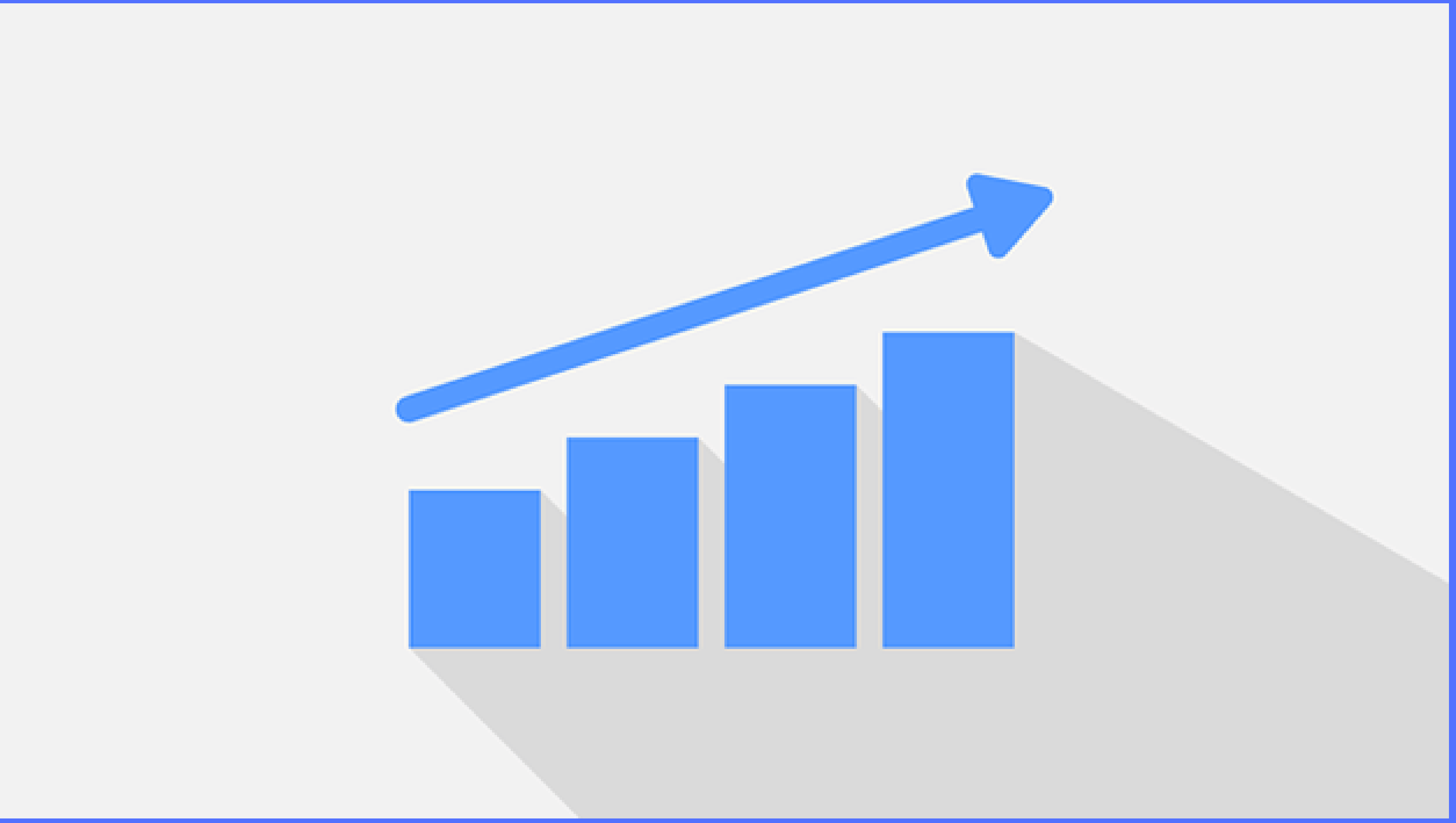


SALES DATA ANALYSIS USING SQL SERVER

Presented by khansa gafer

Batch Name: MIP-DA-05

How to analysis the data with sql server

1 STEP

create a new database in sql server management studio “SSMS”.

2 STEP

import the excel sheet and select the tables

3 STEP

querying this database using SQL to answer various questions

4 STEP

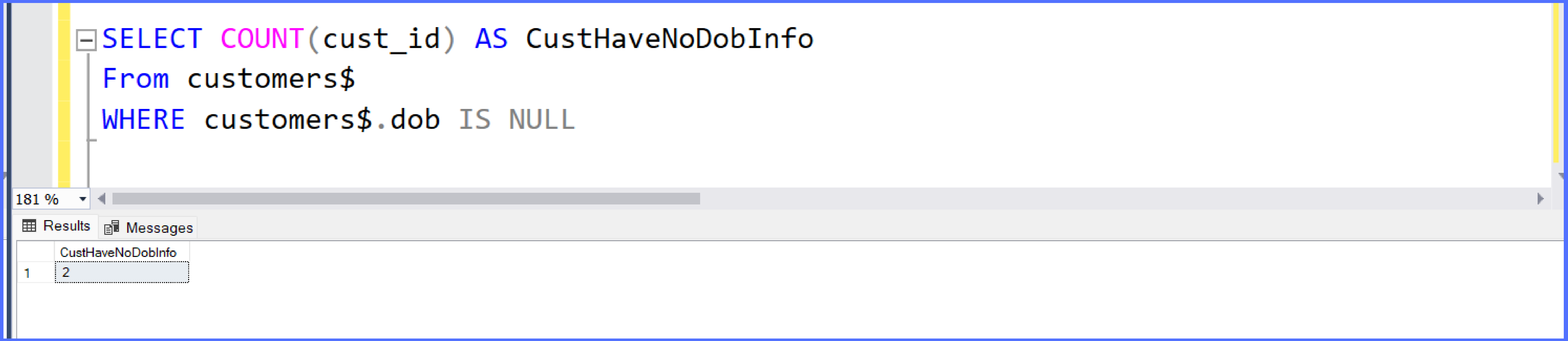
Perform queries to find out different insights which will help to make informed decisions

5 STEP

using power bi for visualization

Q-1

HOW MANY CUSTOMERS DO NOT HAVE DOB INFORMATION AVAILABLE?

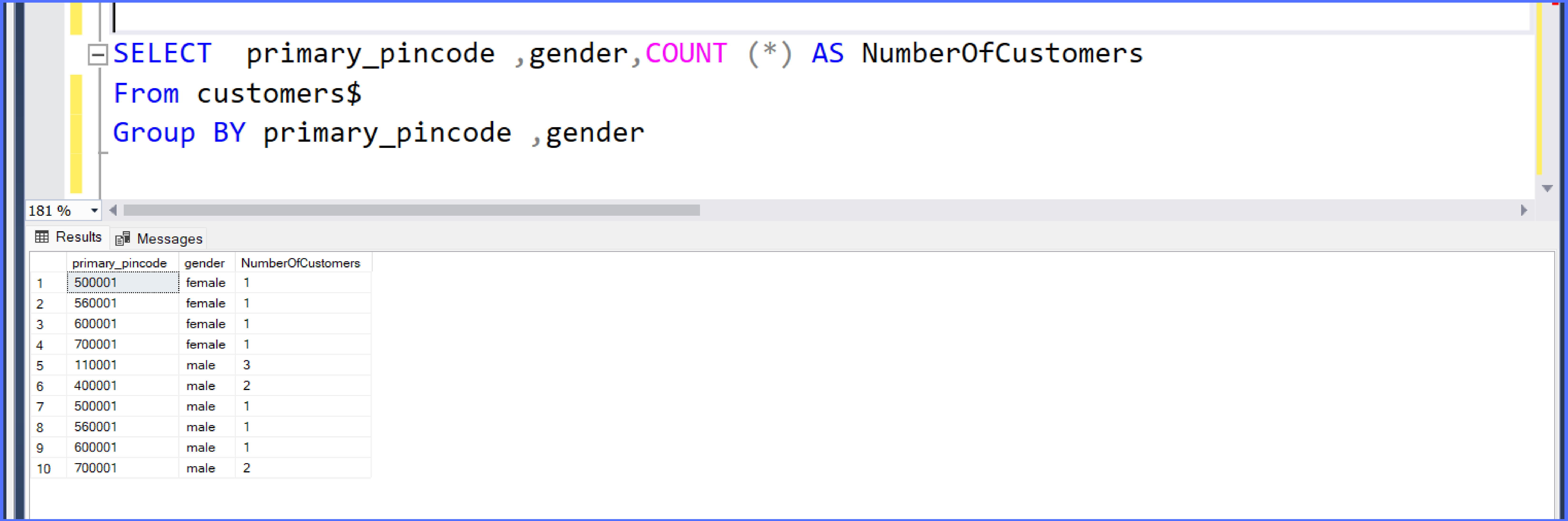


```
SELECT COUNT(cust_id) AS CustHaveNoDobInfo
From customers$
WHERE customers$.dob IS NULL
```

```
   CustHaveNoDobInfo
1  2
```

Q-2

HOW MANY CUSTOMERS ARE THERE IN EACH PINCODE AND GENDER COMBINATION?

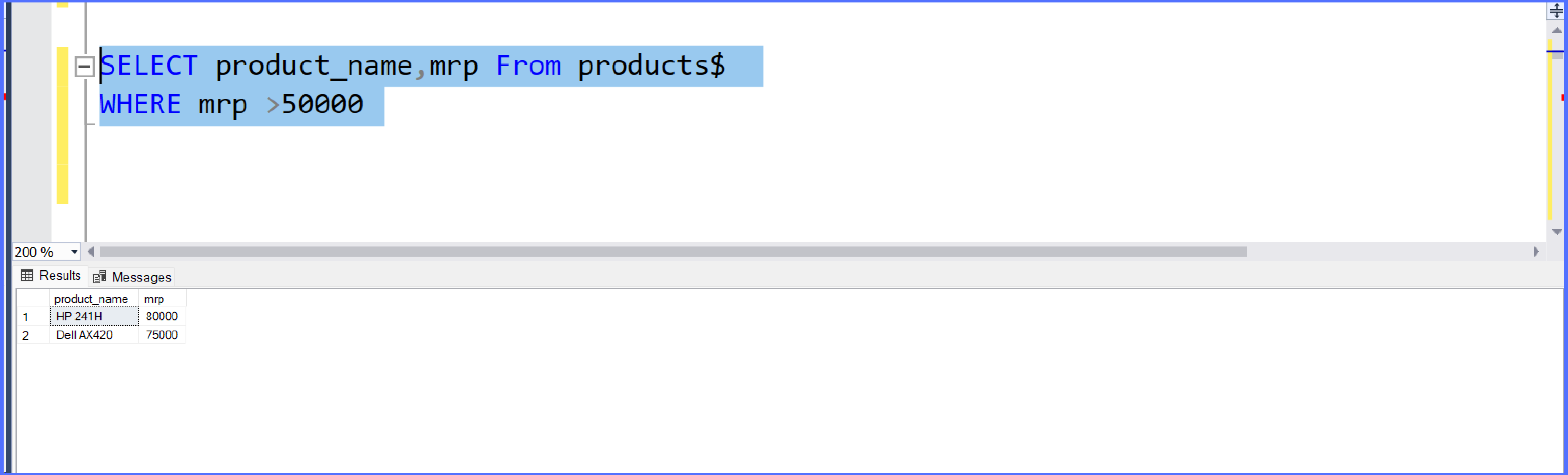


```
SELECT  primary_pincode ,gender,COUNT (*) AS NumberOfCustomers
From customers$
Group BY primary_pincode ,gender
```

	primary_pincode	gender	NumberOfCustomers
1	500001	female	1
2	560001	female	1
3	600001	female	1
4	700001	female	1
5	110001	male	3
6	400001	male	2
7	500001	male	1
8	560001	male	1
9	600001	male	1
10	700001	male	2

Q-3

PRINT PRODUCT NAME AND MRP FOR PRODUCTS WHICH HAVE MORE THAN 50000 MRP?

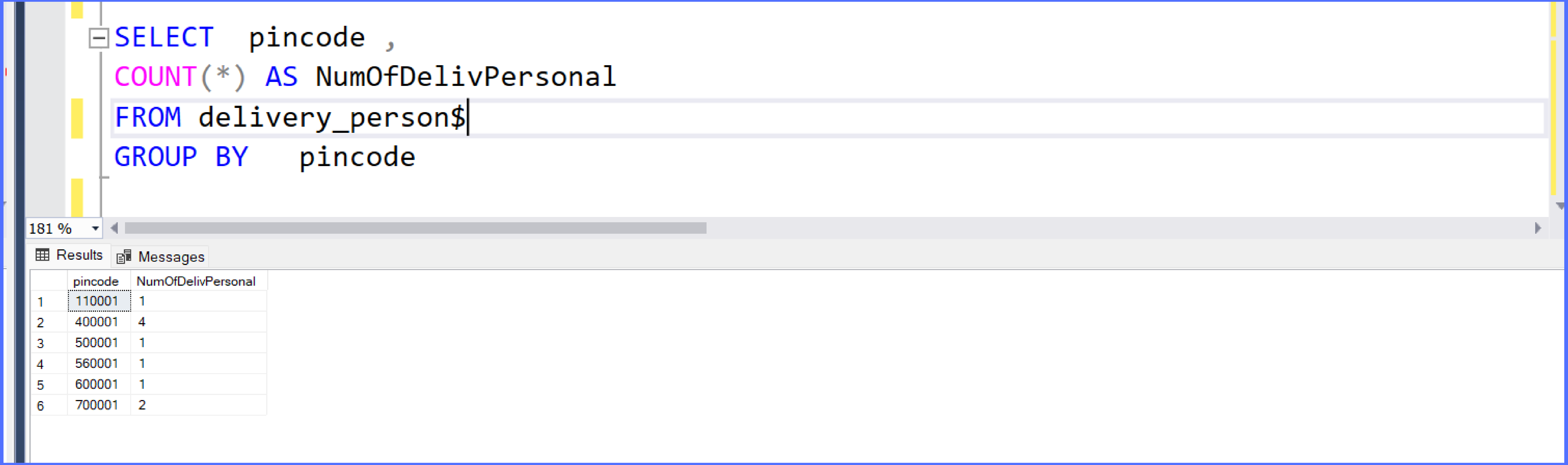


```
SELECT product_name,mrp From products$
WHERE mrp >50000
```

	product_name	mrp
1	HP 241H	80000
2	Dell AX420	75000

Q-4

HOW MANY DELIVERY PERSONAL ARE THERE IN EACH PINCODE?

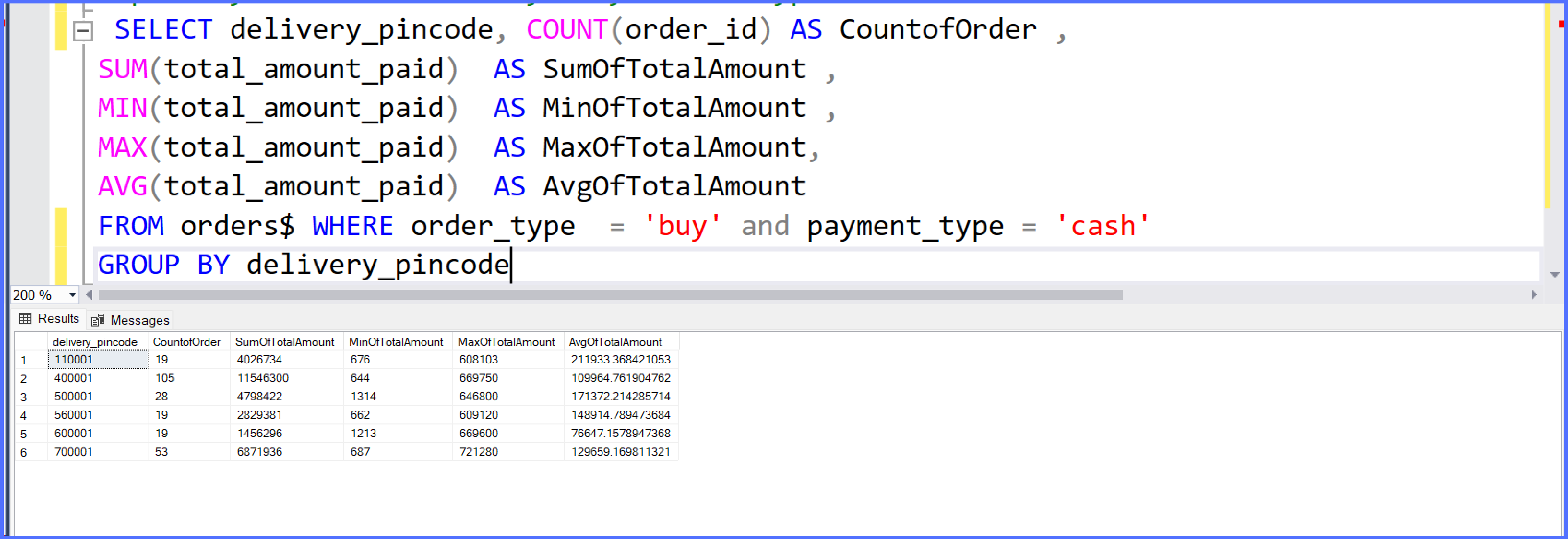


```
SELECT  pincode ,
COUNT(*) AS NumOfDelivPersonal
FROM delivery_person$
GROUP BY   pincode
```

	pincode	NumOfDelivPersonal
1	110001	1
2	400001	4
3	500001	1
4	560001	1
5	600001	1
6	700001	2

Q-5

FOR EACH PIN CODE, PRINT THE COUNT OF ORDERS, SUM OF TOTAL AMOUNT PAID, AVERAGE AMOUNT PAID, MAXIMUM AMOUNT PAID, MINIMUM AMOUNT PAID FOR THE TRANSACTIONS WHICH WERE PAID BY 'CASH'. TAKE ONLY 'BUY' ORDER TYPES

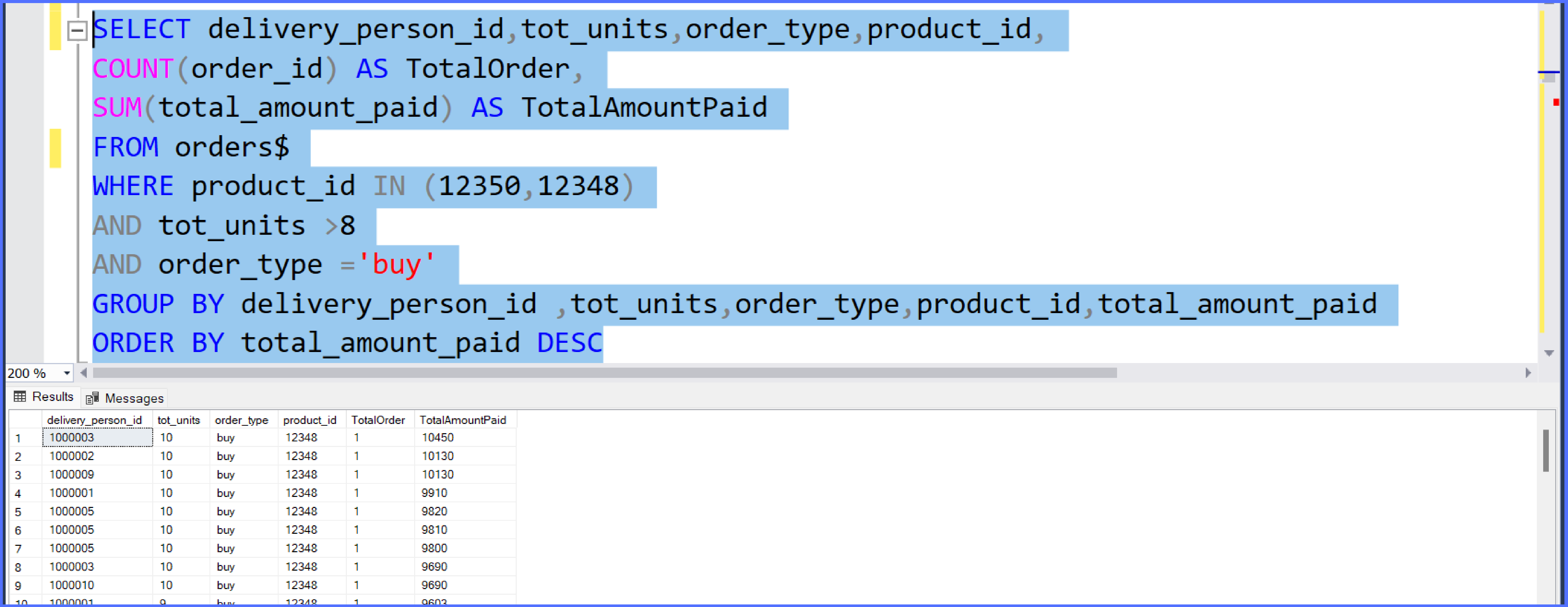


```sql
 SELECT delivery_pincode, COUNT(order_id) AS CountofOrder ,
SUM(total_amount_paid)  AS SumOfTotalAmount ,
MIN(total_amount_paid)  AS MinOfTotalAmount ,
MAX(total_amount_paid)  AS MaxOfTotalAmount,
AVG(total_amount_paid)  AS AvgOfTotalAmount
FROM orders$ WHERE order_type  = 'buy' and payment_type = 'cash'
GROUP BY delivery_pincode
```

	delivery_pincode	CountofOrder	SumOfTotalAmount	MinOfTotalAmount	MaxOfTotalAmount	AvgOfTotalAmount
1	110001	19	4026734	676	608103	211933.368421053
2	400001	105	11546300	644	669750	109964.761904762
3	500001	28	4798422	1314	646800	171372.214285714
4	560001	19	2829381	662	609120	148914.789473684
5	600001	19	1456296	1213	669600	76647.1578947368
6	700001	53	6871936	687	721280	129659.169811321

Q-6

FOR EACH DELIVERY_PERSON_ID, PRINT THE COUNT OF ORDERS AND TOTAL AMOUNT PAID FOR PRODUCT_ID = 12350 OR 12348 AND TOTAL UNITS > 8. SORT THE OUTPUT BY TOTAL AMOUNT PAID IN DESCENDING ORDER. TAKE ONLY 'BUY' ORDER TYPES

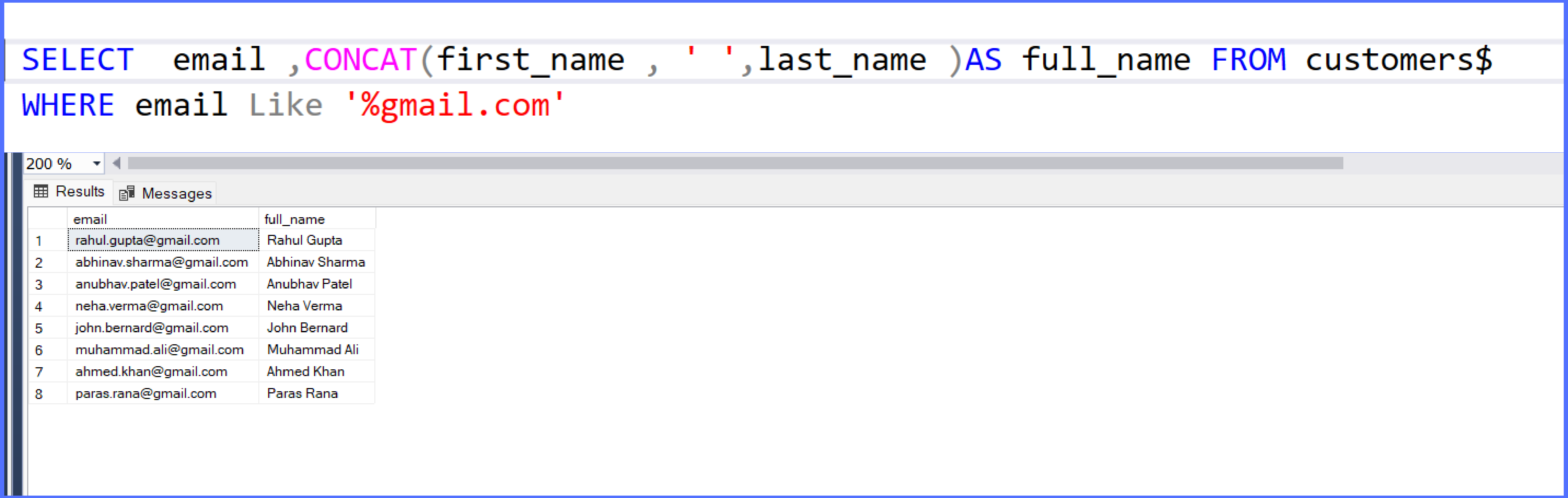


```sql
SELECT delivery_person_id,tot_units,order_type,product_id,
COUNT(order_id) AS TotalOrder,
SUM(total_amount_paid) AS TotalAmountPaid
FROM orders$
WHERE product_id IN (12350,12348)
AND tot_units >8
AND order_type ='buy'
GROUP BY delivery_person_id ,tot_units,order_type,product_id,total_amount_paid
ORDER BY total_amount_paid DESC
```

	delivery_person_id	tot_units	order_type	product_id	TotalOrder	TotalAmountPaid
1	1000003	10	buy	12348	1	10450
2	1000002	10	buy	12348	1	10130
3	1000009	10	buy	12348	1	10130
4	1000001	10	buy	12348	1	9910
5	1000005	10	buy	12348	1	9820
6	1000005	10	buy	12348	1	9810
7	1000005	10	buy	12348	1	9800
8	1000003	10	buy	12348	1	9690
9	1000010	10	buy	12348	1	9690
10	1000001	9	buy	12348	1	9603

Q-7

PRINT THE FULL NAMES (FIRST NAME PLUS LAST NAME) FOR CUSTOMERS THAT HAVE EMAIL ON "GMAIL.COM"?

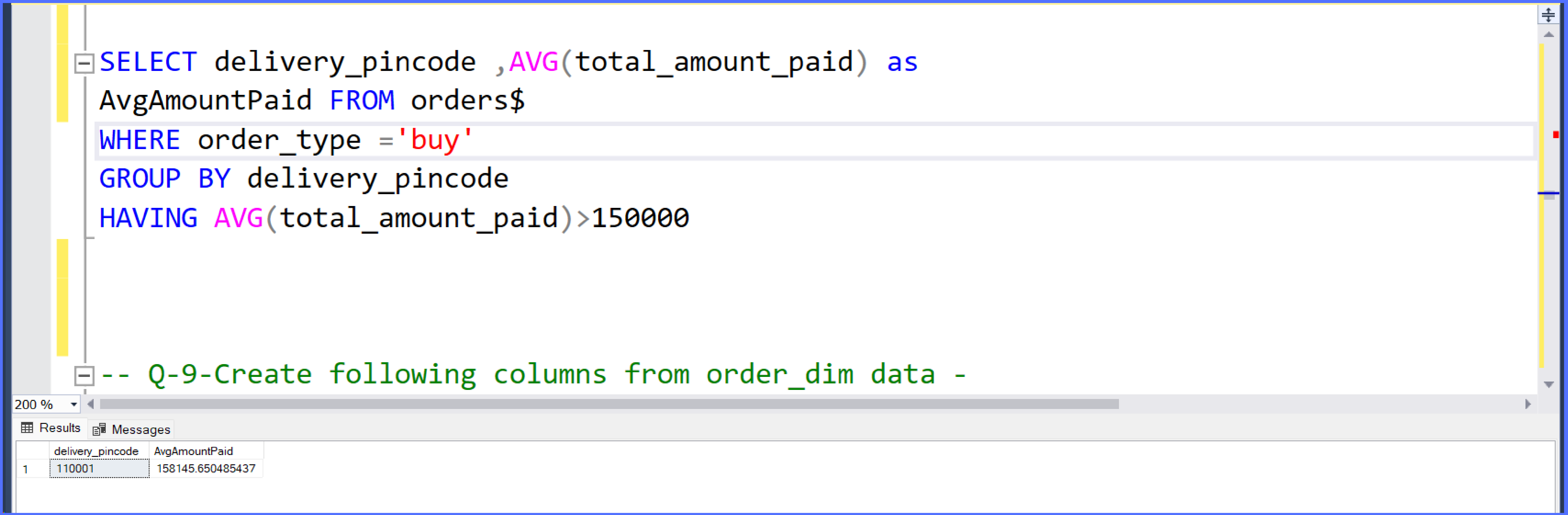


```
SELECT  email ,CONCAT(first_name , ' ',last_name )AS full_name FROM customers$
WHERE email Like '%gmail.com'
```

	email	full_name
1	rahul.gupta@gmail.com	Rahul Gupta
2	abhinav.sharma@gmail.com	Abhinav Sharma
3	anubhav.patel@gmail.com	Anubhav Patel
4	neha.verma@gmail.com	Neha Verma
5	john.bernard@gmail.com	John Bernard
6	muhammad.ali@gmail.com	Muhammad Ali
7	ahmed.khan@gmail.com	Ahmed Khan
8	paras.rana@gmail.com	Paras Rana

Q-8

WHICH PINCODE HAS AVERAGE AMOUNT PAID MORE THAN 150,000? TAKE ONLY 'BUY' ORDER TYPES

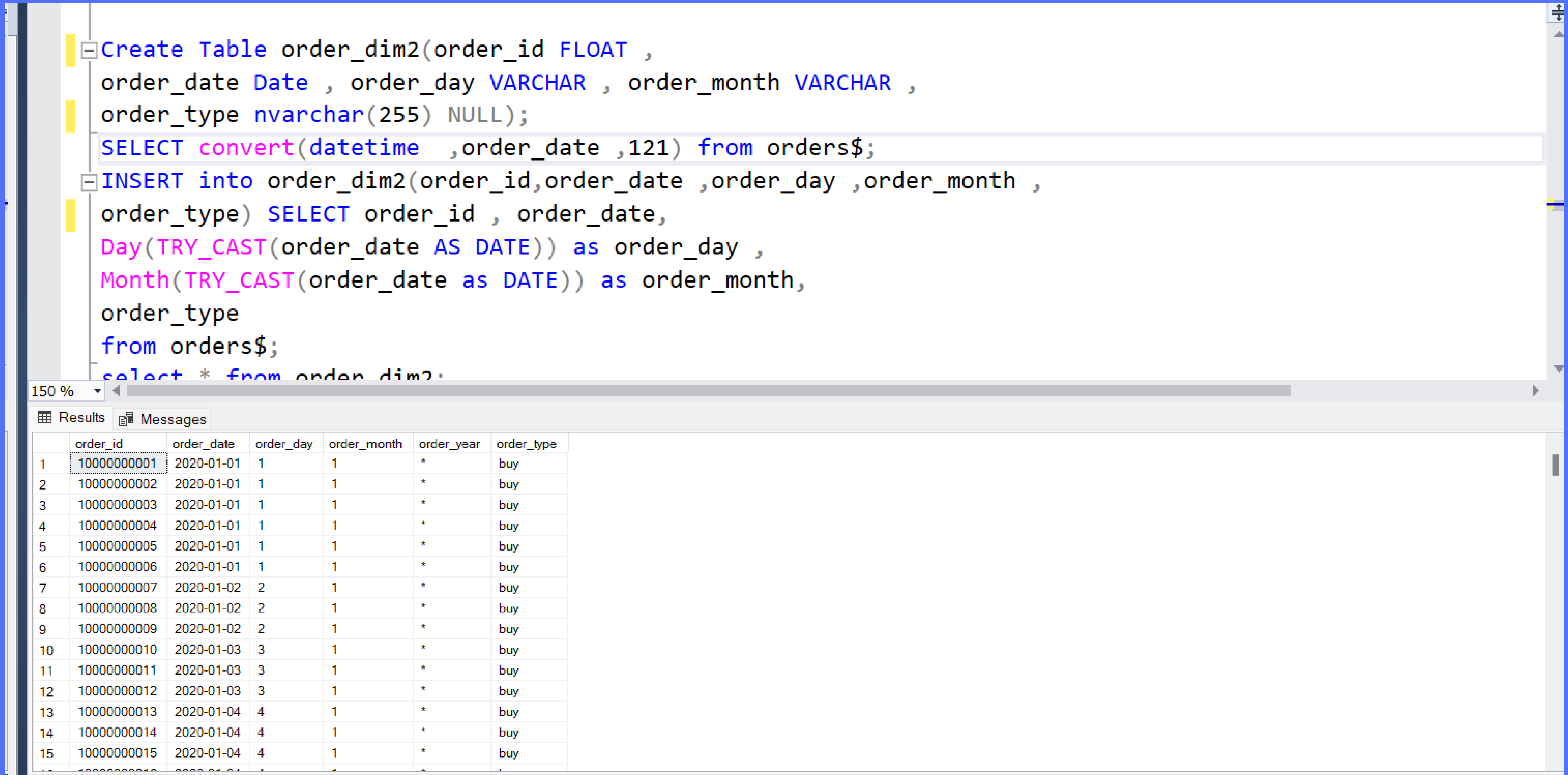


```
SELECT delivery_pincode ,AVG(total_amount_paid) as
AvgAmountPaid FROM orders$
WHERE order_type ='buy'
GROUP BY delivery_pincode
HAVING AVG(total_amount_paid)>150000



-- Q-9-Create following columns from order_dim data -
```

	delivery_pincode	AvgAmountPaid
1	110001	158145.650485437

Q-9

CREATE FOLLOWING COLUMNS FROM ORDER_DIM DATA - 1ORDER_DATE 2- ORDER DAY 3- ORDER MONTH 4- ORDER YEAR

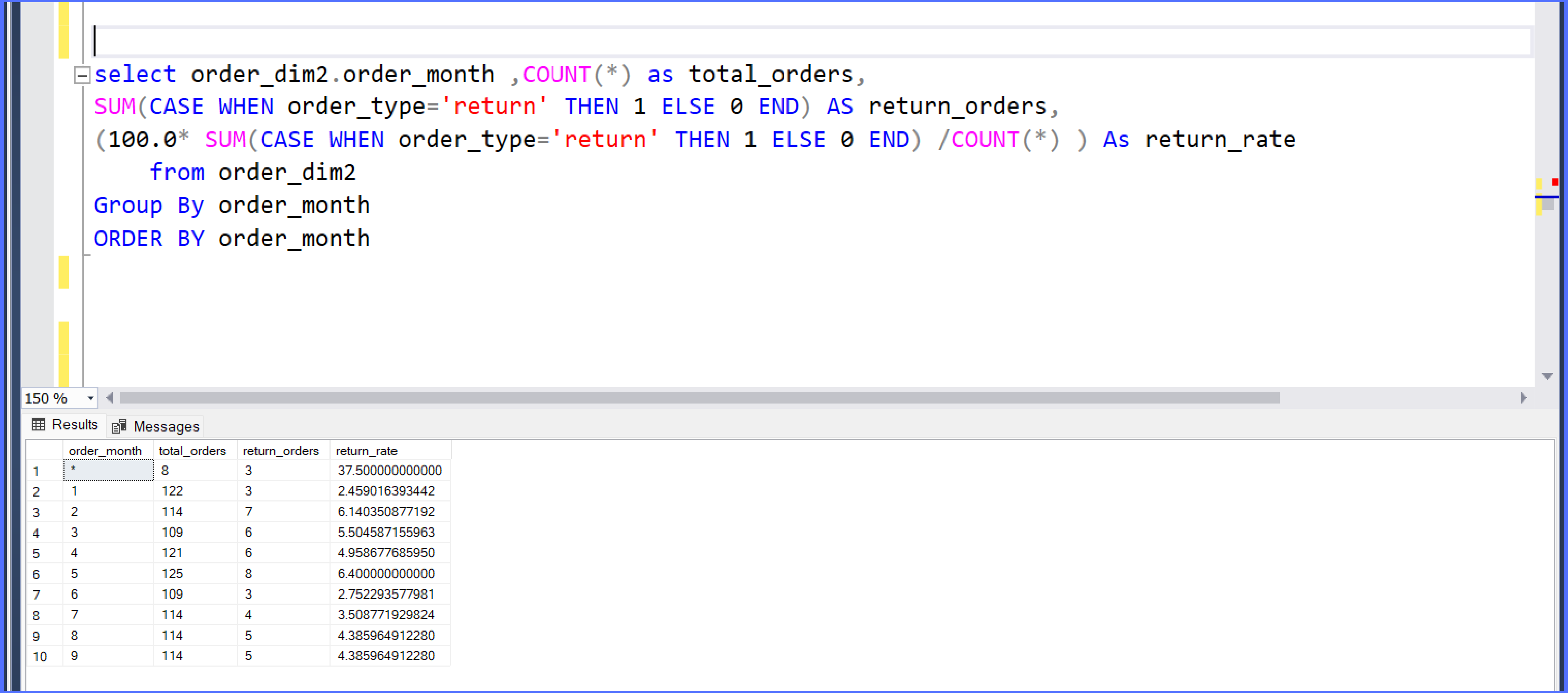


```
Create Table order_dim2(order_id FLOAT ,
order_date Date , order_day VARCHAR , order_month VARCHAR ,
order_type nvarchar(255) NULL);
SELECT convert(datetime  ,order_date ,121) from orders$;
INSERT into order_dim2(order_id,order_date ,order_day ,order_month ,
order_type) SELECT order_id , order_date,
Day(TRY_CAST(order_date AS DATE)) as order_day ,
Month(TRY_CAST(order_date as DATE)) as order_month,
order_type
from orders$;
select * from order_dim2;
```

	order_id	order_date	order_day	order_month	order_year	order_type
1	10000000001	2020-01-01	1	1	*	buy
2	10000000002	2020-01-01	1	1	*	buy
3	10000000003	2020-01-01	1	1	*	buy
4	10000000004	2020-01-01	1	1	*	buy
5	10000000005	2020-01-01	1	1	*	buy
6	10000000006	2020-01-01	1	1	*	buy
7	10000000007	2020-01-02	2	1	*	buy
8	10000000008	2020-01-02	2	1	*	buy
9	10000000009	2020-01-02	2	1	*	buy
10	10000000010	2020-01-03	3	1	*	buy
11	10000000011	2020-01-03	3	1	*	buy
12	10000000012	2020-01-03	3	1	*	buy
13	10000000013	2020-01-04	4	1	*	buy
14	10000000014	2020-01-04	4	1	*	buy
15	10000000015	2020-01-04	4	1	*	buy

Q-10

HOW MANY UNITS HAVE BEEN SOLD BY EACH BRAND? ALSO GET TOTAL RETURNED UNITS FOR EACH BRAND.

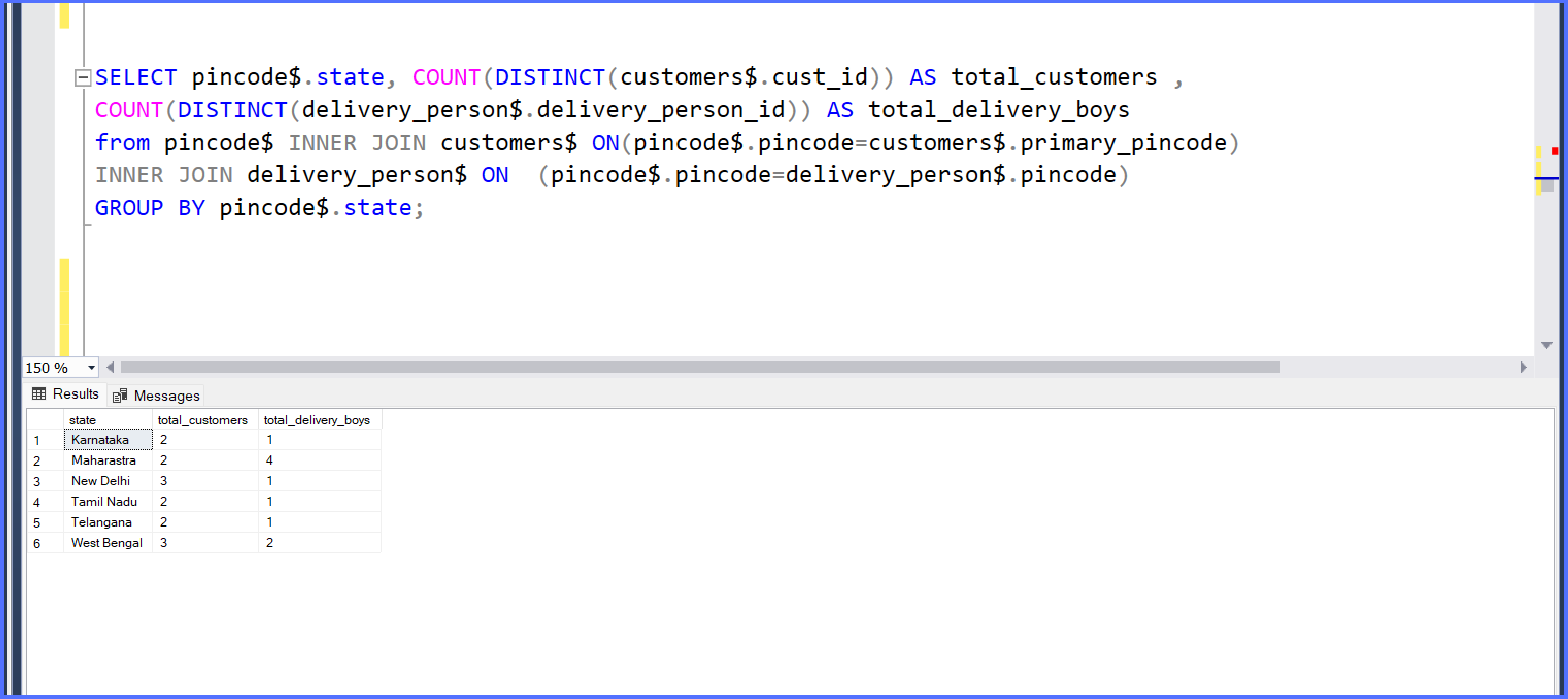


```
select order_dim2.order_month ,COUNT(*) as total_orders,
SUM(CASE WHEN order_type='return' THEN 1 ELSE 0 END) AS return_orders,
(100.0* SUM(CASE WHEN order_type='return' THEN 1 ELSE 0 END) /COUNT(*) ) As return_rate
    from order_dim2
Group By order_month
ORDER BY order_month
```

	order_month	total_orders	return_orders	return_rate
1	*	8	3	37.500000000000
2	1	122	3	2.459016393442
3	2	114	7	6.140350877192
4	3	109	6	5.504587155963
5	4	121	6	4.958677685950
6	5	125	8	6.400000000000
7	6	109	3	2.752293577981
8	7	114	4	3.508771929824
9	8	114	5	4.385964912280
10	9	114	5	4.385964912280

Q-11

HOW MANY DISTINCT CUSTOMERS AND DELIVERY BOYS ARE THERE IN EACH STATE?

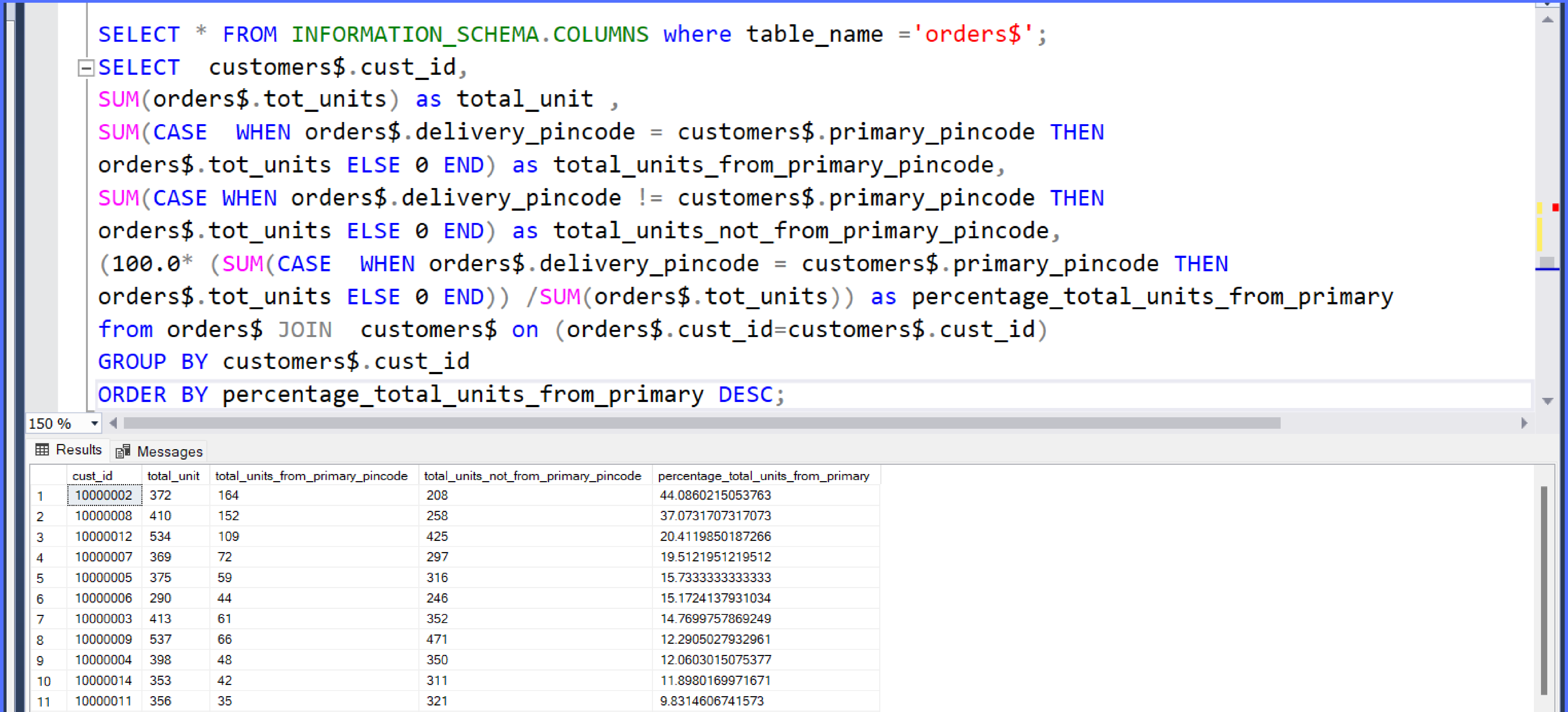


```sql
SELECT pincode$.state, COUNT(DISTINCT(customers$.cust_id)) AS total_customers ,
COUNT(DISTINCT(delivery_person$.delivery_person_id)) AS total_delivery_boys
from pincode$ INNER JOIN customers$ ON(pincode$.pincode=customers$.primary_pincode)
INNER JOIN delivery_person$ ON  (pincode$.pincode=delivery_person$.pincode)
GROUP BY pincode$.state;
```

	state	total_customers	total_delivery_boys
1	Karnataka	2	1
2	Maharastra	2	4
3	New Delhi	3	1
4	Tamil Nadu	2	1
5	Telangana	2	1
6	West Bengal	3	2

Q-12

FOR EVERY CUSTOMER, PRINT HOW MANY TOTAL UNITS WERE ORDERED, HOW MANY UNITS WERE ORDERED FROM THEIR PRIMARY_PINCODE AND HOW MANY WERE ORDERED NOT FROM THE PRIMARY_PINCODE. ALSO CALULATE THE PERCENTAGE OF TOTAL UNITS WHICH WERE ORDERED FROM PRIMARY_PINCODE(REMEMBER TO MULTIPLY THE NUMERATOR BY 100.0). SORT BY THE PERCENTAGE COLUMN IN DESCENDING ORDER.

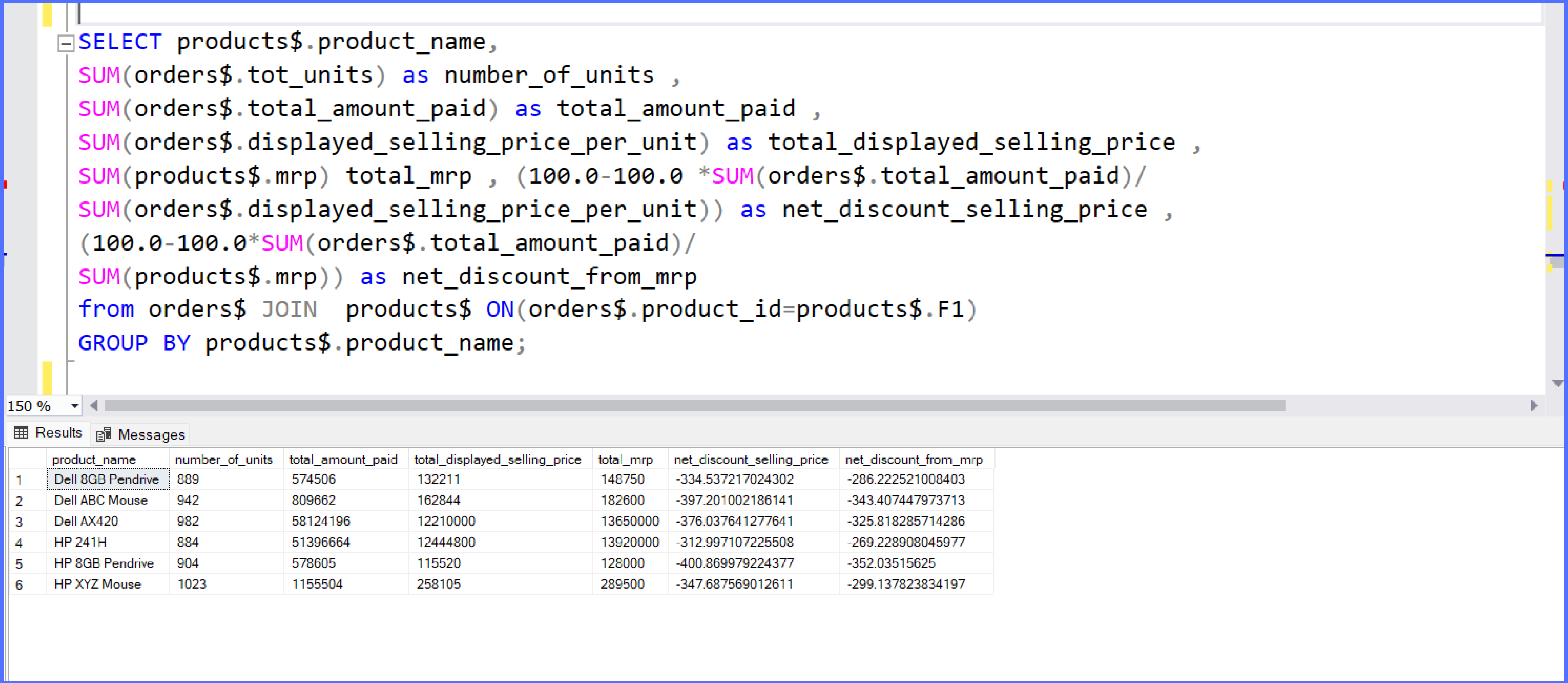


```
SELECT * FROM INFORMATION_SCHEMA.COLUMNS where table_name ='orders$';
SELECT  customers$.cust_id,
SUM(orders$.tot_units) as total_unit ,
SUM(CASE  WHEN orders$.delivery_pincode = customers$.primary_pincode THEN
orders$.tot_units ELSE 0 END) as total_units_from_primary_pincode,
SUM(CASE WHEN orders$.delivery_pincode != customers$.primary_pincode THEN
orders$.tot_units ELSE 0 END) as total_units_not_from_primary_pincode,
(100.0* (SUM(CASE  WHEN orders$.delivery_pincode = customers$.primary_pincode THEN
orders$.tot_units ELSE 0 END)) /SUM(orders$.tot_units)) as percentage_total_units_from_primary
from orders$ JOIN  customers$ on (orders$.cust_id=customers$.cust_id)
GROUP BY customers$.cust_id
ORDER BY percentage_total_units_from_primary DESC;
```

	cust_id	total_unit	total_units_from_primary_pincode	total_units_not_from_primary_pincode	percentage_total_units_from_primary
1	10000002	372	164	208	44.0860215053763
2	10000008	410	152	258	37.0731707317073
3	10000012	534	109	425	20.4119850187266
4	10000007	369	72	297	19.5121951219512
5	10000005	375	59	316	15.7333333333333
6	10000006	290	44	246	15.1724137931034
7	10000003	413	61	352	14.7699757869249
8	10000009	537	66	471	12.2905027932961
9	10000004	398	48	350	12.0603015075377
10	10000014	353	42	311	11.8980169971671
11	10000011	356	35	321	9.8314606741573

Q-13

FOR EACH PRODUCT NAME, PRINT THE SUM OF NUMBER OF UNITS, TOTAL AMOUNT PAID, TOTAL DISPLAYED SELLING PRICE, TOTAL MRP OF THESE UNITS, AND FINALLY THE NET DISCOUNT FROM SELLING PRICE. (I.E. 100.0 - 100.0 * TOTAL AMOUNT PAID / TOTAL DISPLAYED SELLING PRICE) & THE NET DISCOUNT FROM MRP (I.E. 100.0 - 100.0 * TOTAL AMOUNT PAID / TOTAL MRP)

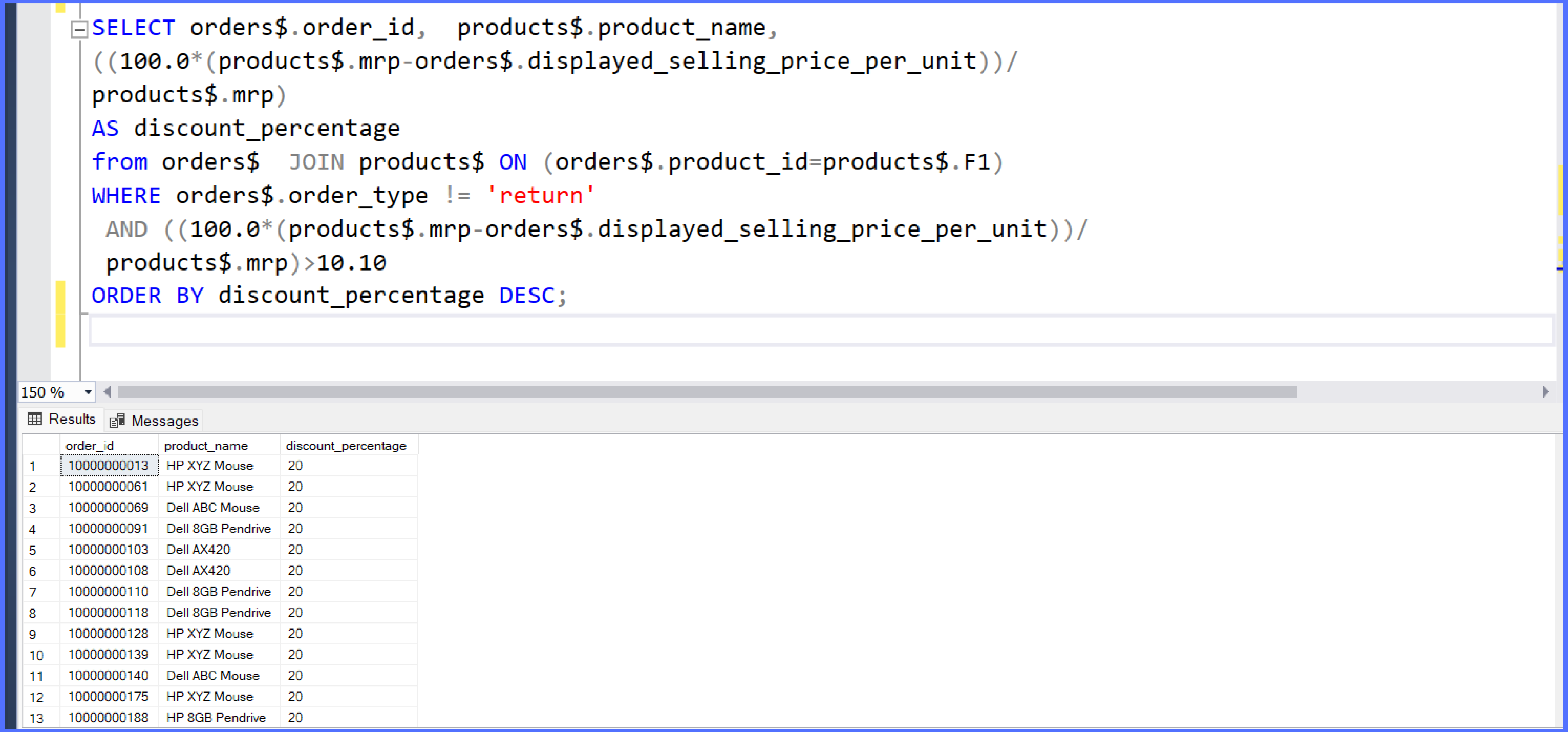


```
SELECT products$.product_name,
SUM(orders$.tot_units) as number_of_units ,
SUM(orders$.total_amount_paid) as total_amount_paid ,
SUM(orders$.displayed_selling_price_per_unit) as total_displayed_selling_price ,
SUM(products$.mrp) total_mrp , (100.0-100.0 *SUM(orders$.total_amount_paid)/
SUM(orders$.displayed_selling_price_per_unit)) as net_discount_selling_price ,
(100.0-100.0*SUM(orders$.total_amount_paid)/
SUM(products$.mrp)) as net_discount_from_mrp
from orders$ JOIN  products$ ON(orders$.product_id=products$.F1)
GROUP BY products$.product_name;
```

	product_name	number_of_units	total_amount_paid	total_displayed_selling_price	total_mrp	net_discount_selling_price	net_discount_from_mrp
1	Dell 8GB Pendrive	889	574506	132211	148750	-334.537217024302	-286.222521008403
2	Dell ABC Mouse	942	809662	162844	182600	-397.201002186141	-343.407447973713
3	Dell AX420	982	58124196	12210000	13650000	-376.037641277641	-325.818285714286
4	HP 241H	884	51396664	12444800	13920000	-312.997107225508	-269.228908045977
5	HP 8GB Pendrive	904	578605	115520	128000	-400.869979224377	-352.03515625
6	HP XYZ Mouse	1023	1155504	258105	289500	-347.687569012611	-299.137823834197

Q-14

. FOR EVERY ORDER_ID (EXCLUDE RETURNS), GET THE PRODUCT NAME AND CALCULATE THE DISCOUNT PERCENTAGE FROM SELLING PRICE. SORT BY HIGHEST DISCOUNT AND PRINT ONLY THOSE ROWS WHERE DISCOUNT PERCENTAGE WAS ABOVE 10.10%.

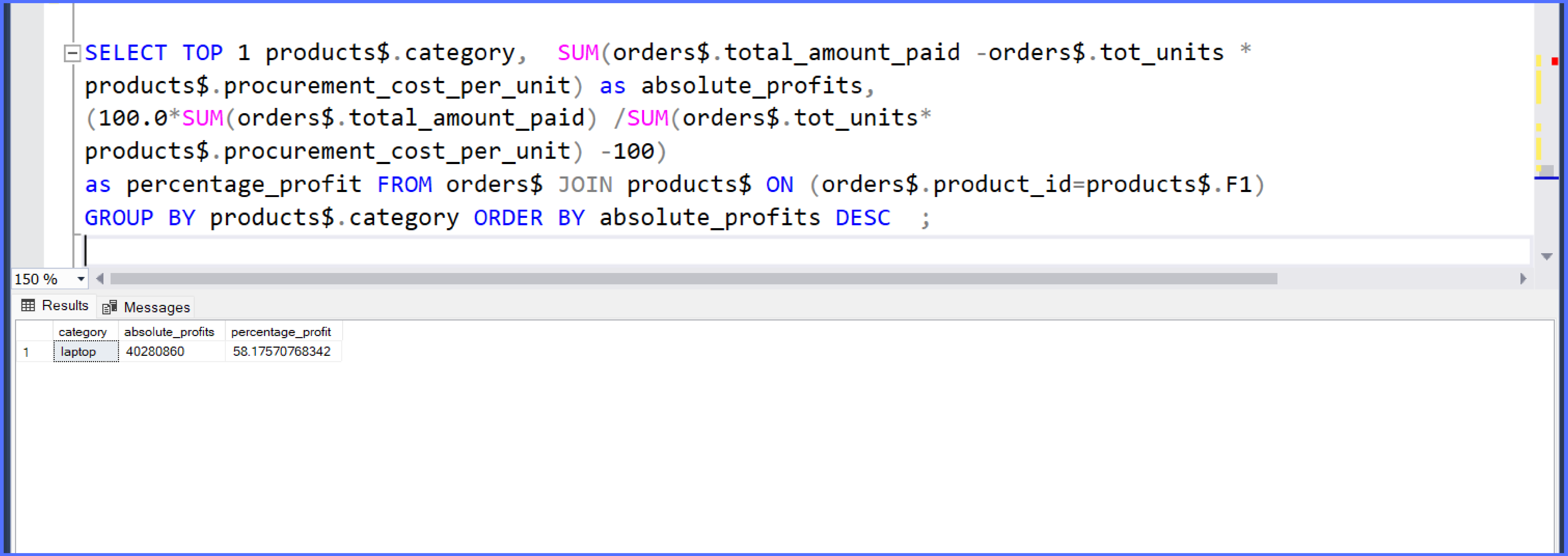


```sql
SELECT orders$.order_id,  products$.product_name,
((100.0*(products$.mrp-orders$.displayed_selling_price_per_unit))/
products$.mrp)
AS discount_percentage
from orders$  JOIN products$ ON (orders$.product_id=products$.F1)
WHERE orders$.order_type != 'return'
 AND ((100.0*(products$.mrp-orders$.displayed_selling_price_per_unit))/
 products$.mrp)>10.10
ORDER BY discount_percentage DESC;
```

	order_id	product_name	discount_percentage
1	10000000013	HP XYZ Mouse	20
2	10000000061	HP XYZ Mouse	20
3	10000000069	Dell ABC Mouse	20
4	10000000091	Dell 8GB Pendrive	20
5	10000000103	Dell AX420	20
6	10000000108	Dell AX420	20
7	10000000110	Dell 8GB Pendrive	20
8	10000000118	Dell 8GB Pendrive	20
9	10000000128	HP XYZ Mouse	20
10	10000000139	HP XYZ Mouse	20
11	10000000140	Dell ABC Mouse	20
12	10000000175	HP XYZ Mouse	20
13	10000000188	HP 8GB Pendrive	20

Q-15

. USING THE PER UNIT PROCUREMENT COST IN PRODUCT_DIM, FIND WHICH PRODUCT CATEGORY HAS MADE MOST PROFIT IN BOTH ABSOLUTE AMOUNT AND PERCENTAGE ABSOLUTE PROFIT = TOTAL AMT SOLD - TOTAL PROCUREMENT COST PERCENTAGE PROFIT = 100.0 * TOTAL AMT SOLD / TOTAL PROCUREMENT COST - 100.0

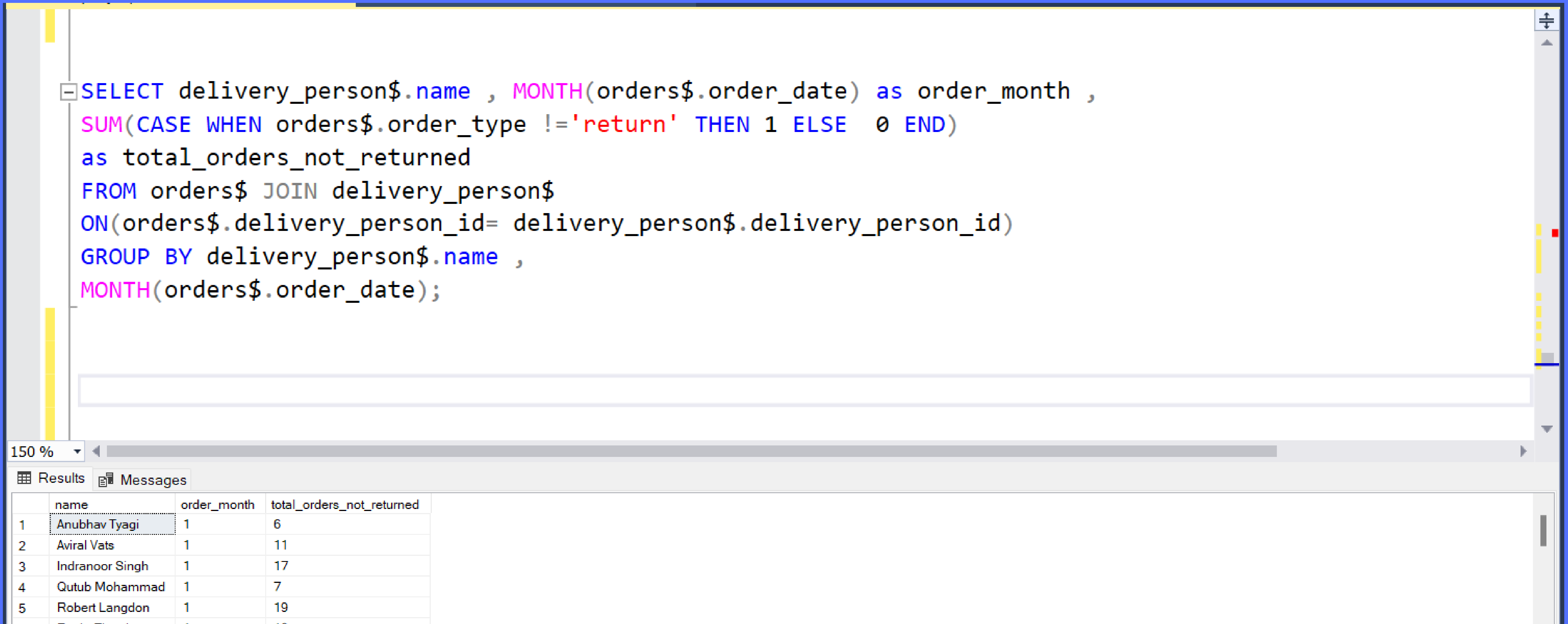


```
SELECT TOP 1 products$.category,  SUM(orders$.total_amount_paid -orders$.tot_units *
products$.procurement_cost_per_unit) as absolute_profits,
(100.0*SUM(orders$.total_amount_paid) /SUM(orders$.tot_units*
products$.procurement_cost_per_unit) -100)
as percentage_profit FROM orders$ JOIN products$ ON (orders$.product_id=products$.F1)
GROUP BY products$.category ORDER BY absolute_profits DESC  ;
```

	category	absolute_profits	percentage_profit
1	laptop	40280860	58.17570768342

Q-16

FOR EVERY DELIVERY PERSON(USE THEIR NAME), PRINT THE TOTAL NUMBER OF ORDER IDS (EXCLUDE RETURNS) BY MONTH IN SEPARATE COLUMNS I.E. THERE SHOULD BE ONE ROW FOR EACH DELIVERY_PERSON_ID AND 12 COLUMNS FOR EVERY MONTH IN THE YEAR

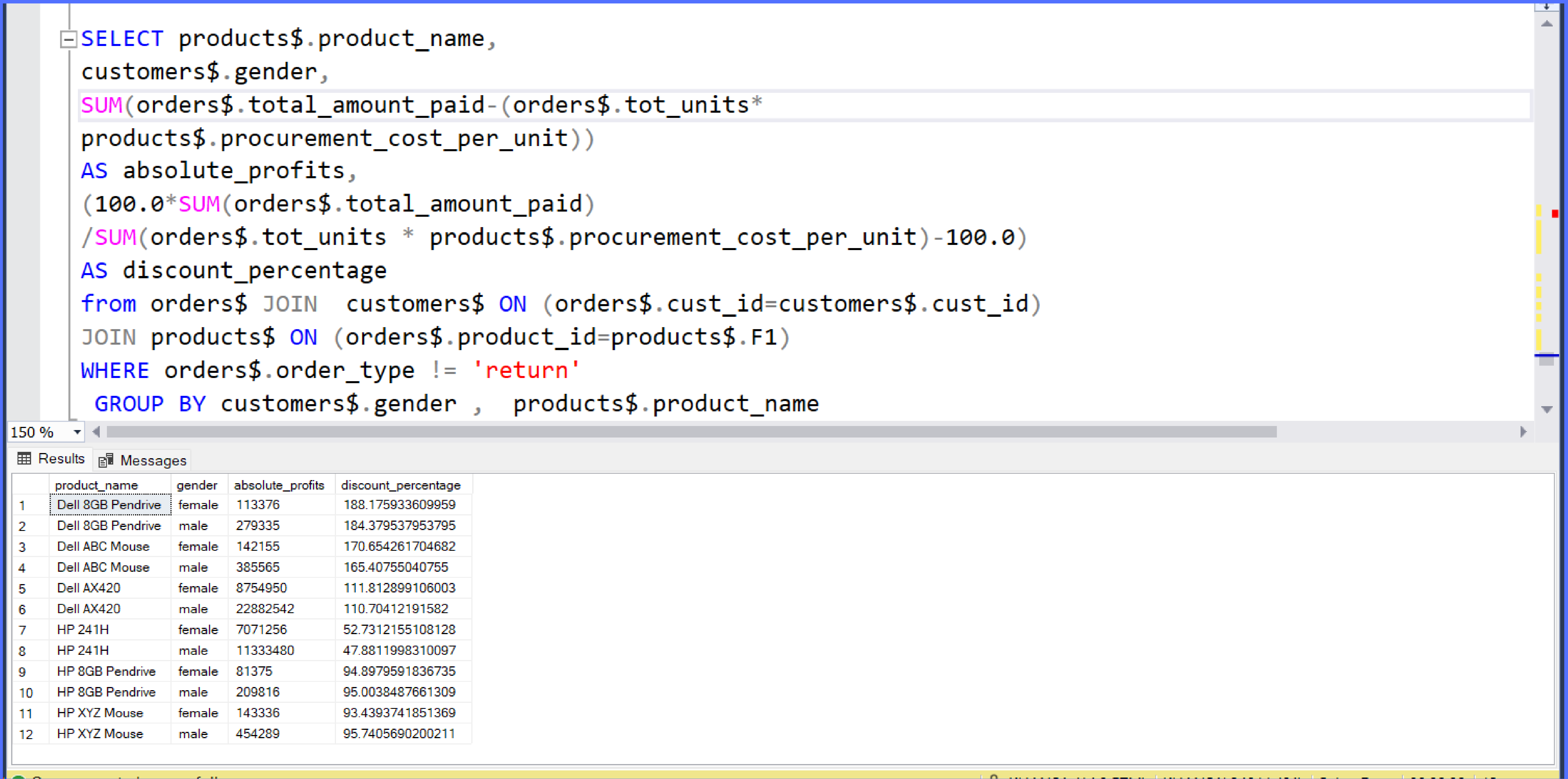


```sql
SELECT delivery_person$.name , MONTH(orders$.order_date) as order_month ,
SUM(CASE WHEN orders$.order_type !='return' THEN 1 ELSE  0 END)
as total_orders_not_returned
FROM orders$ JOIN delivery_person$
ON(orders$.delivery_person_id= delivery_person$.delivery_person_id)
GROUP BY delivery_person$.name ,
MONTH(orders$.order_date);
```

	name	order_month	total_orders_not_returned
1	Anubhav Tyagi	1	6
2	Aviral Vats	1	11
3	Indranoor Singh	1	17
4	Qutub Mohammad	1	7
5	Robert Langdon	1	19

Q-17

FOR EACH GENDER - MALE AND FEMALE - FIND THE ABSOLUTE AND PERCENTAGE PROFIT (LIKE IN Q15) BY PRODUCT NAME

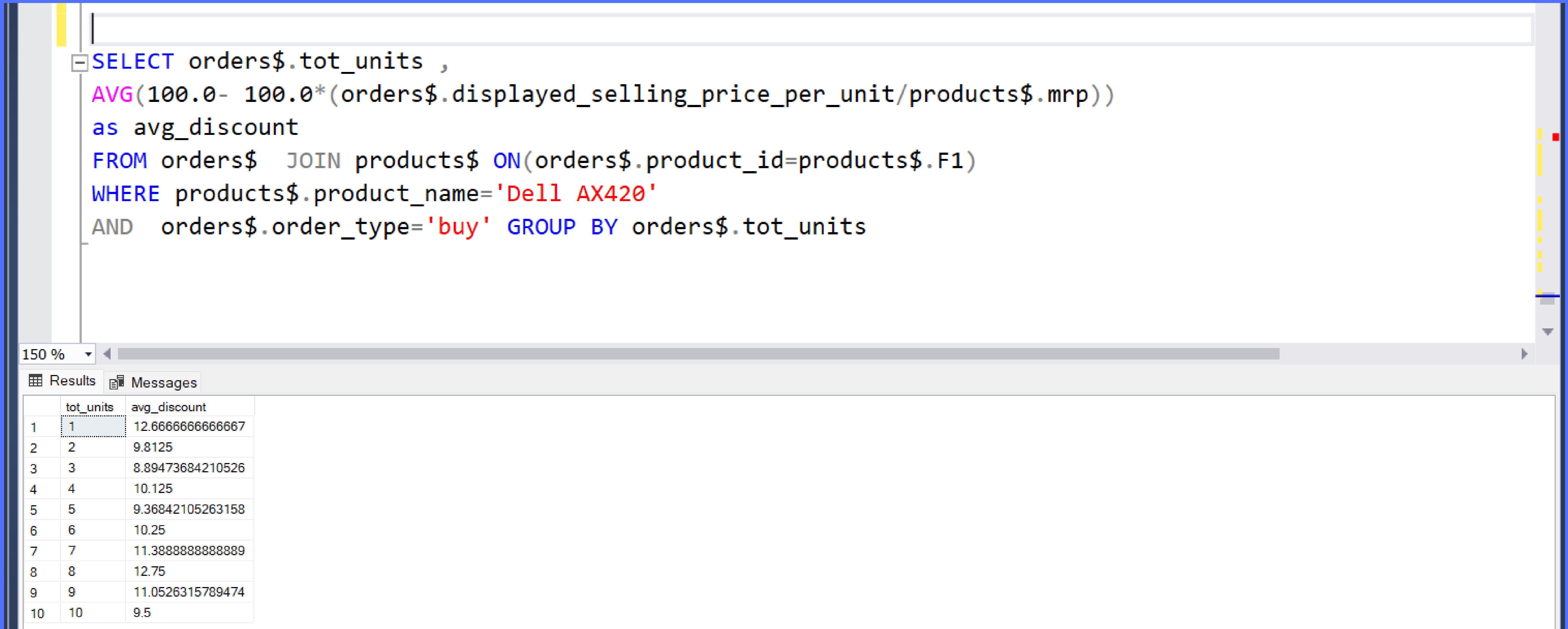


```
SELECT products$.product_name,
customers$.gender,
SUM(orders$.total_amount_paid-(orders$.tot_units*
products$.procurement_cost_per_unit))
AS absolute_profits,
(100.0*SUM(orders$.total_amount_paid)
/SUM(orders$.tot_units * products$.procurement_cost_per_unit)-100.0)
AS discount_percentage
from orders$ JOIN  customers$ ON (orders$.cust_id=customers$.cust_id)
JOIN products$ ON (orders$.product_id=products$.F1)
WHERE orders$.order_type != 'return'
 GROUP BY customers$.gender ,  products$.product_name
```

	product_name	gender	absolute_profits	discount_percentage
1	Dell 8GB Pendrive	female	113376	188.175933609959
2	Dell 8GB Pendrive	male	279335	184.379537953795
3	Dell ABC Mouse	female	142155	170.654261704682
4	Dell ABC Mouse	male	385565	165.40755040755
5	Dell AX420	female	8754950	111.812899106003
6	Dell AX420	male	22882542	110.70412191582
7	HP 241H	female	7071256	52.7312155108128
8	HP 241H	male	11333480	47.8811998310097
9	HP 8GB Pendrive	female	81375	94.8979591836735
10	HP 8GB Pendrive	male	209816	95.0038487661309
11	HP XYZ Mouse	female	143336	93.4393741851369
12	HP XYZ Mouse	male	454289	95.7405690200211

Q-18

GENERALLY THE MORE NUMBERS OF UNITS YOU BUY, THE MORE DISCOUNT SELLER WILL GIVE YOU. FOR 'DELL AX420' IS THERE A RELATIONSHIP BETWEEN NUMBER OF UNITS ORDERED AND AVERAGE DISCOUNT FROM SELLING PRICE? TAKE ONLY 'BUY' ORDER TYPES



```sql
SELECT orders$.tot_units ,
AVG(100.0- 100.0*(orders$.displayed_selling_price_per_unit/products$.mrp))
as avg_discount
FROM orders$  JOIN products$ ON(orders$.product_id=products$.F1)
WHERE products$.product_name='Dell AX420'
AND  orders$.order_type='buy' GROUP BY orders$.tot_units
```

	tot_units	avg_discount
1	1	12.6666666666667
2	2	9.8125
3	3	8.89473684210526
4	4	10.125
5	5	9.36842105263158
6	6	10.25
7	7	11.3888888888889
8	8	12.75
9	9	11.0526315789474
10	10	9.5

CONCLUSION

Through this project it was found that div into sales data has been a adventure challenges encountered have serves as stepping stones for my analytical skills and problem solving abilities ,this project insights that have the potential to drive meaningful change and innovation within bussiness ,use various sql functions ,join,filters helped to clear concepts thoroughly

thank you