🍕 Pizza Sales Analysis — SQL → Power BI End-to-End Workflow

🔹 Step 1: Data Collection

- Dataset: Pizza sales data (tables like orders, order_details, pizzas, pizza_types).
- Tables contain info on order timestamps, pizza sizes, categories, prices, and ingredients.

🔹 Step 2: Data Preparation in SQL

1. **Data Cleaning**
    - Remove nulls and duplicates.
    - Standardize pizza category/size values (e.g., “L” vs. “Large”).
    - Convert date/time columns into proper SQL DATE and TIME formats.

Example:

```
-- Format order date and time
SELECT
  order_id,
  CAST(order_date AS DATE) AS order_date,
  CAST(order_time AS TIME) AS order_time
FROM orders;
```

2. **Join Tables**
    - Join orders with order_details and pizzas to get sales-level data.

```
3. SELECT
4.     o.order_id,
5.     o.order_date,
6.     o.order_time,
7.     p.pizza_type_id,
8.     p.size,
9.     p.price,
```

10. od.quantity,
11. (p.price * od.quantity) AS revenue
12. FROM orders o
13. JOIN order_details od ON o.order_id = od.order_id
14. JOIN pizzas p ON od.pizza_id = p.pizza_id;

- **Step 3: SQL KPI Queries**
  - **Total Revenue**
  - SELECT ROUND(SUM(p.price * od.quantity), 2) AS Total_Revenue
  - FROM order_details od
  - JOIN pizzas p ON od.pizza_id = p.pizza_id;
  - **Total Orders**
  - SELECT COUNT(DISTINCT order_id) AS Total_Orders
  - FROM orders;
  - **Average Order Value (AOV)**
  - SELECT
  - ROUND(SUM(p.price * od.quantity) / COUNT(DISTINCT o.order_id), 2) AS Avg_Order_Value
  - FROM orders o
  - JOIN order_details od ON o.order_id = od.order_id
  - JOIN pizzas p ON od.pizza_id = p.pizza_id;
  - **Best & Worst Selling Pizzas**
  - SELECT
  - p.pizza_type_id,
  - SUM(od.quantity) AS Total_Quantity,
  - ROUND(SUM(p.price * od.quantity), 2) AS Total_Revenue

- FROM order_details od
- JOIN pizzas p ON od.pizza_id = p.pizza_id
- GROUP BY p.pizza_type_id
- ORDER BY Total_Revenue DESC;
- **Category & Size Contribution**
- SELECT
- pt.category,
- p.size,
- ROUND(SUM(p.price * od.quantity), 2) AS Revenue
- FROM order_details od
- JOIN pizzas p ON od.pizza_id = p.pizza_id
- JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
- GROUP BY pt.category, p.size
- ORDER BY Revenue DESC;
- **Time-based Trends (Hour of Day)**
- SELECT
- DATEPART(HOUR, order_time) AS Order_Hour,
- ROUND(SUM(p.price * od.quantity), 2) AS Revenue
- FROM orders o
- JOIN order_details od ON o.order_id = od.order_id
- JOIN pizzas p ON od.pizza_id = p.pizza_id
- GROUP BY DATEPART(HOUR, order_time)
- ORDER BY Order_Hour;

🔹 **Step 4: Export SQL Results**

- Export cleaned and aggregated results as .csv (or connect Power BI directly to SQL).

- Tables to export:
    - **Orders_cleaned.csv**
    - **Sales_KPIs.csv**
    - **Category_Size.csv**
    - **Time_Trends.csv**
    - **Top_Bottom_Pizzas.csv**

🔹 **Step 5: Power BI Dashboard**

1. **Data Import**
    - Connect Power BI to SQL Server (live) or load exported .csv files.
2. **Data Modeling**
    - Build relationships between Orders, Order_Details, Pizzas, and Pizza_Types.
    - Use a **star schema** (fact table = order details, dimensions = pizzas, types, time).
3. **DAX Measures**
    - Total Revenue = SUMX(Order_Details, Order_Details[Quantity] * Pizzas[Price])
    - Total Orders = DISTINCTCOUNT(Orders[Order_ID])
    - Avg Order Value = [Total Revenue] / [Total Orders]
4. **Visuals**
    - KPI cards for **Revenue, AOV, Orders, Quantity**.
    - Bar/column charts for **Top 10 & Bottom 10 pizzas**.
    - Donut chart for **category & size contribution**.
    - Line/area chart for **daily & hourly sales trends**.
    - Filters (date, category, size).
5. **Styling**
    - Used branded colors (blue, orange, green).
    - Added slicers, navigation panels, and bold KPIs (like in your screenshots).

- **Step 6: Insights Delivered**
  - Classic pizzas = top performing category.
  - Large pizzas = ~46% of revenue.
  - Peak hours = lunch (12–1 PM) & evenings (4–7 PM).
  - December (Week 48) = highest sales period.
  - Thai Chicken Pizza = best-selling.
  - Brie Carre Pizza = least selling.