📊 Superstore KPI Dashboard — Full-Stack Guide (SQL Server + Python + React)

A full-stack analytics project powered by:

- 🐘 SQL Server (SSMS)
- 🐍 Python (for data transformation, optional)
- ⚛️ React + D3 (frontend dashboard)

📁 Folder Structure

```
📂 DevProjects
└── 📁 superstore-kpi-dashboard
    ├── 📁 sql/            # SQL schema + KPI scripts
    ├── 📁 python/         # Data cleaning, ETL scripts or notebooks
    ├── 📁 react/          # Frontend React app
    ├── 📁 public-data/    # Cleaned KPI data (JSON or CSV)
    └── 📁 docs/           # Markdown docs like this one
```

1️⃣ Load Superstore Data into SQL Server

☑️ Step 1.1: Create the Database

In **SSMS**:

```sql
CREATE DATABASE superstore_db;
GO
USE superstore_db;
```

☑️ Step 1.2: Create `sales` Table

```sql
CREATE TABLE sales (
  Row_ID INT,
  Order_ID VARCHAR(50),
  Order_Date DATE,
  Ship_Date DATE,
  Ship_Mode VARCHAR(50),
  Customer_ID VARCHAR(50),
  Segment VARCHAR(50),
  Country VARCHAR(50),
  City VARCHAR(100),
  State VARCHAR(100),
  Postal_Code VARCHAR(20),
  Region VARCHAR(50),
  Product_ID VARCHAR(50),
  Category VARCHAR(50),
  Sub_Category VARCHAR(50),
  Product_Name VARCHAR(200),
  Sales FLOAT,
  Quantity INT,
  Discount FLOAT,
  Profit FLOAT
);
```

☑ Step 1.3: Import CSV (2 Options)

🧙 Option A: SSMS GUI

1. Right-click `superstore_db` → `Tasks → Import Flat File`
2. Choose `Superstore.csv`
3. Map columns
4. Finish

🐍 Option B: Python Script

```
import pandas as pd
import pyodbc

df = pd.read_csv("C:/DevProjects/superstore-kpi-dashboard/public-data/superstore.csv")

conn = pyodbc.connect("Driver={SQL Server};Server=localhost;Database=superstore_db;Trusted_Connection=yes;")
cursor = conn.cursor()

for _, row in df.iterrows():
    cursor.execute("""
        INSERT INTO sales (...) VALUES (?, ?, ?, ...);
    """, row["Order ID"], ...)  # Add all required fields

conn.commit()
```

---

2 Clean and Transform Data

☑ Step 2.1: Clean with SQL

```
-- Remove blanks
DELETE FROM sales WHERE Customer_ID IS NULL;

-- Normalize text
UPDATE sales SET Segment = UPPER(Segment);
```

☑ Step 2.2: Optional Python Cleaning

```
df = pd.read_sql("SELECT * FROM sales", conn)
df["Order_Date"] = pd.to_datetime(df["Order_Date"])
df["Segment"] = df["Segment"].str.strip().str.title()
```

---

3 Create KPIs for Dashboard

☑ Step 3.1: Build SQL View

```
CREATE VIEW kpi_dashboard_data AS
SELECT
  FORMAT(Order_Date, 'yyyy-MM') AS Month,
  SUM(Sales) AS Total_Sales,
  AVG(Discount) AS Avg_Discount,
  SUM(Profit) / NULLIF(SUM(Sales), 0) AS Profit_Ratio,
  COUNT(DISTINCT Customer_ID) AS Unique_Customers
FROM sales
GROUP BY FORMAT(Order_Date, 'yyyy-MM');
```

☑ Step 3.2: Export View as JSON

Python:

```
df = pd.read_sql("SELECT * FROM kpi_dashboard_data", conn)
df.to_json("C:/DevProjects/superstore-kpi-dashboard/public-data/dashboard-data.json", orient="records")
```

---

4 Build Modular React Dashboard

☑ Step 4.1: React Project Structure

```
📂 react/
├── 📁 src/
│   ├── 📁 components/
│   │   ├── KpiCard.js
│   │   ├── LineChart.js
│   │   └── Dashboard.js
│   └── App.js
├── 📁 public/
│   └── data/dashboard-data.json
```

☑ Step 4.2: Fetch Data in React

```
useEffect(() => {
  fetch(process.env.PUBLIC_URL + "/data/dashboard-data.json")
    .then((res) => res.json())
    .then((data) => setKpis(data));
}, []);
```

☑ Step 4.3: Display KPIs

```
kpis.map((kpi) => (
  <KpiCard
    title="Total Sales"
    value={kpi.Total_Sales}
    percentage={calculateMoM(kpi)}
    isPositive={kpi.change > 0}
  />
));
```

☑ Step 4.4: Equal-Size KPI Grid CSS

```
.kpi-grid {
  display: flex;
  flex-wrap: wrap;
  gap: 1rem;
}

.kpi-card {
  flex: 1 1 30%;
  max-width: 30%;
  min-width: 200px;
  height: 150px;
  border-radius: 8px;
```

```
    box-shadow: 0px 2px 8px rgba(0, 0, 0, 0.1);
    padding: 1rem;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
  }
```

✅ Final System Overview

Layer	Role
`sql/`	Schema, views, kpi queries
`python/`	Data cleaning, JSON export
`public-data/`	Frontend-ready data
`react/`	Visualize KPIs
`docs/`	Markdown & planning files

🧠 Next Steps

- [ ] Add YoY / MoM % changes to KPIs
- [ ] Add filters (region, category)
- [ ] Build time-series LineChart.js
- [ ] Optionally serve KPIs with FastAPI

🧪 Resources

- Superstore Dataset: https://www.kaggle.com/datasets/vivek468/superstore-dataset-final
- SQL Server Dev Edition: https://aka.ms/sqldev
- SSMS Download: https://aka.ms/ssmsfullsetup
- VS Code Markdown PDF Extension: https://marketplace.visualstudio.com/items?itemName=yzane.markdown-pdf

---

You just built a real-world KPI dashboard stack — from raw CSV to dynamic frontend. Congrats, you're full-stack certified 🧠📈⚛️