

Schéma pagila (vue partielle restreinte au catalogue des films)

1

## Schéma pagila (vue partielle restreinte au catalogue des films)

Créé par [DbSchema](#)

**pagila** est un schéma d'entraînement dérivé du schéma **sakila** de MySQL. C'est une base de donnée fictive destinée à gérer le catalogue et les stocks d'une chaine de magasins de location de DVD (début du XXIème siècle).

Vue d'ensemble des tables portant sur les films

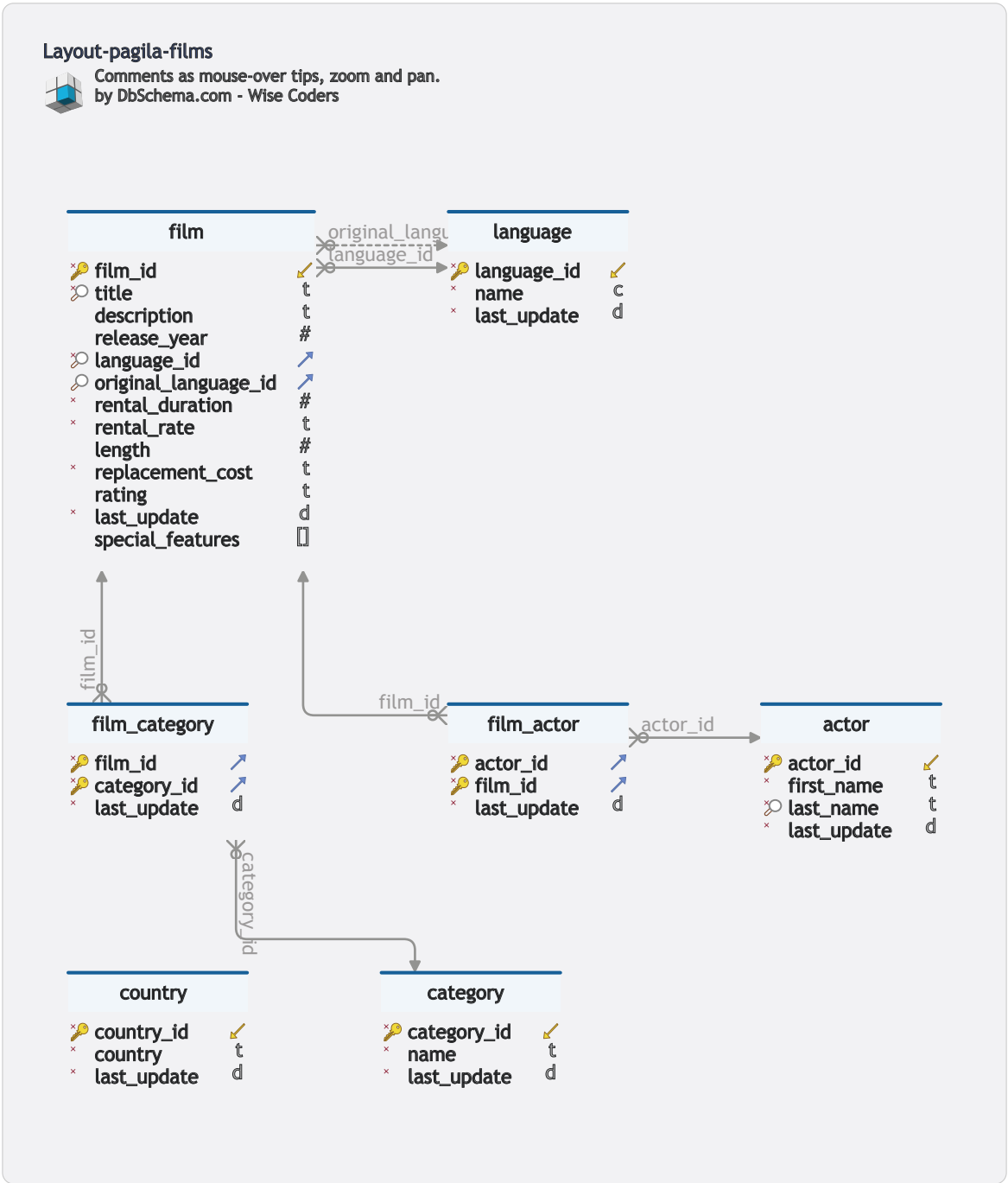


FIG. 1 : Vue partielle de pagila

**Table pagila.actor**

*	actor_id	integer DEFAULT next-val('pagila.actor_actor_id_seq'::reg-class)
*	first_name	varchar(45)
*	last_name	varchar(45)
*	last_update	timestamp DEFAULT now()

Les lignes de la table **actor** sont identifiées par la colonne **actor\_id**. Deux lignes différentes peuvent coïncider sur les colonnes **first\_name** et **last\_name** mais elles doivent se distinguer sur la colonne **actor\_id**. La colonne **actor\_id** est la *clé primaire* de la table **actor**.

**Table pagila.category**

*	category_id	integer DEFAULT next-val('pagila.category_category_id_seq'::reg-class)
*	name	varchar(25)
*	last_update	timestamp DEFAULT now()

Un film relève d'une ou plusieurs catégories (**horror**, **children**, ...). La table **category** liste les catégories utilisées dans **pagila**. Les lignes de cette table sont distinguées par la colonne **category\_id**.

**Table pagila.country**

*	country_id	integer DEFAULT next-val('pagila.country_country_id_seq'::reg-class)
*	country	varchar(50)
*	last_update	timestamp DEFAULT now()

**Table pagila.film**

*	film_id	integer DEFAULT next-val('pagila.film_film_id_seq'::reg-class)
*	title	varchar(255)
	description	text
	release_year	integer
*	language_id	smallint
	original_language_id	smallint
*	rental_duration	smallint DEFAULT 3
*	rental_rate	varchar(4) DEFAULT 4.99
	length	smallint
*	replacement_cost	varchar(5) DEFAULT 19.99
	rating	mpaa_rating DEFAULT 'G'::pagila.mpaa_rating
*	last_update	timestamp DEFAULT now()
	special_features	text[]

La table **film** contient le catalogue de la chaîne de magasins de DVD. Les lignes de cette table sont distinguées par la colonne **film\_id**. Deux lignes distinctes peuvent coïncider sur la colonne **title**.

La colonne **original\_language\_id** permet de repérer la langue de la version originale du film. La colonne **language\_id** permet de repérer la langue dans laquelle le DVD est éditée. Ces informations

doivent être recollées à l'aide de la table `language`. Les valeurs prises par les colonnes `language_id` et `original_language_id` doivent obligatoirement référer à des valeurs rencontrées dans la colonne `language_id` de la table `language`. On parle de *contraintes référentielles*.

### Foreign Keys

film_original_language_id_fkey	( original_language_id ) ref pagila.language (language_id)
film_language_id_fkey	( language_id ) ref pagila.language (language_id)

### Table pagila.film\_actor

*	actor_id	smallint
*	film_id	smallint
*	last_update	timestamp DEFAULT now()

### Foreign Keys

film_actor_film_id_fkey	( film_id ) ref pagila.film (film_id)
film_actor_actor_id_fkey	( actor_id ) ref pagila.actor (actor_id)

La table `film_actor` est une table intermédiaire qui permet de savoir quels acteurs jouent dans quels films.

### Table pagila.film\_category

*	film_id	smallint
*	category_id	smallint
*	last_update	timestamp DEFAULT now()

La table `film_category` est une table intermédiaire qui permet de savoir de quelles catégories relève un film.

### Foreign Keys

film_category_film_id_fkey	( film_id ) ref pagila.film (film_id)
film_category_category_id_fkey	( category_id ) ref pagila.category (category_id)

### Table pagila.language

*	language_id	integer DEFAULT next- val('pagila.language_language_id_seq'::reg- class)
*	name	char(20)
*	last_update	timestamp DEFAULT now()