**Piriya Saengsuwarn**

**Peifeng Jing**

**Mehdi Jaagoub**

# Database Description

The database is designed for an online movie store web application that provides online movie streaming services. Customers can sign up for free on the website or pay to upgrade their membership to receive additional benefits such as promotional discount for renting or buying movies. The web application provides general information about movies such as cast members and directors, of which customers can discover movies that has common directors or actors to their favorite classics. In addition, the web application also includes social feature that customer can write a review to provide community some recommendations and comments of the movie they watched.

The database is also structured to categorize movies, track orders, check sale volume and demand from customers. It can provide data and information to managers, help them to know more about movie market and preference of customers, and help them to make profitable decisions.

To describe tables in our database, we give you several example scenarios that can demonstrate the relationship and application of the tables:

## Registering an Account

CUSTOMER entity is designed for customer registration on the website. The web application then insert a new row into the CUSTOMER table. A new CustomerID, primary key, is automatically generated to uniquely identify the customer when a customer registered on the website. MembershipTypeID attribute in the CUSTOMER table is assigned to track type of customers and identify what kind of discount can be provided to customers, which is decided by another entity which is called PROMOTION.

## Browsing Movies

The MOVIE entity is used for movie storage. A new tuple will be created when website manager decides to add a new movie on the web. This entity contains the name, description, release date and rating of movies. There are several entities, which provides tools to customers for searching their favorite movies.

## Browsing Movies by Genre

Customer can look up movies that has their favorite genre by clicking on the genre tag in the movie browsing page, which is then used to look into the GENRE table to obtain GenreID by matching GenreName. MovieID and GenreID in MOVIE\_GENRE are used to find out movie from genre.

## Browsing Movies by Cast Member

Customer can find movies by looking up their favorite actor or director. CastMemberID is first identified in CAST\_MEMBER entity by matching CastMemberFirstName and CastMemberLastName. Then movies can be found when MovieID with certain CastMemberID are returned. Since one cast member can have one or more role (e.g. Stephen King was both the director and writer of Maximum Overdrive), another associate entity table, CAST\_ROLE, is required to list all the roles from ROLE table that a certain cast member have in a movie.

## Buying/Renting a Movie

Customers can buy/rent movies after they have found what they want, which will add new tuples to ITEM entity. The ITEM is created to describe the base price and optional rental duration of the movie, since there can be more than one offers for a certain movie.

## Adding an Item to the Order

When customers are trying to check out, new tuples will be created in ORDER entity, while discount will be added in PROMOTION entity determined by MembershipTypeID and ItemID. Since there can be more than one items in one order, a LINE\_ITEM associate entity is created to support many-to-many relationship between PROMOTION and ORDER.

## Adding a Payment Method

PAYMENT is created as a table to record each payment made on the website, which contains PaymentID, PaymentTypeID, IsVerified, CardNumber and CardHolderName. The PAYMENT\_TYPE has a relation with PAYMENT and works as a looking-up table.

## Making a Payment to the Order

Customers can make a payment to the order by selecting the payment method in the PAYMENT table that has the same foreign key of CustomerID. PaymentID in PAYMENT table is used to identity corresponding OrderID in ORDER entity. The ProcessedDate attribute in the ORDER table is recorded to keep track of whether movie rental is expired or not.

To look up all the movies that a customer owns, the application takes the CustomerID of a certain customer to get all the payment method that they have from the PAYMENT table. The PaymentID of the PAYMENT table is then used to list all the order from the ORDER table. From the ORDER table the customer can list all the movie they own by navigating from LINE\_ITEM through PROMOTION to ITEM and eventually MOVIE table.

## Writing a Review

A customer can write a review for a movie with review comments and review score (out of 10) which will be recorded in the REVIEW table. One movie can have many reviews from many customers. With this relationship, the REVIEW table is an associate entity table between CUSTOMER and MOVIE table which are identified by ReviewID.