Galvanize Memory

You're making an API that can list, create, read, update, and delete a collection of coffee. The problem is that you're missing a critical part of the API-- the database! Create a database for this API, and interface with it via Knex.js.

Setup

Make a local database and setup a `knexfile` to connect to it.

Migrate

Make a migration that creates the following database table:

`coffee`

key	name	data type
PK	id	auto-incrementing integer
	name	text
	roaster	text
	aroma	integer

Seed

Seed your database with some data:

field	value
id	1
name	Black and Tan
roaster	Ink
aroma	3

field	value
id	2
name	Holiday Roast
roaster	Starbucks
aroma	9

field	value
id	3
name	House Quake
roaster	Denver Coffee
aroma	6

Make sure your next auto-incrementing integer starts with `4`!

Database connection

Make a connection to your database in the `database-connection.js` file with the appropriate environment data.

Queries

Fill out the `queries.js` file with the following:

- `list()` should return a promise that resolves with all of the data in the `coffee` table as an array
- `read(id)` should return a promise that resolves with the record with a matching `id` as an object
- `create(coffee)` should return a promise that inserts a coffee object and resolves to the created database record as an object
- `update(id, coffee)` should return a promise that updates a coffee record matching `id` with the data in `coffee` and resolves to the updated database record as an object
- `delete(id)` should return a promise that removes the record matching `id` and resolves to nothing

Deployment

Deploy this API. Note that you'll need to create a remote database, run your migration and seeds on it, and connect to it in production and your local database in development.

Notes

- You can test your API locally with `npm test`

Add a link to your [deployed API]() here.