# Project Sprint 3

Rating National Parks

# Goals for this sprint

Security

Demonstrate how to set up views and how to grant/restrict what data users can see. Fulfill competency Back-end engineering level 1 and 2 to ensure database exhibits ACID behaviour and identifies privileges to regulate user access using views.

- Success criteria: Create example users with different access to database information

Decide if the database needs all the ACID Attributes and use Transactions to set them up in the database.

- Success criteria: Create transactions queues for the database

# Transactions

**What are they:** Transactions are a unit of work that is executed as a single operation, they ensure data integrity and consistency by allowing multiple database operations (such as inserts, updates, or deletes) to be treated as a single logical unit.

**Why are they important:** Transactions are important because they help maintain integrity and the ACID properties in a database.

**What we are using them for in are database:** Update parks table by adding or removing parks. And adding new weather stations to the Weather stations table

# Grant or revoke users access

**Why this is important:** Granting or revoking user access is important for ensuring that only authorized users have the necessary permissions, when accessing views of performing transactions.

**What is it used for and what are we going to use it for in our database:** We have a free user access to the rating table for trial members of our application. They do not have full access to the application. We also have an avg rating view for each park that full users have access to.

# Views

**What are views:** Views are relations that do not exist physically on the disk, they are virtual tables that are dynamic generated based on the result of a SELECT query.

**Why are they important:** they are important for many reasons. Views can restrict access to specific columns or rows of data, allowing users to see only the data they are authorized to access. They can contain complex queries into a simple manageable object. Views can also improve query performance by pre-computing JOINs or aggregations. And more.

**What we are using for them for in our database:** View for free users to only see limited information on park ratings. View for full users with information about the park and all information viewers would need about ratings. We have a view to select parks that don't have weather stations to allow us to mitigate inconsistencies in the data. View for average rating of each park for quick information on a park

# ACID level in our database.

**Atomicity and Consistency**: We can Guarantee atomicity and consistency by wrapping related queries in transactions.

**Isolation level** in our Rating National Parks is set to repeatable read. This is the default isolation leave for databases made in google cloud console. Repeatable read it ensures consistency and prevents Phantom reads.

**Durability**: using MySQL on the Google Cloud Platform ensures durability by writing committed transactions to storage.

# Goals for this sprint completed

For this sprint our success criteria were to:

- Create example users with different access to database information
- Create transactions queues for the database

We were successful in these criteria by making a full user and free user have differing access to our database tables using views. We also made transactions that drop a park from our tables, add a weather station to our table that links to a park, adds a new park with location and new rating for that park

We have also fulfilled Back-end engineering level 2 this sprint by making a multi user environment by having full and free users able to access system

# Goals for next sprint

Evaluate changing our database into BCNF, 3NF, or 4NF

- Success criteria: Demonstrate completion by showing the decomposition and/or proving that it is in BCNF, 3NF, or 4NF.
- Success criteria: explain why we chose BCNF, 3NF, or 4NF for our tables in Rating National Parks

Create new ERD from our new table configuration and evaluate them using canonical design principles

This would fulfill the course competency data modelling level 3