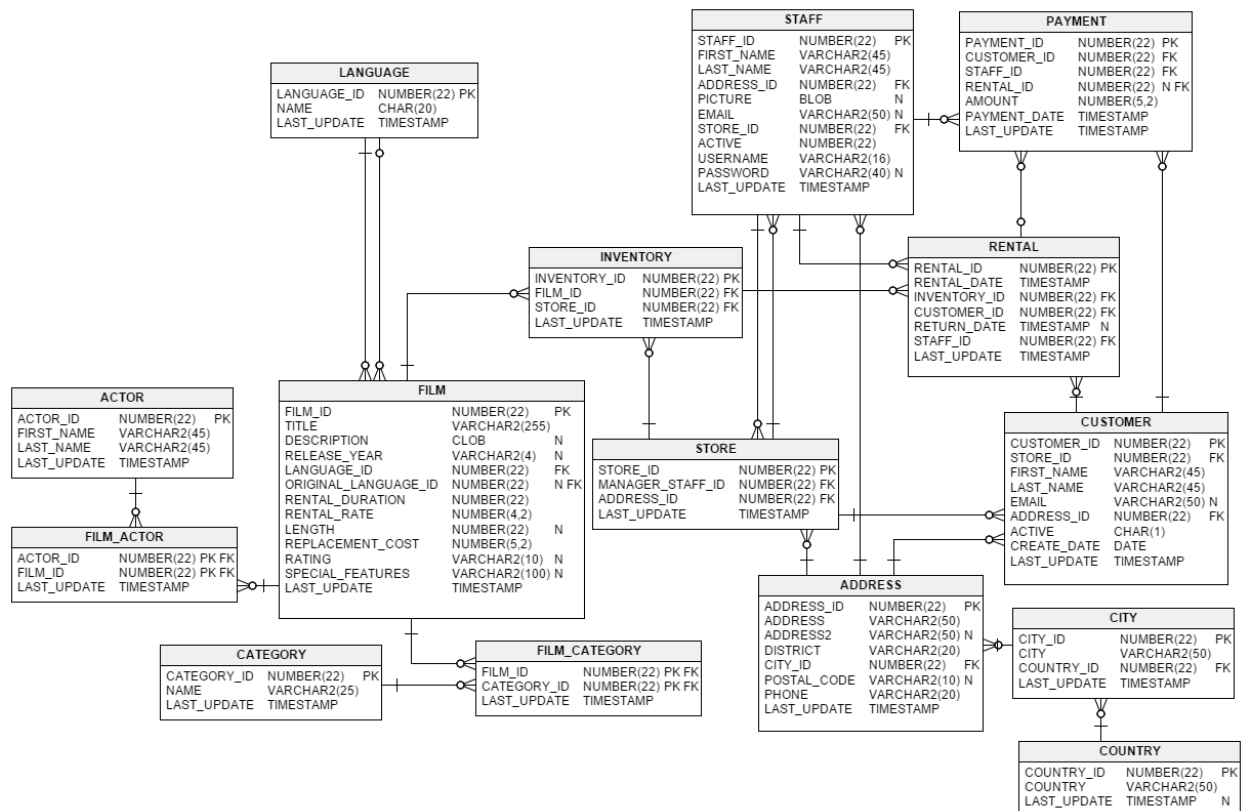


Introduction

The Sakila database is a nicely normalised schema modelling a DVD rental store, featuring things like films, actors, film-actor relationships, and a central inventory table that connects films, stores, and rentals.



Installation

Download from <https://downloads.mysql.com/docs/sakila-db.zip>

A downloadable archive is available in compressed **tar** file or Zip format. The archive contains three files: **sakila-schema.sql**, **sakila-data.sql**, and **sakila.mwb**.

The **sakila-schema.sql** file contains all the **CREATE** statements required to create the structure of the Sakila database including tables, views, stored procedures, and triggers.

The `sakila-data.sql` file contains the `INSERT` statements required to populate the structure created by the `sakila-schema.sql` file, along with definitions for triggers that must be created after the initial data load.

The `sakila.mwb` file is a MySQL Workbench data model that you can open within MySQL Workbench to examine the database structure

To install the Sakila sample database, follow these steps:

1. Extract the installation archive to a temporary location such as `C:\temp\` or `/tmp/`. When you unpack the archive, it creates a directory named `sakila-db` that contains the `sakila-schema.sql` and `sakila-data.sql` files.
2. Connect to the MySQL server using the **mysql** command-line client with the following command:

```
$> mysql -u root -p
```

Enter your password when prompted.

3. Execute the `sakila-schema.sql` script to create the database structure, and execute the `sakila-data.sql` script to populate the database structure, by using the following commands:

```
mysql> SOURCE C:/temp/sakila-db/sakila-schema.sql;
```

```
mysql> SOURCE C:/temp/sakila-db/sakila-data.sql;
```

Replace the paths to the `sakila-schema.sql` and `sakila-data.sql` files with the actual paths on your system.

4. Confirm that the sample database is installed correctly. Execute the following statements. You should see output similar to that shown here.

```
mysql> USE sakila;  
Database changed
```

```
mysql> SHOW FULL TABLES;
```

Tables_in_sakila	Table_type
actor	BASE TABLE
actor_info	VIEW
address	BASE TABLE
category	BASE TABLE
city	BASE TABLE
country	BASE TABLE
customer	BASE TABLE
customer_list	VIEW
film	BASE TABLE
film_actor	BASE TABLE
film_category	BASE TABLE
film_list	VIEW
film_text	BASE TABLE
inventory	BASE TABLE
language	BASE TABLE
nicer_but_slower_film_list	VIEW
payment	BASE TABLE
rental	BASE TABLE
sales_by_film_category	VIEW
sales_by_store	VIEW
staff	BASE TABLE
staff_list	VIEW
store	BASE TABLE

```
23 rows in set (0.01 sec)
```

```
mysql> SELECT COUNT(*) FROM film;
+-----+
| COUNT(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM film_text;
+-----+
| COUNT(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.00 sec)
```

Tables

<https://dev.mysql.com/doc/sakila/en/sakila-structure-tables.html>

Exercises

1. Display the first and last name of each actor in a single column in upper case letters in alphabetic order. Name the column Actor Name.
2. Find all actors whose last name contain the letters GEN:
3. Using IN, display the country_id and country columns of the following countries: Afghanistan, Bangladesh, and China:
4. List the last names of actors, as well as how many actors have that last name.
5. List last names of actors and the number of actors who have that last name, but only for names that are shared by at least two actors

6. The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix the record.
7. Use JOIN to display the first and last names, as well as the address, of each staff member. Use the tables staff and address:
8. List each film and the number of actors who are listed for that film. Use tables film_actor and film. Use inner join.
9. How many copies of the film Hunchback Impossible exist in the inventory system?
10. Using the tables payment and customer and the JOIN command, list the total paid by each customer. List the customers alphabetically by last name
11. The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence, films starting with the letters **K** and **Q** have also soared in popularity. Use subqueries to display the titles of movies starting with the letters **K** and **Q** whose language is English.
12. Use subqueries to display all actors who appear in the film *Alone Trip*.
13. You want to run an email marketing campaign in Canada, for which you will need the names and email addresses of all Canadian customers. Use joins to retrieve this information.
14. Sales have been lagging among young families, and you wish to target all family movies for a promotion. Identify all movies categorized as family films.
15. Create a Stored procedure to get the count of films in the input category (IN category_name, OUT count)
16. Display the most frequently rented movies in descending order.
17. Write a query to display for each store its store ID, city, and country.
18. List the genres and its gross revenue.
19. Create a View for the above query(18)
20. Select top 5 genres in gross revenue view.