SPRING
DISCOVERY

Overview

In this project, we'll be addressing a critical need for a hypothetical new grocery store, Spring Foods™: that of creating a *'single source of truth'* -- a one-stop shop for understanding all the products that have come in and out of the store, along with any intermediary byproducts that were consumed along the way.

The challenges involved are inspired by those that you might face in spinning up a 'lab of the future', where you'd like to be able to track all the samples that come in and what happens to them afterwards.

There is no single correct answer for this project. **We expect you to explore the data on your own with the time you have available, use our questions as guides, and be ready to talk to us about your chosen approaches as well as the next steps you'd take if you had more time.**

Time Expectations

We expect you to send us your work within 4 days of receiving the project (if you need more time or can't start right away, please let us know). There is no enforced minimum or maximum amount of time to be spent — we suggest ~1 hour / day. If you only have an hour or two to work on this total, that's perfectly okay. If you have more time, that's fine too. We are most interested in how you approached the problem, your thought process, and what you would do next if you had more time.

Introduction

Congratulations! You've just been hired as the CTO of California's newest startup grocery store, Spring Foods™.

It's halfway through your first day on the job, and you've been handed the attached directory. It contains a bunch of scattered files -- there are some spreadsheets, some PDFs, and more.

You soon realize that these files represent a mix of:

1. **Purchase logs.** Spring Foods™ purchases its produce directly from farmers and other vendors. The purchase logs include metadata pertaining to every item purchased, which is great! But that metadata is often messy and inconsistent.
2. **Internal records.** Spring Foods™ employees have been responsible for doing their own bookkeeping, and so these too are unconsolidated (but can be considered accurate).

This lack of organization makes it hard for us to answer questions like: "How many apples do we have in-stock? How long has this banana been on the shelf? Where did this stick of butter come from?"

Your goal is to make sense of this data by designing and implementing a one-stop shop for all product tracking at Spring Foods™.

The Challenge

We would like to design a system that meets the following requirements:

- All information from the directories is captured in the system. This includes both the information provided by vendors prior to delivery (the purchase logs), and the information produced by Spring Foods™ afterwards (the internal records).
- The system could be extended to accommodate future orders and future internal protocols. (As an example: if we were to start freezing some of our items on arrival, we'd like to track the dates at which they were frozen and thawed.)
- Technical folks (those with programming experience) can access the data programmatically.
- Non-technical folks (those without) can access the data without writing code.

First, **find a way to organize this data.**

We'll make a couple of assumptions:

- Every item (piece of fruit, carton of milk) will have an identifiable source (tree it came from, cow that was milked).
- Internally, we occasionally use products that we bought to create new products of our own (e.g., milk => butter). In those cases, the input product is used up entirely.
- Any expired items are immediately thrown out.

- (You can safely ignore any files that aren't human-readable.)

You're free (and expected) to make other assumptions -- just document them (when prompted).

You can use literally any tools you want: relational databases, web frameworks, spreadsheets, hosted web applications, a combination, something else entirely. Pick the right tool for the job -- ***we value pragmatism***, feel free to surprise us.

When you've settled on an approach, **find a way to load the data into your system of choice**. This could involve writing code, manual processes, or something else. No right answer. (If you're only able to load a subset of the data, that's okay -- just make sure there's a clear path to loading the rest, given more time.) You will have to make some assumptions along the way -- ***we value decisiveness***, especially in the face of ambiguity.

Then **answer the following questions** using the attached template:

1. Why did you settle on this design? Justify any significant design decisions you made (tools you chose, etc.). What are the limitations of your design? What alternatives did you consider?
2. What assumptions (if any) did you have to make about the data?
3. How many items did Spring Foods™ have in-stock as of January 28, 2019? (How'd you arrive at this number?)
4. How would technical folks go about answering Question #3? Non-technical folks?
5. How would you envision future data getting into the system? What processes and technical solutions would need to be put in place? (Imagine that this system is being used by data scientists, cashiers, and others.)
6. If we were to start freezing some of our items on arrival, what work would be required to track freeze and thaw dates?

(No need to make this lengthy; bullet points or short explanations are fine.)

Submission

In your submission, please include the following (again, using the attached template):

- Any code, links, or instructions necessary for us to understand your setup, along with any relevant instructions. This *could* include any of: screenshots of what you built, a link to a live application, a README with instructions for getting a web application up and running, Python scripts that were used to clean up the data, SQL queries that were run to answer Question #3, etc.
- Answers to the questions in the section above.

Thanks and good luck! Remember, we don't have the right answers here -- in fact, we're eager to see what you come up with.