# Production Dataset Overview

This folder contains the user-facing extracts that downstream analysts and investigators will receive. The files are already cleaned and denormalised enough to be stitched together with simple joins, so consumers do **not** have to understand how the synthetic engine works. Use this guide as the single reference for column meanings, relationships, and safe assumptions.

## Quick reference

| File | Purpose | Primary key(s) | Natural joins | Notes |
|---|---|---|---|---|
| account.csv | One record per bank account. | account_id | br_to_account.account_id, transactions.Account ID | Dates follow YYYY-MM-DD; currency is 3-letter ISO. |
| br_to_account.csv | Junction between business relationships and accounts. | (br_id, account_id) | business_rel.br_id, account.account_id, partner_role.entity_id | relationship_status_code = 1 means active. |
| business_rel.csv | Metadata about each business relationship (BR). | br_id | br_to_account.br_id, partner_role.entity_id | Close date empty ⇒ live relationship. |
| client_onboarding_notes.csv | Unstructured onboarding note per partner. | Partner_ID | partner.partner_id, partner_role.partner_id | UTF-8 text, may contain punctuation and em-dashes. |
| graph.graphml | Property graph describing parties, industries, and their links. | node.@id, edge.@source/@target | Use alongside partner.csv and partner_role.csv. | GraphML 1.0, directed edges; role attribute stores vertex type. |

| | | | | |
|---|---|---|---|---|
| partner.csv | Master list of people or organisations. | partner_id | partner_role.partner_id, partner_country.partner_id, client_onboarding_notes.Partner_ID | Address is multi-line; dates in YYYY-MM-DD. |
| partner_country.csv | Country attributes per partner. | (partner_id, country_type) | partner.partner_id | partner_country_status_code = 1 ⇒ current residence. |
| partner_role.csv | Links partners to business relationships or other parties. | (partner_id, entity_id, br_type_code) | business_rel.br_id, br_to_account.br_id | entity_type is BR for BR relationships; other values are rare but reserved. |
| transactions.csv | Double-entry ledger of account activity. | Transaction ID | account.account_id via Account ID | Datetime is UTC, format YYYY-MM-DD HH:MM. Amounts signed using debit/credit flag. |

# Dataset details

## account.csv

- **Scope:** One row per bank account (retail or partnership).
- **Column glossary:**
  - account_id – Computer-generated UUID that never changes; use this as the unique key.
  - account_iban – International Bank Account Number. Useful for reports, but treat it as display-only because multiple accounts can share similar prefixes.
  - account_currency – Three-letter ISO code telling you which currency an account is denominated in (CHF for Swiss francs, EUR for euro, etc.).
  - account_open_date / account_close_date – Calendar dates in YYYY-MM-DD. A blank close date simply means the account is still active.
- **Safe assumptions:**
  - account_id is globally unique and stable.
  - account_currency always uses ISO 4217 (CHF, EUR, USD, etc.).
  - account_close_date is blank for open accounts; when populated it is on or after the open date.
  - Each account belongs to at least one business relationship via br_to_account.csv.

## br_to_account.csv

- **Scope:** Many-to-many table mapping business relationships (br_id) to the accounts they control.
- **Column glossary:**
  - br_id – Identifier of the business relationship (see business_rel.csv).
  - account_id – Identifier of the linked account.
  - relationship_id – Internal row identifier; handy when you need to refer to a specific connection.
  - relationship_status_code – Simple numeric flag (1 = active, 0 or other values = inactive or historical).
- **Safe assumptions:**
  - (br_id, account_id) pairs are unique.
  - relationship_status_code = 1 indicates the link is active; other codes represent historical states.
  - Every br_id exists in business_rel.csv; every account_id exists in account.csv.

## business_rel.csv

- **Scope:** Business relationships are the contractual shells that bind one or more partners to one or more accounts.
- **Column glossary:**
  - br_id – Stable UUID identifying the relationship; surfaces in several other files.
  - br_open_date – Date the relationship was created (first account signed).
  - br_close_date – Date the relationship formally ended. Blank equals still active.
- **Safe assumptions:**
  - br_id is the authoritative key used throughout the model.
  - br_close_date is blank for live relationships.
  - A relationship can own multiple accounts and have multiple partners (through partner_role.csv).

## client_onboarding_notes.csv

- **Scope:** Free-form onboarding narratives written by relationship managers.
- **Column glossary:**
  - Partner_ID – Matches partner.partner_id.
  - Onboarding_Note – Long text field containing the consultant's description of the client's background, goals, communication preferences, and risk appetite.
- **Safe assumptions:**
  - There is at most one row per partner; some partners may be missing notes.
  - Text is UTF-8 and may contain em dashes, quotes, and multi-sentence paragraphs.
  - Use this file only for qualitative insights—there are no structured fields inside the note.

## graph.graphml

- **Scope:** Directed property graph capturing the ecosystem of partners, industries, and their affiliations.
- **Key components:**
  - Nodes have id (UUID) and a role attribute describing the type (e.g., Individual, Commercial banks).
  - Edges carry a numeric weight that can be interpreted as interaction frequency or strength.
- **Column/attribute glossary:**
  - node.@id – Unique identifier for each vertex; matches partner IDs when the node is a person or

organisation.
• node.data[@key="d0"] – Human-readable description of the node’s role or industry sector.
• edge.@source / edge.@target – Point from the influencing entity to the influenced one. Think of it as “source owns or interacts with target.”
• edge.data[@key="d1"] – Numeric weight; higher values indicate denser or more frequent interactions.
• **Safe assumptions:**
• The GraphML conforms to version 1.0 and can be loaded into NetworkX, Neo4j importers, or Gephi.
• Node IDs align with partner IDs where applicable, so you can cross-reference with partner.csv.
• Edge direction follows the flow of influence or ownership (source → target).

## partner.csv

• **Scope:** Golden record for every individual or organisation that can hold a banking relationship.
• **Column glossary:**
• partner_id – Master key; use it to join to any other partner-facing table.
• industry_gic2_code – Two-digit industry classification (Global Industry Classification Standard). Helpful when segmenting business clients vs. individuals.
• partner_class_code – High-level customer type (e.g., I = Individual, P = Partnership). Matches the codes in partner_role.
• partner_gender – M, F, or blank; not all entities have a gender.
• partner_name – Display name. Individuals are in First Last order; organisations are free text.
• partner_phone_number – Swiss-style number formatted with spaces for readability.
• partner_birth_year – Stored as YYYY-MM-DD when known; blank means the birth date is unavailable or irrelevant for entities.
• partner_address – Multi-line string (street on line 1, postal code + city on line 2).
• partner_open_date / partner_close_date – When the partner was first onboarded and, if relevant, offboarded. Close date blank = still a client.
• **Safe assumptions:**
• partner_id is the join key for all partner-level tables.
• Addresses are stored as multi-line fields separated by newline characters; preserve them when exporting.
• Birth year is stored as YYYY-MM-DD when known; blanks indicate missing data.
• partner_class_code aligns with the values seen in partner_role.partner_class_code.

## partner_country.csv

• **Scope:** Country-level metadata per partner, supporting domicile vs. tax residence vs. mailing distinctions.
• **Column glossary:**
• partner_id – Join key; references partner.csv.
• country_name – Plain-language country.
• partner_country_status_code – 1 means the row is the current active fact; 0 or blank marks a historical state.
• country_type – Specifies what the country represents (e.g., domicile = where the client lives, tax = where taxes are filed).

- **Safe assumptions:**
  - partner_country_status_code = 1 marks the current active row for that country_type.
  - country_type values include domicile, tax, or mailing; new types may appear over time.
  - Every partner has at least one domicile row.

## partner_role.csv

- **Scope:** Canonical link table showing how partners participate in business relationships or reference other partners.
- **Column glossary:**
  - partner_id – Person or organisation taking part in the role.
  - entity_type – Tells you what kind of object is on the other side of the link. BR stands for business relationship; future releases may add LE (legal entity) or ACCT (account).
  - entity_id – Identifier of that other object (e.g., the br_id).
  - relationship_start_date / relationship_end_date – Time window for the role. If end date is blank, the person currently holds the role.
  - br_type_code – Short mnemonic for the role type (VP = single individual, JO = joint, etc.).
  - associated_partner_id – Populated when a role references another partner, such as an authorised signatory.
  - partner_class_code – Mirrors the class code stored in partner.csv for convenience.
- **Safe assumptions:**
  - entity_type = 'BR' rows connect partners to business_rel.br_id; additional entity types (e.g., LE for legal entity) may be added later.
  - br_type_code follows Swiss private banking conventions (VP = Individual, JO = Joint, etc.).
  - associated_partner_id is populated for joint or authorised-representative relationships.
  - Date fields capture the validity window for the role; blank relationship_end_date means active.

## transactions.csv

- **Scope:** Line-level ledger in double-entry form, covering both intra-bank and cross-border payments.
- **Column glossary:**
  - Transaction ID – Unique identifier shared by the debit and credit legs of the same movement. Use this to group both sides of the entry.
  - Debit/Credit – Text value (debit or credit) telling you whether the row removes money from the Account ID or adds to it.
  - Account ID – Internal account that experienced the movement. Join back to account.csv for currency and ownership details.
  - Amount – Positive decimal showing how much money moved. Combine with the debit/credit flag to determine the sign.
  - Balance – Account balance immediately after the transaction finished, expressed in the same Currency column.
  - Currency – ISO code of the account and the transaction (CHF/EUR/USD, etc.).
  - Date – Timestamp down to the minute (YYYY-MM-DD HH:MM). Already in UTC; no timezone math needed.
  - Transfer_Type – Transaction code categorising the payment (e.g., MTACCD for account transfer). Use for classification or scenario filtering.

- counterparty_Account_ID – If present, the other internal account that was credited/debited in the same bank (mirrors double-entry accounting).
- ext_counterparty_Account_ID / ext_counterparty_country – Populated when the other side lives outside the bank. Use these fields to study cross-border flows.

- **Safe assumptions:**
  - Each Transaction ID is unique; debit and credit legs share the same ID but different Debit/Credit values.
  - Amounts are stored as positive numbers; use the debit/credit indicator to infer the sign.
  - Balance is the post-transaction available balance denominated in Currency.
  - Timestamps are UTC and formatted as YYYY-MM-DD HH:MM (24-hour clock).
  - counterparty_Account_ID links to another internal account when populated; otherwise, use the external account fields (ext_counterparty_*).

## General modelling assumptions

- All identifiers are UUIDv4 strings and can be treated as opaque keys.
- Dates use ISO 8601 (YYYY-MM-DD), while datetimes are in UTC with minute precision.
- Monetary amounts are decimal with two places; treat them as high-precision decimals rather than floats when possible.
- The dataset is internally consistent: every foreign key listed above has a matching parent record.
- Personally identifiable information (PII) is fictional but formatted to resemble Swiss data (names, addresses, phone numbers, IBANs, etc.).

Use this README locally (Markdown) or pair it with the generated PDF for stakeholders who prefer fixed documents.