CineLog Database Guide

Plain English Explanations for All 26 Tables

Purpose of this Document
To explain exactly *what* each table stores, *why* it is necessary, and *how* it connects to the rest of the database.

Project Reference Manual

Contents

1 Domain 1: The Core Catalog (Static Data) 2

2 Domain 2: The Connectors (Junctions) 3

3 Domain 3: User Identity 4

4 Domain 4: Logging & Reviews 5

5 Domain 5: Curation 6

6 Domain 6: Social Interaction 6

Domain 1: The Core Catalog (Static Data)

This section holds the data that exists *before* any user signs up. It is the library of films.

Table 1: Movies

What it does: Stores the main details of a film (Title, Poster, Plot). This is the central hub of your database.
Why we need it: Every review, log, and list ultimately points back to a specific row in this table.
Connections:

- Connected to **People** via *Cast_Crew*.
- Connected to **Users** via *Reviews*, *Logs*, and *Favorites*.

Table 2: People

What it does: A single list of every actor, director, writer, and crew member.
Why we need it: To avoid duplicates. If ”Christopher Nolan” directs 10 movies, we only store his name and photo once in this table, instead of typing it 10 times in the Movies table.
Connections: Linked to **Movies** via the *Cast_Crew* table.

Table 3: Streaming_Providers

What it does: A lookup list of services like Netflix, Hulu, Disney+.
Why we need it: If Netflix changes their logo, you update it here once, and it updates on every movie page.
Connections: Linked to **Movies** via *Movie_Streaming*.

Table 4: Genres

What it does: A simple list of categories (Horror, Comedy, Sci-Fi).
Why we need it: It forces consistency. Users can’t accidentally type ”SciFi” and ”Sci-Fi”; they must use the valid ID from this table.

Table 5: Production_Companies

What it does: A list of studios (A24, Warner Bros).
Why we need it: Allows users to filter ”Show me all A24 movies.”

Table 6: Countries

What it does: A standard list of country codes (US, UK, FR).
Why we need it: Essential for analyzing cinema from specific regions.

Domain 2: The Connectors (Junctions)

These tables are purely for linking Domain 1 tables together. They handle "Many-to-Many" relationships.

Table 7: Cast_Crew

What it does: The bridge between **Movies** and **People**.
Why we need it: A movie has many actors; an actor is in many movies. This table stores the specific *Role* (e.g., "Director") and *Character Name* for that specific pairing.

Table 8: Movie_Streaming

What it does: The bridge between **Movies** and **Streaming_Providers**.
Why we need it: Stores the specific URL (Deep Link) to watch *Inception* on *Netflix*. It allows a movie to be on multiple platforms at once.

Table 9: Movie_Genres

What it does: Connects **Movies** to **Genres**.
Why we need it: A movie like *Alien* is both "Horror" and "Sci-Fi". This table allows multiple genre tags per movie.

Table 10: Movie_Companies

What it does: Connects **Movies** to **Production_Companies**.
Why we need it: Many movies are co-productions between multiple studios.

Table 11: Movie_Countries

What it does: Connects **Movies** to **Countries**.
Why we need it: Handles international co-productions (e.g., a film made by France and Germany).

Domain 3: User Identity

Who is using the app?

Table 12: Users

What it does: Stores login info (Email, Password Hash) and profile info (Bio, Avatar).
Connections: Referenced by almost every table in the ”Activity” and ”Social” domains.

Table 13: Follows

What it does: Tracks who follows whom.
How it works: It has two columns, *follower_id* and *following_id*. Both point back to the **Users** table.

Table 14: User_Blocks

What it does: A safety feature. If User A blocks User B, this table records that relationship so the system knows to hide content.

Table 15: User_Favorites

What it does: Stores the specific ”Top 4” movies displayed on a user’s profile header.
Why separate from lists? These 4 movies get special visual treatment in the UI.

Table 16: Watchlist

What it does: A simple ”Plan to Watch” bucket.
Why we need it: It’s a quick bookmarking tool, distinct from a curated ”List”.

Domain 4: Logging & Reviews

This is the core ”Letterboxd” functionality.

Table 17: Diary_Logs

What it does: Records the *event* of watching a movie on a specific date.
Why separate from reviews? You can watch *Star Wars* 5 times. You want 5 diary entries (to track stats), but you might only write 1 review.

Table 18: Reviews

What it does: Stores the written text (essay/opinion) about a movie.
Connections: Can be optionally linked to a **Diary_Log** if the review was written during a specific watch event.

Table 19: Ratings

What it does: Stores the user’s current star score (0.5 - 5.0).
Why separate? A user might rate a movie without writing a word. Or they might change their rating later without changing their review.

Table 20: Tags

What it does: A dictionary of user-created hashtags (e.g., #Halloween, #DateNight).
Why we need it: Allows users to organize their diary logs with custom keywords.

Table 21: Log_Tags

What it does: Connects a specific **Diary_Log** to a **Tag**.

Domain 5: Curation

Table 22: Lists

What it does: The ”Container” for a collection (Title, Description).
Example: ”My Top 10 Horror Movies of 2024”.

Table 23: List_Items

What it does: The actual movies inside the list container.
Crucial Field: *rank_order*. This ensures the movies stay in the order the user placed them (1 to 10).

Domain 6: Social Interaction

Table 24: Interactions (Likes)

What it does: A unified table for all ”Likes”.
How it works: Instead of having 3 tables (ReviewLikes, ListLikes, DiaryLikes), this one table can point to any of them. It saves space and makes counting ”Total Likes” easier.

Table 25: Comments

What it does: Stores user comments on Reviews or Lists.
Advanced Feature: Includes *parent_comment_id* to allow for threaded replies (like Reddit or Facebook).

Table 26: Notifications

What it does: The ”Activity Feed” for a user.
Why we need it: When User A likes User B’s review, this table creates a row so User B sees a red notification dot.