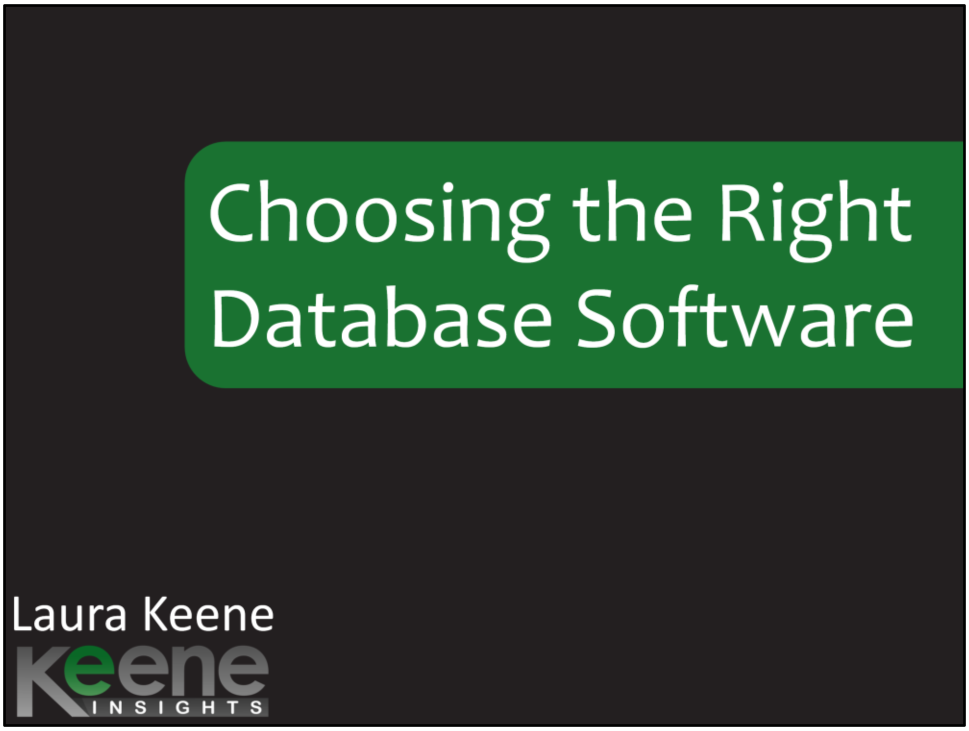
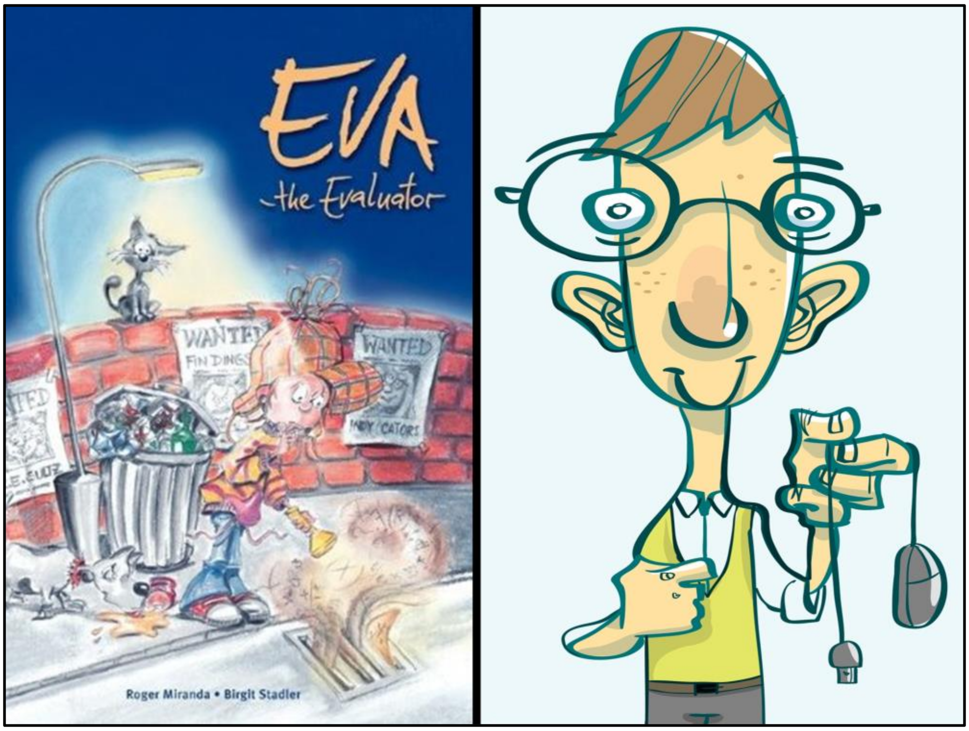


Choosing the Right Database Software

Laura Keene

My name is Laura Keene. I own and operate Keene Insights where, in addition to being an evaluator, I'm also a database developer.



I’ve had one foot in both of these worlds for a long time and from that perspective I’m going to walk you through choosing the right database software.

Two things to note. First, I’ll be talking specifically about larger relational databases like Access, Filemaker Pro, Salesforce and ETO. I won’t be covering data collection software like Survey Monkey or data analysis software like SPSS and SAS.

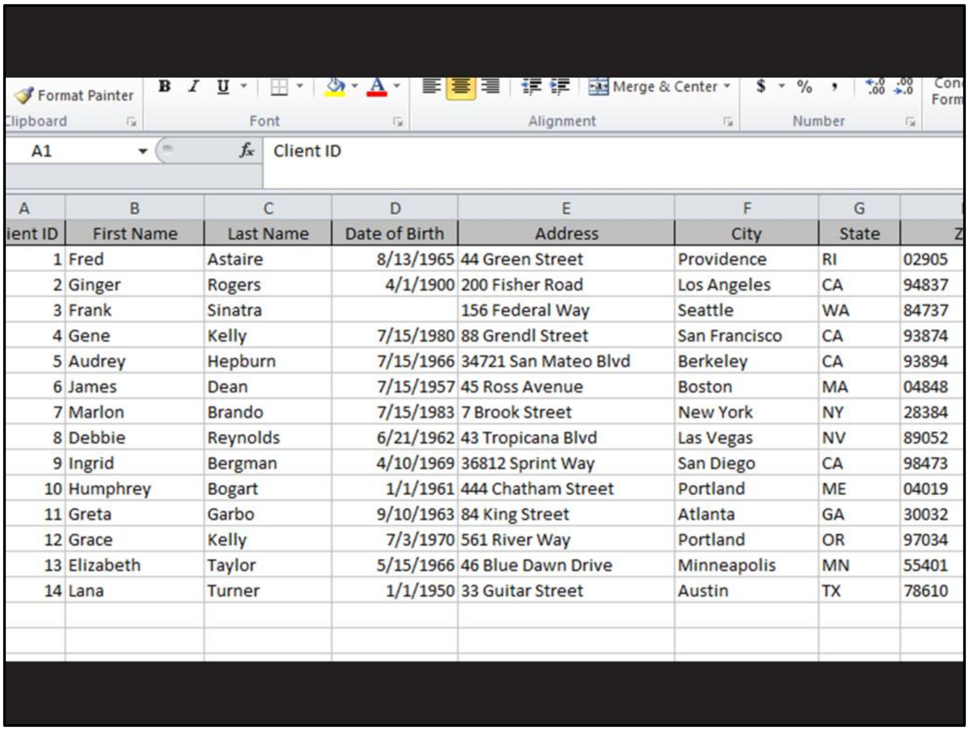
Second, because every context is different and the options are growing and evolving constantly, I can’t give you the right answer, I can’t say, for example, always get ETO. Instead, I’m going to give you the right questions. With the right questions, you can find software that’s the best fit for your needs and your situation.

Agenda

I’m going to start by laying some groundwork: what is a relational database? How has the software landscape changed? And why does this matter to evaluators? Then, we’ll dive into those questions to ask when choosing software: What purpose does it serve? How much will it cost? What software program is the right fit? Who will build and manage it? And, finally, where will it live? For these WHAT, WHO, WHERE questions, we’ll talk about each of the major options and what factors to consider when picking one.

Relational Database Who What What?

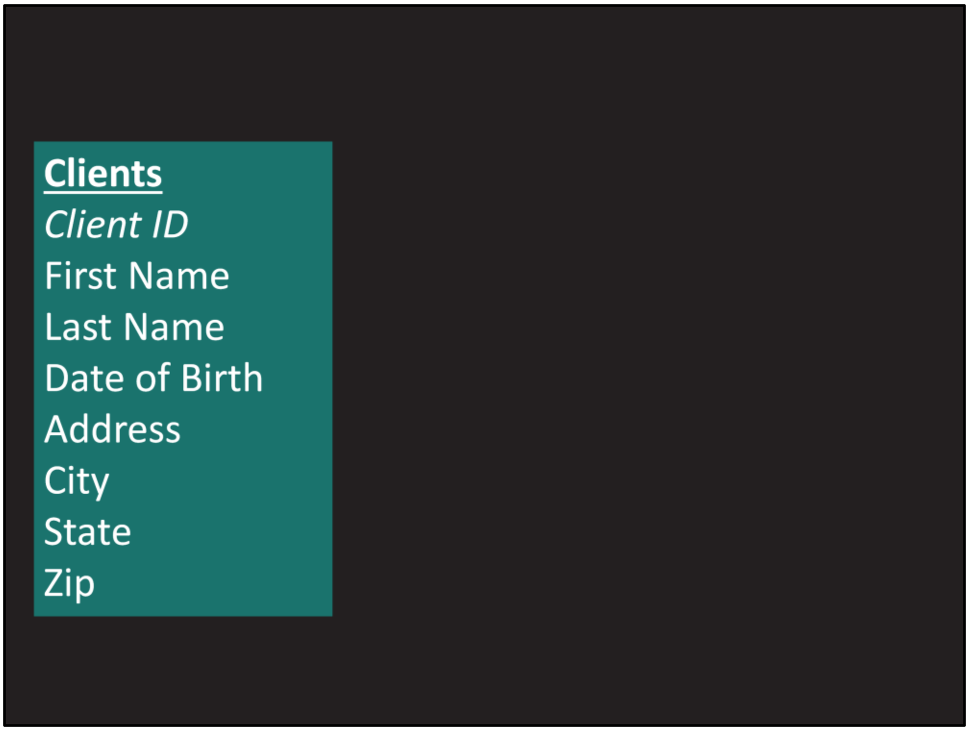
What do I mean when I say relational database?



Client ID

A	B	C	D	E	F	G	
ient ID	First Name	Last Name	Date of Birth	Address	City	State	Z
1	Fred	Astaire	8/13/1965	44 Green Street	Providence	RI	02905
2	Ginger	Rogers	4/1/1900	200 Fisher Road	Los Angeles	CA	94837
3	Frank	Sinatra		156 Federal Way	Seattle	WA	84737
4	Gene	Kelly	7/15/1980	88 Grendl Street	San Francisco	CA	93874
5	Audrey	Hepburn	7/15/1966	34721 San Mateo Blvd	Berkeley	CA	93894
6	James	Dean	7/15/1957	45 Ross Avenue	Boston	MA	04848
7	Marlon	Brando	7/15/1983	7 Brook Street	New York	NY	28384
8	Debbie	Reynolds	6/21/1962	43 Tropicana Blvd	Las Vegas	NV	89052
9	Ingrid	Bergman	4/10/1969	36812 Sprint Way	San Diego	CA	98473
10	Humphrey	Bogart	1/1/1961	444 Chatham Street	Portland	ME	04019
11	Greta	Garbo	9/10/1963	84 King Street	Atlanta	GA	30032
12	Grace	Kelly	7/3/1970	561 River Way	Portland	OR	97034
13	Elizabeth	Taylor	5/15/1966	46 Blue Dawn Drive	Minneapolis	MN	55401
14	Lana	Turner	1/1/1950	33 Guitar Street	Austin	TX	78610

Most people are familiar with spreadsheets and other types of flat files where each column represents a variable or field and each row represents one of whatever it is you’re measuring, clients, trainings, etc. like this example. Take that basic concept and put it on steroids and you have a relational database.



Clients

Client ID

First Name

Last Name

Date of Birth

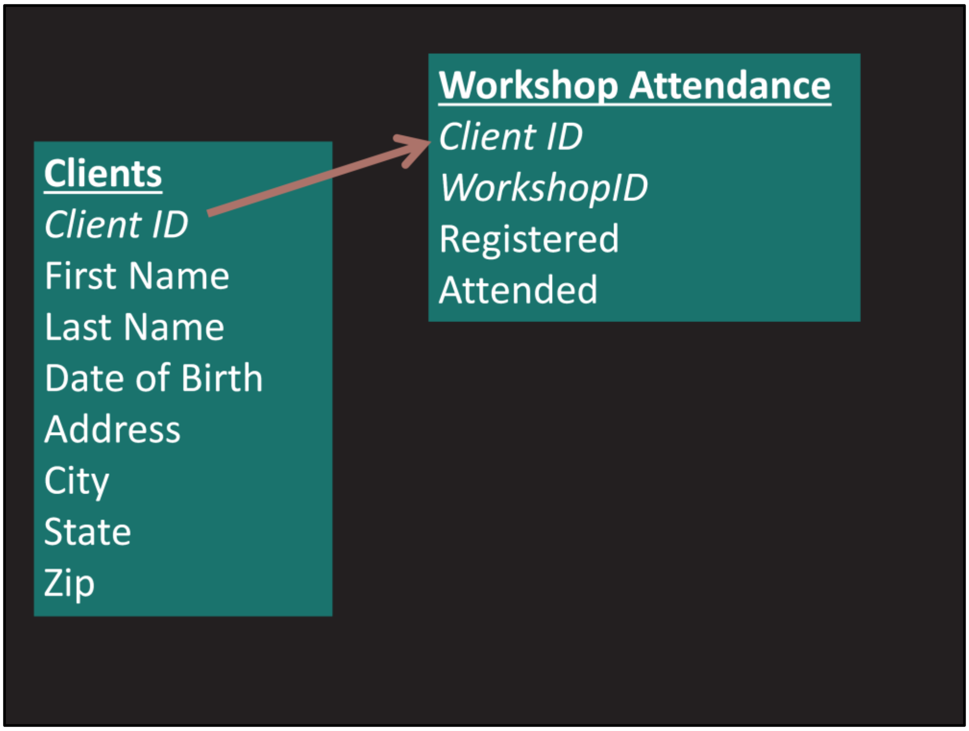
Address

City

State

Zip

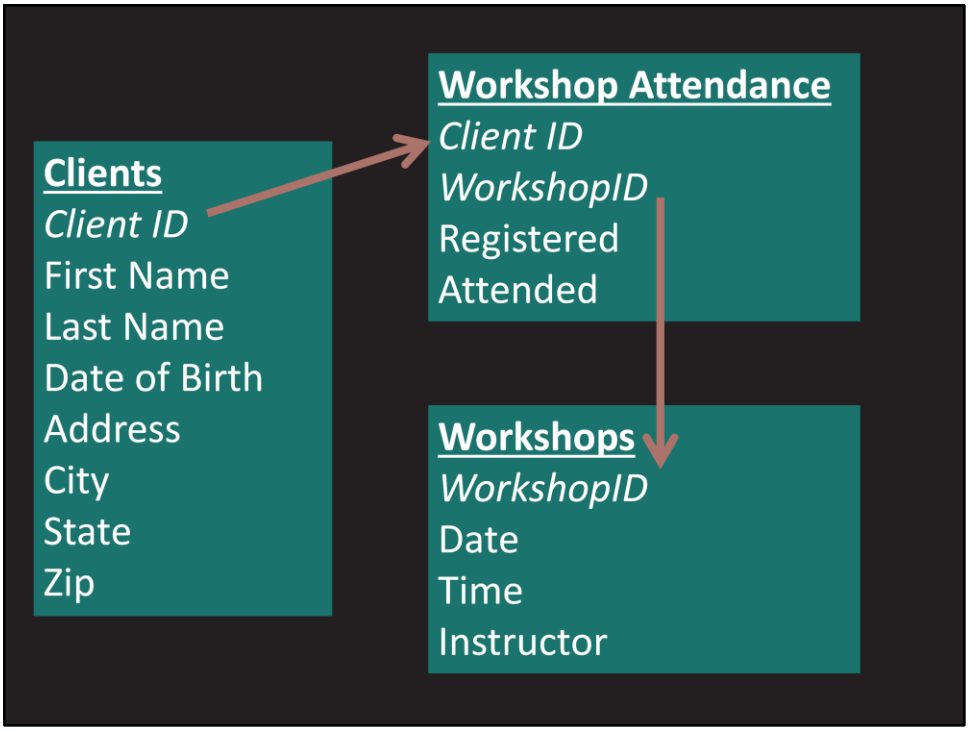
Our basic spreadsheet becomes one table (this table would still look like the spreadsheet, I'm just using this box as shorthand)…



Workshop Attendance
Client ID
WorkshopID
Registered
Attended

Clients
Client ID
First Name
Last Name
Date of Birth
Address
City
State
Zip

…and our client data can now connect to another table, let’s say attendance at workshops…



Clients
Client ID
First Name
Last Name
Date of Birth
Address
City
State
Zip

Workshop Attendance
Client ID
WorkshopID
Registered
Attended

Workshops
WorkshopID
Date
Time
Instructor

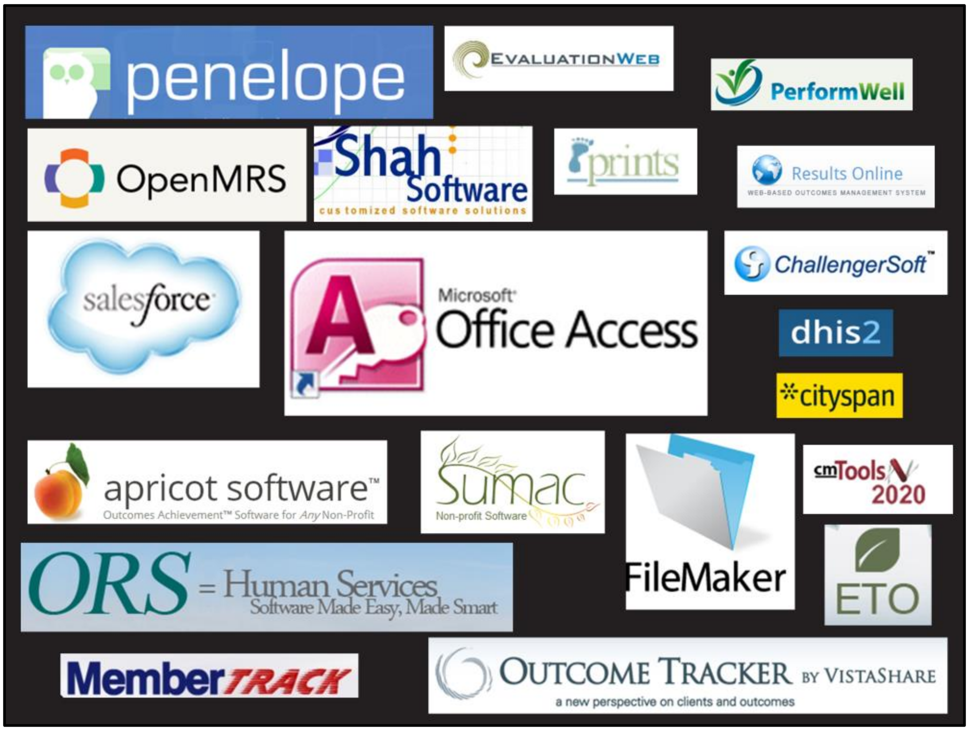
…and then that table can connect to a table that has details about the workshops. With multiple, related tables (a relational database), you have the ability to organize, manage and manipulate much more complex data.

For the most part, people turn to relational databases when they’ve outgrown flat files, like Excel. I’m sure many of you have seen it, the attack of the Excel spreadsheets, growing in size and number until they take over. Along with more robust data management, most relational database software also provides a nice user interface for data entry plus more complex querying and automated reporting features.

The Database Landscape

Now, let’s look at relational databases in the evaluation context: By this I mean databases used by evaluators, nonprofits, foundations, government agencies, educational organizations for managing clients, service delivery data, survey data, secondary data and so on.

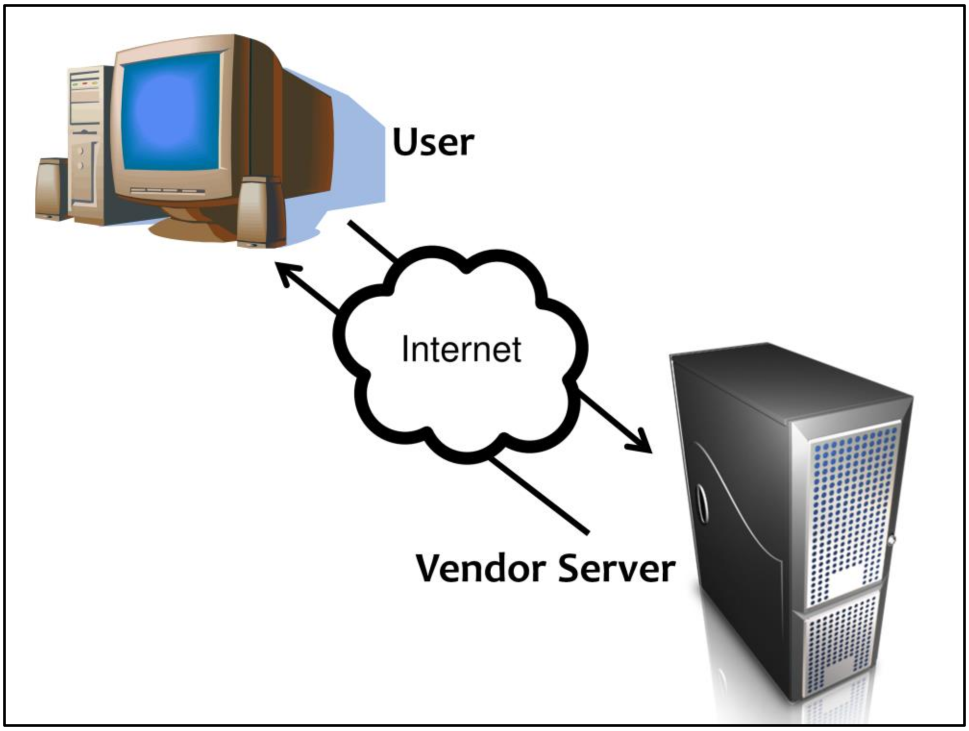
By and large, when I first stepped into the evaluation world about ten years ago there was really only one option for this kind of database that was affordable for the average evaluation budget and provided the needed tools: a custom Microsoft Access database. And there were maybe a few places using Filemaker Pro, which is Apple’s Access equivalent.



Now, the options look more like this and that doesn't even cover all of the options that are out there.

Software as a Service (Saas)

So what’s changed? Most of these new options are web-based, off-the-shelf options. They’re part of a boom in what’s called Software as a Service (SaaS) or hosted software.



With SaaS, the software package and the data are stored on the vendor’s servers (a server is basically just a powerful computer that other computers can talk to). Staff accesses the software and data over the internet, also known as the cloud these days.

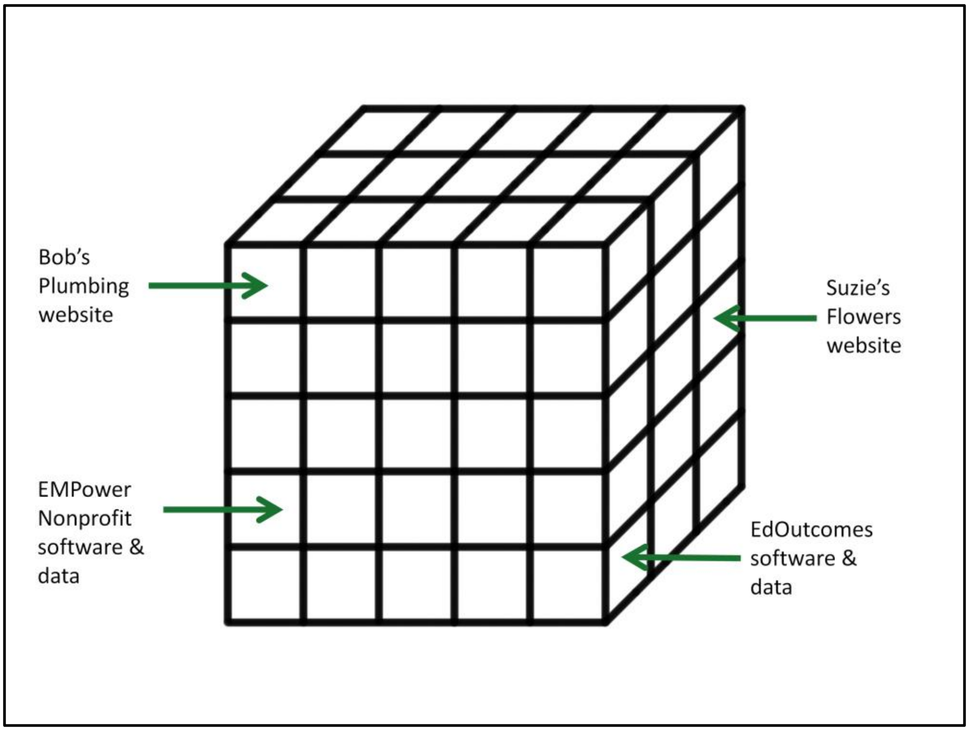
penelope
EvaluationWeb
PerformWell
OpenMRS
Shah Software
customized software solutions
prints
Results Online
WEB-BASED OUTCOMES MANAGEMENT SYSTEM
salesforce
Microsoft Office Access
ChallengerSoft
dhis2
cityspan
apricot software
Outcomes Achievement™ Software for Any Non-Profit
Sumac
Non-profit Software
FileMaker
cmTools 2020
ORS = Human Services
Software Made Easy, Made Smart
ETO
MemberTRACK
Outcome Tracker by VistaShare
a new perspective on clients and outcomes

Facebook, Twitter, Survey Monkey, WebEx and many of the options here are Software as a Service. This SaaS boom is primarily the result of better and cheaper technology. Internet speeds have really ramped up, for example.

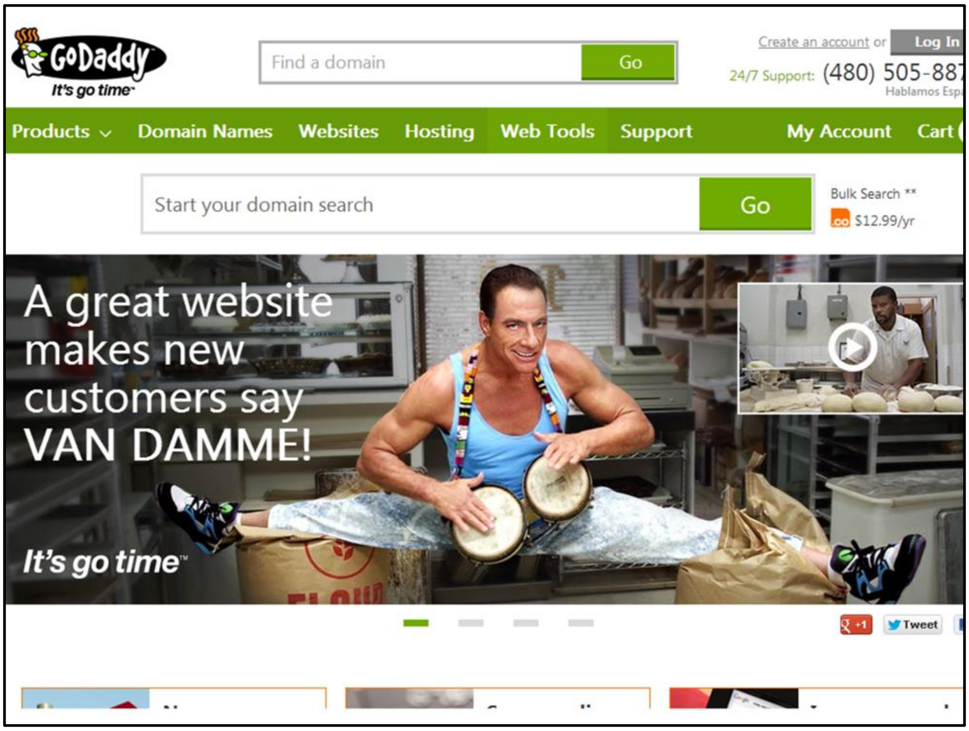
The tools used to build these databases have become more sophisticated.

Storage space is getting cheaper and cheaper. These flash drives are a perfect example of this. You can buy a 64GB flash drive for less than 40 bucks. It feels like it was just yesterday that a 1GB flash drive was the same price. This change makes it much less expensive for vendors to host all of this software and data or pay someone to host it, which bring us to one other big shift…

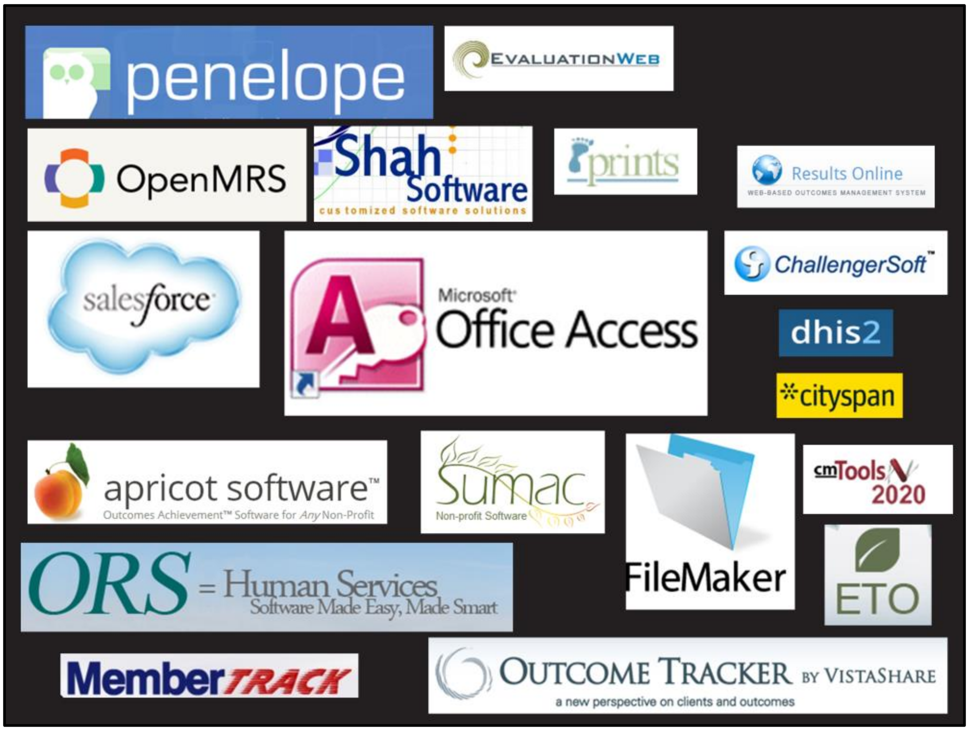
It’s become a lot easier to slice and dice a big server. This dramatically lowers costs for companies that want to build and sell Software as a Service and has allowed a lot of smaller vendors to enter the market. Here’s what I mean by that…these are servers, as I mentioned, they’re essentially more powerful computers that other computers can talk to. Years ago, a big server could only function as one big computer and it was only available to companies who could buy the whole thing and make use of it.



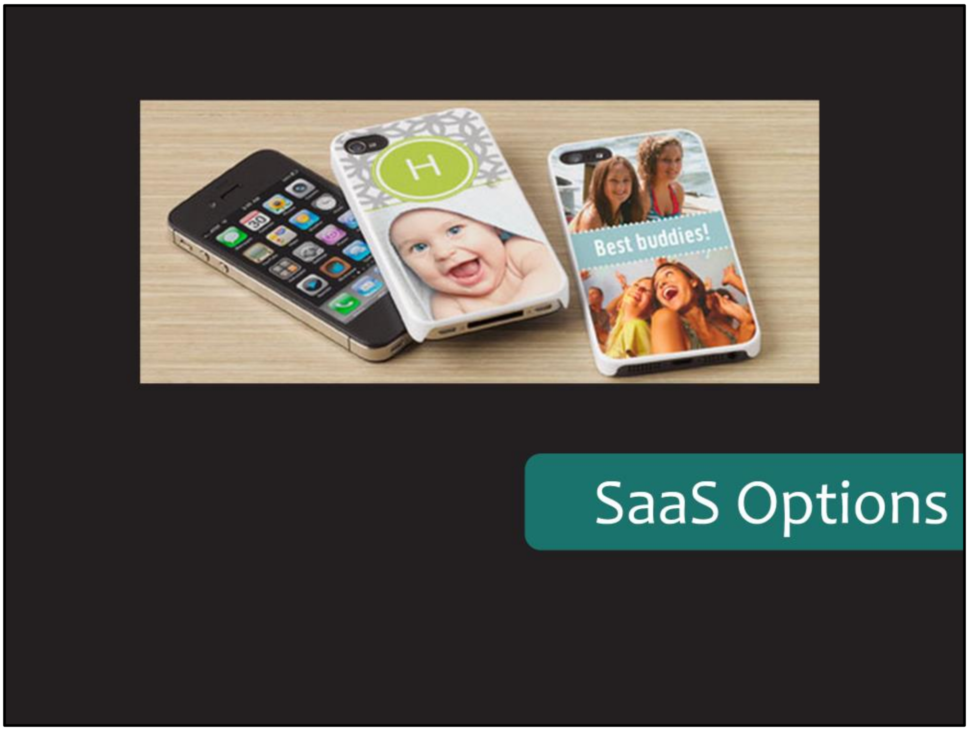
Now, that server can be divided up and function as a bunch of little computers. The whole thing is much more cost effective. One company can house a bunch of servers, handle maintenance, backups, and upgrades and then have a bunch of other companies pay for space on them, and that space is accessed via the internet. Bob's Plumbing website is stored on one part of the server, and Suzie's Flowers website is stored in another part, and so on.



This change has also led to a proliferation of web hosting companies, like GoDaddy, FatCow, and DreamHost, which let <u>you</u> and <u>me</u> buy space on their giant servers for as little as 100 bucks a year and use that space to host our own websites and, if needed, store data. This makes custom websites one of the options when choosing the right software, which will talk more about.



Going back to our context specifically and our SaaS options. What exactly does this type of software look like?

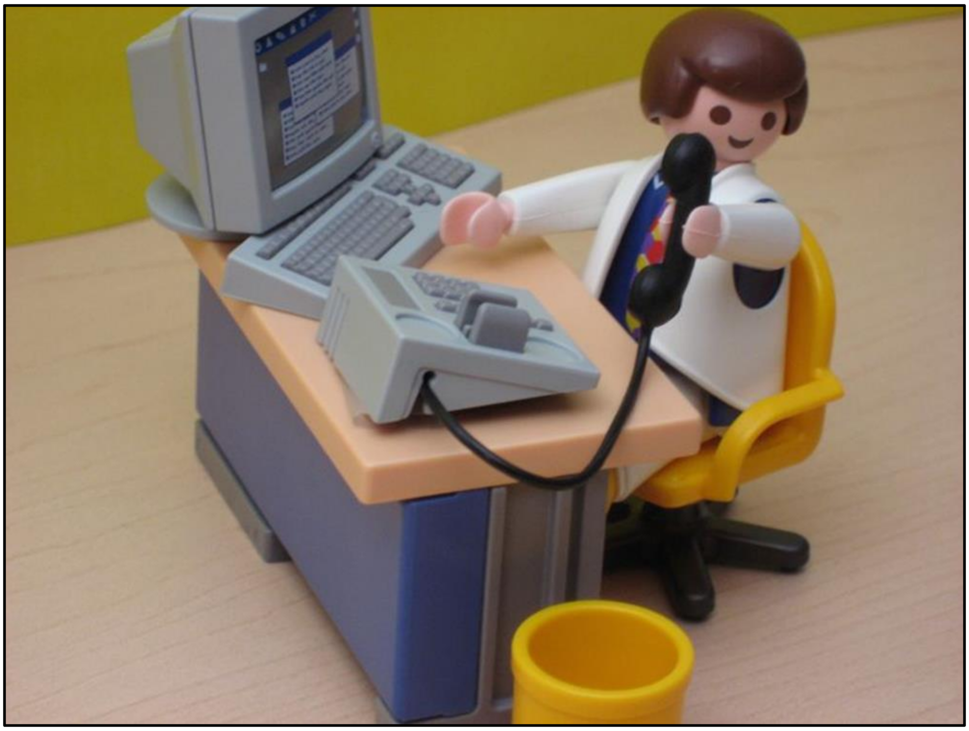


SaaS Options

They are pre-packaged, off-the-shelf options to some extent, the underlying table structure is usually the same across customers. But, most have an interface that can be customized quite a bit. For example, with Apricot’s case management software, each organization can decide which variables they want to collect for each of their clients and for each visit and customize their reports. It’s kind of like being able to create custom cases for your iPhone, the exterior looks different and it’s still an iPhone underneath, but the customizability is much more robust and much more meaningful.

Why do we care?

How is this relevant to evaluators? It generally pop-ups in one of two ways.

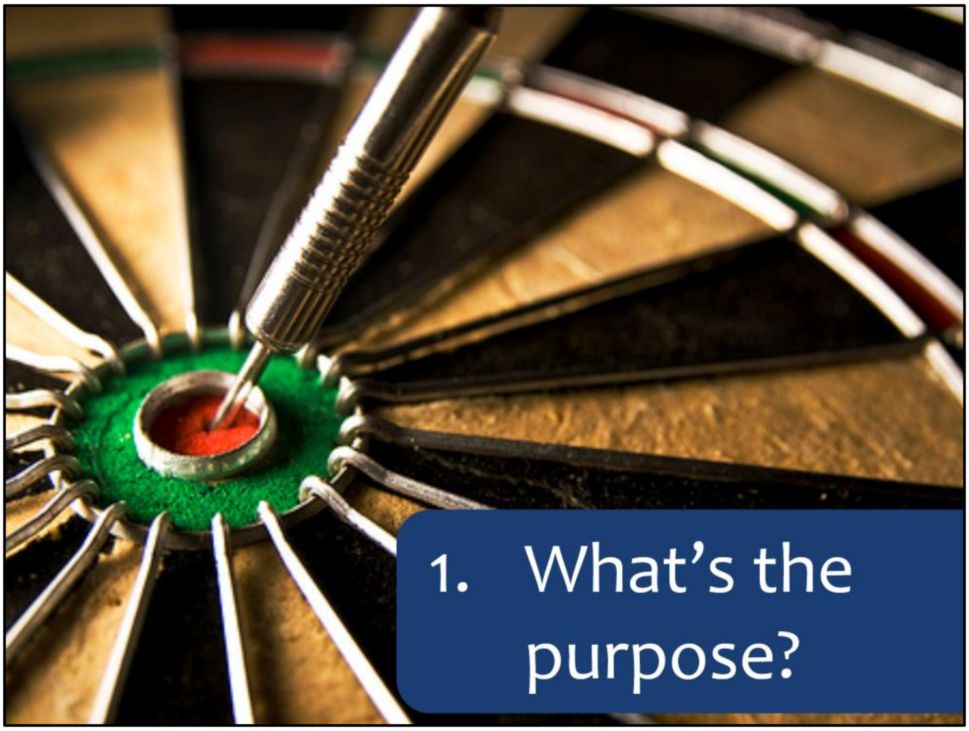


First, the obvious one, whether we’re an internal evaluator or an external evaluator we often need relational databases. We have our own instances of outgrowing Excel spreadsheets, needing nicer data entry interfaces, or wanting to save time and money by automating reporting. All of which leads to having to make some software decisions.

Second, because the programs we work with also need this kind of relational database software, evaluators often end up playing a key role as an interpreter or bridge between software developers and program staff. In addition to having helpful expertise to contribute to the process, we usually need to know what’s going to be collected, how, and eventually be able get our hands on that data so it makes sense to get involved. I also want to mention that the vendors that are part of this SaaS boom are not usually marketing to us as evaluators, they’re marketing to the nonprofits, foundations, government agencies, and educational organizations that we work with.

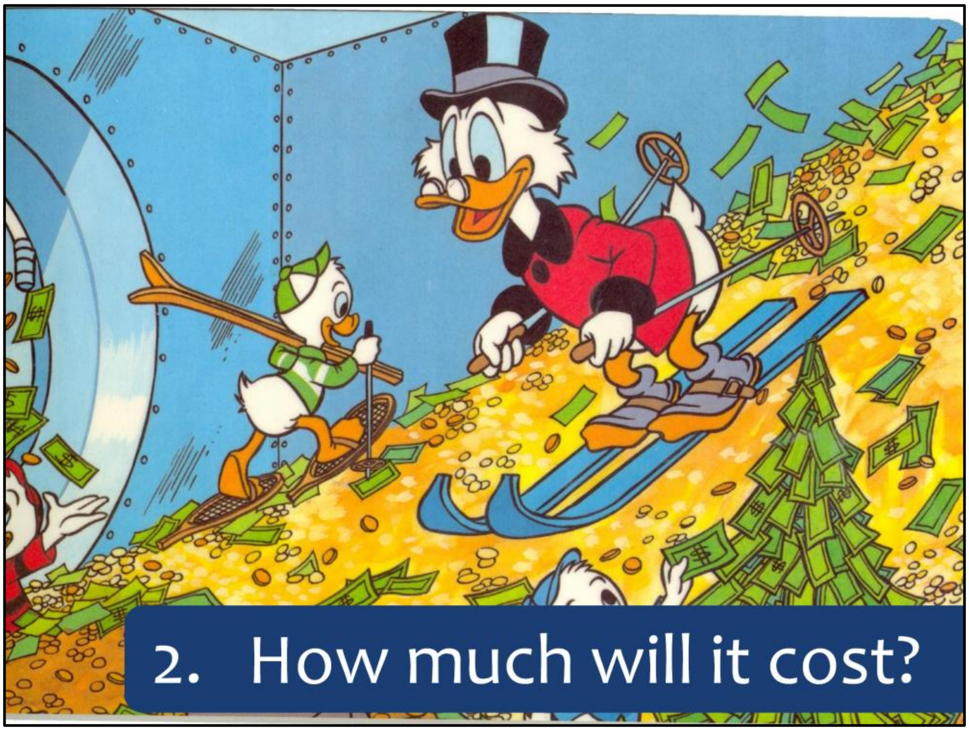
The Right Questions!

Onto the right questions to ask when choosing database software.



1. What’s the purpose?

Question one, what’s the purpose of the database? Who’s going to use it? To do what? How will the data be used? Try to hammer as much of this out as possible BEFORE shopping for a software program. I see a lot of organizations let their software decision drive their data collection decisions, whatever is in the software is what they decide to collect. Typically, the software isn’t up to that task (especially in evaluation contexts) and inevitably, somewhere down the line, they realize it’s not the right fit. It’s best to make the data decisions, and then get the software that’s the right match.

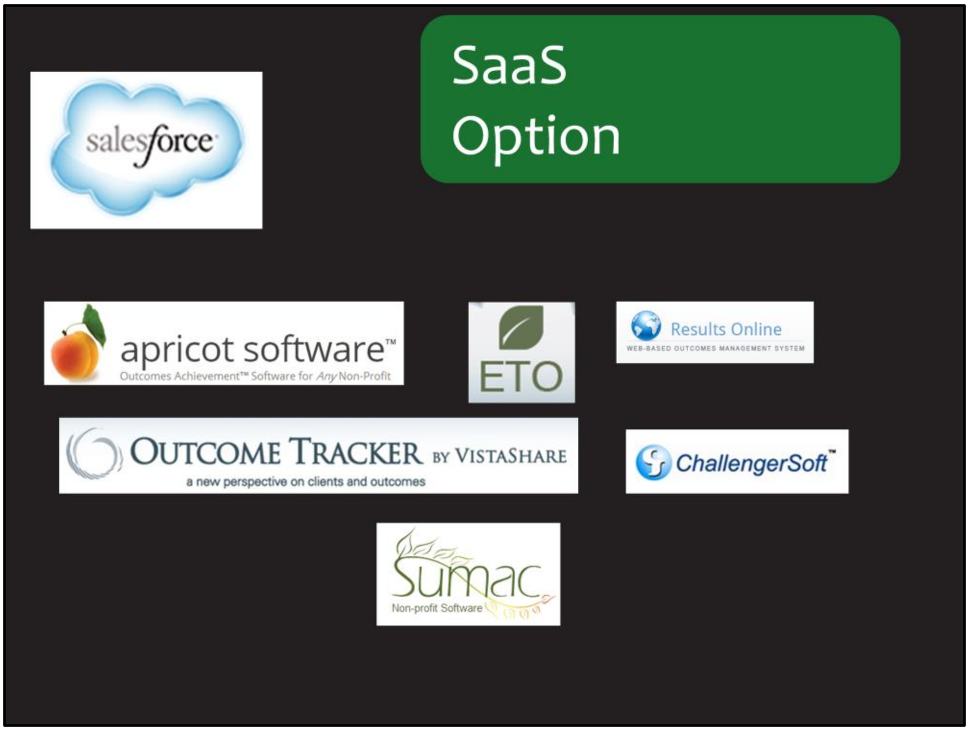


2. How much will it cost?

Question two, how much will it cost? This is a really big question so I’ve incorporated cost considerations into the remaining WHAT, WHO, WHERE sections that follow. Generally, you’re looking at the cost of paying a developer or vendor, the software package itself, and for maintenance, upgrades, and, potentially, web hosting. For SaaS and other pre-packaged options this is all incorporated into an initial setup cost and then a set annual fee. You’ll also want to factor in internal costs (e.g., staff time for planning, training, data entry and management).

3. WHAT software program will you use?

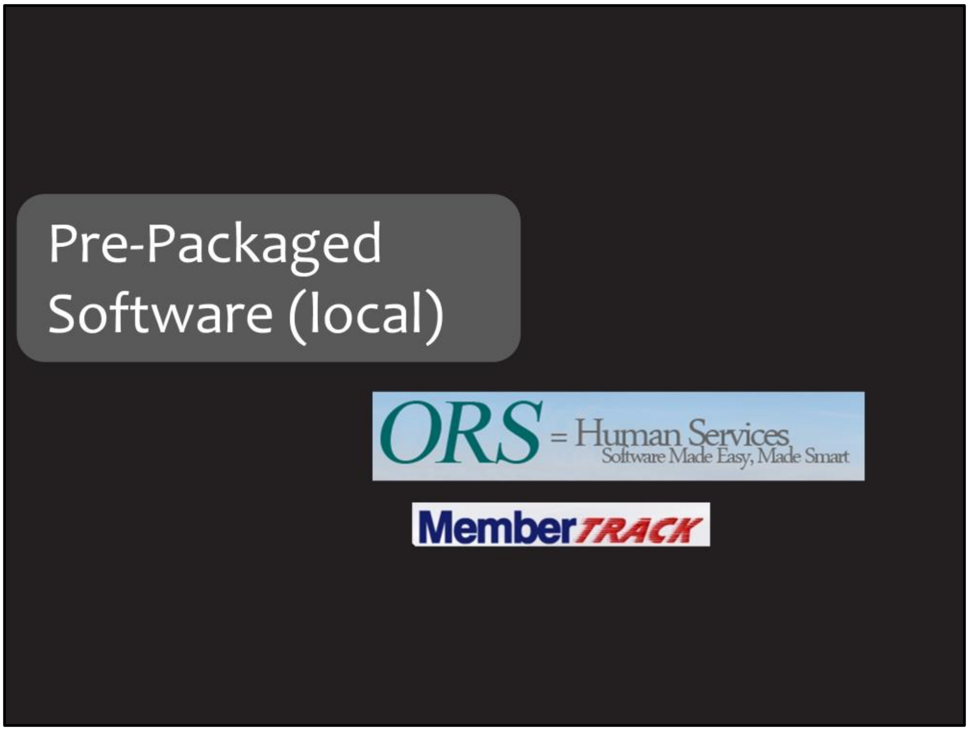
We’ve got these three WHAT, WHO, WHERE questions: I’ll start with the biggest and most complicated, the WHAT question: What software program will you use? The options are endless so we’re going look at four general categories…



SaaS Option

salesforce
apricot software™
Outcomes Achievement™ Software for Any Non-Profit
ETO
Results Online
WEB-BASED OUTCOMES MANAGEMENT SYSTEM
OUTCOME TRACKER BY VISTASHARE
a new perspective on clients and outcomes
ChallengerSoft™
Sumac
Non-profit Software

Option 1 is SaaS. As we've covered, with SaaS, the software package and the data are stored on the vendor's servers and staff accesses it over the internet. Apricot, Efforts to Outcomes, Salesforce and many, many others are Software as a Service.



Pre-Packaged Software (local)

ORS = Human Services
Software Made Easy, Made Smart
MemberTRACK

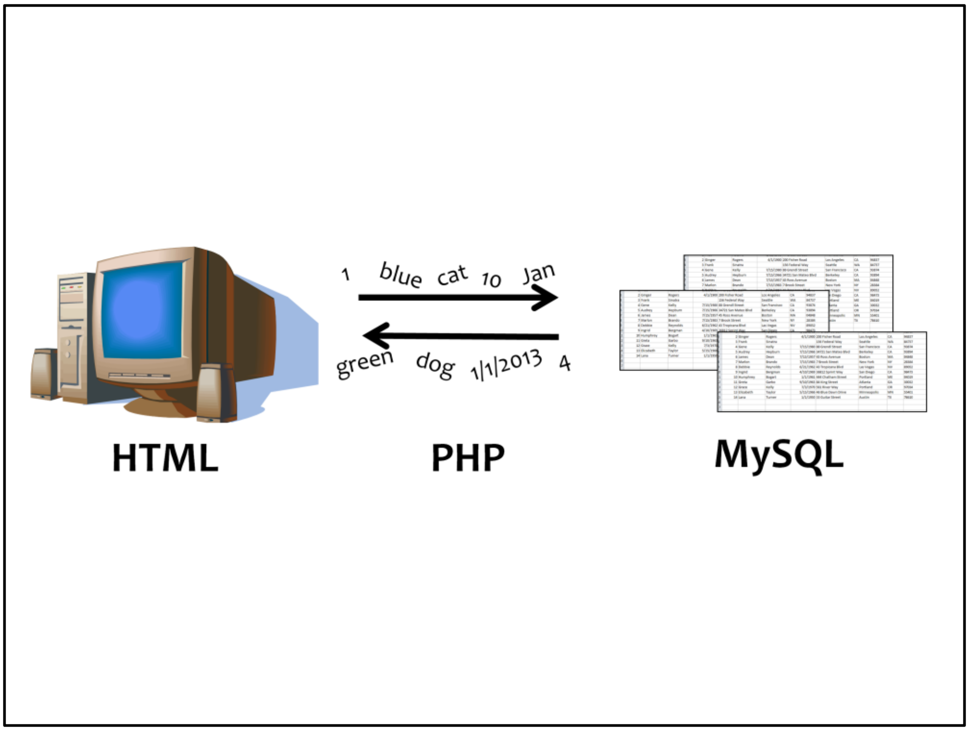
Option two is pre-packaged software that is not internet-based, the software and data are housed at your organization. I think these options are dwindling because it’s more cost effective for companies to go the SaaS route.

Database Builder Software

Option three, is to use a program that is designed for building databases. For these, it's best to also plan on hiring a developer, someone to build it, we'll talk more about that in the WHO section. These options typically have a user interface for designing tables, queries, forms and reports and allow developers to get under the hood and code for more customizability. Examples include well known options like Access and Filemaker. There are also Open Source options, which is free software that is developed and maintained by the community, like DHIS and OpenMIS which are used in international health settings. And, there are some SaaS vendors that offer an environment for people to build in, Salesforce is a good example of this.

Coding
(from scratch)

And, option four, is to use code to build the database from scratch. Again, this would require hiring a developer.

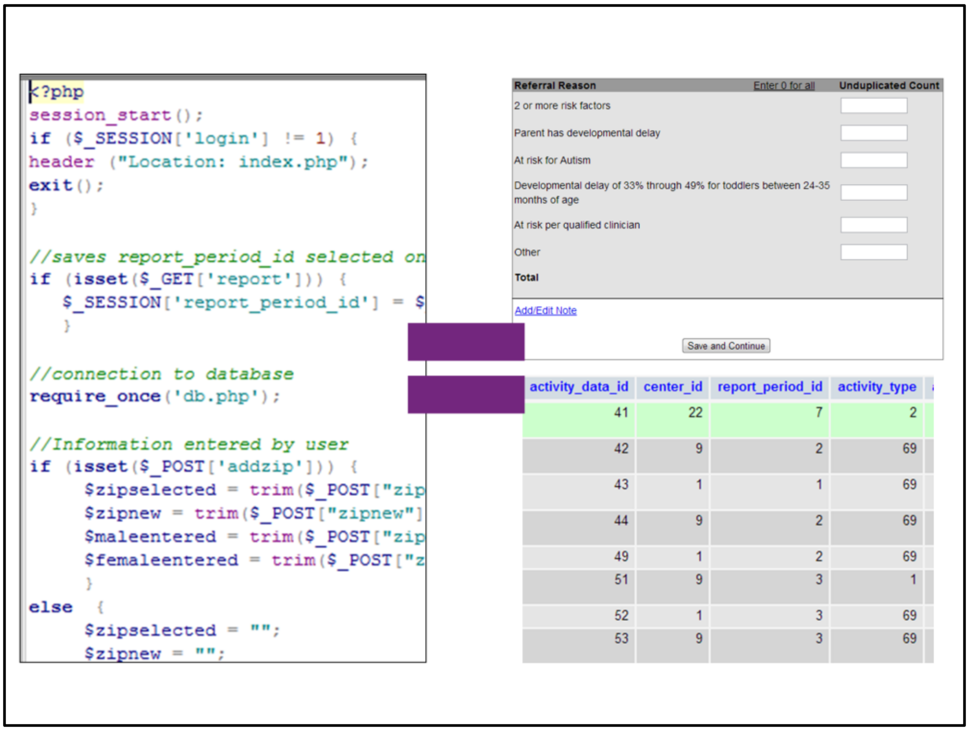


HTML

PHP

MySQL

PHP/MySQL is commonly used for building web-based databases and is a good example to give you an idea of how this works. Under the hood, it's all coding, PHP and MySQL are programming languages that are used to build the database. HTML, which is another language, is used to build the web interface, data are stored in tables using MySQL and PHP moves the data back and forth, saving what the user enters on the web page to the tables or displaying what the user wants to see from the tables on the web page.



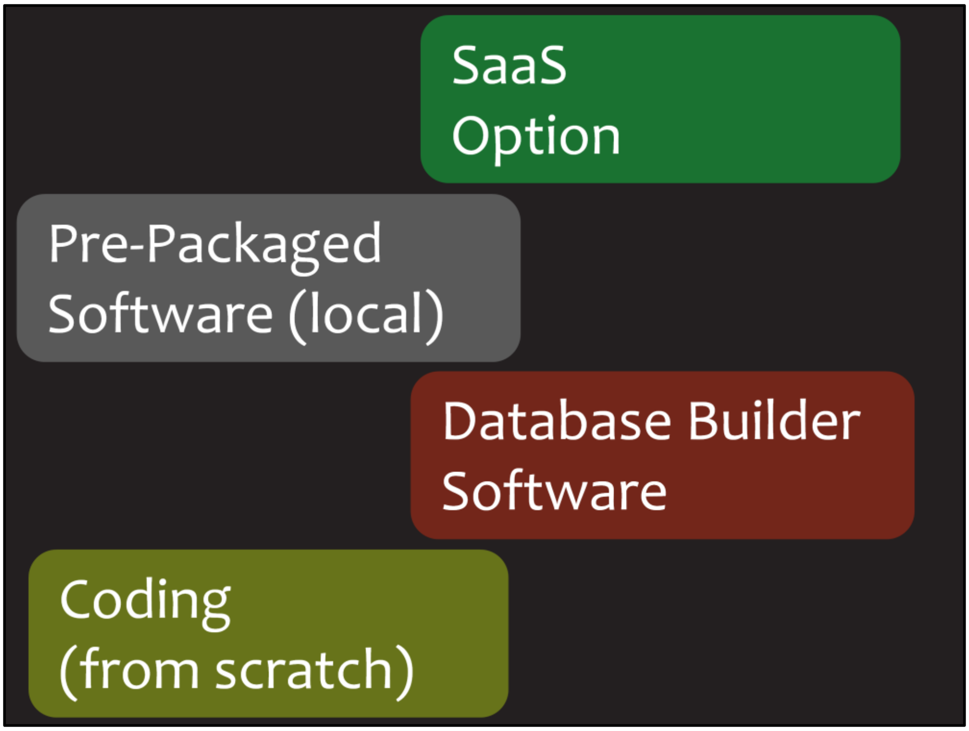
```
<?php
session_start();
if ($_SESSION['login'] != 1) {
header ("Location: index.php");
exit();
}

//saves report_period_id selected on
if (isset($_GET['report'])) {
   $_SESSION['report_period_id'] = $
   }

//connection to database
require_once('db.php');

//Information entered by user
if (isset($_POST['addzip'])) {
     $zipselected = trim($_POST["zip
     $zipnew = trim($_POST["zipnew"]
     $maleentered = trim($_POST["zip
     $femaleentered = trim($_POST["z
     }
else  {
     $zipselected = "";
     $zipnew = "";
```

Referral Reason	Enter 0 for all	Unduplicated Count
2 or more risk factors		
Parent has developmental delay		
At risk for Autism		
Developmental delay of 33% through 49% for toddlers between 24-35 months of age		
At risk per qualified clinician		
Other		
Total		

Add/Edit Note

activity_data_id	center_id	report_period_id	activity_type
41	22	7	2
42	9	2	69
43	1	1	69
44	9	2	69
49	1	2	69
51	9	3	1
52	1	3	69
53	9	3	69

All that coding creates a fully functional database. There are other programming languages that work similarly.



SaaS Option

Pre-Packaged Software (local)

Database Builder Software

Coding (from scratch)

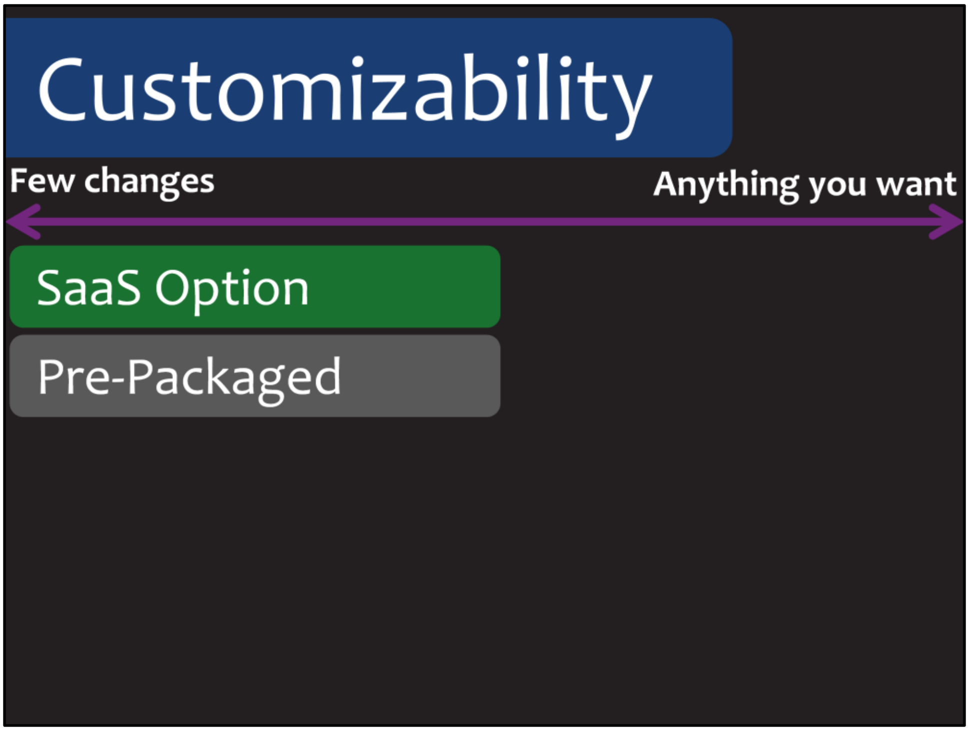
When choosing one of these options, what do you think some of the considerations might be?

Customizability

Few changes

Anything you want

Let’s start with customizability. On a scale of few changes are allowed to you can build anything you can dream up, there is a lot of variability.



Customizability

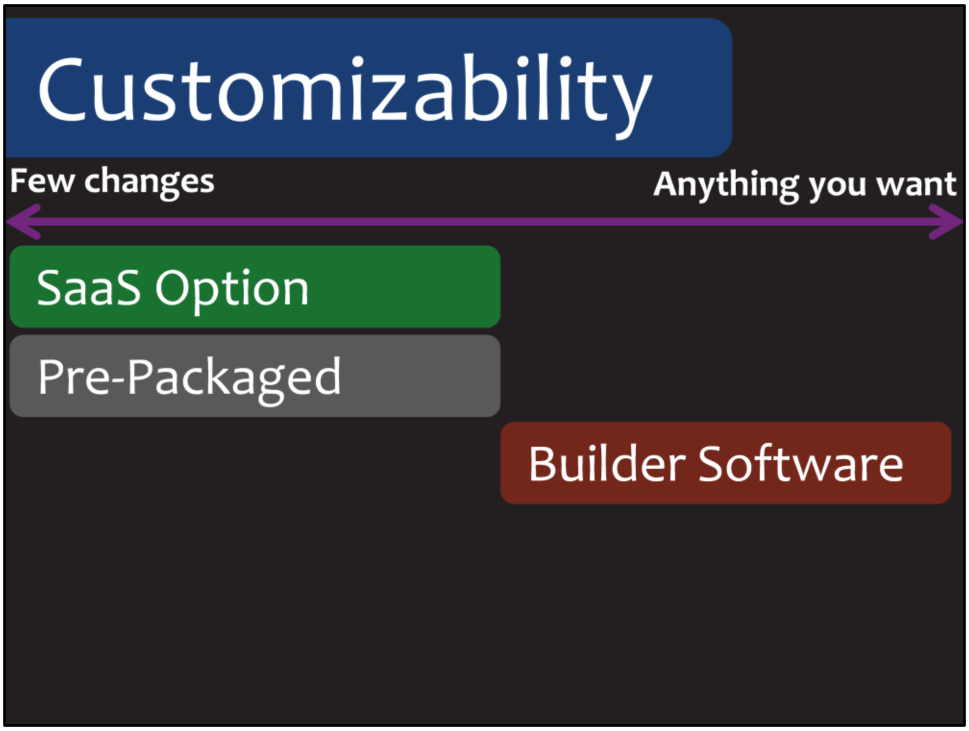
Few changes

Anything you want

SaaS Option

Pre-Packaged

SaaS and pre-packaged options do offer some customizability but the underlying table structure is set. If you go this route you want to focus on finding the right fit. PatronManager, for example, is specifically designed for arts organizations that want to manage events and ticket sales. DenariOnline works well for missionary and child sponsorship programs.



Customizability

Few changes

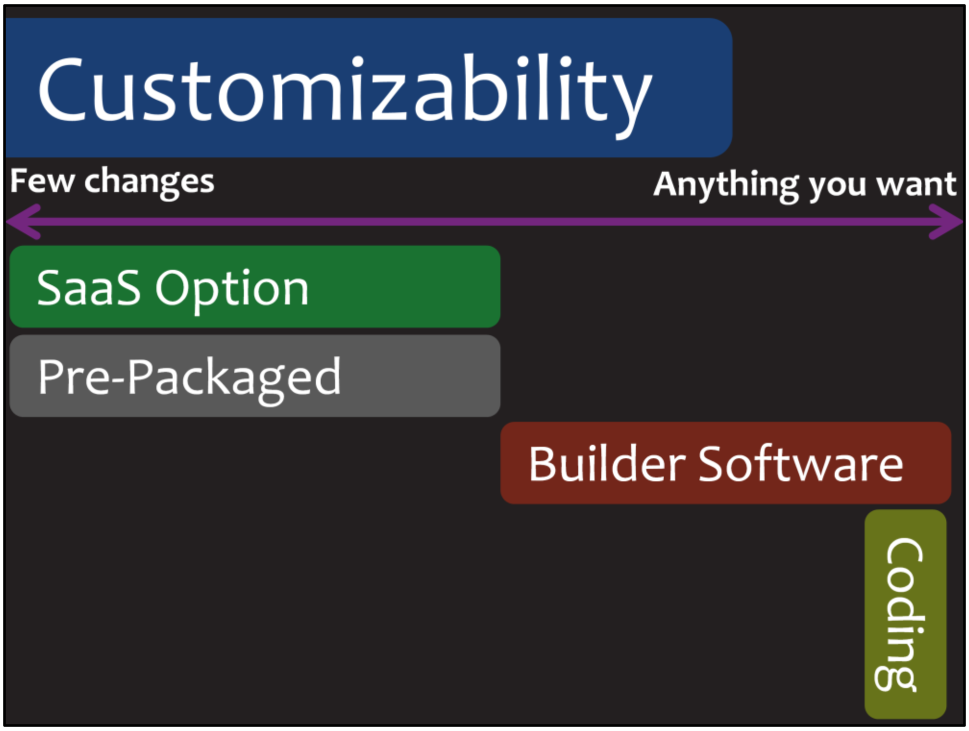
Anything you want

SaaS Option

Pre-Packaged

Builder Software

Database builder software is much more flexible but there’s still a pretty wide range. With Access and FileMaker you can build anything you want, with some open source options its more restrictive.



Customizability

Few changes

Anything you want

SaaS Option

Pre-Packaged

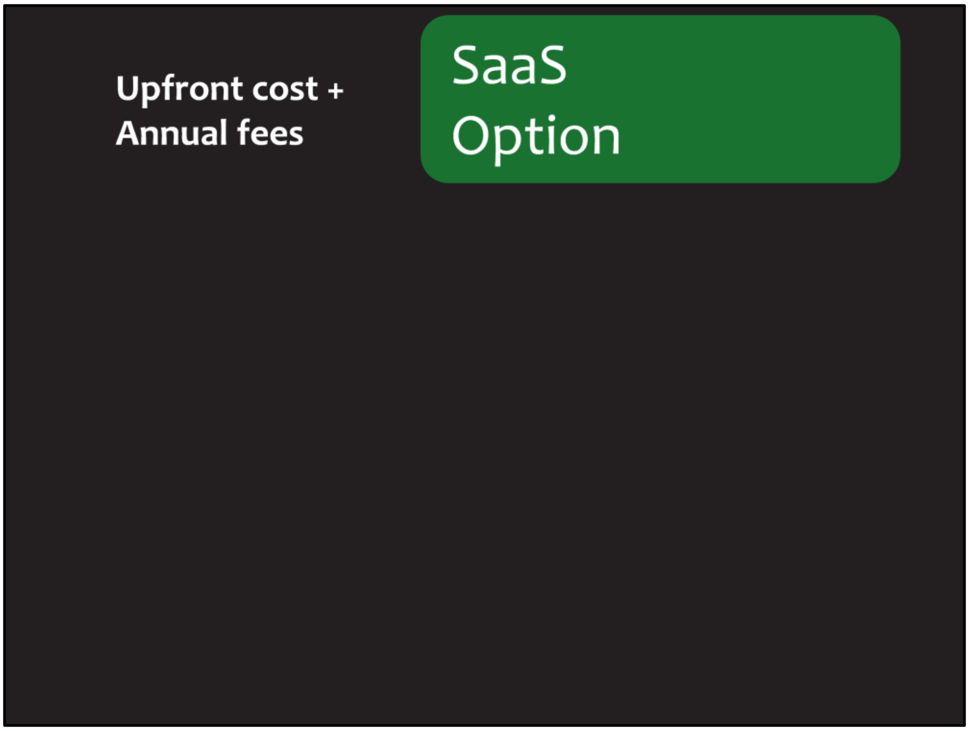
Builder Software

Coding

And with coding languages, you can pretty much build anything you want. Your choice here will depend on your needs. For example, a consulting firm using it for multiple diverse projects is going to need software that’s fully customizable. An after school program that wants to track registration, attendance and activities will most likely be fine with an off-the-shelf option.

Cost

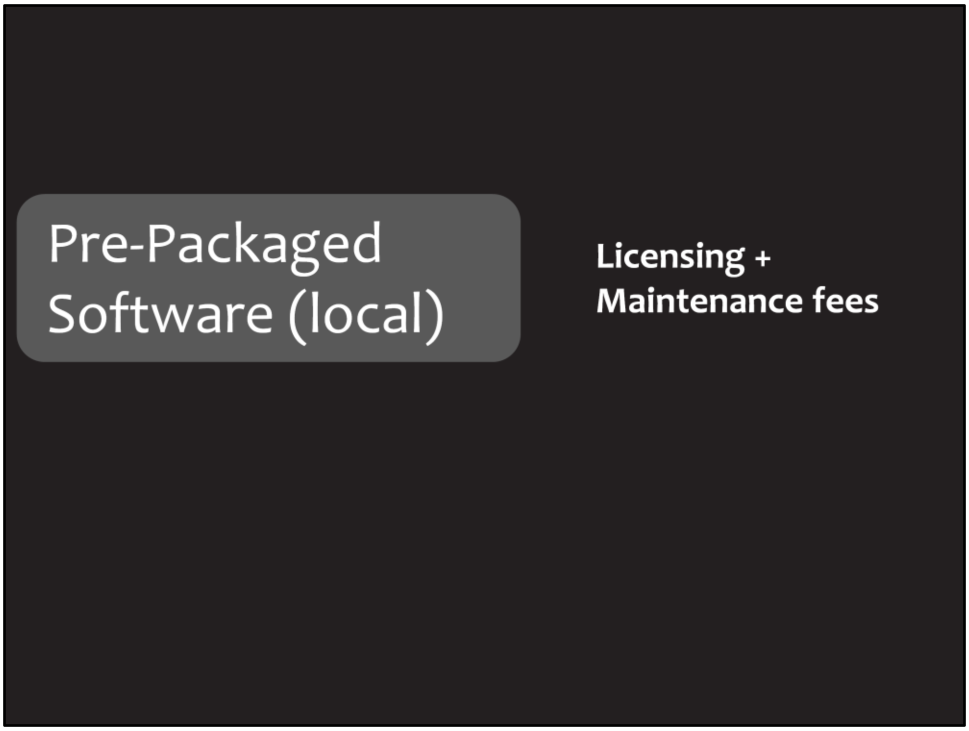
Costs can vary quite a bit. In general though, more complexity and more customizability will cost more, off-the-shelf options are likely to be cheaper. Keep in mind though, a custom built database can result in more savings down the line because you’re getting exactly what you need.



Upfront cost +
Annual fees

SaaS
Option

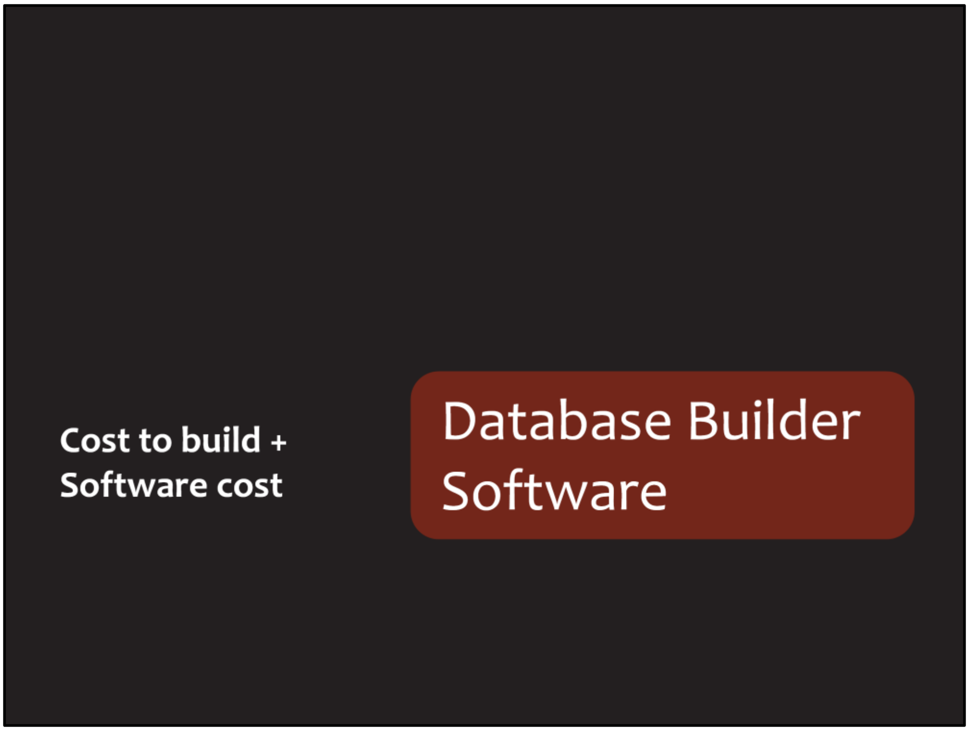
Let’s look at exactly what you’re paying for. SaaS options usually have an upfront cost and then an annual fee, which are all-inclusive, covering customizations, training, support, and maintenance. Prices vary greatly between vendors and are usually linked to the amount of customizability and support you’re getting. I’ve seen some for as little as a few hundred dollars a year and others closer to ten or fifteen thousand a year.



Pre-Packaged Software (local)

Licensing + Maintenance fees

Similarly, with pre-packaged software, you'll pay a licensing fee initially and then fees for ongoing maintenance and support. Here I've seen packages that are a few hundred dollars a year, up to a few thousand dollars a year.



Cost to build +
Software cost

Database Builder Software

With database builder software, a bulk of the cost will be the staff time to build it and developer rates vary quite a bit. Though, you're probably looking at a minimum of $10,000 unless the database is very simple. When it comes to the cost of the software package itself, open source options are free whereas others, like FileMaker Pro, can cost up to a few thousand dollars and then there's everything in between.

Coding
(from scratch)

Cost to build

Most of the tools used to write code are free so it really just comes down to the cost to build it, which tends to be higher, it takes a lot of time to build from scratch. Here, you’re looking at a minimum of $10,000 to $15,000 unless it’s a very simple database.

There's an important trend that I want to talk about for a minute here. It used to be that almost every database out there was totally custom built. Now, there are these SaaS and pre-packaged options, which tend to be less customizable but also cost less. They fill specific needs and niches. Eventually, I think we'll get to a point where there are a handful of great, well-known SaaS and pre-packaged options out there. But, from what I've seen, we seem to be in an experimentation phase, figuring out what generalized, pre-packaged system should really look like. I've seen a lot of finicky, glitchy programs and systems that are so tightly structured that they don't work for anyone. So, for now, I strongly recommend asking a lot of questions and doing as much testing as you can before moving forward with a SaaS or pre-packaged option to make absolutely sure you're getting what you need. The same thing happened in the accounting world, decades ago, all accounting databases were custom built and, eventually, more and more pre-packaged, generalized systems similar to Quickbooks entered the market. It took a while for those options to evolve to the point where they were really good. I think things are still evolving in our world, so just be a little cautious.

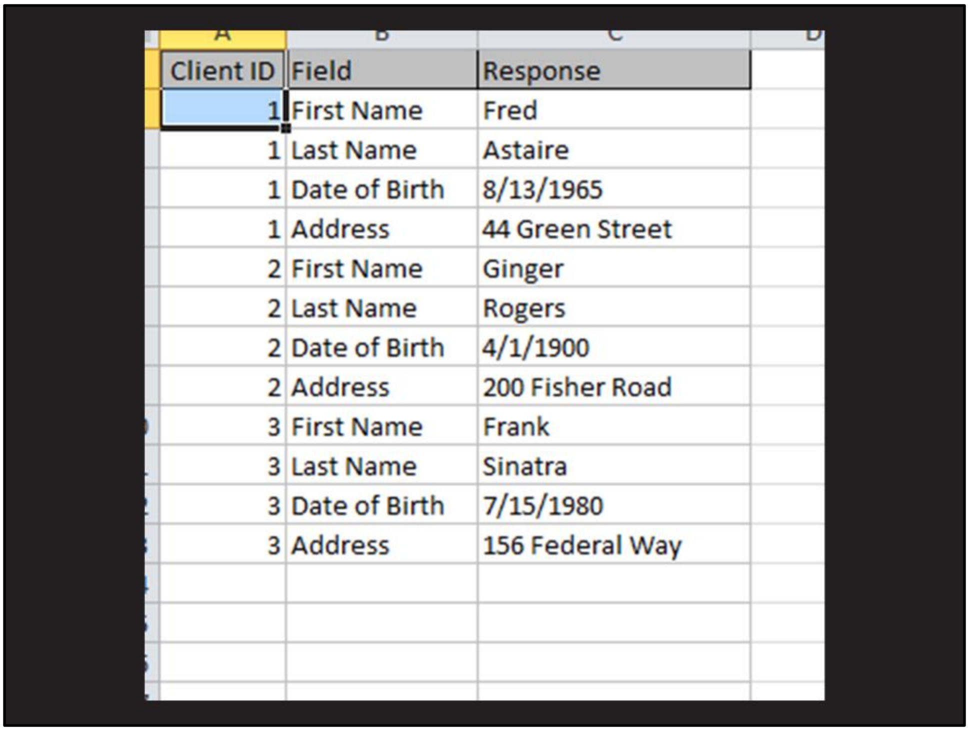
Specific Needs

Regardless of the software program you choose, there are a handful of specific pieces you'll want to explore. If you can get in there and test it out for yourself, do that, otherwise ask your potential vendor or developer about these. Also, have the people who are ultimately going to use the database help test and ask questions (e.g., data entry staff, those folks who will be running the reports).

Specific Needs

Data Tables

First, what do the tables look like? How is your data structured? How to do you get access to it?



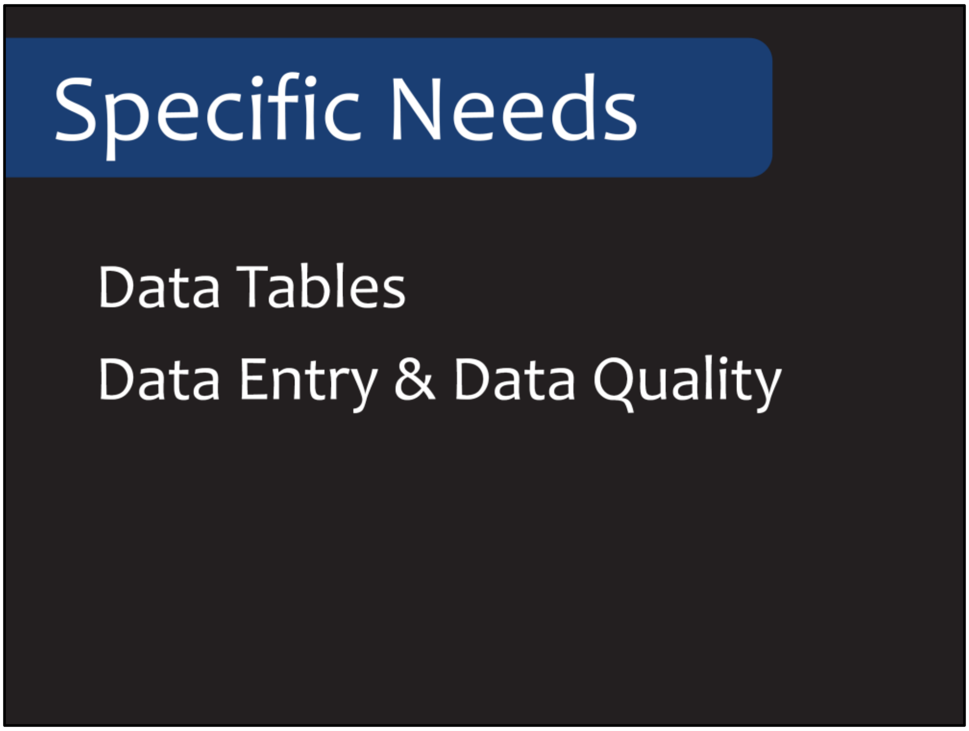
Client ID	Field	Response
1	First Name	Fred
1	Last Name	Astaire
1	Date of Birth	8/13/1965
1	Address	44 Green Street
2	First Name	Ginger
2	Last Name	Rogers
2	Date of Birth	4/1/1900
2	Address	200 Fisher Road
3	First Name	Frank
3	Last Name	Sinatra
3	Date of Birth	7/15/1980
3	Address	156 Federal Way

It’s common practice to develop tables where all of the variables are stored in one field (and coded, unlike the example). It provides a great deal of flexibility from a development perspective. As evaluators, though, we like our data with each variable in a single column. You may need to work with your developer or vendor to make sure you can get that.

Specific Needs

Data Tables

I like to be able to see the tables, get my hands on the raw data, so I sometimes have trouble with systems where I can’t do that. Salesforce is one example. They have what they call ‘virtual tables’, I can never actually see the data in its natural environment. Kinda bugs me from an evaluator standpoint.

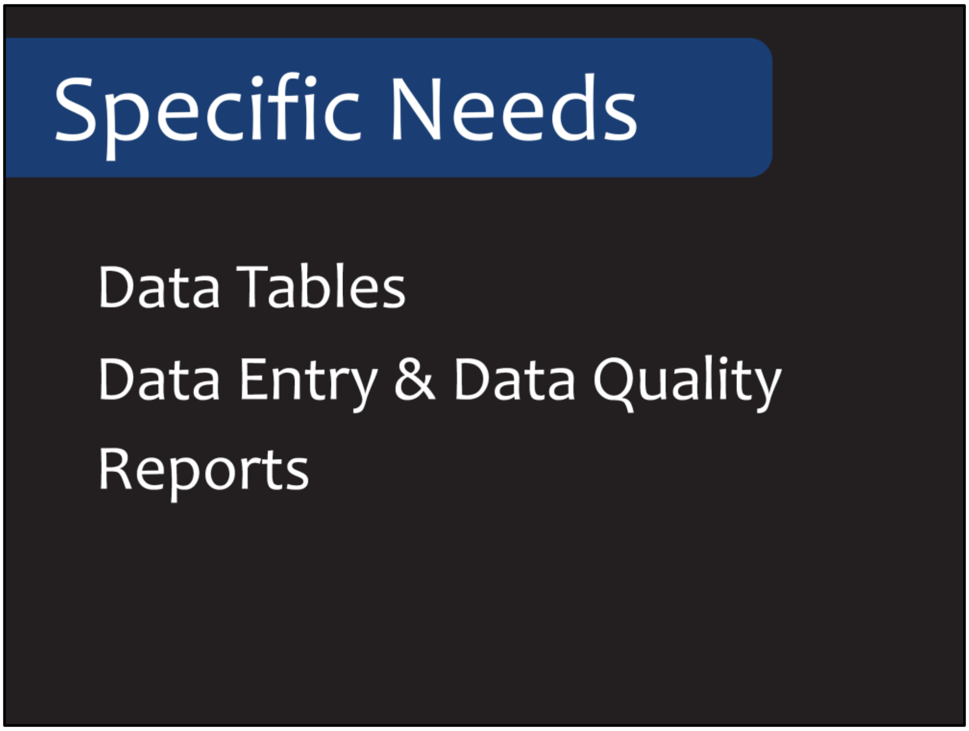


Specific Needs

Data Tables

Data Entry & Data Quality

Next, the data forms. What are the capabilities here? How easy it is it to enter data? How quickly can data be entered? What training will be required for your users? Can you include the data quality and validation checks you need? For example, if a program works only with adolescents, can you check to make sure date of birth is after January 1st, 1993? The benefit of highly customizable options is that you can design forms that look exactly like your paper forms, which makes data entry a breeze.



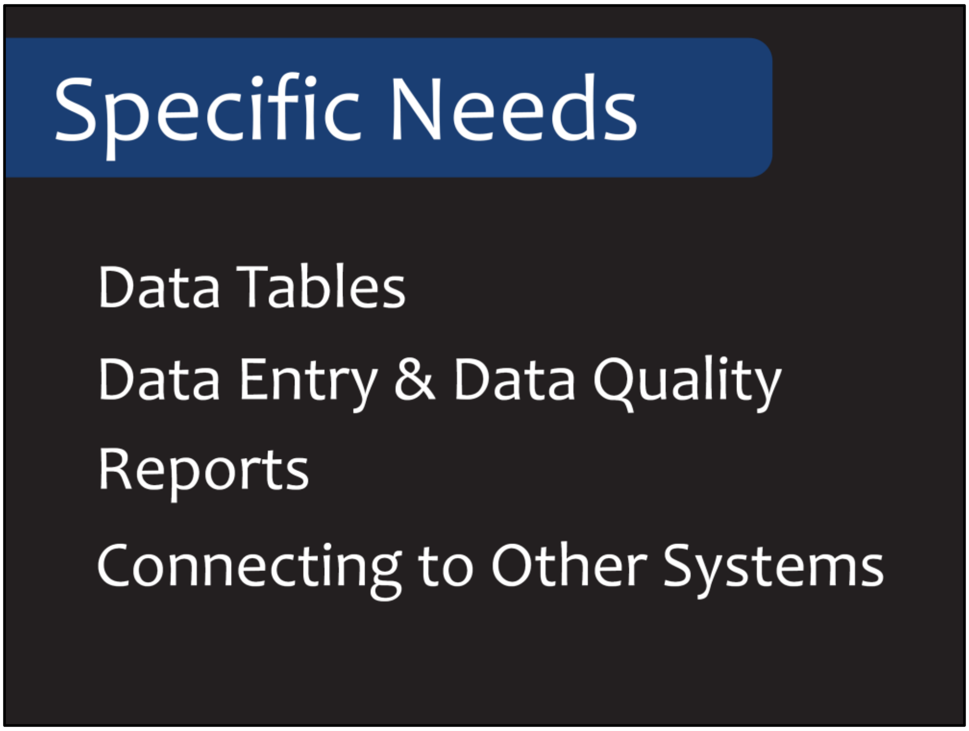
Specific Needs

Data Tables

Data Entry & Data Quality

Reports

Next, reports. As I’m sure you know, good reporting can save so much time. Take a look at what canned reports are provided. Will they meet your internal and external reporting requirements? Is there an option to create custom reports? How user-friendly is this option? How flexible? What training will be required for users? If you’re working with a developer, carefully consider what automated reports will be most valuable.



Specific Needs

Data Tables

Data Entry & Data Quality

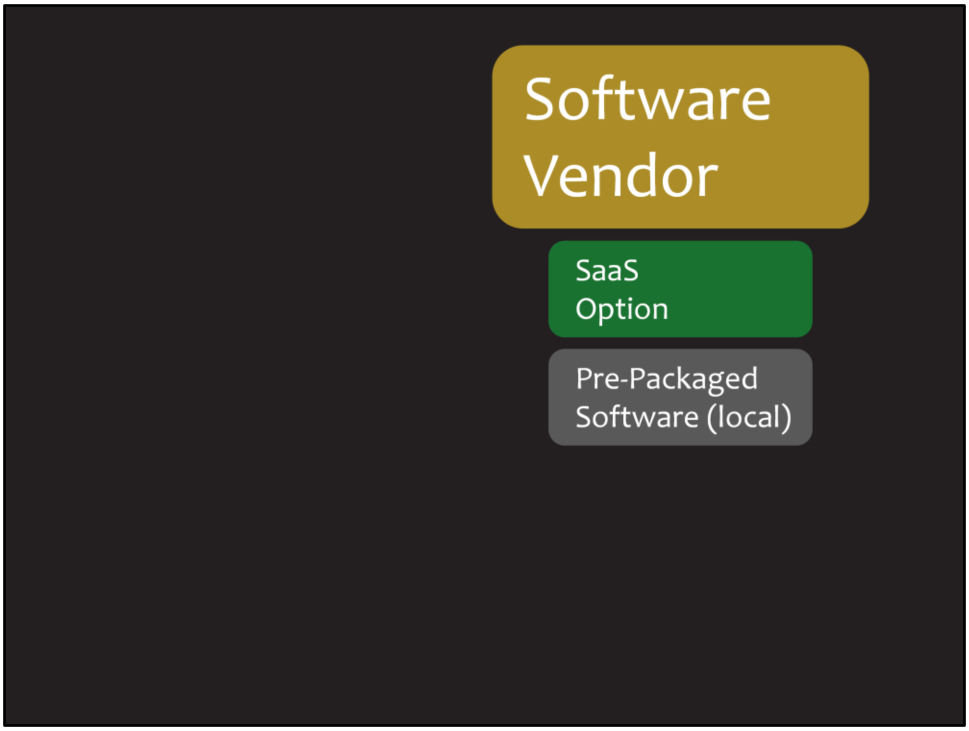
Reports

Connecting to Other Systems

And, lastly, connecting to other systems. You may have data you need to import regularly, can it do that? You may want to get your data into other software for more robust statistical analysis or reporting, can you get data out of the database and into the format you need? Take some time to figure out your exact needs here and bring them to your developer or vendor for consideration. At the very least, you should be able to export an excel or csv file which can be uploaded into just about anything.

4. WHO will build & maintain it?

Onto, question four, who will build and maintain it? As you heard, that came up a lot in the WHAT discussion. Again, we’ll cover your choices and what to consider when picking.

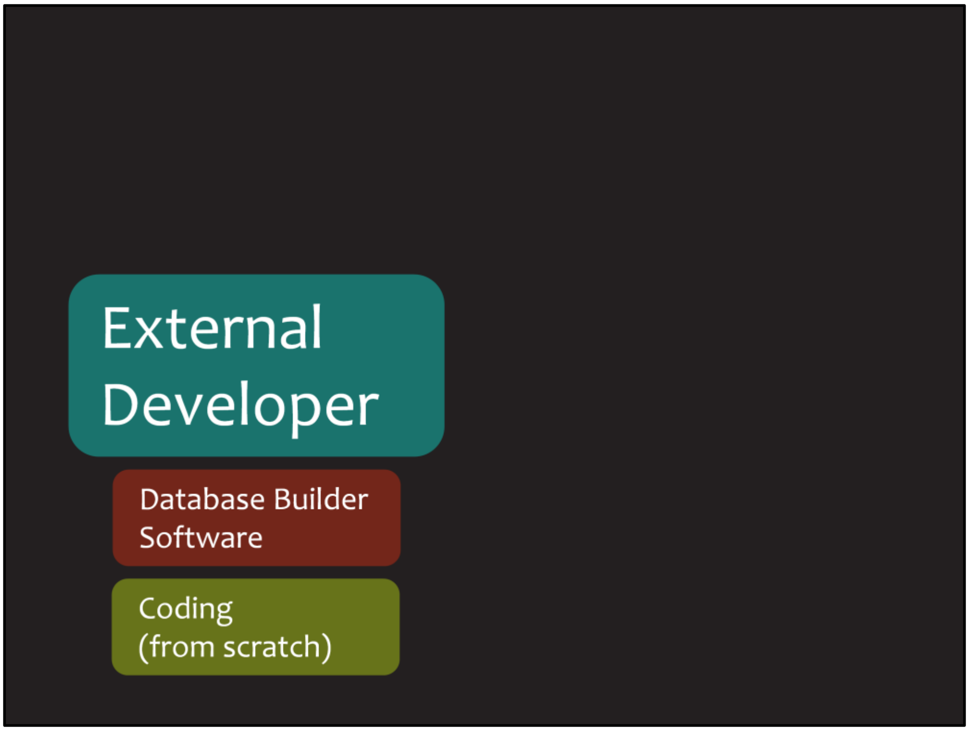


Software Vendor

SaaS Option

Pre-Packaged Software (local)

For SaaS and pre-packaged software, the vendor that sells the software will also customize and maintain it.

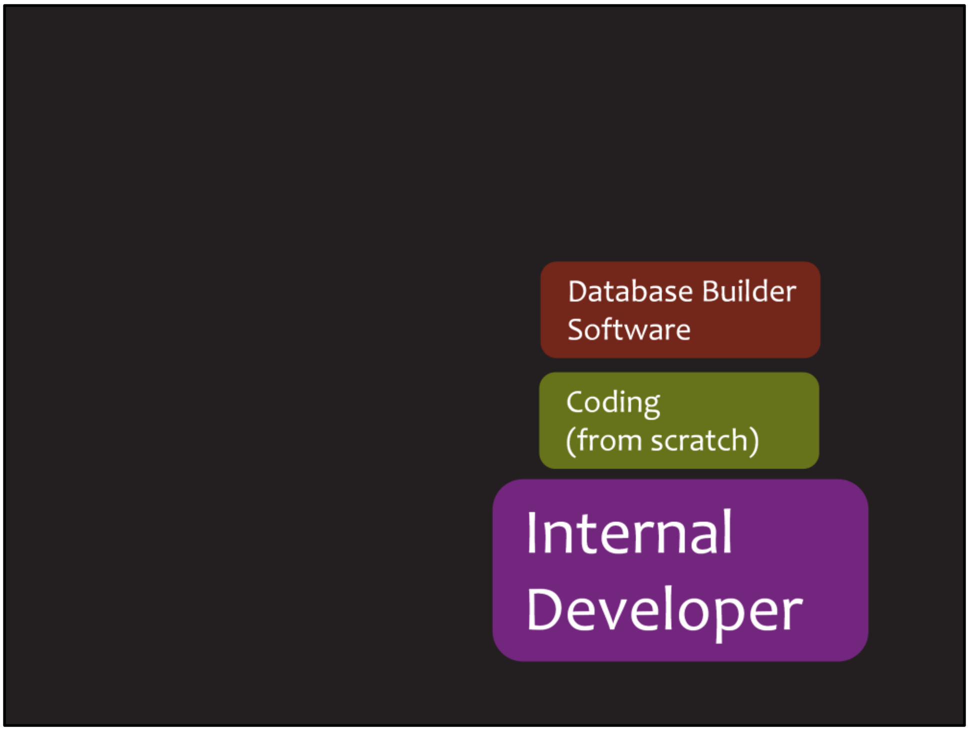


External Developer

Database Builder Software

Coding
(from scratch)

Option two, hire an external developer, either a company or an individual. Typically, the developer would use whatever software program has been chosen.

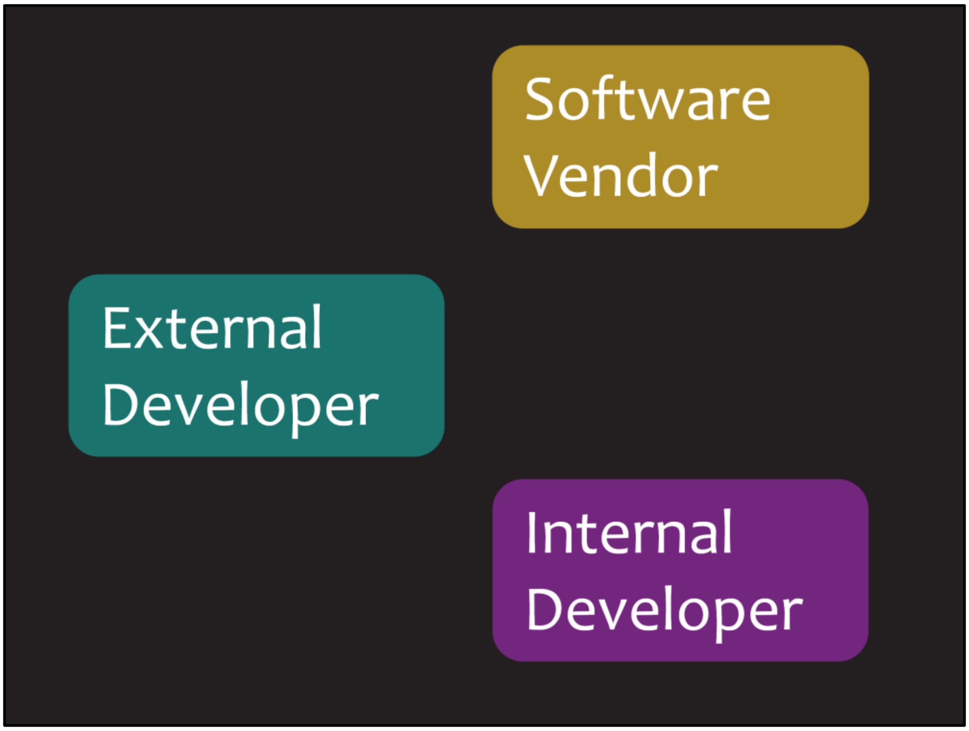


Database Builder Software

Coding (from scratch)

Internal Developer

And, option 3, you can hire a full-time developer. Most medium to large-size research and evaluation consulting companies have at least one internal developer because the needs across projects are so diverse. You'd probably work with the developer to decide which software programs are the best fit for which projects.



Software Vendor

External Developer

Internal Developer

When choosing one of these options, what do you think some of the considerations might be?

Expertise/ Experience

I can’t stress enough how important experience and expertise are. In addition to having the technical skills needed to get the job done, working with a person or a company who understands your needs is INCREDIBLY important. At the very least they need to have a plan for understanding the ins and outs of what you do and what you need. ChallengerSoft, which is a SaaS vendor, for example, has software used primarily by Healthy Start programs. The CEO of ChallengerSoft really understands Healthy Start programs and keeps up with developments in that area. That contextual information is key to providing great database software and support.

Start-up

Whoever your developer or vendor is, ask them what start-up will look like. How long will customization or development take? What training and support are they planning to provide? How will questions be answered and problems resolved? If applicable, are they willing to move any existing data into the new system?

Ongoing Support

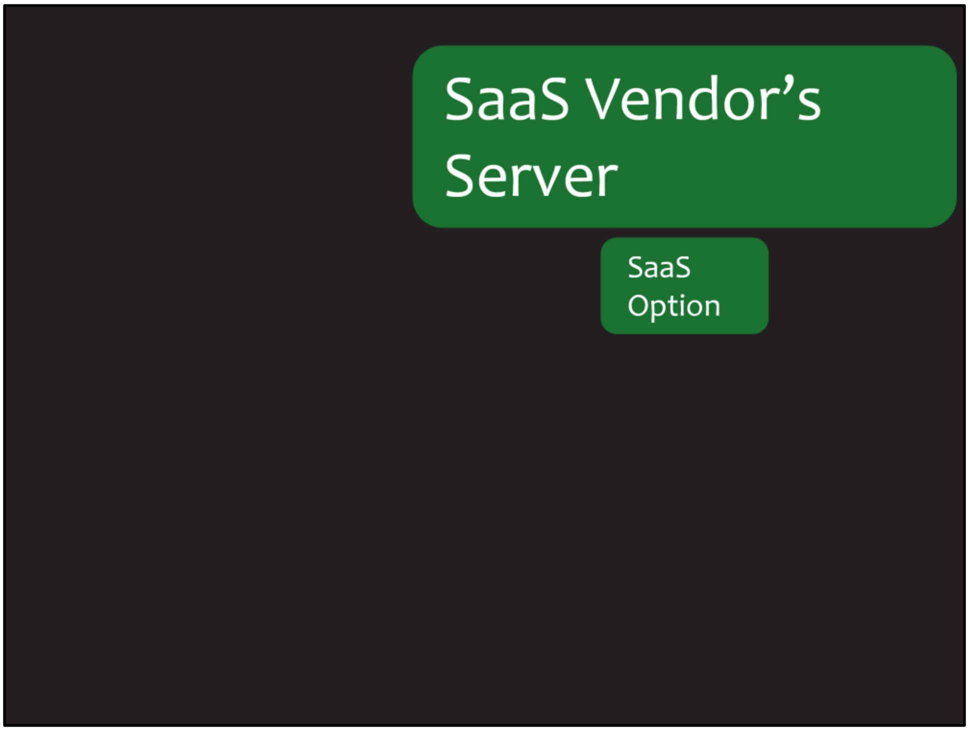
Also ask about ongoing support. How available and responsive will they be? How will questions be answered and problems resolved? How quickly? Over time, you’ll probably need changes, ask how those will be handled. Sometimes external developers want to build it and leave it, leaving you to maintain it. I don’t recommend this, between fixes, enhancements, and user turnover you’re likely to continue to need your developer, find one that plans to stick around.

Cost

Next, let’s look at cost again. For pre-packaged and SaaS options, the software vendor’s costs will be part of the all-inclusive fees. We’ve already discussed development costs for external developers. You’ll also want to factor in the cost of ongoing maintenance, support and upgrades. And, obviously, with an internal developer, you have a salary to pay.

5. WHERE will it live?

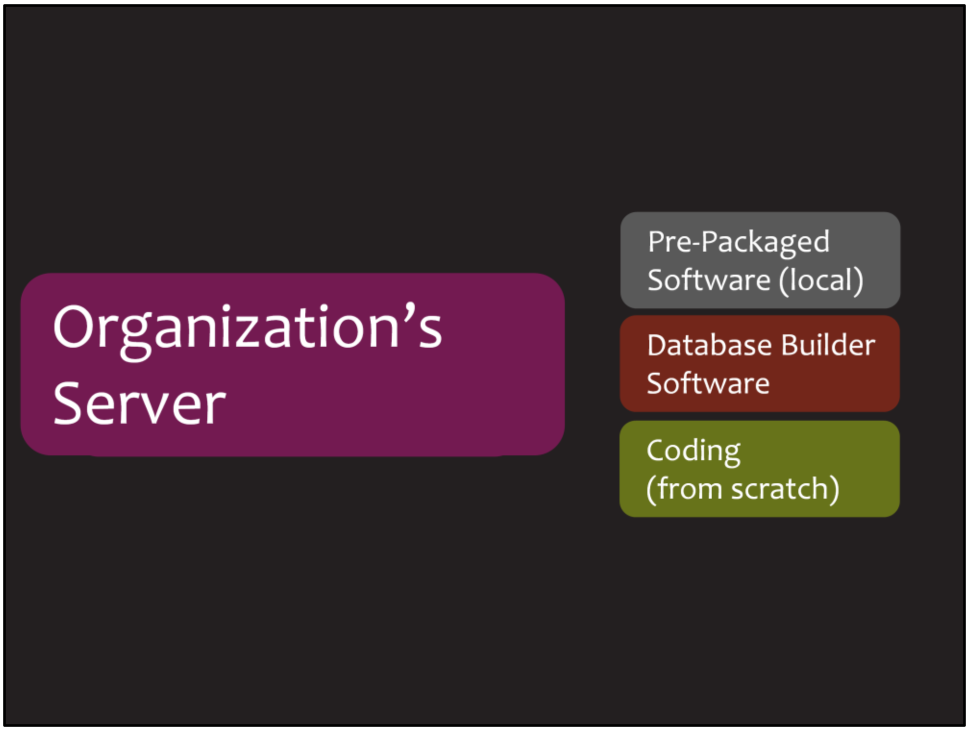
Ok, last big question. WHERE will the database live? Here, again, there are three options. Technically, there's a fourth, which is to have it live on one person's computer. It's not a very good option and I don't recommend it but sometimes the situation calls for it.



SaaS Vendor’s Server

SaaS Option

Of course, if you’re going with a SaaS option, the software and data will live on the SaaS vendor’s server.



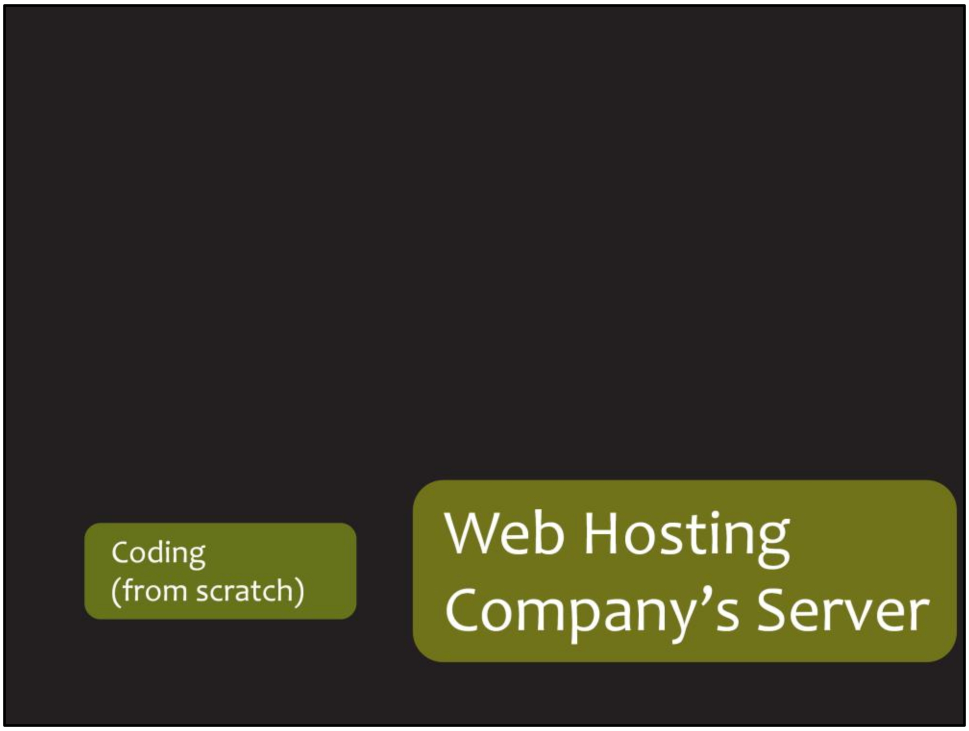
Organization's Server

Pre-Packaged Software (local)

Database Builder Software

Coding (from scratch)

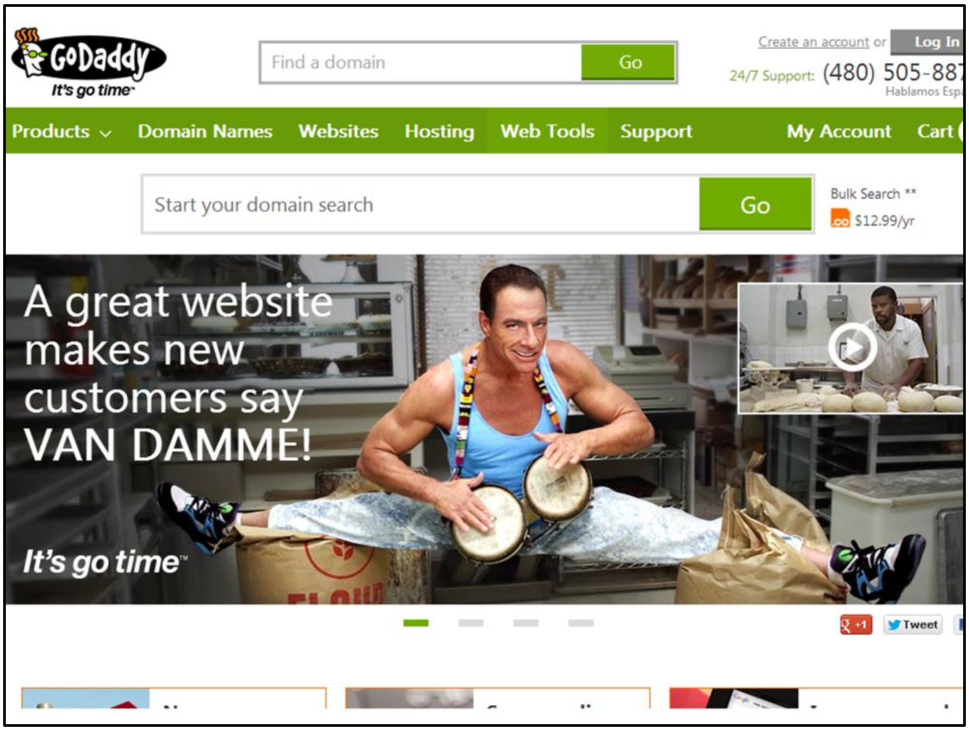
You also have the option of having the software and data live in-house, on your organization's server. This means it would NOT be web-based. Pre-packaged software and most custom built software get housed locally.



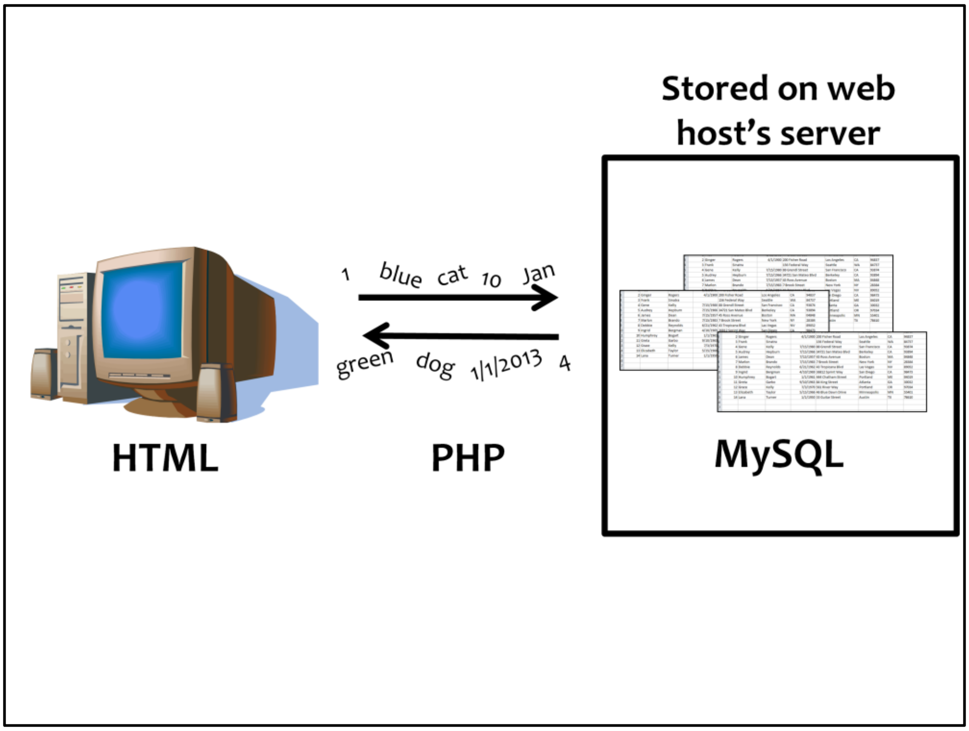
Coding
(from scratch)

Web Hosting
Company's Server

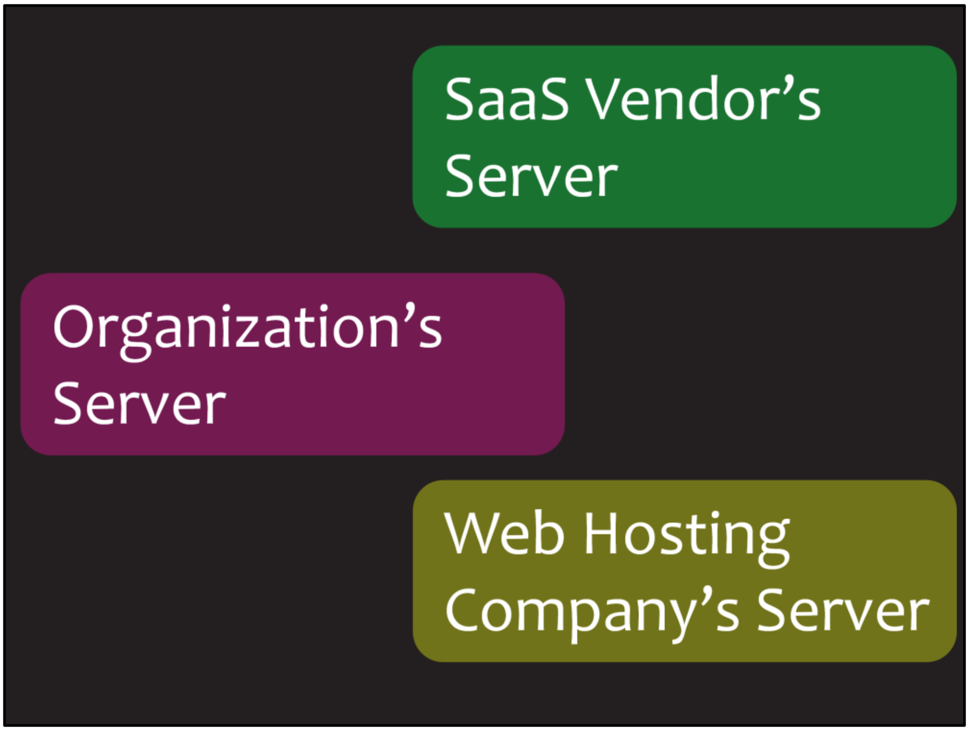
Or, you can use a web hosting company.



As we talked about earlier, there are many companies out there like GoDaddy and DreamHost that will allow you to buy space on their server to host your own website and store data. For this option your software has to be compatible, you can't just plop any database on there.



As far as I know, most of the major web hosting companies primarily support PHP/MySQL databases but there may be other options out there.



SaaS Vendor's Server

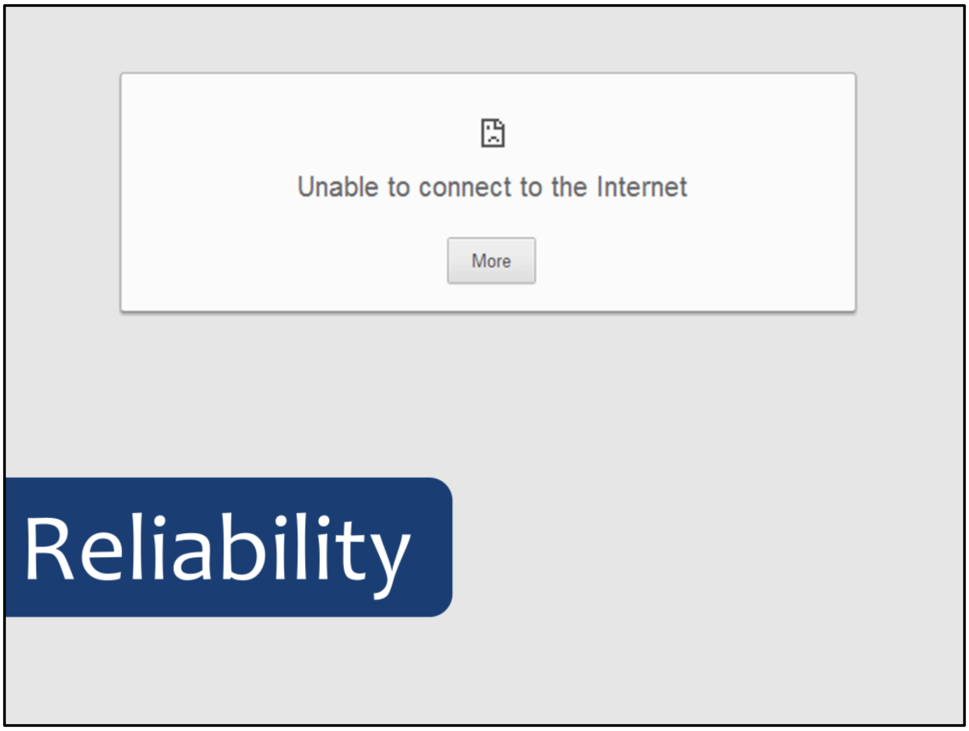
Organization's Server

Web Hosting Company's Server

When choosing one of these options, what do you think some of the considerations might be?

Security

First, security. With the kind of sensitive, client-level data we often work with, this is a key consideration. Wherever the database lives, you'll need to make sure you're meeting HIPAA and HITECH Act requirements and any other confidentiality rules that apply to your data. Be sure to discuss this with your vendor or developer.



Unable to connect to the Internet

Reliability

How reliable is the server? Does it ever crash? If so, how are backups handled? If you’re using a web-based option, think carefully about your internet connection, is it fast enough? How often is your internet down? If your internet connection isn’t up to snuff, I STRONGLY recommend that your database live on your organization’s server.

Cost

And, cost again. With SaaS, server costs are included in your annual fees. If your organization does not have a server and you decide to go that route, you’re looking at a one-time purchase price and IT support. And, with a web hosting company, there’s usually an annual fee, around $100 to $200.

Ok, I’m going to stop there, since I’m sure we’re all ready for a nap after all that. Some final tips…take your time, as you can see, there’s a lot of legwork in getting to the right answer. Ask around, talk to other people who have used whatever software you’re considering. Don’t be afraid to harass the bejeezes out of potential vendors and developers, this is a big decision and you need to get as much information from them as possible. And, lastly, do everything in your power to give the options a good test run.

Thank you!!!

Laura Keene

Thank you so much for coming today!

Contact Info:
laura@keene-insights.com
323-380-0806
keene-insights.com