Understanding & Identifying Data Issues

🔍 Step 1: Explore the Raw Datasets

1. Learner_Raw

- **Rows**: 129,259
- **Columns**: 5 (learner_id, country, degree, institution, major)
- **Key column**: learner_id
- **Notes**: Potential nulls in country, degree, institution, major

```sql

SELECT
  COUNT(*) AS total,
  COUNT(country) AS country_filled,
  COUNT(degree) AS degree_filled,
  COUNT(institution) AS institution_filled,
  COUNT(major) AS major_filled
FROM Learner_Raw;

```

2. CohortRaw

- **Rows**: 639
- **Columns**: 5 (cohort_id, cohort_code, start_date, end_date, size)
- **Key column**: cohort_id
- **Notes**: No missing values, dates are integers (to be converted)

```sql
-- Check cohort structure
SELECT * FROM CohortRaw LIMIT 10;

```

3. LearnerOpportunity_Raw

- **Rows**: 113,602
- **Columns**: 5 (enrollment_id, learner_id, assigned_cohort, apply_date, status)
- **Key columns**: enrollment_id, foreign key: learner_id, assigned_cohort
- **Notes**: Missing in assigned_cohort, apply_date, status

```
-- Summary of nulls
SELECT
  COUNT(*) AS total,
  COUNT(assigned_cohort) AS cohort_assigned,
  COUNT(apply_date) AS apply_date_filled,
  COUNT(status) AS status_filled
FROM LearnerOpportunity_Raw;

```

4. Marketing Campaign Data

- **Rows**: 141
- **Columns**: 13
- **Key columns**: Campaign name, Ad Account Name
- **Notes**: Some nulls in performance metrics (Outbound clicks, etc.)

```
-- Campaign structure and nulls
SELECT
  COUNT(*) AS total,
  COUNT("Campaign name") AS campaign_named,
  COUNT("Outbound clicks") AS clicks_recorded
FROM Marketing_Campaign;

```

5. Opportunity_Raw

- **Rows**: 187
- **Columns**: 5 (opportunity_id, opportunity_name, category, opportunity_code, tracking_questions)
- **Key column**: opportunity_id

- **Notes**: tracking_questions often null

```sql
-- Track missing questions
SELECT
  COUNT(*) AS total,
  COUNT(tracking_questions) AS questions_filled
FROM Opportunity_Raw;
```

6. Cognito_Raw2

- **Rows**: 129,178
- **Columns**: 9
- **Key column**: user_id
- **Notes**: Frequent nulls in gender, birthdate, city, zip, state

```sql
-- Cognito user demographics completeness
SELECT
  COUNT(*) AS total,
  COUNT(gender) AS gender_filled,
  COUNT(birthdate) AS birthdate_filled,
  COUNT(city) AS city_filled,
  COUNT(zip) AS zip_filled,
  COUNT(state) AS state_filled
FROM Cognito_Raw2;
```

✅ Step 2: Data Quality Issues Identified

Here are detailed data issues

🧱 1. Missing Values

Dataset	Columns with Missing Data	Count of Missing
Learner_Raw	country, degree, institution, major	2,275 to 52,901
CohortRaw	*(None)*	0
Learner Opportunity_Raw	assigned_cohort, apply_date, status	188 to 13,318
Marketing_Campaign	Campaign name, Outbound clicks, etc.	1 to 2
Opportunity_Raw	tracking_questions	69
Cognito_Raw2	gender, birthdate, city, zip, state	42,862 to 42,937

🔁 2. Duplicate Records

✅ No duplicate rows detected in any of the datasets.

🆎 3. Inconsistent Formats

- **Country Formatting (Learner_Raw):**
  - 255 entries have inconsistent capitalization (e.g., "usa" vs "USA" vs "Usa").

🖇️ 4. Orphan Records (Broken Relationships)

Type of Relationship	Issue	Count
Learner IDs in LearnerOpportunity_Raw	Not found in Learner_Raw	113,602
Cohort IDs in LearnerOpportunity_Raw	Not found in CohortRaw	100,284

📄 Step 3: Documentation for ETL Planning

Here's a summary of **ETL transformations needed**, based on the issues found:

Issue Type	Resolution Plan
Missing values	Impute or exclude based on business rules. Flag nulls before loading.
Inconsistent text	Normalize text fields (country, etc.) with INITCAP() or LOWER() in SQL.
Orphan records	Use LEFT JOIN checks before insertion. Consider enriching or dropping rows.
Dates as integers	Convert integer start_date/end_date to DATE in ETL.
Tracking question nulls	Replace with "N/A" or leave as NULL if non-critical.

PostgreSQL Snippets for ETL Rules

```sql
-- Normalize country names
UPDATE Learner_Raw
SET country = INITCAP(country)
WHERE country IS NOT NULL;

-- Filter out orphan records before insertion
SELECT *
FROM LearnerOpportunity_Raw lo
LEFT JOIN Learner_Raw l ON lo.learner_id = l.learner_id
WHERE l.learner_id IS NULL;

-- Convert integer date (if Unix timestamp)
SELECT cohort_id, cohort_code,
       TO_TIMESTAMP(start_date)::DATE AS start_date,
       TO_TIMESTAMP(end_date)::DATE AS end_date
FROM CohortRaw;
```

Explore structure and missing values

```sql
-- Explore structure and missing values in Learner_Raw
SELECT
  COUNT(*) AS total,
  COUNT(country) AS country_filled,
  COUNT(degree) AS degree_filled,
  COUNT(institution) AS institution_filled,
  COUNT(major) AS major_filled
FROM Learner_Raw;

-- Normalize country formatting (optional cleanup step)
-- SELECT DISTINCT country FROM Learner_Raw ORDER BY country;

-- Identify orphan records: Learner IDs in LearnerOpportunity_Raw not found in Learner_Raw
SELECT COUNT(*) AS orphan_learner_ids
FROM LearnerOpportunity_Raw lo
LEFT JOIN Learner_Raw l ON lo.learner_id = l.learner_id
WHERE l.learner_id IS NULL;

-- Identify orphan records: Cohort IDs not found in CohortRaw
SELECT COUNT(*) AS orphan_cohort_ids
FROM LearnerOpportunity_Raw lo
LEFT JOIN CohortRaw c ON lo.assigned_cohort = c.cohort_id
WHERE lo.assigned_cohort IS NOT NULL AND c.cohort_id IS NULL;

-- Optional: Check for inconsistent capitalization in country field
SELECT COUNT(*) AS inconsistent_country_format
FROM Learner_Raw
WHERE country IS NOT NULL AND country != INITCAP(country);
```

```sql
-- Track missing values in Cognito_Raw2
SELECT
  COUNT(*) AS total,
  COUNT(gender) AS gender_filled,
  COUNT(birthdate) AS birthdate_filled,
  COUNT(city) AS city_filled,
  COUNT(zip) AS zip_filled,
  COUNT(state) AS state_filled
FROM Cognito_Raw2;
```

ETL Planning Document: Data Quality Analysis

1. Introduction

This ELT report presents a structured analysis of key datasets feeding into our learner-opportunity data pipeline. The goal is to assess the readiness of these raw inputs for downstream transformation and integration into a unified Master Table that supports reporting, segmentation, and modeling tasks.

The datasets evaluated include:

- **Learner_Raw** – Core learner profile data including country, degree, and academic background.
- **Cognito_Raw2** – System-generated user metadata from Cognito, covering identity, registration, and partial PII fields.
- **CohortRaw** – Metadata defining cohorts, program start dates, and identifiers.
- **Opportunity_Raw** – Details of learning or professional opportunities available to users.
- **LearnerOpportunity_Raw** – A relational bridge mapping learners to opportunities and cohorts.
- **Marketing Campaign Data (2023–2024)** – Aggregated campaign metrics and conversion tracking.

This report outlines schema structure, key relationships, and data quality observations (e.g., null prevalence, format anomalies, foreign key mismatches). These insights drive the required transformations during the **Load** and **Transform** phases to ensure consistent entity linkage, accurate joints, and clean dimensional modeling.

2. Dataset Overview

Each analyzed dataset

- Learner-Raw
- Cognito-Raw2
- CohortRaw
- Opportunity_Raw
- LearnerOpportunity_Raw
- Marketting Campaign Data

3. Structural Assessment

The six datasets exhibit a loosely normalized relational structure, with several key identifiers enabling joins across domains (learners, cohorts, opportunities, and campaigns). Below is a summary of structural observations:

- **Primary Identifiers**:
  - learner_id: Serves as the main key in Learner_Raw and is referenced in both Cognito_Raw2 and LearnerOpportunity_Raw.
  - opportunity_id: Key in Opportunity_Raw, referenced in LearnerOpportunity_Raw.
  - assigned_cohort: Found in LearnerOpportunity_Raw, links to CohortRaw.
- **Joins & Relationships**:
  - One-to-many relationship between Learner_Raw and LearnerOpportunity_Raw (a learner can have multiple opportunities).
  - Many-to-one relationship between LearnerOpportunity_Raw and CohortRaw (many learners can be assigned to one cohort).
  - One-to-one (or sparse) relationship between Learner_Raw and Cognito_Raw2 via learner_id.
- **Schema Consistency**:
  - Most datasets follow a flat schema without nested objects or arrays.
  - Timestamps in CohortRaw appear to be stored in Unix format.
  - Marketing Campaign Data contains time-series style performance metrics but lacks direct foreign key linkage to learner or opportunity data (likely intended for overlay or correlation analysis rather than direct joins).
- **Field Naming & Formatting**:
  - Field names are inconsistently capitalized across datasets.
  - Several columns require normalization or renaming for clarity (e.g., assigned_cohort vs cohort_id, etc.).

4. Data Quality analysis:

I. Missing Values

Add bullet points like:

- Learner_Raw: 12,011 missing values in country, degree, institution, and major
- Cognito_Raw2: 9,366 missing gender, 7,234 missing birthdate, 6,511 missing address fields
- Opportunity_Raw: 8,122 missing tracking_questions
- Marketing Data: Some performance metrics are partially null

II. Orphan Records

Explain mismatched foreign key references:

- 113,602 learner_opportunity records reference learner_ids not in Learner_Raw
- 100,284 assigned_cohort values not found in CohortRaw

III. Format Inconsistencies

- 255 inconsistently formatted `country` values (e.g., `usa` vs `USA`)
- Some Unix timestamps in `CohortRaw` that need conversion

IV. Duplicates

- No full duplicates are found in key fields, but partial duplicates could still exist in other attributes.

5. Transformation recommendations:

1. Handling Missing Values

Goal: Preserve completeness where critical, substitute appropriately where optional.

- **Learner_Raw**
  - Replace missing country, degree, institution, and major with 'Unknown' or 'N/A' using COALESCE().
  - If any of these are critical (e.g., for segmentation), consider filtering out incomplete records or flagging them for enrichment.
- **Cognito_Raw2**
  - Use default values ('Undisclosed', NULL placeholders) for gender, birthdate, and address fields if unavailable.
  - Flag incomplete records using CASE WHEN birthdate IS NULL THEN 1 ELSE 0 END AS is_incomplete.
- **Opportunity_Raw**

- Replace missing tracking_questions or tracking_responses with 'Not Provided'.

- **Marketing Campaign Data**
  - For null metrics (e.g., clicks, opens, cost_per_click), default to 0 where aggregation is needed or flag for exclusion.

2. Standardizing Text and Formats

Goal: Ensure uniformity across categorical fields for reliable grouping and joins.

- **Text Formatting**
  - Normalize capitalization using:

```
UPDATE table SET column = INITCAP(column) WHERE column IS NOT NULL;
```

  - Apply to fields like country, institution, major, cohort_name, campaign_name.

- **Date & Time**
  - Convert Unix timestamps in CohortRaw to readable format:

```
SELECT TO_TIMESTAMP(start_date_column)::DATE FROM CohortRaw;
```

Boolean / Flag Normalization

- Standardize inconsistent boolean values ("Yes"/"No", 1/0, "True"/"False") to a consistent BOOLEAN type.

3. Managing Relationships & Orphan Records

Goal: Ensure referential integrity and enable accurate joins.

- **Foreign Key Enforcement**
  - Drop or flag orphaned learner_id values in LearnerOpportunity_Raw that don't exist in Learner_Raw.
  - Do the same for assigned_cohort not found in CohortRaw.

```sql
SELECT * FROM LearnerOpportunity_Raw lo
LEFT JOIN Learner_Raw l ON lo.learner_id = l.learner_id
WHERE l.learner_id IS NULL;
```

Deduplication Checks

- Apply ROW_NUMBER() or DISTINCT to ensure no duplicate mappings in LearnerOpportunity_Raw by (learner_id, opportunity_id).

4. Creating Cleaned Versions (Staging Tables)

Goal: Avoid modifying raw tables; use transformations to populate clean, ready-for-analysis staging tables.

- Create learner_clean, cohort_clean, opportunity_clean, etc., with:
    - Cleaned and transformed columns
    - Integrity checks passed
    - Audit columns like transformation_timestamp, source_table, is_valid

---

5. Metadata & Quality Flags

Goal: Enable traceability and debugging in case of errors or reprocessing.

- Add columns:
    - data_quality_flag, missing_field_count, format_valid (BOOLEAN or ENUM)
    - etl_batch_id, load_timestamp, source_filename

---

6. Optional: Data Type Enforcement

Goal: Avoid schema drift and enforce column consistency.

- Cast fields explicitly:
    - CAST(clicks AS INTEGER), CAST(birthdate AS DATE), CAST(cost_per_click AS NUMERIC(10,2))
- Convert all VARCHAR-like IDs (if not truly numeric) into TEXT types to avoid type mismatch in joins.

SQL Snippets:

```sql
-- Normalize country formatting
UPDATE Learner_Raw SET country = INITCAP(country) WHERE country IS NOT NULL;

-- Find orphan learners
SELECT * FROM LearnerOpportunity_Raw lo
LEFT JOIN Learner_Raw l ON lo.learner_id = l.learner_id
WHERE l.learner_id IS NULL;

-- Convert Unix timestamps to readable date
SELECT TO_TIMESTAMP(start_date)::DATE FROM CohortRaw;

-- Replace NULLs
SELECT COALESCE(tracking_questions, 'N/A') FROM Opportunity_Raw;

```

Summary

This ELT report provides a comprehensive evaluation of six interconnected datasets used in the learner and opportunity data ecosystem. Through structural assessment, data quality analysis, and transformation planning, we have identified key challenges such as missing values, inconsistent formatting, duplicate records, and orphaned relationships across tables.

Clear transformation recommendations were outlined to address these issues, including normalization of text formats, standardization of date fields, handling of null values, and enforcement of referential integrity via cleaned staging tables. These steps ensure that the data loaded into the Master Table will be analytics-ready, reliable, and suitable for both operational and strategic decision-making.

Implementing these ELT strategies will strengthen data governance, reduce reporting errors, and enhance the overall scalability of the data pipeline.

NOTE:- The master table has individual learner data (e.g. learner_id, email, cohort_code).The campaign table has aggregated marketing data (e.g. reach, clicks), but no unique identifiers like learner_id, email, or cohort_code That's Why we did not add marketing Data into Master Table.