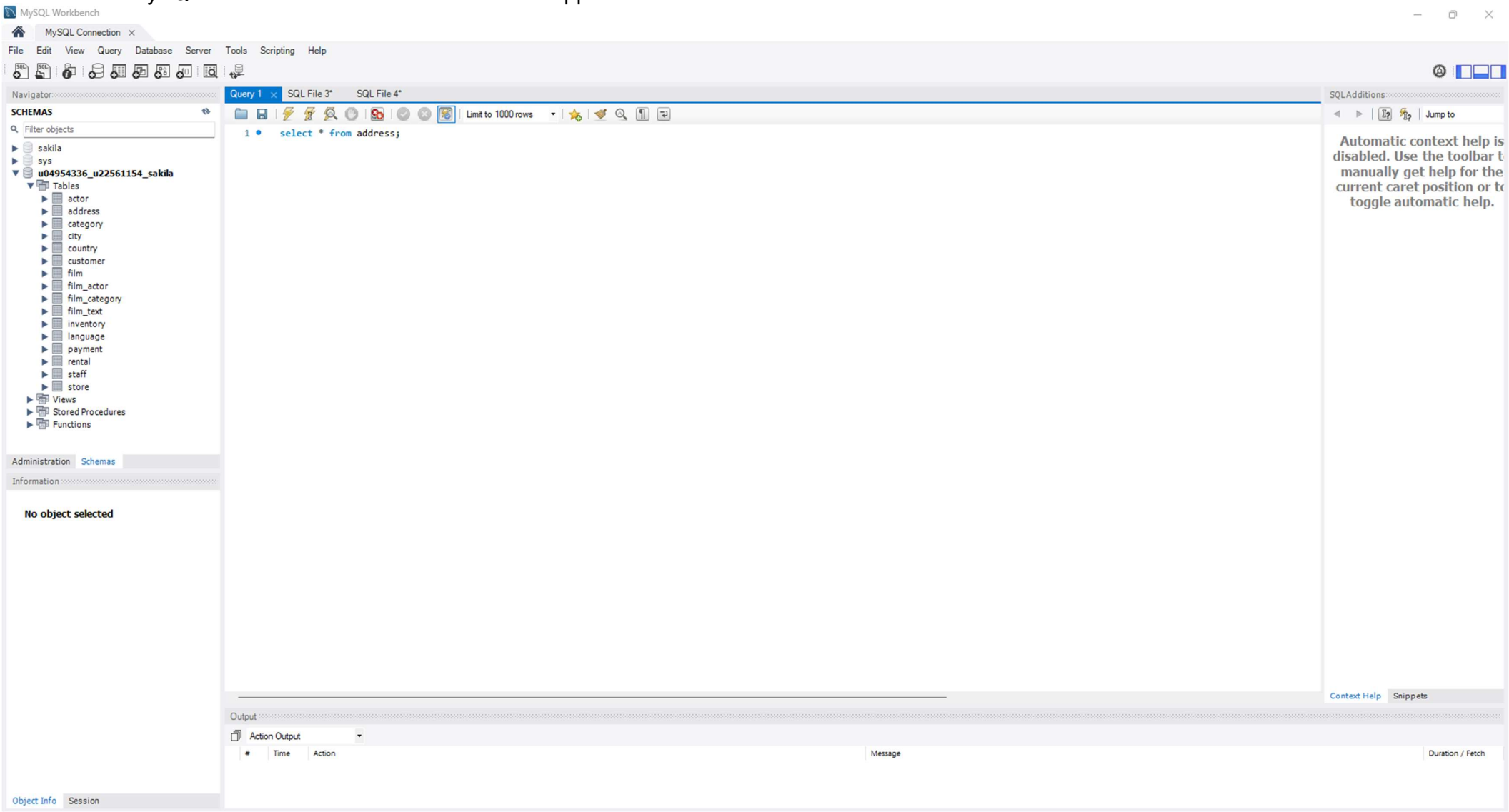
COS 221 PRAC 4

u04954336 and u22561154
Lloyd Creighton and Yi-Rou Hung

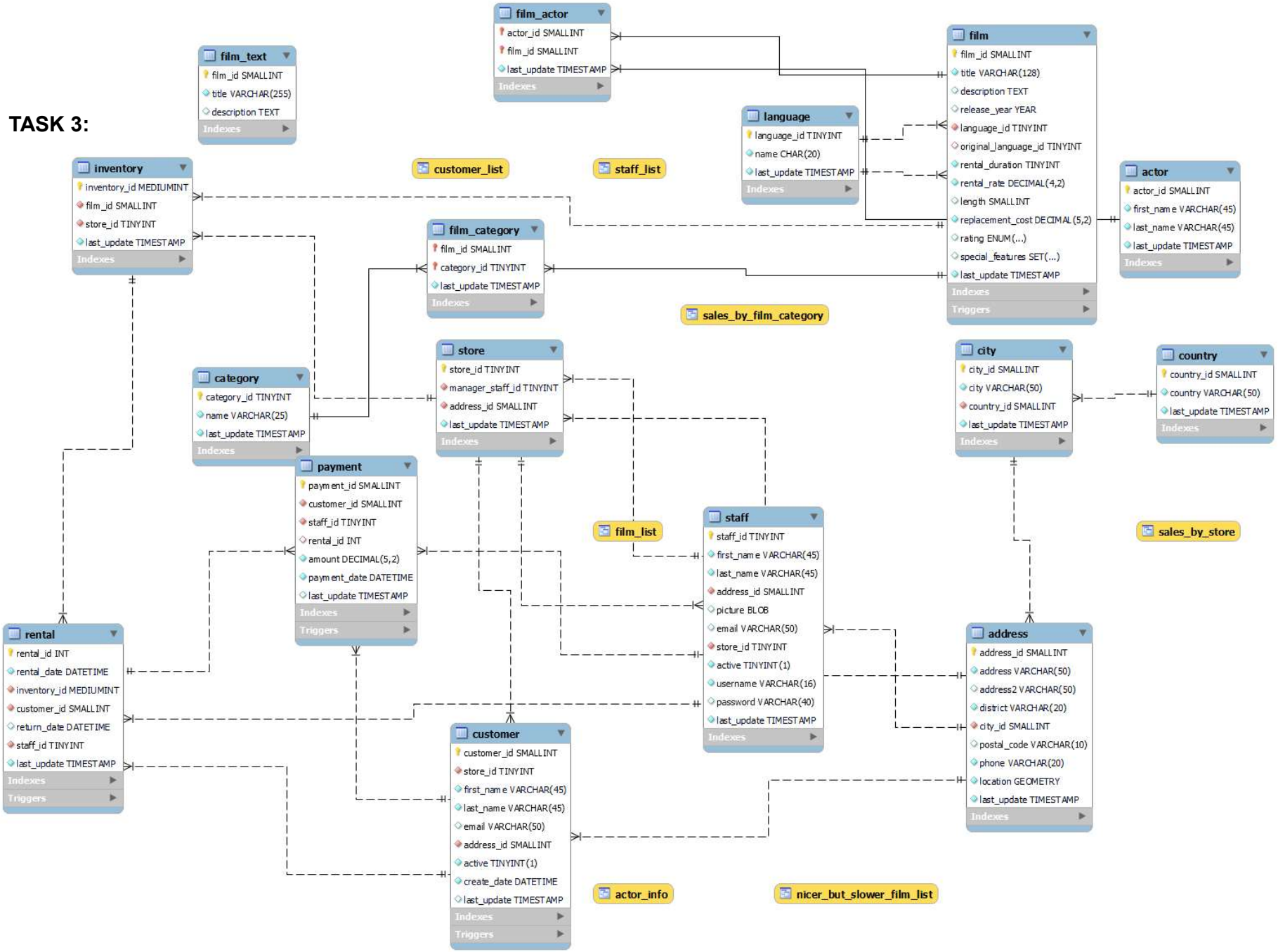
TASK 2:

MySQL Workbench was the chosen canned application.

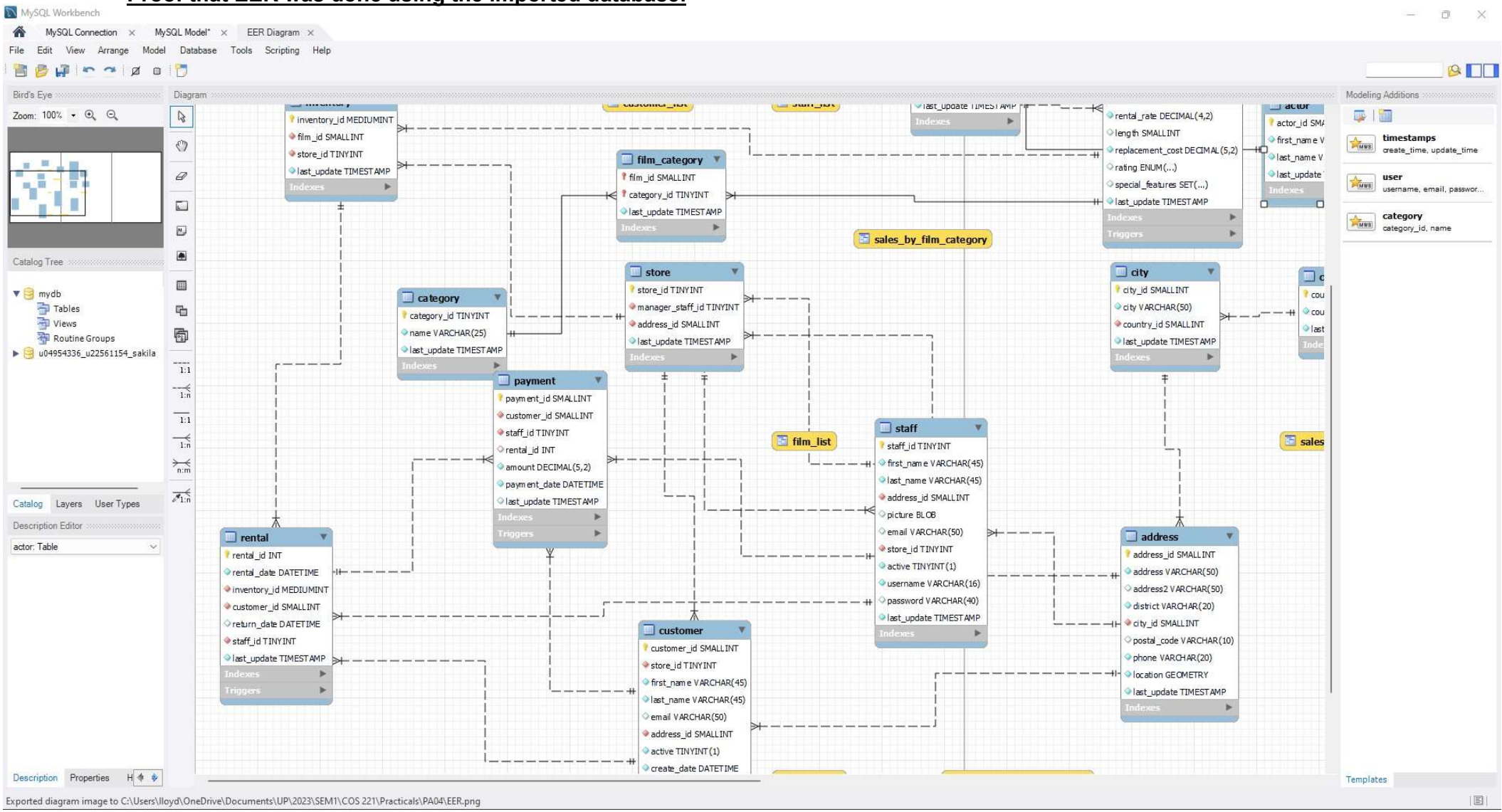


Zoomed in screenshot of the database:

TASK 3:


film_actor
actor_id SMALLINT
film_id SMALLINT
last_update TIMESTAMP
Indexes
film
film_id SMALLINT
title VARCHAR(128)
description TEXT
release_year YEAR
language_id TINYINT
original_language_id TINYINT
rental_duration TINYINT
rental_rate DECIMAL(4,2)
length SMALLINT
replacement_cost DECIMAL(5,2)
rating ENUM(...)
special_features SET(...)
last_update TIMESTAMP
Indexes
Triggers
film_text
film_id SMALLINT
title VARCHAR(255)
description TEXT
Indexes
language
language_id TINYINT
name CHAR(20)
last_update TIMESTAMP
Indexes
actor
actor_id SMALLINT
first_name VARCHAR(45)
last_name VARCHAR(45)
last_update TIMESTAMP
Indexes
inventory
inventory_id MEDIUMINT
film_id SMALLINT
store_id TINYINT
last_update TIMESTAMP
Indexes
customer_list
staff_list
film_category
film_id SMALLINT
category_id TINYINT
last_update TIMESTAMP
Indexes
sales_by_film_category
store
store_id TINYINT
manager_staff_id TINYINT
address_id SMALLINT
last_update TIMESTAMP
Indexes
city
city_id SMALLINT
city VARCHAR(50)
country_id SMALLINT
last_update TIMESTAMP
Indexes
country
country_id SMALLINT
country VARCHAR(50)
last_update TIMESTAMP
Indexes
category
category_id TINYINT
name VARCHAR(25)
last_update TIMESTAMP
Indexes
payment
payment_id SMALLINT
customer_id SMALLINT
staff_id TINYINT
rental_id INT
amount DECIMAL(5,2)
payment_date DATETIME
last_update TIMESTAMP
Indexes
Triggers
film_list
staff
staff_id TINYINT
first_name VARCHAR(45)
last_name VARCHAR(45)
address_id SMALLINT
picture BLOB
email VARCHAR(50)
store_id TINYINT
active TINYINT(1)
username VARCHAR(16)
password VARCHAR(40)
last_update TIMESTAMP
Indexes
sales_by_store
rental
rental_id INT
rental_date DATETIME
inventory_id MEDIUMINT
customer_id SMALLINT
return_date DATETIME
staff_id TINYINT
last_update TIMESTAMP
Indexes
Triggers
address
address_id SMALLINT
address VARCHAR(50)
address2 VARCHAR(50)
district VARCHAR(20)
city_id SMALLINT
postal_code VARCHAR(10)
phone VARCHAR(20)
location GEOMETRY
last_update TIMESTAMP
Indexes
customer
customer_id SMALLINT
store_id TINYINT
first_name VARCHAR(45)
last_name VARCHAR(45)
email VARCHAR(50)
address_id SMALLINT
active TINYINT(1)
create_date DATETIME
last_update TIMESTAMP
Indexes
Triggers
actor_info
nicer_but_slower_film_list

Proof that EER was done using the imported database:



STORE RELATION ANALYSIS:

The store relation holds data that identifies the separate DVD stores in the database. It contains:
a tinyint primary key store_id, which indexes the store. It is auto incremented to ensure a unique key.
a tinyint manager_staff_id which will reference to a staff_id in the staff table.
a smallint address_id which will reference to an address_id in the address table, which will store the address of the store.
a timestamp last_update which will store the time the record was last updated and must thus be changed to the time of the update every time it is updated.

The constraints:
A manager_staff_id must be unique and it has a foreign key constraint to the staff table, as in the staff member in the table must exist, and they cannot manage more than 1 store.
The store_id is unique, and thus cannot have any duplicates.
The address_id has a foreign key constraint; it cannot reference an address that does not exist. It also has a unique constraint; two stores cannot be at the same address.

The store relation links to the staff relation, address relation, inventory relation and customer relation.

The store has a single manager that is part of the staff, and each staff member is under one store, that means that a store will have several staff members attached to it. This is both a 1:1 relationship in terms of manager, and a 1:N relationship for the rest of the staff members (1 store has N staff members).

A store also has an address attached to it, which is referenced in the address field. This is a 1:1 relationship since in the store table, address must be unique.

Inventory has a foreign key relationship to a single store, that means that any given store will have many inventory records attached to it, to keep track of the stock in the store. This is a 1:N relationship between the store and the inventory, 1 store has N inventory records.

In the customer relation, there is a foreign key referencing a store_id, this means that each store will have several customers that are specifically registered at that store. This is a 1:N relationship between the store and customers, 1 store has N customers.