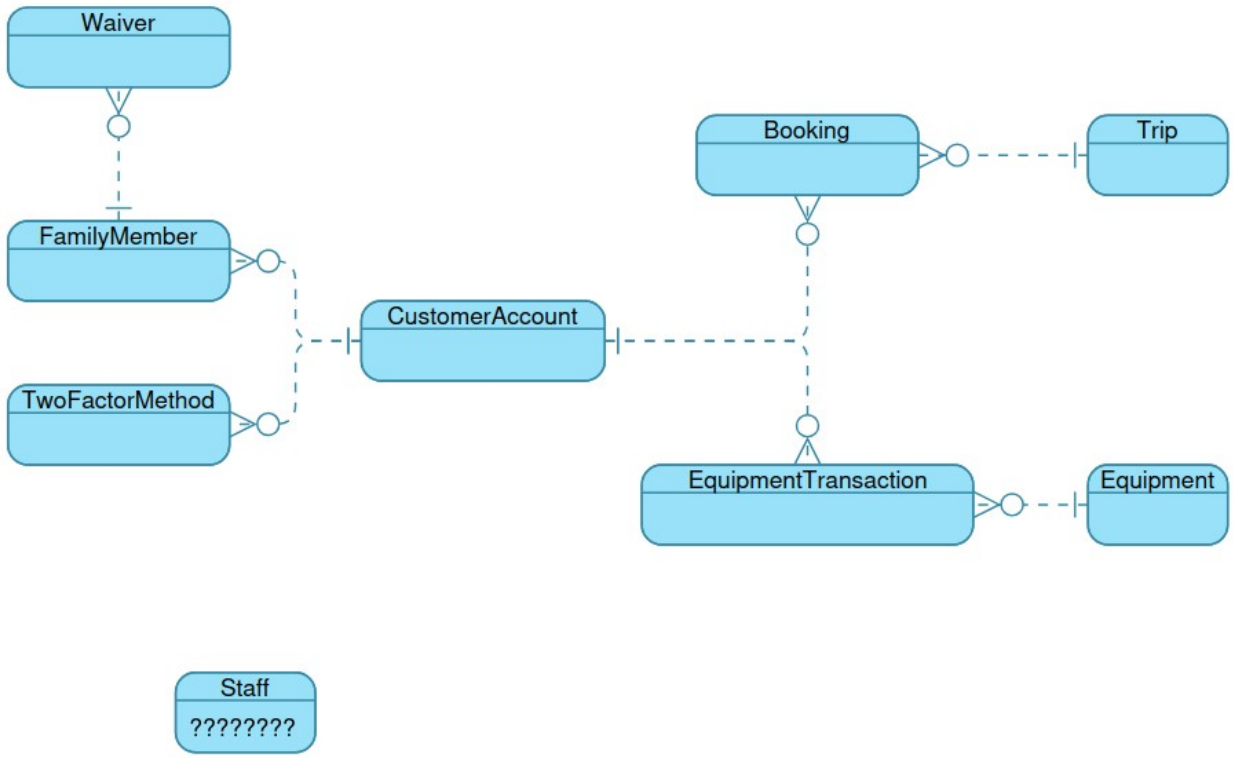
**CSD-310: Database Development and Use**
**Assignment 12.1: Case Study Critique**
**Red Group: Outland Adventures**
**Isaac Ellingson**

## ERD

I can't properly review this ERD. The relationship lines are obscured by other entities that are not taking part in the relationship. For example, the relationship line coming out of Equipment.EquipmentID turns twice *while obscured each time*, ostensibly re-emerging finally to connect to EquipmentTransaction.EquipmentID. The only reason I think this is because of my familiarity with the problem domain and my preconceived notions of what said line *should* link to. I have no way of knowing.

I will be reviewing this design based on these presumed but unknowable relationships. If you see something incorrect here, please disregard my comments on it.




The reason I'm picky here, the point of the ERD is that if you come back in six months and need to write a new query, you will be looking at the ERD, not the database creation SQL. This documentation needs to match the database precisely, and be enough that you could write a query against the database it documents, especially with joins.

## Report Formatting

Most of the reports displayed are in the form

```
Key1: Value1
Key2: Value2
-----------------
Key1: Value1
…
```

This isn't really sufficient to make business decisions. The tabular display of "Booking Trends" is what we want to work towards here. Reports are a *client-facing* representation of our data geared towards helping them answer questions. They need to be clear and focused on the problem.

## Equipment Profit and Rental Performance Report

I feel like I've got a lot of visual noise here, and we're *so close* to giving a really satisfying answer to the client's questions. The database is right for this question, and you've got everything you need here. The sale price minus the initial cost gave you the net sales profit of each equipment item. The rental count, times the rental price, gives you your total rental revenue, and your rental revenue minus your initial cost gives you a net rental profit. What a manager's going to want to see is net sales profit, and net rental profit. Since these numbers are directly comparable, unlike rental revenue vs sales profit, they can be used to make a decision about whether to shut down the rental business.

And based on the sample data, they should definitely shut down the rental business.

The next thing they're going to want to see is a totals line. It's useful to know individually, for example, if people were choosing to rent high-ticket items a lot more, and it was profitable to rent boats out but not low-ticket items like lanterns. But it's also really important to see what the rental vs sales business is like in aggregate. Once again, two key numbers should stand out: total sales profit vs total rental profit.

## Inventory Age Report

Once again, I'm seeing visual noise. I would avoid displaying information not useful to answering the business question, such as "days since purchase" and perhaps even "equipment category".

Additionally, I would recommend sorting the data table by equipment age (descending) and condition (good to bad), so that we start the report with the equipment in most need of attention. Consider switching Condition to an ENUM field to facilitate this kind of sorting. I know you assumed Conditions were entered as freeform text, but imagine the conclusions we could draw with it regularized!

## Booking Trends Report

While nicely formatted and readable, I wonder about calling this a “Trends” report. When a client says “trends”, we’re going to start looking at what’s new. Are participants in Africa down this month? Is there a sudden interest in Southern Europe? We don’t know.

Please consider instead, since the client has expressed their question as one about region performance, using a table grouped by Region, displaying last year’s summed participant counts versus this year’s. You could even compute a difference. It’d be a really good “At a glance” view and give the customer clear signals on what a region is doing.

## Database Design

Overall much of this data gives really strong support to the reports. This design represents good assimilation of domain knowledge, and can produce a very detailed and accurate accounting.

I am most put off by this orphaned Staff entity. It’s not completely forbidden to have a table with no relations, but with it also not appearing in any of the reports, it seems to not be answering any business questions here either. Consider parting with it.

Many tables are in a kind of limbo – FamilyMember, TwoFactorMethod, and Waiver are related to other data, but do not participate in any reports. Going back to the case study, it appears Mei Wong is doing the ecommerce site, so I’m not sure these entities are even really appropriate in this setting.

Booking is acting like a link record in this design, linking accounts to trips in a many-to-many relationship. I like this design – but lets consider duplicates. Is it valid to have twelve duplicates of this record, with different BookingIDs but otherwise identical data? I would feel more comfortable if the database said, “you’ve already registered for this trip”, but it depends on your interpretation of the data, what a duplicate means to you.

Consdier using an ENUM type for Booking.Status accepting only “Pending”, “Confirmed”, or “Cancelled”, to make sorting easier and to prevent typos from entering the database.

I have several questions about CustomerAccount. Is there a difference between AccountName and Username? Is there a difference between AccountID and AccountName? What kinds of values does AccountStatus hold? I would consider keeping a username and a “display name”, or dispensing with the username too and identifying users by email address.

It would appear that in this design each transaction is for multiples of one item, and that no transaction ID can represent a pretty normal-seeming purchase of a lantern, a sleeping bag, and a tent. I feel like this choice is also going to result in a lot of sneaky duplicate data, such as the transactionType and transactionDate being copied across all three “transactions” in our lantern-bag-tent example. If they were rented and then carelessly destroyed that’s then three places to change it into a sale.

### Possibly Unnecessary Advice

In Waiver, there are two boolean fields that may become inconsistent from each other, “SignedByMember” and “SignedByParent”. Consider an enum “SignedBy” field which accepts only “Member” or “Parent”.

TwoFactorMethod appears to have a few inconsistencies brewing where the same customerAccount can have three simultaneous primary two-factor methods, all while two-factor is disabled. Some fields may need to move from one table to another to prevent this. For example, a nullable foreign key on CustomerAccount for PrimaryTwoFactorMethodID would allow the customer to have exactly zero or one primary 2-factor method.

## Conclusions

You’ve given yourself a lot to work with here, but it’s not really shining through. The most important thing is cleaning up the ERD so that relationship lines do not pass under other entities, and formatting all the reports as tables. After that, making sure to really connect the report contents with the questions the client has will greatly strengthen the product.

I know I had a lot of notes, but as I said earlier, in many ways the design is also great for producing clear, detailed, and accurate answers. I especially liked the detailed profit analysis, if you just tweak that report a little bit it’s going to be the star of the show.