SELECT \* FROM inventory.projectfinaldata;

-- remove duplicates

DELETE p

FROM inventory.projectfinaldata p

LEFT JOIN (

SELECT MIN(Patient\_ID) AS min\_rowid

FROM inventory.projectfinaldata

GROUP BY Typeofsales, Patient\_ID, Specialisation, Dept, Dateofbill, Quantity, ReturnQuantity, Final\_Cost, Final\_Sales, RtnMRP, Formulation, DrugName, SubCat, SubCat1

) t ON p.Patient\_ID = t.min\_rowid

WHERE t.min\_rowid IS NULL;

-- handle missing values

UPDATE inventory.projectfinaldata

SET Specialisation = 'Unknown'

WHERE Specialisation IS NULL;

UPDATE inventory.projectfinaldata

SET SubCat = 'Not Available'

WHERE SubCat IS NULL;

-- Repeat similar UPDATE statements for other columns as needed

-- normalize data

UPDATE inventory.projectfinaldata

SET Dateofbill =

CASE

WHEN LOCATE('-', Dateofbill) > 0 THEN Dateofbill

WHEN LOCATE('/', Dateofbill) > 0 THEN Dateofbill

ELSE NULL

END;

-- Removing Outliers

DELETE FROM inventory.projectfinaldata

WHERE Final\_Cost < 0 OR Final\_Cost > 1000000;

DELETE FROM inventory.projectfinaldata

WHERE Final\_Sales < 0 OR Final\_Sales > 1000000;

-- Removing Invalid Records

DELETE FROM inventory.projectfinaldata

WHERE Quantity < 0 OR ReturnQuantity < 0;

-- Calculate mean and variance

SELECT

AVG(Final\_Cost) AS mean\_final\_cost,

VARIANCE(Final\_Cost) AS variance\_final\_cost,

AVG(Final\_Sales) AS mean\_final\_sales,

VARIANCE(Final\_Sales) AS variance\_final\_sales,

AVG(ReturnQuantity) AS mean\_return\_quantity,

VARIANCE(ReturnQuantity) AS variance\_return\_quantity

FROM inventory.projectfinaldata;

-- EDA 1 Meadian Calculation

DELIMITER //

CREATE PROCEDURE CalculateMedian()

BEGIN

DECLARE total\_count INT;

DECLARE middle\_position INT;

DECLARE median\_value DECIMAL(10, 2);

SELECT COUNT(\*) INTO total\_count FROM inventory.projectfinaldata;

SET middle\_position = (total\_count + 1) DIV 2;

SELECT Final\_Sales INTO median\_value

FROM (

SELECT Final\_Sales, @rownum := @rownum + 1 AS rownum

FROM inventory.projectfinaldata, (SELECT @rownum := 0) r

ORDER BY Final\_Sales

) AS ranked

WHERE rownum = middle\_position;

SELECT

COUNT(\*) AS total\_records,

AVG(Final\_Sales) AS avg\_sales,

MIN(Final\_Sales) AS min\_sales,

MAX(Final\_Sales) AS max\_sales,

median\_value AS median\_sales,

STDDEV\_POP(Final\_Sales) AS std\_deviation\_sales

FROM inventory.projectfinaldata;

END //

DELIMITER ;

CALL CalculateMedian();

-- EDA 2 Visualization

-- Number of sales per specialization

SELECT Specialisation, COUNT(\*) AS sales\_count

FROM inventory.projectfinaldata

GROUP BY Specialisation

ORDER BY sales\_count DESC;

-- Daily sales trend

SELECT DATE(Dateofbill) AS bill\_date, SUM(Final\_Sales) AS daily\_sales

FROM inventory.projectfinaldata

GROUP BY bill\_date

ORDER BY bill\_date;

-- EDA 3 Transformation

-- Convert 'Quantity' and 'ReturnQuantity' to positive values

UPDATE inventory.projectfinaldata

SET Quantity = ABS(Quantity), ReturnQuantity = ABS(ReturnQuantity);

-- Calculate total sales for each drug category

SELECT SubCat, SUM(Final\_Sales) AS total\_sales

FROM inventory.projectfinaldata

GROUP BY SubCat

ORDER BY total\_sales DESC;

-- EDA 4 Interpretaion

-- Identify top-selling drugs and their sales

SELECT DrugName, SUM(Final\_Sales) AS total\_sales

FROM inventory.projectfinaldata

GROUP BY DrugName

ORDER BY total\_sales DESC

LIMIT 10;

-- Calculate return rate for each drug

SELECT DrugName, COUNT(\*) AS total\_sales,

SUM(CASE WHEN ReturnQuantity > 0 THEN 1 ELSE 0 END) AS total\_returns,

SUM(CASE WHEN ReturnQuantity > 0 THEN 1 ELSE 0 END) / COUNT(\*) AS return\_rate

FROM inventory.projectfinaldata

GROUP BY DrugName

ORDER BY return\_rate DESC;