--task 1

CREATE DATABASE SALES\_DB;

CREATE SCHEMA SALES\_DB.RAW\_SCHEMA;

CREATE FILE FORMAT SALES\_DB.RAW\_SCHEMA.csv\_format

TYPE='CSV'

FIELD\_OPTIONALLY\_ENCLOSED\_BY='"'

SKIP\_HEADER=1;

CREATE STAGE SALES\_DB.RAW\_SCHEMA.sales\_stage

FILE\_FORMAT = (FORMAT\_NAME = 'csv\_format');

LIST @SALES\_DB.RAW\_SCHEMA.sales\_stage;

--task 2

CREATE OR REPLACE TABLE SALES\_DB.RAW\_SCHEMA.sales\_raw(

transaction\_id STRING,

region STRING,

country STRING,

product STRING,

customer STRING,

sales\_rep STRING,

transaction\_date STRING,

timestamp VARCHAR(30),

quantity STRING,

unit\_price STRING,

total\_amount NUMBER,

order\_status STRING,

payment\_method STRING,

product\_details VARIANT,

customer\_info VARIANT

);

COPY INTO SALES\_DB.RAW\_SCHEMA.sales\_raw

FROM @sales\_stage/sales\_data\_dirty.csv

FILE\_FORMAT=(FORMAT\_NAME = 'csv\_format')

ON\_ERROR='CONTINUE';

--task 3

SELECT COUNT(\*) AS TOTAL,COUNT(DISTINCT transaction\_id) AS DIST\_ID FROM SALES\_DB.RAW\_SCHEMA.sales\_raw;

SELECT COUNT(\*) FROM SALES\_DB.RAW\_SCHEMA.sales\_raw

WHERE TRANSACTION\_ID IS NULL OR

UNIT\_PRICE IS NULL OR

REGION IS NULL OR

COUNTRY IS NULL OR

PRODUCT IS NULL OR

CUSTOMER IS NULL OR

SALES\_REP IS NULL OR

transaction\_date IS NULL OR

timestamp IS NULL OR

quantity IS NULL OR

total\_amount IS NULL OR

order\_status IS NULL OR

payment\_method IS NULL OR

product\_details IS NULL OR

customer\_info IS NULL;

select count (\*) as negative\_values from SALES\_DB.RAW\_SCHEMA.sales\_raw

where try\_to\_number(quantity)<0 or

try\_to\_number(total\_amount)<0 or

try\_to\_number(unit\_price) <0;

--task 4 and 5

create schema sales\_db.clean\_schema;

CREATE OR REPLACE TABLE SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN AS

SELECT

transaction\_id,

region,

country,

product,

customer,

sales\_rep,

TRY\_TO\_DATE(transaction\_date, 'DD-MM-YYYY') AS transaction\_date,

TRY\_TO\_TIMESTAMP\_NTZ(timestamp, 'DD-MM-YYYY HH24:MI') AS transaction\_ts,

CASE

WHEN TRY\_TO\_NUMBER(quantity) < 0 THEN 0

WHEN TRY\_TO\_NUMBER(quantity) IS NULL THEN 0

ELSE TRY\_TO\_NUMBER(quantity)

END AS quantity,

CASE

WHEN TRY\_TO\_NUMBER(unit\_price) < 0 THEN 0

ELSE TRY\_TO\_NUMBER(unit\_price)

END AS unit\_price,

CASE

WHEN TRY\_TO\_NUMBER(total\_amount) < 0 THEN 0

ELSE TRY\_TO\_NUMBER(total\_amount)

END AS total\_amount,

order\_status,

payment\_method,

product\_details:category::STRING AS product\_category,

product\_details:subcategory::STRING AS product\_subcategory,

product\_details:brand::STRING AS product\_brand,

product\_details:specs::STRING AS product\_specs,

CASE

WHEN TRY\_TO\_NUMBER(product\_details:ratings::STRING) < 0 THEN 0

ELSE TRY\_TO\_NUMBER(product\_details:ratings::STRING)

END AS product\_ratings,

customer\_info:segment::STRING AS customer\_segment,

customer\_info:demographics::STRING AS customer\_demographics,

customer\_info:preferences::STRING AS customer\_preferences

FROM SALES\_DB.RAW\_SCHEMA.sales\_raw;

DELETE FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN

WHERE TRANSACTION\_ID IS NULL OR

QUANTITY IS NULL OR

TOTAL\_AMOUNT IS NULL OR

UNIT\_PRICE IS NULL OR

UNIT\_PRICE=0 OR

QUANTITY=0 OR

customer\_segment IS NULL OR

product\_brand IS NULL;

SELECT \* FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN;

--task 6

ALTER TABLE SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN

ADD COLUMN PROFIT\_MARGIN NUMBER;

ALTER TABLE SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN

ADD COLUMN SALES\_QUARTER STRING;

UPDATE SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN

SET profit\_margin = total\_amount - (unit\_price \* quantity),

sales\_quarter = 'Q' ||CAST(EXTRACT(QUARTER FROM TRANSACTION\_DATE)AS STRING)|| '-' || TO\_CHAR(transaction\_date, 'YYYY');

select \* from clean\_schema.sales\_clean;

--task 7

CREATE OR REPLACE TABLE SALES\_DB.CLEAN\_SCHEMA.DIM\_REGION AS

SELECT DISTINCT TRIM(region) AS REGIONS, country FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN;

CREATE OR REPLACE TABLE SALES\_DB.CLEAN\_SCHEMA.DIM\_PRODUCT AS

SELECT DISTINCT product, product\_category, product\_subcategory, product\_brand FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN;

CREATE OR REPLACE TABLE SALES\_DB.CLEAN\_SCHEMA.DIM\_CUSTOMER AS

SELECT DISTINCT customer, customer\_segment, customer\_demographics, customer\_preferences FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN;

CREATE OR REPLACE TABLE SALES\_DB.CLEAN\_SCHEMA.FACT\_SALES AS

SELECT

transaction\_id,

transaction\_date,

transaction\_ts,

customer,

product,

TRIM(region) AS REGIONS,

country,

sales\_rep,

quantity,

unit\_price,

total\_amount,

profit\_margin,

sales\_quarter,

order\_status,

payment\_method,

product\_ratings

FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN;

SELECT \* FROM SALES\_DB.CLEAN\_SCHEMA.FACT\_SALES;

SELECT \* FROM SALES\_DB.CLEAN\_SCHEMA.DIM\_CUSTOMER;

SELECT \* FROM SALES\_DB.CLEAN\_SCHEMA.DIM\_PRODUCT;

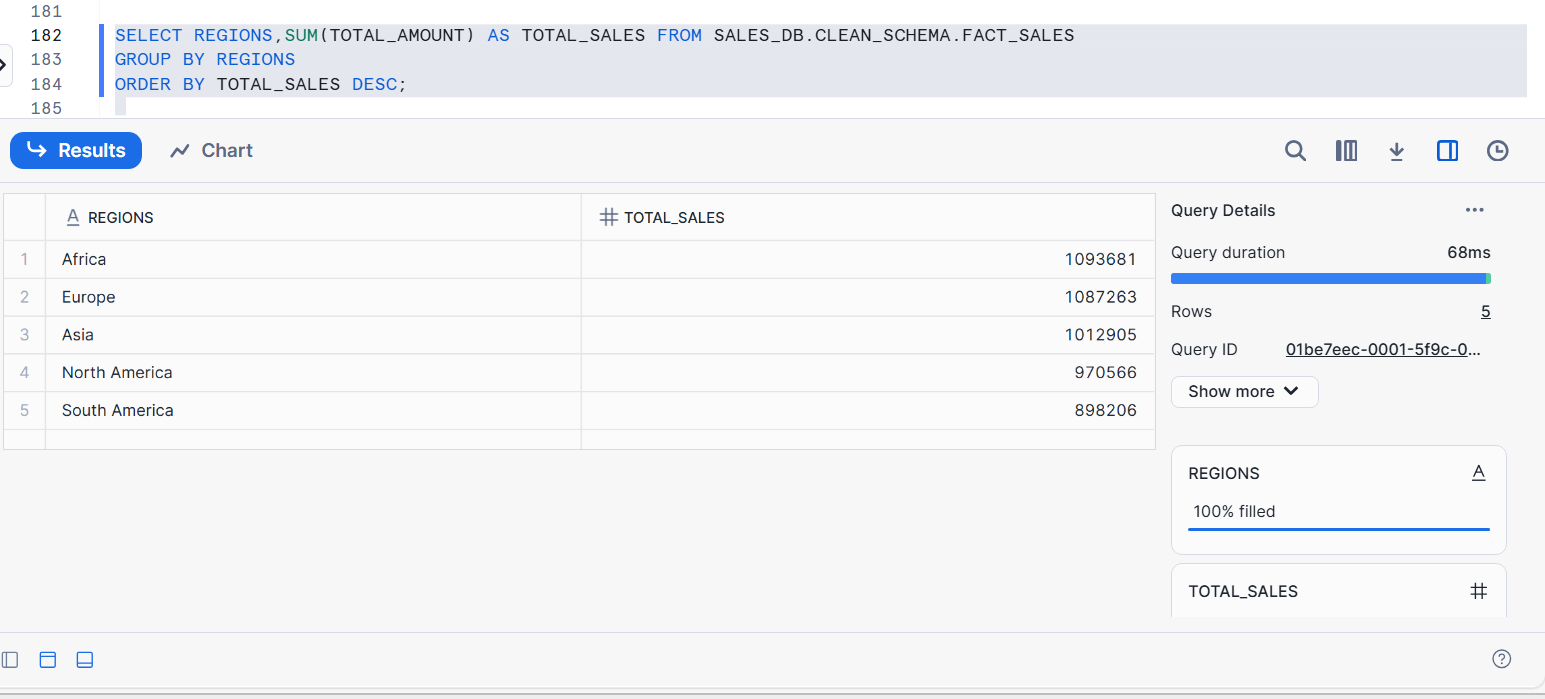
SELECT \* FROM SALES\_DB.CLEAN\_SCHEMA.DIM\_REGION;

--task 8

SELECT REGIONS,SUM(TOTAL\_AMOUNT) AS TOTAL\_SALES FROM SALES\_DB.CLEAN\_SCHEMA.FACT\_SALES

GROUP BY REGIONS

ORDER BY TOTAL\_SALES DESC;

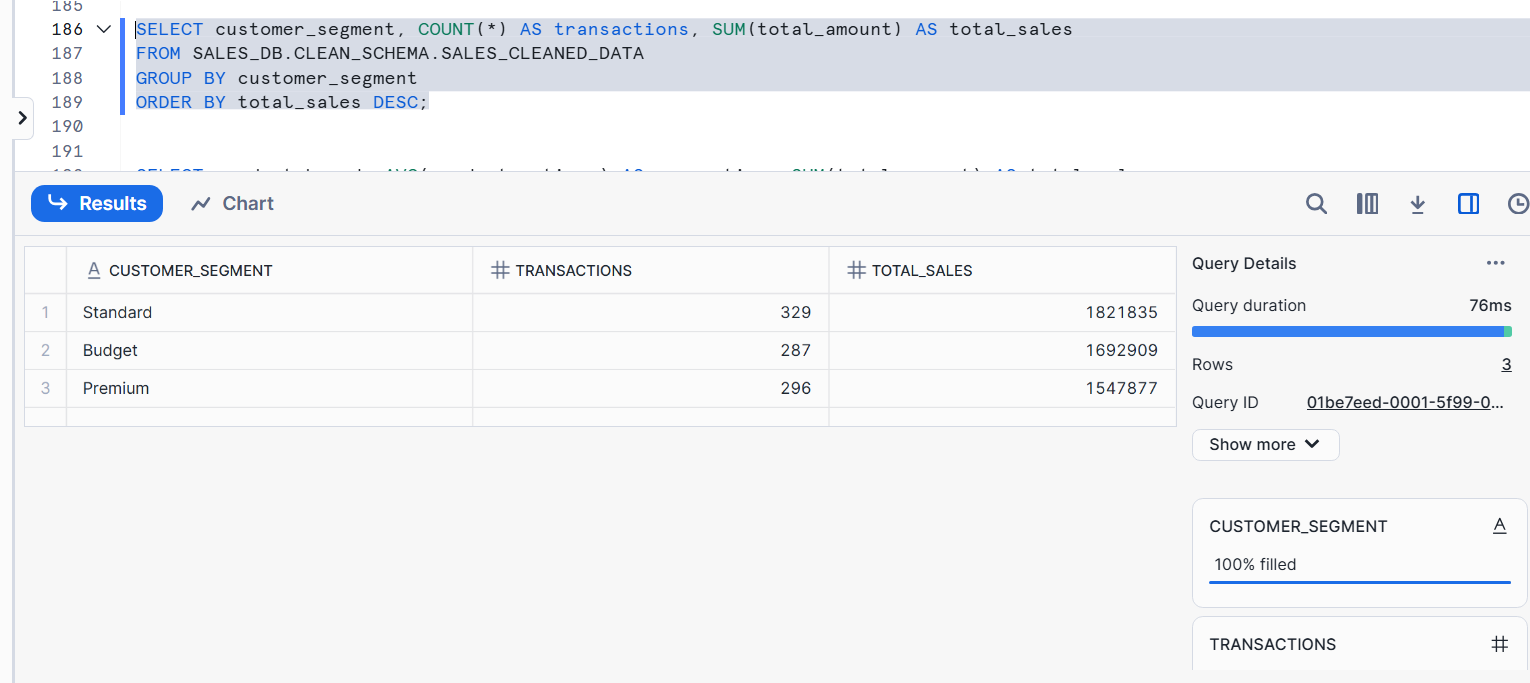


SELECT customer\_segment, COUNT(\*) AS transactions, SUM(total\_amount) AS total\_sales

FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN

GROUP BY customer\_segment

ORDER BY total\_sales DESC;



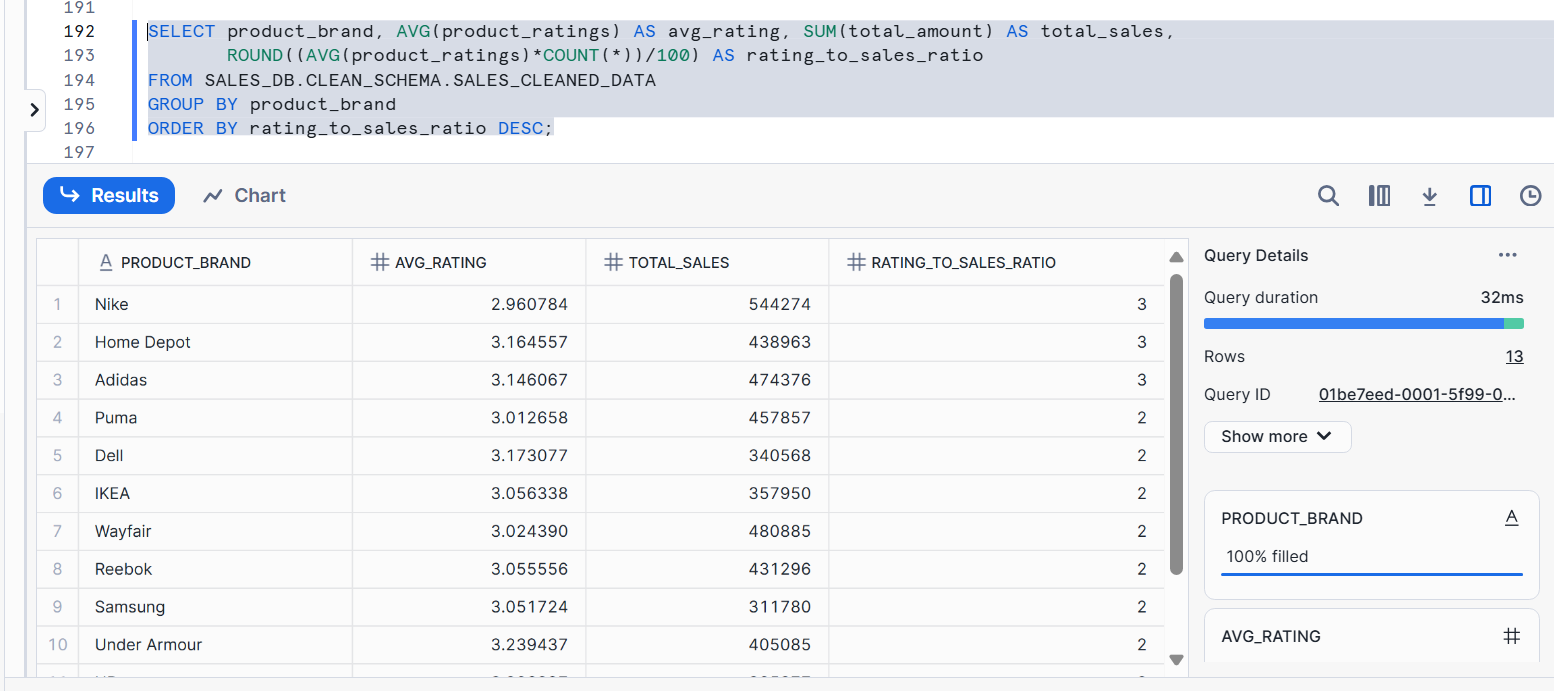
SELECT product\_brand, AVG(product\_ratings) AS avg\_rating, SUM(total\_amount) AS total\_sales,

ROUND((AVG(product\_ratings)\*COUNT(\*))/100) AS rating\_to\_sales\_ratio

FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEAN

GROUP BY product\_brand

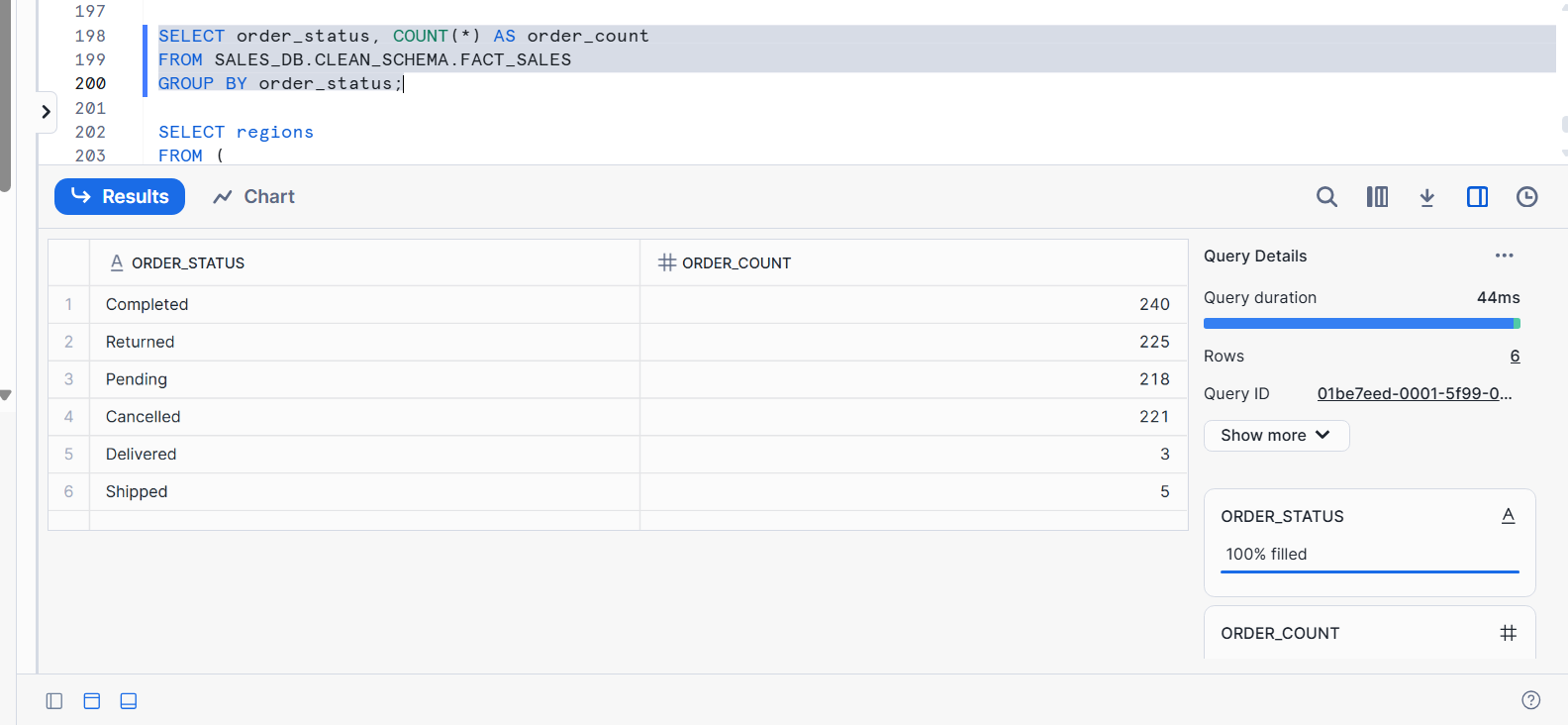
ORDER BY rating\_to\_sales\_ratio DESC;



SELECT order\_status, COUNT(\*) AS order\_count

FROM SALES\_DB.CLEAN\_SCHEMA.FACT\_SALES

GROUP BY order\_status;



SELECT regions

FROM (

SELECT regions, SUM(total\_amount) AS total\_sales

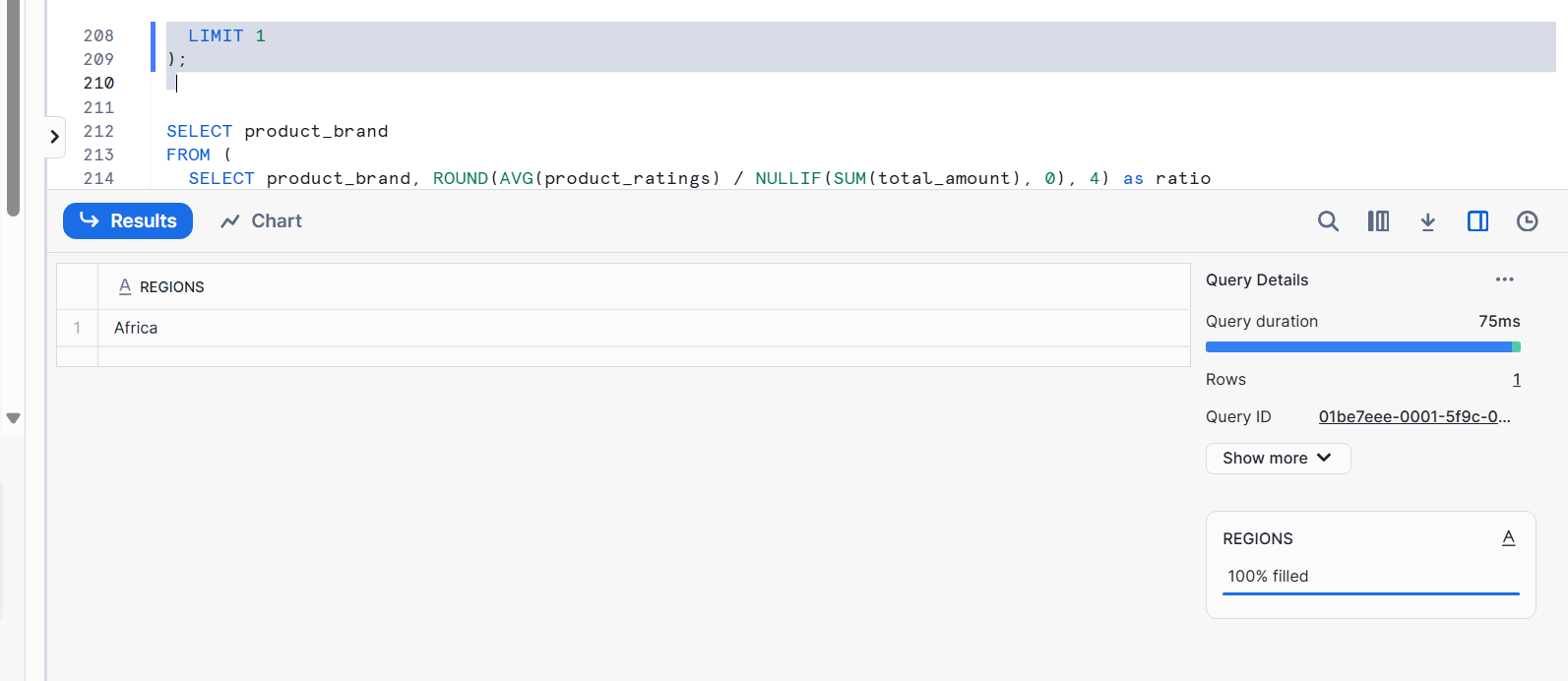
FROM SALES\_DB.CLEAN\_SCHEMA.FACT\_SALES

GROUP BY regions

ORDER BY total\_sales DESC

LIMIT 1

);



SELECT product\_brand

FROM (

SELECT product\_brand, ROUND((AVG(product\_ratings / NULLIF(SUM(total\_amount), 0), 4))) as ratio

FROM SALES\_DB.CLEAN\_SCHEMA.DIM\_PRODUCT

GROUP BY product\_brand

ORDER BY ratio DESC

LIMIT 1

);

