Asset Database Project
A Stepwise Approach
October 2021
Renee Raven

Table of Contents

Scenario

Our client runs a small web design business and often uses open source and creative common digital assets (photos, graphics, audio, video). Each asset may be used in any number of projects. She would like a database to catalog the items she has collected and/or used in projects.

Currently she has over 5,000 assets distributed in five folders: BW Photos, Color Photos, Graphics, Audio, and Video. Her collection continues to grow. She saves the new items in the appropriate folder and titles them with a unique name. However, the number of items, and the lack of a system to search for the items, often leads to trouble finding specific assets.

Constraints & Considerations

- Each asset may have multiple creatives (creators/authors)
- Each asset may be used for any number of projects
- Each project may use any number of assets
- Each creative may produce any number of assets
- Each asset has a primary subject
- Each asset may have any number of secondary assets
- Each asset has a primary mood
- Each asset may have any number of secondary moods
- Each project has 1 and only 1 project owner

Additionally, our client requested the ability to search the database of her collection of digital assets by:

- Name
- Creative(s)
- Type
- Source
- Project(s)
- File size
- Subject(s)
- Mood(s)
- File type
- License
- Resolution
- Dimensions
- Bit rate
- Project owner

Identify Entities

- asset
- creative
- project
- secondary subjects
- secondary moods
- photo specs
- image specs
- audio specs
- video specs

Preliminary List of Attributes with Entities

- asset
    - asset_id
    - asset_name
    - creative(s)
    - asset_type (category - photo, gaphic, audio, video)
    - file_type (category - multiple options)
    - file_size
    - primary_subject
    - primary_mood
    - asset_source
    - license_type
    - project(s)
- creative
    - creative_id
    - creative_name
    - project(s)
    - asset(s)
- project
    - project_id
    - project_name
    - asset(s)
    - creative(s)
    - owner
- subjects
    - asset_id
    - subject
- moods

- o asset_id
- o mood
  - ▪ photo_specs
    - o ps_id
    - o asset_id
    - o dimensions
    - o resolution
    - o color or bw (category - color or bw)
  - ▪ image_specs
    - o gs_id
    - o asset_id
    - o dimensions
    - o resolution
    - o color or bw (category - color or bw)
    - o scalable (category - yes or no)
  - ▪ audio_specs
    - o as_id
    - o asset_id
    - o time
    - o sample_rate
    - o bit_rate
  - ▪ video_specs
    - o vs_id
    - o asset_id
    - o time
    - o resolution
    - o frame_rate

Check Atomicity of Attributes

The asset, creative, and project tables contain attributes where multiple values are possible. Since not every asset is used in a project and not every project is attached to a creative, we can't make a table to connect the three tables.

We need to create 3 associative entity tables built with a combination of foreign keys acting as a primary key and then remove the non-atomic attributes from the asset, creative, and project tables.

New associative tables to add:

- ▪ asset_creative
  - o asset_id
  - o creative_id
- ▪ asset_project
  - o asset_id
  - o project_id

- creative_project
  - creative_id
  - project_id

Identifiers / Keys

- asset
- creative
- project
- secondary subjects
- secondary moods
- photo specs
- image specs
- audio specs
- video specs

Relationships

Assign Char Types to Attributes

Normalize

Revised List of Attributes with Entities

DDL

DML

SQL

Should include special effects? lossy or lossless? list of all possible file_types? Should we include a primary subject, primary mood?