**Database Project 2**

# **Phase 1: Database Design**

## **Conceptual Model**

The conceptual model outlines the main entities and their relationships at a high level.

### **Entities and Relationships:**

1. Customer: Represents users who rent movies or maintain a wishlist.
   * Relationships:
2. Rents Rental
3. Has Wishlist
4. Movie: Represents movies available for rental.
   * Relationships:
5. Belongs to multiple Subcategories.
6. Has multiple Advisory.
7. Features multiple Actors.
8. Directed by one Director.
9. Included in multiple Wishlist.
10. Rented via multiple Rental.
11. Category: Represents movie categories (e.g., Drama) with possible subcategories.
    * Relationships:
12. Has multiple Subcategories.
13. Subcategory: Represents subcategories under main categories (like Thrillers under Drama).
    * Relationships:
14. Belongs to one Category.
15. Associated with multiple Movies.
16. Advisory: Represents content advisories (e.g., Violence, Nudity).
    * Relationships:
17. Associated with multiple Movie
18. Rental: Represents a movie rental transaction.
    * Relationships:
19. Linked to one Customer
20. Linked to one Movie
21. Actor/Actress: Represents actors/actresses in movies.
    * Relationships:
22. Plays roles in multiple Movie
23. Director: Represents directors of movies.
    * Relationships:
24. Directs multiple Movie
25. Wishlist: Represents a customer’s list of desired movies.
    * Relationships:
26. Linked to one Customer
27. Contains multiple Movie

### **Assumptions:**

1. Subcategory Hierarchy: The hierarchy is limited to one level (main category → subcategory).
2. Movie-Subcategory Association: Movies are associated with subcategories (e.g., Thrillers) rather than directly with main categories, but the Subcategory table links to Category, allowing indirect association with main categories (e.g., Drama).
3. One Director per Movie: A movie is directed by exactly one director for simplicity, as the requirements don’t specify multiple directors.
4. Customer Ratings: Customer ratings for rentals (1–5 stars) are optional, as not all rentals may have a rating.
5. Credit Card Storage: Credit card details are stored per rental, not per customer, though customers may have default credit card details.
6. Advisories: The 17 Alberta advisories (like Coarse Language, Gory Scenes) are predefined and shared across movies, with qualifiers like “Crude” and “Gory” embedded in the advisory name (like “Gory Scenes”).
7. Email Validation: Email addresses for customers, actors, and directors follow a standard format (like user@domain.com) and are unique.
8. Movie Availability: The “Coming Soon” binary attribute flags movies not yet available, but no additional availability status is tracked.

## **Logical Model**

The Logical Model expands the Conceptual Model by defining entities, attributes, primary keys (PKs), foreign keys (FKs), and resolving many-to-many relationships with tables. It uses the separate Subcategory table and incorporates all requirements, including Alberta ratings (G, PG, 14A, 18A, R) and advisories.

### **Entities, Attributes, PKs, FKs, and Relationships:**

1. Customer
   * Attributes:
     + CustomerID (PK)
     + FirstName (mandatory)
     + Surname (mandatory,)
     + Email (mandatory)
     + Address (mandatory)
     + PostalCode (mandatory)
     + PhoneNumber (mandatory)
     + DefaultCreditCardNum
     + DefaultCreditCardType
   * Relationships:
     + One-to-Many with Rental (CustomerID as FK in Rental)
     + One-to-Many with Wishlist (CustomerID as FK in Wishlist)
2. Category
   * Attributes:
     + CategoryID (PK)
     + CategoryName (mandatory)
   * Relationships:
     + One-to-Many with Subcategory (CategoryID as FK in Subcategory)
3. Subcategory
   * Attributes:
     + SubcategoryID (PK)
     + SubcategoryName (mandatory)
     + CategoryID (FK,mandatory)
   * Relationships:
     + Many-to-One with Category (CategoryID as FK in Subcategory)
     + Many-to-Many with Movie (via MovieSubcategory, SubcategoryID as FK in MovieSubcategory)
4. MovieSubcategory
   * Attributes:
     + MovieID (PK,FK)
     + SubcategoryID (PK,FK)
   * Relationships:
     + Many-to-One with Movie (MovieID as FK in MovieSubcategory)
     + Many-to-One with Subcategory (SubcategoryID as FK in MovieSubcategory)
5. Movie
   * Attributes:
     + MovieID (PK)
     + Title (mandatory)
     + Duration (mandatory)
     + SDRentalPrice (mandatory)
     + HDRentalPrice (mandatory)
     + Rating (mandatory)
     + NewRelease (mandatory, default 0)
     + MostPopular (mandatory, default 0)
     + ComingSoon (mandatory, default 0)
     + DirectorID (FK, mandatory)
   * Relationships:
     + Many-to-One with Director (DirectorID as FK in Movie)
     + Many-to-Many with Subcategory (via MovieSubcategory, MovieID as FK in MovieSubcategory)
     + Many-to-Many with Advisory (via MovieAdvisory, MovieID as FK in MovieAdvisory)
     + One-to-Many with Rental (MovieID as FK in Rental)
     + One-to-Many with Wishlist (MovieID as FK in Wishlist)
     + Many-to-Many with Actor (via MovieActor, MovieID as FK in MovieActor)
6. Advisory
   * Attributes:
     + AdvisoryID (PK)
     + ShortDescription (mandatory)
     + FullDescription (mandatory)
   * Relationships:
     + Many-to-Many with Movie (via MovieAdvisory, AdvisoryID as FK in MovieAdvisory)
7. MovieAdvisory
   * Attributes:
     + MovieID (PK,FK)
     + AdvisoryID (PK,FK)
   * Relationships:
     + Many-to-One with Movie (MovieID as FK in MovieAdvisory)
     + Many-to-One with Advisory (AdvisoryID as FK in MovieAdvisory)
8. Rental
   * Attributes:
     + RentalID (PK, Integer)
     + CustomerID (FK,mandatory)
     + MovieID (FK, mandatory)
     + RentalDate (mandatory)
     + StartViewingDate
     + ExpiryDate
     + CustomerRating
     + RentalAmount (mandatory)
     + CreditCardNum (mandatory)
     + CreditCardType (mandatory)
   * Relationships:
     + Many-to-One with Customer (CustomerID as FK in Rental)
     + Many-to-One with Movie (MovieID as FK in Rental)
9. Actor
   * Attributes:
     + ActorID (PK)
     + FirstName (mandatory)
     + LastName (mandatory)
     + DateOfBirth (mandatory)
     + Email (mandatory)
   * Relationships:
     + Many-to-Many with Movie (via MovieActor, ActorID as FK in MovieActor)
10. Director
    * Attributes:
      + DirectorID (PK)
      + FirstName (mandatory)
      + LastName (mandatory)
      + DateOfBirth (mandatory)
      + Email (mandatory)
    * Relationships:
      + One-to-Many with Movie (DirectorID as FK in Movie)
11. Wishlist
    * Attributes:
      + WishlistID (PK)
      + CustomerID (FK, mandatory)
      + MovieID (FK, mandatory)
      + DateAdded (mandatory)
    * Relationships:
      + Many-to-One with Customer (CustomerID as FK in Wishlist)
      + Many-to-One with Movie (MovieID as FK in Wishlist)
12. MovieActor
    * Attributes:
      + MovieID (PK,FK)
      + ActorID (PK,FK, Integer, part of composite key)
      + RoleName (mandatory)
    * Relationships:
      + Many-to-One with Movie (MovieID as FK in MovieActor)
      + Many-to-One with Actor (ActorID as FK in MovieActor)