

|  |
| --- |
| Business Template  **Social media: logical daTA MODEL** |
|  |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 Business background 3](#_Toc62212631)

[1.2 Problems. Current Situation 3](#_Toc62212632)

[1.3 The benefits of implementing a database. Project Vision 3](#_Toc62212633)

[2 Model description 3](#_Toc62212634)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

# 

# Business Description

## Business background

The business here is an Instagram-like social media platform. It allows users to create profiles, share content through posts, follow other users and engage with content through likes, comments, and shares. The platform is designed to help people connect with each other, publish video/photo material and hopefully, discover new content that the users will like.

## Problems. Current Situation

The platform could be struggling with inaccurate content recommendations, because the insight gained from database is too messy and too diluted. Structurizing the data better could potentially not only improve the algorithm, but also fasten up the overall performance and maybe increase scalability.

## the Benefits of implementing a database. Project Vision

Organizing all the data (users, posts, interactions) in general can make it easier to access and find information. This will speed up how quickly the platform works, giving users a smoother experience when they use it. As said in previous point, with better organized data, we can also improve the content recommendations. This means that users will see posts and suggestions that match their interests more closely, which will keep them engaged and coming back for more.

As the platform grows, a good database design will help handle more users and more data without slowing down (scalability issue). It will also make sure the information stays accurate and secure. Finally, having all this structured data will help in analyzing how people are using the platform. This information can guide us in making better decisions about content and features and will serve as an advantage to similar businesses.

# Model description

## Definitions & Acronyms

* PK – primary key, not null
* FK - foreign key
* INT - integer type
* VARCHAR – string type
* TEXT – long string type
* NUMERIC – numeric (decimal) type
* GENERATED ALWAYS AS IDENTITY - auto-incrementing column
* NOT NULL – can’t have empty value
* 1:1 – one to one relationship
* 1:m – one to many relationship
* m:m – many to many relationship

## Logical Scheme

Obraz zawierający tekst, zrzut ekranu, diagram, design

Opis wygenerowany automatycznie

*Fig. SOCIAL MEDIA: LOGICAL DATA MODEL*

## Objects

* **User –** stores basic user information, is main table for other quantities

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| User | **user\_id** | Unique identifier for each user, PK | INT, GENERATED ALWAYS AS IDENTITY |
| **username** | Chosen username | VARCHAR(30), NOT NULL, UNIQUE |
| **password** | User’s password | VARCHAR(30), NOT NULL |
| **geolocation\_id** | References Geolocation.geolocation**\_id,** FK | INT, NOT NULL |

**Comments on table relationships:**

* 1:m – users can save multiple posts and publish multiple posts,
* 1:1 – each user has very special location, each special location is reserved by only one user
* M:m – users can have multiple followers and follow multiple people

**Example with data**

|  |  |  |  |
| --- | --- | --- | --- |
| **user\_id** | **username** | **password** | **geolocation\_id** |
| 15234 | DefinitelyNotBot | 123admin123 | 12322 |

* **Geolocation –** stores information about user’s location

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| GEOLOCATION | **geolocation\_id** | Unique identifier for each geolocation entry , PK | INT, GENERATED AS IDENTITY |
| **latitude** | Latitude coordinate of the location | NUMERIC, NOT NULL |
| **longitude** | Longitude coordinate of the location | NUMERIC, NOT NULL |
| **country\_name** | Name of the country corresponding to the location | VARCHAR(30) |

**Comments on table relationships:**

1:1 - Each user has it’s very own special location and each special location has only one user

**Example with data**

|  |  |  |  |
| --- | --- | --- | --- |
| **geolocation\_id** | **latitude** | **longitude** | **country\_name** |
| 12543123 | 37.7749 | -122.4194 | Poland |

* **Post -** contains information about posts made by users

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Post | **post\_id** | Unique identifier for each post, PK | INT, GENERATED ALWAYS AS IDENTITY |
| **user\_id** | References User.user\_id, FK | INT, NOT NULL |
| **content** | Content of the post | TEXT, NOT NULL |

**Comments on table relationships**

1:m – one post can be saved multiple times; it can have multiple comments, likes and shares

M:1 – each user can publish multiple posts

M:m – a post can have multiple hashtags and hashtags can be put into multiple posts

**Example with data**

|  |  |  |
| --- | --- | --- |
| **post\_id** | **user\_id** | **content** |
| 53123231 | 4123213 | "Almost finished my task!" |

* **Saved\_post -** connects users with the posts they have saved

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| **Saved\_post** | **saved\_post\_id** | Unique identifier for each saved post entry | INT, NOT NULL, GENERATED ALWAYS AS IDENTITY |
| **user\_id** | References User.user\_id, FK | INT, NOT NULL |
| **post\_id** | References Post.post\_id, FK | INT, NOT NULL |

**Comments on table relationships**

1:m - each user can save multiple posts, each saved post entry to a specific post (each post can be saved by multiple users, so in core a one-to-many relationship)

**Example with data**

|  |  |  |
| --- | --- | --- |
| **saved\_post\_id** | **user\_id** | **post\_id** |
| 6262211 | 6763452352 | 2546561 |

* **Post\_hashtag –** bridge for m:m relationship between post and hashtag tables

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Post\_hashtag\_bridge | **post\_id** | References Post.post\_id, FK | INT, NOT NULL |
| **hashtag\_id** | References Hashtag.hashtag\_id, FK | INT, NOT NULL |

**Comments on table relationships:**

M:m (bridge) – a post can have multiple hashtags and hashtags can be put into multiple posts

**Example with data**

|  |  |
| --- | --- |
| **post\_id** | **hashtag\_id** |
| 12543123 | 37749 |

* **Hashtag –** stores hashtags used in posts

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Hashtag | **hashtag\_text** | The hashtag text | VARCHAR(50), NOT NULL, UNIQUE |
| **hashtag\_id** | Unique identifier for each hashtag, PK | INT, GENERATED ALWAYS AS IDENTITY |

**Comments on table relationships:**

M:m – a post can have multiple hashtags and hashtags can be put into multiple posts

**Example with data**

|  |  |
| --- | --- |
| **hashtag\_id** | **hashtag\_text** |
| 1234242 | #travel |

* **Like –** stores information about the likes on posts by users

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Like | **like\_id** | Unique identifier for each like | INT, GENERATED AS IDENTITY |
| **post\_id** | References Post.post\_id, FK | INT, NOT NULL |

**Comments on table relationships:**

1:m - Each post can have multiple likes

**Example with data**

|  |  |
| --- | --- |
| **like\_id** | **post\_id** |
| 1234242 | 4324242 |

* **Comment -** stores comments made by users on posts

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Comment | **comment\_id** | Unique identifier for each comment | INT, NOT NULL, GENERATED ALWAYS AS IDENTITY |
| **comment\_text** | Content of the comment | TEXT, NOT NULL |
| **post\_id** | References Post.post\_id, FK | INT, NOT NULL |

**Comments on table relationships**

1:m – each post can have multiple comments

**Example with data**

|  |  |  |
| --- | --- | --- |
| **comment\_id** | **comment\_text** | **post\_id** |
| 6262211 | “Great photo!” | 2546561 |

* **Share –** stores information about posts shared by users

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Share | **share\_id** | Unique identifier for each share | INT, GENERATED AS IDENTITY |
| **post\_id** | References Post.post\_id, FK | INT, NOT NULL |

**Comments on table relationships:**

1:m - Each post can have multiple shares

**Example with data**

|  |  |
| --- | --- |
| **share\_id** | **post\_id** |
| 1234242dd | 4324242 |

* **Following\_user\_bridge –** bridge for m:m relationship between user and followed people

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Following\_user\_bridge | **following\_id** | References Following.following\_id, FK | INT, NOT NULL |
| **user\_id** | References User.user\_id, FK | INT, NOT NULL |

**Comments on table relationships:**

M:m (bridge) – a user can follow multiple people and be followed back by multiple people

**Example with data**

|  |  |
| --- | --- |
| **following\_id** | **user\_id** |
| 12543123 | 423424 |

* **Following –** indicates the users that a particular user is following

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Following | **following\_id** | Unique identifier for the following entry, PK | INT, GENERATED ALWAYS AS IDENTITY |

**Example with data**

|  |
| --- |
| **following\_id** |
| 12312312 |

* **Followers\_user\_bridge –** bridge for m:m relationship between user and people following the user

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Followers\_user\_bridge | **follower\_id** | References Followers.follower\_id, FK | INT, NOT NULL |
| **user\_id** | References User.user\_id, FK | INT, NOT NULL |

**Comments on table relationships:**

M:m (bridge) – a user can be followed multiple people and follow back multiple people

**Example with data**

|  |  |
| --- | --- |
| **follower\_id** | **user\_id** |
| 12543123 | 324234467 |

* **Followers –** indicates the followers of a particular user

|  |  |  |  |
| --- | --- | --- | --- |
| * Table Name | Field name | Field Description | Data Type |
| Followers | **follower\_id** | Unique identifier for the followers entry, PK | INT, GENERATED ALWAYS AS IDENTITY |

**Example with data**

|  |
| --- |
| **following\_id** |
| 1231864 |