

# CineLog Database

## Schema Reference Manual

**Version 1.0**

January 8, 2026

**System Overview**  
26 Tables · 6 Domains · Normalized 3NF

*Designed for Letterboxd Clone Architecture*

# Contents

<b>1 Domain 1: The Core Catalog</b>	<b>2</b>
1.1 Lookup Tables . . . . .	2
<b>2 Domain 2: Metadata Junctions</b>	<b>3</b>
2.1 Standard Junctions . . . . .	3
<b>3 Domain 3: User Identity</b>	<b>3</b>
3.1 Safety & Utilities . . . . .	4
<b>4 Domain 4: Logging &amp; Reviews</b>	<b>5</b>
4.1 Tagging System . . . . .	5
<b>5 Domain 5: Curation (Lists)</b>	<b>5</b>
<b>6 Domain 6: Social Interaction</b>	<b>7</b>

# 1 Domain 1: The Core Catalog

This domain manages static movie data. These entities exist independently of user activity.

Movies		<i>The central registry for film data</i>	
movie_id	INT	<b>PK</b>	Auto-incrementing ID
title	VARCHAR		Film title
release_date	DATE		Official release date
runtime_minutes	INT		Duration in minutes
plot_summary	TEXT		Full synopsis
poster_url	VARCHAR		Path to image resource
tmdb_id	VARCHAR	UQ	External ID for syncing
tmdb_vote_avg	DECIMAL		Cached TMDB rating
tmdb_vote_count	INT		Cached TMDB vote count

  

People		<i>Cast and Crew members</i>	
person_id	INT	<b>PK</b>	Unique Person ID
name	VARCHAR		Full Name
profile_image_url	VARCHAR		Headshot URL

  

Streaming_Providers		<i>Streaming services lookup</i>	
provider_id	INT	<b>PK</b>	Unique Provider ID
name	VARCHAR		e.g., Netflix, Hulu
logo_url	VARCHAR		Provider icon

## 1.1 Lookup Tables

- **Genres:** genre\_id (PK), name (e.g., Horror, Sci-Fi)
- **Production\_Companies:** company\_id (PK), name
- **Countries:** country\_code (PK, ISO-3166), name

## 2 Domain 2: Metadata Junctions

---

These tables handle Many-to-Many relationships for the catalog.

Cast_Crew		Links People to Movies	
movie_id	INT	FK	Ref. Movies
person_id	INT	FK	Ref. People
role	VARCHAR		'Director', 'Actor', 'Writer'
character_name	VARCHAR		Nullable (for Actors)
Movie_Streaming		Where to watch a movie	
movie_id	INT	FK	Ref. Movies
provider_id	INT	FK	Ref. Streaming_Providers
web_link	VARCHAR		Deep link to content
region_code	VARCHAR		e.g., 'US', 'GB'

### 2.1 Standard Junctions

- **Movie\_Genres:** Links movie\_id to genre\_id.
- **Movie\_Companies:** Links movie\_id to company\_id.
- **Movie\_Countries:** Links movie\_id to country\_code.

## 3 Domain 3: User Identity

---

Users		Core authentication & profile	
user_id	INT	PK	Unique User ID
username	VARCHAR	UQ	Display handle
email	VARCHAR	UQ	Login credential
password_hash	VARCHAR		Encrypted string
bio	TEXT		User profile bio
avatar_url	VARCHAR		Profile picture
created_at	TIMESTAMP		Registration date

Follows		<i>Social graph connections</i>	
follower_id	INT	<b>FK</b>	User who follows
following_id	INT	<b>FK</b>	User being followed
created_at	TIMESTAMP		Date followed
User_Favorites		<i>Top 4 profile movies</i>	
user_id	INT	<b>FK</b>	Ref. Users
movie_id	INT	<b>FK</b>	Ref. Movies
sort_order	INT		Position (1-4)
<i>Note: The "Top 4" constraint should be enforced by application logic.</i>			

### 3.1 Safety & Utilities

- **User\_Blocks:** blocker\_id, blocked\_id. Prevents interaction.
- **Watchlist:** user\_id, movie\_id, added\_at.

## 4 Domain 4: Logging & Reviews

---

This domain separates the *Opinion* (Rating) from the *Event* (Diary Log).

Diary_Logs		<i>The specific event of watching</i>	
log_id	INT	<b>PK</b>	Unique Log ID
user_id	INT	<b>FK</b>	Who watched it
movie_id	INT	<b>FK</b>	What they watched
watched_on_date	DATE		Viewing date
rating_snapshot	DECIMAL		Rating given at time of log
is_rewatch	BOOLEAN		Rewatch flag

  

Reviews		<i>Written text reviews</i>	
review_id	INT	<b>PK</b>	Unique Review ID
user_id	INT	<b>FK</b>	Author
movie_id	INT	<b>FK</b>	Subject
log_id	INT	<b>FK</b>	Optional link to Diary
content	TEXT		The review body
contains_spoilers	BOOLEAN		Spoiler flag

  

Ratings		<i>Current score for a movie</i>	
rating_id	INT	<b>PK</b>	Unique Rating ID
user_id	INT	<b>FK</b>	User
movie_id	INT	<b>FK</b>	Movie
score	DECIMAL		0.5 - 5.0 scale

### 4.1 Tagging System

- **Tags:** tag\_id (PK), name (User defined string).
- **Log\_Tags:** Junction linking log\_id to tag\_id.

## 5 Domain 5: Curation (Lists)

---

**Lists***User-curated collections*

list_id	INT	<b>PK</b>	Unique List ID
user_id	INT	<b>FK</b>	Owner
title	VARCHAR		List Title
description	TEXT		Description
privacy	ENUM		'Public', 'Friends', 'Private'

**List\_Items***Movies inside a list*

list_id	INT	<b>FK</b>	Ref. Lists
movie_id	INT	<b>FK</b>	Ref. Movies
rank_order	INT		Sort position
notes	TEXT		Optional annotation

## 6 Domain 6: Social Interaction

This domain uses **Exclusive Arc (Polymorphic)** constraints to handle interactions efficiently.

### Interactions

### *Unified Likes table*

interaction_id	INT	<b>PK</b>	Unique ID
user_id	INT	<b>FK</b>	Who liked it
type	ENUM		'LIKE'
review_id	INT	<b>FK</b>	Nullable Target 1
list_id	INT	<b>FK</b>	Nullable Target 2
diary_id	INT	<b>FK</b>	Nullable Target 3

**Constraint:** Exactly ONE of review\_id, list\_id, or diary\_id must be NOT NULL.

### Comments

### *Unified Threaded Comments*

comment_id	INT	<b>PK</b>	Unique Comment ID
user_id	INT	<b>FK</b>	Author
review_id	INT	<b>FK</b>	Nullable Target 1
list_id	INT	<b>FK</b>	Nullable Target 2
parent_comment_id	INT	<b>FK</b>	Self-ref (Replies)
content	TEXT		Comment text

### Notifications

### *Activity Feed*

notif_id	INT	<b>PK</b>	Unique ID
recipient_id	INT	<b>FK</b>	Who sees it
actor_id	INT	<b>FK</b>	Who caused it
action_type	ENUM		'LIKE', 'FOLLOW', 'COMMENT'
resource_id	INT		ID of target object
is_read	BOOLEAN		Unread status

**End of Schema Definition.**