



## Reflection on the Quick Base Certification Experience May 19, 2020

Yesterday I graduated from [QuickBase University](#) on the first attempt, with a half hour to spare in the allotted time (60 questions in 45 minutes versus the 75 minutes allotted), with a 90% score. (The score required to pass is 70%, and most fail on the first try, with an average score of 56%.)

Here are some differences made by my twelve day, forty-five hour, \$99 investment:

1. **Tabs.** What a sleeper. What a game-changer. I learned how to make much better use of space on forms by using Tabs instead of Section Headings. Now I arrange form sections horizontally across the bottom of a form, then add an invisible Section Heading immediately below the Tab to keep field names positioned above fields. This makes optimal use of screen real estate and makes the form much more intuitive and user friendly. (The use of the invisible section heading is a trick I learned from Sharon Faust, the QuickBase Junkie.) The colorization of Tabs also makes forms' look and feel much more appealing, and adds a layer of meaning to the app if colors are used consistently.
2. **Joiner Tables for many-to-many relationships.** I had already stumbled upon this trick on my own through a trial and error process when I built a Competencies table for a client, in which there is a many-to-many relationship between Employees and Skills. (One employee can have many skills, and one skill can be had by many employees. The Competencies Table cross-tabulates the two.) Learning about Joiner Tables in Quick Base U. validated, affirmed, and solidified my design choice. It bolstered my confidence that I was on to something, that I was doing something right.
3. **Automations** – what a mind-blowing, time-saving game changer. One vital automation is to delete child records when a parent record is deleted. When a project is deleted, so should phases and tasks be deleted. When an invoices is deleted, so should its line items be deleted. Without such automations, your app could become a horrible mess with all kinds of orphaned child records cluttering your tables.
4. **Dynamic Form Rules** enabled me to design much more simple, less cluttered, more user friendly, intelligent forms.
5. **Roles and Permissions** enable me to design more secure applications for my clients.
6. **Web Hooks and APIs.** Wow.
7. **Formula Buttons.** Wow again.
8. **Connected Tables.** Now my client has tables in QuickBase that mirror his Customer, Job, and Project accounts in QuickBooks (within ONE TABLE, mind you), and his revenues via invoice line items. I've contacted QuickBase about including Sales Receipt line items too. When they add those, his revenues will be totally mirrored in QuickBase and related to Customers/Jobs/Projects and will seamlessly and effortlessly agree with QuickBooks, to the penny.
9. **Noticed a bug:** child records on Connected Tables won't clone. I've submitted a support ticket on that with a video of my experience.
10. **Embedding child reports in parent tables using URL fields.** Now I can design what I had previously depended on QuickBase to do for me automatically whenever I created a table relationship. Now I don't have to cross my fingers any more and hope that it works right. I can design it.
11. **Formula Fields.** Now I have a much more clear grasp of field types and how to use different formula

fields appropriately and with less bugs, glitches, and errors. Power and potential here is limited only by the imagination.

12. **Swim Lanes** for app planning. These hadn't yet been developed when I was in business school. Very helpful.
13. **Giffy**. Terrific design tool.
14. **Documenting**. I learned where, how, and when (as you go) to document apps I build.
15. **Home Pages**. I learned that you can build multiple home pages, assign them to roles, and that you can even create code pages that can be referenced in field formulae, sort of like subroutines. The power potential here is mind-blowing. The surface hasn't even yet been scratched.
16. **Email Notifications, Subscriptions, and Reminders**. Enables me to add functionality that users routinely expect.
17. **Calendars**. I learned how to use them and embed them in records so that every important person and thing has his/her/their/its own calendar, and so that events are linked helpfully in ways that are impossible with Google Calendar.
18. **QuickBase – QuickBooks Goddess in Texas**. Learned about Robin Hall of VARC Solutions from review of Empower 2015-2016 Power Point slide decks. Terrific resource.
19. **User Management**. I learned better how to manage users. I had blind spots about this because I was focused on Apps. Users transcend apps; they're managed at a higher level and are kept forever, so it's important to know how to manage the user archive so it doesn't become cluttered, and so that security is ensured and unnecessary expenses are not incurred.
20. **Report Management** was somewhat demystified. I learned where to go to manage reports that are not always visible to me.
21. **Role Emulation and Testing**. Hugely helpful. Now I don't have to take users' time to learn how the app appears to them.
22. **My Own Inventions**. Through their absence in the curriculum, I noticed my own inventions:
  - **Looking up summary fields in child tables**, like hours spent or percentage complete so far. This empowers front-line people to spend time wisely and determine when a task is complete.
  - **Bottom-Up determination of Percentage Complete on a Project**. I found that conventional wisdom uses a top-down approach, which I believe is uninformed, presumptuous, imperious, and foolish.

(Examples of this foolishness include premature software release, or declaring an unfinished, unsafe building ready for occupancy, yielding to peer or political pressure, or arbitrary deadlines.)

With the bottom-up approach, a thing is complete when the people building it say it is, not when higher level managers or other influencers want it to be. They determine percentage complete on Tasks and when Tasks are complete. Those Task completion percentages automatically roll up through Phases to Projects themselves.

Higher level supervisors and managers give front line workers the respect and appreciation they deserve. I feel this is in everyone's long term best interests, and makes proper use of an organization's nerve endings.

- **Whole Day Time Keeping** with an Activity Type field, connected to all relevant tables, helps monitor activity, work in progress, and automatically compiles a collection of useful logs for things like billable work, customer service, training, opportunity recognition and management, and mistake recognition and correction. It removes the question, “Is this noteworthy?” because, as it turns out, **everything** is . . . **especially surprises**.

I encounter a lot of resistance to Whole Day Timekeeping because it's brutally honest. Most people prefer to be selective about what they track and what they don't track; but I realized that if a year consists of almost 6,000 waking hours, and a full time job is 2,000, **what about the other 4,000 hours?**

**What about the other two thirds of my life?**

**Isn't that important too?**

**I say it is.** Indeed, I would say **it's the most important part.**

Since August of 2015, I have been using Quick Base to do Whole Day Time Keeping on myself:

### Work / Leisure by Year

Work / Leisure	Leisure		Work		Totals	
	Hours (tot)	Hours (tot) (% of row)	Hours (tot)	Hours (tot) (% of row)	Hours (tot)	Hours (tot) (% of row)
<u>2015</u>	<u>779.7</u>	67.6%	<u>374.3</u>	32.4%	1154.0	100.0%
<u>2016</u>	<u>2695.7</u>	52.5%	<u>2443.7</u>	47.5%	5139.4	100.0%
<u>2017</u>	<u>4044.6</u>	68.9%	<u>1824.5</u>	31.1%	5869.1	100.0%
<u>2018</u>	<u>4423.3</u>	75.4%	<u>1444.8</u>	24.6%	5868.2	100.0%
<u>2019</u>	<u>3850.9</u>	64.9%	<u>2079.9</u>	35.1%	5930.9	100.0%
<u>2020</u>	<u>1189.5</u>	53.8%	<u>1021.2</u>	46.2%	2210.6	100.0%
<b>Totals (12 groups)</b>	16983.7	64.9%	9188.4	35.1%	26172.1	100.0%

Lessons learned and benefits enjoyed have been invaluable, and are ongoing.

Respectfully submitted,

Kris Freeberg, Economist  
[Making End\\$ Meet](#)