# Agnostic Census Transformation – Master Process Map & Auto‑Detection Design

## 0) Legend (Flowchart / BPMN‑lite)

* **Start/End**: rounded rectangle
* **Process**: rectangle
* **Decision / Gateway**: diamond (Yes/No)
* **Data I/O**: parallelogram
* **Predefined Process (subroutine)**: rectangle with double‑struck edges
* **Data Store**: cylinder
* **Connector / Off‑page**: circle
* **Annotation**: bracketed note

## 1) High‑Level Flow (overview)

(Start) → Ingest File(s) → Profile & Fingerprint → Detect Structure → Normalize → Field Canonicalization → Report Builder → Export → (End)

## 2) Detailed Flow (with decisions)

1. **Start**
2. **Ingest Source** (Data I/O)
   * Accept: Excel (.xlsx/.xls) or CSV
   * Multi‑sheet handling: user selects sheet or auto‑pick largest tabular region
   * Establish table boundaries; convert to Table
3. **Profile & Fingerprint** (Process)
   * Table.Profile (count, null%, min/max/mean, distinct count)
   * Sample first N rows for regex patterns (SSN, DOB, ZIP, State, Phone, Email)
   * Header analysis: tokenize, stem (e.g., dep1\_first, First Name (Dependent 2))
   * **Output**: column fingerprints + dataset fingerprint
4. **Detect Person Grain (Employee vs Dependent)** (Decision)
   * Heuristics: presence of employee SSN vs dependent SSN columns; relationship column; repeated person blocks
   * If missing, infer via value patterns (e.g., Relationship ∈ {Employee, Dependent, Spouse, Child})
5. **Detect Structure A – Row vs Column based** (Decision)
   * **Signals for Column‑based**:
     + Repeating header stems with numeric suffixes: Dep1\_First, Dep2\_First, …
     + Blocks (k consecutive columns) repeating M times (block size ≈ 5–10 typical)
     + High null‑bands per block except within populated dependent counts
   * **Signals for Row‑based**:
     + One set of person columns; multiple rows per employee; presence of Relationship values
6. **Detect Structure B – Plan Grain** (Decision)
   * **Plan‑Per‑Row** (wide per person): repeating plan blocks: Dental\_\*, Vision\_\*, …
   * **Plan‑Per‑Record** (long): duplicate person keys across rows with distinct PlanName / Coverage values
7. **Choose Normalization Strategy** (Gateway)
   * If Column‑based dependents → **Subroutine: Unpivot Dependents**
   * If Plan‑Per‑Record → **Subroutine: Collapse Plans to Wide** (pivot or grouped merge)
   * If Plan‑Per‑Row (repeating plan blocks) → **Subroutine: Unpivot Plan Blocks** then pivot/aggregate as needed
   * Multiple can apply; order = Dependents → Plans
8. **Canonicalize Field Domains** (Process)
   * Map headers to canonical names with multi‑signal scoring:
     + Header text similarity (TF‑IDF / n‑grams / synonyms)
     + Regex/domain checks (SSN, DOB, Email, Phone, ZIP, State, Gender, Relationship)
     + Value distribution signatures (e.g., Gender ≈ {M,F}; State ≈ 50 USPS codes)
     + Position hints (neighbor columns like First/Last near each other)
   * Produce **Candidate Mapping Table** with confidence score per suggestion
9. **User Review & Overrides** (Data I/O)
   * Side‑by‑side preview; highlight low‑confidence fields (< threshold)
   * Allow quick remap via dropdown; accept defaults en masse
10. **Report Builder** (Predefined Process)
    * Header row with dropdowns of **canonical** (or raw) fields
    * Saved presets per carrier; validate required fields; optional formatting rules
11. **Validation & QA** (Process)
    * Key checks: Person count before/after, dependent counts, NULL audits for required fields, SSN/DOB formats
    * Exception report with row IDs and reasons
12. **Export** (Data I/O)
    * Create clean workbook → values only; optional CSV
    * Log: transformation steps, mapping used, timestamps
13. **End**

## 3) Auto‑Detection Heuristics (concrete)

### 3.1 Detect Column‑Based Dependents

* **Header stem clustering**: group headers by stem ignoring trailing digits and keywords (e.g., dep, dependent, child, spouse + ordinal markers 1..10).
* **Block detection**: sliding window to find repeating sequences of fields (e.g., First, Last, DOB, SSN, Relationship, Plan repeated M times). Compute block size consensus; require ≥3 repeated fields across ≥2 blocks.
* **Null‑band pattern**: later dependent blocks show higher null%; monotone increase by block index.

### 3.2 Detect Plan Grain

* **Plan‑Per‑Record (long)**:
  + Duplicate rate: high duplicate count on PersonKey with distinct PlanName/CoverageType/Product values.
  + Row ratio: rows/persons ≫ 1 (e.g., >1.3).
* **Plan‑Per‑Row (wide)**:
  + Repeating plan stems in headers: Dental\_\*, Vision\_\*, Life\_\*, STD\_\*, LTD\_\*, Medical\_\*.
  + Each stem forms a consistent block (e.g., 4–12 columns) with low within‑block null%.

### 3.3 PersonKey Inference (robust when names vary)

* Priority order to form **PersonKey**:
  1. SSN (exact 9 digits, allow dashes; de‑dup using latest DOH if needed)
  2. EmployeeID/MemberID (≥6 alphanumerics)
  3. Composite: FirstName + LastName + DOB normalized
  4. Fallback: stable hash of salient columns

### 3.4 Relationship Inference

* Use regex/synonyms: {Employee, Emp, Sub, Subscriber} → Employee; {Spouse, Domestic Partner} → Spouse; {Child, Dep, Dependent} → Child.
* If absent, infer via position: top row of a person group in row‑based; or presence of employee‑only fields (e.g., DOH).

### 3.5 Field Canonicalization Signals

* **Regex/domain**:
  + SSN: ^(?!000|666|9)\d{3}[- ]?\d{2}[- ]?\d{4}$
  + DOB/DOH: date parse success rate & range (1900–today)
  + ZIP: ^\d{5}(-\d{4})?$, State: 50 USPS codes, Email, Phone
* **Distribution**: low cardinality sets → enums (Gender, Relationship, PlanTier)
* **Text similarity**: cosine similarity on tokenized headers with synonym lists (e.g., First Name ~ Given, FName)
* **Neighbor cues**: First/Last/MI cluster; Address1 near City/State/ZIP
* **Score** = weighted sum; track **Confidence** per field

## 4) Normalization Subroutines (algorithm outlines)

### 4.1 Unpivot Dependents (Column‑based → Row‑based)

* Identify dependent blocks and their field order template.
* For each row, iterate blocks 1..N:
  + If any of the block’s key fields (e.g., First/Last/DOB/SSN) are non‑null → emit a dependent row tagged to PersonKey.
* Result: one employee row + K dependent rows directly below.

### 4.2 Collapse Plans to Wide (Plan‑Per‑Record → Wide per Person)

* Group by PersonKey (+ DepKey when dependents exist).
* Pivot on PlanType (canonical set: Medical, Dental, Vision, Life, STD, LTD, etc.)
* For each plan, project configured attributes (e.g., Carrier, Product, Tier, EffectiveDate) → prefixed columns Vision\_Carrier, Vision\_Tier, …

### 4.3 Unpivot Plan Blocks (Plan‑Per‑Row → Long, then recombine)

* Detect repeating plan stems; for each stem build a map of its attributes.
* Unpivot each block to long with fields (PersonKey, DepKey?, PlanType, Attr, Value) then pivot back to the desired wide structure.

## 5) User Experience (Excel‑centric)

* **Admin tab**: source selection (open workbook dropdown or file picker); options for “Detect structure automatically” + manual override toggles.
* **Profiler pane**: show signals with confidence, e.g., Structure: Column‑based Dependents (92%), Plan grain: per‑record (88%).
* **Mapping pane**: Candidate Mapping Table (Raw Header → Canonical Field, Confidence, Override dropdown)
* **Report Builder tab**: top row dropdowns of **canonical** fields; carrier presets (load/save); required field validator.
* **QA report**: summary stats + exception rows with reasons; link to navigate back to source rows.

## 6) Implementation Notes (Power Query + VBA)

* **Power Query (M)**:
  + Use Table.Profile / Table.Schema for fingerprints
  + Header stem clustering via Text.RegexReplace to strip ordinals (\_?dep(\d+)\_? → \_dep\_)
  + Block detection using column index differences; maintain candidate block sizes and vote
  + Unpivot via Table.UnpivotColumns and custom column to carry block index
  + PersonKey builder function with layered fallbacks
  + Plan pivot using Table.Pivot with aggregators like List.FirstNonNull
* **VBA helpers**:
  + File selection UX; open‑workbook list; trigger PQ refresh chains
  + Export routine to values‑only workbook; log steps and mappings
  + Preset management (carrier layouts) stored in hidden sheet as JSON‑like key/values

## 7) Governance & Safety

* **Deterministic logging**: write a step log (who/when/version, detection decisions, overrides)
* **Reproducibility**: store the auto‑mapping + overrides alongside export
* **PII handling**: optional SSN masking in UI; strict format validation before export

## 8) Starter Flowchart (text)

[Start]  
 → (Load File)  
 → (Profile Columns)  
 → <Row vs Column?>  
 ↘Column→ (Unpivot Dependents)  
 ↘Row → (No change)  
 → <Plan Grain?>  
 ↘Per‑Record→ (Pivot Plans to Wide)  
 ↘Per‑Row → (Unpivot Plan Blocks → Re‑pivot)  
 → (Canonicalize Fields & Build Mapping)  
 → (User Review / Overrides)  
 → (Report Builder + Carrier Presets)  
 → (QA & Exceptions)  
 → (Export Workbook)  
[End]

## 9) Next Steps

* I can convert this into a polished one‑page diagram (Visio/Mermaid/Draw.io) and a scaffolding Excel file with:
  + Admin, Profiler, Mapping, Report Builder, QA tabs
  + Power Query stubs + VBA export/preset modules
* Provide a sample dataset pack to validate each branch (row vs column; plan long vs wide).