**Node, Express, and PostgreSQL: Aggregates with Knex and JavaScript Assignment**

**Instructions**

At "Thinkful Eats", you've been tasked to query some aggregate data about various restaurants in the database. There is currently only one table, restaurants, in the database with the following columns:

* restaurant\_id (primary key)
* restaurant\_name (required string)
* cuisine (required string)
* address (required string)
* rating (optional numeric)

To complete this assignment, you will need to complete the tasks described below to get the tests to pass.

**Existing files**

In this checkpoint, all the required server routes have already set up for you, so you won't have to edit any of the \*.router.js files. The test suite will automatically set up a test database and migrate as well as seed the database with some test data as well. Take some time to understand the content of the existing files.

Do not directly change any of the seed files, as the tests rely on the specific test data set to work.

You will then have to write Knex queries to complete the functions defined inside the \*.service.js and \*.controller.js files.

**Tasks**

In src/restaurants/restaurants.service.js:

1. Complete the count() query builder function to get a count of the number of restaurants stored in the database. Then in src/restaurants/restaurants.controller.js, update the count() route handler.

The HTTP response body structure should look something like this:

{ data: { count: 10 } }

1. Complete the averageRating() query builder function to get an average rating for all the restaurants stored in the database. Then in src/restaurants/restaurants.controller.js, update the averageRating() route handler.

The HTTP response body structure should look something like this:

{ data: { average\_rating: 4.1050000 } }

1. Complete the readHighestRating() function to return the highest rating amongst restaurants. Then in src/restaurants/restaurants.controller.js, update the readHighestRating() route handler.

The HTTP response body structure should look something like this:

{ data: {

max\_rating: "5.00",

}

}