```
-- Step 1: Create raw table
CREATE TABLE sales_records_raw (
    region VARCHAR(50),
    country VARCHAR(50),
    item_type VARCHAR(50),
    sales_channel VARCHAR(30),
    order_priority VARCHAR(20),
    order_date DATE,
    order_id INT PRIMARY KEY,
    ship_date DATE,
    units_sold INT,
    unit_price DECIMAL(10,2),
    unit_cost DECIMAL(10,2),
    total_revenue DECIMAL(10,2),
    total_cost DECIMAL(10,2),
    total_profit DECIMAL(10,2)
);

-- Step 2: Import data from CSV using pgAdmin import tool
-- Use pgAdmin GUI: Right-click table > Import/Export > Choose CSV

-- Step 3: Create lookup tables
CREATE TABLE regions (
    region_id SERIAL PRIMARY KEY,
    region_name VARCHAR(50) UNIQUE
);

CREATE TABLE countries (
    country_id SERIAL PRIMARY KEY,
    country_name VARCHAR(50),
    region_id INT REFERENCES regions(region_id)
);

CREATE TABLE item_types (
    item_type_id SERIAL PRIMARY KEY,
    item_type_name VARCHAR(50) UNIQUE
);

-- Step 4: Populate lookup tables
INSERT INTO regions (region_name)
SELECT DISTINCT region FROM sales_records_raw;

INSERT INTO countries (country_name, region_id)
SELECT DISTINCT sr.country, r.region_id
FROM sales_records_raw sr
JOIN regions r ON sr.region = r.region_name;

INSERT INTO item_types (item_type_name)
SELECT DISTINCT item_type FROM sales_records_raw;

-- Step 5: Create final normalized table
CREATE TABLE sales_records (
    order_id INT PRIMARY KEY,
    country_id INT REFERENCES countries(country_id),
    item_type_id INT REFERENCES item_types(item_type_id),
    sales_channel VARCHAR(30),
    order_priority VARCHAR(20),
    order_date DATE,
    ship_date DATE,
    units_sold INT,
    unit_price DECIMAL(10,2),
    unit_cost DECIMAL(10,2),
    total_revenue DECIMAL(10,2),
    total_cost DECIMAL(10,2),
    total_profit DECIMAL(10,2)
);

-- Step 6: Populate normalized table
INSERT INTO sales_records (
    order_id, country_id, item_type_id, sales_channel, order_priority,
    order_date, ship_date, units_sold, unit_price, unit_cost,
    total_revenue, total_cost, total_profit
)
SELECT
    sr.order_id,
    c.country_id,
    it.item_type_id,
    sr.sales_channel,
    sr.order_priority,
    sr.order_date,
    sr.ship_date,
    sr.units_sold,
    sr.unit_price,
    sr.unit_cost,
    sr.total_revenue,
    sr.total_cost,
    sr.total_profit
FROM sales_records_raw sr
JOIN countries c ON sr.country = c.country_name
JOIN item_types it ON sr.item_type = it.item_type_name;

-- Step 7: Create indexes
CREATE INDEX idx_country_id ON sales_records(country_id);
CREATE INDEX idx_item_type_id ON sales_records(item_type_id);
CREATE INDEX idx_sales_channel ON sales_records(sales_channel);
CREATE INDEX idx_total_profit ON sales_records(total_profit);

-- Step 8: Analyze database
ANALYZE;

--Queries

-- Query 1: Top-selling item types
SELECT it.item_type_name, SUM(sr.units_sold) AS total_units_sold
FROM sales_records sr
JOIN item_types it ON sr.item_type_id = it.item_type_id
GROUP BY it.item_type_name
ORDER BY total_units_sold DESC
```

```
LIMIT 5;

-- Query 2: Total revenue by region
SELECT r.region_name, SUM(sr.total_revenue) AS total_revenue
FROM sales_records sr
JOIN countries c ON sr.country_id = c.country_id
JOIN regions r ON c.region_id = r.region_id
GROUP BY r.region_name
ORDER BY total_revenue DESC;

-- Query 3: Revenue per sales channel
SELECT sales_channel, SUM(total_revenue) AS total_revenue
FROM sales_records
GROUP BY sales_channel
ORDER BY total_revenue DESC;
```