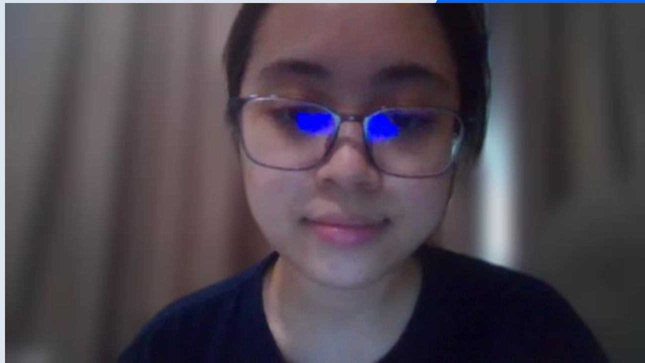
GROUP 30 SEG1201:
SUNNY INSURANCE DATABASE

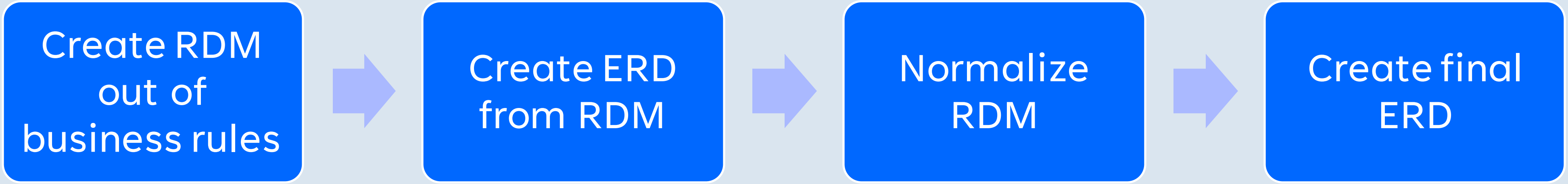
TAN KHAI NIE (22035208)

LESTER KOON ZHY MIN (20068813)

NURHANA SALEHA BINTI MOHAMMAD ZOFIAN (20024253)



SUNNY INSURANCE: Design process

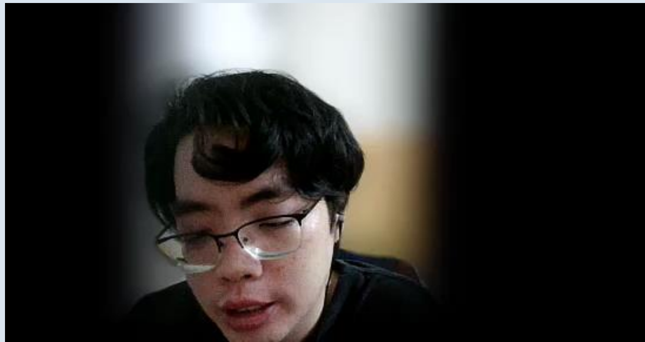


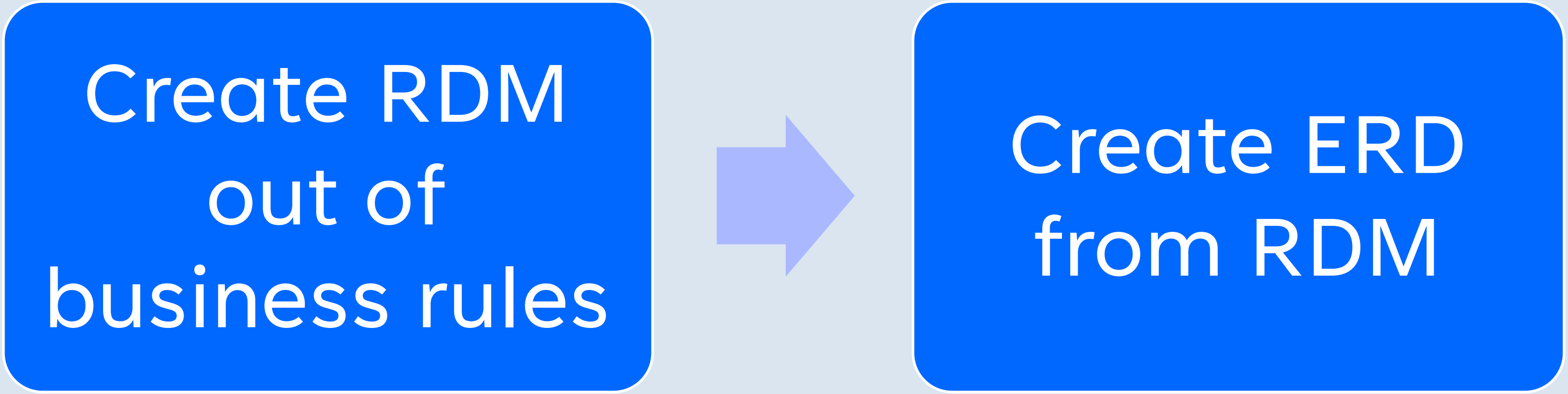
Create RDM out of business rules

Create ERD from RDM

Normalize RDM

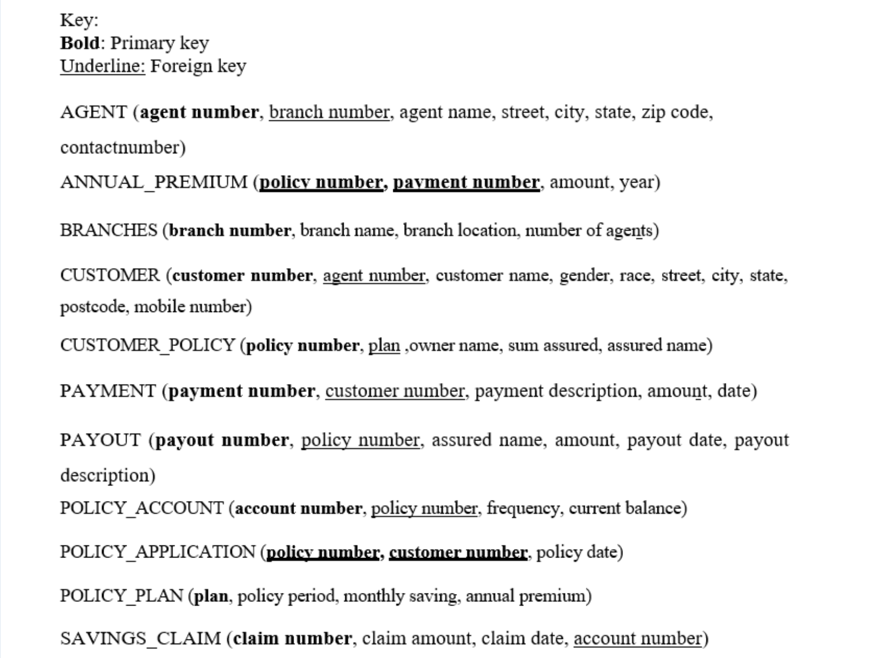
Create final ERD





Create RDM out of business rules

Create ERD from RDM



Key:
Bold: Primary key
Underline: Foreign key

AGENT (**agent number**, branch number, agent name, street, city, state, zip code, contactnumber)

ANNUAL_PREMIUM (**policy number, payment number**, amount, year)

BRANCHES (**branch number**, branch name, branch location, number of agents)

CUSTOMER (**customer number**, agent number, customer name, gender, race, street, city, state, postcode, mobile number)

CUSTOMER_POLICY (**policy number**, plan ,owner name, sum assured, assured name)

PAYMENT (**payment number**, customer number, payment description, amount, date)

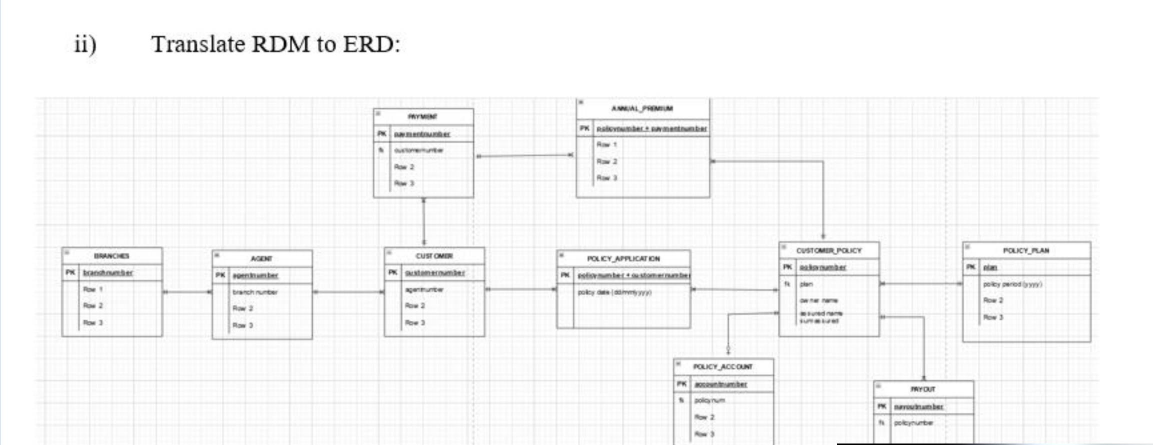
PAYOUT (**payout number**, policy number, assured name, amount, payout date, payout description)

POLICY_ACCOUNT (**account number**, policy number, frequency, current balance)

POLICY_APPLICATION (**policy number, customer number**, policy date)

POLICY_PLAN (**plan**, policy period, monthly saving, annual premium)

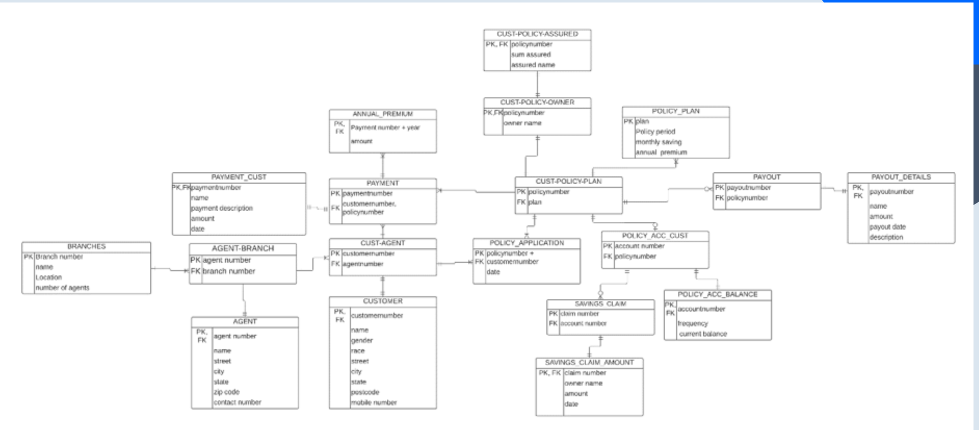
SAVINGS_CLAIM (**claim number**, claim amount, claim date, account number)



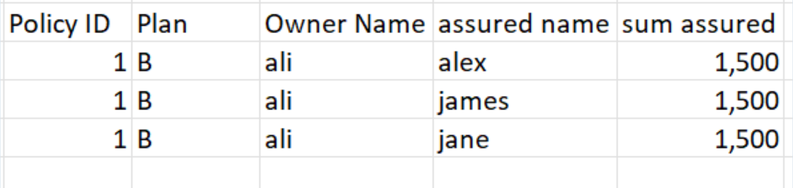
ii) Translate RDM to ERD:

Normalize RDM

Create final ERD

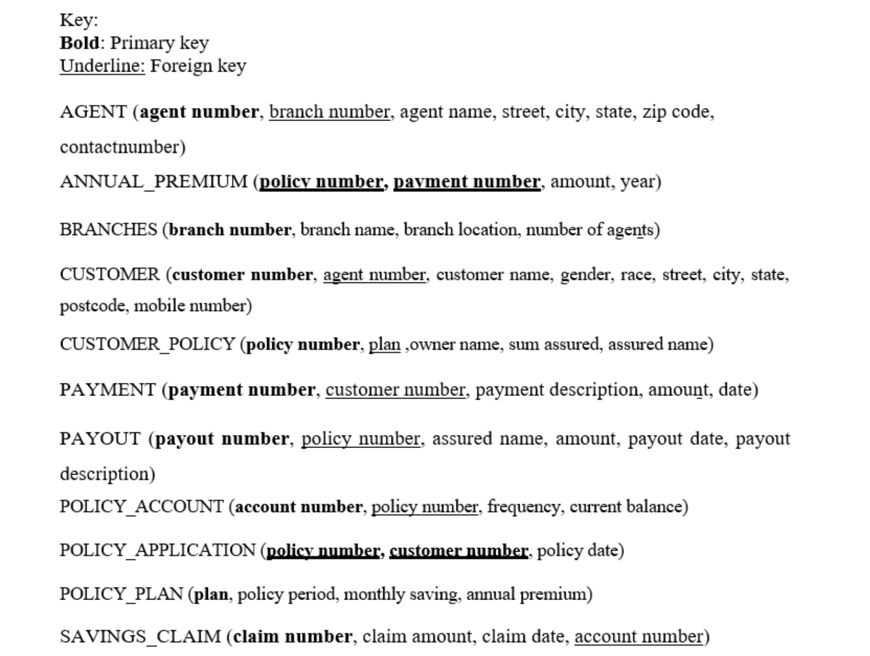


Anomalies:
Delete: PAYOUT
Update: customer_policy
Insert:agent



Policy ID	Plan	Owner Name	assured name	sum assured
1	B	ali	alex	1,500
1	B	ali	james	1,500
1	B	ali	jane	1,500

Example of customer_policy



Key:
Bold: Primary key
Underline: Foreign key

AGENT (**agent number**, branch number, agent name, street, city, state, zip code, contactnumber)

ANNUAL_PREMIUM (**policy number, payment number**, amount, year)

BRANCHES (**branch number**, branch name, branch location, number of agents)

CUSTOMER (**customer number**, agent number, customer name, gender, race, street, city, state, postcode, mobile number)

CUSTOMER_POLICY (**policy number**, plan ,owner name, sum assured, assured name)

PAYMENT (**payment number**, customer number, payment description, amount, date)

PAYOUT (**payout number**, policy number, assured name, amount, payout date, payout description)

POLICY_ACCOUNT (**account number**, policy number, frequency, current balance)

POLICY_APPLICATION (**policy number, customer number**, policy date)

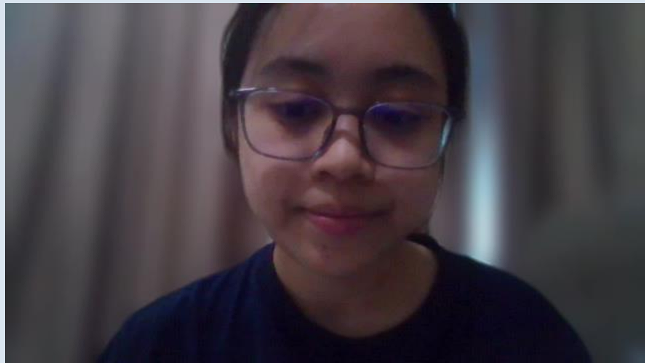
POLICY_PLAN (**plan**, policy period, monthly saving, annual premium)

SAVINGS_CLAIM (**claim number**, claim amount, claim date, account number)

SUNNY INSURANCE

Part 4: Obtaining user query

1. 1 sub query + 2 new date functions
2. 4 table join , 2 user conditions and a GROUP BY clause and HAVING subclause.
3. An outer join and 3 user conditions. One of the conditions uses the IN keyword.
4. a unary join, 4 table aliases and using date functions.

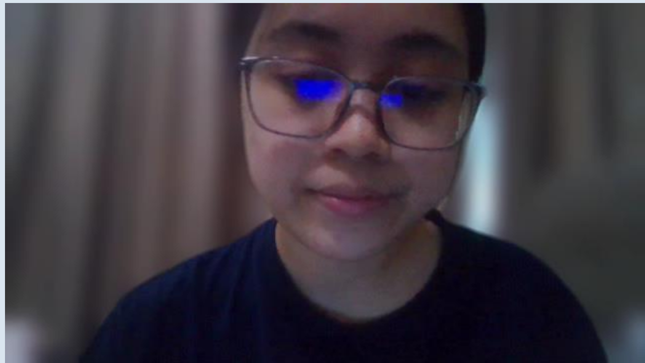


Query case:

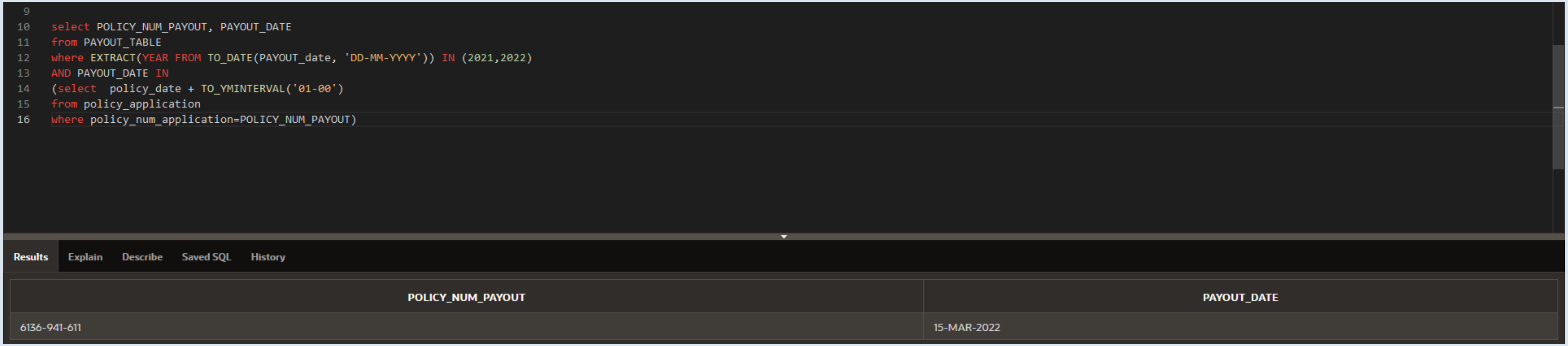
Sunny Insurance recently approved a new policy that states customers that paid their annual premium and has not claimed payout since application will be reduced of their next annual premium amount.

Therefore, the company needs to produce a report for customers that is eligible for the annual premium reduction to inform them.

To notify eligible customer on annual premium reduction, each SQL statement must produce some results.



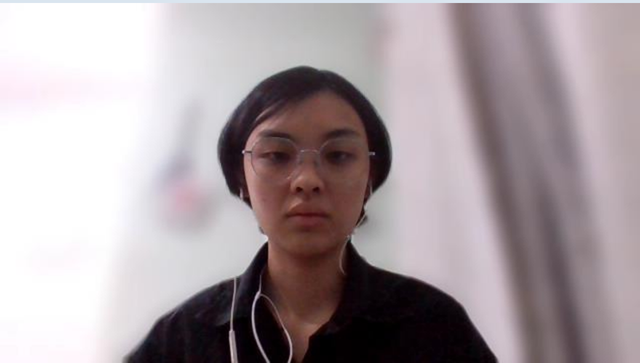
1. Write a user query with 1 sub-query and 2 new date functions.

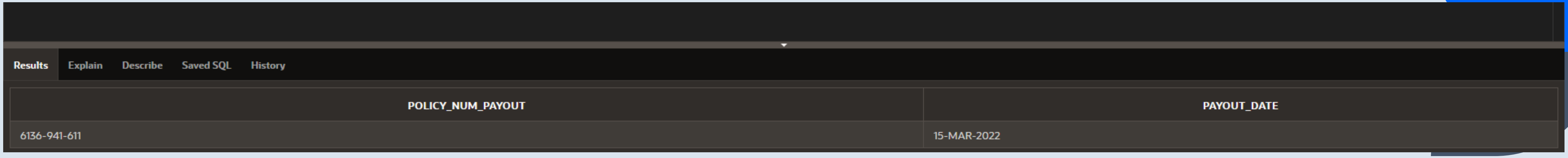


```
select POLICY_NUM_PAYOUT, PAYOUT_DATE
from PAYOUT_TABLE
where EXTRACT(YEAR FROM TO_DATE(PAYOUT_date, 'DD-MM-YYYY')) IN (2021,2022)
AND PAYOUT_DATE IN
(select  policy_date + TO_YMINTERVAL('01-00')
from policy_application
where policy_num_application=POLICY_NUM_PAYOUT)
```

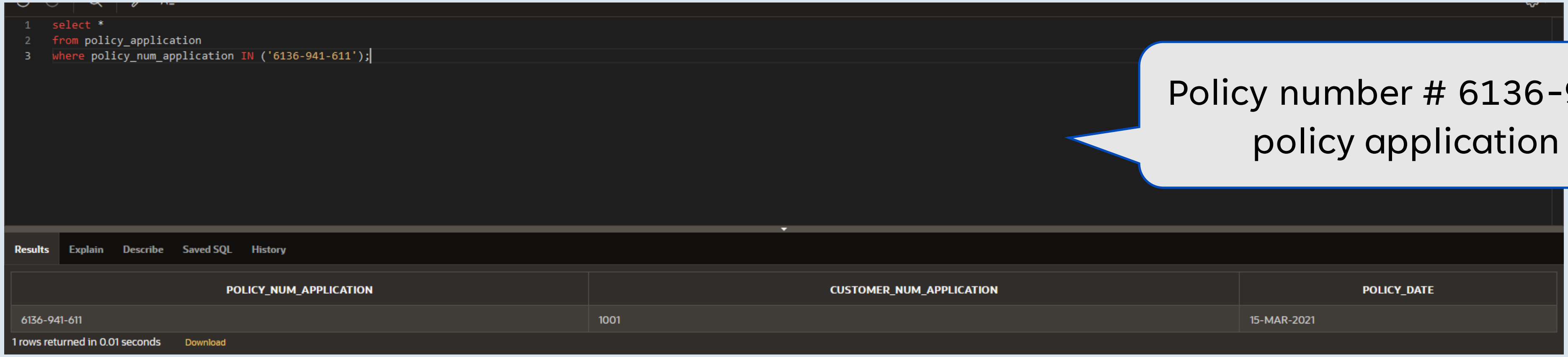
POLICY_NUM_PAYOUT	PAYOUT_DATE
6136-941-611	15-MAR-2022

This query yields results of clients that took pay out at the end of their annual premium date which is one year after policy application date. However, this result does not verify if they have paid annual premium.





POLICY_NUM_PAYOUT	PAYOUT_DATE
6136-941-611	15-MAR-2022

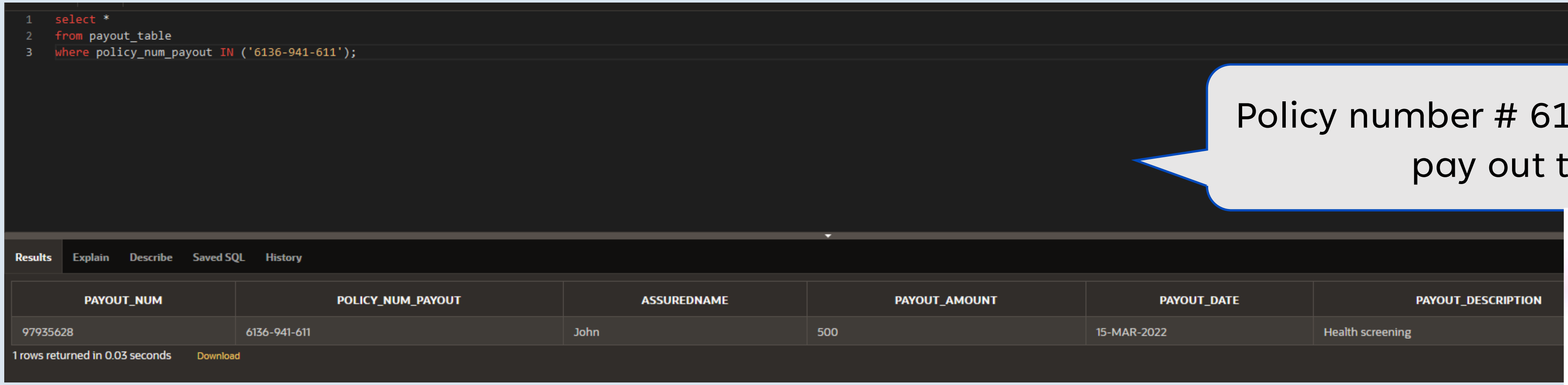


```
select *
from policy_application
where policy_num_application IN ('6136-941-611');
```

POLICY_NUM_APPLICATION	CUSTOMER_NUM_APPLICATION	POLICY_DATE
6136-941-611	1001	15-MAR-2021

```
1 rows returned in 0.01 seconds
```

Policy number # 6136-941-611 in policy application table

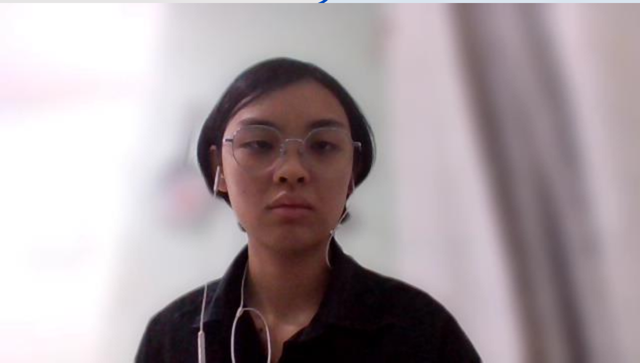


```
select *
from payout_table
where policy_num_payout IN ('6136-941-611');
```

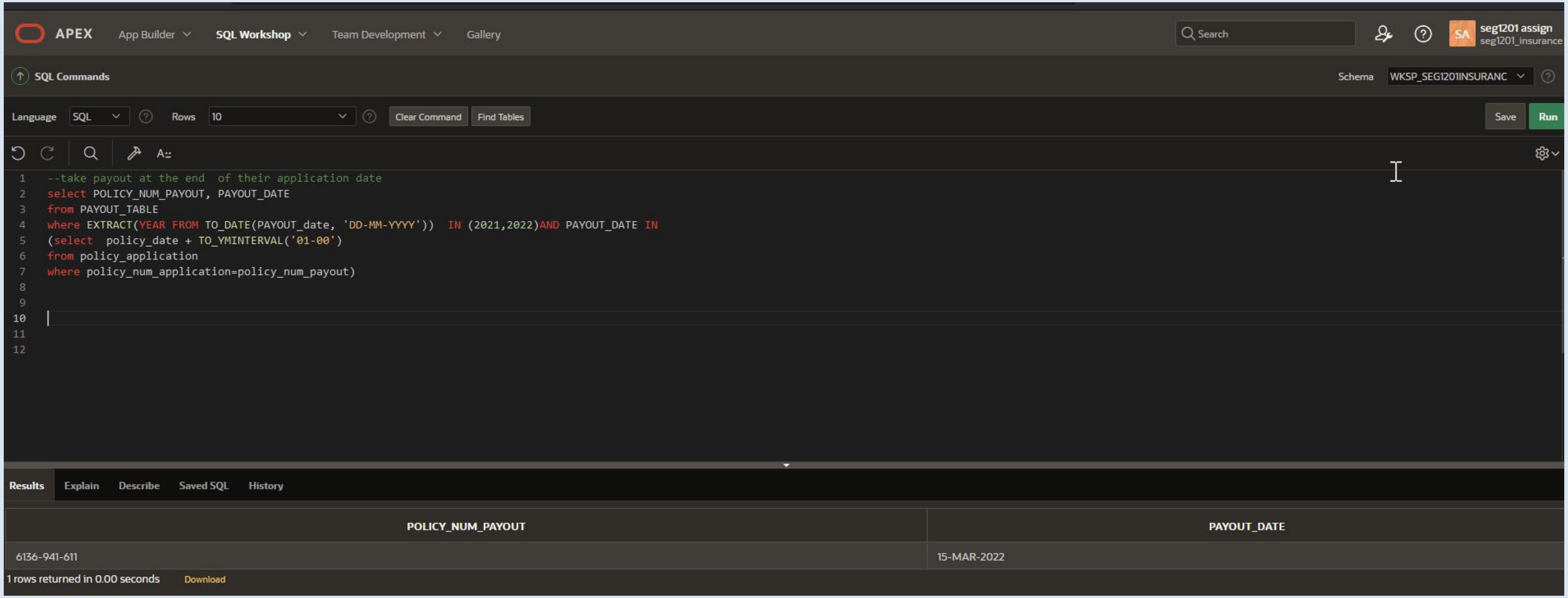
PAYOUT_NUM	POLICY_NUM_PAYOUT	ASSUREDNAME	PAYOUT_AMOUNT	PAYOUT_DATE	PAYOUT_DESCRIPTION
97935628	6136-941-611	John	500	15-MAR-2022	Health screening

```
1 rows returned in 0.03 seconds
```

Policy number # 6136-941-611 in pay out table



Part 1



Schema: WKSP_SEG1201INSURANC

Language: SQL

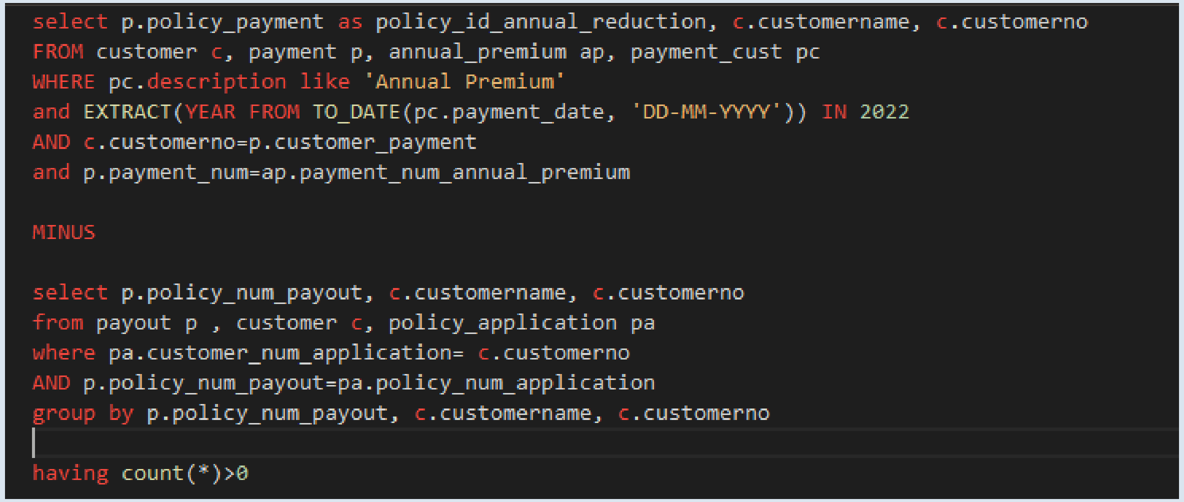
Rows: 10

```sql
--take payout at the end  of their application date
select POLICY_NUM_PAYOUT, PAYOUT_DATE
from PAYOUT_TABLE
where EXTRACT(YEAR FROM TO_DATE(PAYOUT_date, 'DD-MM-YYYY'))  IN (2021,2022)AND PAYOUT_DATE IN
(select  policy_date + TO_YMINTERVAL('01-00')
from policy_application
where policy_num_application=policy_num_payout)
```

POLICY_NUM_PAYOUT	PAYOUT_DATE
6136-941-611	15-MAR-2022

1 rows returned in 0.00 seconds

2. Write a user query with a 4-table join, 2 user conditions and a GROUP BY clause and HAVING subclause.

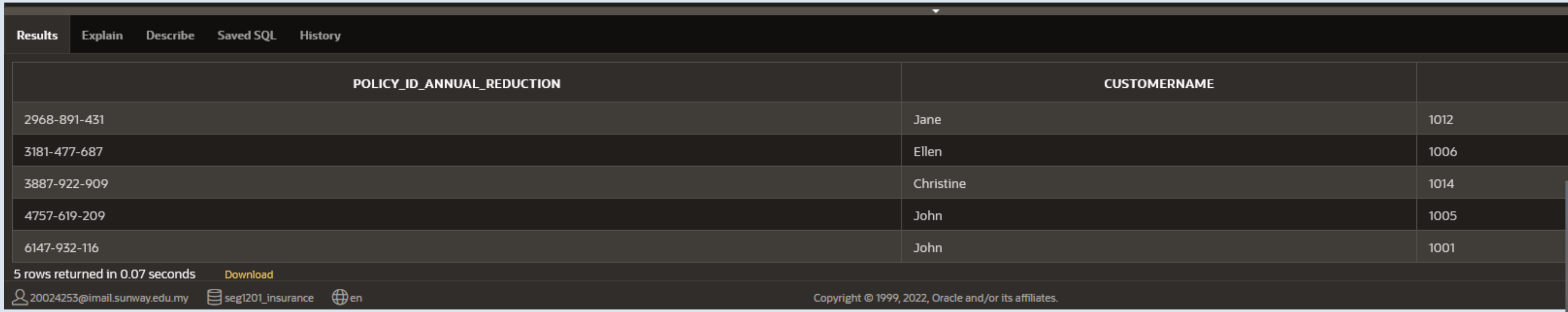


```
select p.policy_payment as policy_id_annual_reduction, c.customername, c.customerno
FROM customer c, payment p, annual_premium ap, payment_cust pc
WHERE pc.description like 'Annual Premium'
and EXTRACT(YEAR FROM TO_DATE(pc.payment_date, 'DD-MM-YYYY')) IN 2022
AND c.customerno=p.customer_payment
and p.payment_num=ap.payment_num_annual_premium

MINUS

select p.policy_num_payout, c.customername, c.customerno
from payout p , customer c, policy_application pa
where pa.customer_num_application= c.customerno
AND p.policy_num_payout=pa.policy_num_application
group by p.policy_num_payout, c.customername, c.customerno

having count(*)>0
```



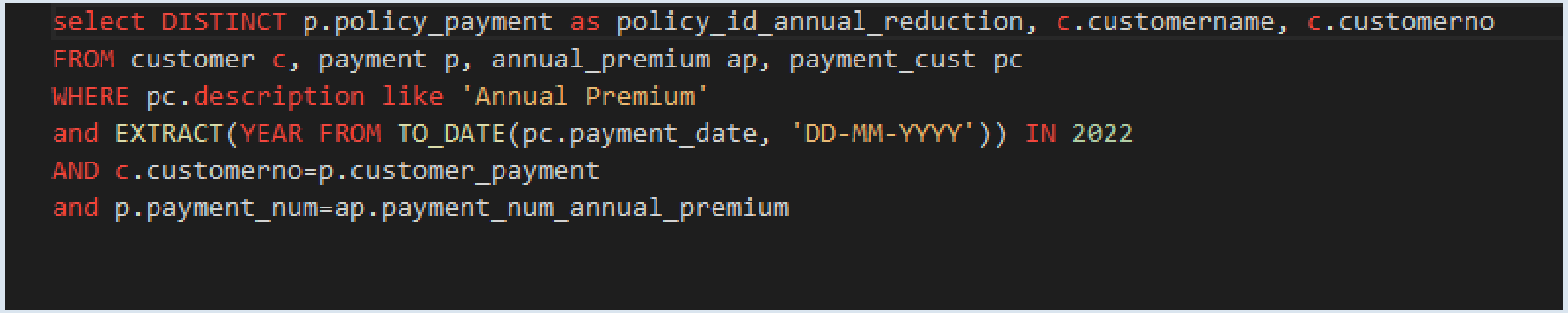
POLICY_ID_ANNUAL_REDUCTION	CUSTOMERNAME	CUSTOMERNO
2968-891-431	Jane	1012
3181-477-687	Ellen	1006
3887-922-909	Christine	1014
4757-619-209	John	1005
6147-932-116	John	1001

5 rows returned in 0.07 seconds Download

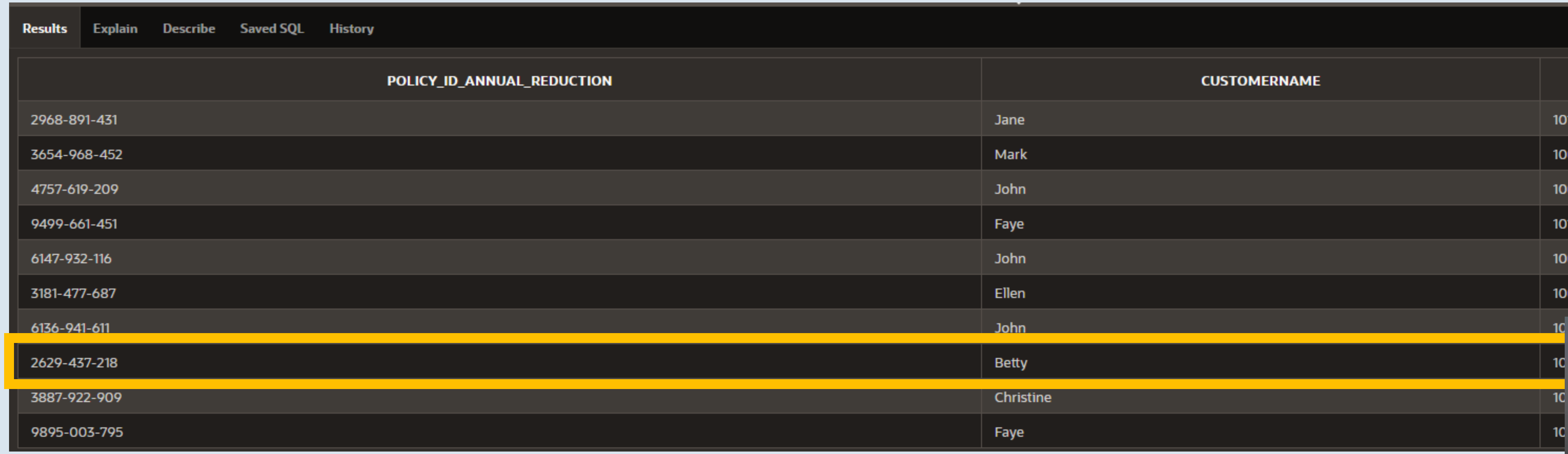
20024253@imail.sunway.edu.my seg1201_insurance en

Copyright © 1999, 2022, Oracle and/or its affiliates.

2. Write a user query with a 4-table join, 2 user conditions (note: the join and user conditions are not the same) and a GROUP BY clause and HAVING subclause.



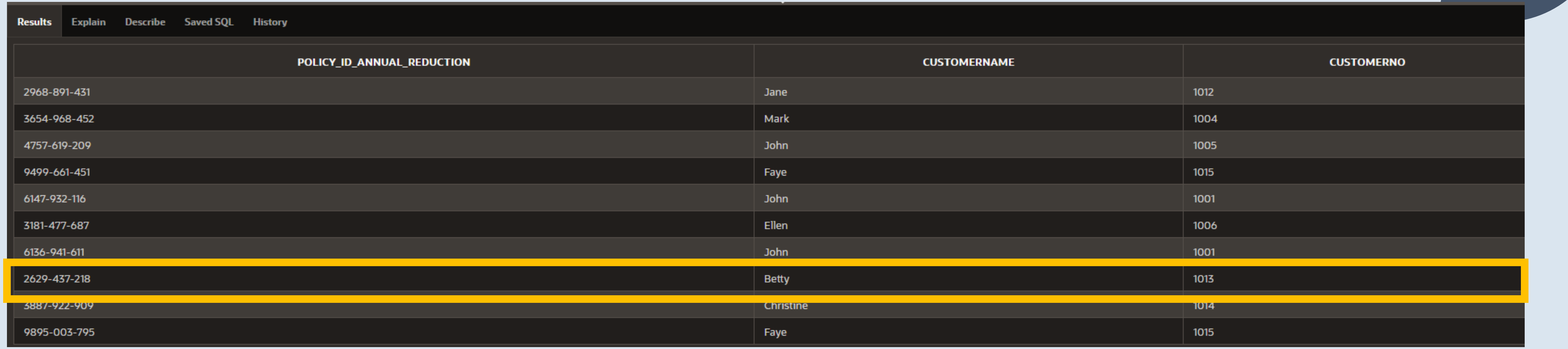
```
select DISTINCT p.policy_payment as policy_id_annual_reduction, c.customername, c.customerno
FROM customer c, payment p, annual_premium ap, payment_cust pc
WHERE pc.description like 'Annual Premium'
and EXTRACT(YEAR FROM TO_DATE(pc.payment_date, 'DD-MM-YYYY')) IN 2022
AND c.customerno=p.customer_payment
and p.payment_num=ap.payment_num_annual_premium
```



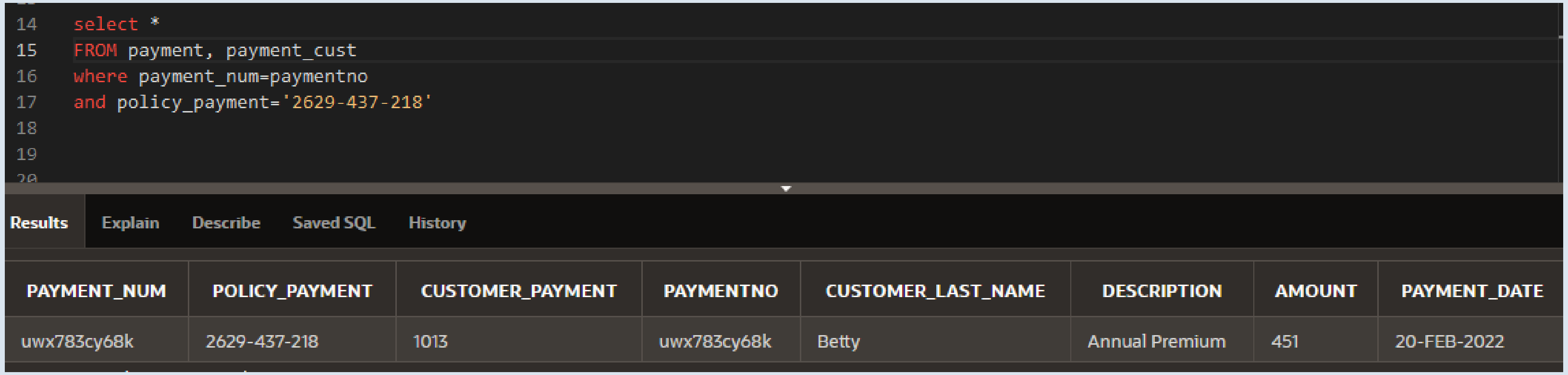
POLICY_ID_ANNUAL_REDUCTION	CUSTOMERNAME	CUSTOMERNO
2968-891-431	Jane	1012
3654-968-452	Mark	1004
4757-619-209	John	1005
9499-661-451	Faye	1015
6147-932-116	John	1001
3181-477-687	Ellen	1006
6136-941-611	John	10
2629-437-218	Betty	10
3887-922-909	Christine	10
9895-003-795	Faye	10

7/19/2023

2. Write a user query with a 4-table join, 2 user conditions (note: the join and user conditions are not the same) and a GROUP BY clause and HAVING subclause.

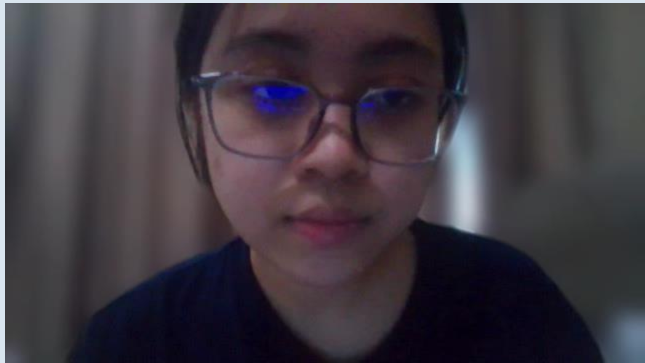


POLICY_ID_ANNUAL_REDUCTION	CUSTOMERNAME	CUSTOMERNO
2968-891-431	Jane	1012
3654-968-452	Mark	1004
4757-619-209	John	1005
9499-661-451	Faye	1015
6147-932-116	John	1001
3181-477-687	Ellen	1006
6136-941-611	John	1001
2629-437-218	Betty	1013
3887-922-909	Christine	1014
9895-003-795	Faye	1015

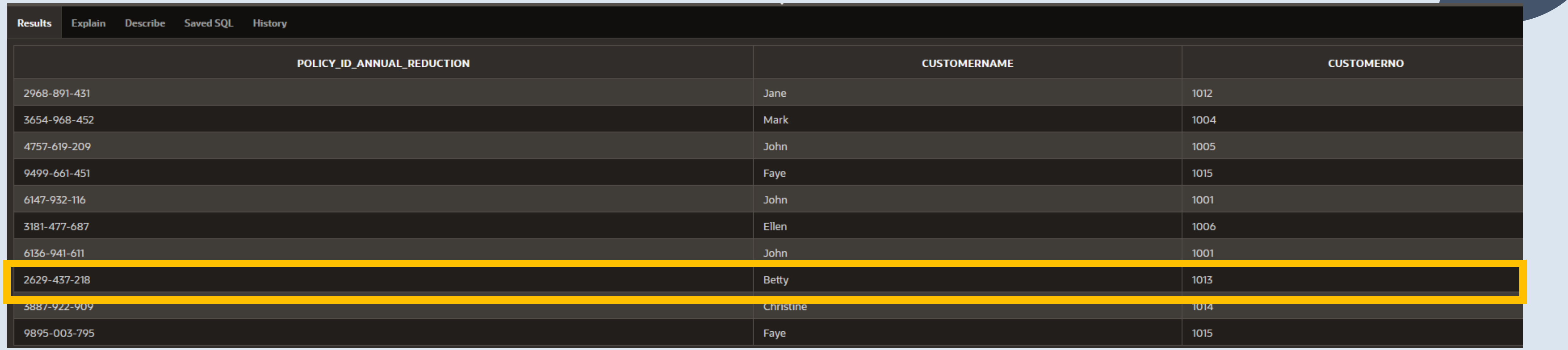


```sql
select *
FROM payment, payment_cust
where payment_num=paymentno
and policy_payment='2629-437-218'
```

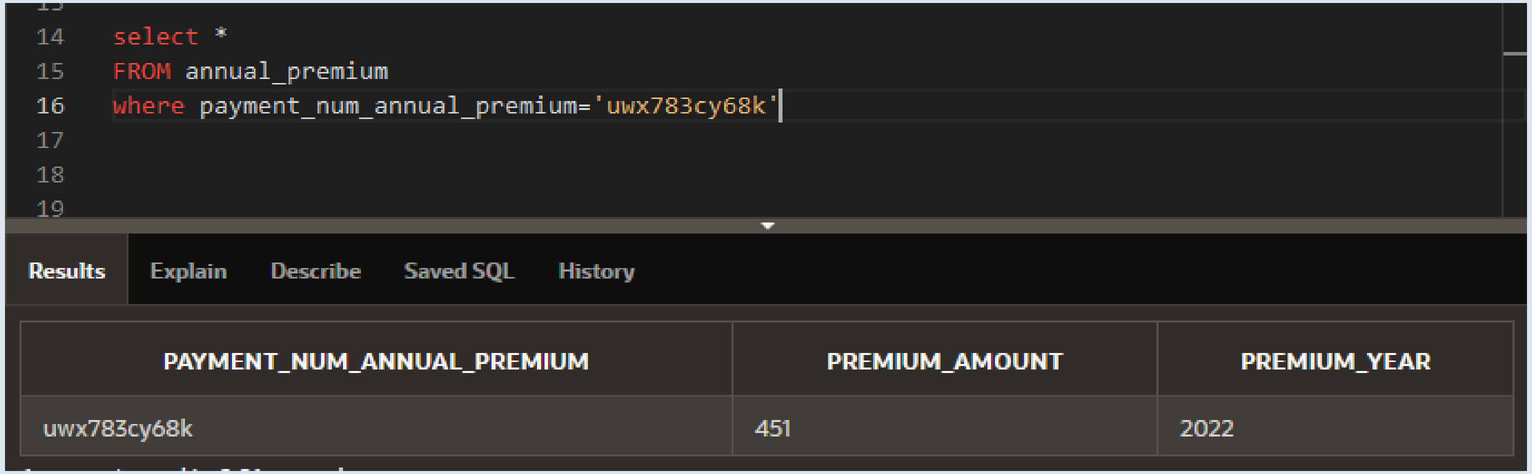
PAYMENT_NUM	POLICY_PAYMENT	CUSTOMER_PAYMENT	PAYMENTNO	CUSTOMER_LAST_NAME	DESCRIPTION	AMOUNT	PAYMENT_DATE
uwx783cy68k	2629-437-218	1013	uwx783cy68k	Betty	Annual Premium	451	20-FEB-2022



2. Write a user query with a 4-table join, 2 user conditions (note: the join and user conditions are not the same) and a GROUP BY clause and HAVING subclause.



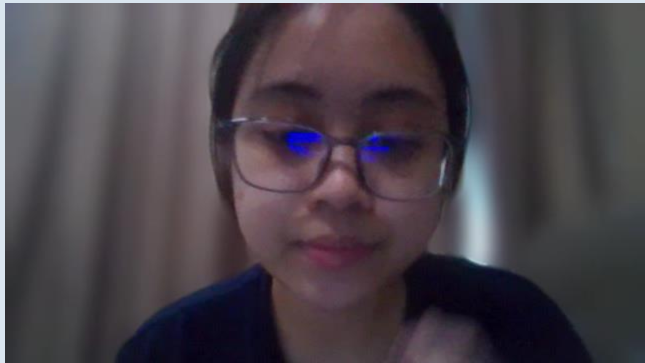
POLICY_ID_ANNUAL_REDUCTION	CUSTOMERNAME	CUSTOMERNO
2968-891-431	Jane	1012
3654-968-452	Mark	1004
4757-619-209	John	1005
9499-661-451	Faye	1015
6147-932-116	John	1001
3181-477-687	Ellen	1006
6136-941-611	John	1001
2629-437-218	Betty	1013
3887-922-909	Christine	1014
9895-003-795	Faye	1015



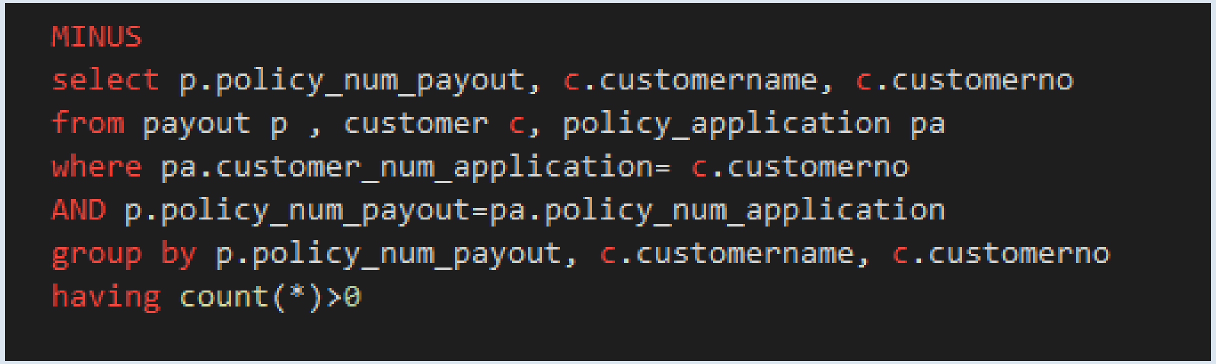
```
select *
FROM annual_premium
where payment_num_annual_premium='uwx783cy68k'
```

PAYMENT_NUM_ANNUAL_PREMIUM	PREMIUM_AMOUNT	PREMIUM_YEAR
uwx783cy68k	451	2022

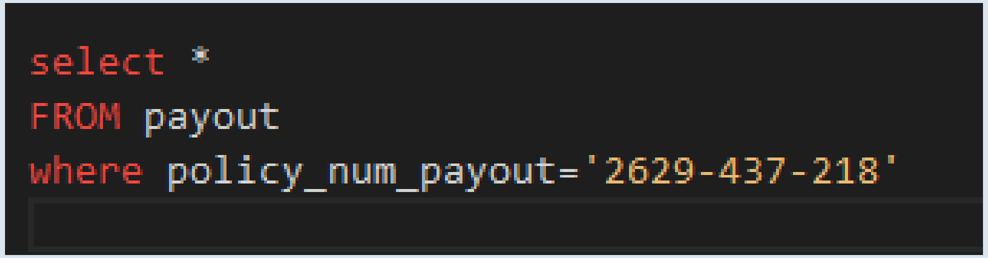
7/19/2023



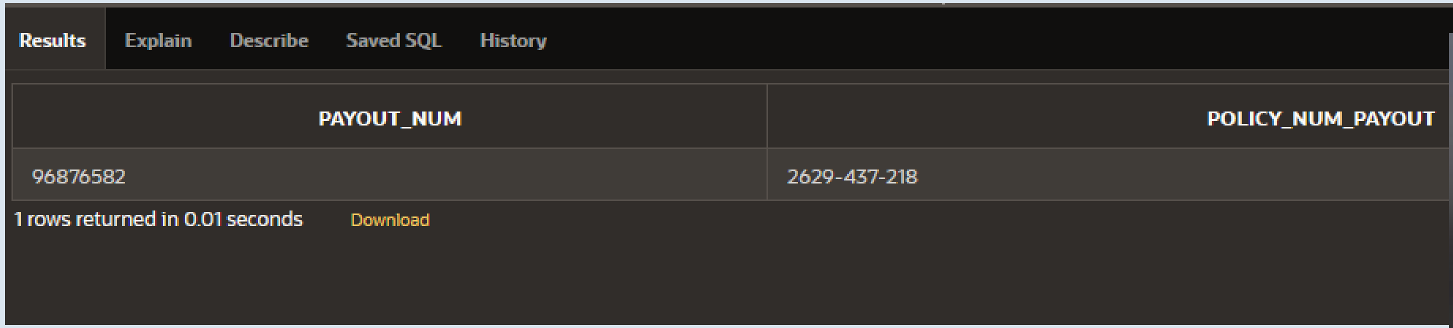
2. Write a user query with a 4-table join, 2 user conditions (note: the join and user conditions are not the same) and a GROUP BY clause and HAVING subclause.



```
MINUS
select p.policy_num_payout, c.customername, c.customerno
from payout p , customer c, policy_application pa
where pa.customer_num_application= c.customerno
AND p.policy_num_payout=pa.policy_num_application
group by p.policy_num_payout, c.customername, c.customerno
having count(*)>0
```

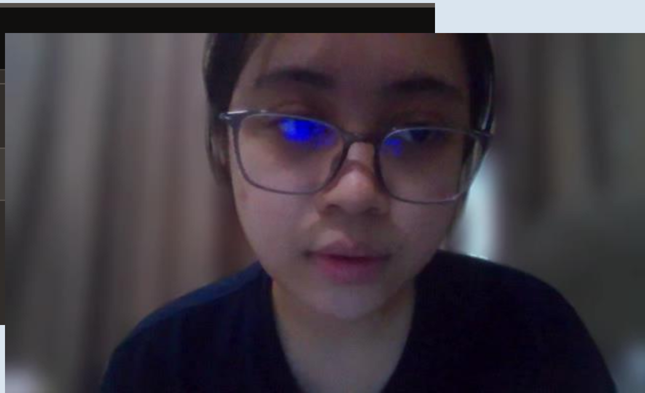


```
select *
FROM payout
where policy_num_payout='2629-437-218'
```

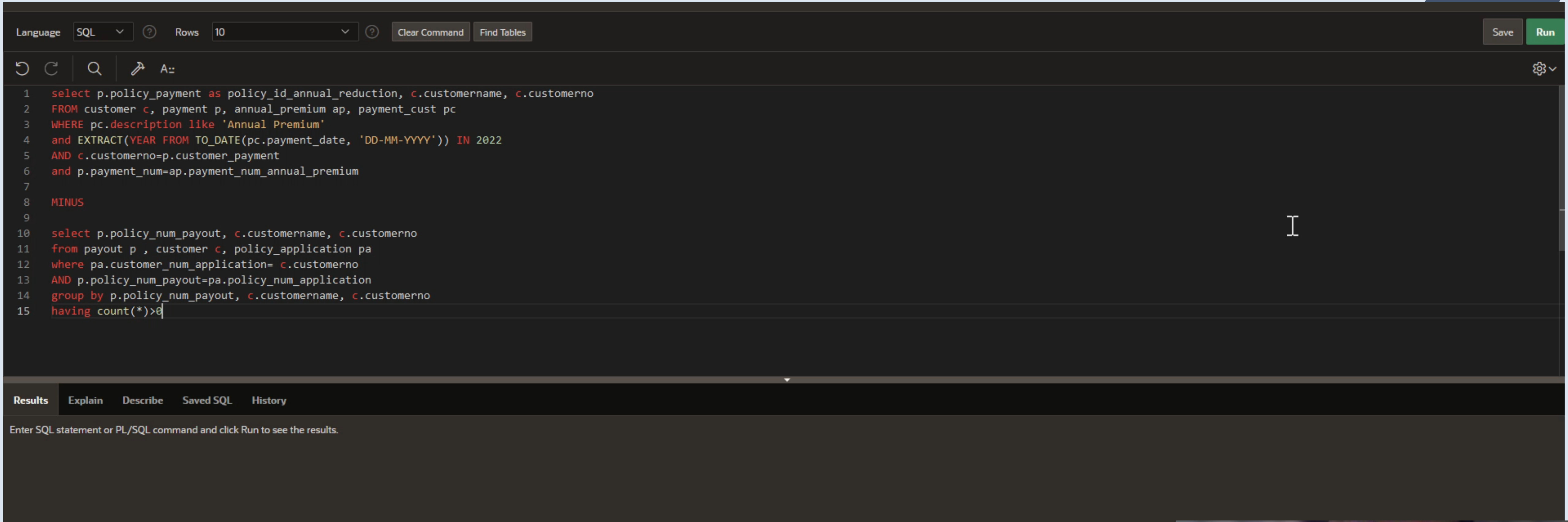


PAYOUT_NUM	POLICY_NUM_PAYOUT
96876582	2629-437-218

1 rows returned in 0.01 seconds



PART 2:



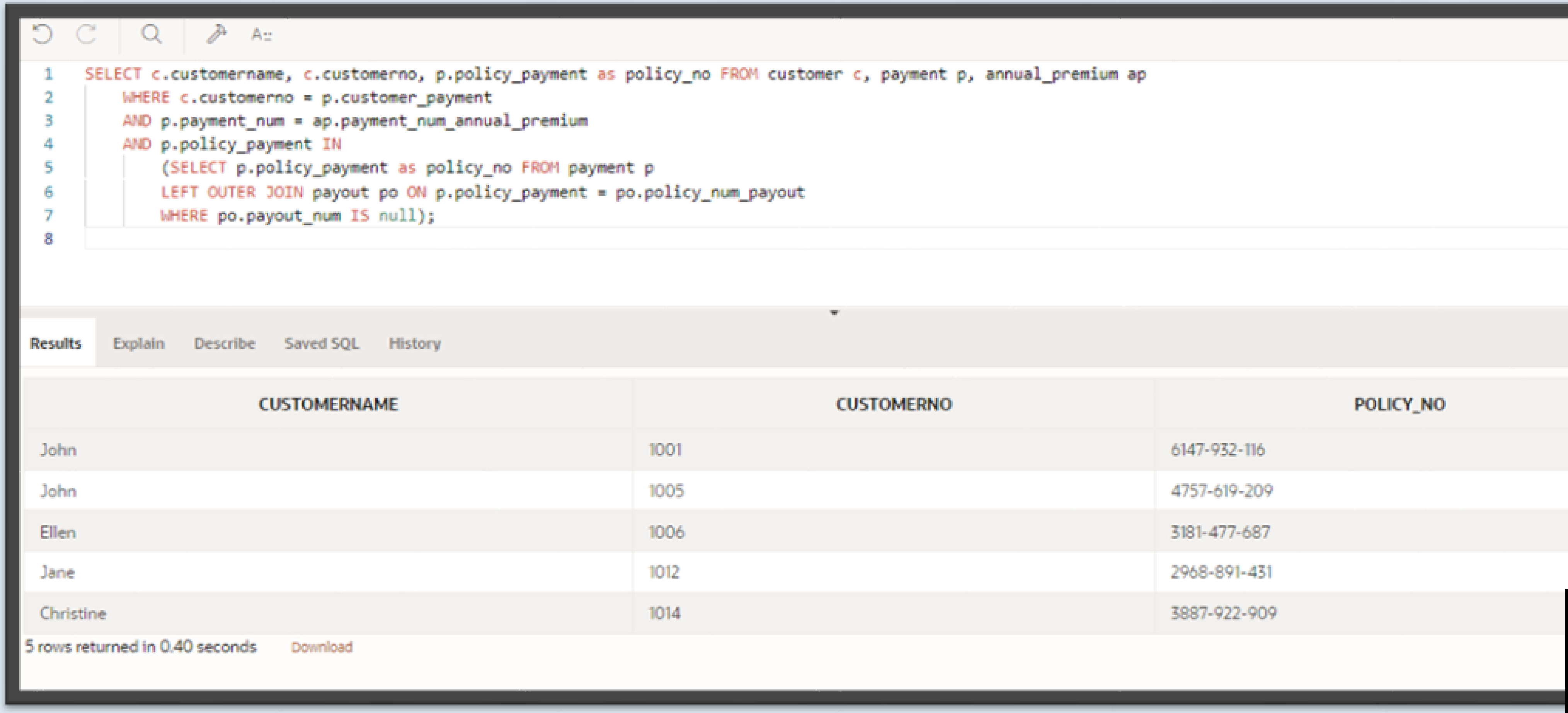
```
select p.policy_payment as policy_id_annual_reduction, c.customername, c.customerno
FROM customer c, payment p, annual_premium ap, payment_cust pc
WHERE pc.description like 'Annual Premium'
and EXTRACT(YEAR FROM TO_DATE(pc.payment_date, 'DD-MM-YYYY')) IN 2022
AND c.customerno=p.customer_payment
and p.payment_num=ap.payment_num_annual_premium

MINUS

select p.policy_num_payout, c.customername, c.customerno
from payout p , customer c, policy_application pa
where pa.customer_num_application= c.customerno
AND p.policy_num_payout=pa.policy_num_application
group by p.policy_num_payout, c.customername, c.customerno
having count(*)>0
```

Enter SQL statement or PL/SQL command and click Run to see the results.

3. Write a user query with an outer join and 3 user conditions to extract the names and customer numbers of customers who are eligible for the reduction. One of the conditions uses the IN keyword.

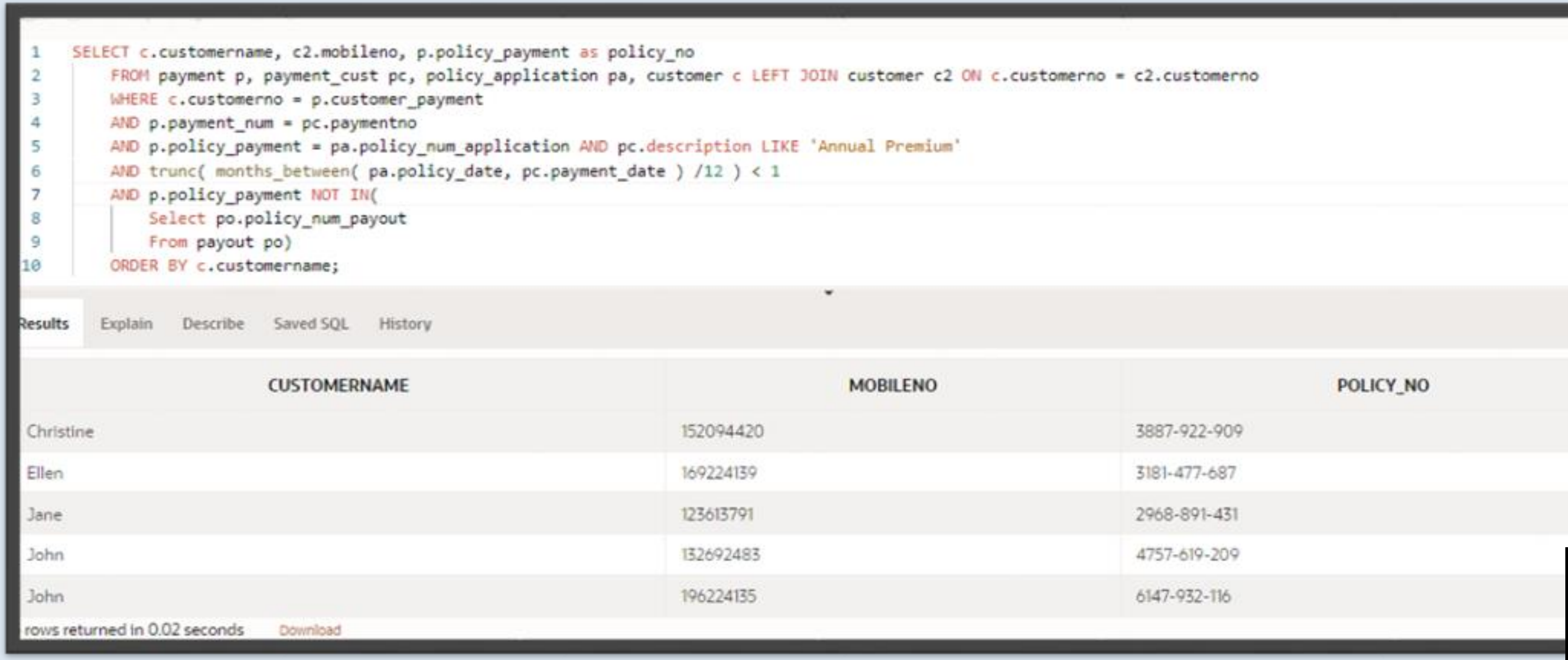


```sql
SELECT c.customername, c.customerno, p.policy_payment as policy_no FROM customer c, payment p, annual_premium ap
    WHERE c.customerno = p.customer_payment
    AND p.payment_num = ap.payment_num_annual_premium
    AND p.policy_payment IN
        (SELECT p.policy_payment as policy_no FROM payment p
        LEFT OUTER JOIN payout po ON p.policy_payment = po.policy_num_payout
        WHERE po.payout_num IS null);
```

CUSTOMERNAME	CUSTOMERNO	POLICY_NO
John	1001	6147-932-116
John	1005	4757-619-209
Ellen	1006	3181-477-687
Jane	1012	2968-891-431
Christine	1014	3887-922-909

5 rows returned in 0.40 seconds Download

4. Write a user query with a unary join, 4 table aliases and using date functions, to ensure that the customer has paid their annual premium on time. Retrieve the name, customer number and policy number for the eligible customers.



```
SELECT c.customername, c2.mobileno, p.policy_payment as policy_no
    FROM payment p, payment_cust pc, policy_application pa, customer c LEFT JOIN customer c2 ON c.customerno = c2.customerno
    WHERE c.customerno = p.customer_payment
    AND p.payment_num = pc.paymentno
    AND p.policy_payment = pa.policy_num_application AND pc.description LIKE 'Annual Premium'
    AND trunc( months_between( pa.policy_date, pc.payment_date ) /12 ) < 1
    AND p.policy_payment NOT IN(
        Select po.policy_num_payout
        From payout po)
    ORDER BY c.customername;
```

CUSTOMERNAME	MOBILENO	POLICY_NO
Christine	152094420	3887-922-909
Ellen	169224139	3181-477-687
Jane	123613791	2968-891-431
John	132692483	4757-619-209
John	196224135	6147-932-116

rows returned in 0.02 seconds

END

Thank you....