Captsone Document

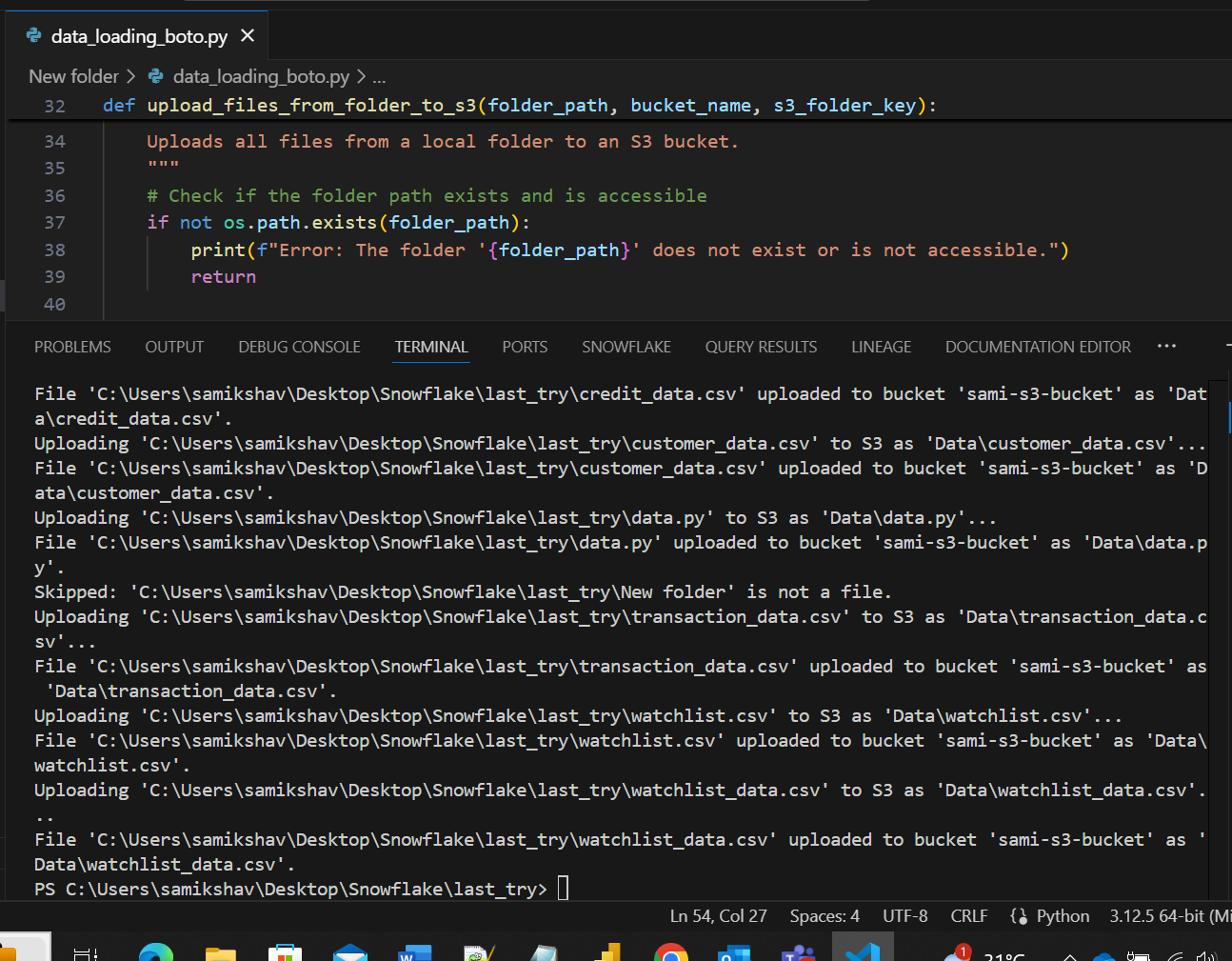
**Project Title**- Advanced ETL Pipeline for Real-Time Fraud Detection and Customer 360 in Global Banking

**#Create Data using Faker Library(data.py)**

Code has been saved in data.py file

**#Created python script which will execute data.py and save data to AWS bucket.**

Code has been saved in data\_loading\_boto.py



1. **Data Ingestion and Storage**:

**A-**Create database and schema

create or replace database GLOBALBANK;

create or replace schema GLOBALBANK.raw\_data;

create or replace schema GLOBALBANK.transform\_data;

create or replace schema GLOBALBANK.analytics;

create or replace schema GLOBALBANK.security;

**B**- Create Tables

---Create transaction table

create or replace table GLOBALBANK.raw\_data.transacation\_raw(

transaction\_id STRING,

customer\_id STRING,

transaction\_date TIMESTAMP\_NTZ,

amount FLOAT,

currency STRING,

transaction\_type STRING,

channel STRING,

merchant\_name STRING,

merchant\_category STRING,

location\_country STRING,

location\_city STRING,

is\_flagged BOOLEAN

);

---Create customer table

create or replace table GLOBALBANK.raw\_data.customer\_raw(

customer\_id STRING,

first\_name STRING,

last\_name STRING,

date\_of\_birth DATE,

gender STRING,

email STRING,

phone\_number STRING,

address STRING,

city STRING,

country STRING,

occupation STRING,

income\_bracket STRING,

customer\_since DATE

);

---Create Account table

create or replace table GLOBALBANK.raw\_data.account\_raw(

account\_id STRING,

customer\_id STRING,

account\_type STRING,

account\_status STRING,

open\_date DATE,

current\_balance FLOAT,

currency STRING,

credit\_limit FLOAT

);

---Create Credit table

create or replace table GLOBALBANK.raw\_data.credit\_raw(

customer\_id STRING,

credit\_score INT,

number\_of\_credit\_accounts INT,

total\_credit\_limit FLOAT,

total\_credit\_used FLOAT,

number\_of\_late\_payments INT,

bankruptcies INT

);

---Create Watchlist

create or replace table GLOBALBANK.raw\_data.watchlist(

entity\_id STRING,

entity\_name STRING,

entity\_type STRING,

risk\_category STRING,

listed\_date DATE,

source STRING);

**C- Create Storage Integration**

create or replace STORAGE INTEGRATION my\_s3\_integration

type = External\_stage

storage\_provider = s3

enabled = true

storage\_aws\_role\_arn = 'arn:aws:iam::975050369468:role/sami-new-role'

storage\_allowed\_locations = ('s3://sami-s3-bucket/Data/');

desc STORAGE INTEGRATION my\_s3\_integration;

**D-Create File Format**

create or replace file format MY\_CSV\_FORMAT

type = 'CSV'

FIELD\_OPTIONALLY\_ENCLOSED\_BY = '"'

SKIP\_HEADER = 1;

**E- Create Stage**

create or replace stage GLOBALBANK.raw\_data.my\_s3\_stage

STORAGE\_INTEGRATION = my\_s3\_integration

URL = 's3://sami-s3-bucket/Data/'

file\_format = MY\_CSV\_FORMAT;

list @GLOBALBANK.raw\_data.my\_s3\_stage;

**F-Create Stream**

-- Stream for transaction\_raw table

CREATE OR REPLACE STREAM GLOBALBANK.raw\_data.transaction\_raw\_stream ON TABLE GLOBALBANK.raw\_data.transacation\_raw;

-- Stream for customer\_raw table

CREATE OR REPLACE STREAM GLOBALBANK.raw\_data.customer\_raw\_stream ON TABLE GLOBALBANK.raw\_data.customer\_raw;

-- Stream for account\_raw table

CREATE OR REPLACE STREAM GLOBALBANK.raw\_data.account\_raw\_stream ON TABLE GLOBALBANK.raw\_data.account\_raw;

-- Stream for credit\_raw table

CREATE OR REPLACE STREAM GLOBALBANK.raw\_data.credit\_raw\_stream ON TABLE GLOBALBANK.raw\_data.credit\_raw;

-- Stream for watchlist table

CREATE OR REPLACE STREAM GLOBALBANK.raw\_data.watchlist\_stream ON TABLE GLOBALBANK.raw\_data.watchlist;

**G-Create Task**

----transaction task

CREATE OR REPLACE TASK GLOBALBANK.raw\_data.transactions\_tsk

WAREHOUSE = compute\_wh

SCHEDULE = '1 minute'

WHEN system$stream\_has\_data('GLOBALBANK.raw\_data.transaction\_raw\_stream')

AS

INSERT INTO GLOBALBANK.raw\_data.transacation\_raw

(transaction\_id,customer\_id,transaction\_date,amount,currency ,transaction\_type ,channel ,merchant\_name ,merchant\_category ,location\_country ,location\_city ,is\_flagged)

SELECT transaction\_id,customer\_id,transaction\_date,amount,currency ,transaction\_type ,channel ,merchant\_name ,merchant\_category ,location\_country ,location\_city ,is\_flagged

FROM GLOBALBANK.raw\_data.transaction\_raw\_stream;

---customer task

CREATE OR REPLACE TASK GLOBALBANK.raw\_data.customer\_tsk

WAREHOUSE = compute\_wh

SCHEDULE = '1 minute'

WHEN system$stream\_has\_data('GLOBALBANK.raw\_data.customer\_raw\_stream')

AS

INSERT INTO GLOBALBANK.raw\_data.customer\_raw

(customer\_id,first\_name,last\_name,date\_of\_birth,gender,email,phone\_number,address,city,country,occupation,income\_bracket,customer\_since)

SELECT customer\_id,first\_name,last\_name,date\_of\_birth,gender,email,phone\_number,address,city,country,occupation,income\_bracket,customer\_since

FROM GLOBALBANK.raw\_data.customer\_raw\_stream;

---account task

CREATE OR REPLACE TASK GLOBALBANK.raw\_data.account\_tsk

WAREHOUSE = compute\_wh

SCHEDULE = '1 minute'

WHEN system$stream\_has\_data('GLOBALBANK.raw\_data.account\_raw\_stream')

AS

INSERT INTO GLOBALBANK.raw\_data.account\_raw

(account\_id,customer\_id,account\_type,account\_status,open\_date,current\_balance,currency,credit\_limit)

SELECT account\_id,customer\_id,account\_type,account\_status,open\_date,current\_balance,currency,credit\_limit

FROM GLOBALBANK.raw\_data.account\_raw\_stream;

----credit task

CREATE OR REPLACE TASK GLOBALBANK.raw\_data.credit\_tsk

WAREHOUSE = compute\_wh

SCHEDULE = '1 minute'

WHEN system$stream\_has\_data('GLOBALBANK.raw\_data.credit\_raw\_stream')

AS

INSERT INTO GLOBALBANK.raw\_data.credit\_raw

(customer\_id,credit\_score,number\_of\_credit\_accounts,total\_credit\_limit,total\_credit\_used,number\_of\_late\_payments,bankruptcies)

SELECT customer\_id,credit\_score,number\_of\_credit\_accounts,total\_credit\_limit,total\_credit\_used,number\_of\_late\_payments,bankruptcies

FROM GLOBALBANK.raw\_data.credit\_raw\_stream;

---watchlist task

CREATE OR REPLACE TASK GLOBALBANK.raw\_data.watchlist\_tsk

WAREHOUSE = compute\_wh

SCHEDULE = '1 minute'

WHEN system$stream\_has\_data('GLOBALBANK.raw\_data.watchlist\_stream')

AS

INSERT INTO GLOBALBANK.raw\_data.watchlist

(entity\_id,entity\_name,entity\_type,risk\_category,listed\_date,source)

SELECT entity\_id,entity\_name,entity\_type,risk\_category,listed\_date,source

FROM GLOBALBANK.raw\_data.watchlist\_stream;

ALTER TASK GLOBALBANK.raw\_data.transactions\_tsk RESUME;

ALTER TASK GLOBALBANK.raw\_data.customer\_tsk RESUME;

ALTER TASK GLOBALBANK.raw\_data.account\_tsk RESUME;

ALTER TASK GLOBALBANK.raw\_data.credit\_tsk RESUME;

ALTER TASK GLOBALBANK.raw\_data.watchlist\_tsk RESUME;

**H-Create Snowpipe**

**---Create pipe for Transaction data**

create or replace pipe transacation\_raw\_pipe

auto\_ingest = true

as

COPY INTO GLOBALBANK.raw\_data.transacation\_raw

FROM @GLOBALBANK.raw\_data.my\_s3\_stage/transaction\_data.csv

FILE\_FORMAT = MY\_CSV\_FORMAT

ON\_ERROR = 'CONTINUE';

alter pipe transacation\_raw\_pipe refresh;

select SYSTEM$PIPE\_STATUS( 'transacation\_raw\_pipe' );

**--Create pipe for customer data**

create or replace pipe customer\_raw\_pipe

auto\_ingest = true

as

COPY INTO GLOBALBANK.raw\_data.customer\_raw

FROM @GLOBALBANK.raw\_data.my\_s3\_stage/customer\_data.csv

FILE\_FORMAT = MY\_CSV\_FORMAT

ON\_ERROR = 'CONTINUE';

alter pipe customer\_raw\_pipe refresh;

**---Create pipe for Account data**

create or replace pipe account\_raw\_pipe

auto\_ingest = true

as

COPY INTO GLOBALBANK.raw\_data.account\_raw

FROM @GLOBALBANK.raw\_data.my\_s3\_stage/account\_data.csv

FILE\_FORMAT = (FORMAT\_NAME = MY\_CSV\_FORMAT)

ON\_ERROR = 'CONTINUE';

alter pipe account\_raw\_pipe refresh;

**---Create pipe on Credit data**

create or replace pipe credit\_raw\_pipe

auto\_ingest = true

as

COPY INTO GLOBALBANK.raw\_data.credit\_raw

FROM @GLOBALBANK.raw\_data.my\_s3\_stage/credit\_data.csv

FILE\_FORMAT = (FORMAT\_NAME = MY\_CSV\_FORMAT)

ON\_ERROR = 'CONTINUE';

alter pipe credit\_raw\_pipe refresh;

**---Create pipe on Watchlist data**

create or replace pipe watchlist\_raw\_pipe

auto\_ingest = true

as

COPY INTO GLOBALBANK.raw\_data.watchlist

FROM @GLOBALBANK.raw\_data.my\_s3\_stage/watchlist\_data.csv

FILE\_FORMAT = (FORMAT\_NAME = MY\_CSV\_FORMAT)

ON\_ERROR = 'CONTINUE';

alter pipe watchlist\_raw\_pipe refresh;

select \* from transacation\_raw;

A screenshot of a computer

Description automatically generated

select \* from customer\_raw;

A screenshot of a computer

Description automatically generated

select \* from account\_raw;

A screenshot of a computer

Description automatically generated

select \* from credit\_raw;

A screenshot of a computer

Description automatically generated

select \* from watchlist;

A screenshot of a computer

Description automatically generated

DBT model tables.

A screenshot of a computer

Description automatically generated

Models are saved in DBT\_model section in github with .sql extension.

select \* from PC\_DBT\_DB.DBT\_SDHADAM.customers\_base;

A screenshot of a computer

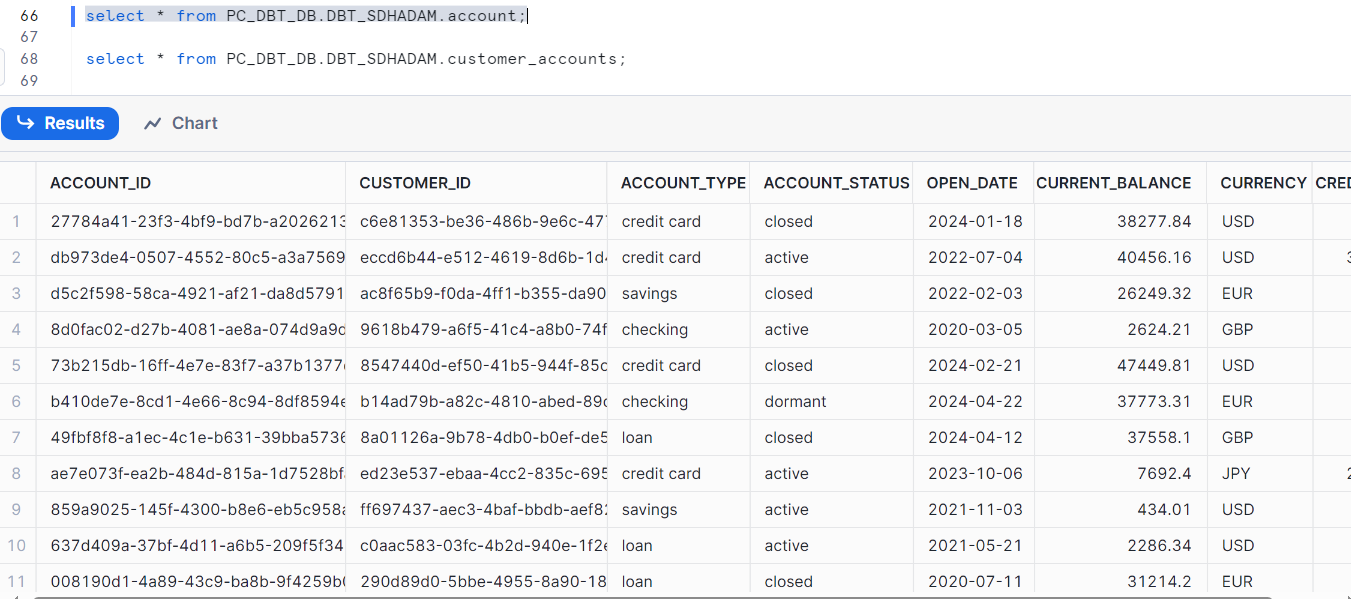
Description automatically generated

select \* from PC\_DBT\_DB.DBT\_SDHADAM.transactions;

A screenshot of a computer

Description automatically generated

select \* from PC\_DBT\_DB.DBT\_SDHADAM.account;



select \* from PC\_DBT\_DB.DBT\_SDHADAM.customer\_accounts;

A screenshot of a computer

Description automatically generated

select \* from PC\_DBT\_DB.DBT\_SDHADAM.customer\_spending;

A screenshot of a computer

Description automatically generated

select \* from PC\_DBT\_DB.DBT\_SDHADAM.customer360;

A screenshot of a computer

Description automatically generated

A barcode with blue lines

Description automatically generated

**Fraud monitoring**  
  
CREATE OR REPLACE VIEW FRAUD\_MONITORING\_VIEW AS

SELECT

CUSTOMER\_ID,

TOTAL\_SPENT,

TOTAL\_TRANSACTIONS,

LAST\_TRANSACTION\_DATE,

CASE

WHEN TOTAL\_SPENT > 10000 THEN 'High Risk'

WHEN TOTAL\_SPENT > 5000 AND TOTAL\_SPENT <= 10000 THEN 'Medium Risk'

ELSE 'Low Risk'

END AS RISK\_LEVEL

FROM CUSTOMER360

ORDER BY TOTAL\_SPENT DESC;

select \* from FRAUD\_MONITORING\_VIEW;

A screenshot of a computer screen

Description automatically generated

SELECT MIN(TOTAL\_SPENT) AS MIN\_SPENT, MAX(TOTAL\_SPENT) AS MAX\_SPENT

FROM CUSTOMER360;

A screenshot of a computer

Description automatically generated

SELECT

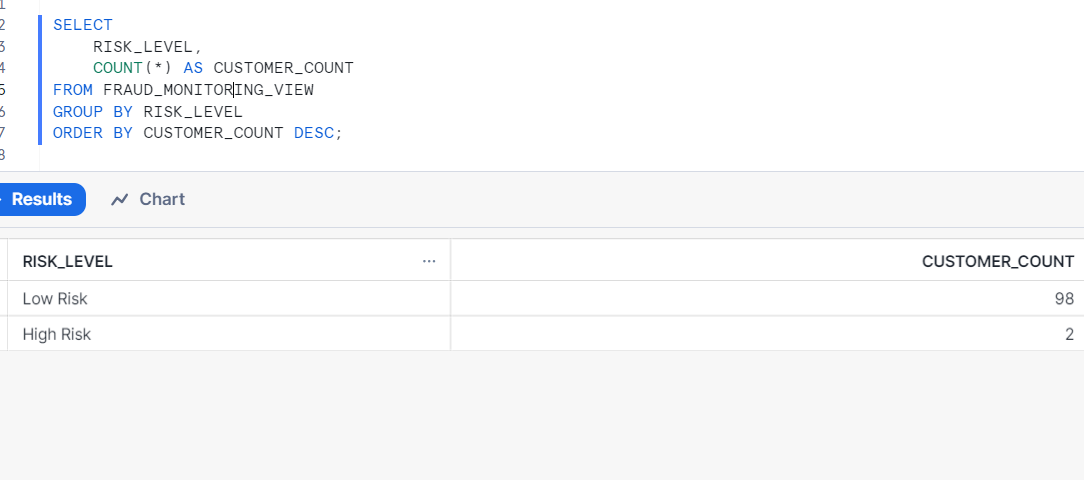
RISK\_LEVEL,

COUNT(\*) AS CUSTOMER\_COUNT

FROM FRAUD\_MONITORING\_VIEW

GROUP BY RISK\_LEVEL

ORDER BY CUSTOMER\_COUNT DESC;



SELECT

MIN(TOTAL\_SPENT) AS MIN\_SPENT,

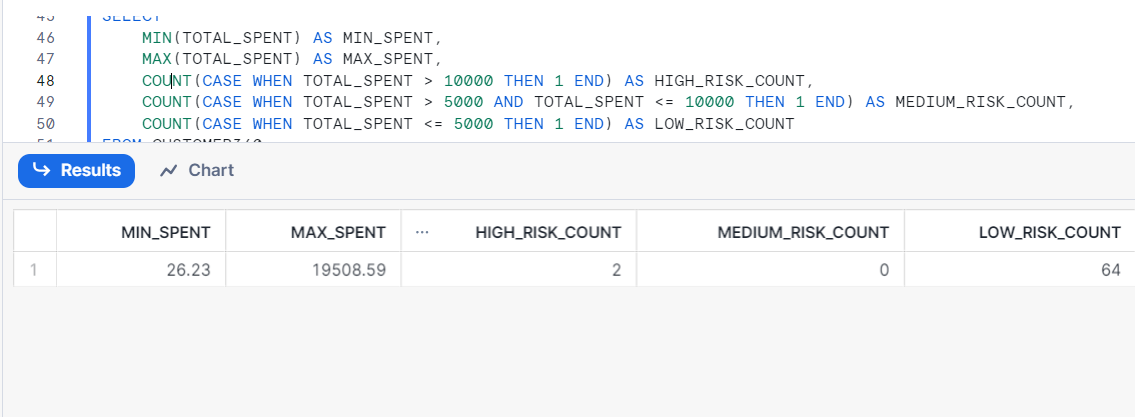
MAX(TOTAL\_SPENT) AS MAX\_SPENT,

COUNT(CASE WHEN TOTAL\_SPENT > 10000 THEN 1 END) AS HIGH\_RISK\_COUNT,

COUNT(CASE WHEN TOTAL\_SPENT > 5000 AND TOTAL\_SPENT <= 10000 THEN 1 END) AS MEDIUM\_RISK\_COUNT,

COUNT(CASE WHEN TOTAL\_SPENT <= 5000 THEN 1 END) AS LOW\_RISK\_COUNT

FROM CUSTOMER360;



SELECT MIN(TOTAL\_SPENT) AS MIN\_SPENT, MAX(TOTAL\_SPENT) AS MAX\_SPENT

FROM CUSTOMER360;

SELECT

RISK\_LEVEL,

COUNT(\*) AS CUSTOMER\_COUNT

FROM FRAUD\_MONITORING\_VIEW

GROUP BY RISK\_LEVEL

ORDER BY CUSTOMER\_COUNT DESC;

CREATE OR REPLACE VIEW FRAUD\_MONITORING\_VIEW AS

SELECT

CUSTOMER\_ID,

TOTAL\_SPENT,

TOTAL\_TRANSACTIONS,

LAST\_TRANSACTION\_DATE,

CASE

WHEN TOTAL\_SPENT > 10000 THEN 'High Risk'

WHEN TOTAL\_SPENT > 2000 AND TOTAL\_SPENT <= 10000 THEN 'Medium Risk'

ELSE 'Low Risk'

END AS RISK\_LEVEL

FROM CUSTOMER360

ORDER BY TOTAL\_SPENT DESC;

SELECT

MIN(TOTAL\_SPENT) AS MIN\_SPENT,

MAX(TOTAL\_SPENT) AS MAX\_SPENT,

COUNT(CASE WHEN TOTAL\_SPENT > 10000 THEN 1 END) AS HIGH\_RISK\_COUNT,

COUNT(CASE WHEN TOTAL\_SPENT > 5000 AND TOTAL\_SPENT <= 10000 THEN 1 END) AS MEDIUM\_RISK\_COUNT,

COUNT(CASE WHEN TOTAL\_SPENT <= 5000 THEN 1 END) AS LOW\_RISK\_COUNT

FROM CUSTOMER360;