Project 1: Assessment of Modernizing the eMovies Model

Jasminder Garcha
CSCI381: Data Modeling
September 14, 2022

INTRODUCTION

Project 1 is a high-level assessment of how I transformed the eMovies Model into the modernized eMoviesSqlServer2019. In the preceding homework, I walked through my process on how I incrementally modernized eMovies. I will now focus on the final revisions I settled on, rather than the steps it took to get there. A brief insight into why I altered the model as such may be integrated into the list. This applies to the Conceptual, Logical, and Physical Data Models.

CONCEPTUAL DATA MODEL (CDM)

1) Add Business Objects and their Definitions:

1) **Person**: A single human being.
2) **Address**: Where a Person or Store is located.
3) **Star**: A Person acting in a Movie.

2) Add Business Rules:

1) A Person/Store can have **one or more** Addresses (Primary & Secondary).
2) Customer, Employee, & Star **is a** Person.
3) A Star can **act** in **one or more** Movies & a Movie can **star one or more** stars.
4) Credit Card, Electronic Payment, & Check **is a** form of Payment.

LOGICAL DATA MODEL (LDM)

1) Change spelling of Entities and their Attributes from ALL CAPS to First Letter Of Each Word Capitalized:

1) Entity PAYMENT → Payment; remove "_" from Movie_Store → MovieStore
2) Attribute payment transaction number →Payment Transaction Number

2) Remove short descriptors (self-documenting):

1) address, address 2 → Primary Address, Secondary Address.
2) No → Number, exp → Expiration, epay → Electronic Payment, soc sec number → Social Security Number
3) email → Customer Email, Employee Email
4) Movie: rating → Movie Rating, …

3) Append "Id" to any attribute ending with the word "Number" (unique identifier):

1) Payment Transaction Number → Payment Transaction Number Id, …

4) Add Star entity:

1) Add Star Id (key), Person Id (foreign key), and Movie Id (foreign key) attributes.

5) Add intermediate entity Movie Star (many-to-many relationship between Movie and Star):

1) Add Star Id (foreign key) and Movie Id (foreign key) attributes.

6) Normalization:

Payment entity:

1) Remove Credit Card Number, Credit Card Expiration, Credit Card Type attributes and add to Credit Card entity.
2) Remove Check Bank Number, Check Number attributes and add to Check entity.
3) Remove Electronic Payment Vendor Number, Electronic Payment Account Number attributes and add to Electronic Payment entity.

4) Add Payment Transaction Number Id (foreign key) attribute to Credit Card, Check, and Electronic Payment entities.

Address entity:
1) Add Address Id (key), Address State, Address City, Address Zipcode attributes.
2) Remove Address, City, Zipcode attributes from Customer, Employee, Store (add Address Id - FK) entities.

Person entity:
1) Remove First Name, Last Name, Address Id, Email, Phone Number, and Social Security Number attributes (if applicable) from Customer, Employee, Star entities and add to Person.
2) Add Person Id (key) attribute to Person entity & Person Id (foreign key) attribute to Customer, Employee, Star entities.

7) Determine whether attributes are optional or non-optional:

Social Security Number – non-optional, Secondary Address – optional, …

PHYSICAL DATA MODEL (PDM)

Forward engineer the LDM.

1) Convert table and column names to PascalCase (by collapsing the spaces between entity/attribute names in the LDM):

1) ~~MO_RENT_REC~~ Movie Rental Record → MovieRentalRecord
2) ~~soc_sec_num~~ Social Security Number → SocialSecurityNumber, …

2) Fully qualify table names with schemas:

1) Accounting.Payment, Accounting.MovieRentalRecord, …
2) Movie.MovieStore, Movie.MovieCopy, …

3) Tighten column data types using business rules, for performance/storage (MS SQL Server specific):

1) EmployeeNumberId: varchar(20) → integer
2) CreditCardExpiration: datetime → date
3) OverdueCharge: integer → numeric, …

4) Required rows altered to NOT NULL (non-optional attributes) & non-required rows altered to NULL (optional attributes):

1) SocialSecurityNumber … NOT NULL
2) SecondaryAddress … NULL

5) Validation:

1) Default values: EmployeeSalary of minimum base salary, …
2) Check constraints: EmployeeAge >= 18, …
3) Triggers: catch invalid data from the front-end.

6) Strengthen weak entities (often bridge tables) with Surrogate Keys:

1) MovieStore: Push MovieNumberId and StoreNumberId columns into its body and add a primary key of MovieStoreId.
2) Same with MovieCopy and MovieStar.

7) Mask personally identifiable information (PII) with SHA256 hash:

1) Columns in Person table:
   Particularly SocialSecurityNumber, …