5. Common SQL functions

The first query you wrote simply returned every row in the database.

```
SELECT * FROM park
```

However, perhaps you don't want to return a long list of results. SQL also offers aggregate functions which can help you reduce data into a single meaningful value. For example, say you want to know the number of rows in the `park` table. Instead of `SELECT * ...`, use the `COUNT()` function and pass in `*` (for all rows) or a column name, and the query will instead return a count of all rows.

```
SELECT COUNT(*) FROM park
```

Another useful aggregate function is the `SUM()` function, for adding up the values in a column. This query filters only national parks (as these are the only entries with a `park_visitors` column that's not null), and adds up the total number of visitors for every park.

```
SELECT SUM(park_visitors) FROM park
WHERE type = "national_park"
```

It's worth noting that you can still use `SUM()` on a null value, but value will simply be treated as zero. The following query will return the same as the one above. However, it's still a good idea to be as specific as possible to avoid bugs when you start using SQL in your apps.

```
SELECT SUM(park_visitors) FROM park
```

In addition to aggregating values, other useful functions exist, like `MAX()` and `MIN()` to get the largest or smallest value respectively.

```
SELECT MAX(area_acres) FROM park
WHERE type = 'national_park'
```

Getting DISTINCT values

You may notice that for some rows, the column has the same value as other rows. For example, the type column only has a finite number of possible values. You can eliminate duplicate values from your query results using the `DISTINCT` keyword. For example, to get all the unique values for the type column, you can use the following query.

```
SELECT DISTINCT type FROM park
```

You can also use `DISTINCT` in an aggregate function, so instead of listing out the unique `type`s and counting them yourself, you can simply return the count.

```
SELECT COUNT(DISTINCT type) FROM park
```

Practice