[redacted]

[redacted]

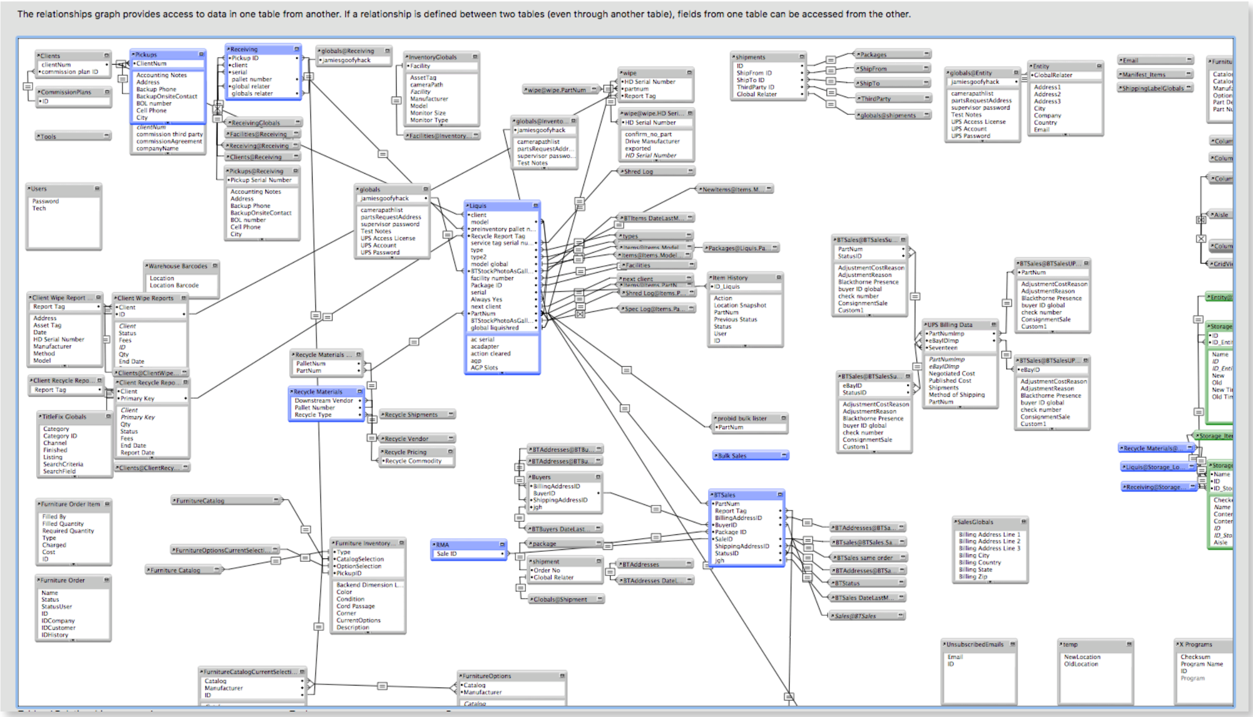
Benjamin Harrison
December, 11th 2017

I. Introduction

[redacted] is the database by which inventory is cataloged and moved through the recycle->test->reuse workflow employed by Liquis.

In it's current iteration, [redacted] has expressed breakage between the DB itself and third-party connectivity (i.e., UPS WorldShip, eBay, etc.).

Having spent a significant amount of time pouring over the backend (see below), I've been able to isolate several problematic areas and the overall problem with the current iteration.



The relationships graph provides access to data in one table from another. If a relationship is defined between two tables (even through another table), fields from one table can be accessed from the other.

II. Problem

There are several factors at play that affect both the performance and functionality of [redacted], that (a) prevent it from being 100% utilized and efficient and (b) eventually led to breakage with third-party connectivity.

Overall, the database is old, with technology used to develop it being at least a decade out-of-date. In this amount of time, each piece of third-party software has probably gone through at least a half dozen or more updates and versions.

Secondly, the database table schema (i.e., backend) was not built using FMI's best practices or by a certified developer, so the logic is extremely hard to follow. It isn't impossible, but is akin to trying to locate a needle in a haystack, if you had to find the haystack first.

Lastly, there are a lot of 'dead' tables and redundancy built into the current system. This bloats it and most assuredly has a negative impact on the performance of the DB, overall.

III. Recommendation

While the frontend could use modernization, the life-blood of any relational database is the solidarity inherent in a well-designed backend. This is not the worst backend I've ever come across or worked with, but there are a lot of things that need to be done and re-done with it.

As such, my recommendation is to build from scratch, a newer, sleeker and vastly more modern solution. A renovation on the current backend could take 8-10 times as long to complete versus just building from scratch.

With this approach in mind, a number of solutions will present themselves:

- The backend will follow a logical path and utilize FMI's best practices, so any other certified developer can pick it up in the future.
- The breakage with third-party apps will be non-existent.
- The frontend will have a more intuitive and user-friendly interface/navigational scheme.
- This solution will be built with both scalability and future-proofing in mind.

IV. Conclusion

Attached as an addendum to this proposal is an estimated scope of work that details both the time and costs that this project will entail. I've made it as thorough as I possibly can.