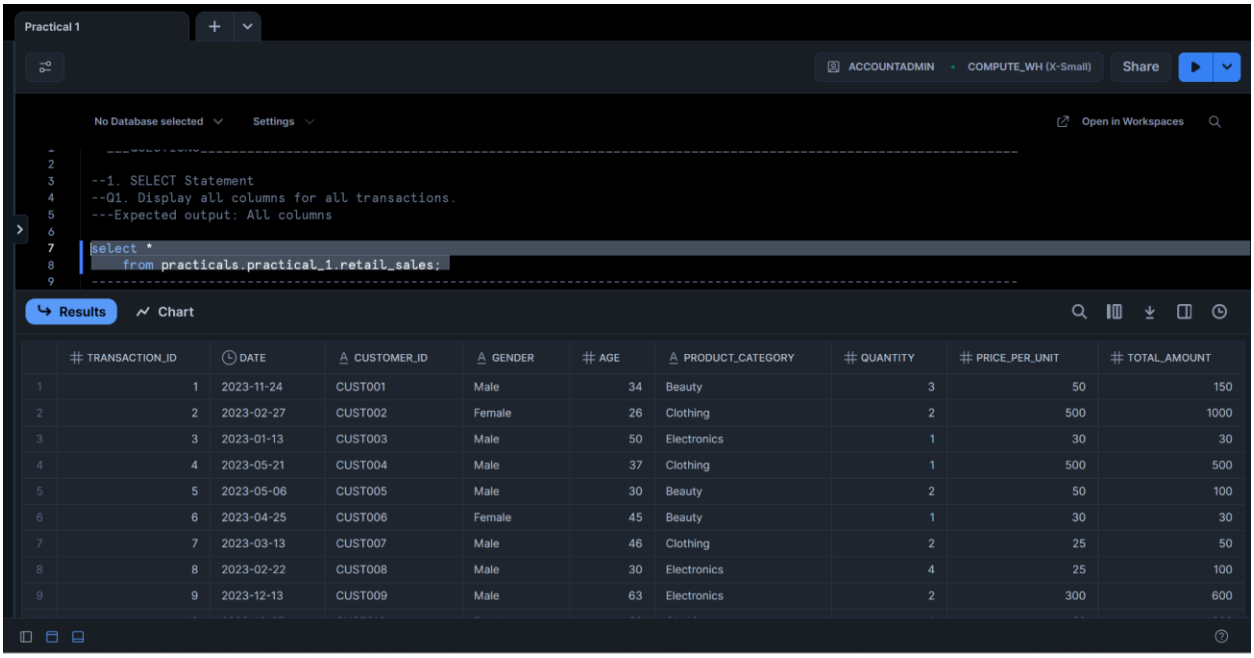
NTSHEMBO MALUEKE

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

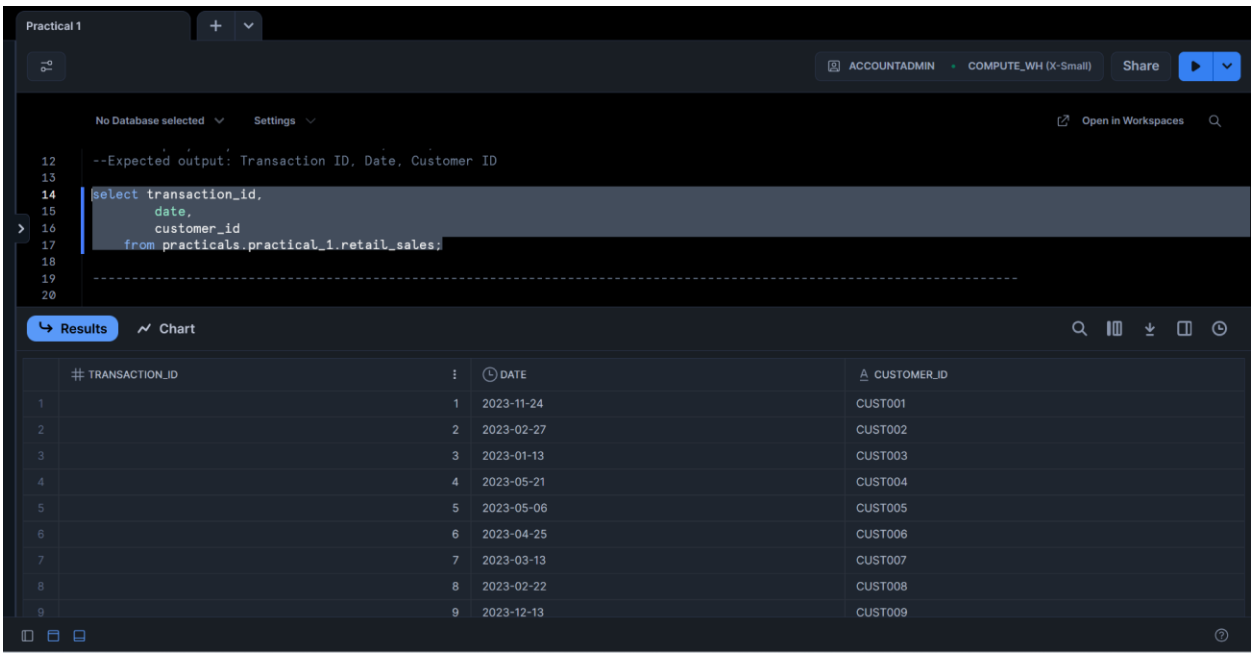
QUESTIONS

Q1



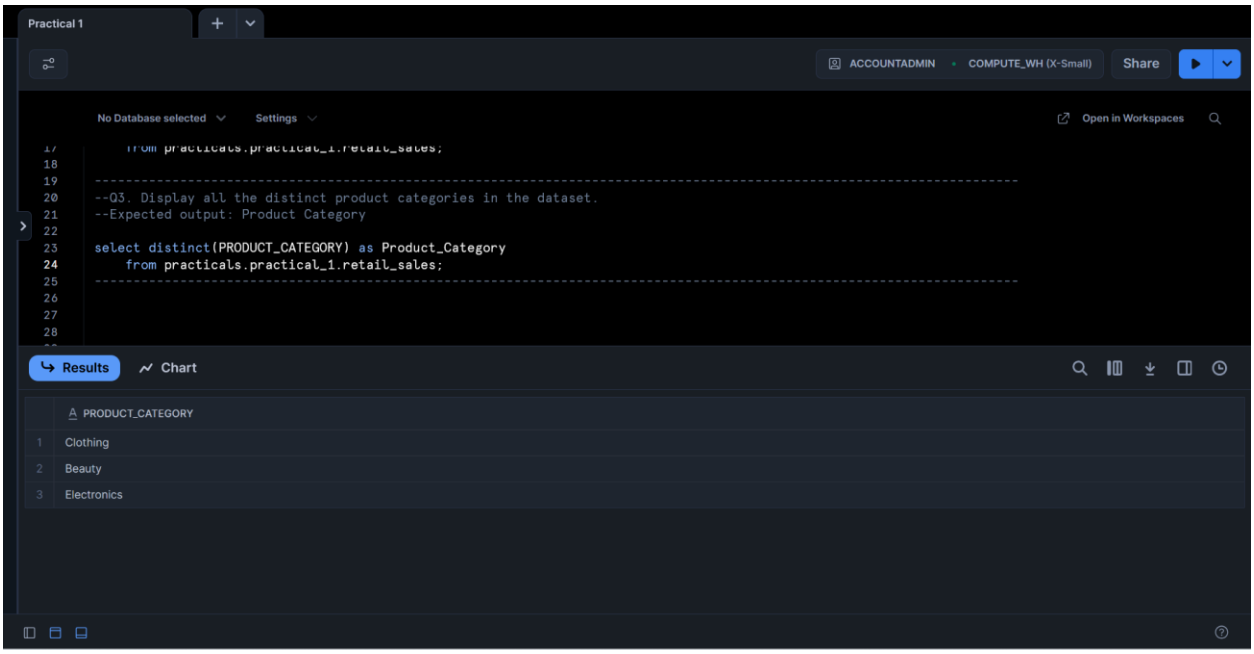
	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
5	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
7	7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
8	8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600

Q2

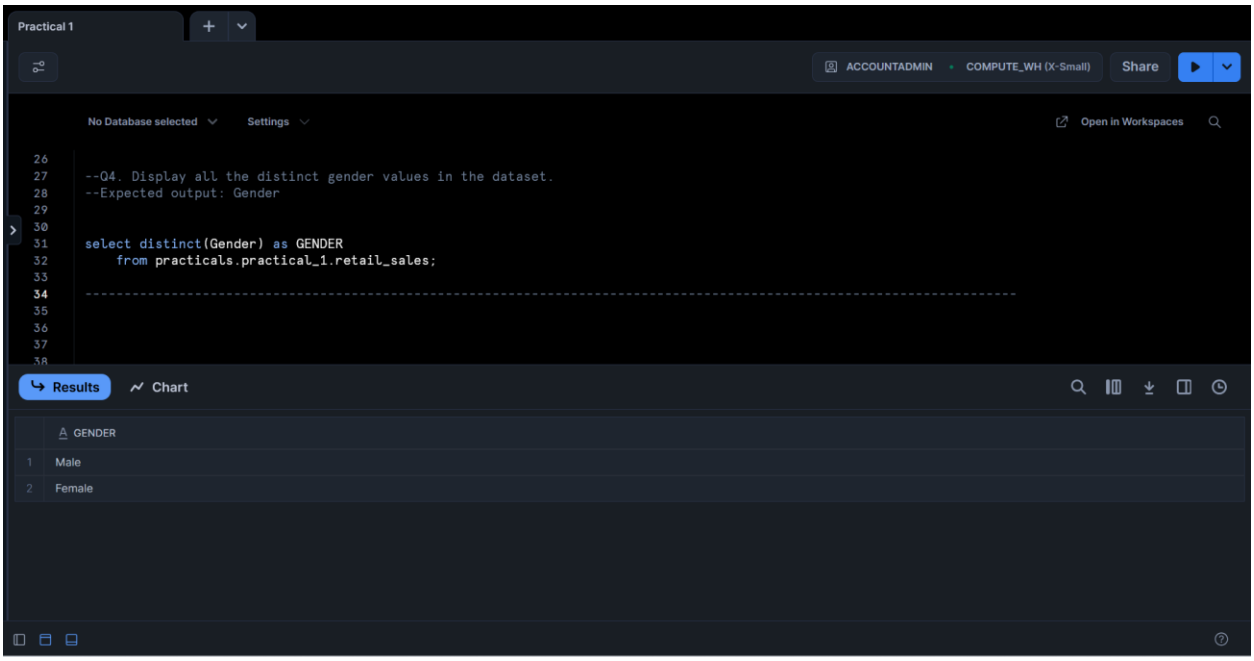


	TRANSACTION_ID	DATE	CUSTOMER_ID
1	1	2023-11-24	CUST001
2	2	2023-02-27	CUST002
3	3	2023-01-13	CUST003
4	4	2023-05-21	CUST004
5	5	2023-05-06	CUST005
6	6	2023-04-25	CUST006
7	7	2023-03-13	CUST007
8	8	2023-02-22	CUST008
9	9	2023-12-13	CUST009

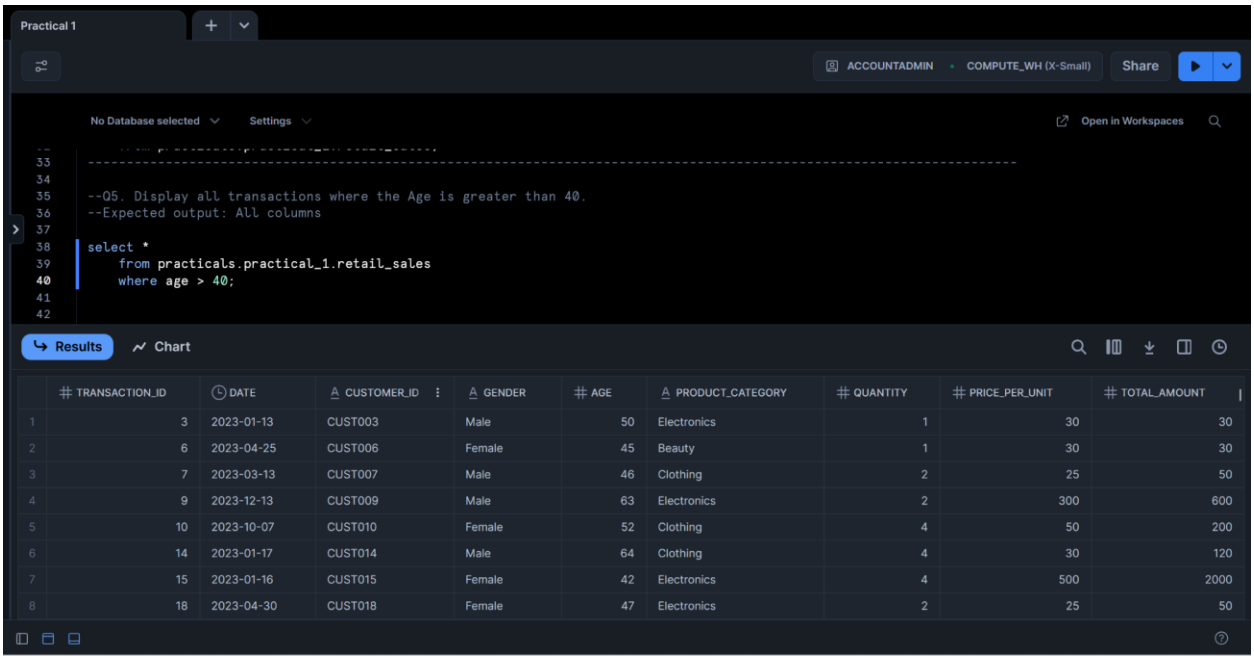
Q3


Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q3. Display all the distinct product categories in the dataset.
--Expected output: Product Category
select distinct(PRODUCT_CATEGORY) as Product_Category
from practicals.practical_1.retail_sales;
Results
Chart
PRODUCT_CATEGORY
1 Clothing
2 Beauty
3 Electronics

Q4

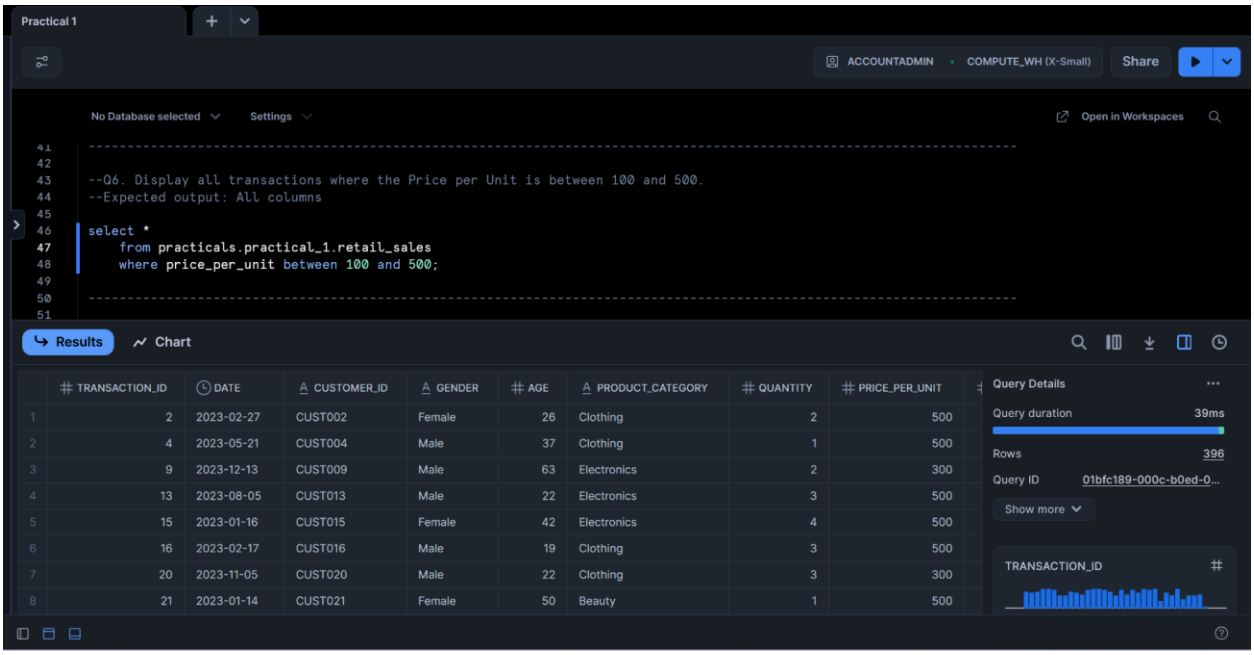

Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q4. Display all the distinct gender values in the dataset.
--Expected output: Gender
select distinct(Gender) as GENDER
from practicals.practical_1.retail_sales;
Results
Chart
GENDER
1 Male
2 Female

Q5



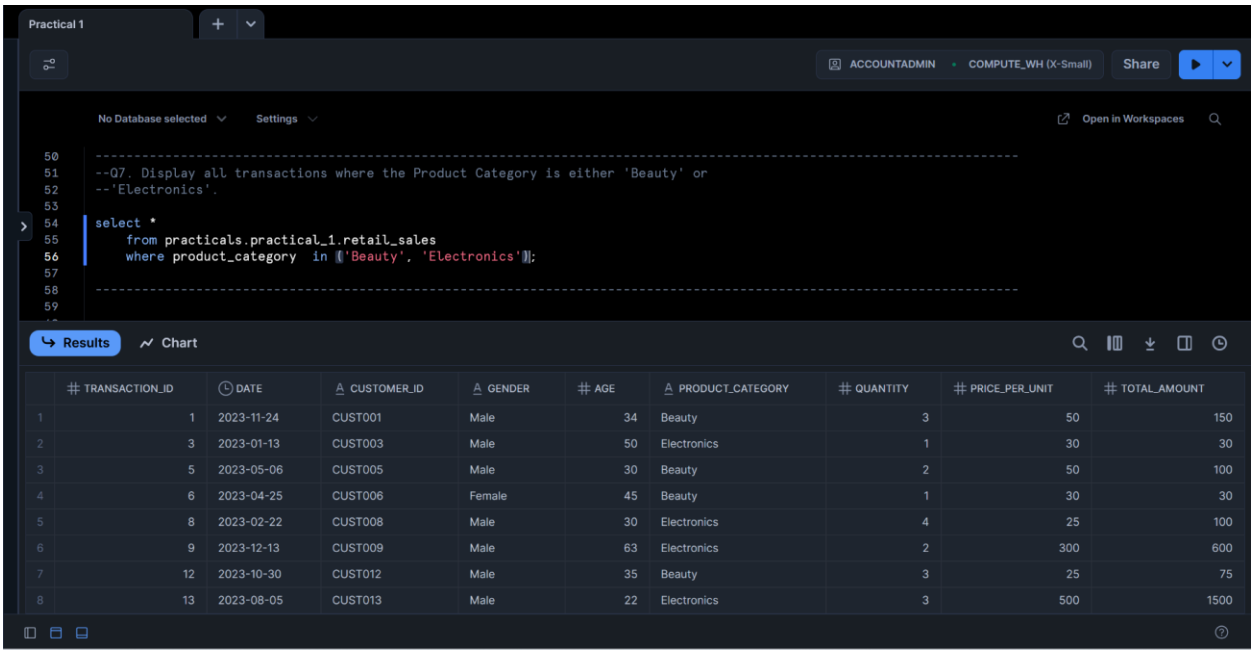
	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
2	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
3	7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
4	9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
5	10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30	120
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000
8	18	2023-04-30	CUST018	Female	47	Electronics	2	25	50

Q6



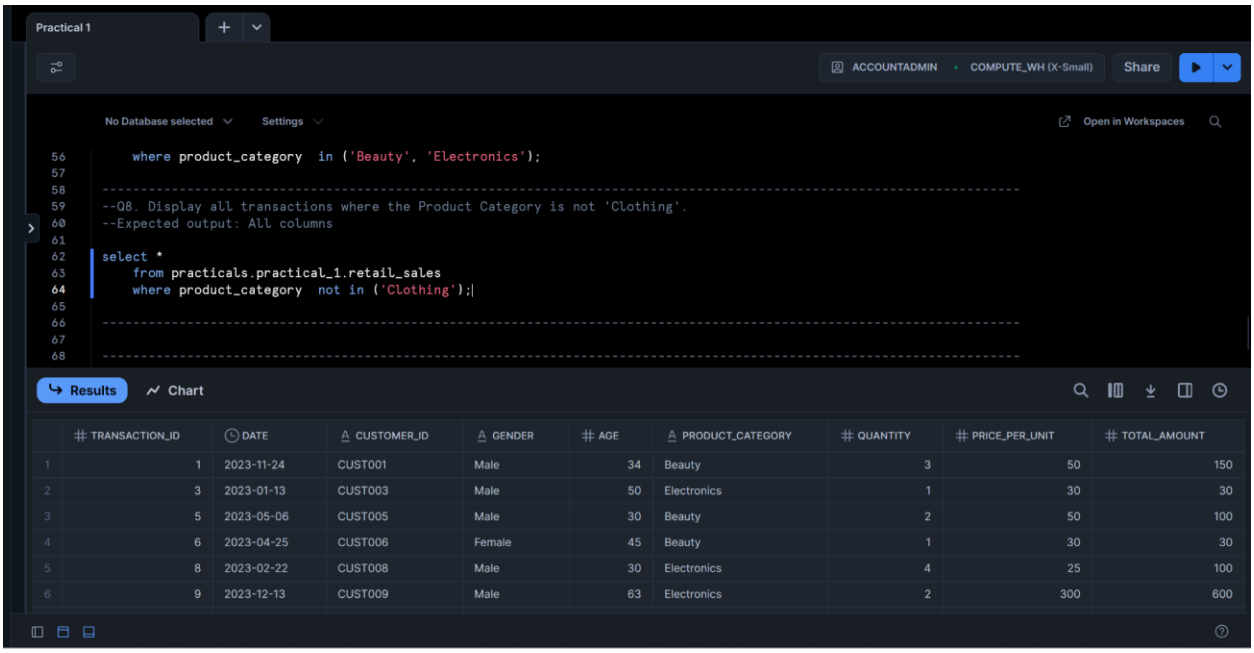
	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500
2	4	2023-05-21	CUST004	Male	37	Clothing	1	500
3	9	2023-12-13	CUST009	Male	63	Electronics	2	300
4	13	2023-08-05	CUST013	Male	22	Electronics	3	500
5	15	2023-01-16	CUST015	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

Q7



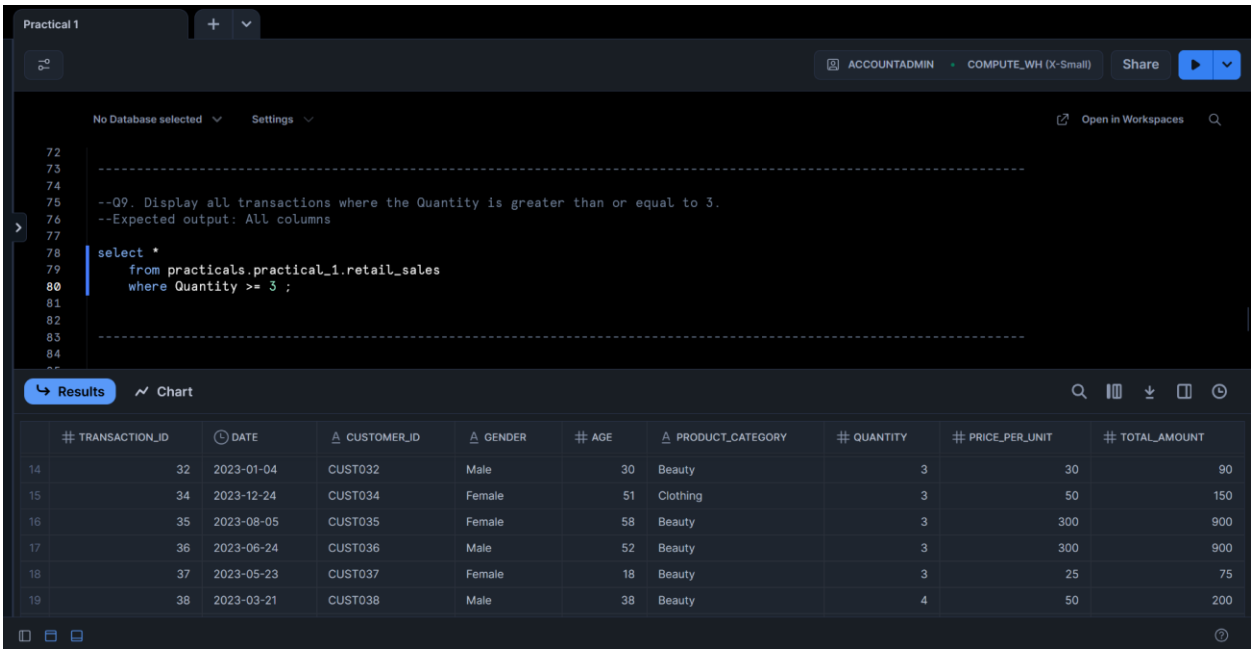
TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500

Q8



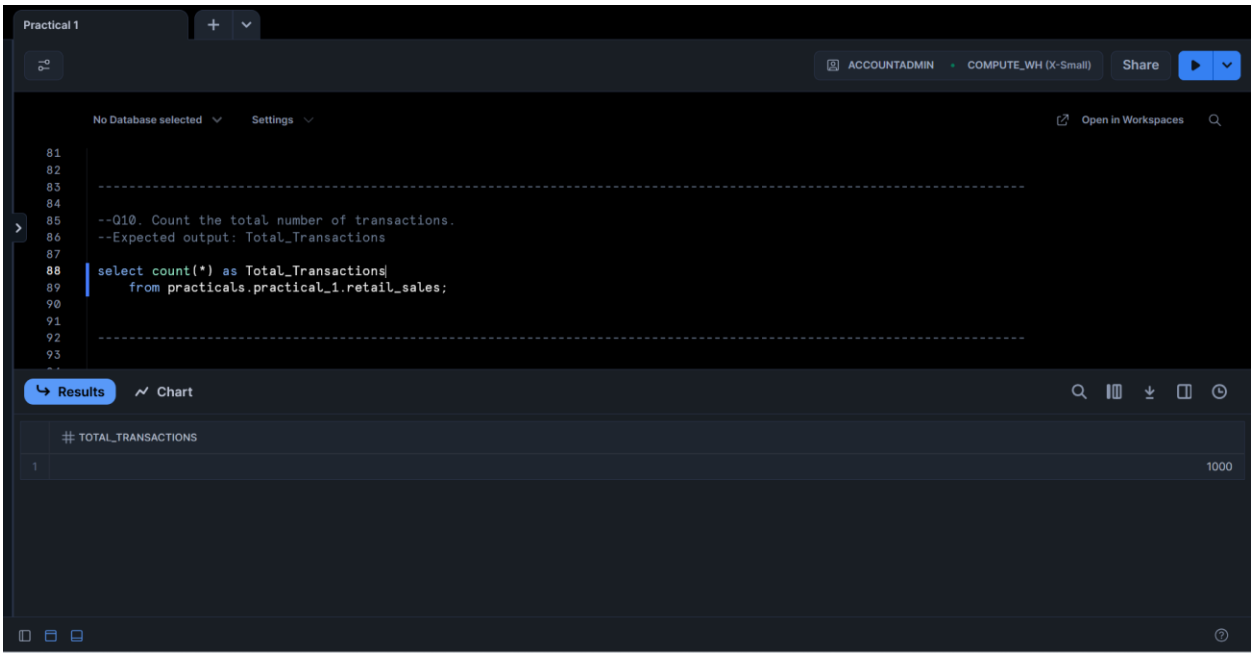
TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	2023-12-13	CUST009	Male	63	Electronics	2	300	600

Q9



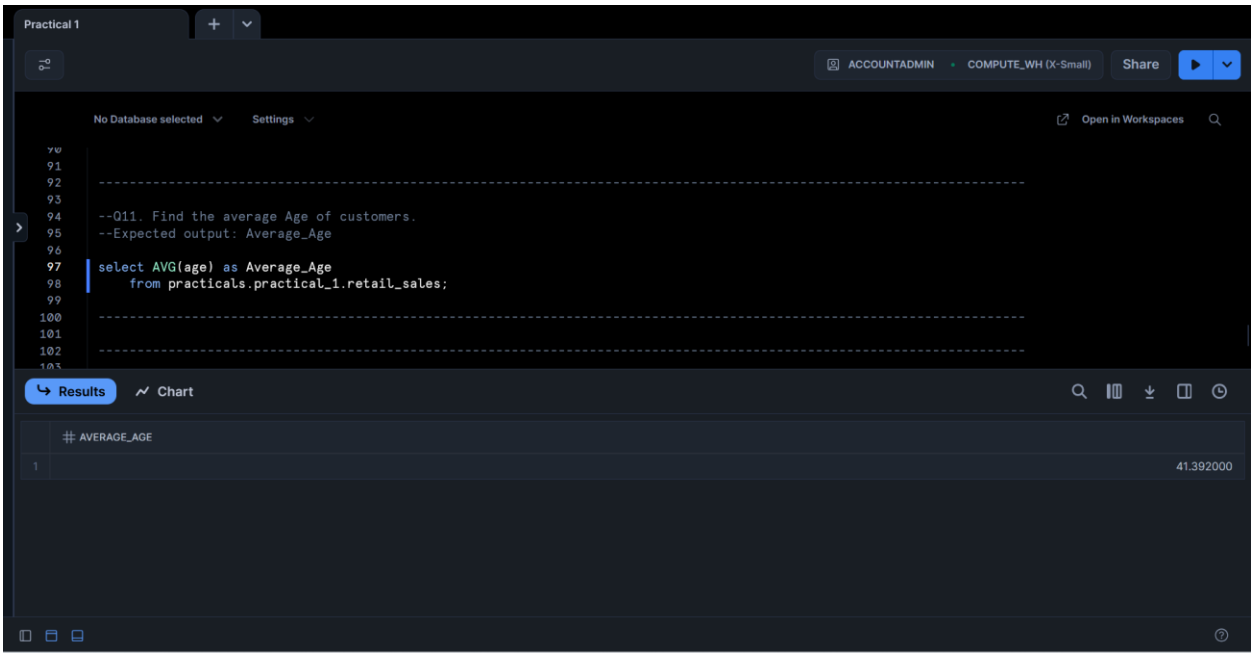
	TRANSACTION_ID	DATE	CUSTOMER_ID	GENDER	AGE	PRODUCT_CATEGORY	QUANTITY	PRICE_PER_UNIT	TOTAL_AMOUNT
14	32	2023-01-04	CUST032	Male	30	Beauty	3	30	90
15	34	2023-12-24	CUST034	Female	51	Clothing	3	50	150
16	35	2023-08-05	CUST035	Female	58	Beauty	3	300	900
17	36	2023-06-24	CUST036	Male	52	Beauty	3	300	900
18	37	2023-05-23	CUST037	Female	18	Beauty	3	25	75
19	38	2023-03-21	CUST038	Male	38	Beauty	4	50	200

Q10

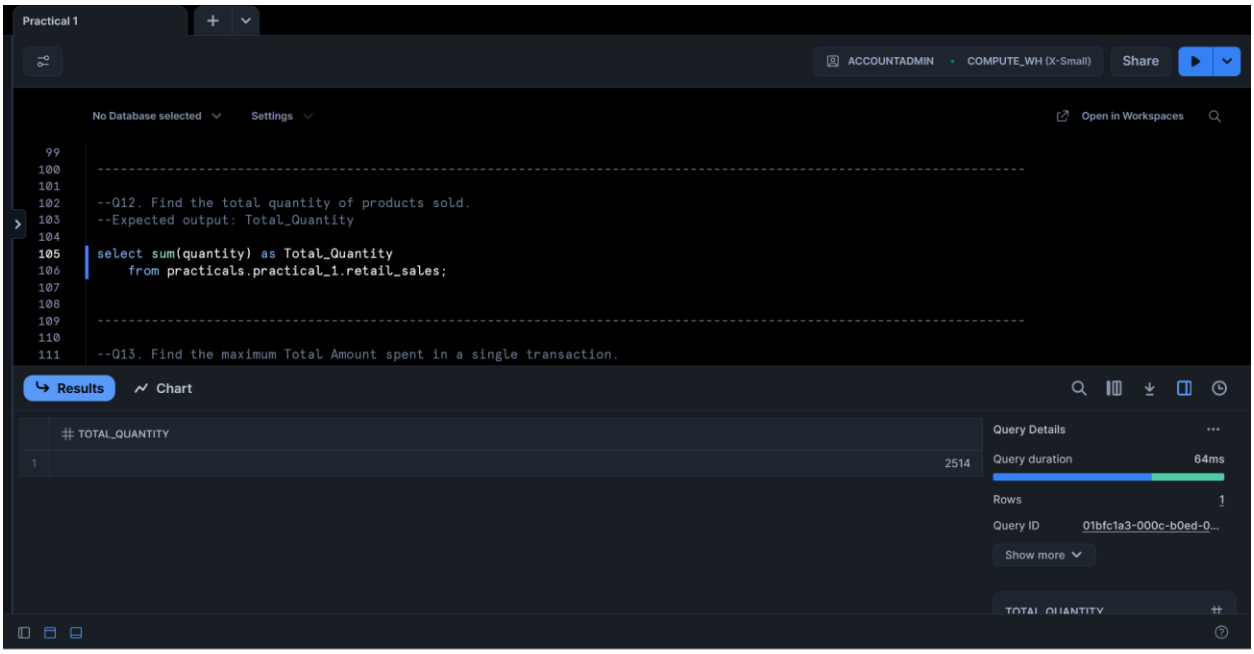


	TOTAL_TRANSACTIONS
1	1000

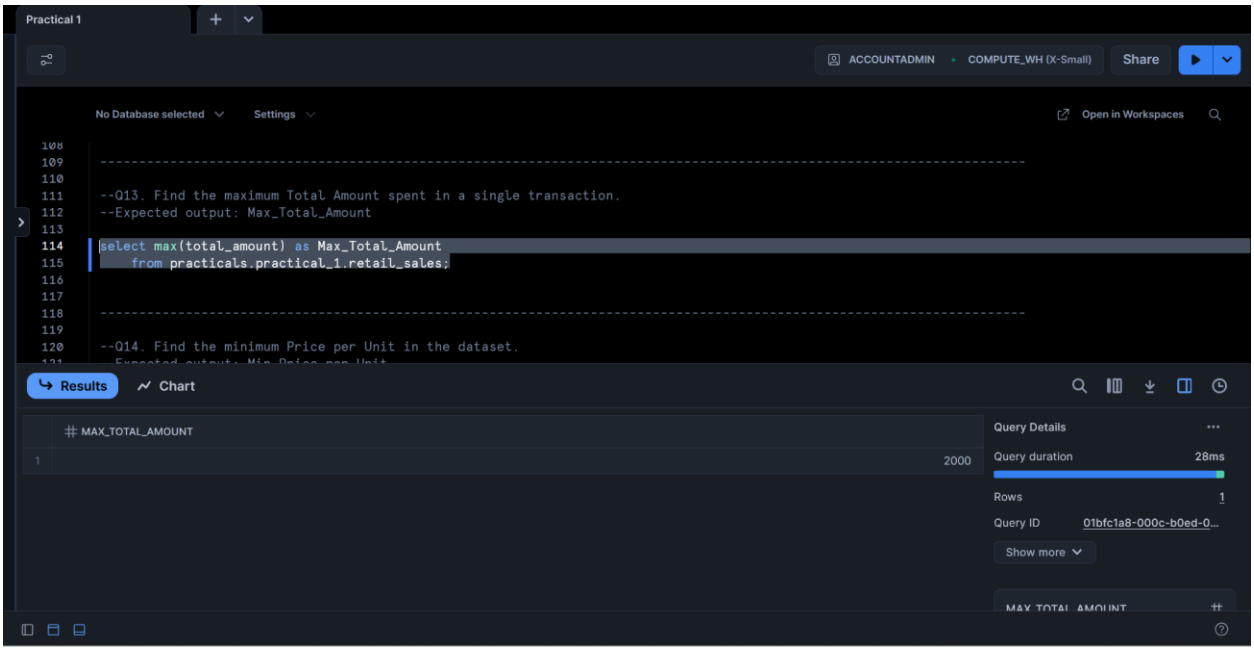
Q11


Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q11. Find the average Age of customers.
--Expected output: Average_Age
select AVG(age) as Average_Age
from practicals.practical_1.retail_sales;
Results
Chart
AVERAGE_AGE
1
41.392000

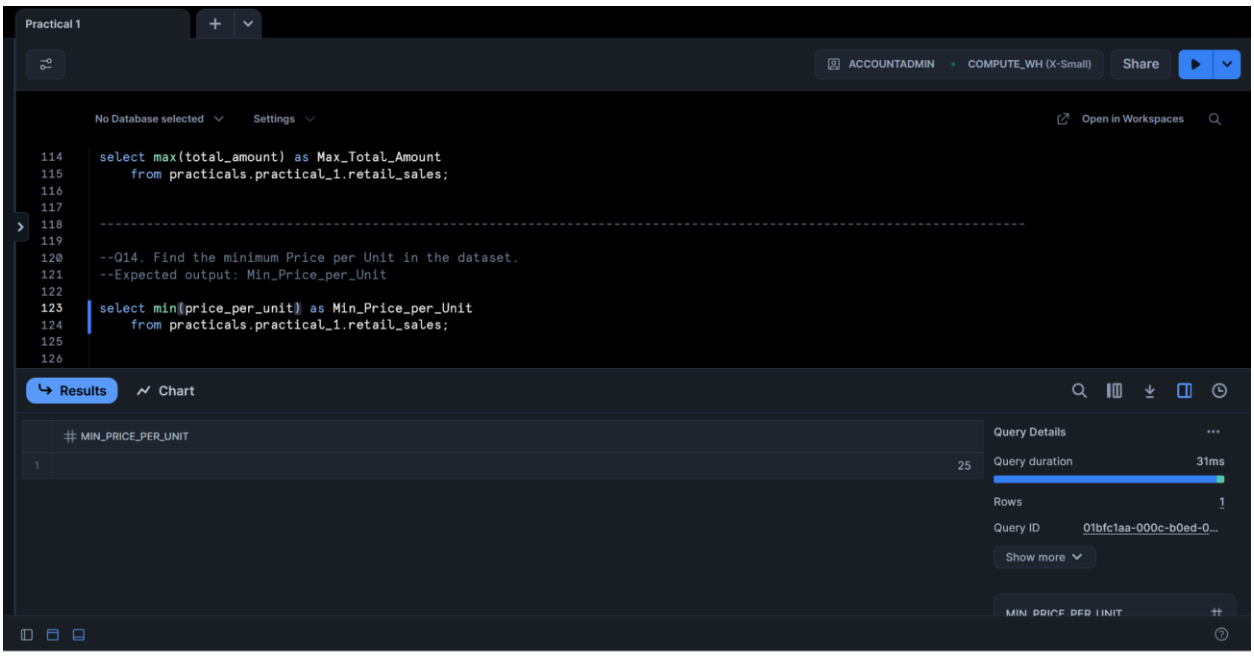
Q12


Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q12. Find the total quantity of products sold.
--Expected output: Total_Quantity
select sum(quantity) as Total_Quantity
from practicals.practical_1.retail_sales;
--Q13. Find the maximum Total Amount spent in a single transaction.
Results
Chart
TOTAL_QUANTITY
1
2514
Query Details
Query duration
64ms
Rows
1
Query ID
01bfc1a3-000c-b0ed-0...
Show more

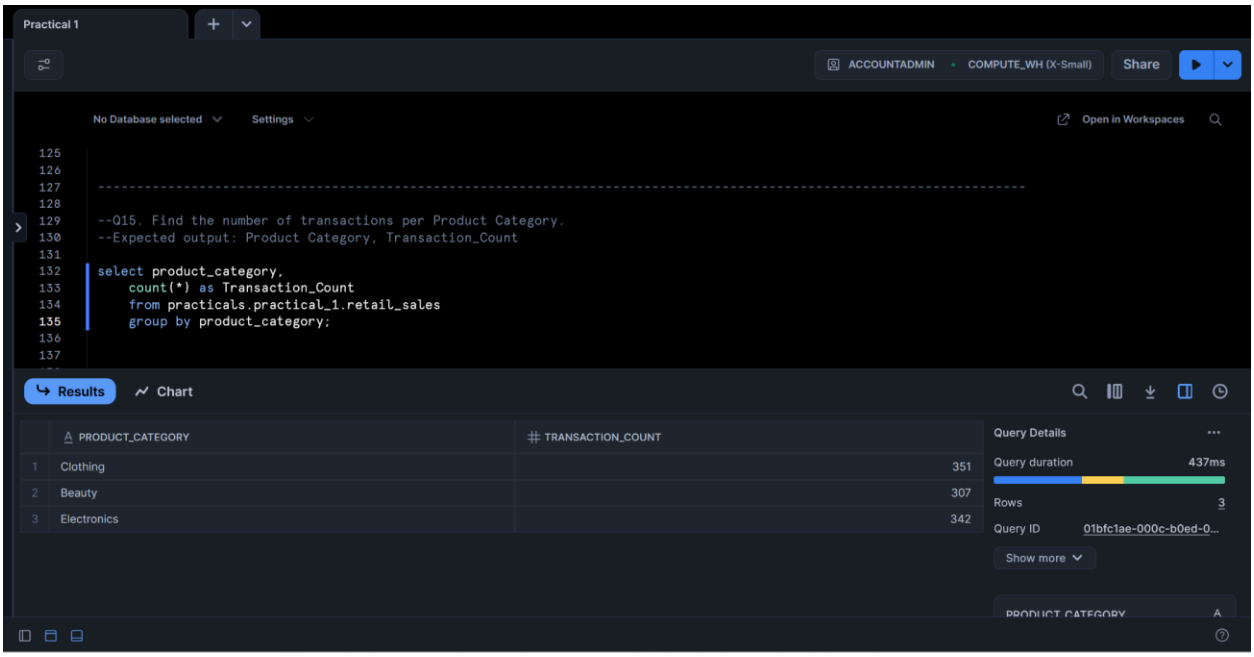
Q13


Practical 1
ACCOUNTADMIN
COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q13. Find the maximum Total Amount spent in a single transaction.
--Expected output: Max_Total_Amount
select max(total_amount) as Max_Total_Amount
from practicals.practical_1.retail_sales;
--Q14. Find the minimum Price per Unit in the dataset.
Results
Chart
MAX_TOTAL_AMOUNT
1
2000
Query Details
Query duration
28ms
Rows
1
Query ID
01bfc1a8-000c-b0ed-0...
Show more

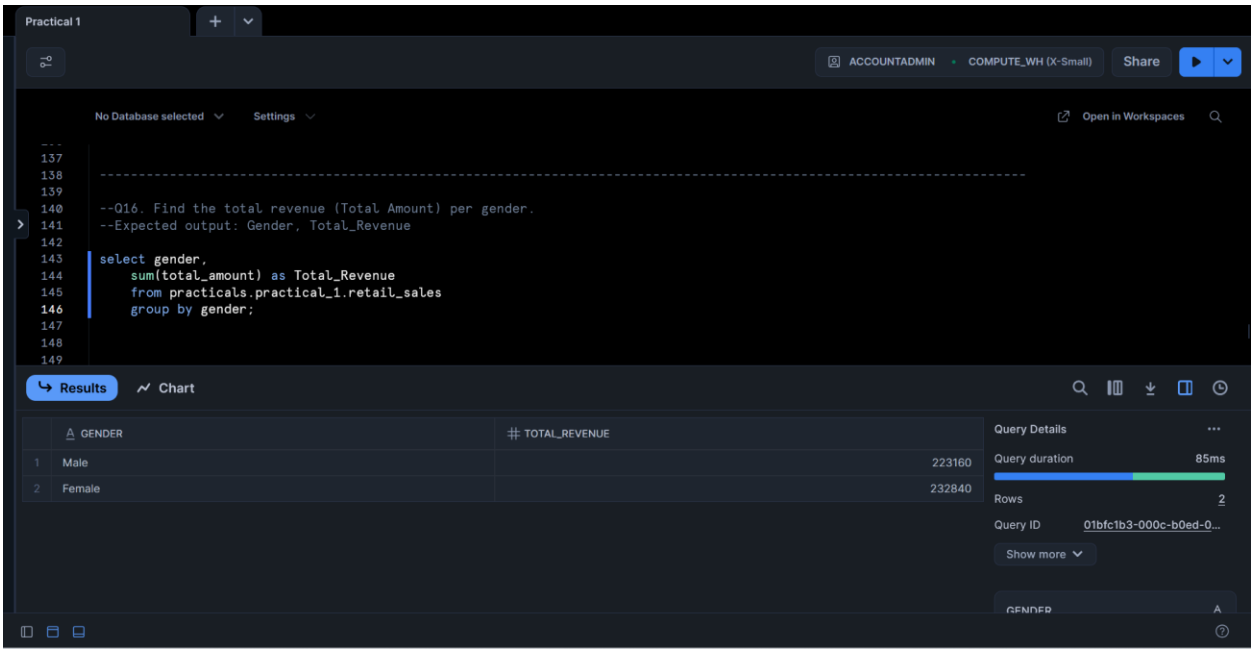
Q14


Practical 1
ACCOUNTADMIN
COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
select max(total_amount) as Max_Total_Amount
from practicals.practical_1.retail_sales;
--Q14. Find the minimum Price per Unit in the dataset.
--Expected output: Min_Price_per_Unit
select min(price_per_unit) as Min_Price_per_Unit
from practicals.practical_1.retail_sales;
Results
Chart
MIN_PRICE_PER_UNIT
1
25
Query Details
Query duration
31ms
Rows
1
Query ID
01bfc1aa-000c-b0ed-0...
Show more

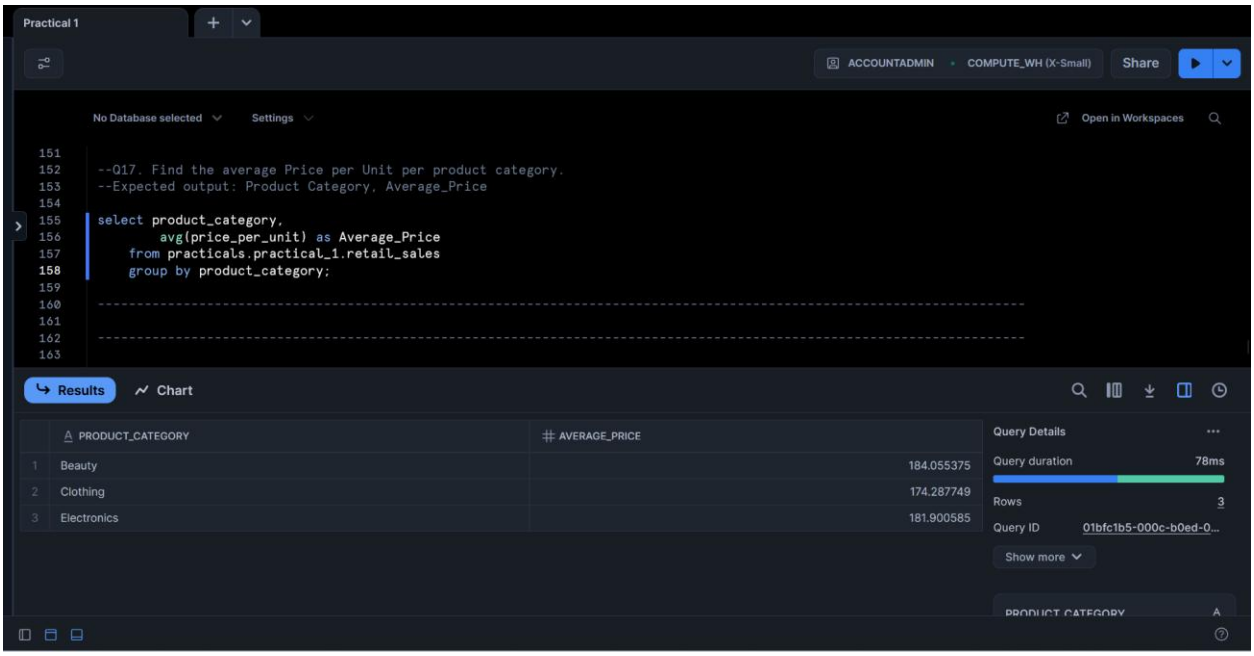
Q15


Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q15. Find the number of transactions per Product Category.
--Expected output: Product Category, Transaction_Count
select product_category,
count(*) as Transaction_Count
from practicals.practical_1.retail_sales
group by product_category;
Results
Chart
PRODUCT_CATEGORY
TRANSACTION_COUNT
1 Clothing 351
2 Beauty 307
3 Electronics 342
Query Details
Query duration 437ms
Rows 3
Query ID 01bfc1ae-000c-b0ed-0...
Show more

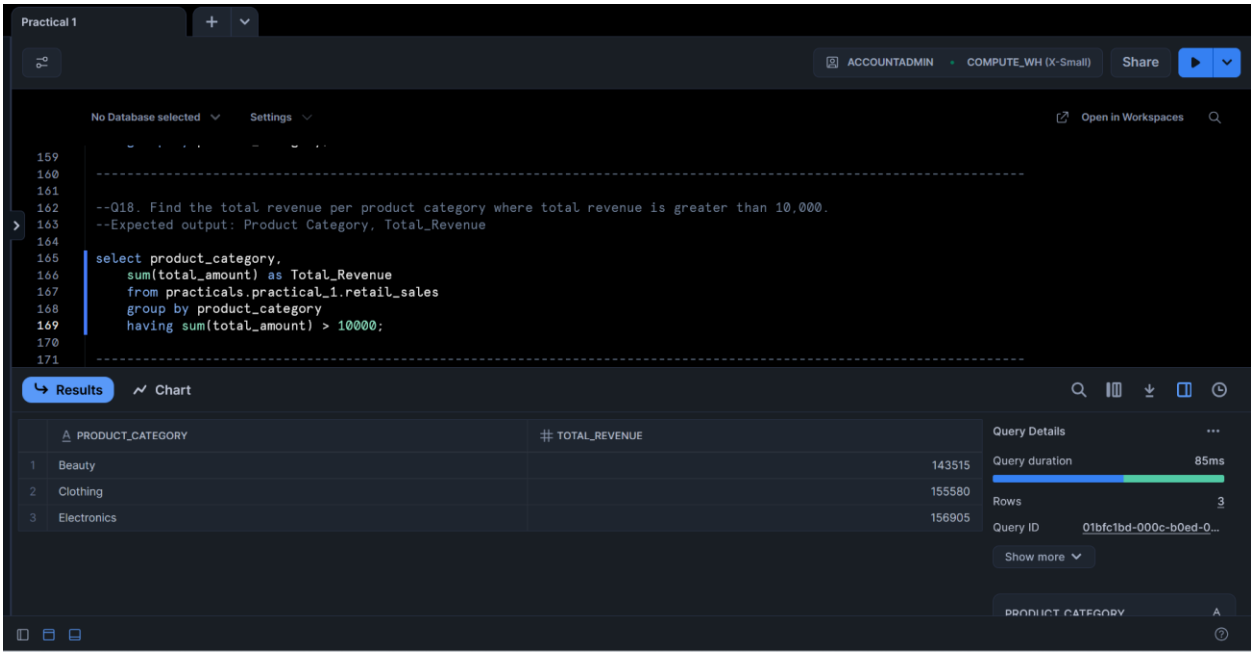
Q16


Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q16. Find the total revenue (Total Amount) per gender.
--Expected output: Gender, Total_Revenue
select gender,
sum(total_amount) as Total_Revenue
from practicals.practical_1.retail_sales
group by gender;
Results
Chart
GENDER
TOTAL_REVENUE
1 Male 223160
2 Female 232840
Query Details
Query duration 85ms
Rows 2
Query ID 01bfc1b3-000c-b0ed-0...
Show more

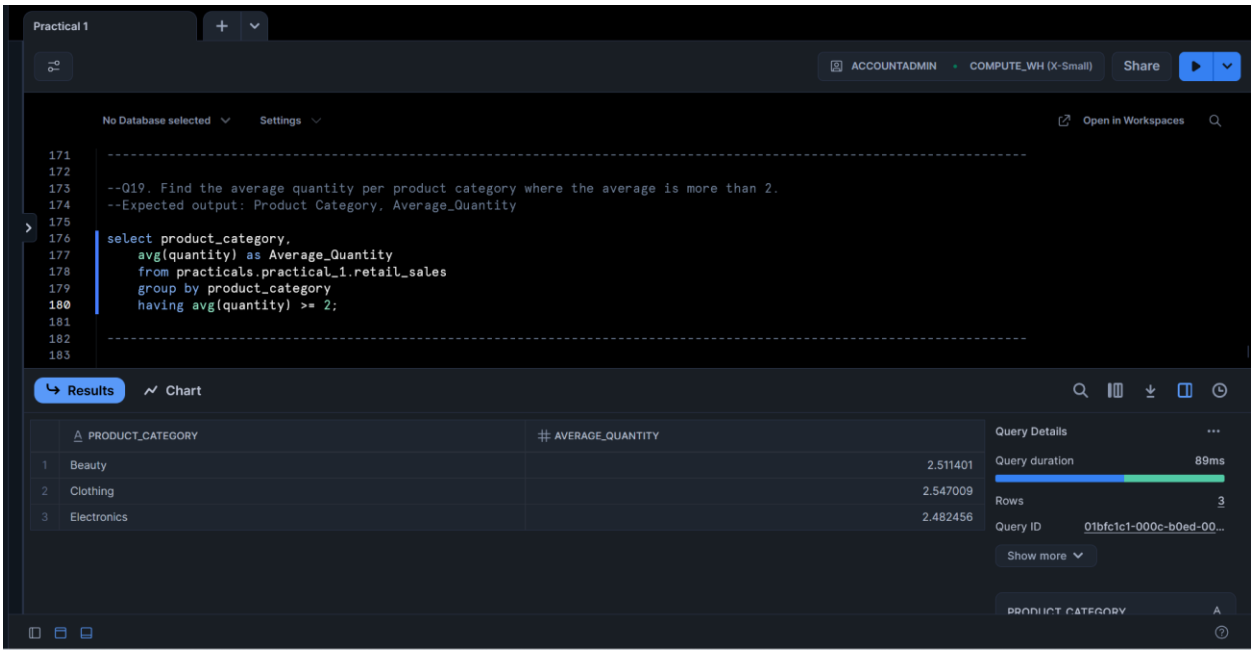
Q17


Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
151
152 --Q17. Find the average Price per Unit per product category.
153 --Expected output: Product Category, Average_Price
154
155 select product_category,
156 avg(price_per_unit) as Average_Price
157 from practicals.practical_1.retail_sales
158 group by product_category;
159
160
161
162
163
Results
Chart
PRODUCT_CATEGORY
AVERAGE_PRICE
1 Beauty 184.055375
2 Clothing 174.287749
3 Electronics 181.900585
Query Details
Query duration 78ms
Rows 3
Query ID 01bfc1b5-000c-b0ed-0...
Show more

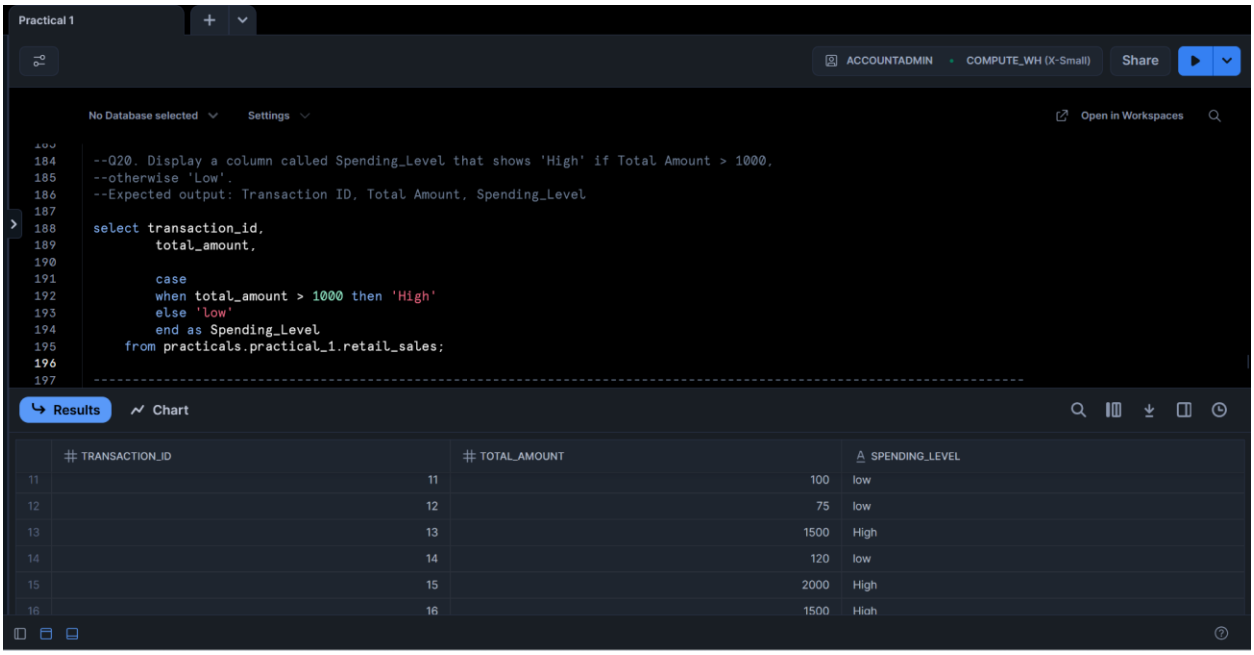
Q18


Practical 1
ACCOUNTADMIN • COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
159
160
161
162 --Q18. Find the total revenue per product category where total revenue is greater than 10,000.
163 --Expected output: Product Category, Total_Revenue
164
165 select product_category,
166 sum(total_amount) as Total_Revenue
167 from practicals.practical_1.retail_sales
168 group by product_category
169 having sum(total_amount) > 10000;
170
171
Results
Chart
PRODUCT_CATEGORY
TOTAL_REVENUE
1 Beauty 143515
2 Clothing 155580
3 Electronics 156905
Query Details
Query duration 85ms
Rows 3
Query ID 01bfc1bd-000c-b0ed-0...
Show more

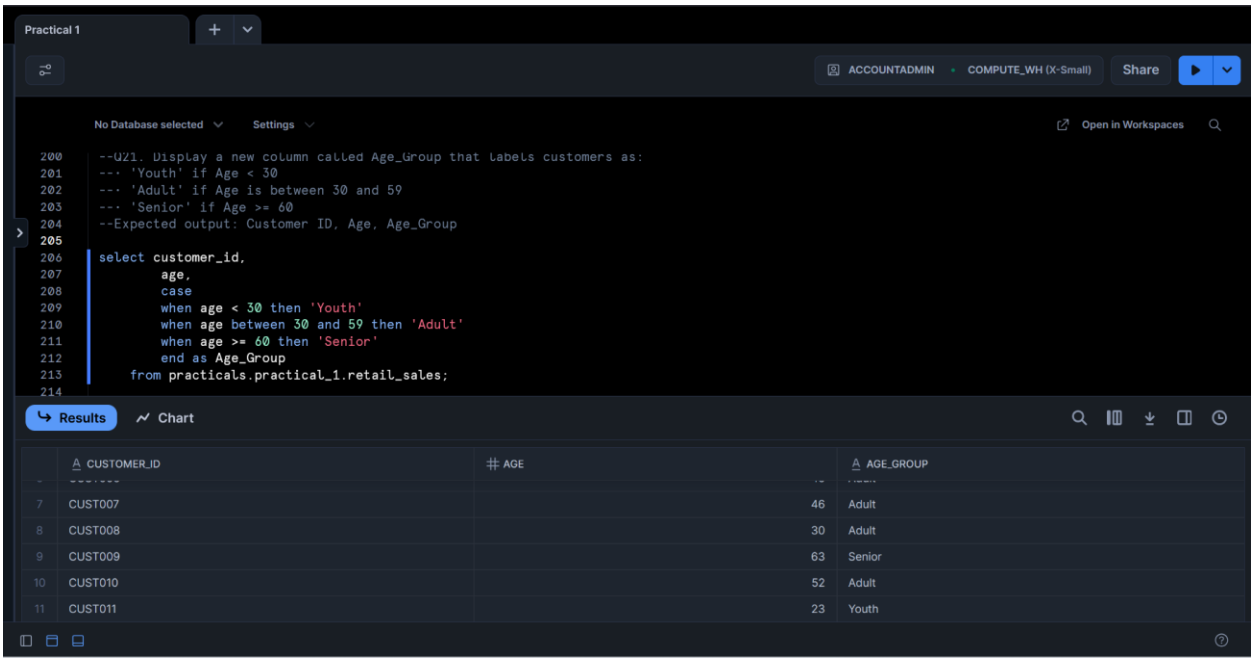
Q19


Practical 1
ACCOUNTADMIN
COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q19. Find the average quantity per product category where the average is more than 2.
--Expected output: Product Category, Average_Quantity
select product_category,
avg(quantity) as Average_Quantity
from practicals.practical_1.retail_sales
group by product_category
having avg(quantity) >= 2;
Results
Chart
PRODUCT_CATEGORY
AVERAGE_QUANTITY
1 Beauty 2.511401
2 Clothing 2.547009
3 Electronics 2.482456
Query Details
Query duration 89ms
Rows 3
Query ID 01bfc1c1-000c-b0ed-00...
Show more

Q20


Practical 1
ACCOUNTADMIN
COMPUTE_WH (X-Small)
Share
No Database selected
Settings
Open in Workspaces
--Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000,
--otherwise 'Low'.
--Expected output: Transaction ID, Total Amount, Spending_Level
select transaction_id,
total_amount,
case
when total_amount > 1000 then 'High'
else 'low'
end as Spending_Level
from practicals.practical_1.retail_sales;
Results
Chart
TRANSACTION_ID
TOTAL_AMOUNT
SPENDING_LEVEL
11 11 100 low
12 12 75 low
13 13 1500 High
14 14 120 low
15 15 2000 High
16 16 1500 High

Q21



Whole code

--BrightLight Data Analytics Coding Practical

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

--
___QUESTIONS_____
_____

--1. SELECT Statement

--Q1. Display all columns for all transactions.

---Expected output: All columns

```
select *
  from practicals.practical_1.retail_sales;

-------------------------------------------------------------------------------------------------------------
-----

--Q2. Display only the Transaction ID, Date, and Customer ID for all records.
--Expected output: Transaction ID, Date, Customer ID

select transaction_id,
      date,
      customer_id
  from practicals.practical_1.retail_sales;

-------------------------------------------------------------------------------------------------------------
-----

--Q3. Display all the distinct product categories in the dataset.
--Expected output: Product Category

select distinct(PRODUCT_CATEGORY) as Product_Category
  from practicals.practical_1.retail_sales;

-------------------------------------------------------------------------------------------------------------
-----

--Q4. Display all the distinct gender values in the dataset.
```

--Expected output: Gender

```
select distinct(Gender) as GENDER
   from practicals.practical_1.retail_sales;
```

--Q5. Display all transactions where the Age is greater than 40.

--Expected output: All columns

```
select *
   from practicals.practical_1.retail_sales
   where age > 40;
```

--Q6. Display all transactions where the Price per Unit is between 100 and 500.

--Expected output: All columns

```
select *
   from practicals.practical_1.retail_sales
   where price_per_unit between 100 and 500;
```

```
-------------------------------------------------------------------------------------------------------------
-----

--Q7. Display all transactions where the Product Category is either 'Beauty' or
--'Electronics'.

select *
  from practicals.practical_1.retail_sales
  where product_category  in ('Beauty', 'Electronics');

-------------------------------------------------------------------------------------------------------------
-----

--Q8. Display all transactions where the Product Category is not 'Clothing'.
--Expected output: All columns

select *
  from practicals.practical_1.retail_sales
  where product_category  not in ('Clothing');

-------------------------------------------------------------------------------------------------------------
-----

--Q9. Display all transactions where the Quantity is greater than or equal to 3.
--Expected output: All columns

select *
```

```
  from practicals.practical_1.retail_sales
  where Quantity >= 3 ;
```

```
--Q10. Count the total number of transactions.
--Expected output: Total_Transactions

select count(*) as Total_Transactions
  from practicals.practical_1.retail_sales;
```

```
--Q11. Find the average Age of customers.
--Expected output: Average_Age

select AVG(age) as Average_Age
  from practicals.practical_1.retail_sales;
```

```
--Q12. Find the total quantity of products sold.
```

--Expected output: Total_Quantity

```
select sum(quantity) as Total_Quantity
   from practicals.practical_1.retail_sales;
```

--Q13. Find the maximum Total Amount spent in a single transaction.

--Expected output: Max_Total_Amount

```
select max(total_amount) as Max_Total_Amount
   from practicals.practical_1.retail_sales;
```

--Q14. Find the minimum Price per Unit in the dataset.

--Expected output: Min_Price_per_Unit

```
select min(price_per_unit) as Min_Price_per_Unit
   from practicals.practical_1.retail_sales;
```

--Q15. Find the number of transactions per Product Category.

--Expected output: Product Category, Transaction_Count

```
select product_category,
   count(*) as Transaction_Count
   from practicals.practical_1.retail_sales
   group by product_category;
```

--Q16. Find the total revenue (Total Amount) per gender.

--Expected output: Gender, Total_Revenue

```
select gender,
   sum(total_amount) as Total_Revenue
   from practicals.practical_1.retail_sales
   group by gender;
```

--Q17. Find the average Price per Unit per product category.

--Expected output: Product Category, Average_Price

```
select product_category,
      avg(price_per_unit) as Average_Price
   from practicals.practical_1.retail_sales
   group by product_category;
```

--Q18. Find the total revenue per product category where total revenue is greater than 10,000.

--Expected output: Product Category, Total_Revenue

```
select product_category,
   sum(total_amount) as Total_Revenue
   from practicals.practical_1.retail_sales
   group by product_category
   having sum(total_amount) > 10000;
```

--Q19. Find the average quantity per product category where the average is more than 2.

--Expected output: Product Category, Average_Quantity

```
select product_category,
   avg(quantity) as Average_Quantity
   from practicals.practical_1.retail_sales
   group by product_category
   having avg(quantity) >= 2;

-------------------------------------------------------------------------------------------------------------
-----

--Q20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000,
--otherwise 'Low'.
--Expected output: Transaction ID, Total Amount, Spending_Level

select transaction_id,
      total_amount,

      case
      when total_amount > 1000 then 'High'
      else 'low'
      end as Spending_Level
   from practicals.practical_1.retail_sales;

-------------------------------------------------------------------------------------------------------------
-----
```

--Q21. Display a new column called Age_Group that labels customers as:

--• 'Youth' if Age < 30

--• 'Adult' if Age is between 30 and 59

--• 'Senior' if Age >= 60

--Expected output: Customer ID, Age, Age_Group

```
select customer_id,
      age,
      case
      when age < 30 then 'Youth'
      when age between 30 and 59 then 'Adult'
      when age >= 60 then 'Senior'
      end as Age_Group
   from practicals.practical_1.retail_sales;
```
