1 Before you begin

2 Overview of a relational database

3 Starter code - Parks database

4 Basic SELECT statements

5 Common SQL functions

6 **Ordering and grouping query results**

7 Inserting and deleting rows

8 Solutions to exercises

9 Congratulations

6. Ordering and grouping query results

In the previous examples, it may have been difficult to find a specific entry. Thankfully, you can also sort results of a `SELECT` statement using an `ORDER BY` clause. You add an `ORDER BY` clause at the end of the query after the `WHERE` clause (if any) and simply specify the column name you'd like to sort by. The following example gets the name of every park in the database, but sorts the results in alphabetical order.

```
SELECT name FROM park
ORDER BY name
```

By default, results are sorted in ascending order, but you can add either the `ASC` or `DESC` keyword to the order by clause to sort in ascending or descending order. You don't need to specify `ASC` as the first query lists results in ascending order to begin with, but if you want to get results in descending order, add the `DESC` keyword to the end of the `ORDER BY` clause.

```
SELECT name FROM park
ORDER BY name DESC
```

To make results easier to read, you also have the option to group them by column. Before the `ORDER BY` clause (if any), you can optionally specify a `GROUP BY` clause and a column. What this does is separate the results into a subset specific to the column in the `GROUP BY`, and for each column, the results will be filtered and ordered according to the rest of the query.

```
SELECT type, name FROM park
GROUP BY type
ORDER BY name
```

This is best understood using an example. Instead of counting all the parks in the database, you can see how many parks of each type are present, and get a separate count for each.

```
SELECT type, COUNT(*) FROM park
GROUP BY type
ORDER BY type
```

Practice

Take the time to apply what you've learned and see if you can write the following query. Be sure to use the Database Inspector to verify that your code works.

Problem 4: Write a SQL query to the top 5 park names along with their visitor count that had the most visitors, in descending order.