**Final Project Retail Sales Performance**

use online\_retail;

DESCRIBE onlineretailnine;

DESCRIBE onlineretailten;

DESCRIBE onlineretailtransactions;

#Add Unique Identifier (if missing) — For Row-Level Ops

ALTER TABLE onlineretailnine ADD COLUMN id INT AUTO\_INCREMENT PRIMARY KEY;

#Removing Duplicates

WITH ranked\_rows AS (

SELECT

id,

ROW\_NUMBER() OVER (

PARTITION BY Invoice, StockCode, Description, Quantity, InvoiceDate, InvoiceTime, CustomerID, Country

ORDER BY id

) AS rn

FROM onlineretailnine

)

DELETE FROM onlineretailnine

WHERE id IN (

SELECT id FROM ranked\_rows WHERE rn > 1

);

# Remove Rows with NULL or Empty Critical Fields

DELETE FROM onlineretailnine

WHERE

CustomerID IS NULL

OR Invoice IS NULL

OR InvoiceDate IS NULL

OR InvoiceTime IS NULL

OR Quantity IS NULL

OR StockCode IS NULL

OR Description IS NULL

OR Country IS NULL

OR TRIM(Description) = '';

# Normalize Country and Description Text (Uppercase, Trim)

UPDATE onlineretailnine

SET

Country = UPPER(TRIM(Country)),

Description = UPPER(TRIM(Description));

#select \* from onlineretailnine;

# Remove Rows with Negative or Zero Quantity

DELETE FROM onlineretailnine

WHERE Quantity <= 0;

#Standardize Time Format (if stored as string)

ALTER TABLE onlineretailnine

MODIFY COLUMN InvoiceTime TIME;

#Trim and Clean StockCode (Remove junk characters)

UPDATE onlineretailnine

SET StockCode = TRIM(BOTH ' "' FROM StockCode)

WHERE StockCode IS NOT NULL;

#Filter Out Test or Dummy Data

DELETE FROM onlineretailnine

WHERE

Description LIKE '%TEST%'

OR Description LIKE '%SAMPLE%'

OR StockCode = 'M';

#Clean Record Count

SELECT COUNT(\*) AS clean\_rows

FROM onlineretailnine

WHERE

CustomerID IS NOT NULL

AND Quantity > 0

AND TRIM(Description) != ''

AND Invoice IS NOT NULL

AND Country IS NOT NULL;

#EDA

# Count Rows

SELECT COUNT(\*) AS total\_rows FROM onlineretailnine;

#Count NULLs per column

SELECT

SUM(CustomerID IS NULL) AS null\_customerid,

SUM(Invoice IS NULL) AS null\_invoice,

SUM(InvoiceDate IS NULL) AS null\_invoicedate,

SUM(InvoiceTime IS NULL) AS null\_invoicetime,

SUM(Quantity IS NULL) AS null\_quantity

FROM onlineretailnine;

#Unique values in categorical columns

SELECT COUNT(DISTINCT Country) AS unique\_countries FROM onlineretailnine;

SELECT COUNT(DISTINCT StockCode) AS unique\_stockcodes FROM onlineretailnine;

#Check for Duplicates

SELECT Invoice, StockCode, Description, Quantity, InvoiceDate, InvoiceTime, CustomerID, Country,

COUNT(\*) as count

FROM onlineretailnine

GROUP BY Invoice, StockCode, Description, Quantity, InvoiceDate, InvoiceTime, CustomerID, Country

HAVING count > 1;

#Quantity Distribution

SELECT Quantity, COUNT(\*) AS freq

FROM onlineretailnine

GROUP BY Quantity

ORDER BY freq DESC

LIMIT 10;

#Top Countries by Transactions

SELECT Country, COUNT(\*) AS transactions

FROM onlineretailnine

GROUP BY Country

ORDER BY transactions DESC;

#Most Common Products

SELECT Description, SUM(Quantity) AS total\_sold

FROM onlineretailnine

GROUP BY Description

ORDER BY total\_sold DESC

LIMIT 10;

#Daily Transaction Trend

SELECT InvoiceDate, COUNT(DISTINCT Invoice) AS num\_invoices

FROM onlineretailnine

GROUP BY InvoiceDate

ORDER BY InvoiceDate;

# Hourly Volume

SELECT HOUR(InvoiceTime) AS hour, COUNT(\*) AS transactions

FROM onlineretailnine

GROUP BY hour

ORDER BY hour;

SELECT

r.Invoice,

r.StockCode,

r.Description,

r.Quantity,

r.InvoiceDate,

r.`Invoice Time`,

r.`Customer ID`,

r.Country,

t.`Transaction Date`,

t.`Transaction Time`

FROM (

SELECT Invoice, StockCode, Description, Quantity, Invoice\_Date, `Invoice\_Time`, `Customer\_ID`, Country

FROM onlineretailnine

UNION ALL

SELECT Invoice, StockCode, Description, Quantity, Invoice\_Date, `Invoice\_Time`, `Customer\_ID`, Country

FROM onlineretailten

) AS r

LEFT JOIN onlineretailtransactions t

ON r.`Customer ID` = t.`Customer ID`

AND r.InvoiceDate = t.`Transaction Date`

AND r.`Invoice Time` = t.`Transaction Time`

LIMIT 1000;

#putting underscore instead of spaces in column names for all three tables

ALTER TABLE onlineretailnine

CHANGE `CustomerID` Customer\_ID INT,

CHANGE `InvoiceTime` Invoice\_Time TEXT,

CHANGE `InvoiceDate` Invoice\_Date TEXT;

ALTER TABLE onlineretailten

CHANGE `Invoice Time` Invoice\_Time TEXT,

CHANGE `Customer ID` Customer\_ID INT,

CHANGE `InvoiceDate` Invoice\_Date TEXT;

ALTER TABLE onlineretailtransactions

CHANGE `Transaction Date` Transaction\_Date TEXT,

CHANGE `Transaction Time` Transaction\_Time TEXT,

CHANGE `Customer ID` Customer\_ID INT;

#quey to join table ten and transaction

SELECT

r.Invoice,

r.StockCode,

r.Description,

r.Quantity,

r.Invoice\_Date,

r.Invoice\_Time,

r.Customer\_ID,

r.Country,

t.Transaction\_Date,

t.Transaction\_Time

FROM (

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

Invoice\_Time,

Customer\_ID,

Country

FROM onlineretailnine

UNION ALL

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

Invoice\_Time,

Customer\_ID,

Country

FROM onlineretailten

) AS r

LEFT JOIN onlineretailtransactions t

ON r.Customer\_ID = t.Customer\_ID

AND r.Invoice\_Date = t.Transaction\_Date

AND r.Invoice\_Time = t.Transaction\_Time

LIMIT 1000;

#

SELECT

r.Invoice,

r.StockCode,

r.Description,

r.Quantity,

r.Invoice\_Date,

r.Invoice\_Time,

r.Customer\_ID,

r.Country,

t.Transaction\_Date,

t.Transaction\_Time

FROM (

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

Invoice\_Time,

Customer\_ID,

Country

FROM onlineretailnine

UNION ALL

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

Invoice\_Time,

Customer\_ID,

Country

FROM onlineretailten

) AS r

LEFT JOIN onlineretailtransactions AS t

ON r.Customer\_ID = t.Customer\_ID

AND r.Invoice\_Date = t.Transaction\_Date

AND r.Invoice\_Time = t.Transaction\_Time;

##Merged three tables

SELECT

r.Invoice,

r.StockCode,

r.Description,

r.Quantity,

r.Invoice\_Date,

r.`Invoice\_Time`,

r.`Customer\_ID`,

r.Country,

t.`Transaction\_Date`,

t.`Transaction\_Time`

FROM (

SELECT Invoice, StockCode, Description, Quantity, Invoice\_Date, `Invoice\_Time`, `Customer\_ID`, Country

FROM onlineretailnine

UNION ALL

SELECT Invoice, StockCode, Description, Quantity, Invoice\_Date, `Invoice\_Time`, `Customer\_ID`, Country

FROM onlineretailten

) AS r

LEFT JOIN onlineretailtransactions t

ON r.`Customer\_ID` = t.`Customer\_ID`

AND r.Invoice\_Date = t.`Transaction\_Date`

AND r.`Invoice\_Time` = t.`Transaction\_Time`

LIMIT 400000;

SELECT

r.Invoice,

r.StockCode,

r.Description,

r.Quantity,

r.Invoice\_Date,

r.Invoice\_Time,

r.Customer\_ID,

r.Country,

t.Transaction\_Date,

t.Transaction\_Time

FROM (

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

Invoice\_Time, -- ✅ fixed here

Customer\_ID,

Country

FROM onlineretailnine

UNION ALL

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

Invoice\_Time, -- ✅ fixed here

Customer\_ID,

Country

FROM onlineretailten

) AS r

LEFT JOIN onlineretailtransactions t -- ✅ check correct table name

ON r.Customer\_ID = t.Customer\_ID

AND r.Invoice\_Date = t.Transaction\_Date

AND r.Invoice\_Time = t.Transaction\_Time

LIMIT 1000;

-- 1. Create the merged table (run this once)

CREATE TABLE merged\_retail\_data AS

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

`Invoice\_Time` AS InvoiceTime,

`Customer\_ID`,

Country

FROM onlineretailnine

UNION ALL

SELECT

Invoice,

StockCode,

Description,

Quantity,

Invoice\_Date,

`Invoice\_Time` AS InvoiceTime,

`Customer\_ID`,

Country

FROM onlineretailten;

-- 2. Add columns for transaction time if needed

ALTER TABLE merged\_retail\_data ADD COLUMN TransactionTime TIME;

-- 3. Update merged table with transaction time from onlineretailtransactions

UPDATE merged\_retail\_data m

JOIN onlineretailtransactions t

ON m.`Customer\_ID` = t.`Customer ID`

AND m.Invoice\_Date = t.`Transaction Date`

AND m.Invoice\_Time = t.`Transaction Time`

SET m.Transaction\_Time = t.`Transaction Time`;

CREATE TABLE merged\_retail\_data AS

SELECT

Invoice AS TransactionID,

StockCode,

Description,

Quantity,

STR\_TO\_DATE(CONCAT(Invoice\_Date, ' ', `Invoice\_Time`), '%d-%m-%Y %H:%i:%s') AS TransactionDateTime,

`Customer\_ID`,

Country

FROM onlineretailnine

UNION ALL

SELECT

Invoice,

StockCode,

Description,

Quantity,

STR\_TO\_DATE(CONCAT(Invoice\_Date, ' ', `Invoice\_Time`), '%d-%m-%Y %H:%i:%s') AS TransactionDateTime,

`Customer\_ID`,

Country

FROM onlineretailten;

SELECT

Invoice AS TransactionID,

StockCode,

Description,

Quantity,

DATE\_FORMAT(STR\_TO\_DATE(CONCAT(Invoice\_Date, ' ', `Invoice\_Time`), '%d-%m-%Y %H:%i:%s'), '%d-%m-%Y') AS TransactionDate,

TIME(STR\_TO\_DATE(CONCAT(Invoice\_Date, ' ', `Invoice\_Time`), '%d-%m-%Y %H:%i:%s')) AS TransactionTime,

`Customer\_ID`,

Country

FROM onlineretailnine

UNION ALL

SELECT

Invoice,

StockCode,

Description,

Quantity,

DATE\_FORMAT(STR\_TO\_DATE(CONCAT(Invoice\_Date, ' ', `Invoice\_Time`), '%d-%m-%Y %H:%i:%s'), '%d-%m-%Y') AS TransactionDate,

TIME(STR\_TO\_DATE(CONCAT(Invoice\_Date, ' ', `Invoice\_Time`), '%d-%m-%Y %H:%i:%s')) AS TransactionTime,

`Customer\_ID`,

Country

FROM onlineretailten;

SELECT

Invoice,

StockCode,

Description,

Quantity,

DATE\_FORMAT(TransactionTimestamp, '%d-%m-%Y') AS TransactionDate,

TIME(TransactionTimestamp) AS TransactionTime,

CustomerID,

Country

FROM

merged\_retail\_data;