**INFO6210 DATA MANAGEMENT AND DATABASE DESIGN**

**Assignment 2: Physical Data Model and Normalization**

**By:**

**Ashwin Lakshman 001353233**

**Navaneeta Naik**

**DOMAIN:** We have chosen a movie database domain for this assignment.

**DATA SOURCES:**

1. IMDB movie data scraped from web page.
2. Twitter User data collected using Tweepy API
3. Director Data collected using Twitter API

**CONCEPTUAL MODEL:**

The conceptual model contains 3 entities:

1. IMDB Movies: The attributes are movie\_names, year\_release, imdb\_ratings, metascores, votes, movie\_description, certificate, runtime, genre,director\_name , star\_cast, gross\_value.

The IMDB entity is related to the User data entity by a one to many relationship. One movie can be commented on, by many users. The IMDB entity is connected with the director entity by a many to one relationship since one director can have multiple movies.

1. Tweepy Data: attributes are post, tweet\_id, retweeted, created time, screen name, movie name.

The User entity is connected with the Director entity using a one to many relationship. One Movie can be watched by and commented on, by a number of users.

1. Twitter Data: attributes are post, tweet\_id, retweeted, created time, director name, movie name.

The director entity is connected with the user entity by using a one to many relationship, as a number of users can be followers to a particular director.

**ENTITY-RELATIONSHIP MODEL:**

|  |
| --- |
|  |

**QUESTIONS ABOUT THE CONCEPTUAL MODEL:**

**1. What are the ranges, data types and format of all of the attributes in your entities?**

The following Datatypes are being used by the attributes: string, int, Bigint, float, Datetime. The ranges have not been set manually. Hence, the attributes are populated with the default ranges.

**USERS TABLE:**

|  |  |
| --- | --- |
| post | varchar(50) |
| tweet\_id | bigint(20) |
| retweeted | varchar(50) |
| timestamp | datetime |
| screenname | varchar(50) |
| name | varchar(50) |

**IMDB MOVIES TABLE:**

|  |  |
| --- | --- |
| movie\_id | int(11) |
| movie\_names | varchar(50) |
| year\_release | int(11) |
| imdb\_ratings | double |
| metascores | int(11) |
| votes | int(11) |
| movie\_description | varchar(50) |
| certificate | int(11) |
| runtime | int(11) |
| genre | varchar(50) |
| director\_name | varchar(50) |
| gross\_value | int(11) |

**DIRECTORS TABLE:**

|  |  |
| --- | --- |
| post | varchar(50) |
| tweet\_id | bigint(20) |
| retweeted | varchar(50) |
| timestamp | datetime |
| screenname | varchar(50) |
| Director\_ name | varchar(50) |

**2. When should you use an entity versus attribute? (Example: address of a person could be modeled as either)**

Entity vs attributes can be used whenever we have an entity which can be split into different entities and framed as different columns. In our database we can take genre as an entity which can be split into different combination of entities. For ex: genre has attributes like (comedy, action) and (comedy ,drama).

**3. When should you use an entity or relationship, and placement of attributes? (Example: a manager could be modeled as either)**

It is used when the tables have correlating data which are dependent on each other or related to each other.

**4. How did you choose your keys? Which are unique?**

The keys for each table was identified by the unique attribute for each row.

The Primary key for the Movies table is movie\_id. (Foreign key - director\_name)

The primary key for the Users table is tweet\_id. (Foreign key - movie\_name)

The Primary key for the directors table is director\_name (foreign key - movie\_name)

**5. Did you model hierarchies using the “ISA” design element? Why or why not?**

No, the model does not use an “IS A” design element. It instead implements the “HAS A” design element, since the data is organized in such a way: The movie has a director, the director has a list of followers and the users have watched a number of movies.

**6. Were there design alternatives? What are their tradeoffs: entity vs. attribute, entity vs.relationship, binary vs. ternary relationships?**

Yes there were design alternatives, to include further data about actors and production houses using other Sources, but was left out to limit scope of project.

A ternary relationship has been implemented in this design.

**7. Where are you going to find real-world data to populate your model?**

The real word data is obtained from multiple sources:

1. Web scraped from IMDB webpage
2. Twitter data using API clients.

**AUDIT:**

The data was audited to confirm that it did not require cleaning as all the obtained data was usable.

**CONTRIBUTIONS:**

Ashwin extracted data from Twitterusing API clients, ER Diagram, and answered half the questions.

Navaneeta scraped data from IMDB using BS, Conceptual Model, and answered the other half.

**REFERENCES:**

1. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
2. <http://docs.tweepy.org/en/latest/api.html>
3. <https://developer.twitter.com/en/docs>
4. <https://docs.python.org/2/library/sqlite3.html>