**Database Model: Cinema Movie Ticket**

**Booking System**

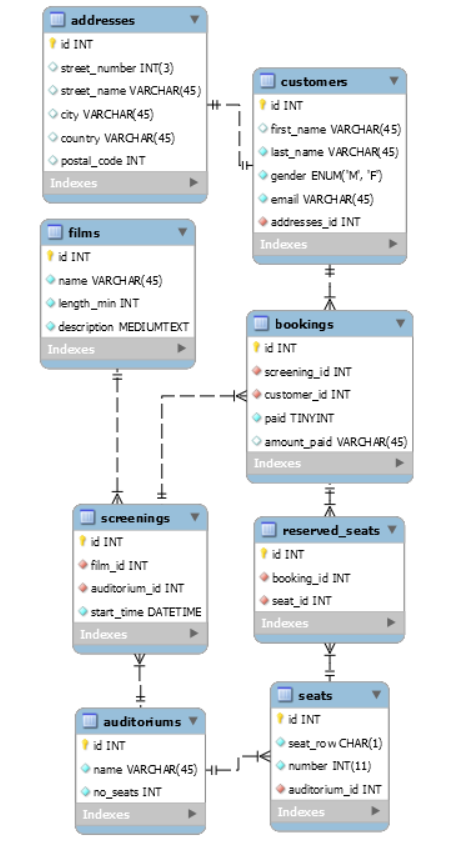
Ngo Trong Hieu (15943) – Software Engineering

# Requirements

Personally (individual work) create a relational database system (model and implementation) regarding specific subject (e-commerce system, system for handling personnel and activities of certain organization, system for ordering and registering goods, books/movies/songs repository, etc.). You can use MySQL database with phpMyAdmin panel or other tools. Task includes:

* Modelling of a system (creation of relational diagrams, ERD diagrams, establishing keys and providing normalization). [1]
* Creating tables and filling them with data (records). [2]
* Being familiar with data types and providing basic SQL queries. [3]
* Providing more advanced SQL queries (queries to multiple tables, aggregated queries and grouping). [4]
* Providing nested queries. [5]

# Relational/ERD diagrams [1]



# Descriptions

The ***films*** table contains data about films which will be shown in the cinema. The primary key is id, which is auto incremented like all primary keys in all other tables. The only mandatory data is title.

The ***auditorium*** table identifies all auditoriums in the cinema. All data is mandatory.

The ***screening*** table contains data of all screenings and all fields are mandatory. A screening must have a related movie, auditorium and start time. We can’t have two showings in same auditorium at the same time.

The ***seats*** table contains a list of all seats we have in auditoriums with each seat assigned to strictly one auditorium. All fields are mandatory.

The ***customers*** table lists all customers using the system. All fields are mandatory.

The ***addresses*** table.

The ***bookings*** table stores data about all of ticket reservations. The customer\_id would contain id of the customer who books ticket. In the same way, if tickets were sold, the attribute paid would be set to True.

The **reserved\_seats** table enables us to make a reservation or one payment for multiple seats. After the employee checks a few free seats on the interface, one record would be added to this table for each of them.

# Key establishments [1]

# Normalization [1]

# Used data types [1]

* **INT:** id. A normal-sized integer that can be signed or unsigned. If signed, the allowable range is from -2147483648 to 2147483647.
* **VARCHAR:** first\_name (customers table). A variable-length string between 1 and 255 characters in length.
* **CHAR(1):** seat\_row (seats table). A fixed-length string between 1 and 255 characters in length, in this case is 1.
* **BOOLEAN/TINYINT(1):** paid (bookings table)
* **ENUM(‘M’,’F):** gender (customers table). Define columns that can contain only a given set of values.
* **DATETIME:** start\_time (screenings table). A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59.
* **MEDIUMTEXT:** description (films table). A field with a maximum length of 16777215 characters.

# Basic queries [3]

# Advanced queries [4]

# Nested queries/Subqueries [5]