-- TASK - 1: SETUP DATABASE AND SCHEMA

CREATE OR REPLACE DATABASE SALES\_DB;

CREATE OR REPLACE SCHEMA RAW\_SCHEMA;

CREATE OR REPLACE SCHEMA FILE\_FORMATS;

CREATE OR REPLACE STAGE SALES\_STAGE;

CREATE OR REPLACE file format SALES\_DB.FILE\_FORMATS.CSVFORMAT

TYPE = CSV

FIELD\_OPTIONALLY\_ENCLOSED\_BY='"'

SKIP\_HEADER=1;

-- TASK - 2: LOAD RAW DATA

CREATE OR REPLACE TABLE SALES\_RAW (

TRANSACTION\_ID VARCHAR(100),

REGION VARCHAR(100),

COUNTRY VARCHAR(100),

PRODUCT VARCHAR(100),

CUSTOMER VARCHAR(100),

SALES\_REP VARCHAR(100),

TRANSACTION\_DATE VARCHAR(100),

TIMESTAMP VARCHAR(100),

QUANTITY VARCHAR(100),

UNIT\_PRICE VARCHAR(100),

TOTAL\_AMOUNT VARCHAR(100),

ORDER\_STATUS VARCHAR(20),

PAYMENT\_METHOD VARCHAR(100),

PRODUCT\_DETAILS STRING,

CUSTOMER\_INFO STRING

);

-- TO CHECK THE ERRORS BEFORE COPYING THE DATA.

COPY INTO RAW\_SCHEMA.SALES\_RAW

FROM @RAW\_SCHEMA.SALES\_STAGE

FILE\_FORMAT = (FORMAT\_NAME = FILE\_FORMATS.CSVFORMAT)

VALIDATION\_MODE = 'RETURN\_ALL\_ERRORS';

-- COPYING THE DATA

COPY INTO RAW\_SCHEMA.SALES\_RAW

FROM @RAW\_SCHEMA.SALES\_STAGE

FILE\_FORMAT = (FORMAT\_NAME = FILE\_FORMATS.CSVFORMAT)

ON\_ERROR = CONTINUE;

SELECT \* FROM RAW\_SCHEMA.SALES\_RAW;

-- TASK - 3: COUNT OF TOTAL VS DISTINCT TRANSACTIONS , NULL VALUES,NEGATIVE COUNT

SELECT COUNT(\*) AS TOTAL\_TRANSACTIONS

FROM RAW\_SCHEMA.SALES\_RAW;

SELECT COUNT(DISTINCT TRANSACTION\_ID) AS DISTINCT\_TRANSACTIONS

FROM RAW\_SCHEMA.SALES\_RAW;

SELECT COUNT(\*) AS NEGATIVE\_COUNT

FROM RAW\_SCHEMA.SALES\_RAW

WHERE UNIT\_PRICE < 0;

SELECT COUNT(\*) AS COUNT\_OF\_NULL\_VALUES FROM RAW\_SCHEMA.SALES\_RAW

WHERE TRANSACTION\_ID 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

UNIT\_PRICE 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;

-- TASK - 4: FLATTEN JSON DATA

CREATE OR REPLACE TABLE CLEAN\_SCHEMA.SALES\_FLATTENED AS

SELECT

TRANSACTION\_ID,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):category::STRING AS PRODUCT\_CATEGORY,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):subcategory::STRING AS PRODUCT\_SUBCATEGORY,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):brand::STRING AS PRODUCT\_BRAND,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):specs::STRING AS PRODUCT\_SPECS,

TRY\_PARSE\_JSON(CUSTOMER\_INFO):segment::STRING AS CUSTOMER\_SEGMENT,

TRY\_PARSE\_JSON(CUSTOMER\_INFO):demographics::STRING AS CUSTOMER\_DEMOGRAPHICS,

TRY\_PARSE\_JSON(CUSTOMER\_INFO):preferences::STRING AS CUSTOMER\_PREFERENCES

FROM RAW\_SCHEMA.SALES\_RAW;

SELECT \* FROM CLEAN\_SCHEMA.SALES\_FLATTENED;

-- CHECKING FOR THE FLATTENED:

-- SELECT

-- TRANSACTION\_ID,

-- REGION,

-- COUNTRY,

-- PRODUCT,

-- CUSTOMER,

-- SALES\_REP,

-- TRANSACTION\_DATE,

-- TIMESTAMP,

-- QUANTITY,

-- UNIT\_PRICE,

-- TOTAL\_AMOUNT,

-- ORDER\_STATUS,

-- PAYMENT\_METHOD,

-- TRY\_PARSE\_JSON(PRODUCT\_DETAILS):category::STRING AS PRODUCT\_CATEGORY,

-- TRY\_PARSE\_JSON(PRODUCT\_DETAILS):subcategory::STRING AS PRODUCT\_SUBCATEGORY,

-- TRY\_PARSE\_JSON(PRODUCT\_DETAILS):brand::STRING AS PRODUCT\_BRAND,

-- TRY\_PARSE\_JSON(PRODUCT\_DETAILS):specs::STRING AS PRODUCT\_SPECS,

-- TRY\_PARSE\_JSON(PRODUCT\_DETAILS):ratings::INT AS PRODUCT\_RATINGS,

-- TRY\_PARSE\_JSON(CUSTOMER\_INFO):segment::STRING AS CUSTOMER\_SEGMENT,

-- TRY\_PARSE\_JSON(CUSTOMER\_INFO):demographics::STRING AS CUSTOMER\_DEMOGRAPHICS,

-- TRY\_PARSE\_JSON(CUSTOMER\_INFO):preferences::STRING AS CUSTOMER\_PREFERENCES

-- FROM RAW\_SCHEMA.SALES\_RAW limit 20;

-- TASK - 5:CLEAN AND TRANSFORM DATA

CREATE OR REPLACE SCHEMA CLEAN\_SCHEMA

CREATE OR REPLACE TABLE SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE AS

SELECT

TRANSACTION\_ID,

REGION,

COUNTRY,

PRODUCT,

CUSTOMER,

SALES\_REP,

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

TIMESTAMP,

TRY\_TO\_NUMBER(QUANTITY) AS QUANTITY,

TRY\_TO\_NUMBER(UNIT\_PRICE) AS UNIT\_PRICE,

TRY\_TO\_NUMBER(TOTAL\_AMOUNT) AS TOTAL\_AMOUNT,

ORDER\_STATUS,

PAYMENT\_METHOD,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):category::STRING AS PRODUCT\_CATEGORY,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):subcategory::STRING AS PRODUCT\_SUBCATEGORY,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):brand::STRING AS PRODUCT\_BRAND,

TRY\_PARSE\_JSON(PRODUCT\_DETAILS):specs::STRING AS PRODUCT\_SPECS,

TRY\_PARSE\_JSON(CUSTOMER\_INFO):segment::STRING AS CUSTOMER\_SEGMENT,

TRY\_PARSE\_JSON(CUSTOMER\_INFO):demographics::STRING AS CUSTOMER\_DEMOGRAPHICS,

TRY\_PARSE\_JSON(CUSTOMER\_INFO):preferences::STRING AS CUSTOMER\_PREFERENCES

FROM SALES\_DB.RAW\_SCHEMA.SALES\_RAW

WHERE TRANSACTION\_ID IS NOT NULL

OR REGION IS NOT NULL OR

COUNTRY IS NOT NULL OR

PRODUCT IS NOT NULL OR

CUSTOMER IS NOT NULL OR

SALES\_REP IS NOT NULL OR

TRANSACTION\_DATE IS NOT NULL OR

TIMESTAMP IS NOT NULL OR

QUANTITY IS NOT NULL OR

UNIT\_PRICE IS NOT NULL OR

TOTAL\_AMOUNT IS NOT NULL OR

ORDER\_STATUS IS NOT NULL OR

PAYMENT\_METHOD IS NOT NULL;

SELECT \* FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

-- DELETING THE NULL VALUES

DELETE FROM SALES\_DB.CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE

WHERE TRANSACTION\_ID 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

UNIT\_PRICE IS NULL OR

TOTAL\_AMOUNT IS NULL OR

ORDER\_STATUS IS NULL OR

PAYMENT\_METHOD IS NULL;

SELECT \* FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

-- TASK - 6: FEATURE ENGINEERING

ALTER TABLE CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE

ADD COLUMN PROFIT\_MARGIN FLOAT;

ALTER TABLE CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE

ADD COLUMN SALES\_QUARTER STRING;

UPDATE CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE

SET

PROFIT\_MARGIN = TOTAL\_AMOUNT \* 0.2,

SALES\_QUARTER = CASE

WHEN MONTH(TRANSACTION\_DATE) BETWEEN 1 AND 3 THEN 'QUARTER-1'

WHEN MONTH(TRANSACTION\_DATE) BETWEEN 4 AND 6 THEN 'QUARTER-2'

WHEN MONTH(TRANSACTION\_DATE) BETWEEN 7 AND 9 THEN 'QUARTER-3'

WHEN MONTH(TRANSACTION\_DATE) BETWEEN 10 AND 12 THEN 'QUARTER-4'

ELSE 'INVALID'

END;

SELECT \* FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

DESC TABLE CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

-- TASK - 7: CREATE FACT AND DIMENSION TABLES

CREATE OR REPLACE SCHEMA STAR\_SCHEMA;

-- FACT TABLES

CREATE OR REPLACE TABLE STAR\_SCHEMA\_SALES\_FACT\_TABLE AS

SELECT

TRANSACTION\_ID,

TRANSACTION\_DATE,

PRODUCT AS PRODUCT\_NAME,

REGION,

COUNTRY,

QUANTITY,

CUSTOMER,

SALES\_REP,

TIMESTAMP,

UNIT\_PRICE,

TOTAL\_AMOUNT,

ORDER\_STATUS,

PAYMENT\_METHOD

FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

SELECT \* FROM STAR\_SCHEMA\_SALES\_FACT\_TABLE;

-- DIMENSIONS TABLES

-- REGION DIMENSION TABLE

CREATE OR REPLACE TABLE STAR\_SCHEMA.DIM\_REGION\_TABLE

AS SELECT DISTINCT PRODUCT,REGION ,COUNTRY

FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

SELECT \* FROM STAR\_SCHEMA.DIM\_REGION\_TABLE;

-- PRODUCT DIMENSION TABLE

CREATE OR REPLACE TABLE STAR\_SCHEMA.DIM\_PRODUCT\_TABLE

AS SELECT

TRANSACTION\_ID,

PRODUCT\_CATEGORY,

PRODUCT\_SUBCATEGORY,

PRODUCT\_BRAND,

PRODUCT\_SPECS,

FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

SELECT \* FROM STAR\_SCHEMA.DIM\_PRODUCT\_TABLE;

-- CUSTOMER DIMENSION TABLE

CREATE OR REPLACE TABLE STAR\_SCHEMA.DIM\_CUSTOMER\_TABLE

AS SELECT

TRANSACTION\_ID,

CUSTOMER\_SEGMENT,

CUSTOMER\_DEMOGRAPHICS,

CUSTOMER\_PREFERENCES

FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

SELECT \* FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE;

-- TASK - 8: REPORTING

SELECT REGION, COUNTRY, SUM(TOTAL\_AMOUNT) AS SUM\_OF\_SALES

FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE

GROUP BY REGION, COUNTRY

ORDER BY SUM\_OF\_SALES DESC;

SELECT CUSTOMER\_SEGMENT, SUM(TOTAL\_AMOUNT) AS SUM\_OF\_SALES

FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE

GROUP BY CUSTOMER\_SEGMENT

ORDER BY SUM\_OF\_SALES DESC;

SELECT PRODUCT\_BRAND, SUM(TOTAL\_AMOUNT) AS SUM\_OF\_SALES

FROM CLEAN\_SCHEMA.SALES\_CLEANED\_TABLE

GROUP BY PRODUCT\_BRAND

ORDER BY SUM\_OF\_SALES DESC;

SELECT ORDER\_STATUS, COUNT(\*)AS ORDER\_COUNT

FROM STAR\_SCHEMA.STAR\_SCHEMA\_SALES\_FACT\_TABLE

GROUP BY ORDER\_STATUS

ORDER BY ORDER\_COUNT DESC;