EPISODE 1145

[INTRODUCTION]

[00:00:00] JM: Databases are the source of truth for every company. Editing the data and the database normally requires writing a query in SQL or a domain-specific querying language. These languages are only accessible to engineers and highly-technical people. BaseDash is a tool for interfacing with the database without requiring the usage of a query language. It allows the user to interface with the database as easily as a spreadsheet.

Max Musing is a founder of BaseDash and he joins the show to talk about how it works and why he built it.

Max, welcome to the show.

[00:00:31] MM: Hey, Jeff. Thank you. Happy to be here.

[00:00:33] JM: So there are many ways that a database can be explored. We can make SQL queries. We can use some simple explorer, that's off-the-shelf database explorer. What are the different ways that people explore databases?

[00:00:50] MM: Yeah. So I guess what we've really been seeing is that there are a lot of tools out there that exist that are really built for engineers to be able to explore this data. So you think about typical sort of SQL database clients. That sort of give you the table view and lets you sort of go through that, write SQL queries to pull data and maybe make edits to that as well.

What we've sort of found is that there aren't any really great tools out there for non-technical people within companies to be able to access and explore this data. And yet they're typically the ones that actually are the ones that need to explore the data or typically make edits.

And so we're actually trying built out a tool that lets these non-engineers within companies be able to explore and edit this data. Really, the alternatives nowadays are typically for companies to build these kind of tools in-house. And so usually what you'll see is some sort of crud

interface that a company builds custom in-house that's basically just like forum to be able to either edit data, create new records or create some sort of custom interface to explore that data.

And so we're building BaseDash, which is this kind of new way to allow these non-engineers to have the standardized interface so that you don't have to build out these custom interfaces for these non-technical people within companies.

[00:02:13] JM: And what would such an interface do?

[00:02:16] MM: Yeah. So you can think of it kind of like if you're familiar with Airtable, it's kind of like this interface that we're building that is a standardized spreadsheet-like or table interface that lets you filter, sort and explore your data in different ways and then save those into views so that it's really easy for non-technical people to effectively build SQL queries without needing to know how to write SQL.

And so from there, once they're created these sort of dashboards or views, what they can do is then just as simply as editing a spreadsheet. They can just double click on a cell, make a change, hit enter. And then that change will go straight through to your SQL database. And what's great about that is that giving these non-technical people access to your database like that means that they don't have to go around bugging engineers to make these changes for them every time they need to. And so it makes that whole process of editing your production database much easier in as really safe way as well.

[00:03:19] JM: Tell me a little bit more about the usage of BaseDash.

[00:03:22] MM: Sure. I guess sort of one of the main functionalities or use cases of BaseDash is really around customer support. And so we see a lot of companies where customer support really needs to access the production database to be able to make changes typically to customer data or data that's somehow generated by a customer.

So if you think of, say, maybe like a social network where a user creates all these sorts of data around posts that they create or their profile, a lot of the times there's not going to be clear ways for the user to be able to manage some of that data. So if you think maybe like early on in a

startup. You might not have something like an edit email field just like as a basic example. So customer support often will get these requests from users saying, “Hey, I need to update my email address or my order address for some order I made.”

And then with BaseDash, what they can do is just jump straight into the database and then make those changes write in the database without having to talk to an engineer to do that. And so sort of getting this direct access to the database allows them to make these changes on their own, which is really useful.

[00:04:44] JM: How do you build a platform on top of a database? What kinds of integrations do you have to make with the database?

[00:04:51] MM: Sure. So that’s a great question. So we’re actually starting out BaseDash with the prime focus on SQL databases. And so that makes it really easy for us to start out, because we just need to be able to connect to SQL databases, which is simple because it’s basically just a standard, right? And so what we can do is basically hook into your database credentials. You can create a user specifically for BaseDash that has read and write access to certain pieces of your database. And then we can just hook straight into that and then be able to pull data from that database and write changes back to it whenever we need to.

We are though sort of planning to expand things beyond just SQL databases. So eventually being able to support any kind of data source that your company has. So if you think of things like Stripe, Zendesk, HubSpot, Salesforce, these are all big data sources that are essential to your company and yet you don’t store the data yourself in a SQL database. They’re stored somewhere else on the cloud.

And so eventually what we want to be able to do with BaseDash is to be able to hook into all of these different integrations and data sources and then be able to pull them all back into BaseDash so that you could then manage everything in one place. BaseDash kind of becomes like your UI layer on top of all these different data sources.

And so that’s where things get a little more tricky, because then we have to sort of build this abstraction layer on top of the different sources that we can have a familiar read-write protocol

to all of these different sources. So that's going to be a big challenge moving forward. But for now since we're just focused on SQL databases, we're basically just using SQL database library that we can just hook into these databases and then make changes as we need to.

[00:06:34] JM: Now, of course, there are companies working on that integration across different data sources. Like you have like data connector companies like FiveTran. But it seems like you are sort of looking to leverage all of the lower-level plumbing that people have been spending a lot of time building out. And you're focused on this higher-level abstraction that just makes it easier for users to play with.

[00:06:58] MM: Exactly. So there are lots of tools out there that are really great, as you mentioned, for sort of connecting to data and sort of bringing things into a single data warehouse. We're really sort of focused on the end product here of trying to build out this actual product that gives you this access to explore and manage the data as you need to. Beyond just sort of pulling the data together into some sort of API or some sort of data warehouse where you can sort of analyze stuff from there, we really want to build the end product where you can just sort of deploy it in a couple minutes and then be able to do anything you need to with that data as an end user.

And so we're kind of in like the no-code or low-code space and that we're sort of trying to replace the need for engineers to have to build-out this kind of tooling themselves. But yeah, we're definitely focused on sort of the end product here rather than sort of the plumbing in place to build it yourself.

[00:07:55] JM: Can you give an example of a prototypical user type who uses BaseDash?

[00:08:01] MM: Sure. I guess a good example might be sort of on the operations side of things. So all the time – Maybe to give a clear example, we've got a user who does sort of like delivery of items for their companies. So they'll sort of be like an Amazon type company where you can pick an order through them and then they'll have to deliver products to your house.

And so instead of building out like a whole back office around that sort of like a Shopify type thing where they would have to like build out an order management system and customer

management. They just use BaseDash to hook into their database so that they can see all the orders in real time as they come in and then be able to fulfill them, send out delivery carriers to be able to deliver those items, and then be able to change the status of those orders as they're happening. So we're kind of acting as like their back office that lets them than access all of their data and manage it as it comes in in real time.

[00:09:00] JM: The vision for low-code often has this element of triggers and integrations with other platforms. Like you might change something in an Airtable table and it triggers some Zapier trigger to go do something else, which triggers an AWS Lambda function to go do something else. Do you have a vision for that being a part of BaseDash?

[00:09:21] MM: Yeah. Actually we do. So I think that's going to start off with sort of like a Zapier integration to start where you can sort of huck into different kinds of platforms to be able to manage things like you were talking about. But I think eventually we will want to build out our own sort of automation workflow system around BaseDash. I think that there's a really interesting opportunity there, because since we'll already have these integrations to all your different services, we can make it really easy to build these kinds of automation specifically around the data management space.

So if you think about a lot of the kind of automations that companies are building within things like Zapier, a lot of the time it's really just sort of when one piece of data changes in one software change some other piece of data in another integration. And so since we're so focused on the data management aspect of things, we can really build that out in a really easy way because we sort of can build this higher-level abstraction layer on top of all the different data sources. So that's something we definitely want to explore.

Yeah, that's probably going to start with sort of manual workflows where you can just build out workflows that touch all these different data sources at once, and then eventually will build in triggers that you can then automate those workflows .

[00:10:40] JM: You've integrated with a variety of different databases. What have been the hardest integrations to build?

[00:10:46] MM: It's actually been pretty easy to start. Since we're just focused on those SQL databases, we don't have to worry about all these huge differences and sort of how they work. And so we're actually using a library called Sequalize, which makes the integration with all the main SQL databases really easy.

There are few differences in sort of how they work in terms of different kinds of data types or different kinds of support for how they handle different kinds of SQL queries. And so we've had to sort of manage the way we handle those differences. But, really, since were focused on the SQL databases, it's been pretty much as like a single effort to get them all working in one go. But that's really going to change once we start to have to worry about different kinds of integrations with non-SQL databases.

So we're planning to add things like MongoDB and Firestore. That's going to be a lot trickier, because now we have to worry about on sort of like interface side of things, how do we change something that's often a deeply nested data store to be able to view that in the table view? So that's going to be a big technical challenge and design challenge as well.

[00:11:57] JM: There are a bunch of these different sophisticated SQL explorer type situation applications. How are you planning to differentiate from the other ones on the market?

[00:12:08] MM: I think a lot of the existing tools out there are really built for a specific user, and that is really the individual engineer within a tech company. The main difference that we have is that instead of individual engineers, we're whole teams of typically non-engineers within companies. And so in practice what this looks like is we have to be able to support sort of multiplayer collaborative workflows.

And so collaboration is actually a really big part of BaseDash and a lot of the ways that we design features in the product as a whole. And so everything from sort of the standard table interface that you get when you're editing data is fully collaborative. So kind of like Google Sheets where you can see other people's cursors. You can see updates coming in real time, and you can also share any of the views that you create.

So as I mentioned, we have sort of like a view builder where you can build-out SQL queries sort

of through the UI really easily. All of those sort of stiff dashboards and reports that you create through views can then be shared across your whole team. And so everything's been designed to be collaborative from the start, and also everything's been designed from the start to be used by non-technical users as supposed to solely engineers.

And so that really means that we've had to design things in a very different way so that you can't just like click a button and accidentally delete your whole database, for example. And so, really, I think that's one thing that a lot of these existing database explorers and clients really haven't been focused on that I think really differentiates what we're on.

[00:13:55] JM: There is a search functionality in BaseDash. So when I load my database into BaseDash, I can search through it. Is that to say that you build indexes on top of the databases that get used with BaseDash?

[00:14:12] MM: So currently we actually don't. Currently we're basically just run a query that looks through all your columns, which isn't super-efficient. But yeah, we do plan to sort of build things on top your database that makes things a lot easier and faster for you.

So definitely, indexing is going to be a big part of that. We'll probably add some sort of way to maybe suggest indexes that you can add to speed up certain parts of how you're using BaseDash. I think this kind speaks to maybe an interesting point more generally about BaseDash, which is that there're a lot of things that we have to build into BaseDash that already exists within SQL databases that we kind of have to replicate.

So if you think about like views, views are pretty essential part of SQL databases at its core, and yet we had to basically rebuild a view system on top of that in BaseDash, and the reason for that is because we're planning to add all of these integrations with different data sources. And so, yeah, we could just build views within your actual SQL database, but eventually what we want to be able to do is hook into all of your different data sources and build views that connect all of those together. And so all of these things like views and roles that are all part of SQL databases already, we have to kind of build an abstraction layer on top of that. It will work with all these different integration sources.

[00:15:35] JM: You have a change log feature so you can track changes that occur across records in BaseDash. Tell me about how the change log is built.

[00:15:47] MM: Sure. So any time someone makes a change within BaseDash, we basically just keep track of that with a record in our database so that any time a cell is changed, we can just tell, “Hey, this was the value before the change. This was the value after the change. This is who made the change and at what time.”

And what that lets us do is build this sort of almost like a Git log of all the manual changes that are going on within your database. So that’s really cool for a couple of reasons. First of all, just for security purposes. It can provide sort of like an audit history or audit log of everything going on within your database. You can check who's making what changes. But also, again, it sort of builds on top of this idea of being a multiplayer collaborative experience.

So suddenly now you can see what other people on your team are actually doing in your database. You can start to see, “Hey, this is an area that maybe we’ll want to automate more, because a lot of people are making manual changes on this kind of record.” And so that's really cool. We also support comments on records. So you can sort of write a comment describing why you made a certain change within a record. So maybe you get a customer support request ticket and then you can sort of link that identifier into the record as you make the change, which is really cool.

One thing also to note is that we actually don't store a copy of your data anywhere within BaseDash, again, for like security and privacy reasons. We don't want to be responsible for keeping a copy of everything that you have. And so we just sort of connect straight to your database and make pull records straight from your database as we need to. And then also write changes straight to your database whenever you need to make a change. And so the edit history is actually the only aspect of your data that we need to store and we only really need to keep the before-and-after value of every change.

[00:17:39] JM: What has been the hardest engineering problem to solve with BaseDash?

[00:17:44] MM: I think the hardest thing is actually something we haven't done yet that we’re

actually just still working on, which is the integrations as I mentioned. And that's can it be really because we have to build out that abstraction layer on top of these different sources. And so, effectively, what we have to do is build-out kind of like a standardized API that lets us read and write to any data source. And so that includes things like your internal database, whether that'd be a SQL database, or no SQL, or external APIs that we have an idea of what the documentation looks like for that. So things like Stripe, Zendesk, HubSpot, Salesforce, or even internal APIs that maybe we don't have access to good documentation for. So that's going to be a really big problem that we're working on right now.

And then on top of that, beyond just sort of connecting to those sources and reading and writing data from them, what we also want to be able to do is then connect those different sources together, so within views. Sort of like how in a SQL database you can join tables together. What we want to be able to do is be able to join different data sources together. So effectively creating these kinds of views that join across tables in different data sources. That's a really big technical challenge that we're going to be working on. And I've got some ideas for how that might work, but I really have no idea how it's going to end up actually looking once we build it out. So we'll see how that goes.

[00:19:14] JM: Tell me your ideas.

[00:19:16] MM: Yeah. The main idea that we have is to build-out sort of like a temporary SQL database that we could basically pull data from all these sources into and then basically just run SQL queries on top of that. And then as soon as you're done with that, we can just throw away the database. Again, we don't want have to store any of this data permanently.

And so we could just have this sort of temporary data store that we can run all of your views on top of. That's sort of the main idea we have now. We'll see how that actually ends up working in practice.

[00:19:53] JM: Yeah, you could use like Spanner or something to load all that data into. And it wouldn't necessarily be the fastest thing in the world, but it would work, right?

[00:20:04] MM: Yeah, I think so. I think performance is a big part of BaseDash that we want to

be really important. In general, I think user experience is really important for a product like this, especially where you're taking something that is so fundamentally technical. Something that in history has really just been a tool for engineers and bringing that to non-technical people within a company, user experience is just very important there.

And so design has been a really big aspect of BaseDash. And so I think performance really plays a big part in the UX of the product. And so we're going to have to try and come up with ways to make all of this pulling and pushing of information to these different data sources really performant. So it's going to be a big engineering task.

[00:20:52] JM: Well, if you're going to do a join across like Red Shift and Snowflake, there's no way that's going to be fast ever, I don't think.

[00:21:02] MM: Yeah, that's true. I think we aren't really trying to target those kinds of huge data warehouses where you just sort of like throw all of your data into Red Shift or Snowflake and then mostly you just use it for analysis. I think we're really sort of targeting the kinds of data sources where they are more so used for purposes where you actually need to be able to read and write on a regular basis.

So if you think of something like – I don't know, say Stripe data. Stripe data is something where obviously you need to be able to read your Stripe data to see sort of where customer data is at and subscriptions and stuff like that. But also it's the kind of data that changes a lot and often changes through manual circumstances, right? So a lot of the time a customer will reach out and say, "Hey, I need to cancel my subscription, or I need to change my subscription to a different plan," or do something weird that's maybe not part of your application that the user can do themselves.

And so in these kinds of data sources, usually they're not as huge and as impossible to sort of work within performant ways. So I think we're going to try and make it as fast as possible. But we'll see how possible that is with some of the bigger data sources, especially once you start to join across different sources. That's going to be kind of weird.

[00:22:22] JM: Did you build BaseDash – Is this an Electron app? It looks like Electron.

[00:22:26] MM: Yeah. So it's right now just a web app. We don't have a desktop version yet. I've actually been thinking about sort of bringing it over to an electron app. But yeah, for now it's just a web app.

[00:22:38] JM: And what's the stack look like?

[00:22:41] MM: Yeah. So on the frontend we're using React. It's all pretty standard. We're not using any sort of UI frameworks or anything like that. It's all pretty custom. We wanted a lot of control over the whole interface and user experience. And on the backend, we're using NodeJS with Express. And also as I mentioned for all of those sorts of SQL database connections right now, we're using Sequalize to handle all of that, which makes our lives way easier than having to integrate with all of the individual SQL libraries.

[00:23:14] JM: Tell me a little bit more about your vision for the no-code future.

[00:23:20] MM: Sure. I think when you really look at internal tools within companies you start to realize that companies spend insane amounts of time and resources building these kinds of internal tools, which are effectively just very simple interfaces on top of a data source. And I think that's crazy that every company has to end up building these integrations, and authentication, and audit history, and searching, and filtering, and all of the different pieces of functionality into their custom tooling themselves.

And so with BaseDash, we're really just trying to eliminate the need for companies to have to build-out these tools in-house by themselves. And so there's going to be a lot of aspects to BaseDash, which sort of play into this no-code future. But I think we're really sort of still building on top of the idea that companies will still have a SQL database. They will still have code hooking everything up and we're just trying to provide that UI layer on top of that, which eliminates the need for companies to have to build custom tools themselves.

So we've got a lot of plans for sort of how that's going to look in the future beyond just sort of the data viewer and explorer. We do want to go more into the automation space as I was mentioning with workflows and triggers. And then we probably will build out some sort of like an

app builder or like I guess more select an Airtable blocks like tool where you sort of build slightly more custom data views on top of what we already have. So beyond just sort of the table view that lets you just view and edit data in bulk, we'll also want to have ways to have custom views where you can pull data in really easily from these sources and then write back to them again with this sort of, I guess, standardized read/write interface that we can build on top of your sources.

[00:25:22] JM: Do you anticipate more common interfaces between low-code tools? Or do you expect the kind of status quo of glue code through AWS Lambda or Zapier? Is that going to be the future?

[00:25:38] MM: I think that like Zapier has been really great in being able to hook up all these different tools together in the no-code space. And that's kind of in the glue for a lot of companies that are really focused on no-code. We're definitely going to be adding a Zapier integration at some point for BaseDash so that you will be able to add triggers so that when you change something in BaseDash, you'll be able to trigger actions on your API or on any other kind of integration that Zapier has.

I think that that's probably going to become more common within companies to have these sort of low-code or no-code integrations where you can really have non-technical people build their basic business logic. And I think that's already sort of starting to happen, where non-technical people can really use Zapier to hook up automations both for their own personal workflows, but then also for automations that need to happen at a company level. So I think that's going to sort of increase a lot in the future.

[00:26:44] JM: How big is your team at BaseDash right now?

[00:26:46] MM: Right now it's just me working on it. So solo founder. Yeah, one man team. Yeah, I've handled everything from engineering, to design, and marketing, sales, all that kind of stuff. But definitely planning to expand the team right now. We actually just raised our seed round. And yeah, we're going to be hiring some engineers in. So if anyone's interested, we're sort of recruiting and hiring right now. So you can check us out and apply.

[00:27:15] JM: What's on your feature list?

[00:27:16] MM: So I guess the big ones are going to be the workflows and integrations with all these different kinds of data sources. So figuring out how to build out that abstraction layer on top of these sources and automations on top of that so we can of triggers that push to all these different sources with also a Zapier integration as well. And then, really, I think a big focus as I mentioned is kind of like the UX of this product. So we're really kind of trying to build at this tool that feels amazing to use. We're really trying to sort of follow in these footsteps of like Superhuman or Linear, those kinds of apps that have kind of built this like cult following around them just because they have such great UX and product. So that's something we're really looking for.

[00:28:04] JM: Well, Max, is there anything else you want to add about BaseDash? Anything else you want to discuss?

[00:28:09] MM: Yeah. I think that's basically what we're working on right now. We've got a lot planned for the future. I think what we've got right now is really just sort of just starting out into this space. I think that there's going to be a lot of interesting things that pop up as we keep moving forward. But, yeah, that's sort of where we're at right now.

[00:28:27] JM: Here's one question. Have you thought about making it open source?

[00:28:30] MM: Yes. That's something that I have been thinking about. I haven't decided to take the leap yet, but I think that's something that we will keep thinking about for the future.

[00:28:43] JM: What's the cost-benefit analysis there?

[00:28:47] MM: Well, definitely the benefit there is really just for adaption. I think people just love open source and being able to sort of see how things grow and be able to contribute to that and build a community around that. And that's something that we really want to work on, is sort of the community aspect of building a tool like this.

[00:29:04] JM: That's the thing, is that there doesn't seem to be much cost.

[00:29:07] MM: Yeah. I guess sort of looking at how we're approaching the business model here a little bit, one kind of interesting thing, usually with open source projects you sort of have like the free tier being the host it yourself version of it. Where is then you usually charge for having like a cloud-based hosted model of the product. It gets a little complicated with what we're doing, because for us as an internal tool, on-premises deployment is usually what enterprises want. And that's actually where we expect to make a lot of money. So it kind of flips things on its head a little bit, because usually we would want to have a cheaper or free plan be the one that's used by smaller companies. But for internal tooling, a lot of the time that's going to be for the bigger companies. So it's going to be a little weird to figure out how we handle on-prem deployments versus the open source model. So we're going to have to think a little bit about how that would work in an open-source model.

[00:30:08] JM: Well, Max, thanks for coming on the show. It's been a real pleasure talking to you.

[00:30:11] MM: Yeah, thanks for having me. Great to chat with you.

[END]