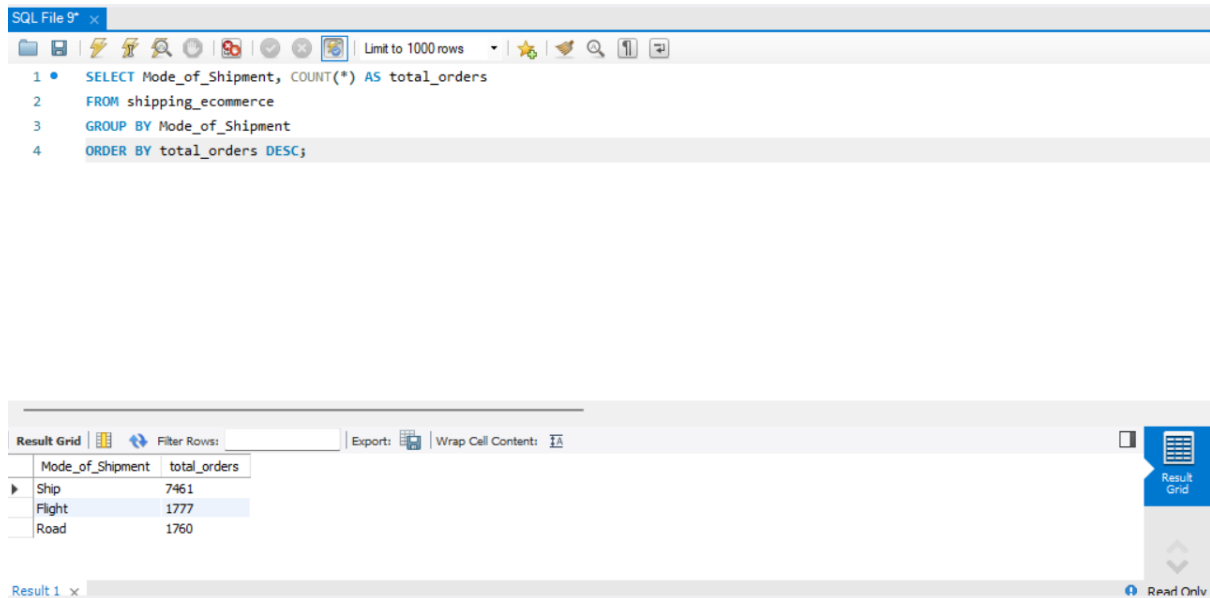


-- 1. Most common mode of shipment

```
SELECT Mode_of_Shipment, COUNT(*) AS total_orders  
  
FROM shipping_ecommerce  
  
GROUP BY Mode_of_Shipment  
  
ORDER BY total_orders DESC;
```



The screenshot shows a SQL query editor window titled "SQL File 9". The query is as follows:

```
1 SELECT Mode_of_Shipment, COUNT(*) AS total_orders  
2 FROM shipping_ecommerce  
3 GROUP BY Mode_of_Shipment  
4 ORDER BY total_orders DESC;
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has two columns: "Mode_of_Shipment" and "total_orders". The results are as follows:

Mode_of_Shipment	total_orders
Ship	7461
Flight	1777
Road	1760

The "Result Grid" tab is active, and the "Read Only" status is indicated at the bottom right.

-- 2. Basic analysis using SELECT, WHERE, ORDER BY, GROUP BY

```
SELECT Product_importance, Gender, COUNT(*) AS order_count  
  
FROM shipping_ecommerce  
  
WHERE Discount_offered > 10  
  
GROUP BY Product_importance, Gender  
  
ORDER BY order_count DESC;
```

SQL File 9*

```

1  -- 2.Basic analysis using SELECT, WHERE, ORDER BY, GROUP BY
2  • SELECT Product_importance, Gender, COUNT(*) AS order_count
3    FROM shipping_ecommerce
4    WHERE Discount_offered > 10
5    GROUP BY Product_importance, Gender
6    ORDER BY order_count DESC;

```

Result Grid

Product_importance	Gender	order_count
low	F	627
medium	F	599
low	M	597
medium	M	553
high	M	137
high	F	134

Result 2 × Read Only

-- 3. Total and average weight shipped per warehouse block

```

SELECT Warehouse_block,
       COUNT(*) AS total_orders,
       SUM(Weight_in_gms) AS total_weight,
       AVG(Weight_in_gms) AS avg_weight
FROM shipping_ecommerce
GROUP BY Warehouse_block
ORDER BY total_weight DESC;

```

SQL File 9*

```

1  -- 3. Total and average weight shipped per warehouse block
2  SELECT Warehouse_block,
3         COUNT(*) AS total_orders,
4         SUM(Weight_in_gms) AS total_weight,
5         AVG(Weight_in_gms) AS avg_weight
6  FROM shipping_ecommerce
7  GROUP BY Warehouse_block
8  ORDER BY total_weight DESC;

```

Result Grid

Warehouse_block	total_orders	total_weight	avg_weight
F	3666	13349327	3641.3876
C	1833	6674560	3641.3312
B	1833	6664240	3635.7010
D	1834	6655305	3628.8468
A	1832	6621909	3614.5791

Result 3 × Read Only

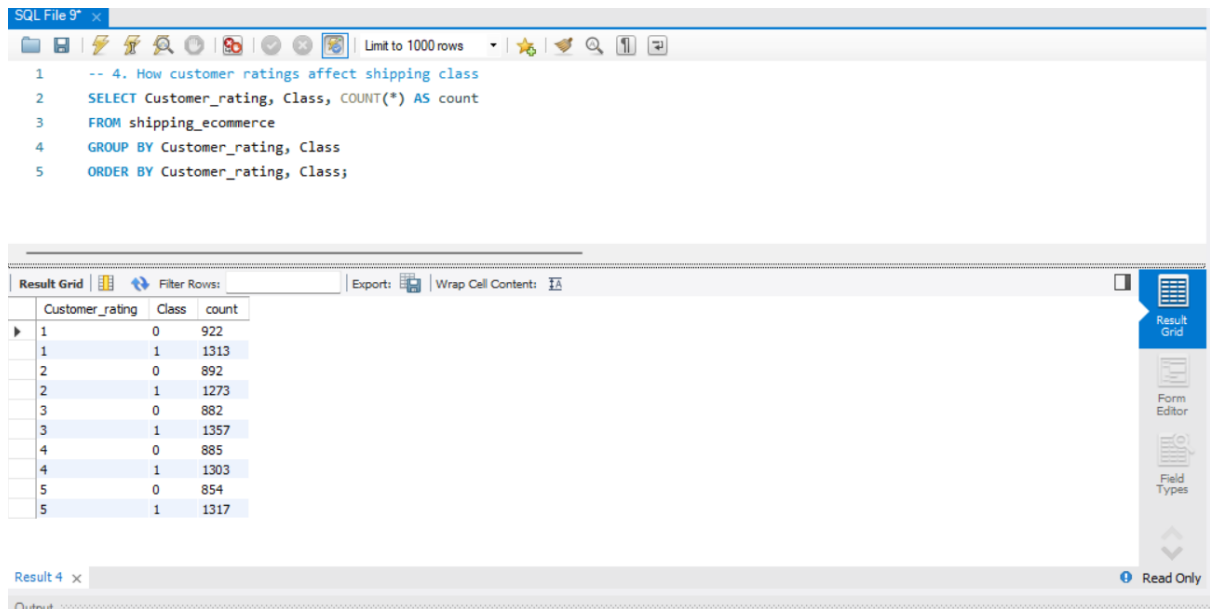
-- 4. How customer ratings affect shipping class

```
SELECT Customer_rating, Class, COUNT(*) AS count

FROM shipping_ecommerce

GROUP BY Customer_rating, Class

ORDER BY Customer_rating, Class;
```



The screenshot shows a SQL editor window titled "SQL File 9" with a toolbar and a "Limit to 1000 rows" dropdown. The SQL code is as follows:

```
1 -- 4. How customer ratings affect shipping class
2 SELECT Customer_rating, Class, COUNT(*) AS count
3 FROM shipping_ecommerce
4 GROUP BY Customer_rating, Class
5 ORDER BY Customer_rating, Class;
```

Below the editor is the "Result Grid" showing the output of the query. The grid has columns for "Customer_rating", "Class", and "count". The results are as follows:

Customer_rating	Class	count
1	0	922
1	1	1313
2	0	892
2	1	1273
3	0	882
3	1	1357
4	0	885
4	1	1303
5	0	854
5	1	1317

The interface includes a "Filter Rows" section, an "Export" button, and a "Wrap Cell Content" dropdown. On the right side, there are buttons for "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 4" and "Read Only".

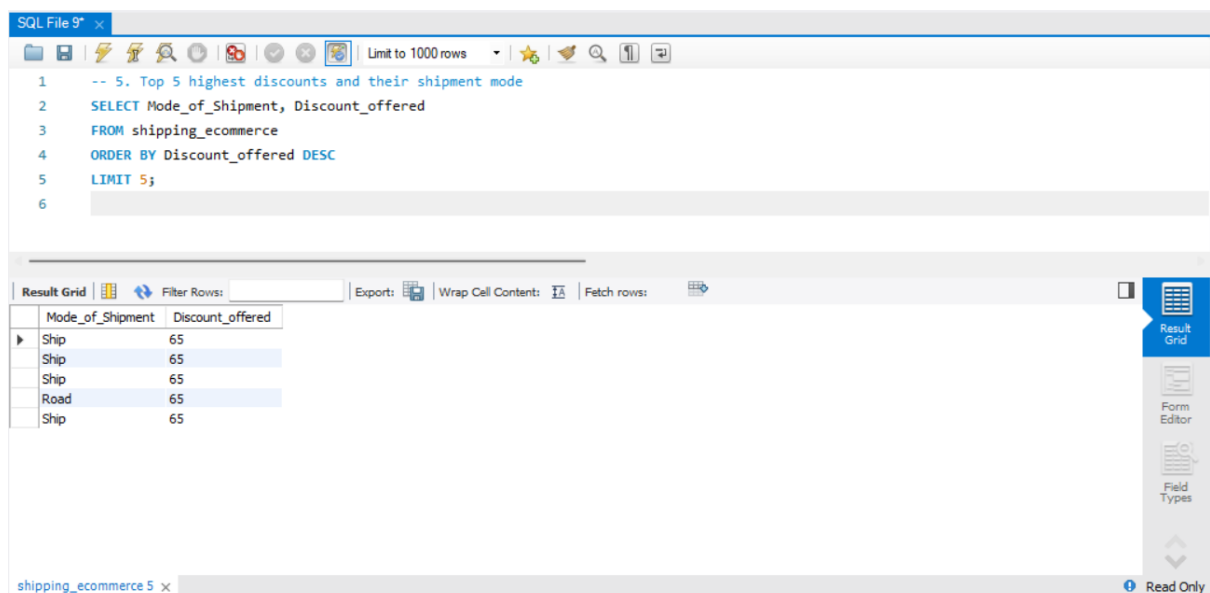
```
-- 5. Top 5 highest discounts and their shipment mode

SELECT Mode_of_Shipment, Discount_offered

FROM shipping_ecommerce

ORDER BY Discount_offered DESC

LIMIT 5;
```



The screenshot shows a SQL editor window titled "SQL File 9" with a toolbar and a "Limit to 1000 rows" dropdown. The SQL code is as follows:

```
1 -- 5. Top 5 highest discounts and their shipment mode
2 SELECT Mode_of_Shipment, Discount_offered
3 FROM shipping_ecommerce
4 ORDER BY Discount_offered DESC
5 LIMIT 5;
```

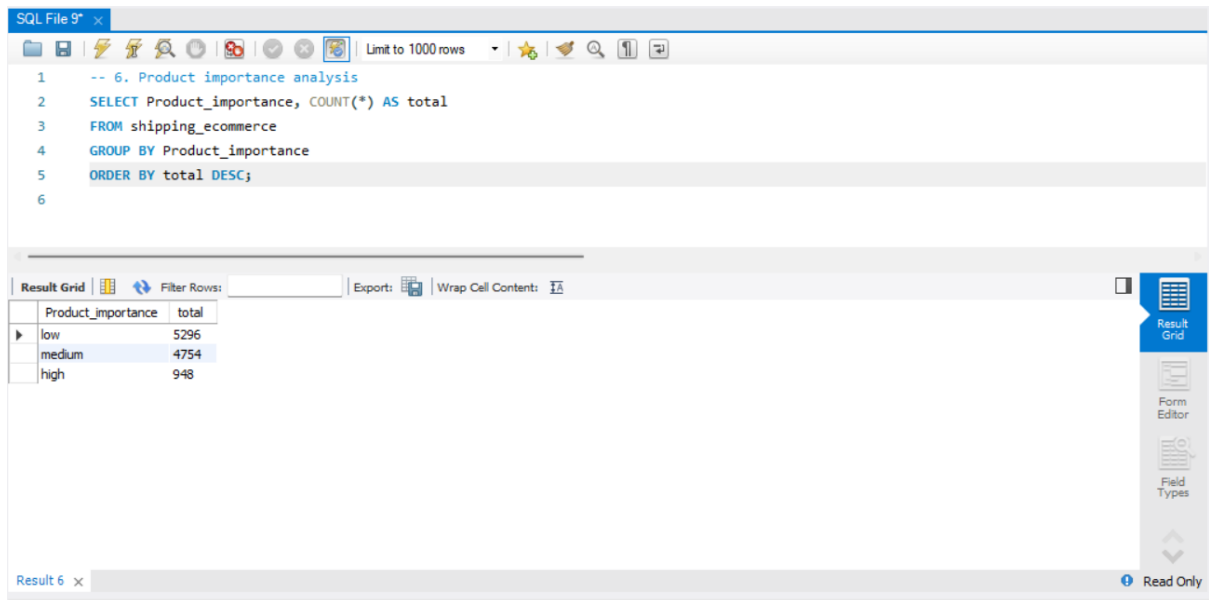
Below the editor is the "Result Grid" showing the output of the query. The grid has columns for "Mode_of_Shipment" and "Discount_offered". The results are as follows:

Mode_of_Shipment	Discount_offered
Ship	65
Ship	65
Ship	65
Road	65
Ship	65

The interface includes a "Filter Rows" section, an "Export" button, and a "Wrap Cell Content" dropdown. On the right side, there are buttons for "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "shipping_ecommerce 5" and "Read Only".

```
-- 6. Product importance analysis
```

```
SELECT Product_importance, COUNT(*) AS total  
  
FROM shipping_ecommerce  
  
GROUP BY Product_importance  
  
ORDER BY total DESC;
```



The screenshot shows a SQL IDE window titled "SQL File 9". The query editor contains the following SQL code:

```
1  -- 6. Product importance analysis  
2  SELECT Product_importance, COUNT(*) AS total  
3  FROM shipping_ecommerce  
4  GROUP BY Product_importance  
5  ORDER BY total DESC;  
6
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are as follows:

Product_importance	total
low	5296
medium	4754
high	948

The IDE interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Filter Rows" input field. The bottom status bar indicates "Result 6" and "Read Only".

```
-- 7. Gender-wise shipping behavior  
  
SELECT Gender,  
  
    COUNT(*) AS total_orders,  
  
    AVG(Weight_in_gms) AS avg_weight,  
  
    AVG(Discount_offered) AS avg_discount  
  
FROM shipping_ecommerce  
  
GROUP BY Gender;
```

SQL File 9*

```

2  SELECT Gender,
3      COUNT(*) AS total_orders,
4      AVG(Weight_in_gms) AS avg_weight,
5      AVG(Discount_offered) AS avg_discount
6  FROM shipping_ecommerce
7  GROUP BY Gender;
8

```

Result Grid

Gender	total_orders	avg_weight	avg_discount
M	5454	3639.9078	13.1808
F	5544	3627.9372	13.5635

Result 7 x Read Only

-- 8. Prior purchases vs shipping class

```

SELECT Prior_purchases, Class, COUNT(*) AS count
FROM shipping_ecommerce
GROUP BY Prior_purchases, Class
ORDER BY Prior_purchases;

```

SQL File 9*

```

1  -- 8. Prior purchases vs shipping class
2  SELECT Prior_purchases, Class, COUNT(*) AS count
3  FROM shipping_ecommerce
4  GROUP BY Prior_purchases, Class
5  ORDER BY Prior_purchases;

```

Result Grid

Prior_purchases	Class	count
2	0	974
2	1	1625
3	0	1421
3	1	2534
4	0	983
4	1	1171
5	0	645
5	1	642
6	0	247
6	1	314
7	0	44
7	1	92
8	0	45
8	1	83
10	0	76
10	1	102

Result 8 x Read Only

-- 9. Orders with high discount & low weight (anomaly check)

```

SELECT *
FROM shipping_ecommerce

```

WHERE Discount_offered > 20 AND Weight_in_gms < 3000;

SQL File 9" x

Limit to 1000 rows

```
1 -- 9. Orders with high discount & low weight (anomaly check)
2 SELECT *
3 FROM shipping_ecommerce
4 WHERE Discount_offered > 20 AND Weight_in_gms < 3000;
5
```

Result Grid

Filter Rows: Export: Wrap Cell Contents: Fetch rows:

	Customer_care_calls	Customer_rating	Prior_purchases	Discount_offered	Weight_in_gms	Warehouse_block	Mode_of_Shipment	Product_importance	Gender	Class
5	3	5	21	1011	D	Road	low	M	1	
4	5	4	21	1014	B	Ship	low	M	1	
2	3	8	21	1033	A	Ship	medium	M	1	
3	1	3	21	1047	A	Road	high	F	1	
4	1	2	21	1075	C	Flight	high	F	1	
2	5	3	21	1113	F	Ship	low	F	1	
5	4	6	21	1145	B	Flight	medium	F	1	
5	3	3	21	1187	F	Ship	medium	M	1	
4	4	2	21	1495	A	Ship	medium	M	1	
4	3	3	21	1588	F	Ship	low	F	1	
4	2	2	21	1617	B	Ship	medium	M	1	
3	2	4	21	1636	D	Ship	medium	F	1	
4	2	3	21	1637	F	Ship	low	M	1	
5	5	3	21	1648	A	Ship	medium	F	1	
5	2	4	21	1674	B	Ship	low	M	1	
4	5	3	21	1683	B	Ship	low	F	1	
4	4	5	21	1727	F	Ship	low	M	1	
4	3	3	21	1955	F	Flight	low	F	1	
3	4	3	21	2003	A	Ship	low	F	1	
4	5	4	21	2026	B	Road	medium	M	1	
5	3	2	21	2053	F	Ship	medium	F	1	
2	5	2	21	2070	F	Ship	low	F	1	
3	2	3	21	2351	A	Flight	medium	M	1	
5	1	3	21	2351	A	Flight	high	M	1	

shipping_ecommerce 9 x

Read On

-- 10. Which block has highest failed deliveries (if Class=0 means failure)

SELECT Warehouse_block, COUNT(*) AS failed_deliveries

FROM shipping_ecommerce

WHERE Class = 0

GROUP BY Warehouse_block

ORDER BY failed_deliveries DESC;

SQL File 9*

```
1 -- 10. Which block has highest failed deliveries (if Class=0 means failure)
2 SELECT Warehouse_block, COUNT(*) AS failed_deliveries
3 FROM shipping_ecommerce
4 WHERE Class = 0
5 GROUP BY Warehouse_block
6 ORDER BY failed_deliveries DESC;
```

Result Grid

	Warehouse_block	failed_deliveries
▶	F	1472
	A	757
	C	739
	D	738
	B	729

Result 10 × Read Only

-- 11. Correlation-like query (Discount vs Class)

```
SELECT Class,
        AVG(Discount_offered) AS avg_discount,
        COUNT(*) AS orders
FROM shipping_ecommerce
GROUP BY Class;
```

SQL File 9* x

Limit to 1000 rows

```

1  -- 11. Correlation-like query (Discount vs Class)
2  SELECT Class,
3         AVG(Discount_offered) AS avg_discount,
4         COUNT(*) AS orders
5  FROM shipping_ecommerce
6  GROUP BY Class;

```

Result Grid

	Class	avg_discount	orders
▶	1	18.6637	6563
	0	5.5454	4435

Result 11 x

Read Only

-- 13. Aggregate functions

```

SELECT Gender,
       SUM(Weight_in_gms) AS total_weight,
       AVG(Discount_offered) AS avg_discount
FROM shipping_ecommerce
GROUP BY Gender;

```

SQL File 9* x

Limit to 1000 rows

```

1  -- 13. Aggregate functions
2  SELECT Gender,
3         SUM(Weight_in_gms) AS total_weight,
4         AVG(Discount_offered) AS avg_discount
5  FROM shipping_ecommerce
6  GROUP BY Gender;

```

Result Grid

	Gender	total_weight	avg_discount
▶	M	19852057	13.1808
	F	20113284	13.5635

