Building the Master Table & ETL Process

Definition:

The master table centralizes learner information, their cohort and opportunity assignments, and associated campaign performance, also give a detailed analysis of the relationship between datasets.

```sql
CREATE TABLE master_learner_summary (
    learner_id TEXT PRIMARY KEY,
    learner_name TEXT,
    email TEXT,
    birthdate DATE,
    gender TEXT,
    address TEXT,

    country TEXT,
    degree TEXT,
    institution TEXT,
    major TEXT,

    cohort_name TEXT,
    cohort_type TEXT,
    cohort_start_date DATE,
    cohort_end_date DATE,

    opportunity_id TEXT,
    opportunity_name TEXT,
    opportunity_stage TEXT,
    opportunity_status TEXT,
    opportunity_source TEXT,

    campaign_name TEXT,
    clicks INTEGER DEFAULT 0,
    opens INTEGER DEFAULT 0,
    cost_per_click NUMERIC(10,2),

    etl_batch_id TEXT,
    load_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    record_status TEXT DEFAULT 'active'
);

```

Figure 1 Sql

Step 1: Plan the Master Table Structure

1. Key Columns for Master Table

To create a holistic view, the Master Table will combine learner, cohort, opportunity, and campaign data.

Primary Key (PK):

- learner_id (from Learner_Raw)

Essential Fields:

- learner_name, email, country, institution, degree, major
- birthdate, gender, address (from Cognito)
- cohort_name, start_date, end_date, cohort_type (from CohortRaw)
- opportunity_name, opportunity_status, stage, source (from Opportunity_Raw)
- campaign_name, clicks, opens, cost_per_click (from Marketing Campaign)

Relationships and Keys

Table	Key(s)	Relationship
Learner_Raw	learner_id (PK)	1:M with LearnerOpportunity_Raw
LearnerOpportunity_Raw	learner_id, opportunity_id (Composite PK)	Bridge table for M:M
Opportunity_Raw	opportunity_id (PK)	FK in bridge table
CohortRaw	cohort_id or name (PK)	FK via assigned_cohort
Cognito_Raw2	email or sub (PK)	Joined on email with Learner_Raw
Marketing Campaign	account_name, campaign_name	Join via learner/campaign metadata

2. Master Table structure

Column	Data Type	Notes
learner_id	TEXT	PK
learner_name	TEXT	From Cognito (if available) or Learner
email	TEXT	Joined across Cognito + Learner

Column	Data Type	Notes
country	TEXT	Standardized
birthdate	DATE	Converted from text
degree, institution, major	TEXT	Standardized formatting
cohort_name	TEXT	FK from Cohort
start_date, end_date	DATE	Converted from Unix timestamp
opportunity_name	TEXT	Joined from Opportunity
opportunity_status	TEXT	Standardized
campaign_name	TEXT	Joined based on opportunity or learner info
clicks, opens	INTEGER	Nulls default to 0
cost_per_click	NUMERIC(10,2)	Rounded and standardized

Step 2: Extract Data from Source Tables

Identify Relevant Tables and Columns

Dataset	Columns Extracted
Learner_Raw	learner_id, country, degree, institution, major
Cognito_Raw2	email, birthdate, gender, address
LearnerOpportunity_Raw	learner_id, opportunity_id, assigned_cohort
Opportunity_Raw	opportunity_id, opportunity_name, stage, status, source

Dataset	Columns Extracted
CohortRaw	cohort_name, start_date, end_date, cohort_type
Marketing Campaign Data	campaign_name, account_name, clicks, opens, cost_per_click

Join Conditions

- Join Learner_Raw ⇌ LearnerOpportunity_Raw via learner_id
- Join LearnerOpportunity_Raw ⇌ Opportunity_Raw via opportunity_id
- Join LearnerOpportunity_Raw ⇌ CohortRaw via assigned_cohort
- Join Learner_Raw ⇌ Cognito_Raw2 via email
- Join learner_id or campaign_name ⇌ Marketing Campaign Data via appropriate mapping

---

Step 3: Transform Data for Consistency & Accuracy

Cleaning Steps

Transformation Type	Description
Missing Values	Replace nulls with 'Unknown' or 'Not Provided'; flag incomplete rows
De-duplication	Use DISTINCT, ROW_NUMBER() to remove repeats
Text Standardization	Use INITCAP() for name fields, LOWER() for identifiers
Date Conversion	Convert timestamps with TO_TIMESTAMP(unix_time)::DATE
Numeric Cleanup	Round off cost_per_click, default missing metrics to 0
Flagging Rules	Mark invalid or orphan records for audit or exclusion

---

Step 4: Load Cleaned Data into Master Table

Loading Strategy

MASTER TABLE AND ETL PROCESS

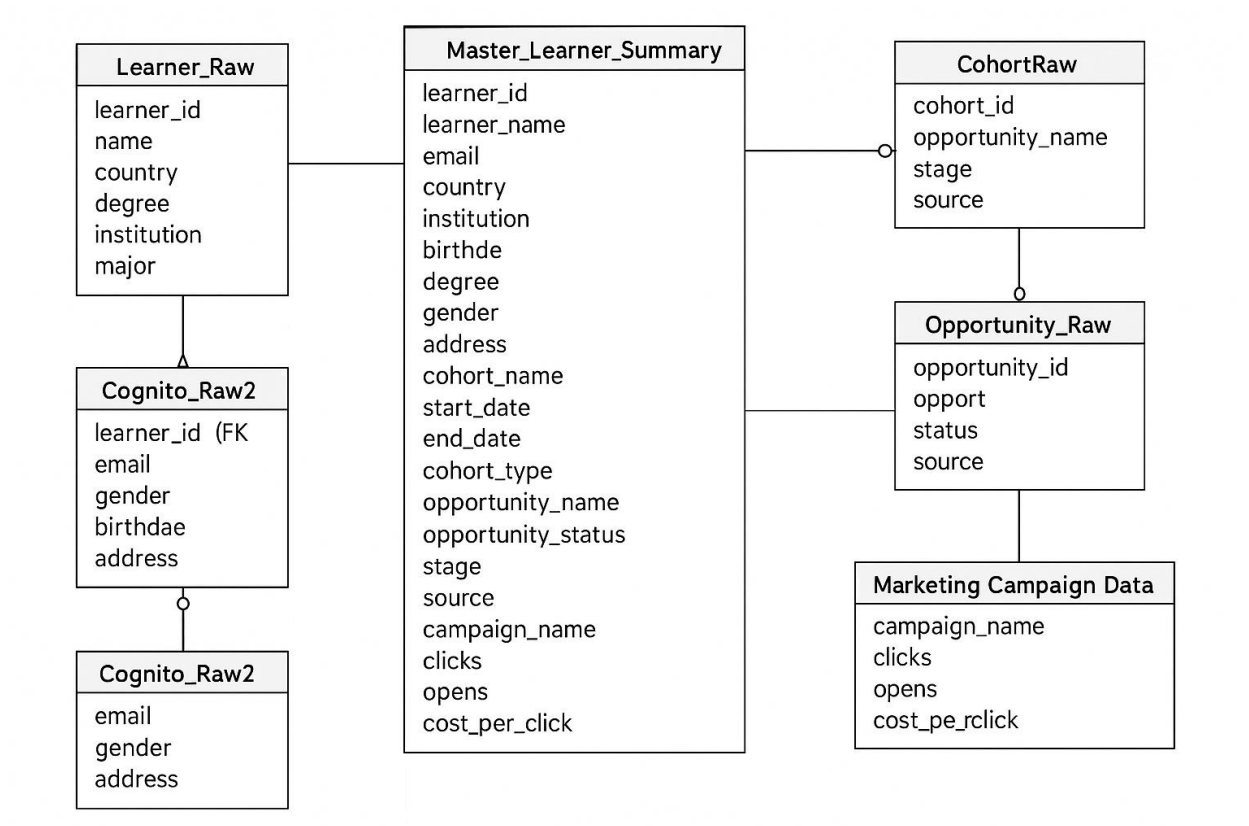
- Load clean transformed data into master_learner_summary
- Validate PK/FK mappings during load using NOT EXISTS checks
- Add ETL tracking columns: etl_batch_id, load_timestamp, record_status

```sql
INSERT INTO master_learner_summary (
    learner_id,
    learner_name,
    email,
    birthdate,
    gender,
    address,
    country,
    degree,
    institution,
    major,
    cohort_name,
    cohort_type,
    cohort_start_date,
    cohort_end_date,
    opportunity_id,
    opportunity_name,
    opportunity_stage,
    opportunity_status,
    opportunity_source,
    campaign_name,
    clicks,
    opens,
    cost_per_click,
    etl_batch_id,
    load_timestamp,
    record_status
)
SELECT
    lr.learner_id,
    COALESCE(cr.full_name, 'Unknown') AS learner_name,
    LOWER(lr.email) AS email,
    TO_DATE(cr.birthdate, 'YYYY-MM-DD') AS birthdate,
    INITCAP(cr.gender) AS gender,
    cr.address,
    INITCAP(lr.country) AS country,
    INITCAP(lr.degree) AS degree,
    INITCAP(lr.institution) AS institution,
    INITCAP(lr.major) AS major,

    ch.cohort_name,
    ch.cohort_type,
    ch.start_date,
    ch.end_date,

    opp.opportunity_id,
    opp.opportunity_name,
    opp.stage,
    opp.status,
    opp.source,

```

```
    load_timestamp,
    record_status
)
SELECT
    lr.learner_id,
    COALESCE(cr.full_name, 'Unknown') AS learner_name,
    LOWER(lr.email) AS email,
    TO_DATE(cr.birthdate, 'YYYY-MM-DD') AS birthdate,
    INITCAP(cr.gender) AS gender,
    cr.address,
    INITCAP(lr.country) AS country,
    INITCAP(lr.degree) AS degree,
    INITCAP(lr.institution) AS institution,
    INITCAP(lr.major) AS major,

    ch.cohort_name,
    ch.cohort_type,
    ch.start_date,
    ch.end_date,

    opp.opportunity_id,
    opp.opportunity_name,
    opp.stage,
    opp.status,
    opp.source,

    mc.campaign_name,
    COALESCE(mc.clicks, 0),
    COALESCE(mc.opens, 0),
    ROUND(COALESCE(mc.cost_per_click, 0.0)::numeric, 2),

    gen_random_uuid()::text AS etl_batch_id,
    CURRENT_TIMESTAMP,
    'active' AS record_status

FROM stg_learner_raw lr
LEFT JOIN stg_cognito_raw cr ON LOWER(lr.email) = LOWER(cr.email)
LEFT JOIN stg_learner_opportunity lo ON lr.learner_id = lo.learner_id
LEFT JOIN stg_opportunity_raw opp ON lo.opportunity_id = opp.opportunity_id
LEFT JOIN stg_cohort_raw ch ON lr.assigned_cohort = ch.cohort_id
LEFT JOIN stg_marketing_campaign mc ON mc.email = lr.email
WHERE lr.learner_id IS NOT NULL;

```

Virtual representation of master table

MASTER TABLE AND ETL PROCESS



ETL Automation Suggestions

- Use SQL procedures or Python (e.g., Airflow, dbt) to orchestrate:
    - Extract queries per dataset
    - Transform logic (joins, formatters, NULL handling)
    - Insert into Master Table with INSERT INTO ... SELECT ...
    - Log file generation and error handling

To develop an ETL workflow that extracts, transforms, and loads data in the correct order.

ETL Workflow for master_learner_Table:

STEP 1: Extracting data (Raw → Staging)

MASTER TABLE AND ETL PROCESS

Extract data from the raw source tables into staging tables for transformation. This step preserves raw data and avoids modifying it directly.

Extract Tables:

Raw Table	Staging Table	Key Columns
Learner_Raw	stg_learner_raw	learner_id, email, degree, major
Cognito_Raw2	stg_cognito_raw	email, gender, birthdate, address
CohortRaw	stg_cohort_raw	cohort_id, cohort_name, start_date
LearnerOpportunity_Raw	stg_learner_opportunity	learner_id, opportunity_id
Opportunity_Raw	stg_opportunity_raw	opportunity_id, opportunity_name
Marketing Campaign Data All Accounts	stg_marketing_campaign	email, clicks, opens, cost_per_click

---

STEP 2: Transform data

Apply data cleaning and standardization logic in the staging area before loading into the master table.

🔧 **Transformations:**

Transformation Type	Description
Missing Value Handling	Use COALESCE() for defaults on nulls (e.g., 'Unknown', 0, empty string)
Format Normalization	Convert email to lowercase, names to InitCap, dates to YYYY-MM-DD
Duplicate Removal	Use DISTINCT or ROW_NUMBER()/RANK() to filter duplicates

MASTER TABLE AND ETL PROCESS

Transformation Type	Description
Data Type Consistency	Use CAST() or TO_DATE() for proper typing
Relationship Mapping	Join across staging tables using email, learner_id, cohort_id, opportunity_id
Business Rules	Generate etl_batch_id, set record_status = 'active', set load timestamp

Loading of data

The **Loading** is the final step of the ETL workflow where the **transformed, cleaned, and enriched data** from the staging tables is **inserted into the Master Table**.

Purpose of loading data in to master table:

To consolidate validated and normalized data from multiple sources into a single, structured table that's ready for analysis, reporting, or dashboarding.

Major Characteristics:

Task	Description
Insert Clean Data	Insert all transformed rows from staging tables using INSERT INTO or MERGE logic.
Maintain Integrity	Ensure proper relationships (via foreign keys), valid data types, and required values.
Track Metadata	Attach etl_batch_id, load_timestamp, and record_status to enable traceability and audit.
De-duplication	Filter out or ignore duplicate records to prevent redundancy in the Master Table.

MASTER TABLE AND ETL PROCESS

Task	Description
Transactional Control	Run within a transaction block to ensure either complete success or rollback on failure.

Load Logic Highlights (from your ETL):

- Join **learner**, **cognito**, **cohort**, **opportunity**, and **marketing** data using keys like email, learner_id, opportunity_id.
- Normalize:
    - Emails → lowercase
    - Text fields → INITCAP()
    - Dates → parsed with TO_DATE()
    - Costs and metrics → rounded and defaulted
- Insert only **valid learners** (learner_id IS NOT NULL).