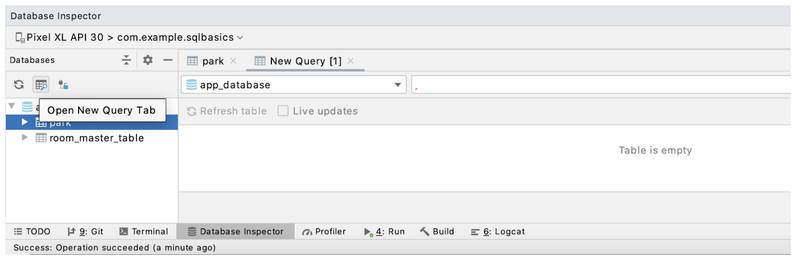
4. Basic SELECT statements

For the following exercises, you'll run the queries in the Database Inspector. Make sure you select the correct table in the left pane (park), click the **Open New Query Tab** button and you should see a text box where you can type SQL commands.



A SQL statement is a command, sort of like a line of code, that accesses (either reading or writing) a database. The most basic thing you can do in SQL is simply getting all the data in a table. To do this, you start with the word `SELECT`, meaning that you want to read data. Then, you add a star (`*`). This is where you would specify the columns you want to select, and using a star is shorthand for selecting all columns. Then, use the `FROM` keyword followed by the name of the data table, `park`. Run the following command in the Database Inspector, and observe the entire table with all the rows and columns.

```
SELECT * FROM park
```

If you want to only select a specific column instead of all columns in the data table, you can specify a column name.

```
SELECT city FROM park
```

You can also select multiple specific columns, each separated with a comma.

```
SELECT name, established, city FROM park
```

Sometimes selecting all the rows in a database isn't entirely necessary. You can add clauses-part of a SQL statement-to further narrow down your results.

One clause is `LIMIT`, which allows you to set a limit on the number of rows returned. So instead of returning all 23 results, the following query only returns the first five.

```
SELECT name FROM park
LIMIT 5
```

One of the most common and useful clauses is the `WHERE` clause. A `WHERE` clause lets you filter results based on one or more columns.

```
SELECT name FROM park
WHERE type = "national_park"
```

> **Note:** Unlike in Kotlin, where the = operator is used for assignment and == is used for comparison, in SQL, you only use a single equal sign to compare two values.

There's also a "not equal to" (`!=`) operator. The following query lists all parks over 100,000 acres that are not a `recreation_area`. With `WHERE` clauses, you can also use Boolean operators like `AND` or `OR` to add more than one condition.

```
SELECT name FROM park
WHERE type != "recreation_area"
AND area_acres > 100000
```

Practice

SQL queries can be useful to answer a variety of questions about your data, and the best way to practice is to write your own queries. Over the next few steps, you'll have the opportunity to write a query to answer a particular question. Be sure to test it in the Database Inspector before moving on.

All exercises will build on the cumulative knowledge from all previous sections, and there will be walkthroughs at the end of the codelab to check your answers.

> **Problem 1:**
>
> Write a SQL query to get the names of all parks with fewer than 1,000,000 visitors.