**Star Schema Data Warehouse Implementation**

**Objective**

To design and implement a star schema in PostgreSQL that supports analytical queries on Walmart’s weekly sales data. This includes creating normalized dimension tables and a centralized fact table for efficient querying and reporting.

**Dataset Overview**

**Source File:** Walmart\_Sales.csv  
**Fields:**

* **Store:** Store ID
* **Date:** Weekly record date
* **Weekly\_Sales:** Sales revenue for the week
* **Holiday\_Flag:** 1 if holiday week, else 0
* **Temperature:** Average temperature
* **Fuel\_Price:** Cost per gallon
* **CPI:** Consumer Price Index
* **Unemployment:** Regional unemployment rate

**Schema Design**

**The star schema consists of the following tables:**

1. Fact Table: fact\_weekly\_sales  
 - date\_key (FK to dim\_date)  
 - store\_key (FK to dim\_store)  
 - weekly\_sales  
 - holiday\_flag  
 - temperature  
 - fuel\_price  
 - cpi  
 - unemployment  
  
2. Dimension Table: dim\_date  
 - date\_key (Primary Key)  
 - date  
 - week  
 - month  
 - quarter  
 - year  
  
3. Dimension Table: dim\_store  
 - store\_key (Primary Key)

**Implementation Steps**

**1. Database Creation**

***CREATE DATABASE retail\_dw OWNER postgres;***

**2. Schema & Table Setup**

**Tables were created inside the star schema, including a staging table for CSV import:**

***CREATE SCHEMA IF NOT EXISTS star;***

***SET search\_path = star;***

**Staging table for raw CSV**

***CREATE TABLE IF NOT EXISTS staging\_sales (***

***store INTEGER,***

***date DATE,***

***weekly\_sales NUMERIC(14,2),***

***holiday\_flag SMALLINT,***

***temperature REAL,***

***fuel\_price NUMERIC(6,3),***

***cpi REAL,***

***unemployment REAL***

***);***

**Dimension: Date**

***CREATE TABLE IF NOT EXISTS dim\_date (***

***date\_key INTEGER PRIMARY KEY, -- YYYYMMDD***

***date DATE NOT NULL,***

***week SMALLINT NOT NULL,***

***month SMALLINT NOT NULL,***

***quarter SMALLINT NOT NULL,***

***year INTEGER NOT NULL***

***);***

**Dimension: Store**

***CREATE TABLE IF NOT EXISTS dim\_store (***

***store\_key INTEGER PRIMARY***

***);***

**Fact: Weekly Sales**

***CREATE TABLE IF NOT EXISTS fact\_weekly\_sales (***

***date\_key INTEGER NOT NULL***

***REFERENCES dim\_date(date\_key),***

***store\_key INTEGER NOT NULL***

***REFERENCES dim\_store(store\_key),***

***weekly\_sales NUMERIC(14,2) NOT NULL,***

***holiday\_flag SMALLINT NOT NULL,***

***temperature REAL,***

***fuel\_price NUMERIC(6,3),***

***cpi REAL,***

***unemployment REAL,***

***PRIMARY KEY (date\_key, store\_key)***

***);***

**3. Data Import Using \copy**

**Used \copy to load local CSV:**

***\copy staging\_sales(store, date, weekly\_sales, holiday\_flag, temperature, fuel\_price, cpi, unemployment)***

***FROM 'C:/Users/Manan/Downloads/Walmart\_Sales.csv'***

***WITH (FORMAT csv, HEADER true);***

**4. Populating Dimension Tables**

**dim\_date:**

***INSERT INTO dim\_date(date\_key, date, week, month, quarter, year)***

***SELECT***

***EXTRACT(YEAR FROM date)::INT \* 10000 +***

***EXTRACT(MONTH FROM date)::INT \* 100 +***

***EXTRACT(DAY FROM date)::INT AS date\_key,***

***date,***

***TO\_CHAR(date, 'IW')::SMALLINT AS week,***

***EXTRACT(MONTH FROM date)::SMALLINT,***

***EXTRACT(QUARTER FROM date)::SMALLINT,***

***EXTRACT(YEAR FROM date)::INT***

***FROM (SELECT DISTINCT date FROM staging\_sales) d***

***ON CONFLICT (date\_key) DO NOTHING;***

**dim\_store:**

***INSERT INTO dim\_store(store\_key)***

***SELECT DISTINCT store FROM staging\_sales***

***ON CONFLICT (store\_key) DO NOTHING;***

**5. Populating Fact Table**

***INSERT INTO fact\_weekly\_sales(date\_key, store\_key, weekly\_sales, holiday\_flag, temperature, fuel\_price, cpi, unemployment)***

***SELECT***

***EXTRACT(YEAR FROM date)::INT \* 10000 +***

***EXTRACT(MONTH FROM date)::INT \* 100 +***

***EXTRACT(DAY FROM date)::INT AS date\_key,***

***store,***

***weekly\_sales,***

***holiday\_flag,***

***temperature,***

***fuel\_price,***

***cpi,***

***unemployment***

***FROM staging\_sales***

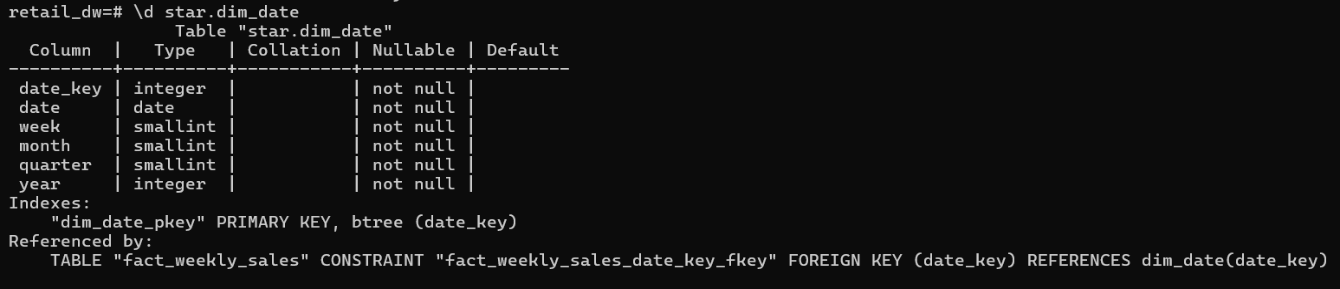
***ON CONFLICT (date\_key, store\_key) DO NOTHING;***

**Validation &Testing**

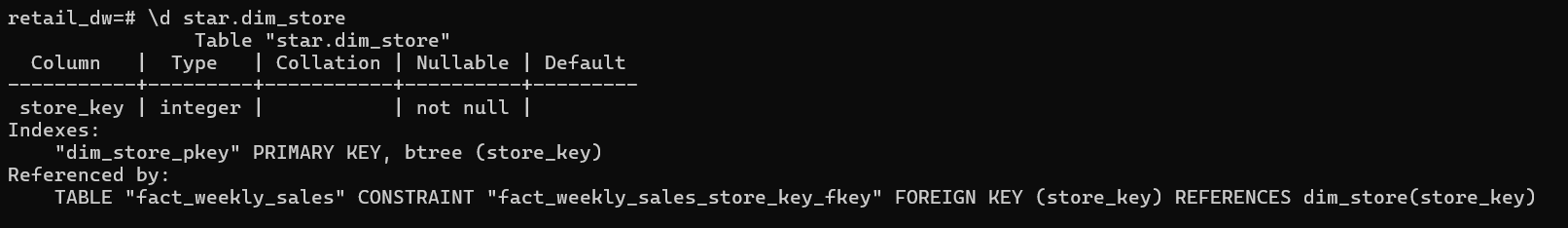
**Verify Structure of Each Table**

**Check columns and primary keys:**

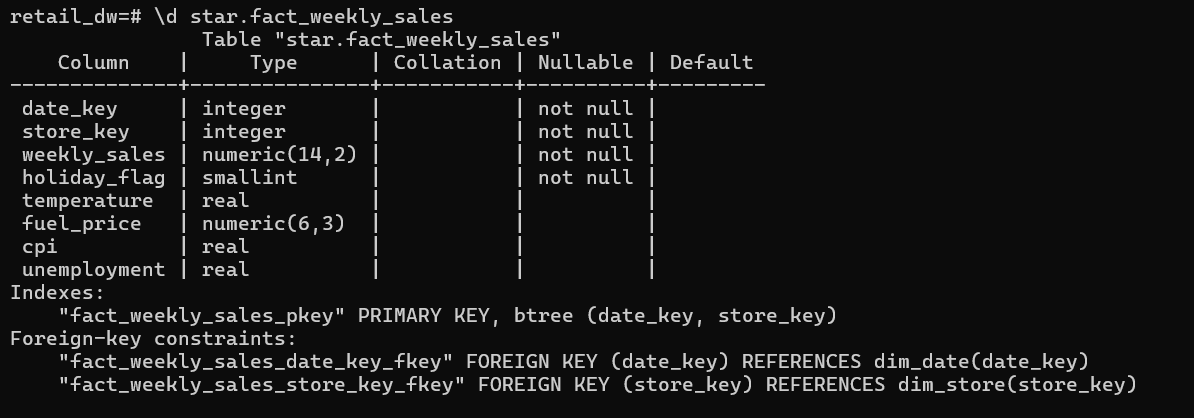
***\d star.dim\_date***



***\d star.dim\_store***



***\d star.fact\_weekly\_sales***

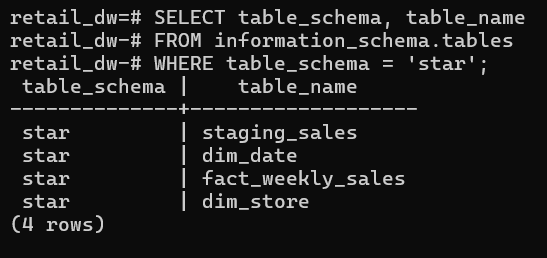


**Schema Introspection**

***SELECT table\_schema, table\_name***

***FROM information\_schema.tables***

***WHERE table\_schema = 'star';***



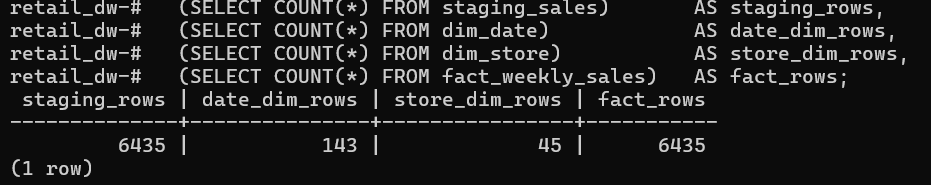
**Row Counts**

***SELECT***

***(SELECT COUNT(\*) FROM star.dim\_date) AS dim\_date\_rows,***

***(SELECT COUNT(\*) FROM star.dim\_store) AS dim\_store\_rows,***

***(SELECT COUNT(\*) FROM star.fact\_weekly\_sales) AS fact\_rows;***



**Join Query Test**

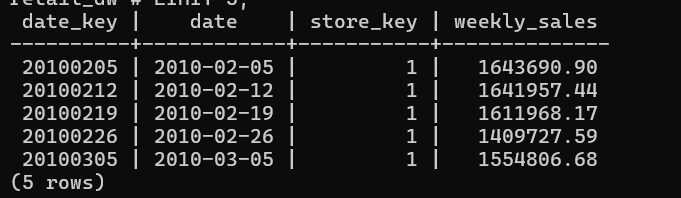
***SELECT f.date\_key, d.date, f.store\_key, f.weekly\_sales***

***FROM star.fact\_weekly\_sales f***

***JOIN star.dim\_date d ON f.date\_key = d.date\_key***

***JOIN star.dim\_store s ON f.store\_key = s.store\_key***

***LIMIT 5;***



**Outcome**

* Star schema successfully implemented with normalized dimensions.
* Referential integrity established via foreign keys.
* Dataset prepared for analytical workloads, dashboards, or BI tools.