MINI PROJECT
INSURANCE MANAGEMENT BY USING MS SQL

TABLES:

- Customer
- Policy
- Payment
- Claim

ATTRIBUTE:

- Customer

  CUSTOMER_ID
  CUSTOMER_NAME
  AGE
  GENDER
  OCCUPATION
  ANNUAL_INCOME
  ADDRESS
  PHONE_NUMBER
  EMAIL_ID

- Policy

  POLICY_ID
  CUSTOMER_ID
  POLICY_NAME
  POLICY_TYPE
  MAX_NO_OF_YEARS
  PREMIUM_RATE
  MAX_SUM_ASSURED

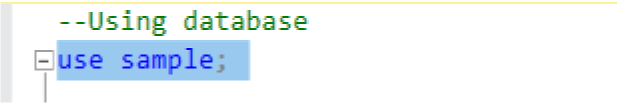
- Payment

  PAYMENT_ID
  CUSTOMER_ID
  AMOUNT
  PAYMENT_DATE
  PAYMENT_MODE

- Claim

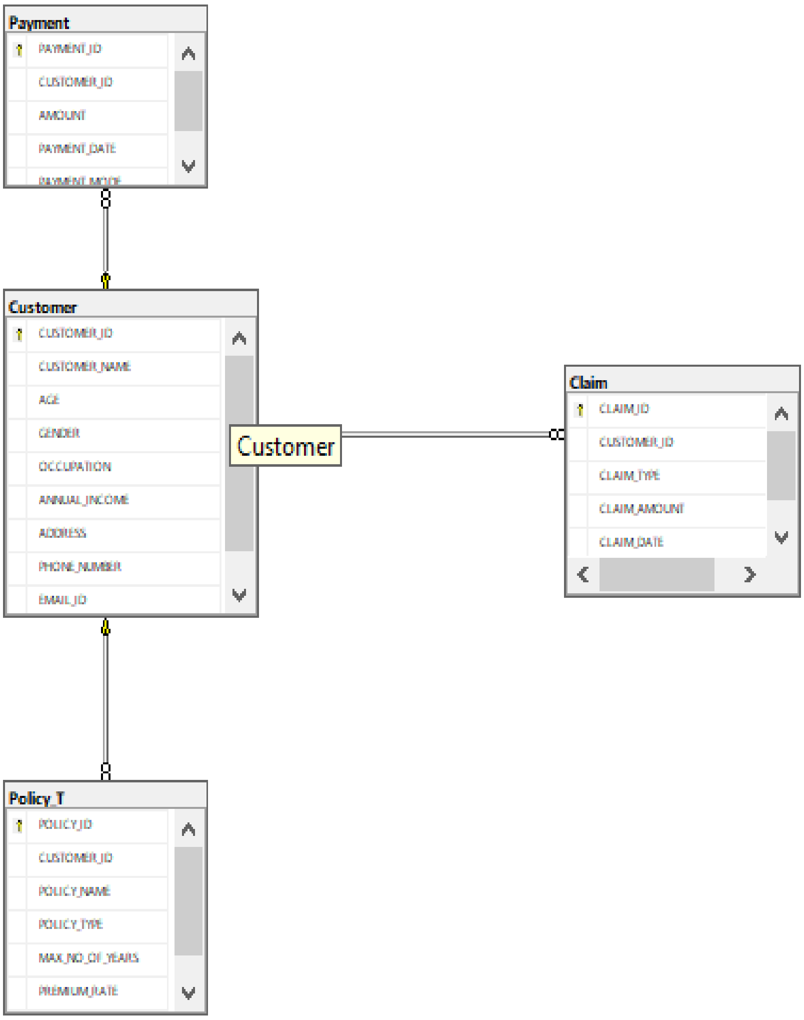
  CLAIM_ID
  CUSTOMER_ID
  CLAIM_TYPE
  CLAIM_AMOUNT
  CLAIM_DATE

USING THE DATABASE SAMPLE:



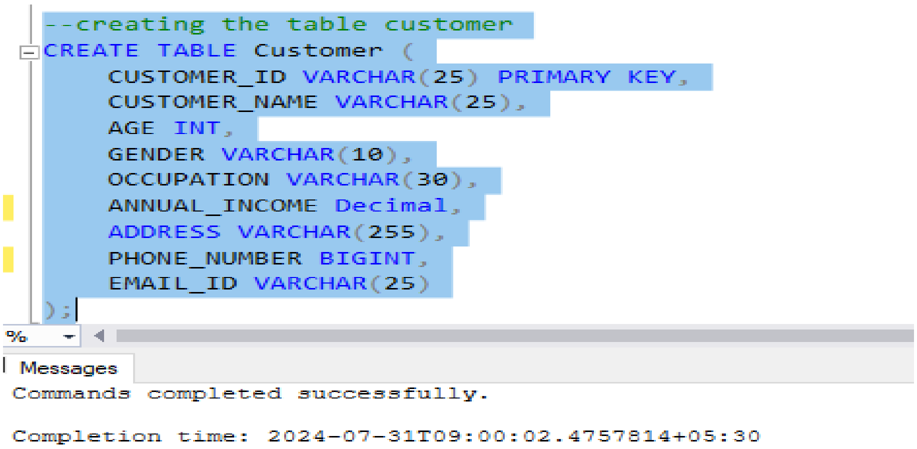
```
--Using database
use sample;
```

DATABASE DIAGRAM



CREATING THE TABLE:

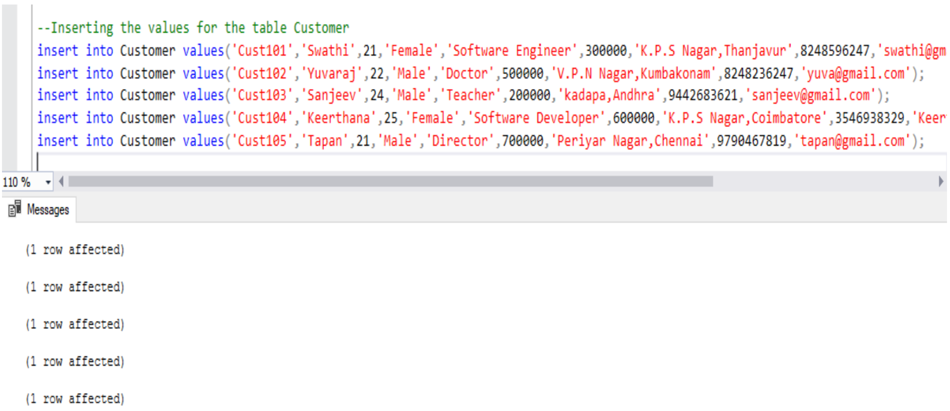
Creating the table Customer



```
--creating the table customer
CREATE TABLE Customer (
    CUSTOMER_ID VARCHAR(25) PRIMARY KEY,
    CUSTOMER_NAME VARCHAR(25),
    AGE INT,
    GENDER VARCHAR(10),
    OCCUPATION VARCHAR(30),
    ANNUAL_INCOME Decimal,
    ADDRESS VARCHAR(255),
    PHONE_NUMBER BIGINT,
    EMAIL_ID VARCHAR(25)
);
```

```
Commands completed successfully.

Completion time: 2024-07-31T09:00:02.4757814+05:30
```

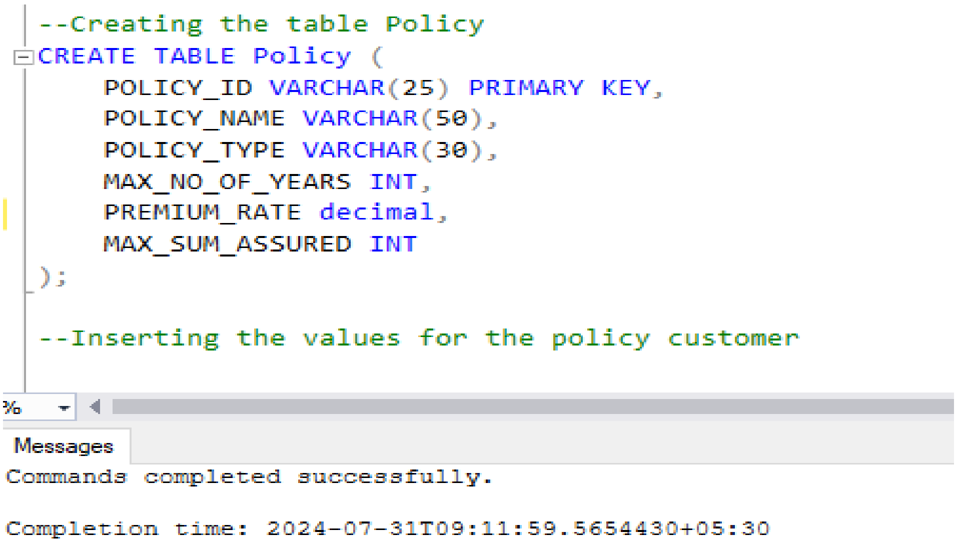
Inserting the values in the Customer table



```
--Inserting the values for the table Customer
insert into Customer values('Cust101','Swathi',21,'Female','Software Engineer',300000,'K.P.S Nagar,Thanjavur',8248596247,'swathi@gm
insert into Customer values('Cust102','Yuvaraj',22,'Male','Doctor',500000,'V.P.N Nagar,Kumbakonam',8248236247,'yuva@gmail.com');
insert into Customer values('Cust103','Sanjeev',24,'Male','Teacher',200000,'kadapa,Andhra',9442683621,'sanjeev@gmail.com');
insert into Customer values('Cust104','Keerthana',25,'Female','Software Developer',600000,'K.P.S Nagar,Coimbatore',3546938329,'Keer
insert into Customer values('Cust105','Tapan',21,'Male','Director',700000,'Periyar Nagar,Chennai',9790467819,'tapan@gmail.com');
```

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

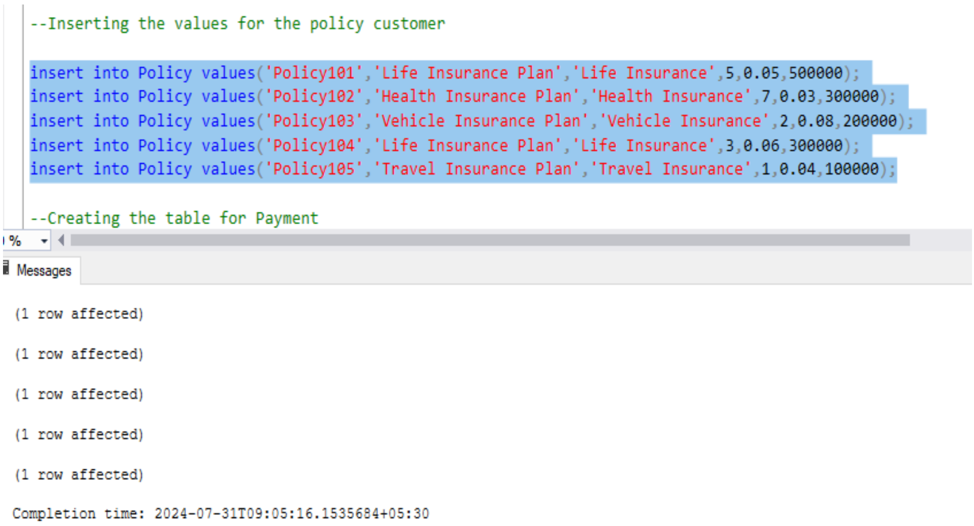
Creating the table Policy



```
--Creating the table Policy
CREATE TABLE Policy (
    POLICY_ID VARCHAR(25) PRIMARY KEY,
    POLICY_NAME VARCHAR(50),
    POLICY_TYPE VARCHAR(30),
    MAX_NO_OF_YEARS INT,
    PREMIUM_RATE decimal,
    MAX_SUM_ASSURED INT
);

--Inserting the values for the policy customer
```

```
Commands completed successfully.

Completion time: 2024-07-31T09:11:59.5654430+05:30
```

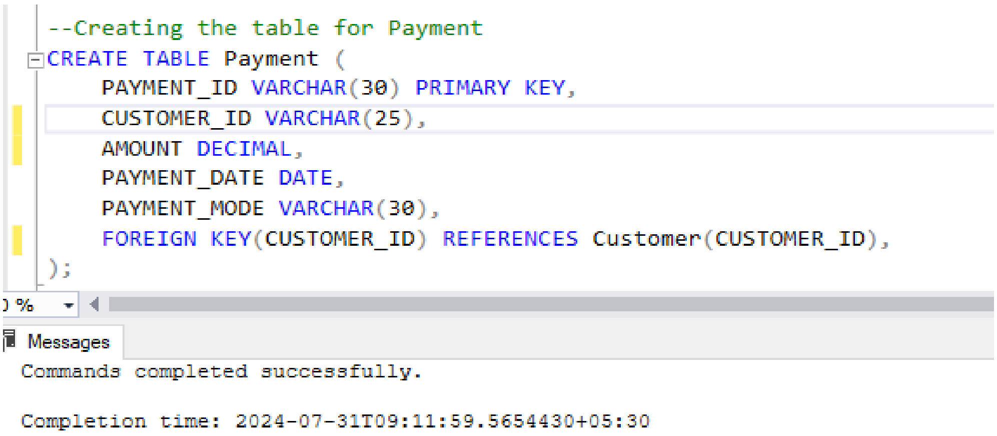
Inserting the values in the Policy table



```
--Inserting the values for the policy customer

insert into Policy values('Policy101','Life Insurance Plan','Life Insurance',5,0.05,500000);
insert into Policy values('Policy102','Health Insurance Plan','Health Insurance',7,0.03,300000);
insert into Policy values('Policy103','Vehicle Insurance Plan','Vehicle Insurance',2,0.08,200000);
insert into Policy values('Policy104','Life Insurance Plan','Life Insurance',3,0.06,300000);
insert into Policy values('Policy105','Travel Insurance Plan','Travel Insurance',1,0.04,100000);

--Creating the table for Payment
```

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-07-31T09:05:16.1535684+05:30
```

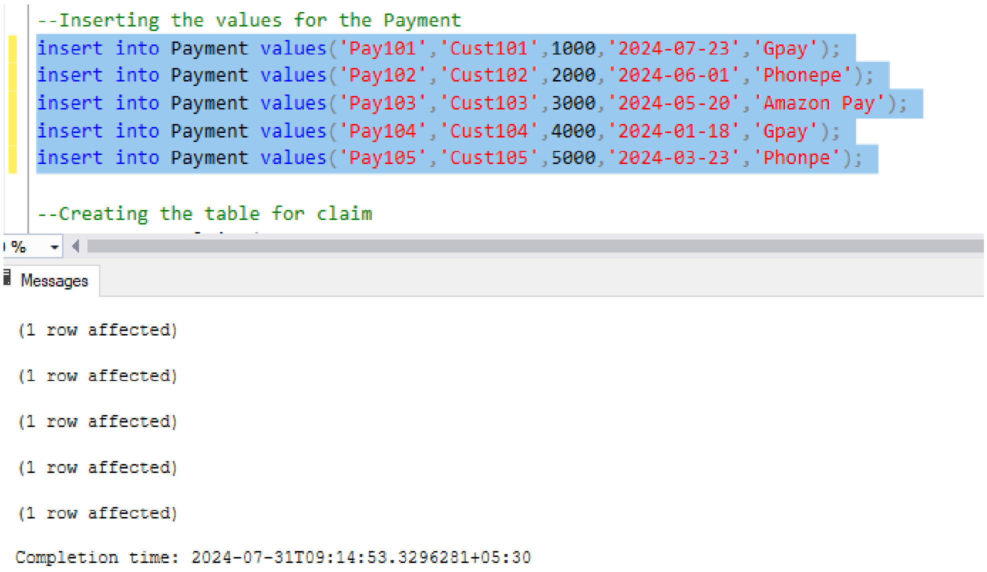
Creating the table Payment



```
--Creating the table for Payment
CREATE TABLE Payment (
    PAYMENT_ID VARCHAR(30) PRIMARY KEY,
    CUSTOMER_ID VARCHAR(25),
    AMOUNT DECIMAL,
    PAYMENT_DATE DATE,
    PAYMENT_MODE VARCHAR(30),
    FOREIGN KEY(CUSTOMER_ID) REFERENCES Customer(CUSTOMER_ID),
);
```

```
Commands completed successfully.

Completion time: 2024-07-31T09:11:59.5654430+05:30
```

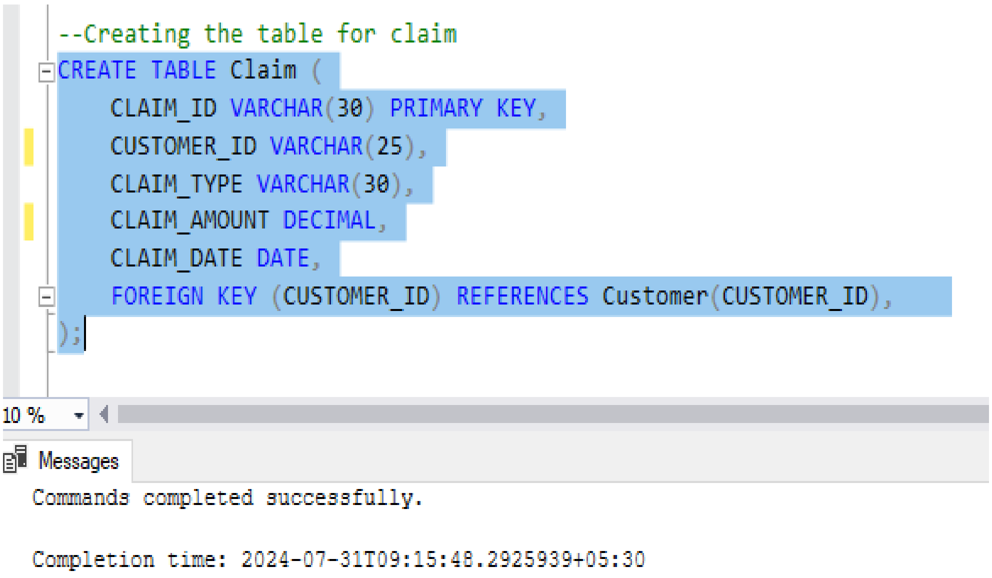
Inserting the values in the Payment table



```
--Inserting the values for the Payment
insert into Payment values('Pay101','Cust101',1000,'2024-07-23','Gpay');
insert into Payment values('Pay102','Cust102',2000,'2024-06-01','Phonepe');
insert into Payment values('Pay103','Cust103',3000,'2024-05-20','Amazon Pay');
insert into Payment values('Pay104','Cust104',4000,'2024-01-18','Gpay');
insert into Payment values('Pay105','Cust105',5000,'2024-03-23','Phonpe');

--Creating the table for claim
```

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-07-31T09:14:53.3296281+05:30
```

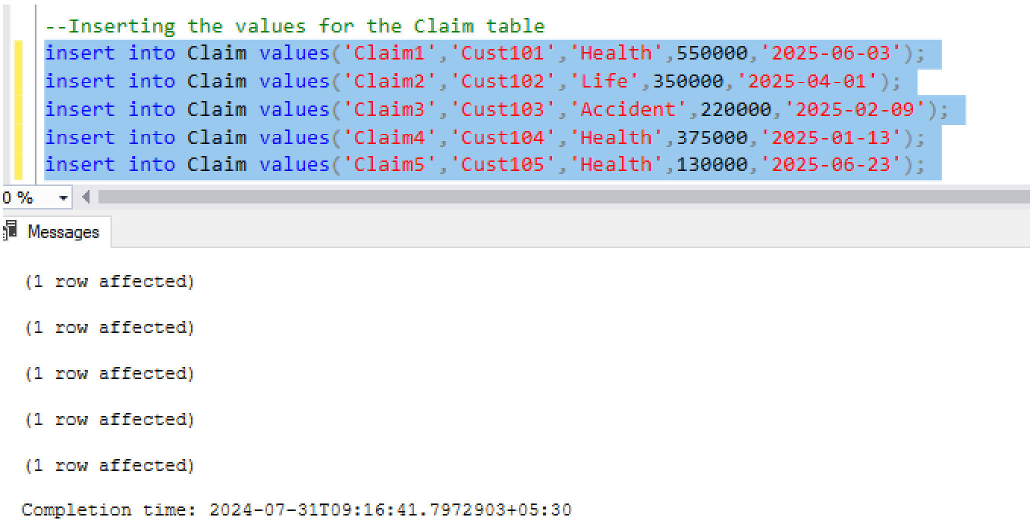
Creating the table Claim



```
--Creating the table for claim
CREATE TABLE Claim (
    CLAIM_ID VARCHAR(30) PRIMARY KEY,
    CUSTOMER_ID VARCHAR(25),
    CLAIM_TYPE VARCHAR(30),
    CLAIM_AMOUNT DECIMAL,
    CLAIM_DATE DATE,
    FOREIGN KEY (CUSTOMER_ID) REFERENCES Customer(CUSTOMER_ID),
);
```

```
Commands completed successfully.

Completion time: 2024-07-31T09:15:48.2925939+05:30
```

Inserting the values in the Claim table



```
--Inserting the values for the Claim table
insert into Claim values('Claim1','Cust101','Health',550000,'2025-06-03');
insert into Claim values('Claim2','Cust102','Life',350000,'2025-04-01');
insert into Claim values('Claim3','Cust103','Accident',220000,'2025-02-09');
insert into Claim values('Claim4','Cust104','Health',375000,'2025-01-13');
insert into Claim values('Claim5','Cust105','Health',130000,'2025-06-23');
```

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-07-31T09:16:41.7972903+05:30
```

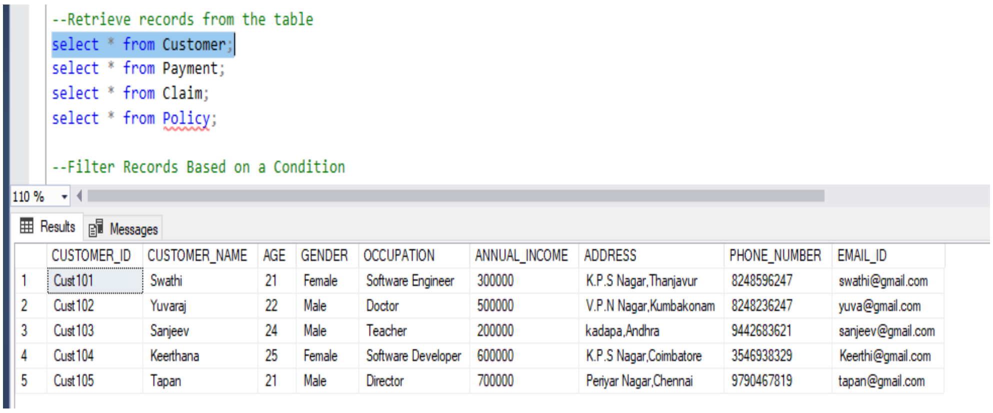
1. Retrieve All Records from a Table

Query Task: Select all records from the customers table.

QUERY:

select * from Customer;

OUTPUT:



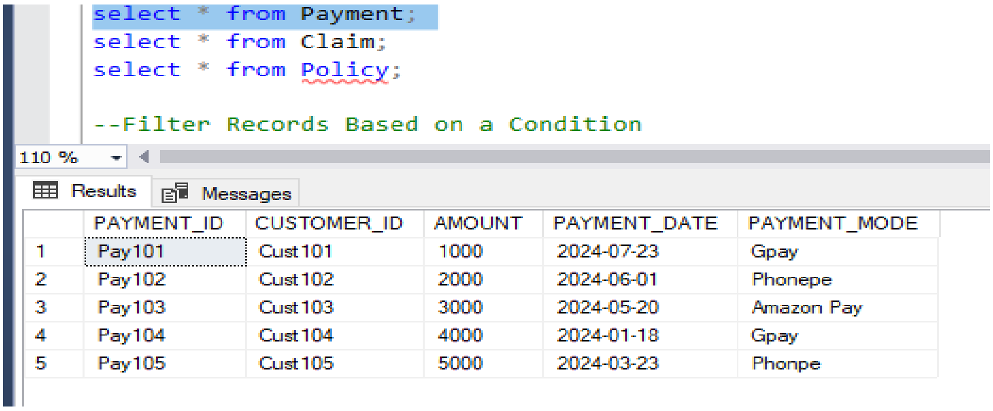
```
--Retrieve records from the table
select * from Customer;
select * from Payment;
select * from Claim;
select * from Policy;

--Filter Records Based on a Condition
```

	CUSTOMER_ID	CUSTOMER_NAME	AGE	GENDER	OCCUPATION	ANNUAL_INCOME	ADDRESS	PHONE_NUMBER	EMAIL_ID
1	Cust101	Swathi	21	Female	Software Engineer	300000	K.P.S Nagar,Thanjavur	8248596247	swathi@gmail.com
2	Cust102	Yuvaraj	22	Male	Doctor	500000	V.P.N Nagar,Kumbakonam	8248236247	yuva@gmail.com
3	Cust103	Sanjeev	24	Male	Teacher	200000	kadapa,Andhra	9442683621	sanjeev@gmail.com
4	Cust104	Keerthana	25	Female	Software Developer	600000	K.P.S Nagar,Coimbatore	3546938329	Keerthi@gmail.com
5	Cust105	Tapan	21	Male	Director	700000	Periyar Nagar,Chennai	9790467819	tapan@gmail.com

QUERY:

select * from Payment;

OUTPUT:



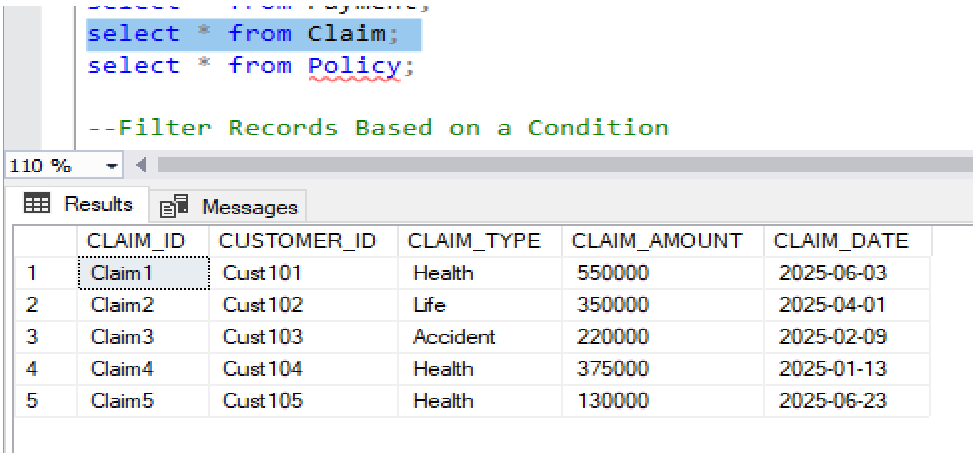
```
select * from Payment;
select * from Claim;
select * from Policy;

--Filter Records Based on a Condition
```

	PAYMENT_ID	CUSTOMER_ID	AMOUNT	PAYMENT_DATE	PAYMENT_MODE
1	Pay101	Cust101	1000	2024-07-23	Gpay
2	Pay102	Cust102	2000	2024-06-01	Phonepe
3	Pay103	Cust103	3000	2024-05-20	Amazon Pay
4	Pay104	Cust104	4000	2024-01-18	Gpay
5	Pay105	Cust105	5000	2024-03-23	Phonpe

QUERY:

select * from Claim;

OUTPUT:



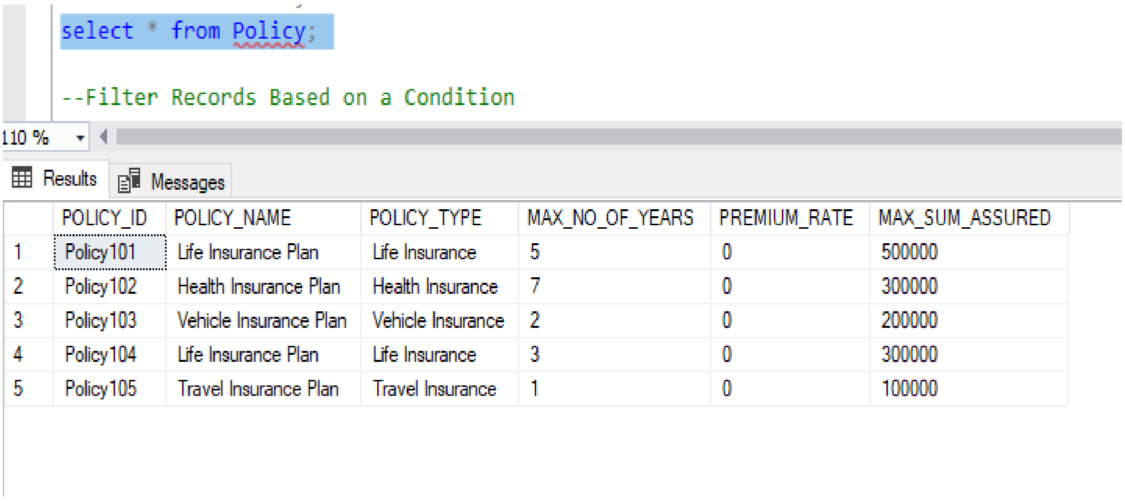
```
select * from Claim;
select * from Policy;

--Filter Records Based on a Condition
```

	CLAIM_ID	CUSTOMER_ID	CLAIM_TYPE	CLAIM_AMOUNT	CLAIM_DATE
1	Claim1	Cust101	Health	550000	2025-06-03
2	Claim2	Cust102	Life	350000	2025-04-01
3	Claim3	Cust103	Accident	220000	2025-02-09
4	Claim4	Cust104	Health	375000	2025-01-13
5	Claim5	Cust105	Health	130000	2025-06-23

QUERY:

select * from Policy_T;

OUTPUT:



```
select * from Policy;

--Filter Records Based on a Condition
```

	POLICY_ID	POLICY_NAME	POLICY_TYPE	MAX_NO_OF_YEARS	PREMIUM_RATE	MAX_SUM_ASSURED
1	Policy101	Life Insurance Plan	Life Insurance	5	0	500000
2	Policy102	Health Insurance Plan	Health Insurance	7	0	300000
3	Policy103	Vehicle Insurance Plan	Vehicle Insurance	2	0	200000
4	Policy104	Life Insurance Plan	Life Insurance	3	0	300000
5	Policy105	Travel Insurance Plan	Travel Insurance	1	0	100000

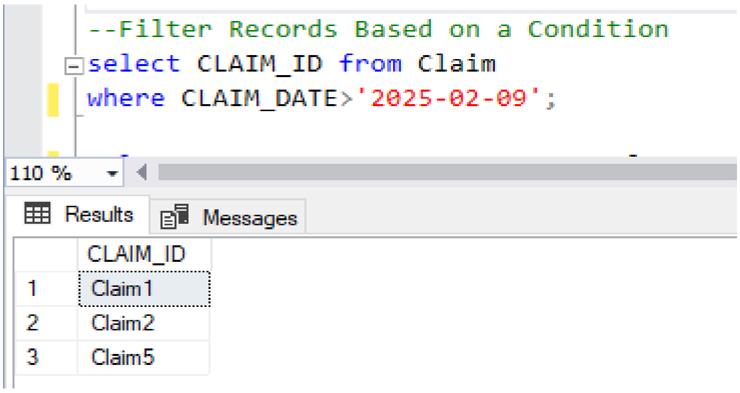
2. Filter Records Based on a Condition

Query Task: Select all orders from the orders table where the order date is after January 1, 2023.

QUERY:

select CLAIM_ID from Claim
where CLAIM_DATE>'2025-02-09';

OUTPUT:



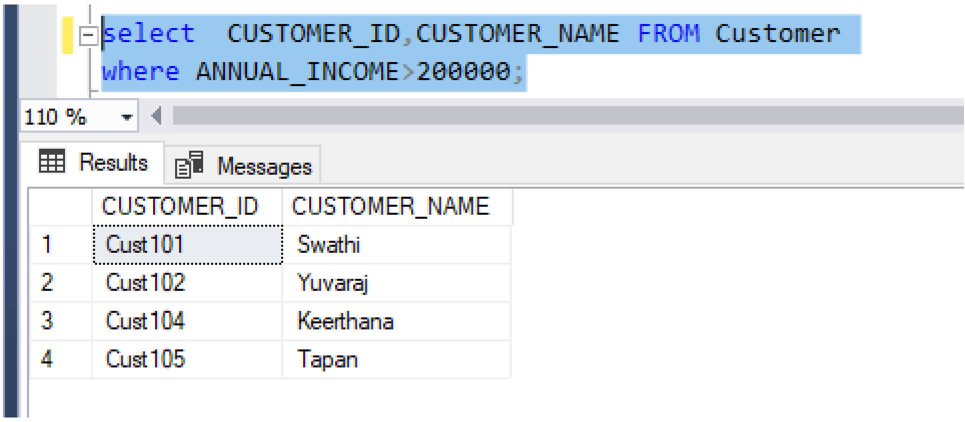
```
--Filter Records Based on a Condition
select CLAIM_ID from Claim
where CLAIM_DATE>'2025-02-09';
```

	CLAIM_ID
1	Claim1
2	Claim2
3	Claim5

QUERY:

select CUSTOMER_ID,CUSTOMER_NAME FROM Customer
where ANNUAL_INCOME>200000;

OUTPUT:



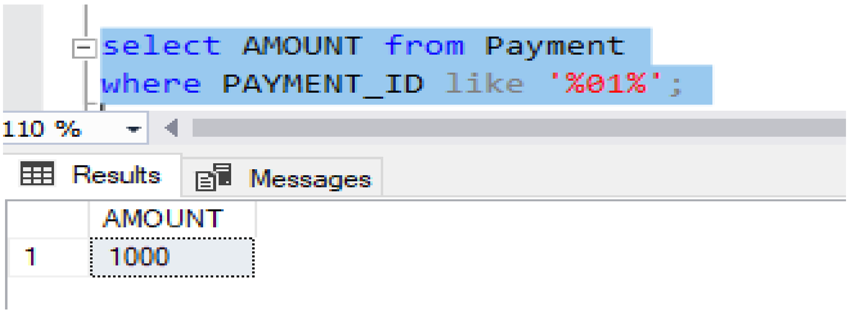
```
select  CUSTOMER_ID,CUSTOMER_NAME FROM Customer
where ANNUAL_INCOME>200000;
```

	CUSTOMER_ID	CUSTOMER_NAME
1	Cust101	Swathi
2	Cust102	Yuvaraj
3	Cust104	Keerthana
4	Cust105	Tapan

QUERY:

select AMOUNT from Payment
where PAYMENT_ID like '%01%';

OUTPUT:



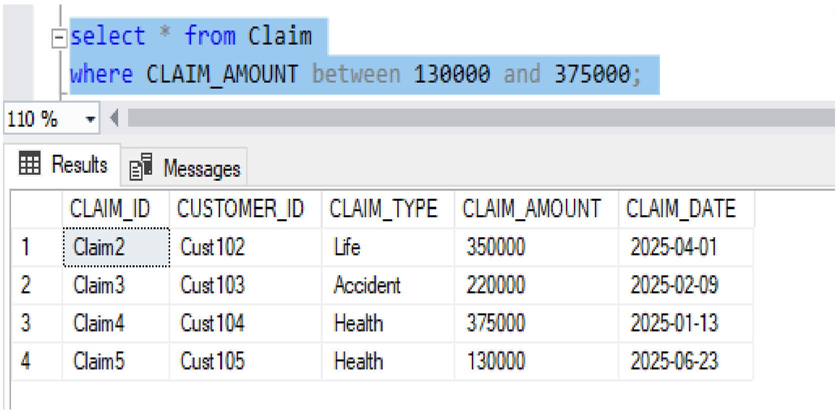
```
select AMOUNT from Payment
where PAYMENT_ID like '%01%';
```

	AMOUNT
1	1000

QUERY:

```
select * from Claim
where CLAIM_AMOUNT between 130000 and 375000;
```

OUTPUT:



```
select * from Claim
where CLAIM_AMOUNT between 130000 and 375000;
```

110 %

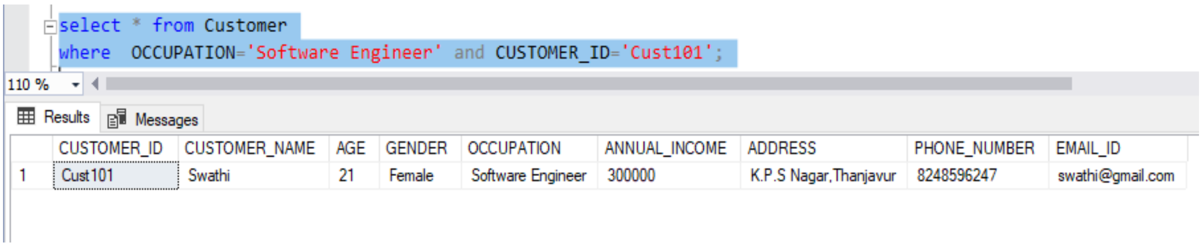
Results | Messages

	CLAIM_ID	CUSTOMER_ID	CLAIM_TYPE	CLAIM_AMOUNT	CLAIM_DATE
1	Claim2	Cust102	Life	350000	2025-04-01
2	Claim3	Cust103	Accident	220000	2025-02-09
3	Claim4	Cust104	Health	375000	2025-01-13
4	Claim5	Cust105	Health	130000	2025-06-23

QUERY:

```
select * from Customer
where  OCCUPATION='Software Engineer' and CUSTOMER_ID='Cust101';
```

OUTPUT:



```
select * from Customer
where  OCCUPATION='Software Engineer' and CUSTOMER_ID='Cust101';
```

110 %

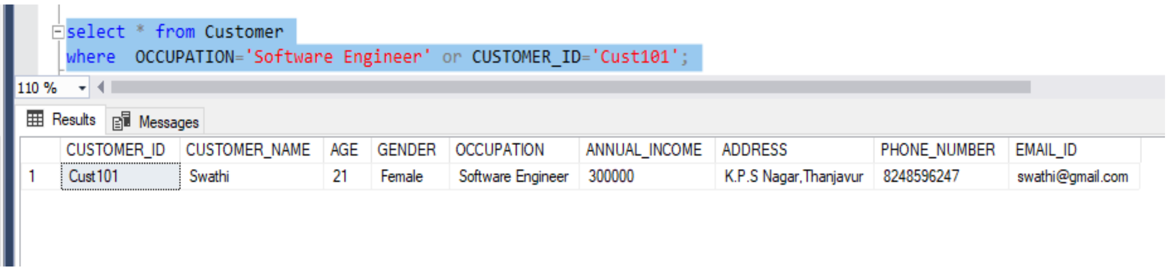
Results | Messages

	CUSTOMER_ID	CUSTOMER_NAME	AGE	GENDER	OCCUPATION	ANNUAL_INCOME	ADDRESS	PHONE_NUMBER	EMAIL_ID
1	Cust101	Swathi	21	Female	Software Engineer	300000	K.P.S Nagar,Thanjavur	8248596247	swathi@gmail.com

QUERY:

```
select * from Customer
where  OCCUPATION='Software Engineer' or CUSTOMER_ID='Cust101';
```

OUTPUT:



```
select * from Customer
where  OCCUPATION='Software Engineer' or CUSTOMER_ID='Cust101';
```

110 %

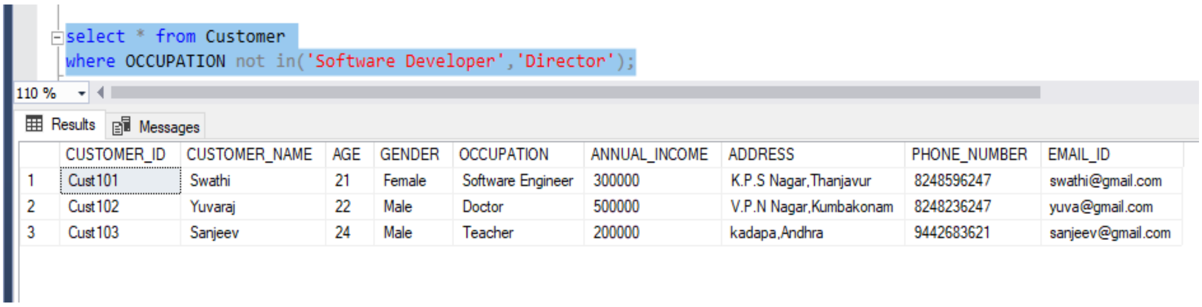
Results | Messages

	CUSTOMER_ID	CUSTOMER_NAME	AGE	GENDER	OCCUPATION	ANNUAL_INCOME	ADDRESS	PHONE_NUMBER	EMAIL_ID
1	Cust101	Swathi	21	Female	Software Engineer	300000	K.P.S Nagar,Thanjavur	8248596247	swathi@gmail.com

QUERY:

```
select * from Customer
where OCCUPATION not in('Software Developer','Director');
```

OUTPUT:



```
select * from Customer
where OCCUPATION not in('Software Developer','Director');
```

110 %

Results | Messages

	CUSTOMER_ID	CUSTOMER_NAME	AGE	GENDER	OCCUPATION	ANNUAL_INCOME	ADDRESS	PHONE_NUMBER	EMAIL_ID
1	Cust101	Swathi	21	Female	Software Engineer	300000	K.P.S Nagar,Thanjavur	8248596247	swathi@gmail.com
2	Cust102	Yuvaraj	22	Male	Doctor	500000	V.P.N Nagar,Kumbakonam	8248236247	yuva@gmail.com
3	Cust103	Sanjeev	24	Male	Teacher	200000	kadapa,Andhra	9442683621	sanjeev@gmail.com

3. Join Two Tables

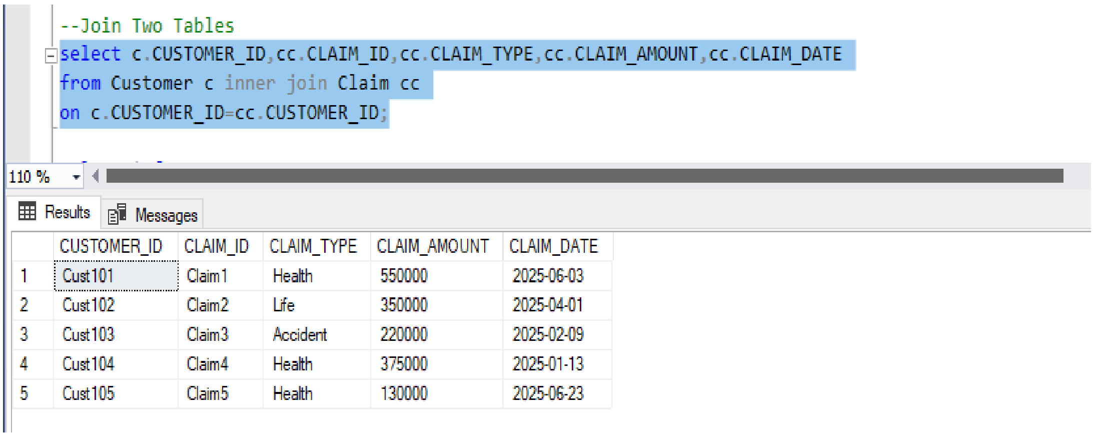
Query Task: Retrieve the names of customers along with their order IDs from the customers and orders tables.

Hint: Use an INNER JOIN/outer join/cross join to combine data from both tables based on a common column.

QUERY:

```
select
c.CUSTOMER_ID,cc.CLAIM_ID,cc.CLAIM_TYPE,cc.CLAIM_AMOUNT,cc.CLAIM_DATE
from Customer c inner join Claim cc
on c.CUSTOMER_ID=cc.CUSTOMER_ID;
```

OUTPUT:

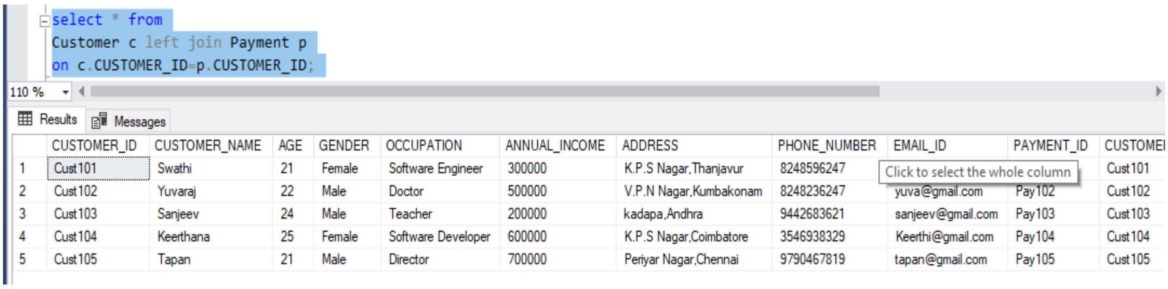


```
--Join Two Tables
select c.CUSTOMER_ID,cc.CLAIM_ID,cc.CLAIM_TYPE,cc.CLAIM_AMOUNT,cc.CLAIM_DATE
from Customer c inner join Claim cc
on c.CUSTOMER_ID=cc.CUSTOMER_ID;
```

	CUSTOMER_ID	CLAIM_ID	CLAIM_TYPE	CLAIM_AMOUNT	CLAIM_DATE
1	Cust101	Claim1	Health	550000	2025-06-03
2	Cust102	Claim2	Life	350000	2025-04-01
3	Cust103	Claim3	Accident	220000	2025-02-09
4	Cust104	Claim4	Health	375000	2025-01-13
5	Cust105	Claim5	Health	130000	2025-06-23

QUERY:

```
select * from
Customer c left join Payment p
on c.CUSTOMER_ID=p.CUSTOMER_ID;
```



```
select * from
Customer c left join Payment p
on c.CUSTOMER_ID=p.CUSTOMER_ID;
```

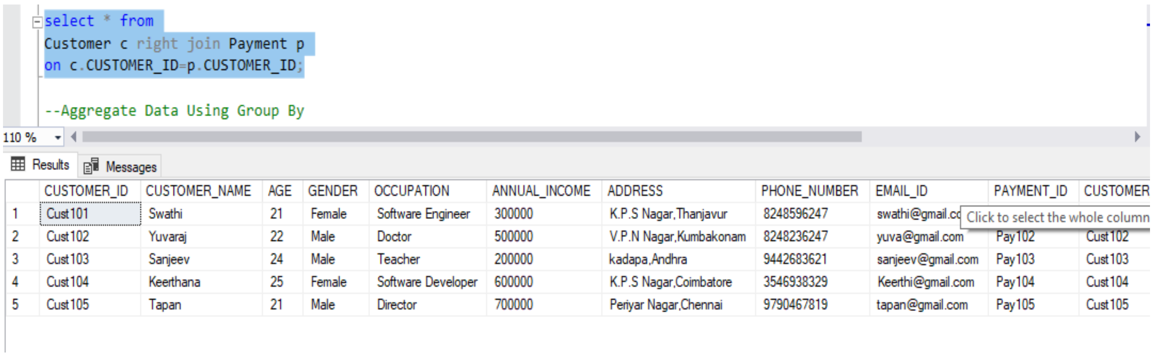
	CUSTOMER_ID	CUSTOMER_NAME	AGE	GENDER	OCCUPATION	ANNUAL_INCOME	ADDRESS	PHONE_NUMBER	EMAIL_ID	PAYMENT_ID	CUSTOMEI
1	Cust101	Swathi	21	Female	Software Engineer	300000	K.P.S Nagar,Thanjavur	8248596247	Click to select the whole column		Cust101
2	Cust102	Yuvaraj	22	Male	Doctor	500000	V.P.N Nagar,Kumbakonam	8248236247	yuva@gmail.com	Pay102	Cust102
3	Cust103	Sanjeev	24	Male	Teacher	200000	kadapa,Andhra	9442683621	sanjeev@gmail.com	Pay103	Cust103
4	Cust104	Keerthana	25	Female	Software Developer	600000	K.P.S Nagar,Coimbatore	3546938329	Keerthi@gmail.com	Pay104	Cust104
5	Cust105	Tapan	21	Male	Director	700000	Periyar Nagar,Chennai	9790467819	tapan@gmail.com	Pay105	Cust105

OUTPUT:

QUERY:

```
select * from
Customer c right join Payment p
on c.CUSTOMER_ID=p.CUSTOMER_ID;
```

OUTPUT:



```
select * from
Customer c right join Payment p
on c.CUSTOMER_ID=p.CUSTOMER_ID;

--Aggregate Data Using Group By
```

	CUSTOMER_ID	CUSTOMER_NAME	AGE	GENDER	OCCUPATION	ANNUAL_INCOME	ADDRESS	PHONE_NUMBER	EMAIL_ID	PAYMENT_ID	CUSTOMER
1	Cust101	Swathi	21	Female	Software Engineer	300000	K.P.S Nagar,Thanjavur	8248596247	swathi@gmail.c		
2	Cust102	Yuvaraj	22	Male	Doctor	500000	V.P.N Nagar,Kumbakonam	8248236247	yuva@gmail.com	Pay102	Cust102
3	Cust103	Sanjeev	24	Male	Teacher	200000	kadapa,Andhra	9442683621	sanjeev@gmail.com	Pay103	Cust103
4	Cust104	Keerthana	25	Female	Software Developer	600000	K.P.S Nagar,Coimbatore	3546938329	Keerthi@gmail.com	Pay104	Cust104
5	Cust105	Tapan	21	Male	Director	700000	Periyar Nagar,Chennai	9790467819	tapan@gmail.com	Pay105	Cust105

4.

Aggregate Data Using Group By

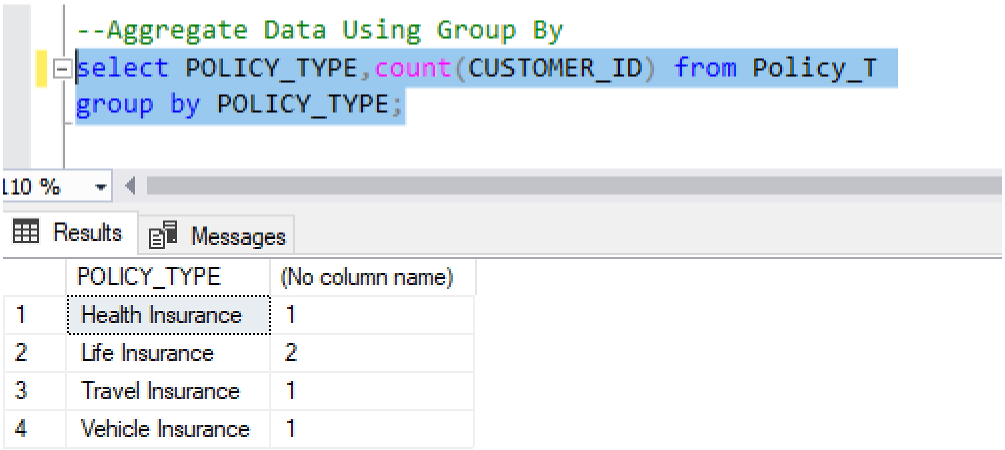
Query Task: Find the total number of orders placed by each customer.

Hint: Use the GROUP BY clause to group records and COUNT to aggregate.

QUERY:

```
select POLICY_TYPE,count(CUSTOMER_ID) from Policy_T

group by POLICY_TYPE;
```

OUTPUT:



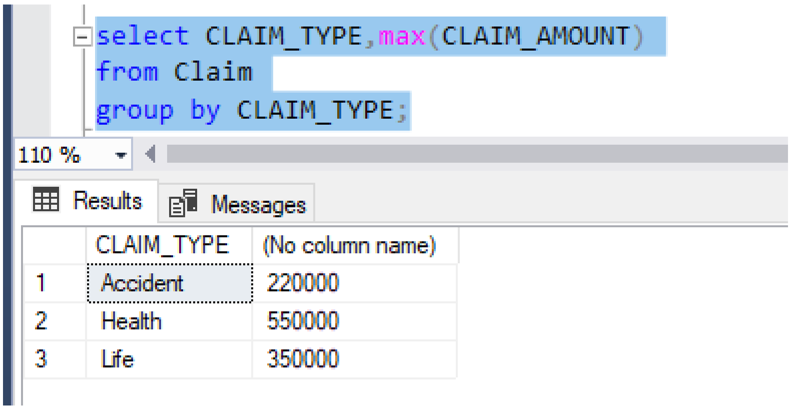
```
--Aggregate Data Using Group By
select POLICY_TYPE,count(CUSTOMER_ID) from Policy_T
group by POLICY_TYPE;
```

	POLICY_TYPE	(No column name)
1	Health Insurance	1
2	Life Insurance	2
3	Travel Insurance	1
4	Vehicle Insurance	1

QUERY:

```
select CLAIM_TYPE,max(CLAIM_AMOUNT)
from Claim
group by CLAIM_TYPE;
```

OUTPUT:



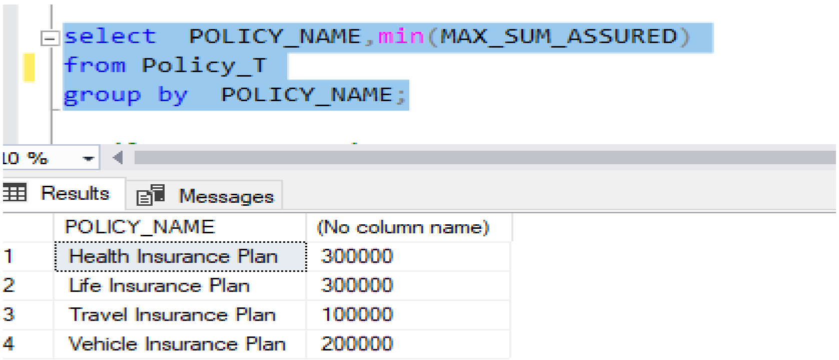
```
select CLAIM_TYPE,max(CLAIM_AMOUNT)
from Claim
group by CLAIM_TYPE;
```

	CLAIM_TYPE	(No column name)
1	Accident	220000
2	Health	550000
3	Life	350000

QUERY:

```
select  POLICY_NAME,min(MAX_SUM_ASSURED)
from Policy_T
group by  POLICY_NAME;
```

OUTPUT:



```
select   POLICY_NAME,min(MAX_SUM_ASSURED)
from Policy_T
group by   POLICY_NAME;
```

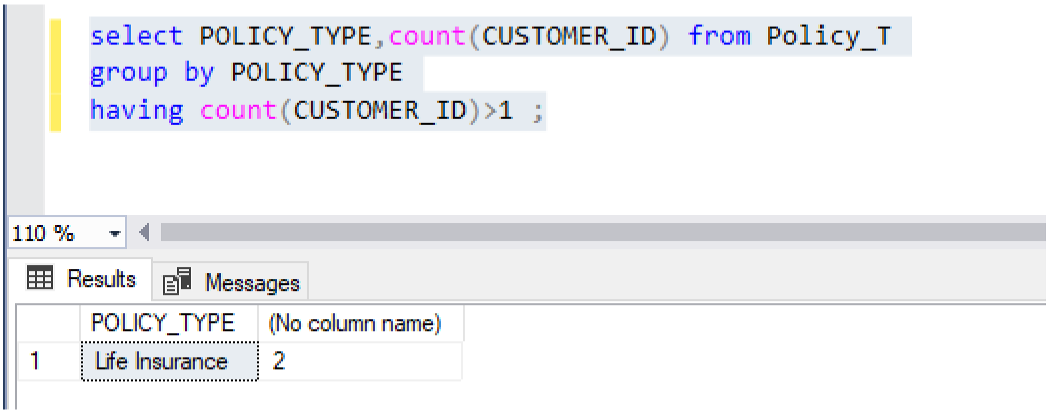
	POLICY_NAME	(No column name)
1	Health Insurance Plan	300000
2	Life Insurance Plan	300000
3	Travel Insurance Plan	100000
4	Vehicle Insurance Plan	200000

5. Filter Groups Using HAVING

Query Task: Retrieve the customer IDs and their total number of orders, but only for customers who have placed more than 5 orders.

QUERY:

```
SELECT POLICY_TYPE,count(CUSTOMER_ID) from Policy_T
group by POLICY_TYPE
having count(CUSTOMER_ID)>1 ;
```



```
select POLICY_TYPE,count(CUSTOMER_ID) from Policy_T
group by POLICY_TYPE
having count(CUSTOMER_ID)>1 ;
```

	POLICY_TYPE	(No column name)
1	Life Insurance	2

6. **Order Results Using ORDER BY**

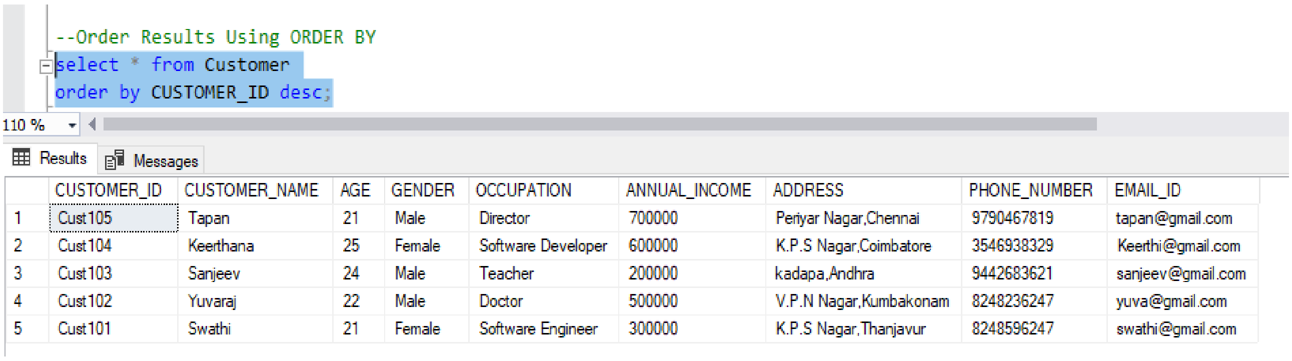
Query Task: Select all products from the products table and order them by price in descending order.

Hint: Use the ORDER BY clause to sort the results

QUERY:

```
select * from Customer
order by CUSTOMER_ID desc;
```

OUTPUT;



```
--Order Results Using ORDER BY
select * from Customer
order by CUSTOMER_ID desc;
```

	CUSTOMER_ID	CUSTOMER_NAME	AGE	GENDER	OCCUPATION	ANNUAL_INCOME	ADDRESS	PHONE_NUMBER	EMAIL_ID
1	Cust105	Tapan	21	Male	Director	700000	Periyar Nagar,Chennai	9790467819	tapan@gmail.com
2	Cust104	Keerthana	25	Female	Software Developer	600000	K.P.S Nagar,Coimbatore	3546938329	Keerthi@gmail.com
3	Cust103	Sanjeev	24	Male	Teacher	200000	kadapa,Andhra	9442683621	sanjeev@gmail.com
4	Cust102	Yuvaraj	22	Male	Doctor	500000	V.P.N Nagar,Kumbakonam	8248236247	yuva@gmail.com
5	Cust101	Swathi	21	Female	Software Engineer	300000	K.P.S Nagar,Thanjavur	8248596247	swathi@gmail.com

7. Retrieve Data with a Subquery

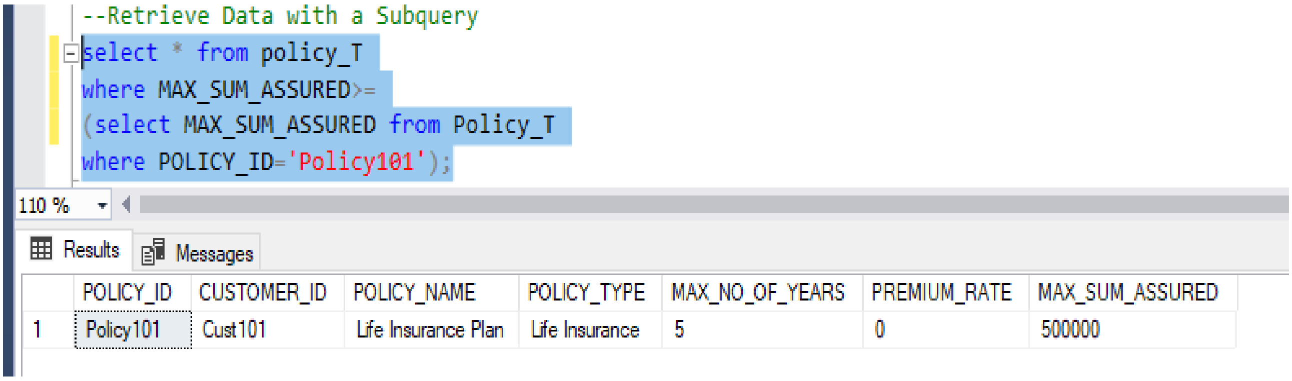
Query Task: Find the names of customers who have placed orders with a total amount greater than Rs.1000.

Hint: Use a subquery to calculate the total order amount for each customer

QUERY:

```
select * from policy_T
where MAX_SUM_ASSURED>=
(select MAX_SUM_ASSURED from Policy_T

where POLICY_ID='Policy101');
```

OUTPUT:



```
--Retrieve Data with a Subquery
select * from policy_T
where MAX_SUM_ASSURED>=
(select MAX_SUM_ASSURED from Policy_T
where POLICY_ID='Policy101');
```

	POLICY_ID	CUSTOMER_ID	POLICY_NAME	POLICY_TYPE	MAX_NO_OF_YEARS	PREMIUM_RATE	MAX_SUM_ASSURED
1	Policy101	Cust101	Life Insurance Plan	Life Insurance	5	0	500000

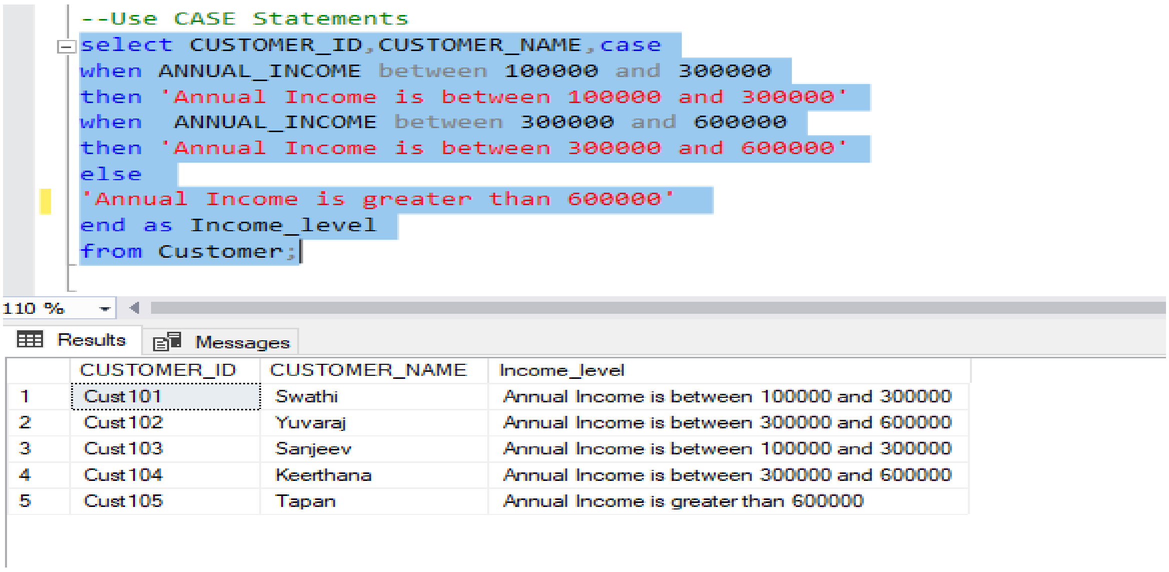
8. Use CASE Statements

Query Task: Retrieve order details along with a column that indicates if the order amount is 'High', 'Medium', or 'Low'.

QUERY:

```
select CUSTOMER_ID,CUSTOMER_NAME,case
when ANNUAL_INCOME between 100000 and 300000
then 'Annual Income is between 100000 and 300000'
when  ANNUAL_INCOME between 300000 and 600000
then 'Annual Income is between 300000 and 600000'
else
'Annual Income is greater than 600000'
end as Income_level
from Customer;
```

OUTPUT:



```
--Use CASE Statements
select CUSTOMER_ID,CUSTOMER_NAME,case
when ANNUAL_INCOME between 100000 and 300000
then 'Annual Income is between 100000 and 300000'
when  ANNUAL_INCOME between 300000 and 600000
then 'Annual Income is between 300000 and 600000'
else
'Annual Income is greater than 600000'
end as Income_level
from Customer;
```

	CUSTOMER_ID	CUSTOMER_NAME	Income_level
1	Cust101	Swathi	Annual Income is between 100000 and 300000
2	Cust102	Yuvaraj	Annual Income is between 300000 and 600000
3	Cust103	Sanjeev	Annual Income is between 100000 and 300000
4	Cust104	Keerthana	Annual Income is between 300000 and 600000
5	Cust105	Tapan	Annual Income is greater than 600000