

<Camping Registry>

Final Project CS195

Prepared by: Brendan Renaud & Quinn Davis

Date: 5/2/2025

Contents

1. Project Definition, Scope Case	4
1.1. Problem statement	4
1.2. Users.....	4
1.3. Data Requirement	4
1.4. Constraints.....	4
1.5. Real World Requirement.....	5
1.6. Information Requirement.....	5
1.7. Competitive Advantage	5
2. Conceptional Design	5
2.1. Entity Relationship Diagram.....	6
2.2. Normalization.....	6
3. Logical Design	7
3.1. Single Query	7
3.2. Multi-Query	8
3.3. Stored Procedures / Functions	10
3.4. Miscellaneous	10
4. Design Overview.....	11

Related Documents

<insert name and description of the scripts that are provided for the project>

Document Title	Quick Description
5_single_table_queries.sql	Our 5 single table queries
5_multi_table_queries.sql	Our 5 multi table queries
Area_inserts.sql	The inserts for area table
Photo_inserts.sql	The inserts for photo table
Rating_inserts.sql	The inserts for rating table
Site_inserts.sql	The inserts for site table
Comment_inserts.sql	The inserts for comment table
Rule_inserts.sql	The inserts for rule table
User_inserts.sql	The inserts for user table
ERD.pdf	The erd for our database
Fetch_liked_user_sites_PROC.sql	The stored procedure for retrieving liked user sites
Insert_saved_user_sites_PROC.sql	The stored procedure for inserting a saved user sites

1. Project Definition, Scope Case

1.1. Problem statement

Campsites are often remote and lack useful information about the type of site, whether it's free or paid, or how easy it is to access. We have created a database that stores all useful information about campsites so that people looking to camp have everything they need to be prepared.

1.2. Users

Our target users are outdoor enthusiasts and campers who often struggle to find detailed and reliable information about campsites when planning their trips.

1.3. Data Requirement

We will be storing various types of data, including:

- Numerical data (integers) for ratings and fees
- Textual data (varchars and strings) for names, descriptions, and comments
- Dates for tracking timestamps
- Boolean values for features like showers or electrical hookups
- Geographic data (latitude and longitude) in the format of decimals for mapping site location

1.4. Constraints

Some data will have constraints to ensure accuracy and consistency. For example:

- Phone numbers must be 10 digits
 - Zip codes must be 5 digits
 - Ratings must be between 0 and 10
 - Emails must be unique for each user
- These constraints help keep the data clean and prevent incorrect entries.

1.5. Real World Requirement

The database represents real-world applications like:

- A user registering an account
- Users liking or reviewing a campsite
- Users submitting photos or comments
- Admins adding new campsites and rules
- Campers searching for campsites based on features or location

1.6. Information Requirement

We expect to retrieve information such as:

- A list of top-rated campsites
- Campsites near/in a certain city or zip code
- Sites with specific facilities (e.g., restrooms or showers)
- User-submitted ratings, comments, and photos for a site
- Reports showing most liked or reviewed sites

1.7. Competitive Advantage

This project gives a competitive advantage by offering a centralized, easy-to use system where users can view and share detailed, real time information about campsites. Unlike generic review platforms, this system is built specifically for outdoor camping and includes specific features that campers care about most.

Conceptional Design

The ERD consists of the following entities:

- USER: Stores user account details
- SITE: Stores information about each campsite
- AREA: Stores location data for broader regions
- RATING: Stores user ratings on different aspects of a site
- COMMENT: Stores user comments
- PHOTO: Stores image file paths
- RULE and SITERULE: Manage campsite-specific rules
- LIKED_USER_SITE: Tracks which users liked which sites

1.8. Entity Relationship Diagram

The following information is the high-level model and design for the scope of this project.



1.9. Normalization

The following information is the high-level construction and optimization of the database for the scope of this project. This can be visual or an explanation that should be detailed as to why you believe this is in 1NF and 2NF

The design of our database meets the requirements for both 1st and 2nd normal form. Our database is in 1NF because all of our entities contain single valued attributes, ex: comments and photos is placed in its own table because there can be multiple values for one site. We have no partial dependencies in any of our entities, meaning we only have attributes that directly relate to the primary key. We also satisfy 2NF because we have created all necessary intermediate tables to logically separate data without.

Logical Design

<The following section represents the queries that the business will use for functional purposes>

1.10. Single Query

Query #1: Retrieve comments for a specific campsite

```
SELECT comment, timestamp FROM comment WHERE site_ID = 1;
```

Purpose: This query fetches all comments and their timestamps associated with a particular campsite (in this case, site #1), allowing users to view feedback about that specific location.

Query #2: Retrieve photo filepaths for a specific campsite

```
SELECT filepath FROM photo WHERE site_ID = 1;
```

Purpose: This query retrieves the file locations of all photos related to a specific campsite (site #1), enabling the system to display images of that campsite to users.

Query #3: Retrieve area details for a specific campsite

```
SELECT state, county, street_address, zipcode FROM area WHERE name = "Hartman Rocks";
```

Purpose: This query gathers location information (state, county, street address, zipcode) for a camping area based on its name

Query #4: Find users located in Gunnison

```
SELECT first_name, last_name FROM user WHERE city = "gunnison";
```

Purpose: This query identifies all users who live in Gunnison by retrieving their names, which could be useful for location-based features or analytics.

Query #5: Find camping areas in Gunnison

```
SELECT area_ID FROM area WHERE city = "gunnison";
```

Purpose: This query locates all camping areas located in Gunnison by their IDs, helping users find local camping options. The id is used to load additional information about the area to display on a route.

1.11. Multi-Query

Query #1: Retrieve all the site IDs for campsites that are in the city of gunnison

```
SELECT s.site_ID FROM site as s JOIN area as a ON s.area_ID = a.area_ID  
WHERE a.city LIKE "%gunnison%" LIMIT 5;
```

Purpose: This query finds campsites located in Gunnison by joining the site and area tables, then filtering for areas with "gunnison" in the city name. It returns up to 5 site IDs to be used to load additional information from the web app.

Query #2: Retrieve all of the site IDs that belong to hartman rocks camping area

```
SELECT s.site_ID FROM site as s JOIN area as a ON s.site_ID = a.site_ID  
WHERE a.name = "Hartman Rocks";
```

Purpose: This query identifies all campsites within the Hartman Rocks camping area by joining the site and area tables and filtering for the specific area name.

Query #3: Find the top 5 highest-rated camping sites based on average overall rating

```
SELECT s.site_ID, AVG(r.cleanliness + r.accessibility + r.quietness + r.activities  
+ r.amenities + r.cost) as overall_score FROM site as s JOIN rating as r ON  
s.site_ID = r.site_ID GROUP BY s.site_ID ORDER BY overall_score DESC  
LIMIT 5;
```

Purpose: This query calculates an overall score for each campsite by averaging all rating categories, then returns the top 5 highest-rated sites.

Query #4: Find the top 5 rated sites based on average rating for a single category

Camp Site Registry

Final Project: Renaud & Davis

```
SELECT s.site_ID, AVG(r.cleanliness) as cleanliness_score FROM site as s
JOIN rating as r ON s.site_ID = r.site_ID GROUP BY s.site_ID ORDER BY
cleanliness_score DESC LIMIT 5;
```

Purpose: This query finds the top 5 campsites based solely on their cleanliness ratings, allowing users to find sites that excel in this specific category.

Query #5: Find the average overall rating for a site

```
SELECT AVG(r.cleanliness + r.accessibility + r.quietness + r.activities +
r.amenities + r.cost)/6 as avg FROM site as s JOIN rating as r ON s.site_ID =
r.site_ID WHERE s.site_ID = 1;
```

Purpose: This query calculates the overall average rating for a specific campsite (site #1) by averaging all six rating categories together.

Query #6: Find the average rating for a single category for a site

```
SELECT AVG(r.cleanliness) FROM site as s JOIN rating as r ON s.site_ID =
r.site_ID WHERE s.site_ID = 1;
```

Purpose: This query retrieves the average cleanliness rating for a specific campsite (site #1), showing how well the site performs in just this one category.

Query #7: Find all the rules related to a single site

```
SELECT r.rule FROM rule as r JOIN site_rule as sr ON r.rule_ID = sr.rule_ID
JOIN site as s ON s.site_ID = sr.site_ID WHERE sr.site_ID = 1;
```

Purpose: This query retrieves all rules that apply to a specific campsite (site #1) by joining three tables to connect rules to sites through the junction table.

Query #8: Retrieve all photo filepaths for campsites in a specific camping area

```
SELECT p.filepath FROM site as s JOIN photo as p ON s.site_ID = p.site_ID
JOIN area as a ON s.area_ID = a.area_ID WHERE a.name = "Hartman Rocks";
```

Purpose: This query finds all photo filepaths for campsites located in the Hartman Rocks camping area by joining the site, photo, and area tables.

1.12. Stored Procedures / Functions

< List all store procedures/functions and a brief explanation of the purpose of each query>

1.13. Miscellaneous

< describe any additional features that you may have included that enriches the project.>

2. Design Overview

Why did you design it the way that you did, what is the purpose of your tables. Don't just list the tables with the column names.

We designed our database to be easy to use, organized, and clear. Each table has a unique purpose that ties into real world needs of campers. For example the user table stores basic information about each person, like their name and where there from. The site table stores all important details about a campsite. Like if it has restrooms, electricity, or a fee. We also included an area table to organize campsites by region with constraints of city, county, and state to make specific and easier to locate. We have also added a rating table to handle everything related to reviews, so people can leave feedback for example on how clean or quite a particular place is. Users can also upload photos to help others see exactly what the site looks like. We added to liked_user_site to track which users liked which sites, and setup tables rule and site_rule to show what rules about to any given site without repeating the same rule to every site. Overall we tried to design things in a way that keeps things simple, avoid redundancy, and just make sure everything connects in a logical way.