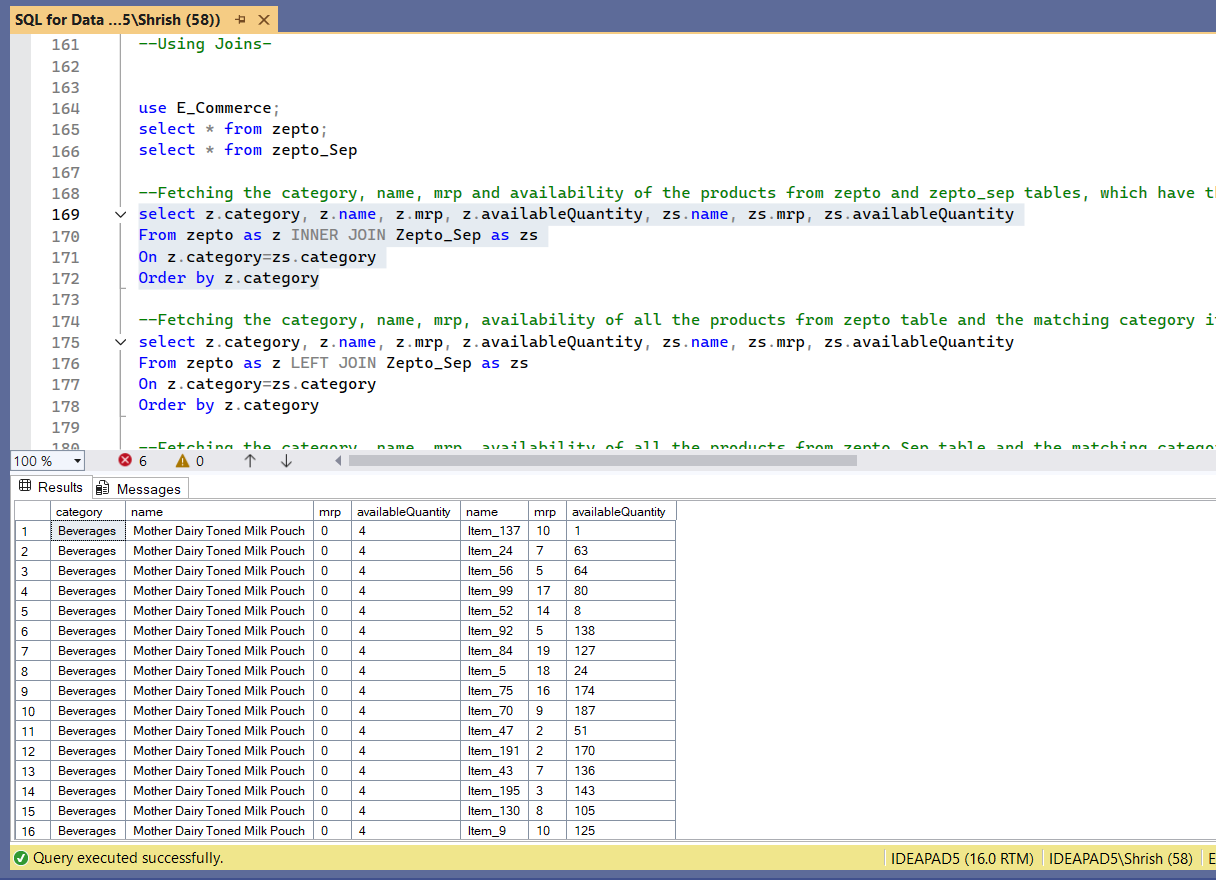
SELECT DISTINCT name, weightInGms, discountedSellingPrice, ROUND(discountedSellingPrice/weightInGms,2) AS price\_per\_gram

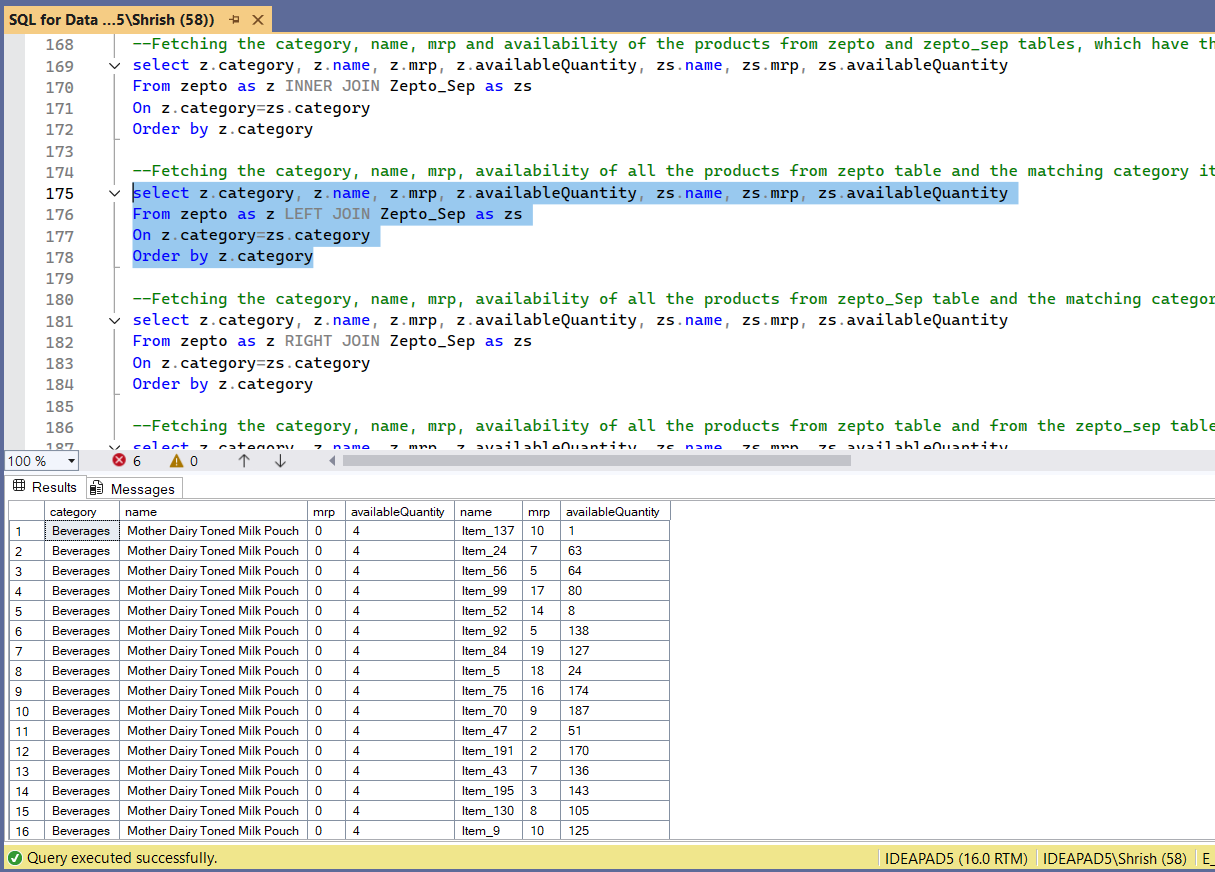
FROM zepto WHERE weightInGms >= 100 ORDER BY price\_per\_gram;  


SELECT DISTINCT name, weightInGms,  
CASE WHEN weightInGms < 1000 THEN 'Low'  
 WHEN weightInGms < 5000 THEN 'Medium'  
 ELSE 'Bulk'  
 END AS weight\_category  
FROM zepto;  


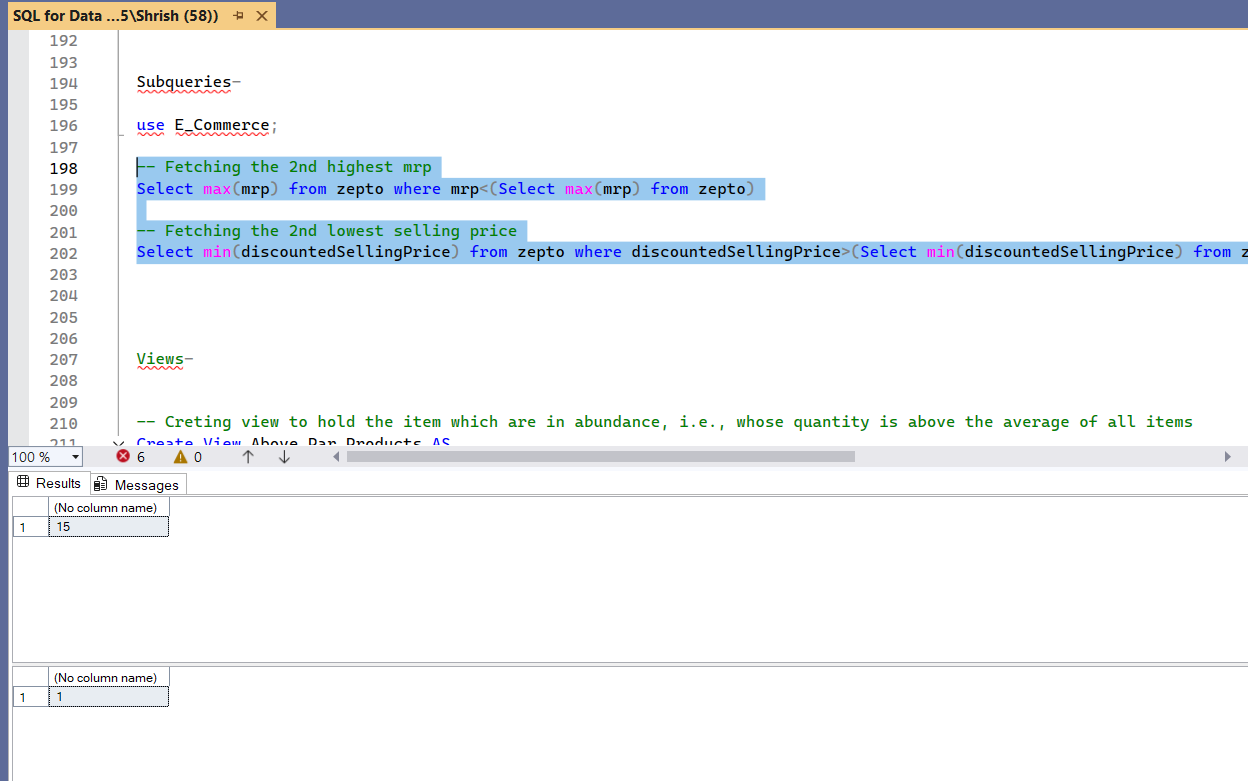
SELECT category, SUM(weightInGms \* availableQuantity) AS total\_weight FROM zepto GROUP BY category ORDER BY total\_weight;  


Joins-

select z.category, z.name, z.mrp, z.availableQuantity, zs.name, zs.mrp, zs.availableQuantity  
From zepto as z INNER JOIN Zepto\_Sep as zs  
On z.category=zs.category  
Order by z.category  
  


select z.category, z.name, z.mrp, z.availableQuantity, zs.name, zs.mrp, zs.availableQuantity  
From zepto as z LEFT JOIN Zepto\_Sep as zs  
On z.category=zs.category  
Order by z.category  


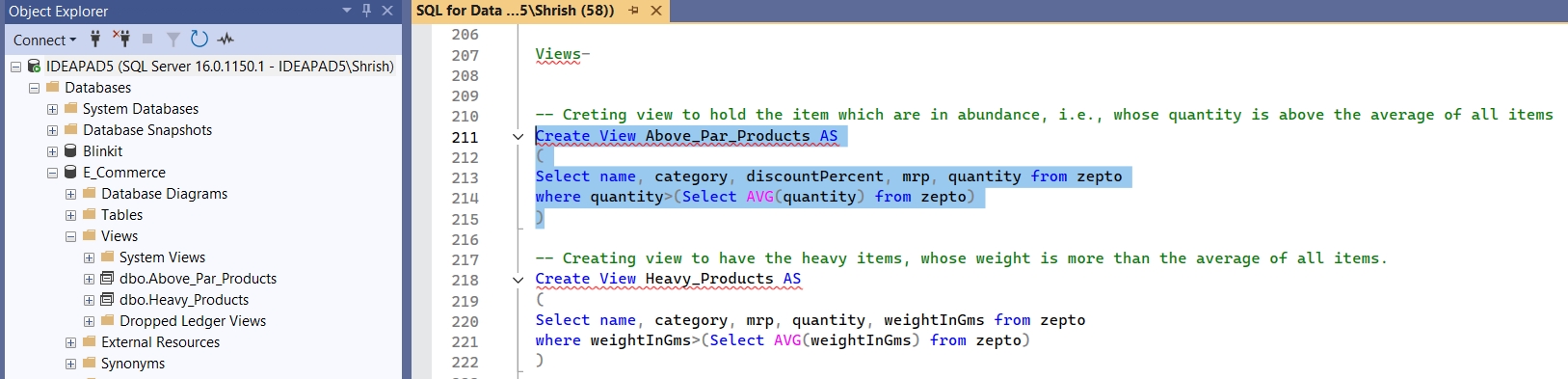
-- Fetching the 2nd highest mrp  
Select max(mrp) from zepto where mrp<(Select max(mrp) from zepto)

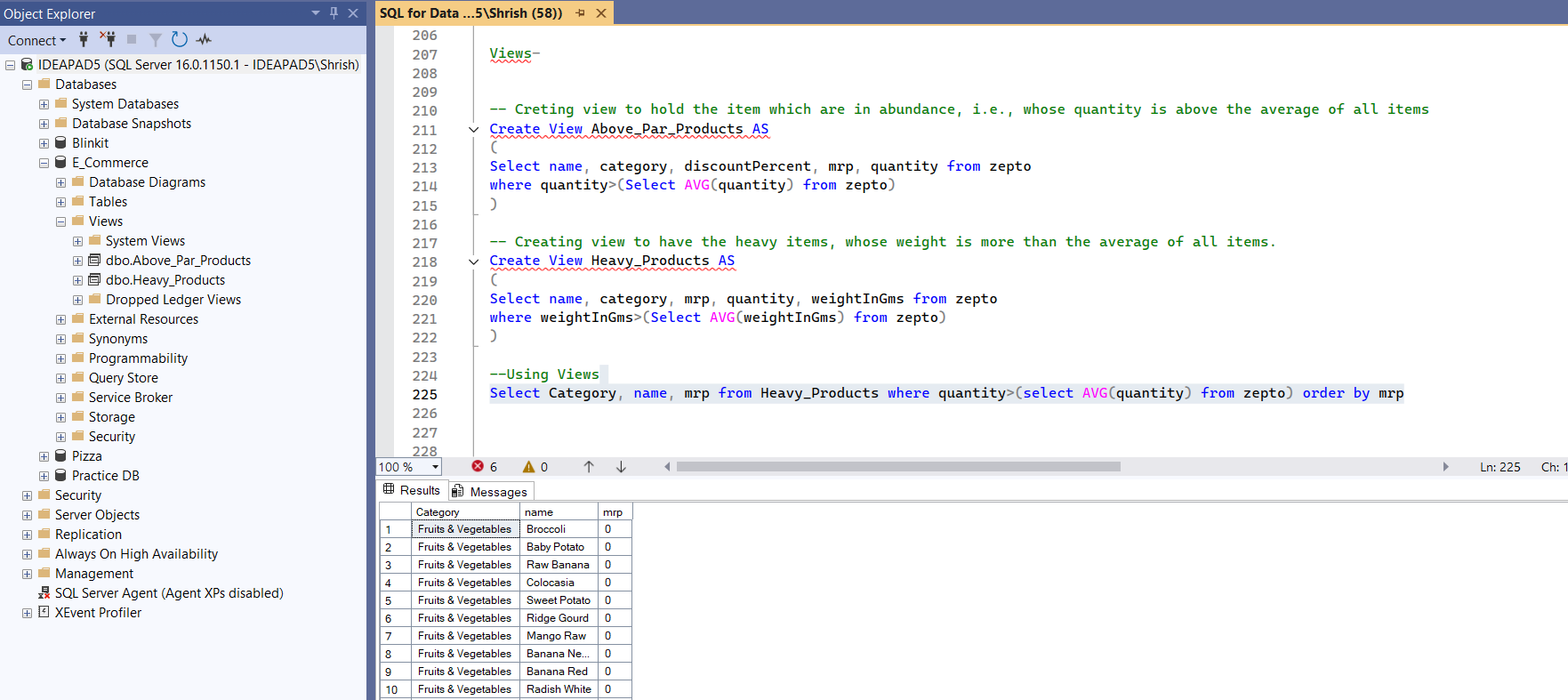
-- Fetching the 2nd lowest selling price  
Select min(discountedSellingPrice) from zepto where discountedSellingPrice>(Select min(discountedSellingPrice) from zepto)  
  


Creating Views-  
  
  
-- Creating view to hold the item which are in abundance, i.e., whose quantity is above the average of all items

Create View Above\_Par\_Products AS  
(  
Select name, category, discountPercent, mrp, quantity from zepto  
where quantity>(Select AVG(quantity) from zepto)  
)

-- Creating view to have the heavy items, whose weight is more than the average of all items.  
Create View Heavy\_Products AS  
(  
Select name, category, mrp, quantity, weightInGms from zepto  
where weightInGms>(Select AVG(weightInGms) from zepto)  
)



Select Category, name, mrp from Heavy\_Products where quantity>(select AVG(quantity) from zepto) order by mrp  


--Creating Indices for bot the tables on primary keys of both the tables-  
  
CREATE INDEX prime\_index ON zepto (ID);  
CREATE INDEX prime\_index ON zepto\_Sep (ID);

