**Project Deliverable 1 & 2**

**Elias Joseph**

**Summary**

Like any business, a gym can have a lot of moving and interacting parts once the operation becomes large enough. That is why I have designed a database that helps facilitate the operation of a gym. Specifically, a chain, as having multiple locations will make having well organized data an even larger necessity. This database will include all the facilities, as well as the employees and features of each, in addition to all the members of the gym, and their relevant information. It will also include details about classes offered, and member check-ins, to allow for a better analysis of what services are the most popular.

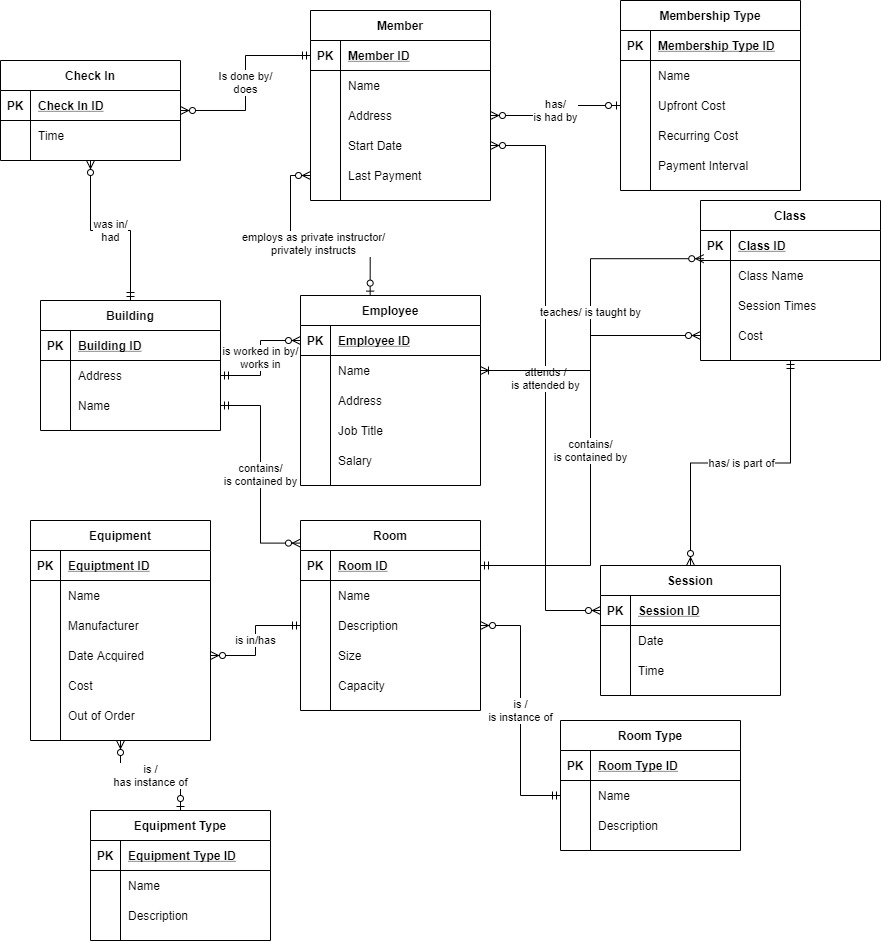
The primary stakeholders would be the owners of the gym. Having a well-organized database containing all the information necessary to run the business would reduce costs, as well as make data retrieval more time efficient. Having easy access to data would also allow better analysis of current business practices, so that those practices could be improved. The customers are also a stakeholder, as data mismanagement could make the product harder to used. Having a well-organized database would mean they would face less disruptions such as problems checking in, failing automatically to renew memberships, issues on the data management end. Any improvements to customer experience made as a result of this data would also benefit them.

As stated earlier, this database could be used to answer questions about how best to run the business. Some examples are:

1. What Membership plan is the most popular / most profitable?
2. How likely are Members to use a different Gym from their usual one? Would it be worth it to try and allow Members to participate in their classes / personal training sessions, or would that be too much work to implement?
3. How do active classes effect general attendance? Does the room they take place in affect attendance?
4. At what times are people most likely to use the Gym?
5. Does the presence of certain room types increase the popularity of a facility?

At a high level, this database represents a gym chain made up of separate facilities. Each facility has rooms, and employees (although employees don’t necessarily have a facility, as they may be working remote, or work at an HQ building). There are also customers, who each have a membership plan. There will be a defined ‘No Membership’ plan, so every member in the database must have a plan entered. This will help track and automate payments. Customers may also hire a private trainer, creating an optional link between member and employee. The database will also track classes offered. The classes will have rooms and instructors, and will be broken up into sessions, with specific times and attendees. For more detail, see the full conceptual model below.

**Conceptual Model**



**Glossary**

**Building**: A specific gym location, with an address, and a name more readable than a primary key. Has employees who work there, rooms within it, and also records the instances of members checking into the gym.

**Check In**: An instance of a member checking into a gym. Records the time and date where the member attended one of the gyms.

**Class**: A class offered by the gym for more structured exercise. A class is made up of sessions, has a room in which it takes place, and has one or more employees as instructors.

**Employee**: An employee of the company. This table contains information such as name, and salary. Employees can have gyms that they work with, although this is optional to allow for the possibility of remote workers. They also can teach classes, or privately instruct members.

**Equipment:** A piece of equipment at the gym. Is contained in a room and has a specific equipment type. This table records information such as cost, and manufacturer, as well as having a Boolean value to indicate whether the piece of equipment needs maintenance.

**Equipment Type:** A table that specifies the overall types of equipment that might appear in the gym.

**Member:** A table containing the gym members, as well as their information. Members have check ins which are recorded. They have a membership type which indicates what services they have access to and how much and how frequently they are charged. They can also attend sessions of classes and can hire a private instructor to teach them.

**Membership Type:** A specific type of membership offered by the gym, which tracks how much and how often members should be charged. None is considered a valid membership type, for members who currently don’t have active membership. This will make it easier to reinstate people’s membership.

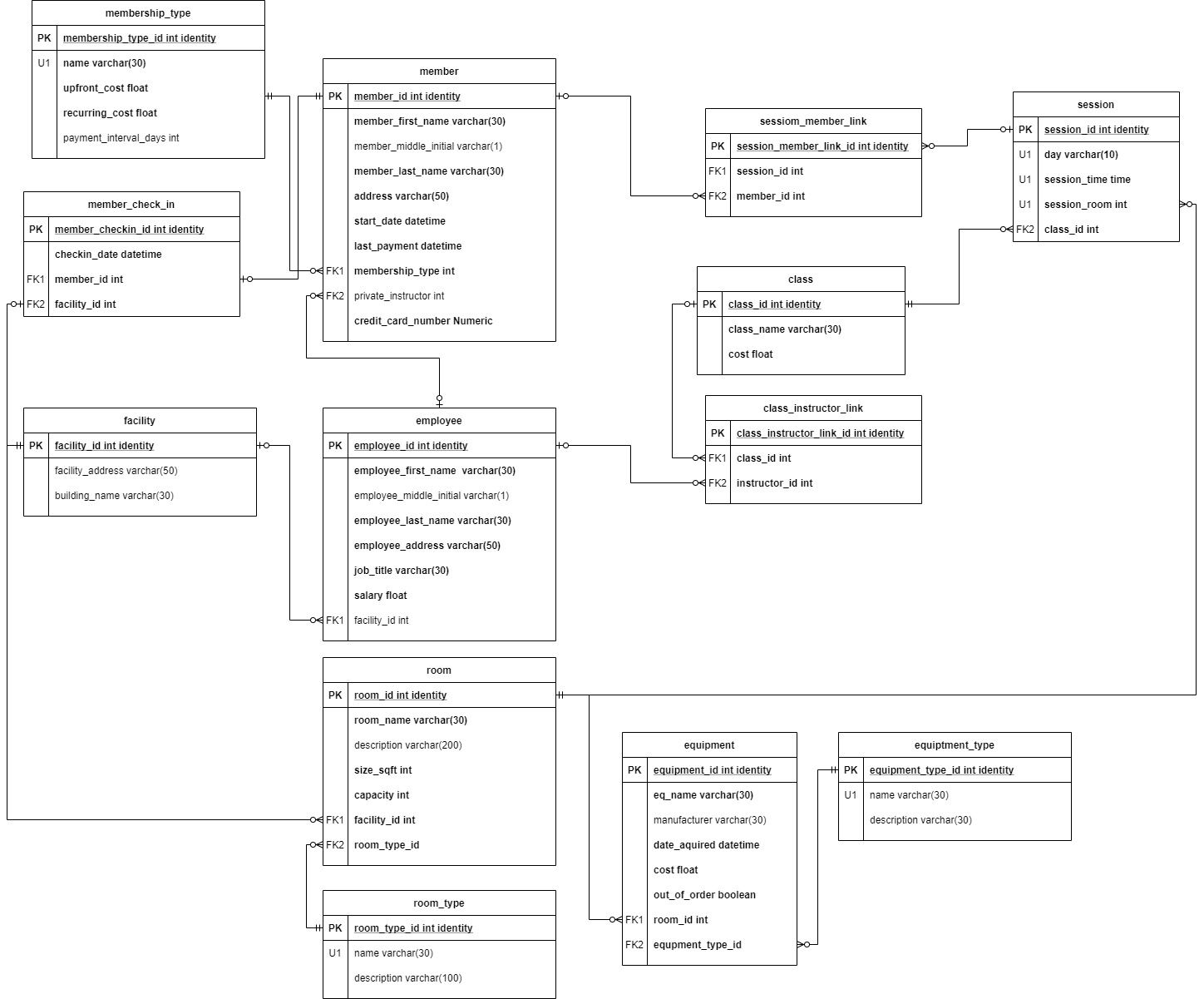
**Room:** A room within a building. Rooms have information such as capacity, as well as the equipment that should be in the room, and classes that take place there. They also have a specific type, such as ‘pool’ or ‘weight room’.

**Room Type:** A specific archetype of a room such as ‘pool’ or ‘weight room’.

**Session:** A specific session of a class with a specific time and date. Sessions have members as attendees.

**Logical Model**

The logical model shows a clearer example of how I will implement the database. I added two surrogate keys, being session\_member\_link and class\_instructor\_link. Theas are both many to many relationships. Sessions have multiple attendees, and members can attend multiple sessions of different classes. Employees can teach multiple classes, and a class could be taught by multiple people. I implemented the primary keys exclusively as integers, because that is easier to manage. Most descriptive values I used varchar.



**Deliverable 2**

**Data Questions:**

**Question 1: What Membership plan is the most popular / most profitable?**

To answer this question, I plotted the number of people subscribed to each plan, as well as the amount of money each plan was earning.

The premium plan is the most popular and is bringing in by far the most money.

Chart, bar chart

Description automatically generated

Chart

Description automatically generated

**Question 2: How likely are Members to use a different Gym from their usual one? Would it be worth it to try and allow Members to participate in their classes / personal training sessions, or would that be too much work to implement?**

To answer this question, I looked at a heatmap of which gyms each member signs into.

Chart, waterfall chart, surface chart

Description automatically generated

It looks like members primarily stick to one gym, so it would not be worth it to implement cross facility classes.

**Question 3: How do active classes effect general attendance? Does the room they take place in affect attendance?**

To answer this question, I graphed the distribution of the number of visits of members who have either hired a private instructor or signed up for a class, vs the number of visits of those who did not.

Chart, box and whisker chart

Description automatically generated

As you can see from this graph, members who participate in training attend far more often than those who do not.

**Question 4: At what times are people most likely to use the Gym?**

To answer this question, I looked at histograms of the times people checked. When looking at the date, the number of visits follow a pattern of spiking and then dipping back down. The visits tend to be between 8:00 am and 8:00 pm.

Chart, histogram

Description automatically generated

Chart, bar chart

Description automatically generated

**Question 5: Does the presence of certain room types increase the popularity of a facility?**

**A picture containing text, receipt, screenshot

Description automatically generated**

To answer this question, I looked at a linear regression model for the number of visits a facility received vs the number of each room type.

Overall, it looks like the room types do not significantly affect visits.

**Example of Data Entry Form for New Member:**

Here is an example of what a user-friendly data entry form could look like. All the other data in the member table can be generated automatically.

Text

Description automatically generated

**Summary:**

To create my repeatable script, I started by creating the tables. For each entity I created, I added a ‘drop [entity] if exists’ clause. After creating the tables, I added several procedures to handle actions such as modifying a member’s membership plan or changing the role of an employee. Next, I added some randomly generated data. I used the text file in this GitHub repository (https://gist.github.com/zubietaroberto/b1e14a1f7cc307749d02c99df398c84a) to create names and addresses. Afterwards, I wrote my select statements as well as some helper functions so that I could query the solutions to my data questions. I’ve attached the code in a file called ‘Elias-Joseph-Repeatable-Script.sql’

**Reflection:**

Overall, I think I underestimated how many relations really exist in a database. There are a lot of patterns that should exist that don’t because I didn’t have time to add them to my data generation. For example, I have people with no active membership taking classes, and people’s check ins don’t line up with when they have classes to attend.

Something else that caught me off guard with this project as well as with this class, was that working with Microsoft Access was much more difficult than working with SQL studio, so I wish I had budgeted more time to figuring that out.

Overall, I think this project will make me consider the relationships between entities a lot more. I’ve worked a lot with data frames in the past, but those force the data into a 2D shape. This project has made me think about how to use multiple tables to avoid entering a lot of redundant data.