Rent-A-Film Database

1. Installation

The Rent-A-Film sample database is available from here. A downloadable file is available in compressed `Zip` format. The file contains two files: `rent-a-film-schema.sql`, and `rent-a-film-data.sql`

The `rent-a-film-schema.sql` file contains all the `CREATE` statements required to create the structure of the Rent-A-Film database including tables, views, stored procedures, and triggers.

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

To install the Rent-A-Film 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 `rent-a-film-db` that contains the `rent-a-film-schema.sql` and `rent-a-film-data.sql` files.

2. Connect to the MySQL server using the `mysql` command-line client with the following command:

```
shell> mysql -u root -p
```

Enter your password when prompted. A non-`root` account can be used, provided that the account has privileges to create new databases.

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

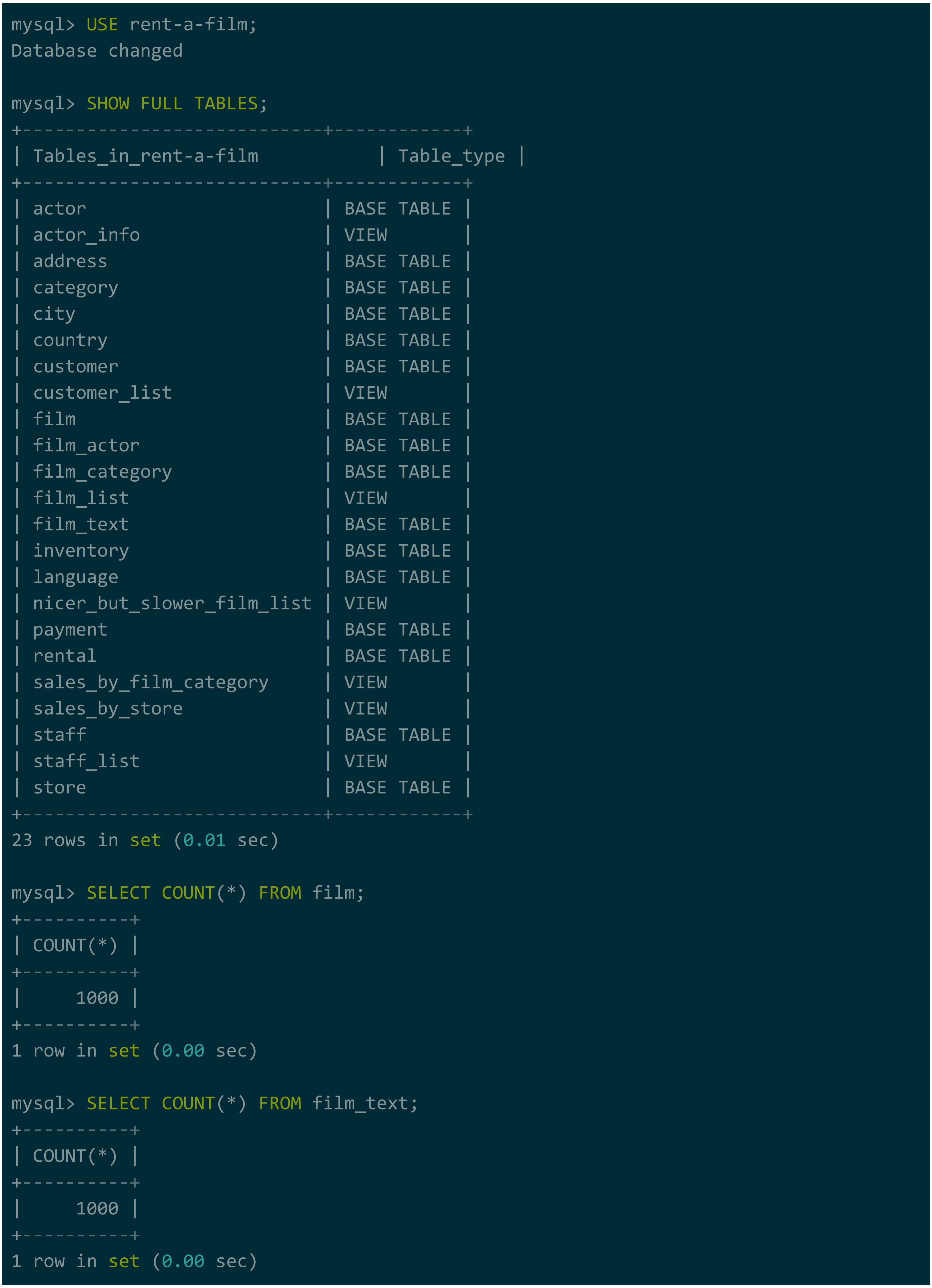
```
mysql> SOURCE C:/temp/rent-a-film-db/rent-a-film-schema.sql;

mysql> SOURCE C:/temp/rent-a-film-db/rent-a-film-data.sql;
```

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

Note
On Windows, use slashes rather than backslashes when executing the `SOURCE` command.

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



```
mysql> USE rent-a-film;
Database changed

mysql> SHOW FULL TABLES;
+----------------------------+------------+
| Tables_in_rent-a-film            | 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)
```

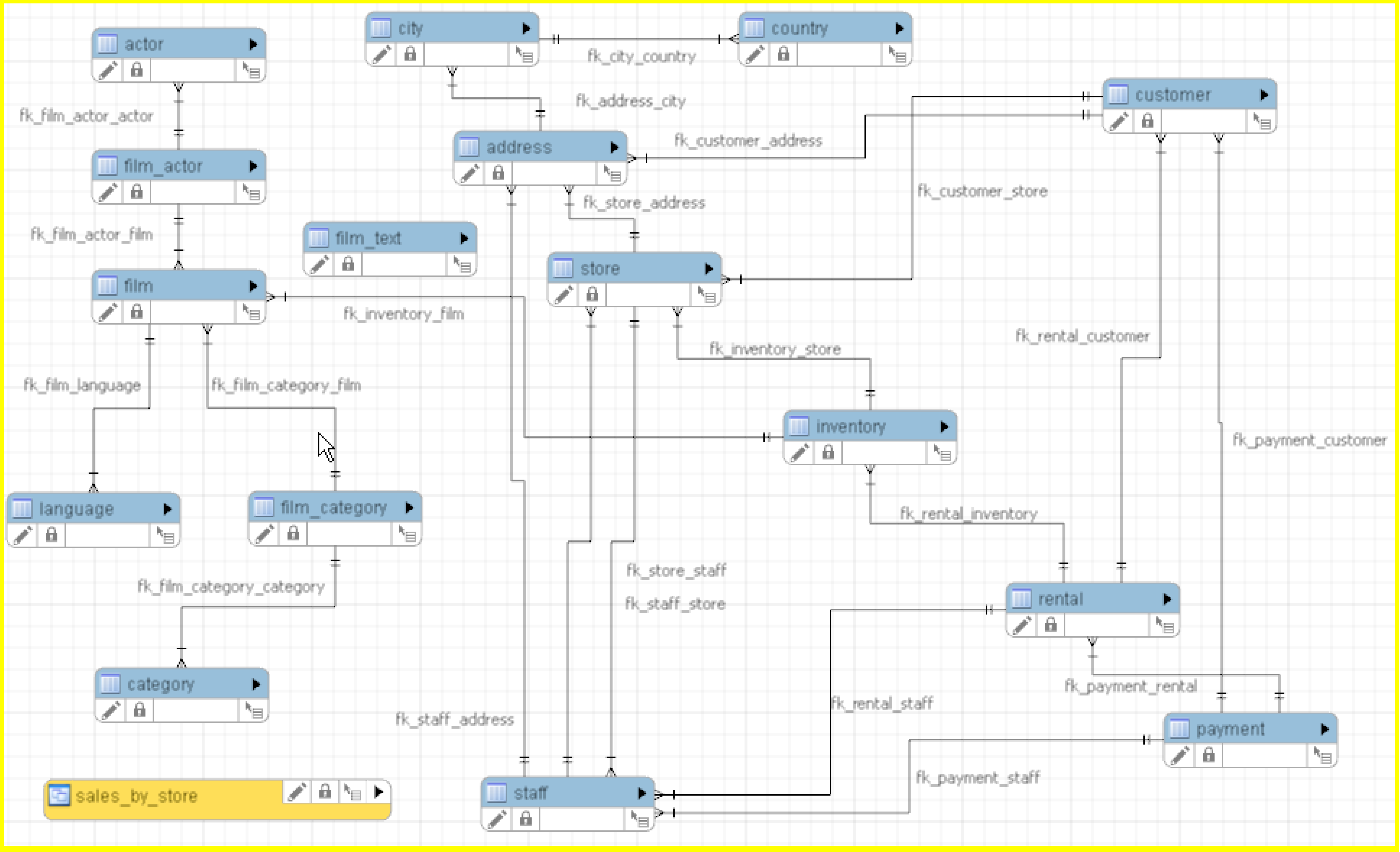


Figure 1 The Rent-A-Film Database Schema

2.1 Tables

The following sections describe the tables that make up the Rent-A-Film sample database, in alphabetic order.

2.1.1 The actor Table

The `actor` table lists information for all actors.

The `actor` table is joined to the `film` table by means of the `film_actor` table.

Columns

- `actor_id`: A surrogate primary key used to uniquely identify each actor in the table.
- `first_name`: The actor first name.
- `last_name`: The actor last name.
- `last_update`: When the row was created or most recently updated.

2.1.2 The address Table

The `address` table contains address information for customers, staff, and stores.

The `address` table primary key appears as a foreign key in the `customer`, `staff`, and `store` tables.

Columns

- `address_id`: A surrogate primary key used to uniquely identify each address in the table.
- `address`: The first line of an address.
- `address2`: An optional second line of an address.
- `district`: The region of an address, this may be a state, province, prefecture, etc.
- `city_id`: A foreign key pointing to the `city` table.
- `postal_code`: The postal code or ZIP code of the address (where applicable).
- `phone`: The telephone number for the address.
- `last_update`: When the row was created or most recently updated.
- `location`: A Geometry column with a spatial index on it.

2.1.3 The category Table

The category table lists the categories that can be assigned to a film.

The category table is joined to the film table by means of the film_category table.

Columns

- category_id: A surrogate primary key used to uniquely identify each category in the table.
- name: The name of the category.
- last_update: When the row was created or most recently updated.

2.1.4 The city Table

The city table contains a list of cities.

The city table is referred to by a foreign key in the address table and refers to the country table using a foreign key.

Columns

- city_id: A surrogate primary key used to uniquely identify each city in the table.
- city: The name of the city.
- country_id: A foreign key identifying the country that the city belongs to.
- last_update: When the row was created or most recently updated.

2.1.5 The country Table

The country table contains a list of countries.

The country table is referred to by a foreign key in the city table.

Columns

- country_id: A surrogate primary key used to uniquely identify each country in the table.
- country: The name of the country.
- last_update: When the row was created or most recently updated.

2.1.6 The customer Table

The customer table contains a list of all customers.

The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign keys.

Columns

- `customer_id`: A surrogate primary key used to uniquely identify each customer in the table.

- `store_id`: A foreign key identifying the customer “home store.Customers” are not limited to renting only from this store, but this is the store at which they generally shop.

- `first_name`: The customer first name.

- `last_name`: The customer last name.

- `email`: The customer email address.
- `address_id`: A foreign key identifying the customer address in the `address` table.

- `active`: Indicates whether the customer is an active customer. Setting this to `FALSE` serves as an alternative to deleting a customer outright. Most queries should have a `WHERE active = TRUE` clause.

- `create_date`: The date the customer was added to the system. This date is automatically set using a trigger during an `INSERT`.
- `last_update`: When the row was created or most recently updated.

2.1.7 The film Table

The `film` table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the `inventory` table.

The `film` table refers to the `language` table and is referred to by the `film_category`, `film_actor`, and `inventory` tables.

Columns

- `film_id`: A surrogate primary key used to uniquely identify each film in the table.

- `title`: The title of the film.

- `description`: A short description or plot summary of the film.

- `release_year`: The year in which the movie was released.

- `language_id`: A foreign key pointing at the `language` table; identifies the language of the film.

- `original_language_id`: A foreign key pointing at the `language` table; identifies the original language of the film. Used when a film has been dubbed into a new language.

- `rental_duration`: The length of the rental period, in days.

- `rental_rate`: The cost to rent the film for the period specified in the `rental_duration` column.

- `length`: The duration of the film, in minutes.

•replacement_cost: The amount charged to the customer if the film is not returned or is returned in a damaged state.

•rating: The rating assigned to the film. Can be one of G, PG, PG-13, R, or NC-17.

•special_features: Lists which common special features are included on the DVD. Can be zero
or more of: Trailers, Commentaries, Deleted Scenes, Behind the Scenes.

- last_update: When the row was created or most recently updated.

2.1.8 The film_actor Table

The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing the actor and film.

The film_actor table refers to the film and actor tables using foreign keys.

Columns:

•actor_id: A foreign key identifying the actor.

•film_id: A foreign key identifying the film.

•last_update: When the row was created or most recently updated.

2.1.9 The film_category Table

The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the film_category table listing the category and film.

The film_category table refers to the film and category tables using foreign keys.
Columns:

•film_id: A foreign key identifying the film.

•category_id: A foreign key identifying the category.

•last_update: When the row was created or most recently updated.

2.1.10 The film_text Table

The film_text table contains the film_id, title and description columns of the film table, with the contents of the table kept in synchrony with the film table by means of triggers on film table INSERT, UPDATE and DELETE operations

Columns

- `film_id`: A surrogate primary key used to uniquely identify each film in the table.
- `title`: The title of the film.
- `description`: A short description or plot summary of the film.

The contents of the `film_text` table should never be modified directly. All changes should be made to the `film` table instead.

2.1.11 The inventory Table

The `inventory` table contains one row for each copy of a given film in a given store.

The `inventory` table refers to the `film` and `store` tables using foreign keys and is referred to by the `rental` table.

Columns

- `inventory_id`: A surrogate primary key used to uniquely identify each item in inventory.
- `film_id`: A foreign key pointing to the film this item represents.
- `store_id`: A foreign key pointing to the store stocking this item.
- `last_update`: When the row was created or most recently updated.

2.1.12 The language Table

The `language` table is a lookup table listing the possible languages that films can have for their language and original language values.

The `language` table is referred to by the `film` table.

Columns

- `language_id`: A surrogate primary key used to uniquely identify each language.
- `name`: The English name of the language.
- `last_update`: When the row was created or most recently updated.

2.1.13 The payment Table

The `payment` table records each payment made by a customer, with information such as the amount and the rental being paid for (when applicable).

The `payment` table refers to the `customer`, `rental`, and `staff` tables.

Columns

•`payment_id`: A surrogate primary key used to uniquely identify each payment.

•`customer_id`: The customer whose balance the payment is being applied to. This is a foreign key reference to the `customer` table.

•`staff_id`: The staff member who processed the payment. This is a foreign key reference to the `staff` table.

•`rental_id`: The rental that the payment is being applied to. This is optional because some payments are for outstanding fees and may not be directly related to a rental.

•`amount`: The amount of the payment.

•`payment_date`: The date the payment was processed.

•`last_update`: When the row was created or most recently updated.

2.1.14 The rental Table

The `rental` table contains one row for each rental of each inventory item with information about who rented what item, when it was rented, and when it was returned.

The `rental` table refers to the `inventory`, `customer`, and `staff` tables and is referred to by the `payment` table.

Columns

•`rental_id`: A surrogate primary key that uniquely identifies the rental.

•`rental_date`: The date and time that the item was rented.

•`inventory_id`: The item being rented.

•`customer_id`: The customer renting the item.

•`return_date`: The date and time the item was returned.

•`staff_id`: The staff member who processed the rental.

•`last_update`: When the row was created or most recently updated.

2.1.15 The staff Table

The `staff` table lists all staff members, including information for email address, login information, and picture.

The `staff` table refers to the `store` and `address` tables using foreign keys, and is referred to by the `rental`, `payment`, and `store` tables.

Columns

- `staff_id`: A surrogate primary key that uniquely identifies the staff member.
- `first_name`: The first name of the staff member.
- `last_name`: The last name of the staff member.
- `address_id`: A foreign key to the staff member’s address in the `address` table.
- `picture`: A `BLOB` containing a photograph of the employee.
- `email`: The staff member’s email address.
- `store_id`: The staff member “home store. The” employee can work at other stores but is generally assigned to the store listed.
- `active`: Whether this is an active employee. If employees leave, their rows are not deleted from this table; instead, this column is set to `FALSE`.
- `username`: The user name used by the staff member to access the rental system.
- `password`: The password used by the staff member to access the rental system. The password should be stored as a hash using the `SHA2()` function.
- `last_update`: When the row was created or most recently updated.

2.1.16 The store Table

The `store` table lists all stores in the system. All inventory is assigned to specific stores, and staff and customers are assigned a “home store”.

The `store` table refers to the `staff` and `address` tables using foreign keys and is referred to by the `staff`, `customer`, and `inventory` tables.

Columns

- `store_id`: A surrogate primary key that uniquely identifies the store.
- `manager_staff_id`: A foreign key identifying the manager of this store.
- `address_id`: A foreign key identifying the address of this store.
- `last_update`: When the row was created or most recently updated.