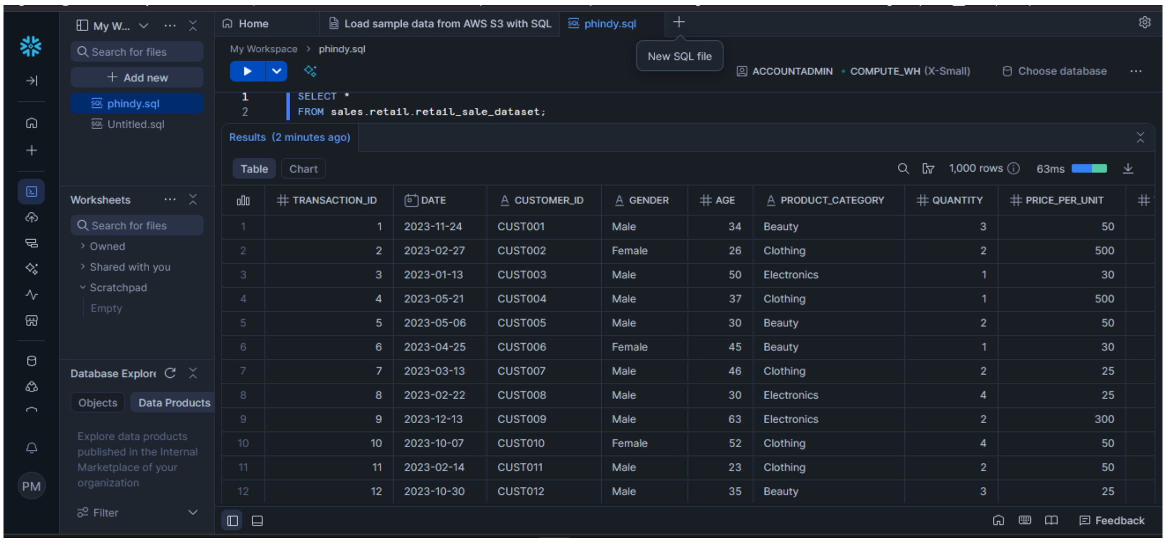
Phindile Mnisi

Practical 1: SQL Fundamentals (Snowflake-Basic SQL Syntax)

1. SELECT Statement

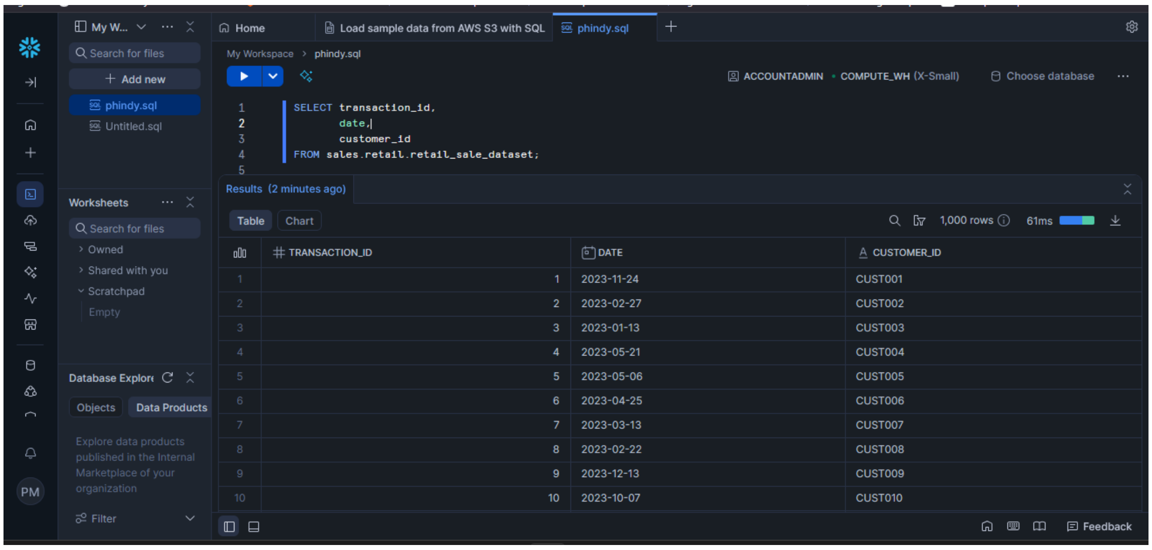


```sql
SELECT *
FROM sales.retail.retail_sale_dataset;
```

Results (2 minutes ago)

1,000 rows 63ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500
5	5	2023-05-06	CUST005	Male	30	Beauty	2	50
6	6	2023-04-25	CUST006	Female	45	Beauty	1	30
7	7	2023-03-13	CUST007	Male	46	Clothing	2	25
8	8	2023-02-22	CUST008	Male	30	Electronics	4	25
9	9	2023-12-13	CUST009	Male	63	Electronics	2	300
10	10	2023-10-07	CUST010	Female	52	Clothing	4	50
11	11	2023-02-14	CUST011	Male	23	Clothing	2	50
12	12	2023-10-30	CUST012	Male	35	Beauty	3	25



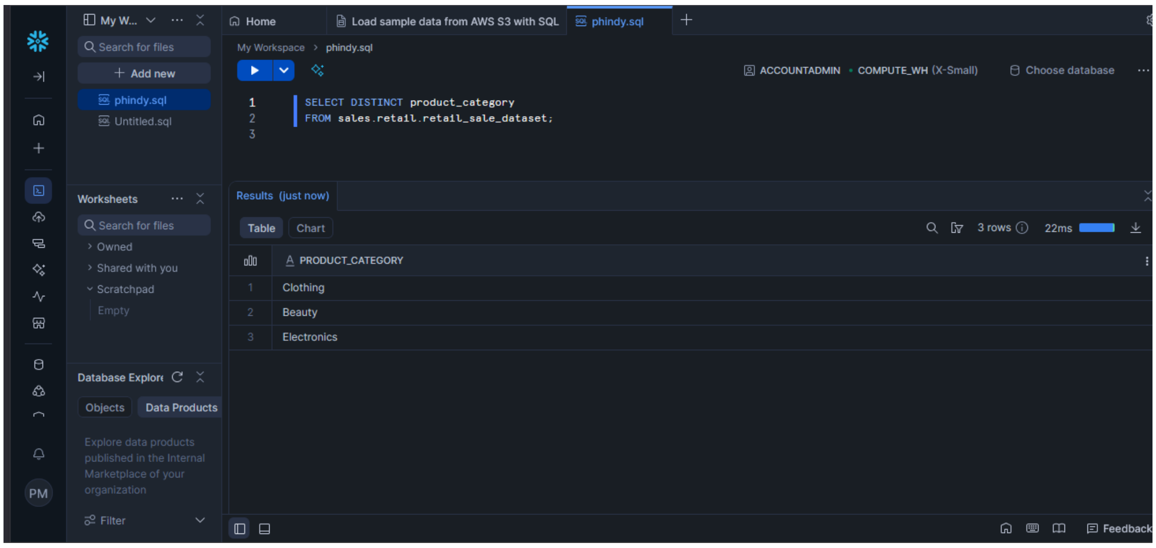
```sql
SELECT transaction_id,
       date,
       customer_id
FROM sales.retail.retail_sale_dataset;
```

Results (2 minutes ago)

1,000 rows 61ms

	TRANSACTION_ID	DATE	CUSTOMER_ID
1	1	2023-11-24	CUST001
2	2	2023-02-27	CUST002
3	3	2023-01-13	CUST003
4	4	2023-05-21	CUST004
5	5	2023-05-06	CUST005
6	6	2023-04-25	CUST006
7	7	2023-03-13	CUST007
8	8	2023-02-22	CUST008
9	9	2023-12-13	CUST009
10	10	2023-10-07	CUST010

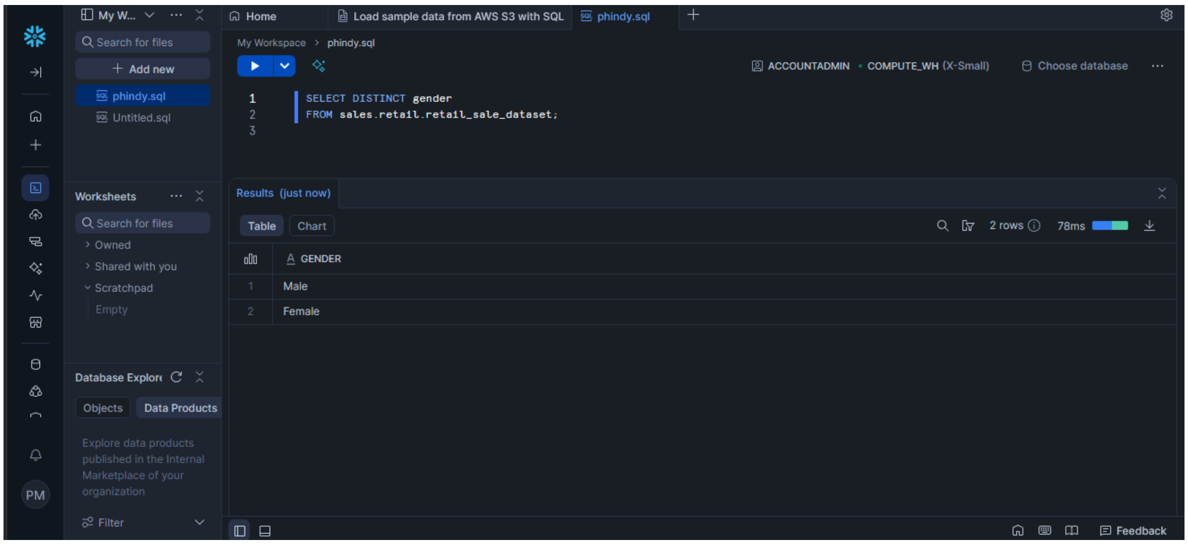
2. SELECT DISTINCT Statement



```sql
SELECT DISTINCT product_category
FROM sales.retail.retail_sale_dataset;
```

Results (just now) — 3 rows, 22ms

	PRODUCT_CATEGORY
1	Clothing
2	Beauty
3	Electronics

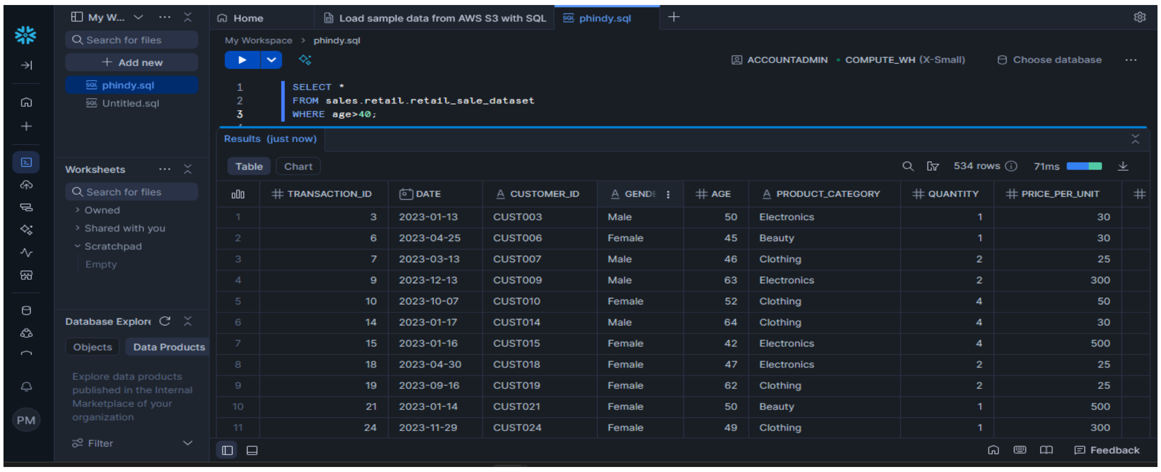


```sql
SELECT DISTINCT gender
FROM sales.retail.retail_sale_dataset;
```

Results (just now) — 2 rows, 78ms

	GENDER
1	Male
2	Female

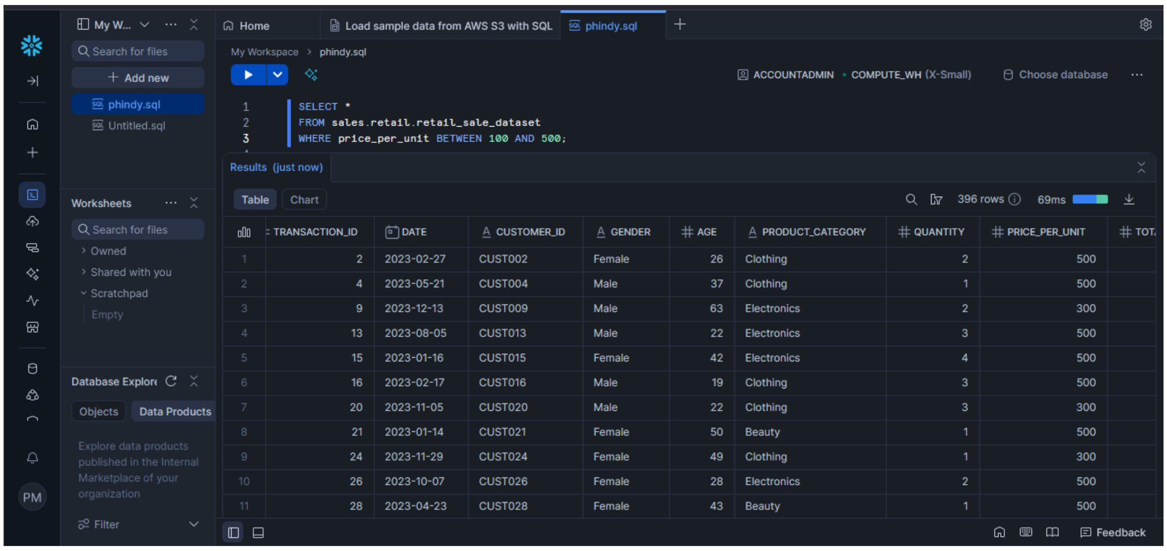
3. WHERE Clause



```sql
SELECT *
FROM sales.retail.retail_sale_dataset
WHERE age>40;
```

Results (just now) — 534 rows, 71ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDE	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
1	3	2023-01-13	CUST003	Male	50	Electronics	1	30
2	6	2023-04-25	CUST006	Female	45	Beauty	1	30
3	7	2023-03-13	CUST007	Male	46	Clothing	2	25
4	9	2023-12-13	CUST009	Male	63	Electronics	2	300
5	10	2023-10-07	CUST010	Female	52	Clothing	4	50
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500
8	18	2023-04-30	CUST018	Female	47	Electronics	2	25
9	19	2023-09-16	CUST019	Female	62	Clothing	2	25
10	21	2023-01-14	CUST021	Female	50	Beauty	1	500
11	24	2023-11-29	CUST024	Female	49	Clothing	1	300



Home | Load sample data from AWS S3 with SQL | phindy.sql

My Workspace > phindy.sql

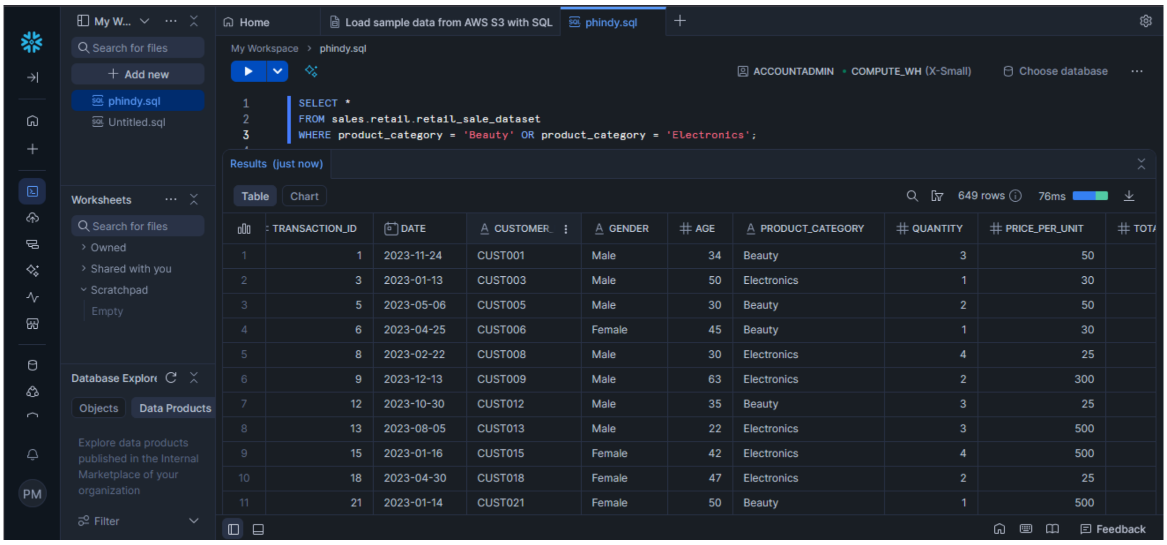
ACCOUNTADMIN • COMPUTE_WH (X-Small) | Choose database

```sql
SELECT *
FROM sales.retail.retail_sale_dataset
WHERE price_per_unit BETWEEN 100 AND 500;
```

Results (just now)

Table | Chart — 396 rows — 69ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOT
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	
2	4	2023-05-21	CUST004	Male	37	Clothing	1	500	
3	9	2023-12-13	CUST009	Male	63	Electronics	2	300	
4	13	2023-08-05	CUST013	Male	22	Electronics	3	500	
5	15	2023-01-16	CUST015	Female	42	Electronics	4	500	
6	16	2023-02-17	CUST016	Male	19	Clothing	3	500	
7	20	2023-11-05	CUST020	Male	22	Clothing	3	300	
8	21	2023-01-14	CUST021	Female	50	Beauty	1	500	
9	24	2023-11-29	CUST024	Female	49	Clothing	1	300	
10	26	2023-10-07	CUST026	Female	28	Electronics	2	500	
11	28	2023-04-23	CUST028	Female	43	Beauty	1	500	



Home | Load sample data from AWS S3 with SQL | phindy.sql

My Workspace > phindy.sql

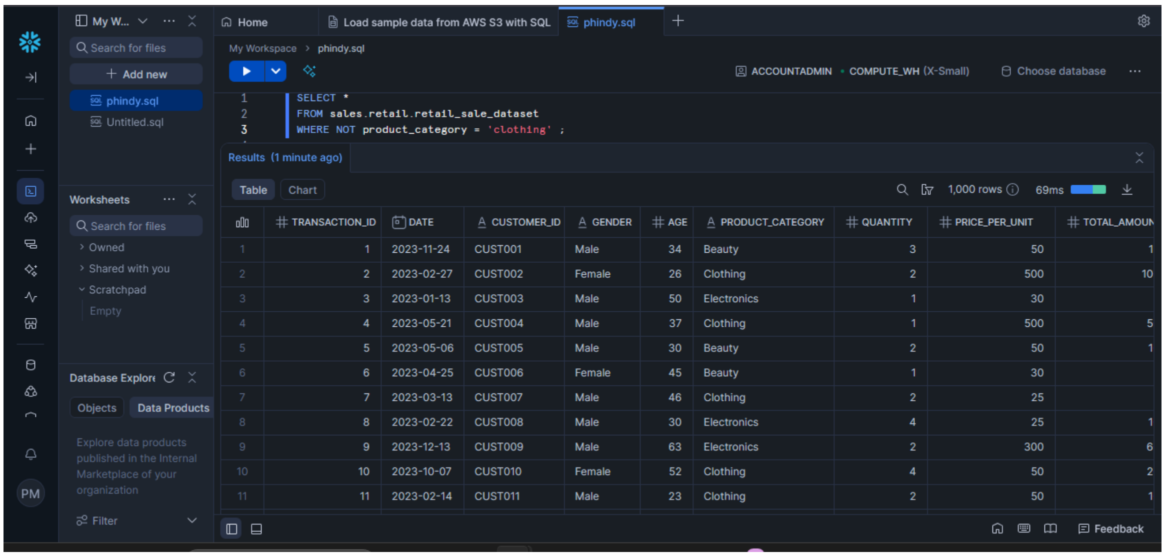
ACCOUNTADMIN • COMPUTE_WH (X-Small) | Choose database

```sql
SELECT *
FROM sales.retail.retail_sale_dataset
WHERE product_category = 'Beauty' OR product_category = 'Electronics';
```

Results (just now)

Table | Chart — 649 rows — 76ms

	TRANSACTION_ID	DATE	CUSTOMER_	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTA
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50	
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30	
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25	
6	9	2023-12-13	CUST009	Male	63	Electronics	2	300	
7	12	2023-10-30	CUST012	Male	35	Beauty	3	25	
8	13	2023-08-05	CUST013	Male	22	Electronics	3	500	
9	15	2023-01-16	CUST015	Female	42	Electronics	4	500	
10	18	2023-04-30	CUST018	Female	47	Electronics	2	25	
11	21	2023-01-14	CUST021	Female	50	Beauty	1	500	



Home | Load sample data from AWS S3 with SQL | phindy.sql

My Workspace > phindy.sql

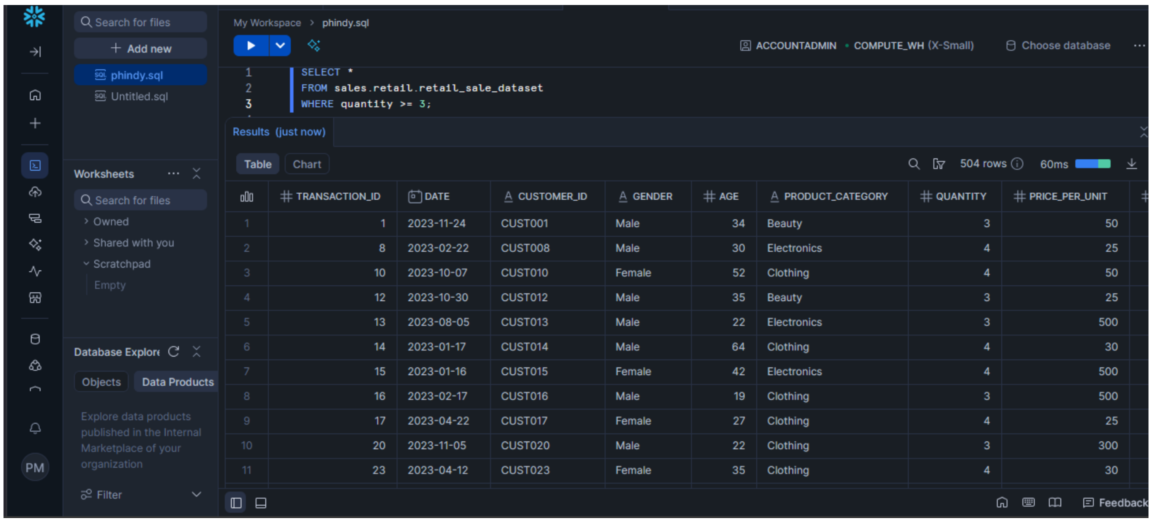
ACCOUNTADMIN • COMPUTE_WH (X-Small) | Choose database

```sql
SELECT *
FROM sales.retail.retail_sale_dataset
WHERE NOT product_category = 'clothing' ;
```

Results (1 minute ago)

Table | Chart — 1,000 rows — 69ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUN
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	1
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500	10
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30	
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500	5
5	5	2023-05-06	CUST005	Male	30	Beauty	2	50	1
6	6	2023-04-25	CUST006	Female	45	Beauty	1	30	
7	7	2023-03-13	CUST007	Male	46	Clothing	2	25	
8	8	2023-02-22	CUST008	Male	30	Electronics	4	25	1
9	9	2023-12-13	CUST009	Male	63	Electronics	2	300	6
10	10	2023-10-07	CUST010	Female	52	Clothing	4	50	2
11	11	2023-02-14	CUST011	Male	23	Clothing	2	50	1

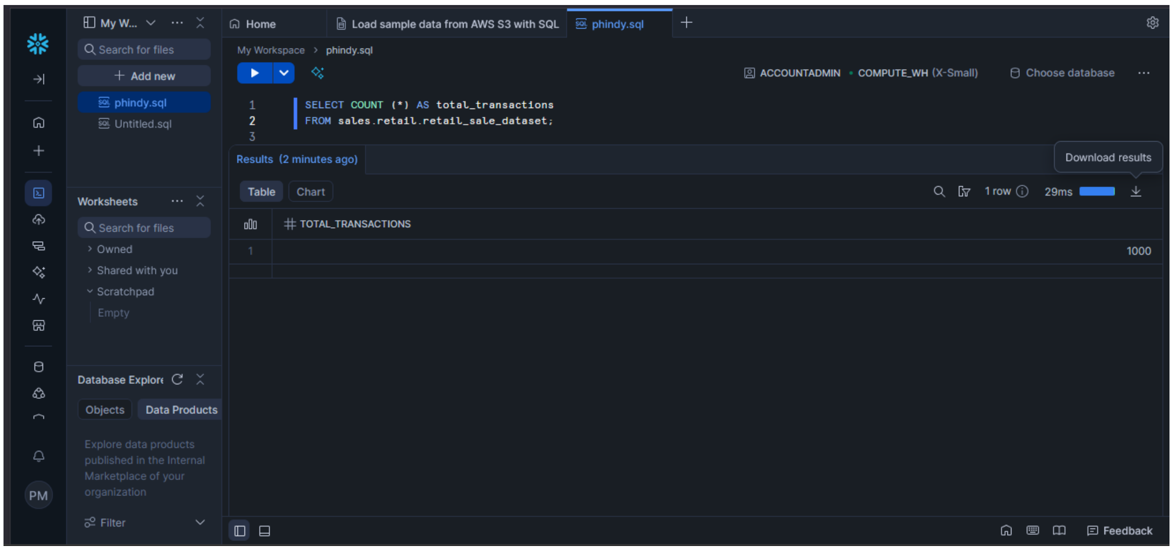


```sql
SELECT *
FROM sales.retail.retail_sale_dataset
WHERE quantity >= 3;
```

Results (just now) — 504 rows, 60ms

	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	8	2023-02-22	CUST008	Male	30	Electronics	4	25
3	10	2023-10-07	CUST010	Female	52	Clothing	4	50
4	12	2023-10-30	CUST012	Male	35	Beauty	3	25
5	13	2023-08-05	CUST013	Male	22	Electronics	3	500
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500
8	16	2023-02-17	CUST016	Male	19	Clothing	3	500
9	17	2023-04-22	CUST017	Female	27	Clothing	4	25
10	20	2023-11-05	CUST020	Male	22	Clothing	3	300
11	23	2023-04-12	CUST023	Female	35	Clothing	4	30

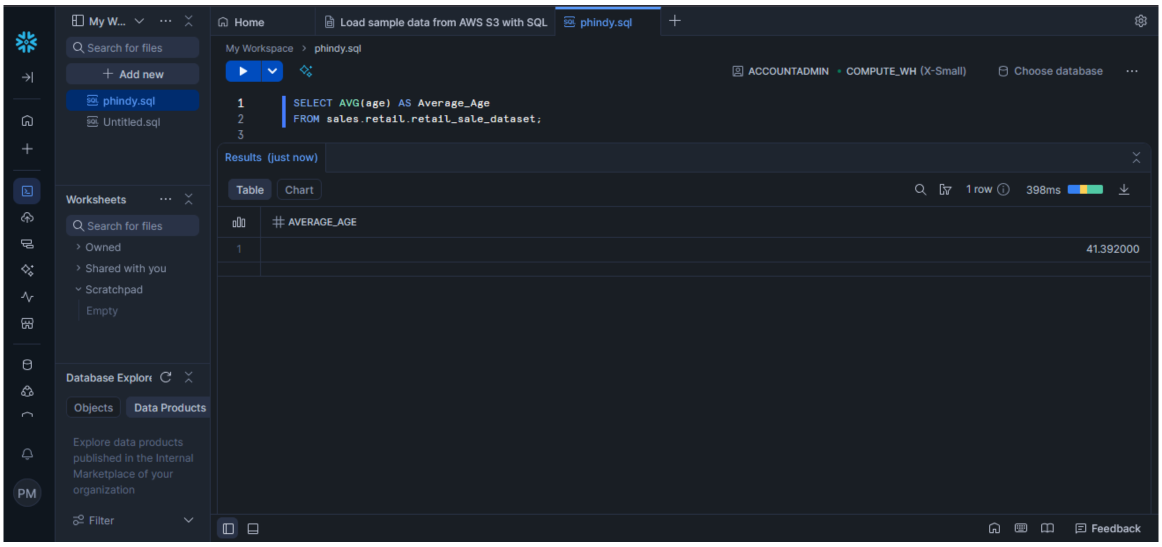
4. Aggregate Functions



```sql
SELECT COUNT (*) AS total_transactions
FROM sales.retail.retail_sale_dataset;
```

Results (2 minutes ago) — 1 row, 29ms

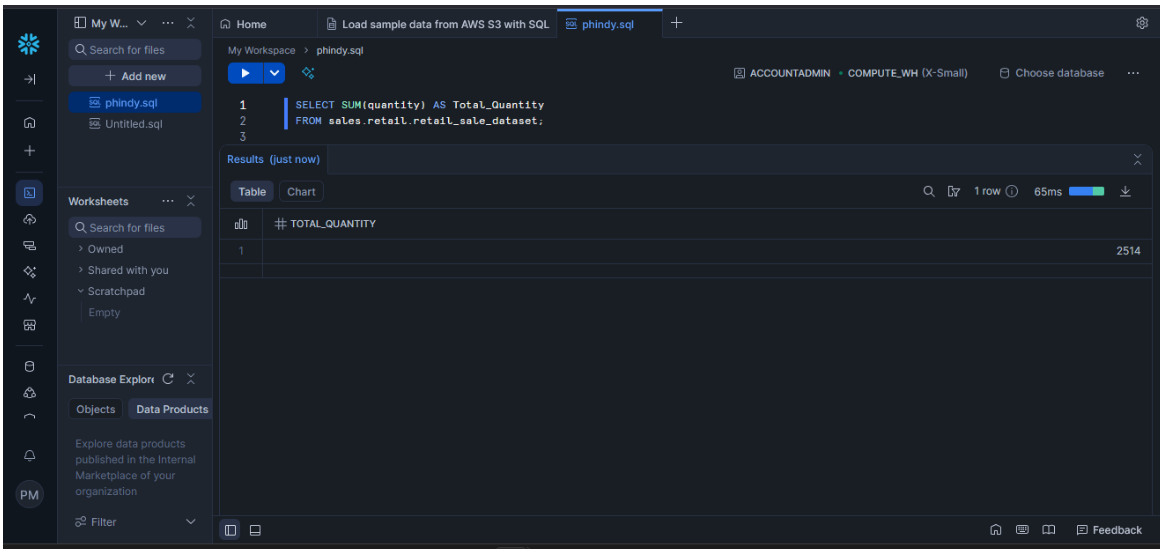
	TOTAL_TRANSACTIONS
1	1000



```sql
SELECT AVG(age) AS Average_Age
FROM sales.retail.retail_sale_dataset;
```

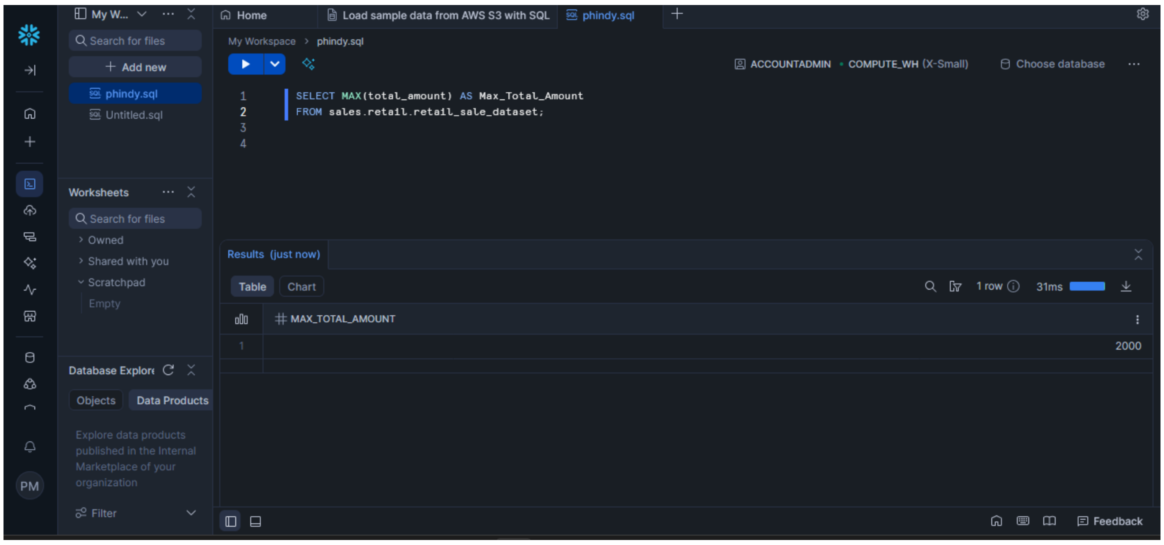
Results (just now) — 1 row, 398ms

	AVERAGE_AGE
1	41.392000



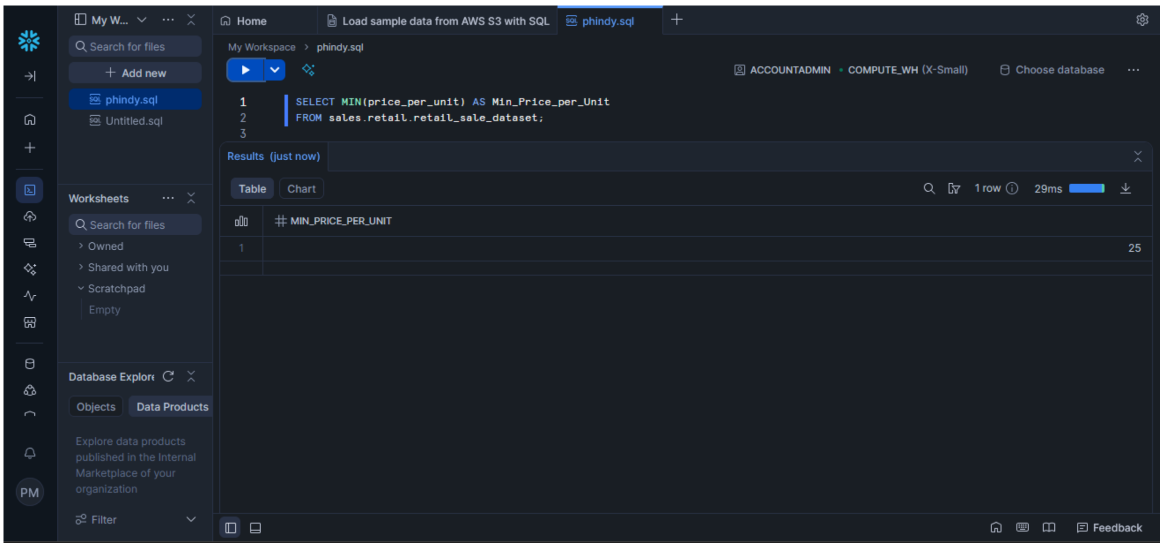
```
SELECT SUM(quantity) AS Total_Quantity
FROM sales.retail.retail_sale_dataset;
```

```
  TOTAL_QUANTITY
1           2514
```



```
SELECT MAX(total_amount) AS Max_Total_Amount
FROM sales.retail.retail_sale_dataset;
```

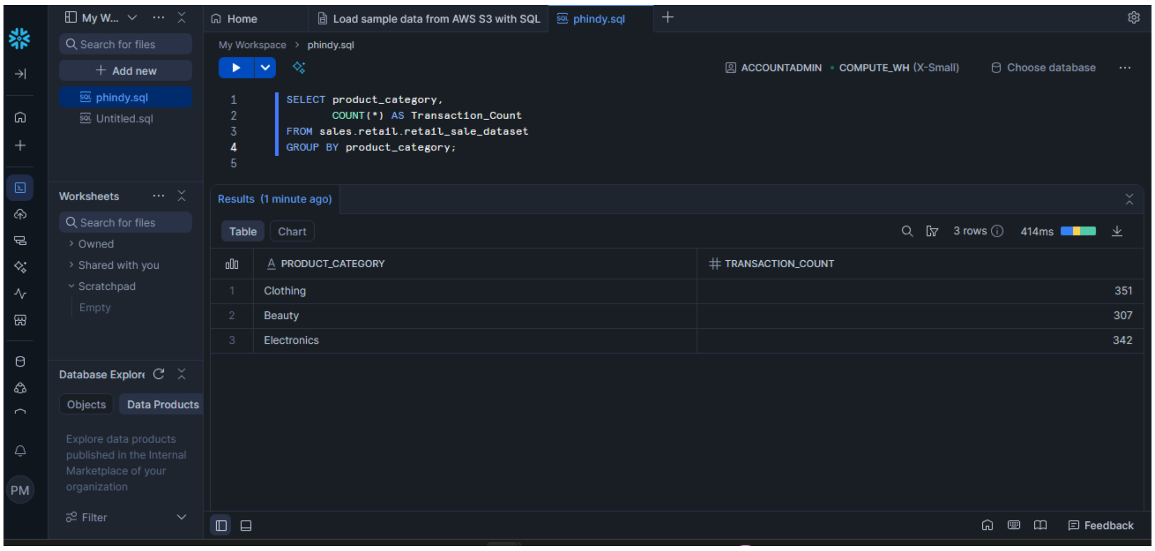
```
  MAX_TOTAL_AMOUNT
1             2000
```



```
SELECT MIN(price_per_unit) AS Min_Price_per_Unit
FROM sales.retail.retail_sale_dataset;
```

```
  MIN_PRICE_PER_UNIT
1                 25
```

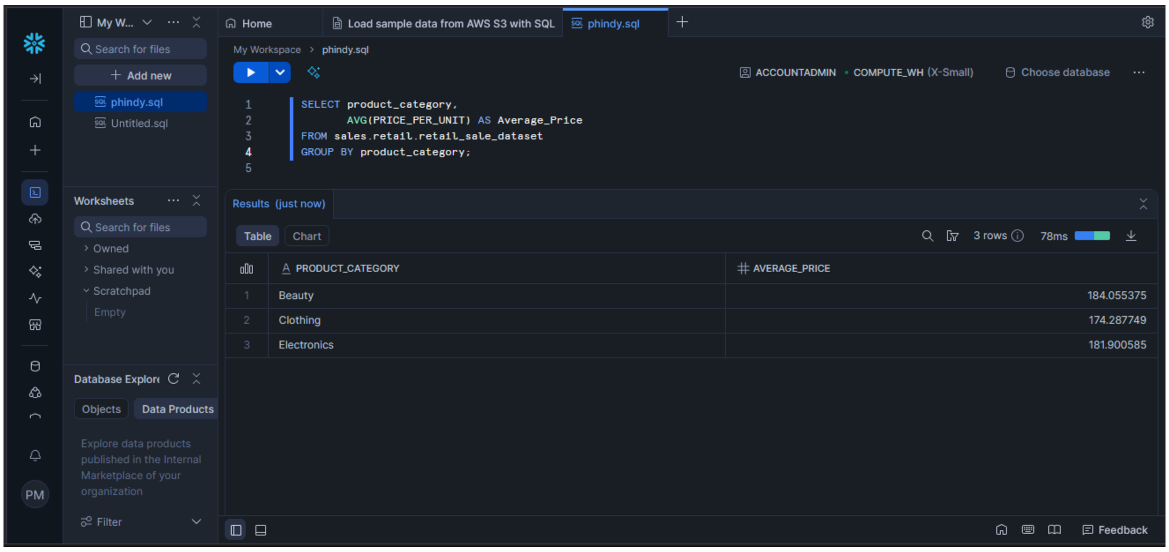
5. GROUP BY Statement



```sql
SELECT product_category,
       COUNT(*) AS Transaction_Count
FROM sales.retail.retail_sale_dataset
GROUP BY product_category;
```

Results (1 minute ago) — 3 rows, 414ms

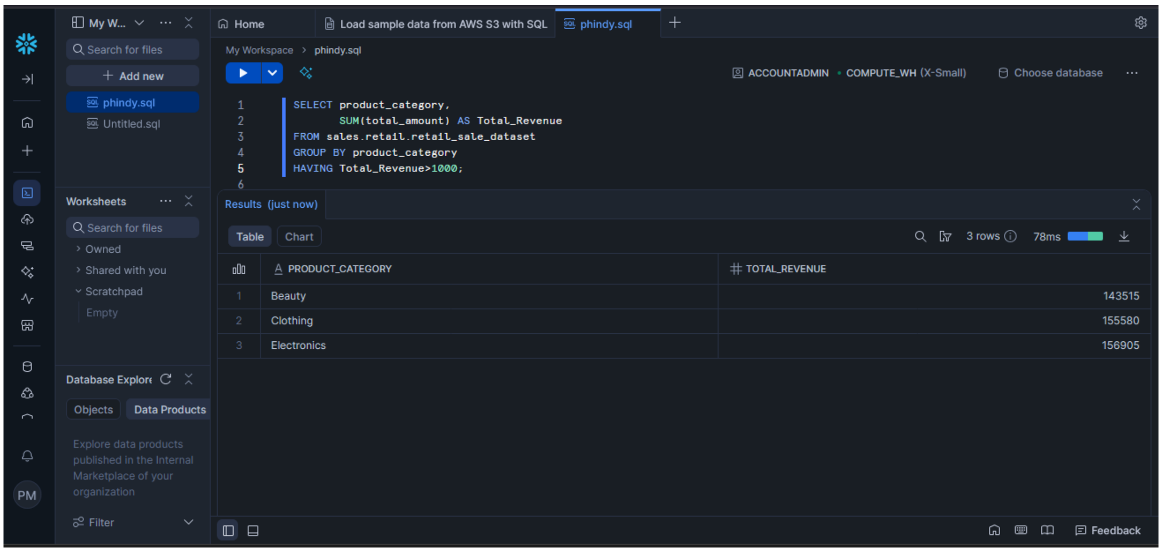
	PRODUCT_CATEGORY	TRANSACTION_COUNT
1	Clothing	351
2	Beauty	307
3	Electronics	342



```sql
SELECT product_category,
       AVG(PRICE_PER_UNIT) AS Average_Price
FROM sales.retail.retail_sale_dataset
GROUP BY product_category;
```

Results (just now) — 3 rows, 78ms

	PRODUCT_CATEGORY	AVERAGE_PRICE
1	Beauty	184.055375
2	Clothing	174.287749
3	Electronics	181.900585

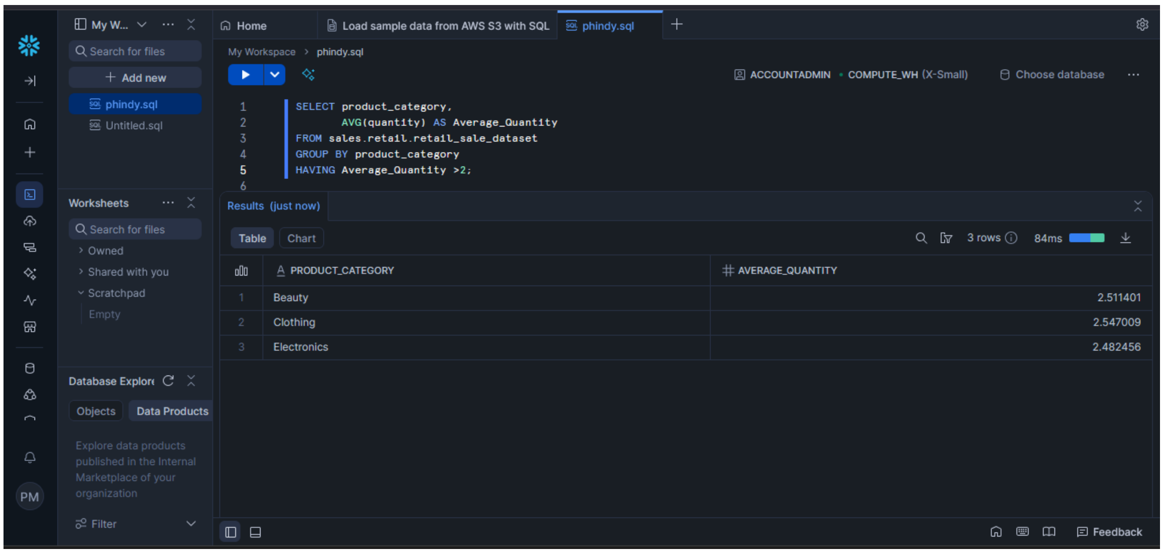


```sql
SELECT product_category,
       SUM(total_amount) AS Total_Revenue
FROM sales.retail.retail_sale_dataset
GROUP BY product_category
HAVING Total_Revenue>1000;
```

Results (just now) — 3 rows, 78ms

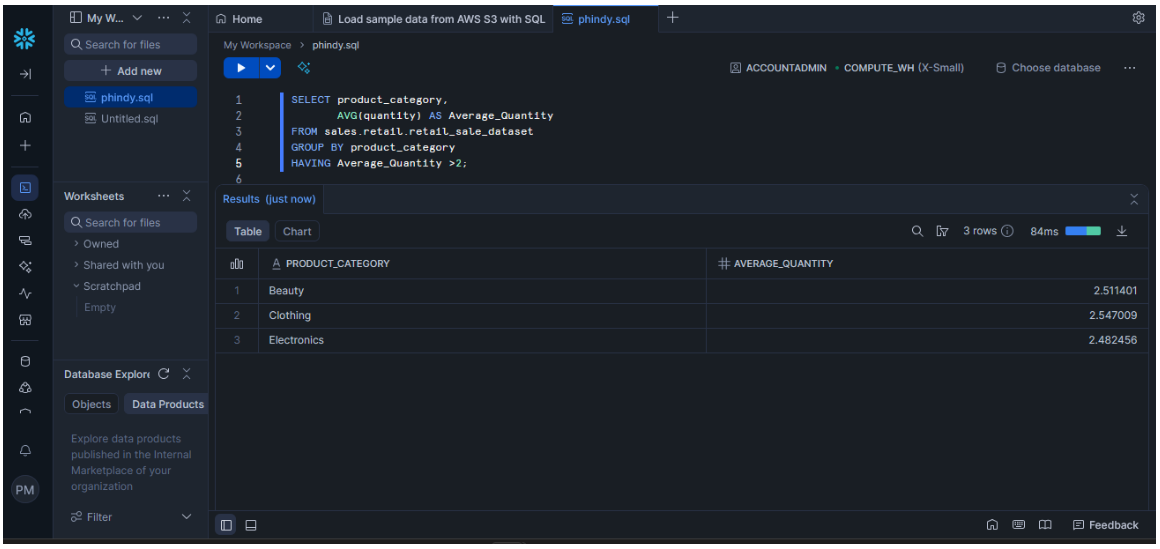
	PRODUCT_CATEGORY	TOTAL_REVENUE
1	Beauty	143515
2	Clothing	155580
3	Electronics	156905

6. HAVING Clause



```sql
SELECT product_category,
       AVG(quantity) AS Average_Quantity
FROM sales.retail.retail_sale_dataset
GROUP BY product_category
HAVING Average_Quantity >2;
```

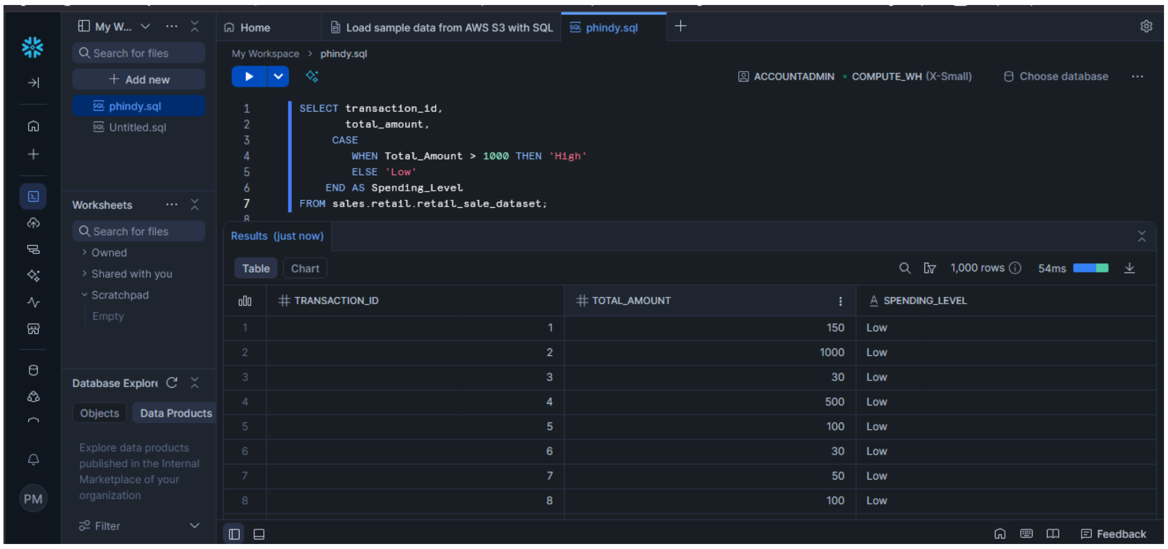
	PRODUCT_CATEGORY	AVERAGE_QUANTITY
1	Beauty	2.511401
2	Clothing	2.547009
3	Electronics	2.482456



```sql
SELECT product_category,
       AVG(quantity) AS Average_Quantity
FROM sales.retail.retail_sale_dataset
GROUP BY product_category
HAVING Average_Quantity >2;
```

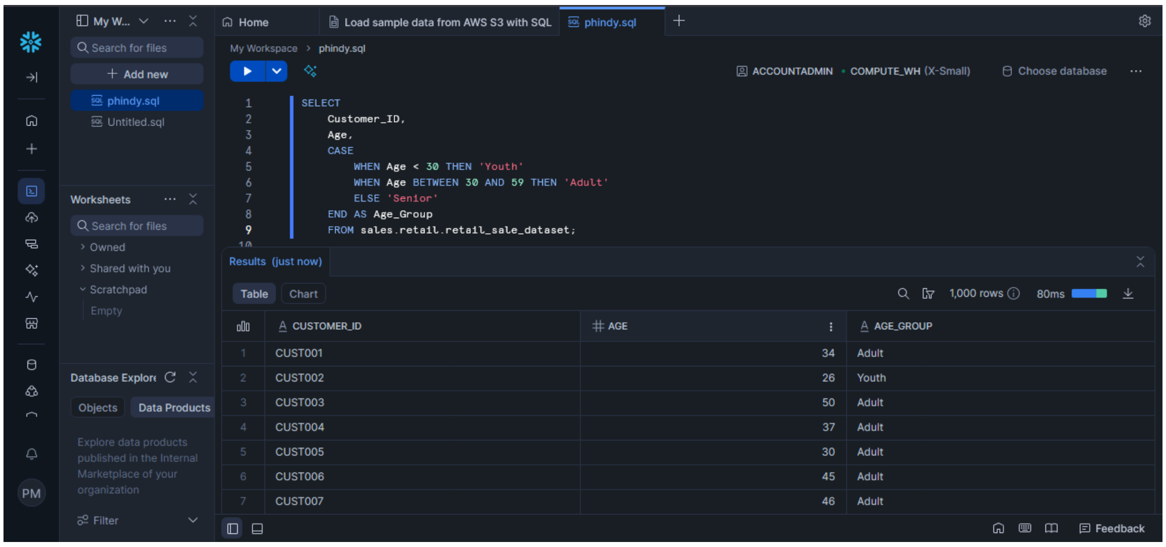
	PRODUCT_CATEGORY	AVERAGE_QUANTITY
1	Beauty	2.511401
2	Clothing	2.547009
3	Electronics	2.482456

7. CASE Statement



```sql
SELECT transaction_id,
       total_amount,
     CASE
        WHEN Total_Amount > 1000 THEN 'High'
        ELSE 'Low'
    END AS Spending_Level
FROM sales.retail.retail_sale_dataset;
```

	TRANSACTION_ID	TOTAL_AMOUNT	SPENDING_LEVEL
1	1	150	Low
2	2	1000	Low
3	3	30	Low
4	4	500	Low
5	5	100	Low
6	6	30	Low
7	7	50	Low
8	8	100	Low



```
SELECT
    Customer_ID,
    Age,
    CASE
        WHEN Age < 30 THEN 'Youth'
        WHEN Age BETWEEN 30 AND 59 THEN 'Adult'
        ELSE 'Senior'
    END AS Age_Group
    FROM sales.retail.retail_sale_dataset;
```

Results (just now) — 1,000 rows, 80ms

	CUSTOMER_ID	AGE	AGE_GROUP
1	CUST001	34	Adult
2	CUST002	26	Youth
3	CUST003	50	Adult
4	CUST004	37	Adult
5	CUST005	30	Adult
6	CUST006	45	Adult
7	CUST007	46	Adult