


# Schéma : pagila (films)

pagila, film, relationnel, pattes de corbeau, crowfoot

2024-09-20

 Présentation de la partie film du schéma pagila, pour l'entraînement à PostGres.

- **L3 MIASHS/Ingémath**
- [Université Paris Cité](#)
- Année 2024-2025
- [Course Homepage](#)
- [Moodle](#)



This is a layout. You can create multiple layouts with the same or different tables.  
Double-click the table headers to edit.

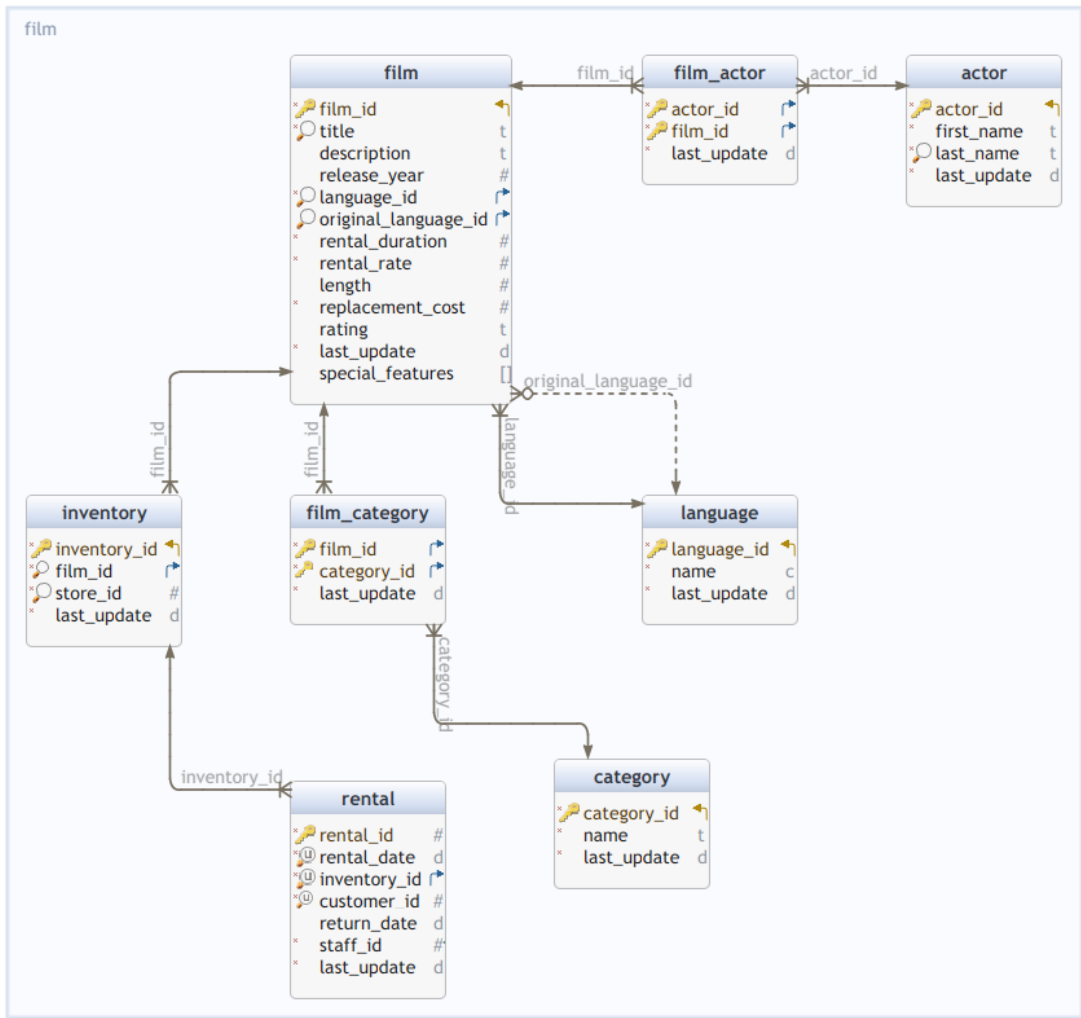


FIG. 1 : Schema pagila créé par DbSchema

### Contexte

Description de quelques tables from the **pagila** database, spécifiquement les tables liées au catalogue de films (sans considérer les tables liées au fichier client, à la gestion des locations, aux magasins).

[Pagemaison de la base de données pagila.](#)

[Pagemaison de la base de données sakila](#)

### Table pagila.actor

Idx	Name	Data Type
*	actor_id	integer DEFAULT nextval('pagila.actor_actor_id_seq'::regclass)
*	first_name	varchar(45)
*	last_name	varchar(45)
*	last_update	timestamp DEFAULT now()

`actor_id` sert de clé primaire. Noter le mécanisme de génération de nouvelles clés grâce au comportement par défaut et à l'usage de la fonction `nextval()` opérant sur une séquence associée à la table `actor`.

La colonne `last_update` sert à dater les mises à jour.

### Table `pagila.category`

Idx	Name	Data Type
*	category_id	integer DEFAULT nextval('pagila.category_category_id_seq'::regclass)
*	name	varchar(25)
*	last_update	timestamp DEFAULT now()

### Table `pagila.film`

Idx	Name	Data Type
*	film_id	integer DEFAULT nextval('pagila.film_film_id_seq'::regclass)
*	title	varchar(255)
	description	text
	release_year	year
*	language_id	smallint
	original_language_id	smallint
*	rental_duration	smallint DEFAULT 3
*	rental_rate	numeric(4,2) DEFAULT 4.99
	length	smallint
*	replacement_cost	numeric(5,2) DEFAULT 19.99
	rating	mpaa_rating DEFAULT 'G'::pagila.mpa_rating
*	last_update	timestamp DEFAULT now()
	special_features	text[]

### Foreign Keys

Type	Name	On
	<code>film_original_language_id_fkey</code>	( <code>original_language_id</code> ) ref <code>pagila.language (language_id)</code>
	<code>film_language_id_fkey</code>	( <code>language_id</code> ) ref <code>pagila.language (language_id)</code>

Les colonnes `film_id` (clé primaire), `original_language_id` et `language_id` sont utilisées pour réaliser des (équi)-jointures avec les tables `film_actor`, `film_category`, `film_language`

Un film est censé n'avoir qu'une seule langue originale. L'association *film a pour langue originale la langue xxx* est de un vers plusieurs. On peut la représenter à l'aide d'un attribut dans la table `film`.

### Table pagila.film\_actor

Idx	Name	Data Type
*	actor_id	smallint
*	film_id	smallint
*	last_update	timestamp DEFAULT now()

C'est une table intermédiaire qui permet de représenter/traduire l'association plusieurs à plusieurs *acteur joue dans film*.

#### Foreign Keys

Type	Name	On
	film_actor_film_id_fkey	( film_id ) ref pagila.film (film_id)
	film_actor_actor_id_fkey	( actor_id ) ref pagila.actor (actor_id)

### Table pagila.film\_category

Idx	Name	Data Type
*	film_id	smallint
*	category_id	smallint
*	last_update	timestamp DEFAULT now()

#### Foreign Keys

Type	Name	On
	film_category_film_id_fkey	( film_id ) ref pagila.film (film_id)
	film_category_category_id_fkey	( category_id ) ref pagila.category (category_id)

C'est une table intermédiaire qui permet de représenter/traduire l'association plusieurs à plusieurs *film relève de catégorie*.

### Table pagila.language

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