Bradley Smith

Dr. Antonios

CSC 400

8/19/2020

IMC Database

An Aggregated Data Solution to Streamline Service

Bradley Smith

CSC 400

Dr. Imad Antonios

8/18/2020

BACKGROUND

I work for IMC, LLC which is a Managed Service Provider (MSP) that performs IT service for dozens of clients. Our focus is the health, performance, and security of our clients' computers and networks. To accomplish this, we use various software to monitor and secure their devices. These software include:

- ConnectWise Automate
    - Grants remote access to the machine and monitors the hardware/operating system running conditions.
    - Each client has one or more locations for their business. Each location is comprised of computers. Clients are billed per computer at each location.
- Cloudberry
    - Cloudberry performs data backup and restoration both locally and in the cloud.
    - Each client may or may not have this service, but when they do it is billed based on the amount of data they are backing up.
- Sentinel One
    - Sentinel One is an AI based antivirus and antimalware. It performs automated scans and reports the status of each machine.
    - Clients are billed for each computer that has a license of this software.
- Malwarebytes
    - Malwarebytes is just antimalware. It performs similar automated scans with status reports as Sentinel One but does not have an antivirus component.
    - Clients are billed for each computer that has a license of this software.
- Microsoft 365
    - Some of our clients have their email hosted through us on Microsoft 365.
    - Each client is billed for each email that has a license attached.

When it comes time for our billing department to push out a monthly bill, they are tasked with getting the updated numbers from each software. Currently that process takes an exorbitant amount of time. To confirm the numbers of ConnectWise automate licenses, they would have to navigate to each location within the software and manually record a number presented at the top, then move to the next location. This process takes some loading time but must be done for all 90 locations. For our software Sentinel One and Malwarebytes, they must log into the web portals, navigate the menus, organize the cites and groups within the portal, and manually count the machines listed. For Cloudberry we need to remote into each machine, load up the software and total each backup set for each computer that has backups for each location. Lastly, for Microsoft 365, we currently keep track of the monthly additions and subtractions of licenses. Manually counting them per domain is a pain separating those with and without licenses but do have emails on each domain.

My solution for this problem is to combine these databases from each software onto one platform so we can generate tables which report the proper numbers for each client and location we service. At the same time, we can use these databases we gather and use other features they report. For example, which software is out of date? What machines have security issues? From these specialized queries, as a project manager, I can make projects based off these custom reports that cannot be found anywhere else.

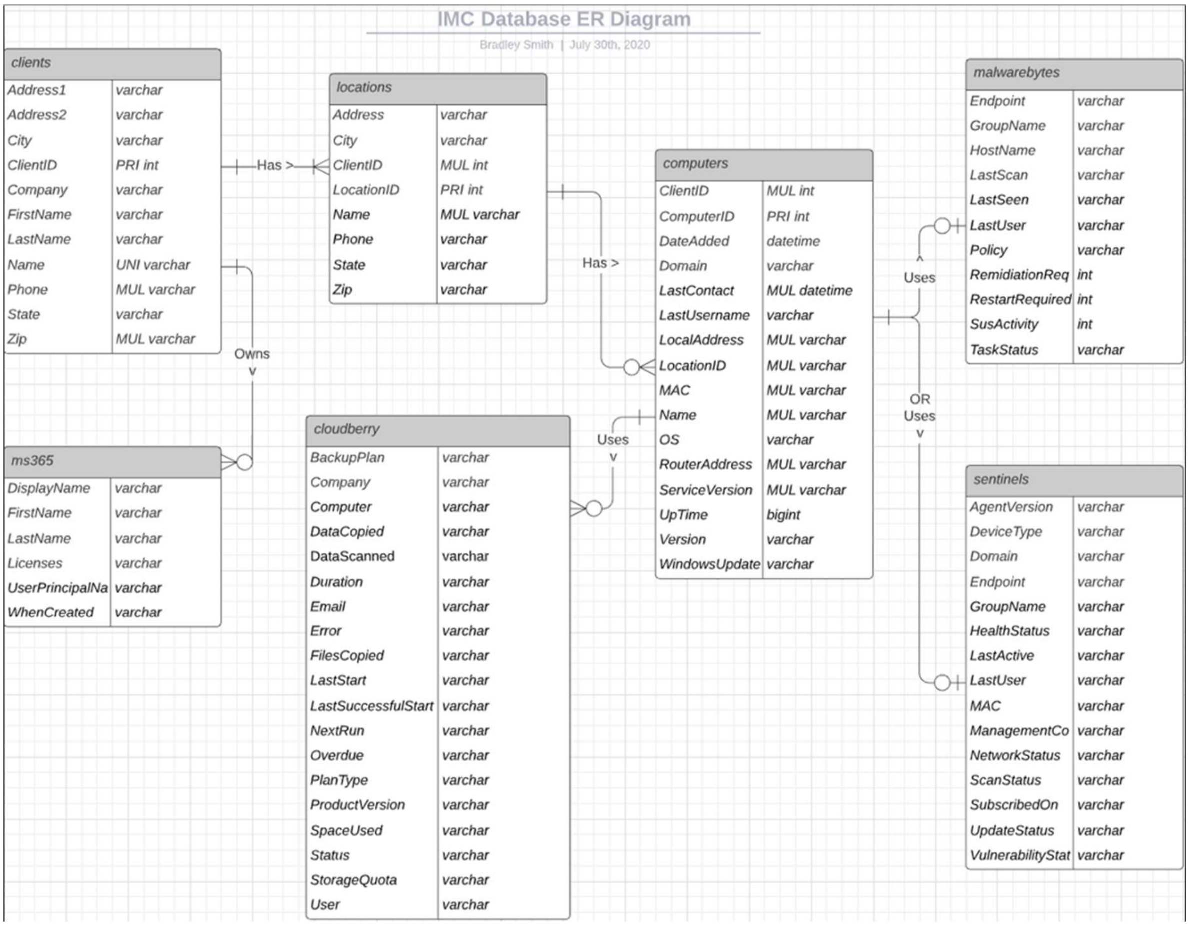
IMPLEMENTATION

This project was accomplished in 5 stages:

1. Collect the Databases
2. Combine the Tables
3. Design the Queries
4. Host the Results
5. Polish the Website

1. Collect the Databases

Each service had their data exportable in one way or another. For instance, ConnectWise Automate needed to be exported as an SQL Database, most others were exported as a CSV. I met with each service provider to find the best way to get the data I needed. In the future, updated copies of these databases will be available directly from the portals so there will be no need to contact their support staff.



clients	
Address1	varchar
Address2	varchar
City	varchar
ClientID	PRI int
Company	varchar
FirstName	varchar
LastName	varchar
Name	UNI varchar
Phone	MUL varchar
State	varchar
Zip	MUL varchar

locations	
Address	varchar
City	varchar
ClientID	MUL int
LocationID	PRI int
Name	MUL varchar
Phone	varchar
State	varchar
Zip	varchar

computers	
ClientID	MUL int
ComputerID	PRI int
DateAdded	datetime
Domain	varchar
LastContact	MUL datetime
LastUsername	varchar
LocalAddress	MUL varchar
LocationID	MUL varchar
MAC	MUL varchar
Name	MUL varchar
OS	varchar
RouterAddress	MUL varchar
ServiceVersion	MUL varchar
UpTime	bigint
Version	varchar
WindowsUpdate	varchar

malwarebytes	
Endpoint	varchar
GroupName	varchar
HostName	varchar
LastScan	varchar
LastSeen	varchar
LastUser	varchar
Policy	varchar
RemidiationReq	int
RestartRequired	int
SusActivity	int
TaskStatus	varchar

ms365	
DisplayName	varchar
FirstName	varchar
LastName	varchar
Licenses	varchar
UserPrincipalNa	varchar
WhenCreated	varchar

cloudberry	
BackupPlan	varchar
Company	varchar
Computer	varchar
DataCopied	varchar
DataScanned	varchar
Duration	varchar
Email	varchar
Error	varchar
FilesCopied	varchar
LastStart	varchar
LastSuccessfulStart	varchar
NextRun	varchar
Overdue	varchar
PlanType	varchar
ProductVersion	varchar
SpaceUsed	varchar
Status	varchar
StorageQuota	varchar
User	varchar

sentinels	
AgentVersion	varchar
DeviceType	varchar
Domain	varchar
Endpoint	varchar
GroupName	varchar
HealthStatus	varchar
LastActive	varchar
LastUser	varchar
MAC	varchar
ManagementCo	varchar
NetworkStatus	varchar
ScanStatus	varchar
SubscribedOn	varchar
UpdateStatus	varchar
VulnerabilityStat	varchar

Figure 1.1 Data Tables

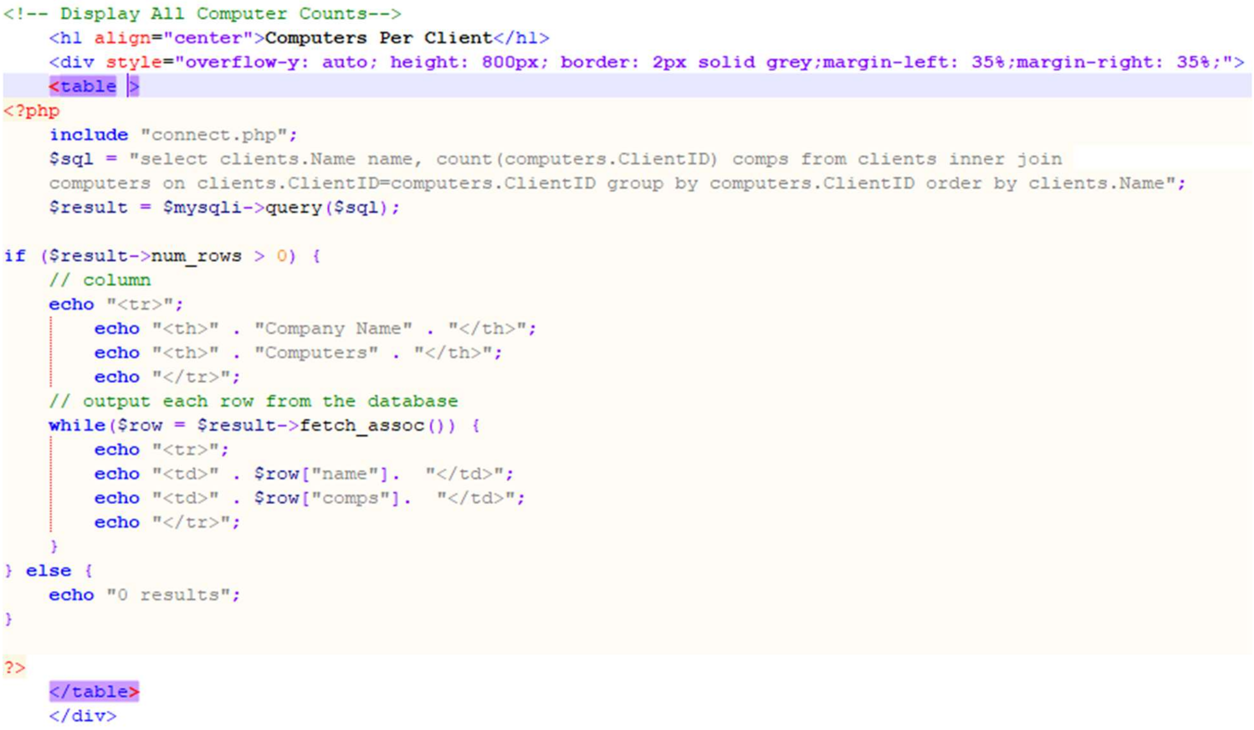
2. Combine the Tables

Once I got copies of each database, I combined them into one database as tables. The data came with quite a few fields I did not need so I deleted fields I would not need for the project. As a future consideration, I would collect the database in its entirety but in the queries, only use the fields I needed. I uploaded them onto a software called XAMPP which allows me to see, configure, and host the database.

Some complications came where two different tables had similar values comprised of different domains. For example, the Endpoint in Malwarebytes table is the same as the Name in the Computers table. But the domain for Endpoint was considered various characters of length 50 where for Computers it was characters of length 25. This subtle difference can create issues when designing queries

3. Design the Queries

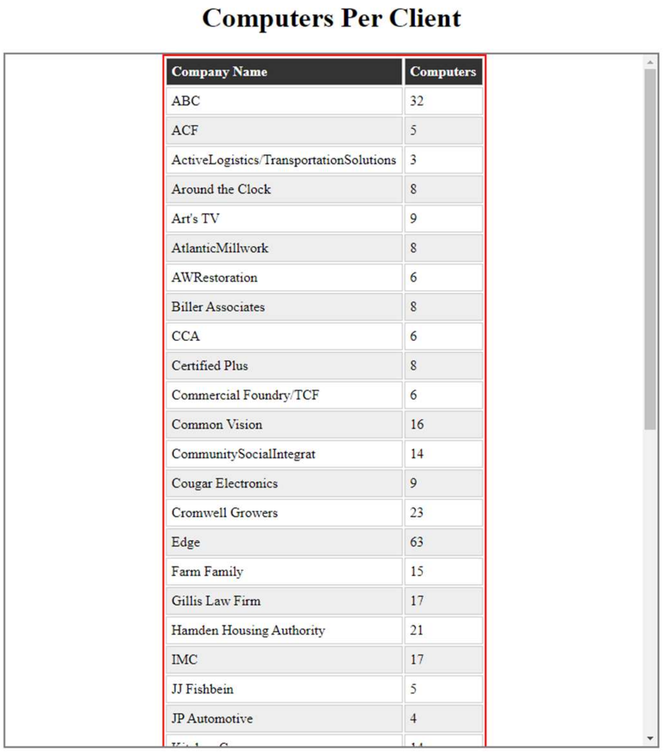
Once the tables were combined, I was able to create and test some queries. The reports we are interested in involved connecting those tables across like values.



```
<!-- Display All Computer Counts-->
    <h1 align="center">Computers Per Client</h1>
    <div style="overflow-y: auto; height: 800px; border: 2px solid grey;margin-left: 35%;margin-right: 35%;">
    <table >
<?php
    include "connect.php";
    $sql = "select clients.Name name, count(computers.ClientID) comps from clients inner join
    computers on clients.ClientID=computers.ClientID group by computers.ClientID order by clients.Name";
    $result = $mysqli->query($sql);

if ($result->num_rows > 0) {
    // column
    echo "<tr>";
        echo "<th>" . "Company Name" . "</th>";
        echo "<th>" . "Computers" . "</th>";
        echo "</tr>";
    // output each row from the database
    while($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row["name"].  "</td>";
        echo "<td>" . $row["comps"].  "</td>";
        echo "</tr>";
    }
} else {
    echo "0 results";
}

?>
    </table>
    </div>
```

Figure 2.1 Computers per Client Code

In this block of code, we generate one such report. We join the clients and computers tables across a common value of ClientID. We take the Company Name and a count of the computers and display that result. It does this for each client in the database and displays it like this.



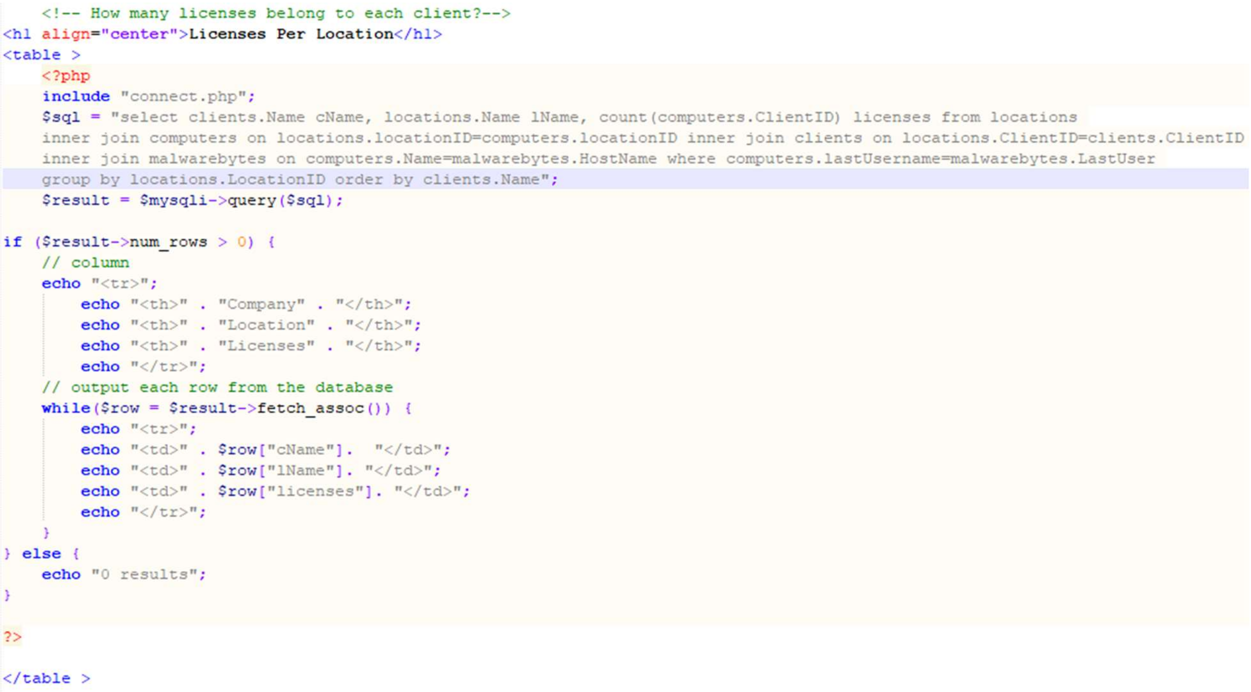
Computers Per Client

Company Name	Computers
ABC	32
ACF	5
ActiveLogistics/TransportationSolutions	3
Around the Clock	8
Art's TV	9
AtlanticMillwork	8
AWRestoration	6
Biller Associates	8
CCA	6
Certified Plus	8
Commercial Foundry/TCF	6
Common Vision	16
CommunitySocialIntegrat	14
Cougar Electronics	9
Cromwell Growers	23
Edge	63
Farm Family	15
Gillis Law Firm	17
Hamden Housing Authority	21
IMC	17
JJ Fishbein	5
JP Automotive	4

Figure 3 Computers per Client Report

This scrollable table allows our billing department to see those computer totals at a glance per client.

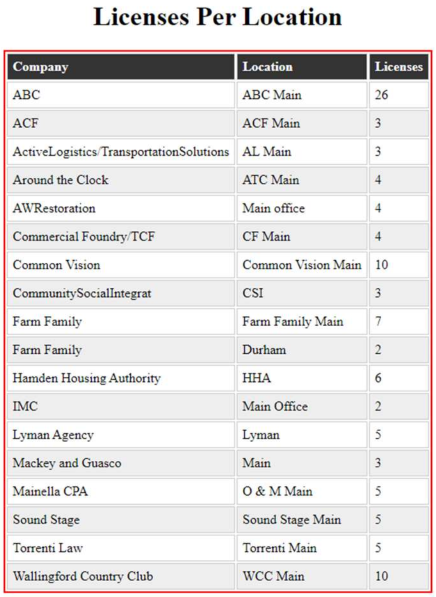
A more difficult query was getting a count of Malwarebytes licenses. The Clients and Computers tables came from the same source, ConnectWIse. The Malwarebytes table is coming from Malwarebytes so the data they collect is different and in a different format.



```
    <!-- How many licenses belong to each client?-->
<h1 align="center">Licenses Per Location</h1>
<table >
    <?php
    include "connect.php";
    $sql = "select clients.Name cName, locations.Name lName, count(computers.ClientID) licenses from locations
    inner join computers on locations.locationID=computers.locationID inner join clients on locations.ClientID=clients.ClientID
    inner join malwarebytes on computers.Name=malwarebytes.HostName where computers.lastUsername=malwarebytes.LastUser
    group by locations.LocationID order by clients.Name";
    $result = $mysqli->query($sql);

if ($result->num_rows > 0) {
    // column
    echo "<tr>";
        echo "<th>" . "Company" . "</th>";
        echo "<th>" . "Location" . "</th>";
        echo "<th>" . "Licenses" . "</th>";
        echo "</tr>";
    // output each row from the database
    while($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row["cName"].  "</td>";
        echo "<td>" . $row["lName"]. "</td>";
        echo "<td>" . $row["licenses"]. "</td>";
        echo "</tr>";
    }
} else {
    echo "0 results";
}

?>

</table >
```

Figure 4 Malwarebytes Licenses per Location Code

The query for this is a great deal more complex. For this, we connected the locations table to the computers table with the ClientID, the locations table to the clients table with the ClientID, and the computers table to the Malwarebytes table with the computer name and hostname where the LastUsername was the same as the LastUser. Thankfully, these fields were collected in the same way and proved effective at matching machines properly. Using a less variable field like a MAC address would be preferable if Malwarebytes collected such information. Beyond the complexity of the query, the simplicity of the report speaks for itself.



Licenses Per Location

Company	Location	Licenses
ABC	ABC Main	26
ACF	ACF Main	3
ActiveLogistics/TransportationSolutions	AL Main	3
Around the Clock	ATC Main	4
AWRestoration	Main office	4
Commercial Foundry/TCF	CF Main	4
Common Vision	Common Vision Main	10
CommunitySocialIntegrat	CSI	3
Farm Family	Farm Family Main	7
Farm Family	Durham	2
Hamden Housing Authority	HHA	6
IMC	Main Office	2
Lyman Agency	Lyman	5
Mackey and Guasco	Main	3
Mainella CPA	O & M Main	5
Sound Stage	Sound Stage Main	5
Torrenti Law	Torrenti Main	5
Wallingford Country Club	WCC Main	10

Figure 5 Malwarebytes Licenses Per Location Report

4. Host the Results

XAMPP is the software I used to house the database and troubleshoot the queries that would eventually be presented onto a webpage. XAMPP uses PHPMyAdmin to connect to the database and host PHP scripts onto the local host. Security and privacy were a concern for our clients, so we considered the idea of whether this information should be hosted locally or on the web. We decided for those of us who will be using this service, we would only be using it locally in the office so securing it as a local host was sufficient.

Within the PHP scripts there are calls to connect to the database, run the queries, and echo the results in specific formats. Each data table has a few reports and search functions encoded. I also created one “Monthlies” report site where all the most relevant reports are in one place.

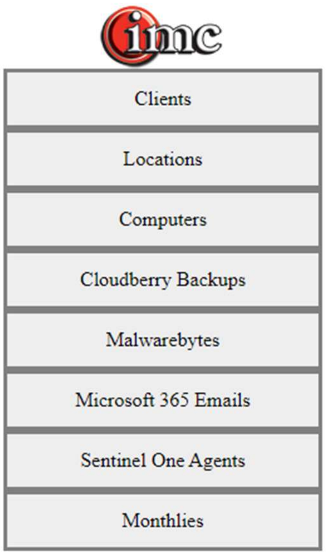


Figure 6 Front Page Access to All Reports

5. Polish the Website

Function is one thing, but form is another and is just as important. In its infancy, the results were presented as just lines of text. Cluttered, jagged, and formless as it was, I was able to use some HTML to place the results into tables. Now organized, I used some CSS and styling to give the site the same color scheme as the business. Some quality of life improvements came in with placing the tables in scrollable windows, adding a navigation bar so you would not have to go back to the start, and a back button designed as the IMC logo.

STATE OF IMPLEMENTATION

The site is fully functional and serves its intended purposes.

1. Monthly reports are properly presented both in their own pages and on the monthlies page.
2. Additional queries such as software versions and at-risk machines are also presenting on their relevant pages.
3. My coworkers have access to the reports and can use the information for the monthly totals.

TESTING AND EVALUATION

The arduous work of testing the queries is done. They have been evaluated with the collected totals from the relevant software. There is one query which presents all the data from all the reports in one table which needs work. Because of the data formatting issue implicit in the data collection methods of the software providers, this type of full report may or may not be possible.

LESSONS LEARNED AND REFLECTION

The value of this project and the conceptualization of the solution are simple enough to see and understand. The amount of time it will save my staff will be invaluable each month. If I was to do this again, I would take the data as a whole and filter the reports after the fact. This would make future imports and updates easier for myself. Writing the HTML and PHP were quite new to me. Fine tuning the SQL queries took quite a bit of research as to why they were failing when they did. I had to learn how the software views the folder structure so navigating pages was possible.

This project may have taken a few months to complete, but the education that has taken place to make this project even possible for me has been 4 years in the making. Each aspect of this project is built on the knowledge passed down to me by my professors. My first exposure to coding was in college writing a “Hello World!” program as all computer science students do. From there, learning about data structures, python coding, HTML, CSS, SQL, those skills built from separate projects. Understanding the underlying fundamentals and logic of syntax came through years of practice while in school. SQL queries such as these are complicated to say the least. It requires an understanding of how tables communicate to each other, how they are joined, and what the tables themselves and have in what forms.

I used sites like stack overflow for hints to solving some coding problems. When the syntax was a bit off, or the type or join I was using was not keeping the information I wanted, I went searching for answers online. Things like the styling were readily available on w3schools. Reading through the documentation I learned new things that were possible. When I was at the point of applying polish to the website, I changed things like the coloring, centering the reports, and most importantly, putting the reports in sub windows. I painstakingly went through the formatting of the style to make sure it presented cleanly on a full resolution view, and on smaller viewing perspectives as well. I am sure there are templates that could accomplish similar tasks, but I have always been more engaged in learning the code from the bottom up. Though the end product may look more simplistic, I know I built it from the ground up. There have been projects in the past which heavily depended on the work and templates of others. I find it a better expression of my level of mastery to write the code myself. I understand a certain competency that is achieved by taking templates and APIs and adapting them to your own work. For example, getting live updates of the databases may be possible through Malwarebytes' API. But major changes like that are better saved for future considerations.

VERSION 2

Time is always the enemy in projects like these. My future considerations for this project would be automating the data upload and update process. These numbers are only referenced once a month for the monthlies, so a constant update does not matter so much. There are other reports in here that are more service ticket based. Places where we can go in, see an issue as an IT team and respond to it. It would help for these tables to be updated automatically when there is a security risk or out of date software to respond to.

I would also like to have implemented better queries that compared information a bit better. There are some things in the final report that fail because of the name scheme used by the different software providers. For example, the Computers table does keep track of the MAC address, as does the Sentinel One table. It would be perfect to compare these two values from each database since they are, by nature, unique to the device compared to any other device in the world. Unfortunately, the format of the MAC address made them un-comparable. One table had them separated with colons whereas the other table had them separated with hyphens. This meant when they were compared for likeness, they would appear different to the query given the character differences. Lastly, I would include printable reports so the reports could be shared over emails or in person.