**-Set Up the Raw Data Table**

Connect to your PostgreSQL database and create a table to hold the raw data.

CREATE TABLE raw\_data (

InvoiceNo VARCHAR(50),

StockCode VARCHAR(50),

Description VARCHAR(255),

Quantity INT,

InvoiceDate DATE,

UnitPrice DECIMAL(10, 2),

CustomerID INT,

Country VARCHAR(100),

TotalPrice DECIMAL(10, 2)

);

**-Import the CSV File into PostgreSQL**

Transfer the CSV file to your server or local machine where PostgreSQL is running. Then use the \copy command in psql or the import tool in pgAdmin.

**Using**psql**:**

\copy raw\_data FROM '/path/to/Cleaned\_Online\_Retail\_Corrected.csv' WITH CSV HEADER;

**-Create Dimension and Fact Tables**

-- Time Dimension Table

CREATE TABLE time\_dim (

time\_key SERIAL PRIMARY KEY,

invoice\_date DATE NOT NULL,

year INT NOT NULL,

quarter INT NOT NULL,

month INT NOT NULL,

day INT NOT NULL,

day\_of\_week INT NOT NULL,

week\_of\_year INT NOT NULL

);

-- Customer Dimension Table

CREATE TABLE customer\_dim (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255),

country\_id INT,

FOREIGN KEY (country\_id) REFERENCES country\_dim(country\_id)

);

-- Product Dimension Table

CREATE TABLE product\_dim (

product\_id SERIAL PRIMARY KEY,

stock\_code VARCHAR(50),

description VARCHAR(255),

unit\_price DECIMAL(10, 2)

);

-- Country Dimension Table

CREATE TABLE country\_dim (

country\_id SERIAL PRIMARY KEY,

country\_name VARCHAR(100) NOT NULL

);

-- Sales Fact Table

CREATE TABLE sales\_fact (

sales\_id SERIAL PRIMARY KEY,

customer\_id INT,

product\_id INT,

time\_key INT,

quantity INT,

total\_price DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES customer\_dim(customer\_id),

FOREIGN KEY (product\_id) REFERENCES product\_dim(product\_id),

FOREIGN KEY (time\_key) REFERENCES time\_dim(time\_key)

);

**-Insert Data into Dimension Tables**

INSERT INTO time\_dim (invoice\_date, year, quarter, month, day, day\_of\_week, week\_of\_year)

SELECT DISTINCT

InvoiceDate,

EXTRACT(YEAR FROM InvoiceDate),

EXTRACT(QUARTER FROM InvoiceDate),

EXTRACT(MONTH FROM InvoiceDate),

EXTRACT(DAY FROM InvoiceDate),

EXTRACT(DOW FROM InvoiceDate),

EXTRACT(WEEK FROM InvoiceDate)

FROM raw\_data;

INSERT INTO country\_dim (country\_name)

SELECT DISTINCT Country FROM raw\_data WHERE Country IS NOT NULL;

INSERT INTO customer\_dim (customer\_id, country\_id)

SELECT DISTINCT

CustomerID,

(SELECT country\_id FROM country\_dim WHERE country\_name = raw\_data.Country LIMIT 1)

FROM raw\_data

WHERE CustomerID IS NOT NULL

ON CONFLICT (customer\_id) DO NOTHING;

INSERT INTO product\_dim (stock\_code, description, unit\_price)

SELECT DISTINCT StockCode, Description, UnitPrice

FROM raw\_data;

INSERT INTO sales\_fact (customer\_id, product\_id, time\_key, quantity, total\_price)

SELECT

cd.customer\_id,

pd.product\_id,

td.time\_key,

rd.Quantity,

rd.TotalPrice

FROM raw\_data rd

INNER JOIN customer\_dim cd ON rd.CustomerID = cd.customer\_id

INNER JOIN product\_dim pd ON rd.StockCode = pd.stock\_code

INNER JOIN time\_dim td ON rd.InvoiceDate = td.invoice\_date;

-The querying the data warehouse for analysis can start once the tables are set up

Example of the query for analysis is the query for getting the total sales by country

SELECT c.country\_name, SUM(sf.total\_price) AS total\_sales

FROM sales\_fact sf

JOIN customer\_dim cd ON sf.customer\_id = cd.customer\_id

JOIN country\_dim c ON cd.country\_id = c.country\_id

GROUP BY c.country\_name

ORDER BY total\_sales DESC;

**Top Selling products**

SELECT

p.description AS product\_name,

SUM(sf.quantity) AS total\_quantity\_sold

FROM sales\_fact sf

JOIN product\_dim p ON sf.product\_id = p.product\_id

GROUP BY p.description

ORDER BY total\_quantity\_sold DESC

LIMIT 10;

**Sales by region**SELECT

co.country\_name,

SUM(sf.total\_price) AS total\_sales

FROM sales\_fact sf

JOIN customer\_dim c ON sf.customer\_id = c.customer\_id

JOIN country\_dim co ON c.country\_id = co.country\_id

GROUP BY co.country\_name

ORDER BY total\_sales DESC;

**Monthly sales trend**

SELECT

TO\_CHAR(t.invoice\_date, 'YYYY-MM') AS month,

SUM(sf.total\_price) AS total\_sales

FROM sales\_fact sf

JOIN time\_dim t ON sf.time\_key = t.time\_key

GROUP BY TO\_CHAR(t.invoice\_date, 'YYYY-MM')

ORDER BY month ASC;