

Introduction

In today's world where social media can become one's career, needs of content creators are growing. This project proposes the database of an integrated platform that connects Figma with popular social media networks, allowing users to design and schedule posts directly within the Figma interface. By simplifying the content creation process and ensuring compliance with platform-specific requirements, this solution enhances user efficiency and effectiveness. The project includes detailed entity definitions, relationships, and business rules to ensure data integrity and efficient management of user designs, posts, and platforms.

Part 1: Business Case

Social media content creator uses all kinds of tools to create content, one of the tools that they use is Figma. I am creating a business that aims to develop an integrated platform that allows users to design content in Figma and schedule it for posting on multiple social media platforms directly from within the Figma interface. The platform will support the scheduling of posts across various platforms like Facebook, Instagram, and LinkedIn, while adhering to platform-specific requirements like image sizes and formats.

Entities:

- User: Represents a user who creates designs and schedules posts.
- Design: Represents the designs created in Figma by users.
- Post: Contains the details of the scheduled posts.
- Platform: Represents social media platforms where the posts will be published (e.g., Facebook, Instagram).
- Scheduler: Handles the scheduling of posts to specific platforms.

Business Rules:

- A User can create multiple Designs in Figma.
- A Design can be used in multiple Posts, but each Post can have only one Design.
- A User can schedule multiple Posts to be published on different Platforms.
- Each Post must be associated with one Platform, but a Platform can have many Posts.
- A Scheduler must handle the timing and order of posts for each Platform.

- Posts must include attributes like scheduled date, time, and platform-specific constraints (image size, format).

Cardinality and Relationship Details:

- A User is mandatory and can create many Designs, but a Design belongs to only one User (1-to-Many, mandatory on both sides).
- A Design is mandatory for a Post, and a Post must be linked to exactly one Design (1-to-1, mandatory).
- A User is optional for Posts, but a Post must be created by one User (1-to-Many, optional on User side).
- A Post must be published on a Platform (1-to-Many, mandatory for both Post and Platform).
- The Scheduler must manage multiple Posts but is mandatory for every Platform (1-to-Many, mandatory for both Scheduler and Platform).

Part 2: Database Design

Entities and Attributes:

User:

- UserID (Primary Key) – Unique identifier for each user.
- UserName – The name of the user.
- Email – The email of the user for communication.
- Password – User's encrypted password.

Design:

- DesignID (Primary Key) – Unique identifier for each design.
- UserID (Foreign Key) – Refers to the user who created the design.
- DesignName – Name of the design.
- CreatedDate – Date when the design was created.

Post:

- PostID (Primary Key) – Unique identifier for each post.
- DesignID (Foreign Key) – Refers to the design being used in the post.
- UserID (Foreign Key) – Refers to the user who created the post.
- PlatformID (Foreign Key) – Refers to the platform on which the post will be published.
- PostContent – The content of the post (e.g., caption).
- ScheduledDateTime – Date and time when the post is scheduled to be published.

Platform:

- PlatformID (Primary Key) – Unique identifier for each platform.
- PlatformName – Name of the platform (e.g., Facebook, Instagram).
- ImageSizeConstraint – The required image size for the platform.

Scheduler:

- SchedulerID (Primary Key) – Unique identifier for each scheduler instance.
- PostID (Foreign Key) – Refers to the post being scheduled.
- ScheduledDateTime – The date and time when the post is scheduled.

Design Decisions and Justifications:

Primary Keys (PKs):

- Each table has a unique Primary Key to uniquely identify each record in the table.
- I chose surrogate keys for simplicity and consistency in indexing.

Foreign Keys (FKs):

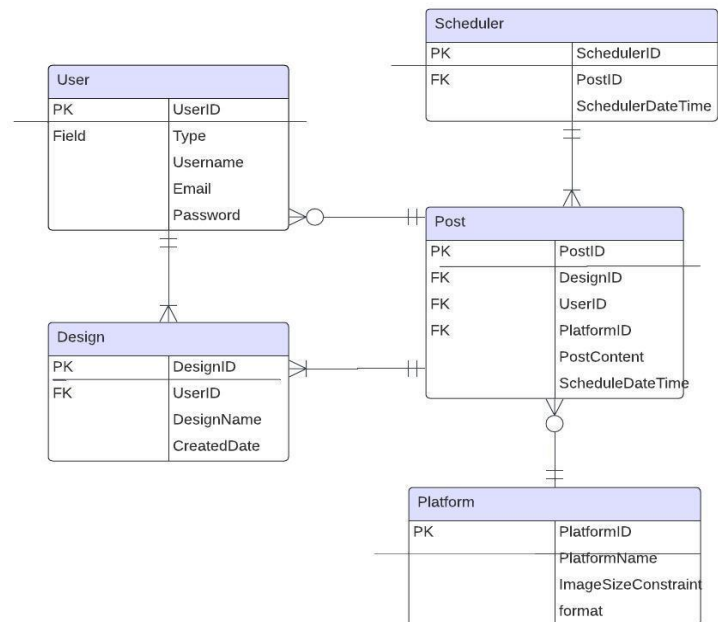
- The foreign Keys are used to establish relationships between the tables.
- For example, the Post table has foreign keys linking to the User, Design, and Platform tables to indicate the relationships between posts, users, designs, and platforms.

Relationships:

- User-Design: A User can create many Designs, but each Design is created by only one User (1-to-Many).
- Design-Post: A Post uses one Design, and a Design can be used in multiple Posts (1-to-Many).
- User-Post: A User can create multiple Posts, but each Post is created by only one User (1-to-Many).
- Post-Platform: A Post is published on one Platform, and a Platform can host multiple Posts (1-to-Many).
- Post-Scheduler: A Scheduler is responsible for scheduling a Post to a platform at a specific time (1-to-1, since each post is scheduled at a specific time).

Constraints:

- NOT NULL constraints on mandatory fields such as UserName, DesignName, and PostContent ensure that these fields are always populated.
- Foreign Key Constraints are implemented to ensure data integrity and proper linkage between tables.



Part 3: DB Implementation

```

CREATE DATABASE SocialMediaScheduler;
USE SocialMediaScheduler;

```

```

CREATE TABLE User (
    UserID INT PRIMARY KEY AUTO_INCREMENT,
    UserName VARCHAR(100) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    Password VARCHAR(255) NOT NULL
);

```

```

CREATE TABLE Design (
    DesignID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT NOT NULL,
    DesignName VARCHAR(100) NOT NULL,
    CreatedDate DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE
);

```

```

CREATE TABLE Platform (
    PlatformID INT PRIMARY KEY AUTO_INCREMENT,
    PlatformName VARCHAR(50) NOT NULL,

```

```

        ImageSizeConstraint VARCHAR(50) NOT NULL
    );

CREATE TABLE Post (
    PostID INT PRIMARY KEY AUTO_INCREMENT,
    DesignID INT NOT NULL,
    UserID INT NOT NULL,
    PlatformID INT NOT NULL,
    PostContent TEXT NOT NULL,
    ScheduledDateTime DATETIME NOT NULL,
    FOREIGN KEY (DesignID) REFERENCES Design(DesignID) ON DELETE
    CASCADE,
    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE SET NULL,
    FOREIGN KEY (PlatformID) REFERENCES Platform(PlatformID) ON DELETE
    CASCADE
);

CREATE TABLE Scheduler (
    SchedulerID INT PRIMARY KEY AUTO_INCREMENT,
    PostID INT NOT NULL,
    ScheduledDateTime DATETIME NOT NULL,
    FOREIGN KEY (PostID) REFERENCES Post(PostID) ON DELETE CASCADE
);

```

Part 4: Data Entry

```

INSERT INTO User (UserName, Email, Password)
VALUES ('JohnDoe', 'john@example.com', 'hashedpassword1'),
       ('JaneSmith', 'jane@example.com', 'hashedpassword2');

INSERT INTO Platform (PlatformName, ImageSizeConstraint)
VALUES ('Facebook', '1080x1080'),
       ('Instagram', '1080x1080'),
       ('LinkedIn', '1200x627');

INSERT INTO Design (UserID, DesignName, CreatedDate)
VALUES (1, 'Summer Campaign Design', '2024-10-01'),
       (2, 'Winter Campaign Design', '2024-10-05');

INSERT INTO Post (DesignID, UserID, PlatformID, PostContent,
                  ScheduledDateTime)
VALUES (1, 1, 1, 'Check out our summer sale!', '2024-10-25 10:00:00'),
       (2, 2, 2, 'Winter deals are here!', '2024-11-01 14:00:00');

INSERT INTO Scheduler (PostID, ScheduledDateTime)

```

```
VALUES (1, '2024-10-25 10:00:00'),  
       (2, '2024-11-01 14:00:00');
```

Part 5: Query

```
SELECT Post.PostContent, Design.DesignName, Platform.PlatformName,  
       Post.ScheduledDateTime  
FROM Post  
JOIN Design ON Post.DesignID = Design.DesignID  
JOIN Platform ON Post.PlatformID = Platform.PlatformID  
WHERE Platform.PlatformName = 'Instagram';
```

```
SELECT Post.PostContent, Design.DesignName, Platform.PlatformName,  
       Post.ScheduledDateTime  
FROM Post  
JOIN Design ON Post.DesignID = Design.DesignID  
JOIN Platform ON Post.PlatformID = Platform.PlatformID  
JOIN User ON Post.UserID = User.UserID  
WHERE User.UserName = 'JohnDoe';
```

PHP Script

```
<?php  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "SocialMediaScheduler";  
  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
  
$sql = "SELECT Post.PostContent, Design.DesignName, Platform.PlatformName,  
          Post.ScheduleDateTime  
        FROM Post  
        JOIN Design ON Post.DesignID = Design.DesignID
```

```
JOIN Platform ON Post.PlatformID = Platform.PlatformID
JOIN User ON Post.UserID = User.UserID
WHERE User.UserName = 'JohnDoe';
```

```
$result = $conn->query($sql);
```

```
if ($result->num_rows > 0) {
```

```
    while($row = $result->fetch_assoc()) {
        echo "Post Content: " . $row["PostContent"]. " - Design Name: " .
        $row["DesignName"]. " - Platform: " . $row["PlatformName"]. " - Scheduled Date/Time:
        " . $row["ScheduleDateTime"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```

Conclusion

The database design effectively captures the essential components of the integrated platform, ensuring that relationships between users, designs, posts, and social media platforms are well-defined. By enforcing key constraints and utilizing foreign keys, the database maintains data integrity while supporting scalability for future enhancements. Overall, this structured approach provides a solid foundation for the platform, enabling efficient content management and scheduling for users.