INSTAGRAM DATABASE CLONE.

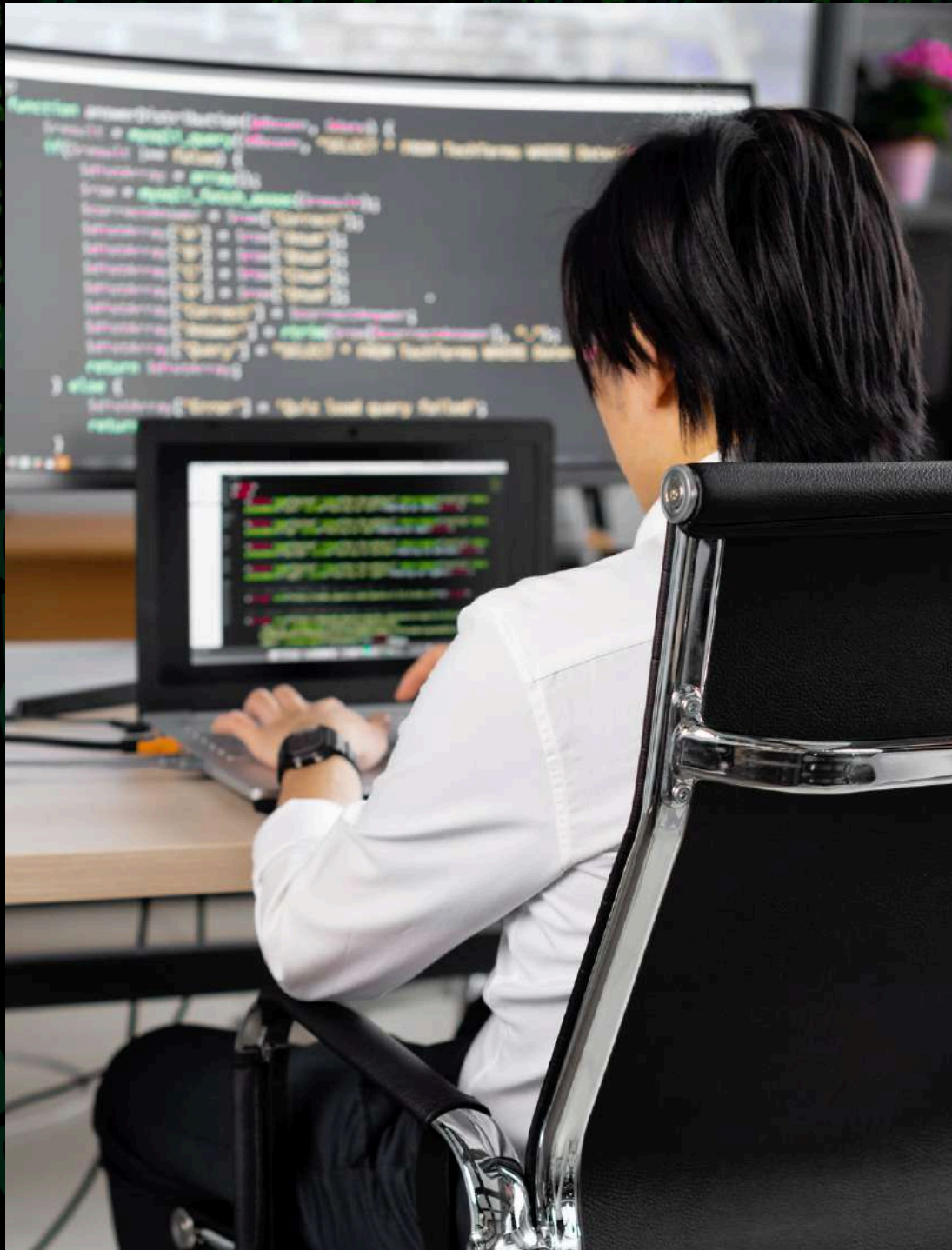## PRESENTATION

PROJECT OVERVIEW

- Briefly explain the goal of the project (e.g., creating a database schema for a social media application).
- Mention key features, such as users posting photos, adding comments, liking photos, and following other users.
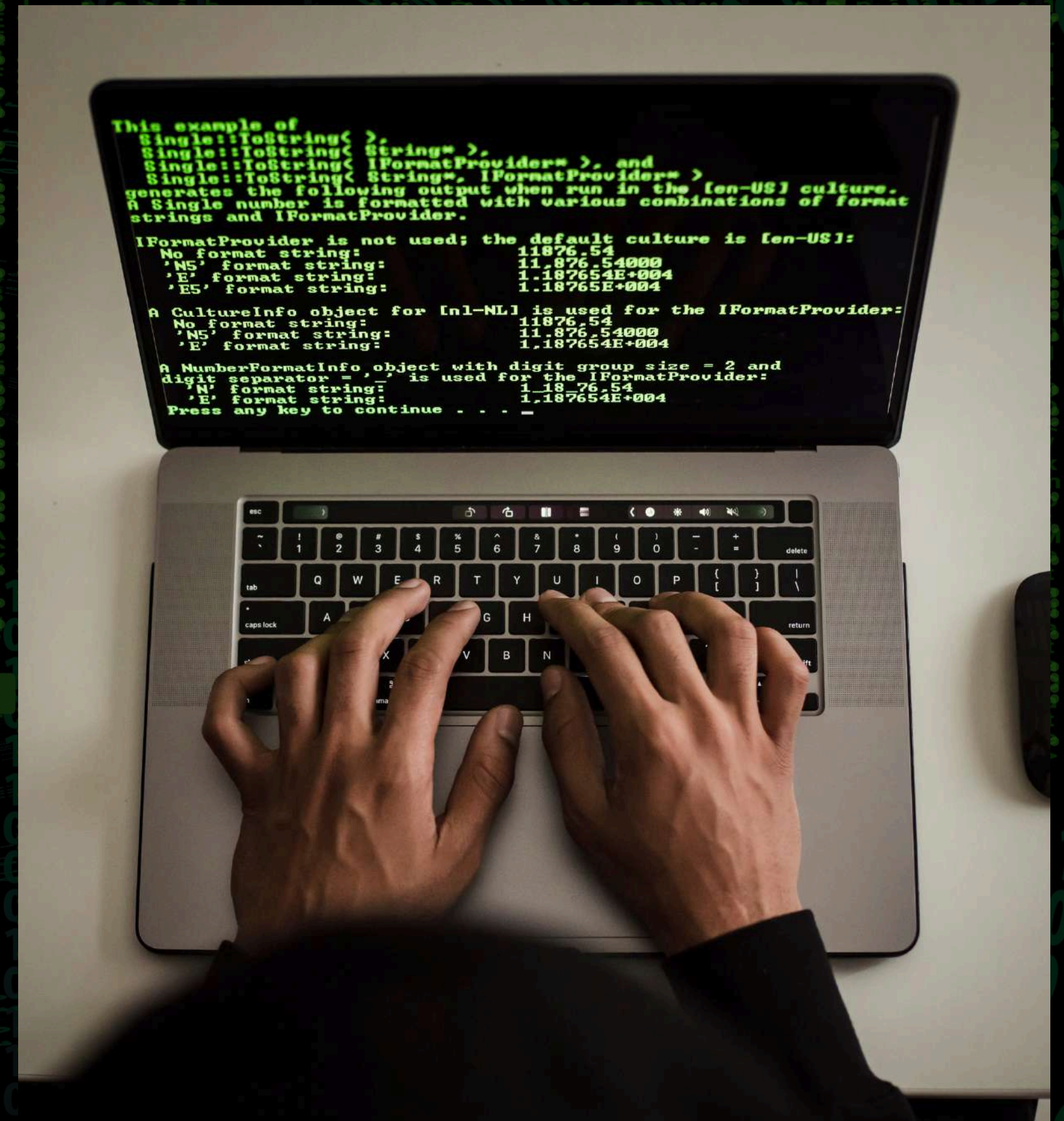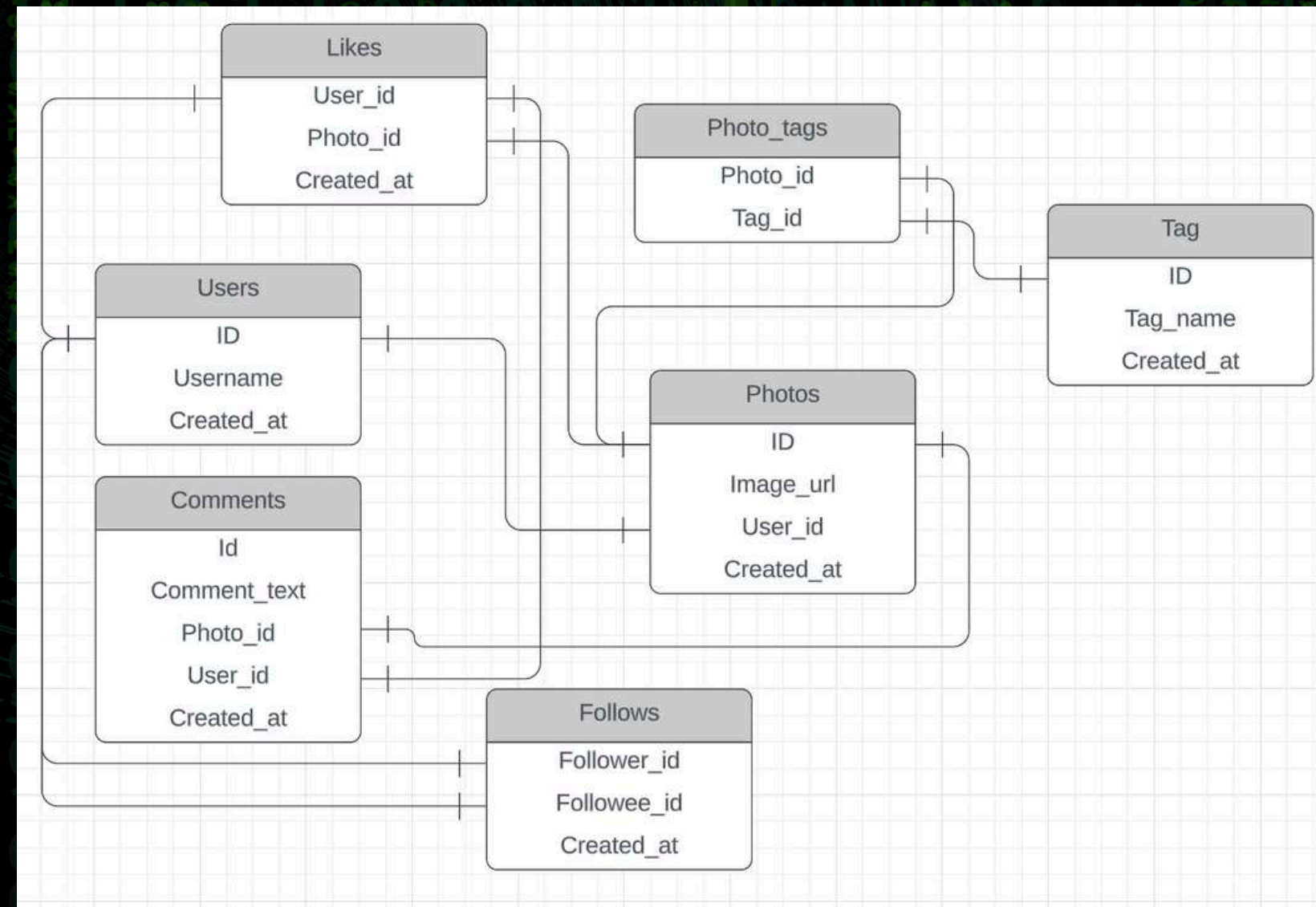
DATABASE DESIGN

DATABASE STRUCTURE

- Discuss the importance of relational database design for maintaining relationships between different entities.
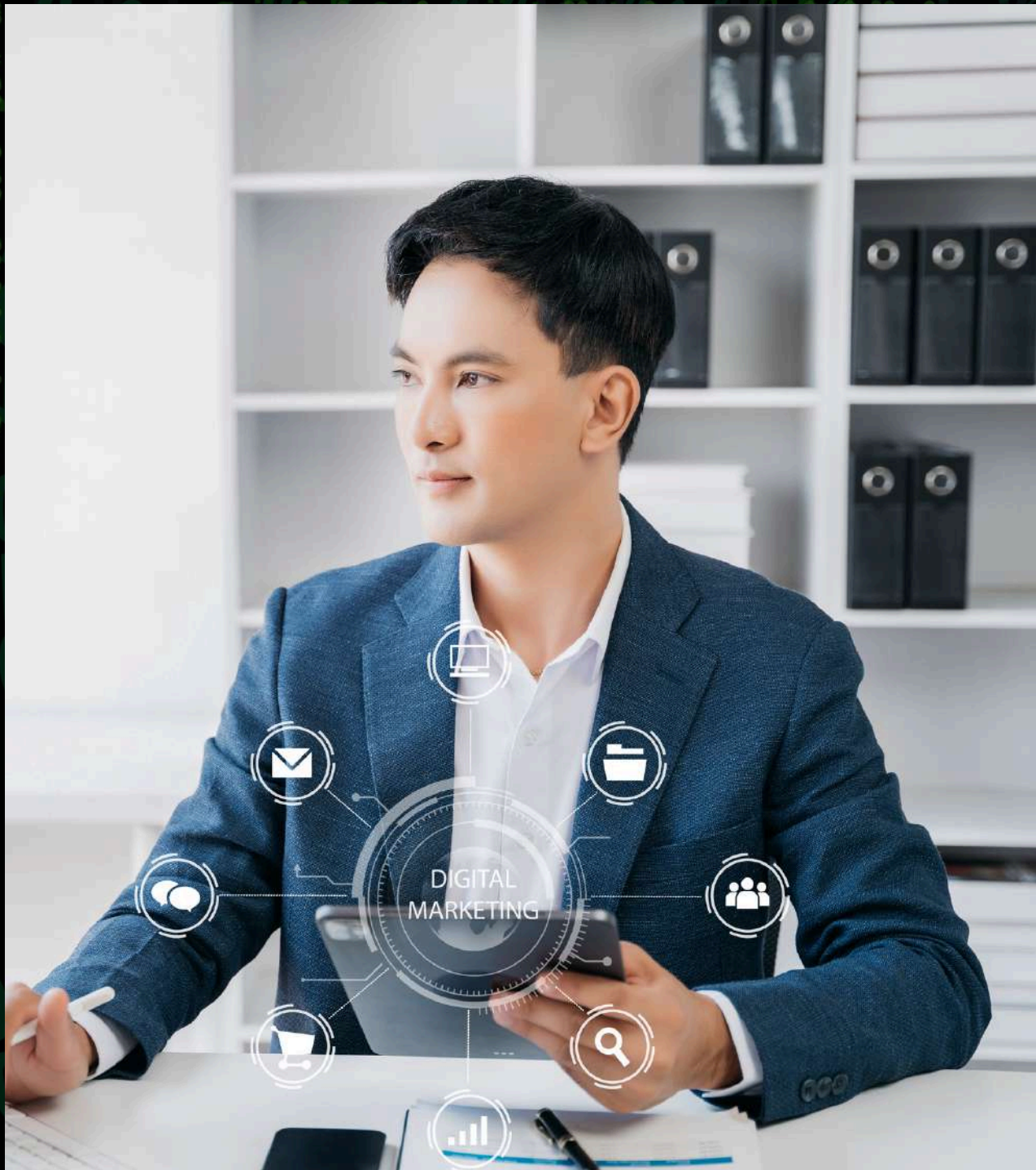- Mention normalization and why it's important for reducing redundancy.

USERS TABLE

»

Content

- Columns: id, username, created_at
- Description: Stores user information, including unique usernames and creation timestamps.

```sql
CREATE TABLE users (
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) UNIQUE NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);
```
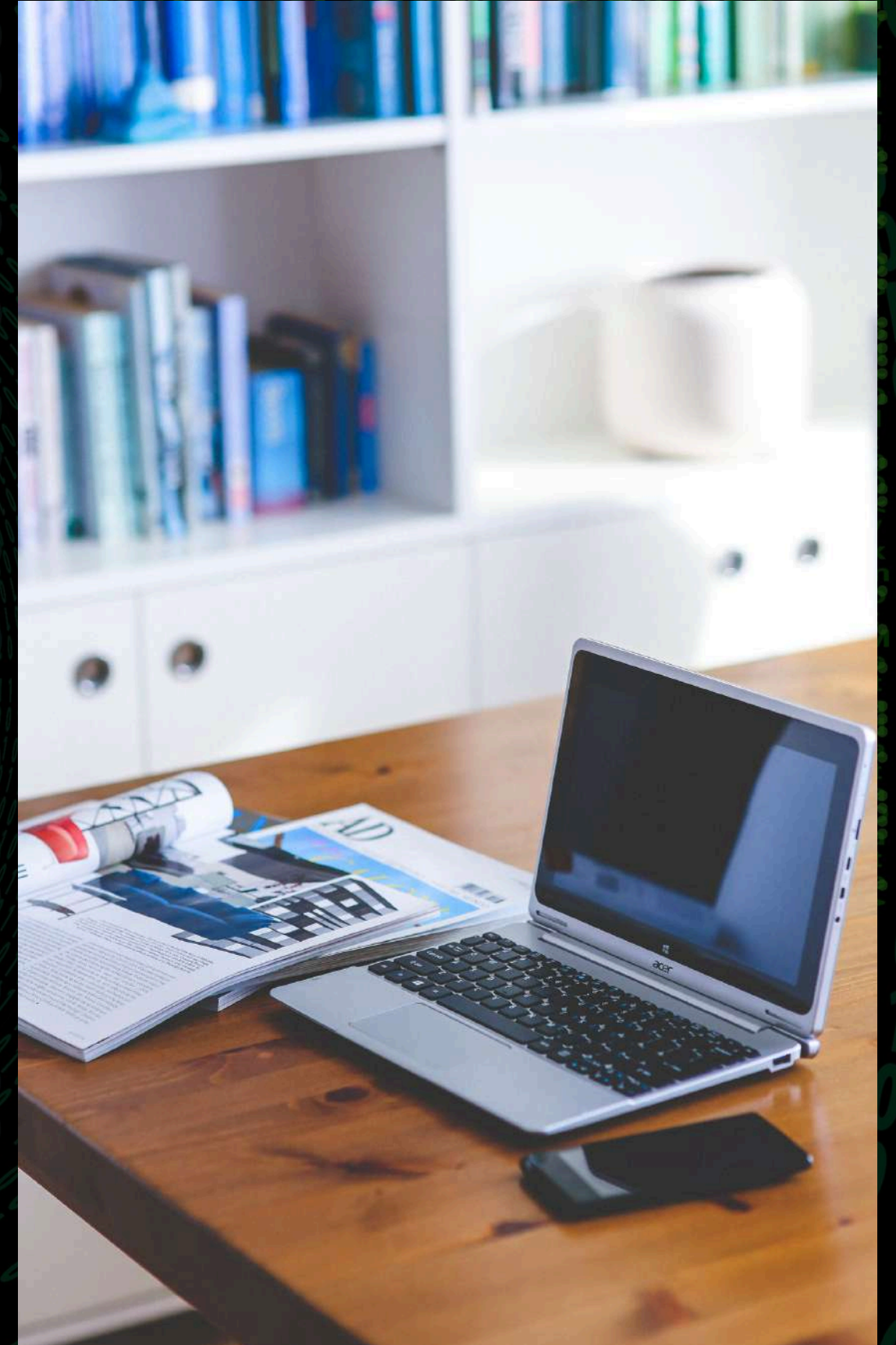
PHOTOS TABLE

Content

- Columns: id, image_url, user_id, created_at
- Description: Stores photos uploaded by users and the relationship with the users table via user_id.

```sql
CREATE TABLE photos (
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(255) NOT NULL,
    user_id INTEGER NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id)
);
```

COMMENTS TABLE

CONTENTS

- Columns: id, comment_text, photo_id, user_id, created_at
- Description: Stores comments made by users on photos, with foreign key references to both users and photos.

```sql
CREATE TABLE comments (
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
    comment_text VARCHAR(255) NOT NULL,
    photo_id INTEGER NOT NULL,
    user_id INTEGER NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(photo_id) REFERENCES photos(id),
    FOREIGN KEY(user_id) REFERENCES users(id)
);
```
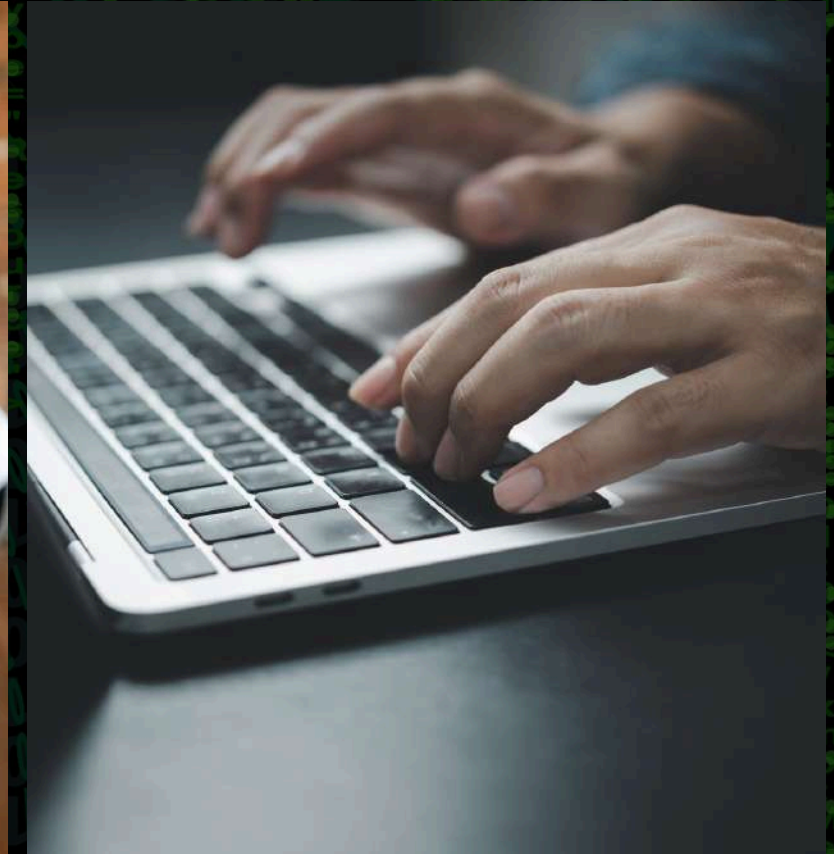
LIKES TABLE

Content

- **Columns: user_id, photo_id, created_at**
- **Description: Tracks which users liked which photos. Primary key is a composite of user_id and photo_id.**

```sql
CREATE TABLE likes (
  user_id INTEGER NOT NULL,
  photo_id INTEGER NOT NULL,
  created_at TIMESTAMP DEFAULT NOW(),
  FOREIGN KEY(user_id) REFERENCES users(id),
  FOREIGN KEY(photo_id) REFERENCES photos(id),
  PRIMARY KEY(user_id, photo_id)
);
```
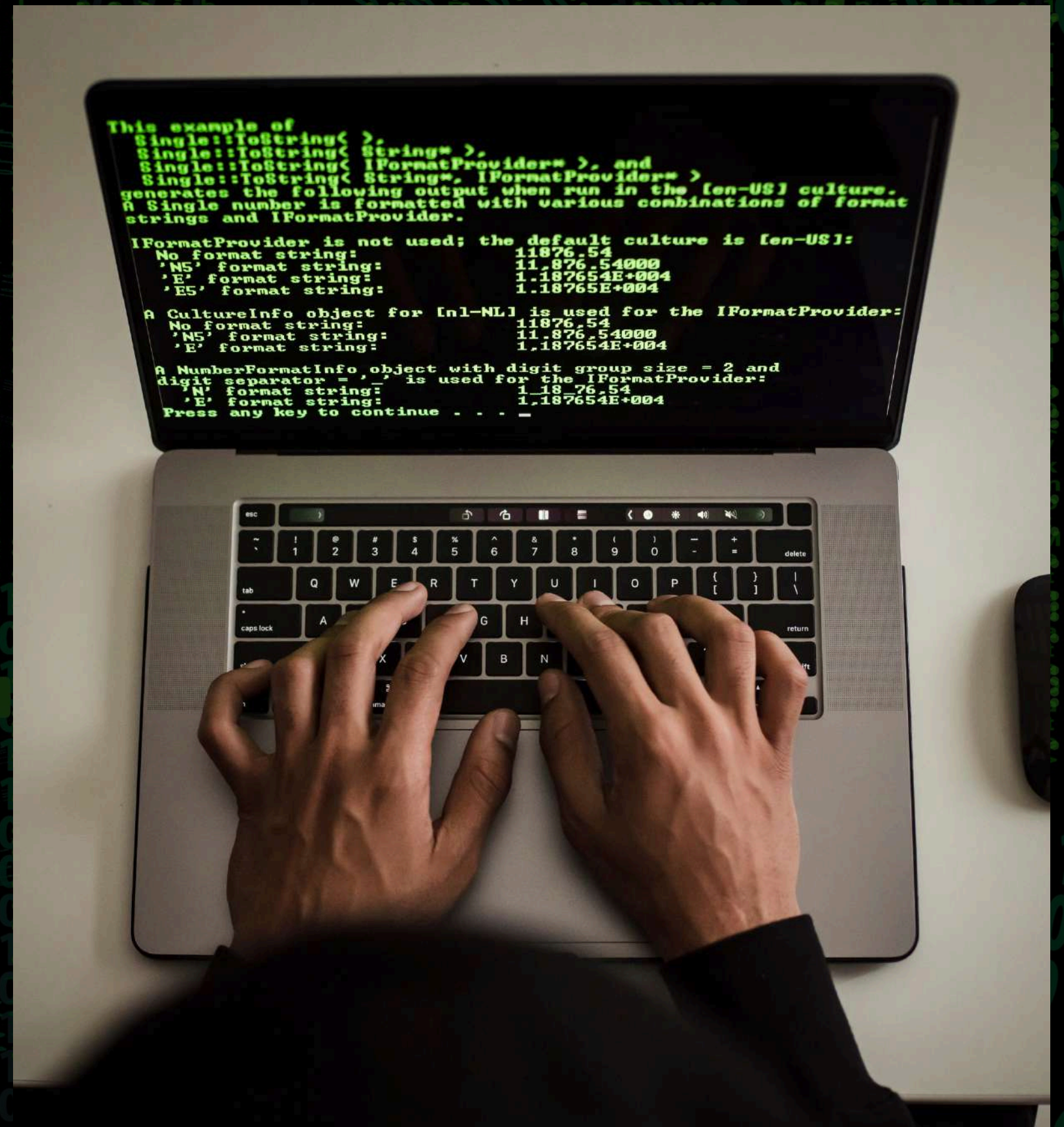
FOLLOWS TABLE

CONTENT
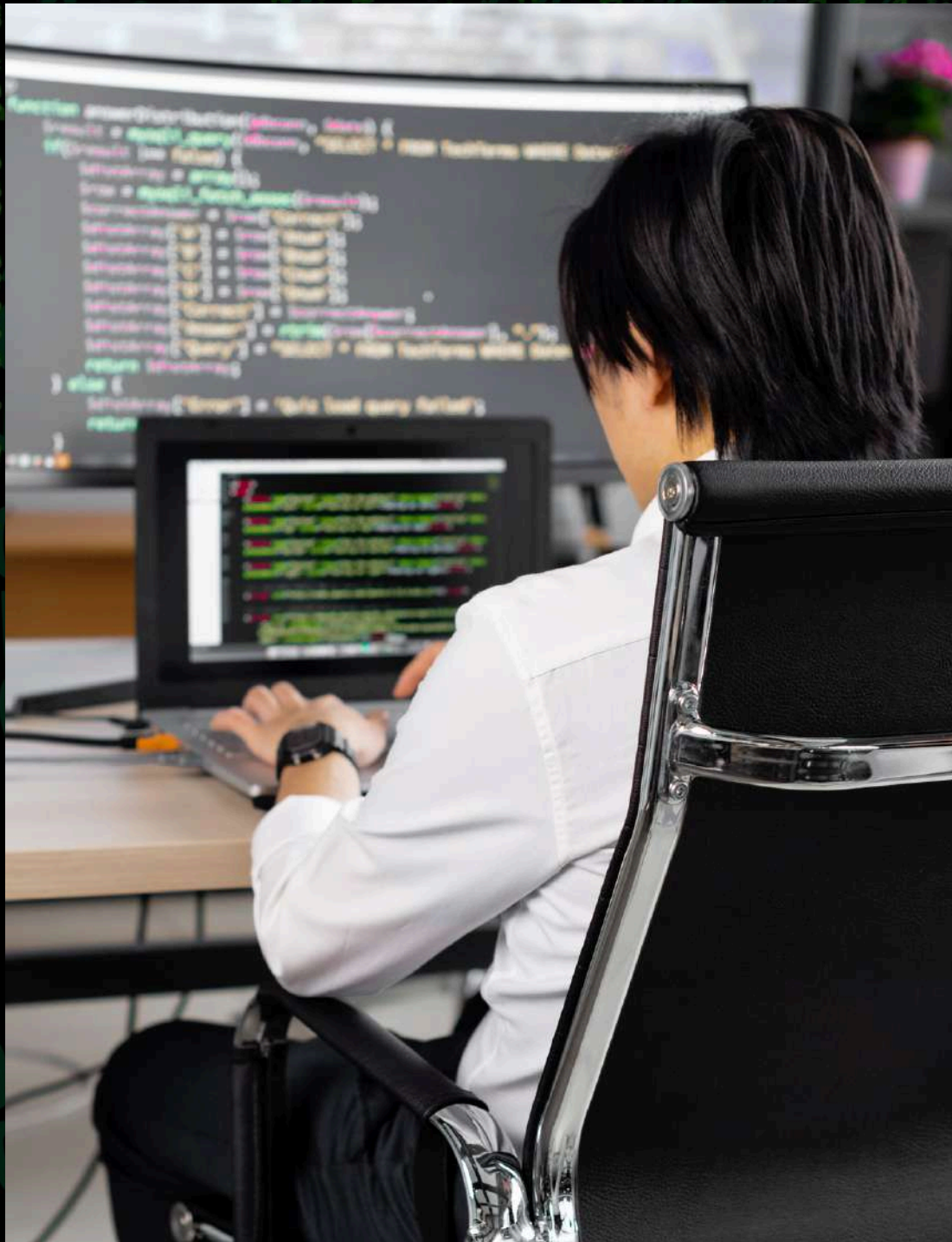
- Columns: follower_id, followee_id, created_at
- Description: Tracks follow relationships between users, with foreign keys referencing the users table.

```sql
CREATE TABLE follows (
    follower_id INTEGER NOT NULL,
    followee_id INTEGER NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(follower_id) REFERENCES users(id),
    FOREIGN KEY(followee_id) REFERENCES users(id),
    PRIMARY KEY(follower_id, followee_id)
);
```

TAGS AND PHOTO TAGS TABLES

»

Content

- **Tags Table:**
  - **Columns: id, tag_name, created_at**
  - **Stores the names of tags applied to photos.**
- **Photo Tags Table:**
  - **Columns: photo_id, tag_id**
  - **Links photos with tags via foreign keys.**

```sql
CREATE TABLE tags (
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
    tag_name VARCHAR(255) UNIQUE,
    created_at TIMESTAMP DEFAULT NOW()
);


CREATE TABLE photo_tags (
    photo_id INTEGER NOT NULL,
    tag_id INTEGER NOT NULL,
    FOREIGN KEY(photo_id) REFERENCES photos(id),
    FOREIGN KEY(tag_id) REFERENCES tags(id),
    PRIMARY KEY(photo_id, tag_id)
);
```

CONCLUSION

THE IG-CLONE SUCCESSFULLY MODELS THE CORE FEATURES OF A SOCIAL MEDIA APPLICATION THROUGH AN EFFICIENT RELATIONAL DATABASE SCHEMA. KEY TABLES SUCH AS USERS, PHOTOS, COMMENTS, LIKES, FOLLOWS, TAGS, AND PHOTO TAGS WORK TOGETHER TO HANDLE VARIOUS INTERACTIONS BETWEEN USERS AND CONTENT.

HIGHLIGHTS:

- DATA INTEGRITY: FOREIGN KEYS MAINTAIN RELATIONSHIPS BETWEEN USERS, PHOTOS, COMMENTS, LIKES, AND TAGS.
- NORMALIZATION: REDUCES REDUNDANCY, ENSURING AN EFFICIENT DATABASE STRUCTURE.
- SCALABILITY: THE SCHEMA IS DESIGNED TO SUPPORT FUTURE ENHANCEMENTS LIKE ADDITIONAL FEATURES OR LARGER DATASETS.

THIS DATABASE FRAMEWORK PROVIDES A SOLID FOUNDATION FOR BUILDING SCALABLE AND INTERACTIVE SOCIAL MEDIA PLATFORMS.