## Entity-Relationship Overview:

A close-up of a diagram

Description automatically generated

## Table-by-Table Summary:

### 1. suppliers.csv

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| supplier\_id | UUID | Unique ID for each supplier |
| supplier\_name | String | Company name |
| contact\_email | String | Supplier email |
| Phone | String | Contact number |
| Address | String | Address with street, city, zip, etc. |

**Rows**: 50  
**Usage**: Used to match with Purchase Orders; ideal for supplier-wise analysis (volume, reliability, etc.)

### 2. products.csv

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| product\_id | UUID | Unique ID for each product |
| product\_name | String | Name of the product |
| Category | Enum | One of: Electronics, Furniture, Office Supplies, Food |
| unit\_price | Float | Price per unit ($5 - $500) |

**Rows**: 200  
**Usage**: Enables category/product-level cost and sales insights; links to purchase orders.

### 3. purchase\_orders.csv

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| po\_id | UUID | Unique purchase order ID |
| supplier\_id | UUID | Linked to suppliers |
| product\_id | UUID | Linked to products |
| order\_date | Date | When the PO was placed |
| quantity | Integer | Units ordered (1–100) |
| unit\_price | Float | From products |
| total\_amount | Float | unit\_price \* quantity |

**Rows**: 8,000  
**Usage**: Core fact table for procurement; links suppliers with products. Can be aggregated over time, supplier, or category.

### 4. shipments.csv

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| shipment\_id | UUID | Unique ID for shipment |
| po\_id | UUID | Linked to purchase\_orders |
| shipment\_date | Date | Shipment date (1–14 days after order) |
| quantity | Integer | Quantity shipped (same as PO for simplicity) |

**Rows**: ~7,600 (95% of POs shipped)  
**Usage**: Analyze delivery efficiency, delay metrics, and fulfillment rate.

### 5. invoices.csv

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| invoice\_id | UUID | Unique ID |
| po\_id | UUID | Links to purchase\_orders |
| invoice\_date | Date | 2–10 days after order\_date |
| due\_date | Date | invoice\_date + 30 days |
| amount\_due | Float | From purchase\_orders.total\_amount |

**Rows**: 8,000  
**Usage**: Essential for accounts payable; basis for cash flow and overdue invoice analysis.

### 6. payments.csv

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| payment\_id | UUID | Unique ID for payment |
| invoice\_id | UUID | Linked to invoices |
| payment\_date | Date | Paid within 1–30 days of invoice date |
| paid\_amount | Float | 80%–100% of amount\_due |

**Rows**: ~7,200 (90% of invoices paid)  
**Usage**: Analyze DSO (days sales outstanding), cash collection ratio, and payment delays.

### Relationships Summary:

|  |  |  |  |
| --- | --- | --- | --- |
| **Table A** | **Table B** | **Relationship** | **Type** |
| suppliers | purchase\_orders | 1-to-many | via supplier\_id |
| Products | purchase\_orders | 1-to-many | via product\_id |
| purchase\_orders | shipments | 1-to-zero/one | via po\_id |
| purchase\_orders | invoices | 1-to-1 | via po\_id |
| invoices | payments | 1-to-zero/one | via invoice\_id |

### Analysis & dbt Modeling Ideas:

Here are some starter questions + potential dbt models you can build:

|  |  |  |
| --- | --- | --- |
| **Topic** | **dbt Model Name** | **Description** |
| Total Spend per Supplier | supplier\_spend | Join POs + Invoices grouped by supplier\_id |
| Shipment Delays | shipment\_delays | Compare order\_date and shipment\_date |
| Invoice Aging | invoice\_aging | Days between invoice and payment or today |
| Payment Status | payment\_status | Paid, Partially Paid, Unpaid |
| Category Spend | category\_costs | Join products + POs; group by category |
| Cash Flow Summary | cashflow\_summary | Total amount\_due vs. paid\_amount by month |

## Date Ranges

* All data is generated for the **last 2 years** (from today backward).
* This allows for:
  + **Monthly/quarterly trend analysis**
  + **Year-over-year comparisons**
  + **Lag-based metrics** like payment delays, shipment delays, etc.