🧱 1. Data Model Overview

We'll use a **star schema**, which is ideal for Power BI.
👉 **Fact tables** store transactional data (amounts, dates, statuses).
👉 **Dimension tables** store descriptive information (parties, currencies, codes, time).

🌟 Fact Table

`FactPayments` — contains **one row per transaction** (e.g., pacs.008).

Field Name	Type	Description
PaymentID	Text	Unique internal ID (e.g., combination of MsgId + InstrId).
MsgId	Text	Message identifier from `pacs.008` or `pain.001`.
InstrId	Text	Instruction identifier.
EndToEndId	Text	End-to-end reference (links customer → interbank → beneficiary).
PaymentDate	Date	Transaction initiation or settlement date.
SettlementDate	Date	Interbank settlement date.
Amount	Decimal	Transaction amount.
CurrencyCode	Text	ISO 4217 currency code (e.g., EUR, USD).
DebtorID	Text	Key to Debtor dimension.
CreditorID	Text	Key to Creditor dimension.
DebtorAgentBIC	Text	BIC of debtor's bank.
CreditorAgentBIC	Text	BIC of creditor's bank.
PurposeCode	Text	Payment purpose (e.g., SALA, SUPP).
StatusCode	Text	Processing status (e.g., ACSP, RJCT).
ChargeBearer	Text	Who pays the charges (e.g., SHAR, DEBT, CRED).
ProcessingTimeMinutes	Decimal	Derived: time between initiation and settlement.

🧍 Dimension Tables

🧭 `DimParty` (Debtors & Creditors)

Field	Description
PartyID	Key
Name	Name of company or individual
IBAN	Account IBAN
CountryCode	ISO 2-letter country code
City	Optional
LEI	Legal Entity Identifier (if available)

👉 Used twice in the model — one relationship for DebtorID and one for CreditorID.

💱 `DimCurrency`

Field	Description
CurrencyCode	ISO 4217 code
CurrencyName	e.g., Euro, US Dollar
CurrencySymbol	€, $

📅 `DimDate`

Field	Description
Date	Date key
Year	Year
Quarter	Quarter
Month	Month name
MonthNumber	Numeric month
Day	Day
WeekNumber	Week of year

👉 Linked to both `PaymentDate` and `SettlementDate`.

🗒️ DimPurposeCode

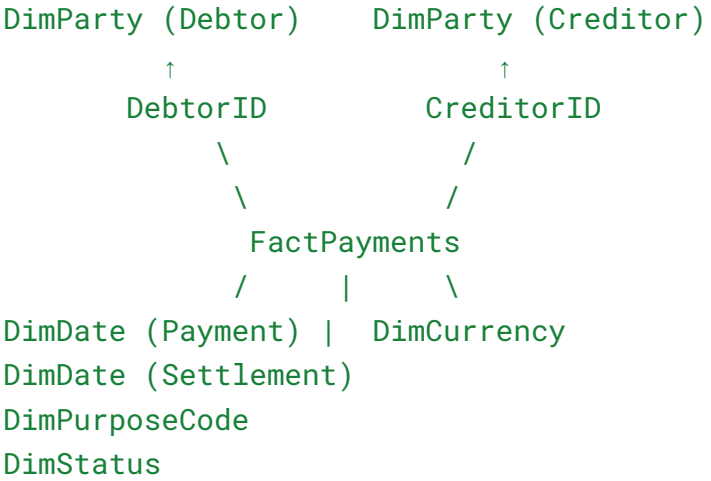
Field	Description
PurposeCode	Code from ISO 20022 list
Description	e.g., SALA = Salary Payment, SUPP = Supplier Payment

---

🧭 DimStatus

Field	Description
StatusCode	e.g., ACSP, RJCT
Description	Accepted for Settlement, Rejected, Pending, etc.

🔗 2. Relationships Diagram



👉 In Power BI, you'll create **two relationships to DimParty**:

- One active for Debtor
- One inactive for Creditor (you can use DAX `USERELATIONSHIP` in measures to switch)

📊 3. Dashboard Layout

Here's a recommended **Power BI dashboard structure**, divided into **pages** (tabs):

🟡 Page 1: Payments Overview (Executive Dashboard)

KPIs at top:

- Total Payments (count)
- Total Amount (sum)
- % Rejected Transactions
- Average Processing Time (minutes)

Main visuals:

- 📅 **Line chart** — Daily or monthly volume and value trends.
- 🌍 **Map** — Debtor → Creditor flows by country (using CountryCode).
- 📊 **Bar chart** — Top 10 Purpose Codes by volume/value.
- 🧮 **Table** — High-value transactions with status.

🟢 Page 2: Operational Monitoring

Focus: Real-time or daily payment processing.

- 🔸 Matrix by DebtorAgentBIC → CreditorAgentBIC showing total amount and number of transactions.
- ⏱️ Processing time distribution histogram (to detect delays).
- 🧭 Status breakdown (ACSP, RJCT, etc.) — stacked bar or donut chart.
- Filters for date, currency, status, purpose.

🗒️ Page 3: Reconciliation

Goal: Match customer instructions (pain.001) with interbank and settlement messages.

- Table linking EndToEndId between FactPayments and `camt.054` (credits received).
- KPIs: Number of unmatched transactions, delayed settlements.
- Drill-through to individual transaction timeline:
    - pain.001 → pacs.008 → pacs.002 → camt.054

🕵️ Page 4: Regulatory / Compliance

- Filter payments by PurposeCode or Country.
- Top corridors (e.g., PT → DE, US → UK) by volume.
- Identify transactions missing LEI, PurposeCode, or structured remittance.
- List high-value payments by purpose for reporting.

📈 Page 5: Advanced Analytics (Optional)

If you have larger datasets:

- Time-series forecasting of volumes.
- Anomaly detection for unusual amounts or routing.
- Correlation between purpose codes and processing delays.

⚡ 4. Why This Works Well

- ISO 20022 data has **clear identifiers (MsgId, EndToEndId)** that make joining different messages easy.

- Fields like **BIC, PurposeCode, Country, Status** are standard — ideal for consistent visuals.
- Power BI can **incrementally refresh** and scale to large payment volumes.
- You can blend **operational + regulatory + strategic views** in one place.

🧠 5. Next Steps to Build It

1. 🧾 **Extract & transform** ISO 20022 XML messages into structured tables (Power Query or ETL).
2. 🗃️ **Create the star schema** as shown above.
3. ⚙️ Build **relationships** in Power BI.
4. 📊 **Design visuals** page by page using the fields.
5. 🧠 Add **DAX measures** for KPIs (e.g., processing time, rejection rate).
6. 🔐 Apply row-level security if needed (e.g., per business unit or region).

🚀 In Short

✅ Yes — ISO 20022 payment data lends itself *perfectly* to a structured **Power BI dashboard**.
With the star schema above, you can build:

- 📊 Executive overviews
- 🧭 Operational dashboards
- 🧾 Reconciliation tools
- 🕵️ Compliance reports
- 💡 Strategic analytics

Once the ETL is in place, this dashboard becomes a **real-time "payment control tower"** 🏦✨