Ask the client to present the most common scenarios in order to find some common computational patterns. I assume these are the 8 mentioned in the case.

This would give us an idea to what level we can aggregate the data and still compute the numbers for the simulations.

It would also give us an idea how to organize the simulation engine (i.e. which simulation parameters that can be configured in the application/dashboard)

What output does the client expect of these simulations?

- Visualization of the vendor revenues
- Just a number with the total revenue
- Visual summary of all 20.000 simulations

Do they need to keep all the data or just the data of last month?

Data cleaning, some distances/fares are below zero?

Project Plan:

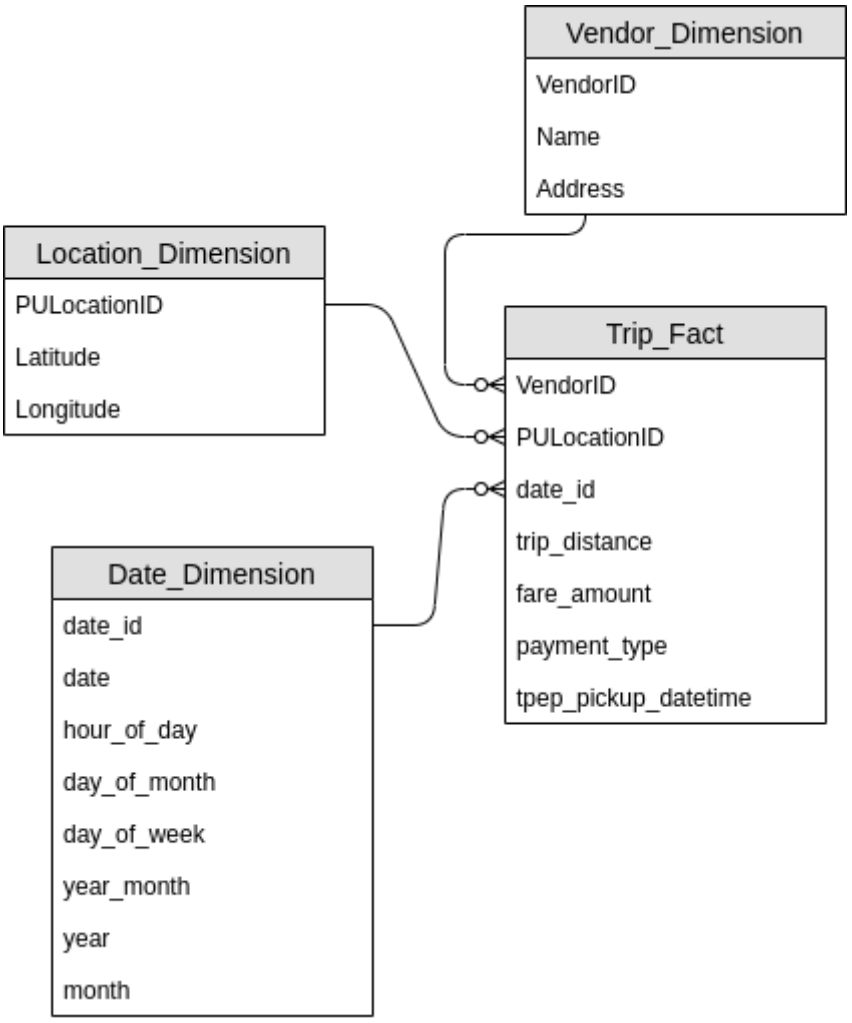
1. Store data and load in data warehouse
2. Make a few sample scenarios for the client to give feedback on
3. Iterate
4. Add the full set of required scenarios

Challenges

Given that they want to run 20k simulations per month on a month's data, we need to be able to process each simulation in roughly 2 minutes if done serially. Probably we don't want the computations to take an entire month. So many parallel queries will be made.

If they want to run simulations using all data (not just last month's) the size of the data might also become a limiting factor.

Data Model



Data layer and technologies

We need to be able to filter on **pickup period** (month of year) and **payment type**. We need to be able to group records by **VendorID and/or PULocationID**. So these four are **dimensions**.

The sum over trip distance and fare amount represent the **metrics** of interest.

So we would definitely need **indices** on VendorID and PULocationID. The data could be **partitioned** over the dimensions year_month and payment type. This way only one partition needs to be accessed or reindexed (see docs SNAP - Spark Extension for BI)

Some scenarios require very similar views on the data model. For example all given scenarios can be computed based on an aggregate **view** where the fare and distance is aggregated over the vendor, pickup location and payment type.

Every month the data can be pulled and stored in an object storage (like S3), partitioned per year-month. From there it can be processed and loaded into a column-oriented data store:

- Data is relational and the scenarios seem to be a match for SQL queries
- certain columns don't contain many different values and could be very well compressed.

Redshift could be used as a column-oriented data warehouse solution. Or Oracle's Spark native SNAP extension.

Redshift allows to add compute nodes dynamically for increased number of concurrent queries.

Simulation Engine

Some of the simulations might need more complex formulas (DAX) on top of the defined tables or views.

Either these functions need to be predefined in python (possibly parametrized) and then run against the views or these formulas need to be expressed and stored in a BI tool.

If you don't use a BI tool, you'll probably want to create an API and a web application where you can select your scenarios and parameters.