**INFO 6210 Data Management and Database Design Physical Data Model and Social Media Assignment 2 Report**

**ABSTRACT**

This project aims to find the details of the **Books** through various social media sources like Twitter, and Reddit and other sources like goodreads website, goodreads API call and datasets. This domain is converted into entities that represent consumers, producers and companies. We can make a conceptual and physical database schema of the data. It Is then populated into sqlite database to query the it and gain information by answering questions.

**PART I – CONCEPTUAL MODEL**

The domain exhibited here is a Books Domain in which the consumer is our User, Producer is Author and Company is Publishers. From the Twitter API, we have gathered Author and Publisher details (like Followers, Favourited etc) along with the Tweets of Author and Publishers which are then stored in the database. Similarly from Reddit API, we have collected Author and User Posts are stored into separate tables. Below are the attributes and tables used along with their keys.

**CONCEPTUAL DATABASE SCHEMA**

For the company, Nerd Analytics we have designed the conceptual database in such a way that the data is present in the entities accurately and is stored in the sqlite database. Two social media account (Twitter and Reddit) are used in order to get the data and Goodreads website is scrapped for the books data. Other relevant book data is taken from Kaggle data source.

The data is planned to be stored in different relevant tables. From Reddit API, we have created two tables, *Reddit\_Posts* and *Reddit\_User\_Posts* in which we have taken likes, dislikes, posts and hashtags.

From the Twitter API, we have gathered Publisher’s and Author’s followers, tweets, hashtags, retweet counts, friends count and timestamp(when it was posted).This data is distributed accordingly to four tables, *Twitter\_Publisher*, *Twitter\_Author*, *Publisher\_Tweets* and *Author\_Tweets.* The other four tables, *Publisher*, *Author*, *Prices* and *Books* are created in a manner which has publisher, author, books and price related detailed information. Now if we want any relevant information we can use joins between these tables and have the desired output.

Below is the conceptual design of the Books domain. Each entity is shown in the box with the Table Name at the top. The Attributes are written below the table name. Each entity is connected with other entity with the cardinalities as depicted by the crow’s foot notation. For example, Publisher and Publisher\_Tweets has one to many cardinality.

A screenshot of a cell phone

Description automatically generated

**QUESTIONS ABOUT CONCEPTUAL MODEL**

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

**Twitter\_Author**:- This table contain the Author’s description and help us to find his Twitter followers, his tweets etc.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| Name | nvarchar(100) |
| ID | nvarchar(100) |
| Created\_AT | timestamp |
| Favourited | int |
| Followers | int |
| Description | nvarchar(4000) |
| Friend\_Count | int |
| Tweets | int |

**Twitter\_Publisher**:- This table contain the Publisher’s description and help us to find his Twitter followers, his tweets etc.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| Name | nvarchar(100) |
| ID | nvarchar(100) |
| Created\_AT | timestamp |
| Favourited | int |
| Followers | int |
| Description | nvarchar(4000) |
| Friend\_Count | int |
| Tweets | int |

**Author\_Tweets**:- This table contain the Author’s tweets and help us to know how many people has retweeted his posts, what are his popular hashtags and people who have favourited it.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| ScreenName | nvarchar(100) |
| Name | nvarchar(100) |
| Tweets | nvarchar(4000) |
| HashTags | nvarchar(100) |
| Favourites | int |
| RetweetCount | int |
| Time | timestamp |
| Original\_User | nvarchar(100) |
| Retweeted | nvarchar(100) |

**Publisher\_Tweets**:- This table contain the Publisher’s tweets and help us to know how many people has retweeted his posts, what are his popular hashtags and people who have favourited it.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| ScreenName | nvarchar(100) |
| Name | nvarchar(100) |
| Tweets | nvarchar(4000) |
| HashTags | nvarchar(100) |
| Favourites | int |
| RetweetCount | int |
| Time | timestamp |
| Original\_User | nvarchar(100) |
| Retweeted | nvarchar(100) |

**Reddit\_Posts**:- This table has the posts and their up votes(like) and down votes(dislike) for different users.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| Title | nvarchar(100) |
| Post | nvarchar(4000) |
| Post\_Author | nvarchar(100) |
| Post\_ID | nvarchar(100) |
| Post\_Time | timestamp |
| Post\_Edited | nvarchar(100) |
| Up\_Votes | int |
| Down\_Votes | int |

**Reddit\_User\_Posts**:- This table contains the posts and their popular hashtags(subreddits).

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| Name | nvarchar(100) |
| Post | nvarchar(4000) |
| Time | timestamp |
| SubReddit | nvarchar(100) |

**Author**:- This table contains the author details like name, gender, birth date etc.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| AuthorName | nvarchar(100) |
| WorksCount | int |
| Gender | nvarchar(100) |
| About | nvarchar(100) |
| BornDate | nvarchar(100) |
| DiedDate | nvarchar(100) |
| PublicationInfluenced | nvarchar(100) |

**Books**:- This tale has book details and their respective ratings along with publication year.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| Goodreads\_book\_id | int |
| Work\_id | int |
| Books\_count | int |
| Isbn | nvarchar(100) |
| New\_ISBN | nvarchar(100) |
| Original\_publication\_year | nvarchar(100) |
| Original\_title | nvarchar(1000) |
| Language\_code | nvarchar(100) |
| Average\_rating | nvarchar(100) |
| Rating\_count | int |
| Work\_rating\_count | int |
| Work\_text\_reviews\_count | int |
| Ratings\_1 | int |
| Ratings\_2 | int |
| Ratings\_3 | int |
| Ratings\_4 | int |
| Ratings\_5 | int |

**Prices**:- This table contains the price of the books and the characters present in that particular book.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| Title | nvarchar(1000) |
| ISBN | nvarchar(100) |
| Characters | nvarchar(2000) |
| Price | nvarchar(100) |
| Settings | nvarchar(2000) |
| Awards | nvarchar(4000) |

**Publishers**:- This table has Publishers information and will help us know the books published by a publisher and their reviews.

|  |  |
| --- | --- |
| **Attribute Name** | **Type** |
| Title | nvarchar(1000) |
| Issue | nvarchar(100) |
| BooksFormat | nvarchar(100) |
| Authors | nvarchar(100) |
| Pages | nvarchar(100) |
| Description | nvarchar(4000) |
| AverageRating | nvarchar(100) |
| RatingCount | int |
| TestReviews | int |
| PublicationYear | nvarchar(100) |
| Publisher | nvarchar(100) |
| Language | nvarchar(100) |
| Ebook | int |
| ISBN | nvarchar(100) |
| NewISBN | nvarchar(100) |

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

An entity is an independent object which has several attributes where as an attribute is a characteristic that is associated to an entity. So, if you would like to drill down further into an attribute of an entity, a separate entity for that attribute can be created and now this entity will have its own attributes. For eg, we have an Author entity that contains data for the Authors, Name, gender etc are the authors attributes. Also, we have an attribute Author in the publisher table because the publisher works with the author to publish books. Here, the field author is an attribute.

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

We choose an entity based on the design and the information required. If we need more information regarding an attribute, we create a separate entity for that attribute so that it can have other attributes associated to it.

1. How did you choose your keys? Which are unique?

This number is universal and is unique for each book. We have ISBN as an attribute in our Books table that acts as primary key here similarly Publishers use the ISBN for distribution purposes as well hence we have the field ISBN in the Publishers table as well. When we join the Books and the Publishers the attribute ISBN in the Books table acts as the primary key and the attribute in the Publishers table acts as a foreign key.

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

The model is based on “Has a” design model. Each entity is associated to another using a “Has a” relationship. For example, An Author manages their twitter account. Hence, an Author “Has a” Twitter account.

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

Yes, there were design alternatives that we have presented. The trade offs can be between entity vs attribute and bimary vs ternary relationships. However, in the present model there cannot be any tradeoff between entity vs relationship.

Entity vs Attribute: We have created an Author entity but it is also an attribute in the Books table as we need more specifications about the author. Hence, have author as an entity.

Binary vs Ternary: We have created a Books table and it is associated with Publisher and Authors. This is identified as a ternary relationship. We have several other binary relationships for eg , Author and their twitter Accounts.

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

Data can be found on Goodreads, Twitter, Kaggle, Reddit.

**PART II – PHYSICAL MODEL**

**ENTITY RELATIONSHIP DIAGRAM**

A close up of a logo

Description automatically generated

Above is the ER Diagram of the model presented. The entities are represented in the rectangular boxes. Entity price is a weak entity and is shown in double line rectangular box as Price is decided by the Publisher.

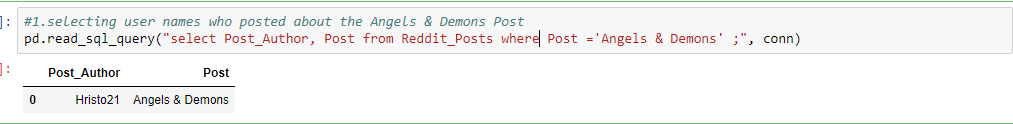
The attributes are represented in the circles and each entity has it’s own attributes. The primary keys are underlined inside their circles.

The diamond shapes represent the relationship between the entities as how each entity is connected with each other.

QUERIES

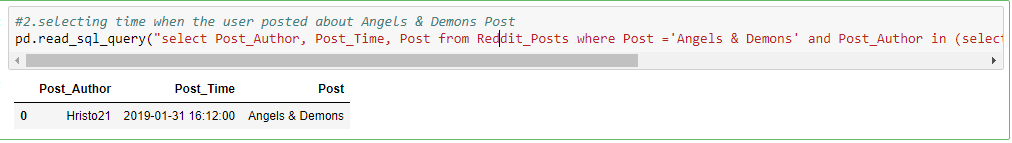
1. What user posted this (e.g. tweet, facebook post, IG post, etc.)?

select Post\_Author, Post from Reddit\_Posts where Post ='Angels & Demons' ;



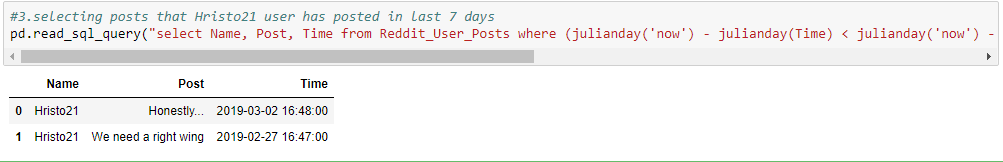
1. When did the user post this (e.g. tweet, facebook post, IG post, etc.)?

select Post\_Author, Post\_Time, Post from Reddit\_Posts where Post ='Angels & Demons' and Post\_Author in (select Post\_Author from Reddit\_Posts where Post ='Angels & Demons') ;



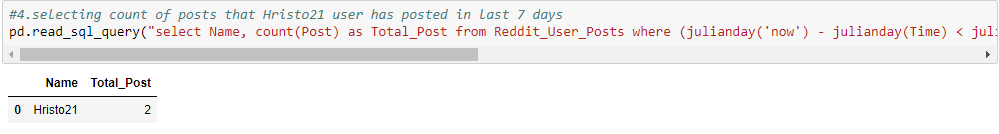
1. What posts has this user posted in the past 7 days?

select Name, Post, Time from Reddit\_User\_Posts where (julianday('now') - julianday(Time) < julianday('now') - (julianday('now')-7)) and Name in (select Post\_Author from Reddit\_Posts where Post ='Angels & Demons');



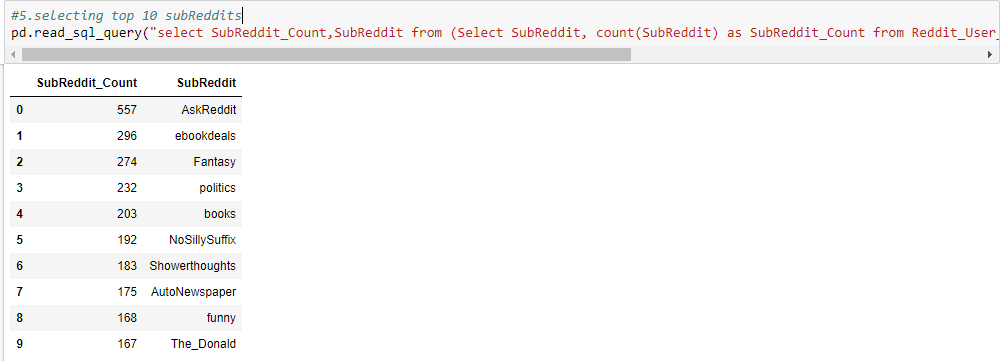
1. How many post has this user posted in the past 7 days?

select Name, count(Post) as Total\_Post from Reddit\_User\_Posts where (julianday('now') - julianday(Time) < julianday('now') - (julianday('now')-7)) and Name in (select Post\_Author from Reddit\_Posts where Post ='Angels & Demons') group by Name;



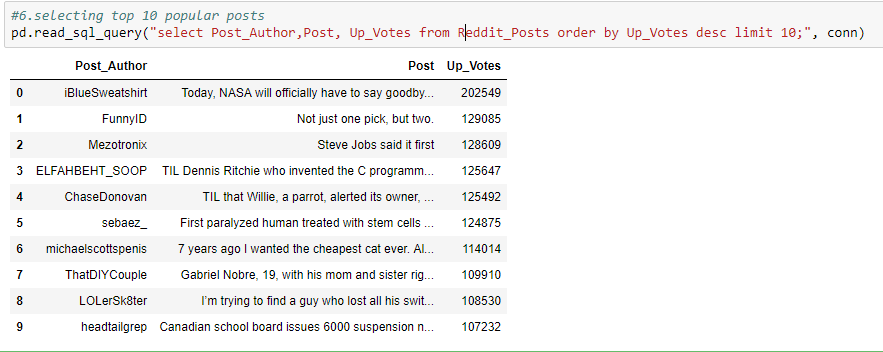
1. What keywords/ hashtags are popular?

select SubReddit\_Count,SubReddit from (Select SubReddit, count(SubReddit) as SubReddit\_Count from Reddit\_User\_Posts group by SubReddit) order by SubReddit\_Count desc limit 10;



1. What posts are popular?

select Post\_Author,Post, Up\_Votes from Reddit\_Posts order by Up\_Votes desc limit 10;



**CONCLUSION**

From this Assignment we were able to perform and learned the following:

1. Extract data from Twitter and Reddit APIs.
2. Identify Attributes and entities to create a database.
3. Create associations among several entities.
4. Create a conceptual database.
5. Improve a conceptual database to create a physical database.
6. Write SQL queries.

**CONTRIBUTIONS**  
  
Rashika Moza: 50%  
Akash Srivastava: 50%

**CITATIONS**

1. <https://github.com/nikbearbrown/INFO_6210>
2. <https://www.sqlite.org/datatype3.html>
3. <https://www.w3schools.com/sql/default.asp>
4. <https://praw.readthedocs.io/en/latest/getting_started/quick_start.html>
5. <http://docs.tweepy.org/en/v3.5.0/>
6. https://stackoverflow.com/questions/11131958/what-is-the-maximum-characters-for-the-nvarcharmax

**LICENSE**

Copyright 2019 Akash Srivastava, Rashika Moza

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2019 Akash Srivastava, Rashika Moza

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.