Prompt 1

A screenshot of a computer

Description automatically generated

Prompt 2 - Create Tables

CREATE TABLE users (

userid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

username VARCHAR(20),

address VARCHAR(255),

city VARCHAR(100),

state CHAR(2),

zip INT

);

CREATE TABLE locations (

itemid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

type INT,

description VARCHAR(255),

lng REAL,

lat REAL

);

CREATE TABLE photographs (

photoid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

locationid INT,

userid INT,

FOREIGN KEY (locationid) REFERENCES locations(itemid),

FOREIGN KEY (userid) REFERENCES users(userid)

);

A screenshot of a computer

AI-generated content may be incorrect.

Prompt 3 - Alter Tables

ALTER TABLE locations MODIFY type INT NOT NULL;

ALTER TABLE locations MODIFY description VARCHAR(255) NOT NULL;

ALTER TABLE locations MODIFY lng REAL NOT NULL;

ALTER TABLE locations MODIFY lat REAL NOT NULL;

ALTER TABLE users MODIFY name VARCHAR(255) NOT NULL;

ALTER TABLE users MODIFY username VARCHAR(20) NOT NULL;

ALTER TABLE photographs MODIFY photoid INT NOT NULL;

ALTER TABLE photographs MODIFY locationid INT NOT NULL;

Prompt 4

CREATE UNIQUE INDEX idx\_userid ON users (userid);

CREATE UNIQUE INDEX idx\_locationid ON photographs (locationid);

A screenshot of a computer

AI-generated content may be incorrect.

Prompt 5

A screenshot of a computer

AI-generated content may be incorrect.

Prompt 6

A screenshot of a computer

AI-generated content may be incorrect.

Prompt 7

ALTER TABLE photographs ADD COLUMN userid INT AFTER locationid;

A screenshot of a computer

AI-generated content may be incorrect.

Prompt 8

A screenshot of a computer

AI-generated content may be incorrect.

Now, to recap, the photographs table has the userid column added to it. Because the userid column was disconnected from the users table, there was no way of knowing whether each userid in the photographs table was a valid user. That's when invalid data may be there where users' userid values in the photographs table are not present in the users' table, breaking referential integrity.

To fix that, I enforced a foreign key in the photographs table. Photographs.userid is now a foreign key to the users.id column. With this, it ensures that no other userid value can ever be inserted within the photographs table. Furthermore, I enabled what is known as ON DELETE CASCADE, meaning that if a user is removed from the users table, all photographs tied to that user will also be deleted, which ensures data consistency and prevents orphaned records.

This solution ensures that the data is well-formed, which serves to keep the database consistent.

Prompt 9

INSERT INTO locations (type, description, lng, lat)

VALUES

(1, 'Independence Hall', 794.35, 651.43),

(2, '6709 Wonder Street', 323.41, 412.22),

(1, 'Sunrise', 221.45, 132.43),

(2, '356 A Street', 123.32, 222.43),

(1, 'Mountains', 34.12, 87.99),

(2, '900 Star Street', 1071.9, 206.45),

(1, 'Moonrise', 816.2, 111.2),

(2, '183714 N North Street', 176.11, 11.176);

INSERT INTO photographs (photoid, locationid, userid)

VALUES

(1, 1, 1),

(2, 2, 1),

(3, 3, 3),

(4, 4, 4);

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Prompt 10

A screenshot of a computer

AI-generated content may be incorrect.

Prompt 11

A screenshot of a computer

AI-generated content may be incorrect.

SELECT name FROM users, photographs WHERE users.userid = photographs.userid;

Prompt 12

A screenshot of a computer

AI-generated content may be incorrect.

SELECT DISTINCT name FROM users, photographs WHERE users.userid = photographs.userid;