Harvard Artifacts Collection – ETL & SQL Analytics System

Author: PS Naveen Kumar

1. Introduction

This project presents the design and implementation of an end-to-end ETL and SQL analytics system built using the Harvard Art Museums API. The system extracts artifact data, transforms nested JSON into structured relational tables, loads it into a TiDB Cloud (MySQL-compatible) database, and enables interactive SQL querying through a Streamlit web application. The project demonstrates practical data engineering skills including API integration, data normalization, SQL schema design, and database-driven analytics using a user-friendly interface.

The digitization of museum collections has enabled open access to large volumes of cultural and historical data. The Harvard Art Museums API provides structured information on thousands of artifacts, including metadata, media attributes, and color composition. However, the raw API data is not directly suitable for structured analysis. This project addresses this gap by building a complete pipeline that transforms API responses into relational tables and supports analytical exploration through SQL.

2. Problem Statement

- The primary challenge addressed by this project is the absence of a unified system that allows users to:
- Extract large volumes of artifact data from a public API
- Transform nested JSON data into normalized relational tables
- Persist data reliably in a cloud SQL database
- Execute analytical SQL queries through a simple, interactive interface

3. Objectives

- Develop an ETL pipeline using Python and the Harvard Art Museums API
- Fetch up to 2,500 artifacts per classification
- Normalize API data into three relational tables (metadata, media, colors)
- Prevent duplicate database inserts using primary keys and INSERT IGNORE
- Support TiDB Cloud (MySQL-compatible) database with SSL
- Build an interactive Streamlit UI for data extraction, migration, and querying
- Implement 30 predefined SQL analytical queries

Harvard Artifacts Collection – ETL & SQL Analytics System

Author: PS Naveen Kumar

4. Literature Review

Existing research in cultural informatics emphasizes digitization, metadata curation, and open museum datasets. While many museums provide public APIs, fewer implementations demonstrate complete ETL workflows combined with relational storage and interactive analytics. This project contributes a practical, end-to-end implementation that integrates data extraction, transformation, SQL modeling, and user-driven analytics within a single system.

5. System Architecture

- The system architecture consists of the following components:
- Data Source: Harvard Art Museums API
- ETL Layer: Python with requests
- Data Processing: In-memory transformation using Pandas-compatible structures
- Storage Layer: TiDB Cloud (MySQL protocol, SSL enabled)
- Application Layer: Streamlit web application
- Analytics Layer: SQL query engine with predefined analytical queries

Flow:

API → Data Extraction → Normalization → SQL Database → Streamlit UI → SQL Analytics

6. ETL Pipeline Design

The ETL process fetches artifact records page-by-page (25 pages × 100 records per page).

Each artifact record is split into:

- Metadata attributes
- Media-related attributes
- Color composition attributes

The pipeline ensures data consistency by resetting state between runs and uses primary keys and foreign key relationships to maintain referential integrity during SQL insertion.

Harvard Artifacts Collection – ETL & SQL Analytics System

Author: PS Naveen Kumar

7. Database Schema

Table 1: metadata

Column Name	Description
id	Primary key (Artifact ID)
title	Artifact title
culture	Cultural origin
period	Historical period
century	Century of creation
medium	Material used
dimensions	Physical dimensions
description	Artifact description
department	Museum department
classification	Artifact category
accessionyear	Year acquired
accessionmethod	Method of acquisition

Table 2: media

Column Name	Description
objid	Primary key, FK → metadata.id
imagecount	Number of images
mediacount	Total media files
colorcount	Number of colors
db_rank	Database rank
datebegin	Creation start year
dateend	Creation end year

Table 3: colors

Column Name	Description
objid	FK → metadata.id
color	Color name
spectrum	Color spectrum
hue	Color hue
percent	Coverage percentage
css3	CSS color code

Harvard Artifacts Collection – ETL & SQL Analytics System

Author: PS Naveen Kumar

8. Implementation

- The system is implemented in Python using:
- requests for API communication
- SQLAlchemy and PyMySQL for database interaction
- Streamlit for the web interface
- streamlit-option-menu for UI navigation

The application supports secure connections to TiDB Cloud using SSL certificates and dynamically creates the target database and tables during migration.

9. Streamlit Application

- The Streamlit UI provides the following functionality:
- Selection of artifact classifications
- Extraction of artifact data from the API
- Preview of metadata, media, and color tables
- Migration of processed data into TiDB Cloud
- Execution of predefined SQL queries
- Display of query results in tabular format

10. SQL Analytics

- The project includes 30 analytical SQL queries, covering:
- Cultural and classification analysis
- Media and image-based insights
- Color and hue distribution analysis
- Time-based queries using periods and centuries
- Multi-table joins across metadata, media, and colors
- Ranking-based and aggregation queries

These queries enable structured exploration of the dataset for academic and analytical use cases.

Harvard Artifacts Collection – ETL & SQL Analytics System

Author: PS Naveen Kumar

11. TiDB Cloud Integration

- TiDB Cloud is used as the primary database backend due to its MySQL compatibility and cloud scalability. The application:
- Connects using SSL certificates
- Creates the database programmatically
- Uses SQLAlchemy engines for query execution
- Ensures secure and reliable data persistence

12. Results & Discussion

- The ETL pipeline successfully processed 12,500+ artifact records
- Data normalization enabled efficient relational querying
- Duplicate inserts were prevented using primary keys and INSERT IGNORE
- SQL queries revealed meaningful patterns across culture, classification, and color usage
- The Streamlit interface simplified database interaction for non-technical users

13. Conclusion

This project demonstrates a complete and practical data engineering work flow using a real-world public API. It showcases skills in ETL design, relational database modeling, SQL analytics, and interactive application development. The system is scalable, reproducible, and suitable for academic exploration or portfolio presentation.