EPISODE 685

[INTRODUCTION]

[00:00:00] JM: LinkedIn is an organization with thousands of employees. An enterprise of that size starts to develop problems with data collaboration. Data collaboration is the process of sharing and analyzing data with multiple users, such as data scientists, business analysts and engineers. How do data scientists know what questions to ask? How do business analysts know the right way to query a database? How does a data engineer even find where the right database is within the company's sprawling infrastructure, and how can these different users share information with each other so that redundant work is avoided?

When Adam Weinstein was at LinkedIn, he saw these problems firsthand. The process of accessing and utilizing data felt slow and broken. Engineers would have to search through the company's wiki to find out how to leverage data, and the wiki was often out of date. For example, you might have to figure out a query, the semantics of a query to access the right database piece which somebody else in the company has already figured out, and that's redundant work. When an engineer would leave the company, there was not this durable institutional memory of how that engineer worked with data.

Adam used his experience at LinkedIn as inspiration for Cursor, which is a tool for data collaboration. Cursor allows different users in the data pipeline to share datasets, queries, access patterns and comments about data within a company. Cursor is used by LinkedIn, Slack, Apple and other companies. Adam is the CEO of Cursor and he joins the show for an interview about the problems and opportunities of data collaboration. This is another show in a series of shows about the data systems problems within companies. We've kind of stumbled into doing a lot of shows about this, but if you're interested in this topic, we've done shows about Dremio, Outlier.ai, Uber. Uber's data platform solved some of these data problems at Uber. Not that they are solved completely. There are just perennial problems at companies in modern engineering. So it was interesting to hear Adam's side of it.

Before we get started, I want to mention we are looking for sponsors for Q4. If you're interested in reaching the 50,000 engineers that listen to Software Engineering Daily, I'd love to hear from you. You can email me, jeff@softwareengineeringdaily.com. You can also check out our website. We've got a form for – You could fill out the form and tell us what you're looking for, and we can get back to you about advertising opportunities. We're also hiring. You can go to softwareengineeringdaily.com/jobs if you are looking to be a writer or a podcaster with us, we've got some openings, and I'd love to hear from you.

[SPONSOR MESSAGE]

[00:02:53] JM: Nobody becomes a developer to solve bugs. We like to develop software because we like to be creative. We like to build new things, but debugging is an unavoidable part of most developers' lives. So you might as well do it as best as you can. You might as well debug as efficiently as you can. Now you can drastically cut the time that it takes you to debug.

Rookout rapid production debugging allows developers to track down issues in production without any additional coding. Any redeployment, you don't have to restart your app. Classic debuggers can be difficult to set up, and with the debugger, you often aren't testing the code in a production environment. You're testing it on your own machine or in a staging server.

Rookout lets you debug issues as they are occurring in production. Rookout is modern debugging. You can insert Rookout non-breaking breakpoints to immediately collect any piece of data from your live code and pipeline it anywhere. Even if you never thought about it before or you didn't create instrumentation to collect it, you can insert these nonbreaking breakpoints on the fly.

Go to rook out.com/sedaily to start a free trial and see how Rookout works. See how much debugging time you can save with this futuristic debugging tool. Rookout integrates with modern tools like Slack, Datadog, Sentry and New Relic.

Try the debugger of the future, try Rookout at @rookout.com/sedaily. That's R-O-O-K-O-U-T.com/sedaily. Thanks to Rookout for being a new sponsor of Software Engineering Daily.

[INTERVIEW]

[00:04:57] JM: Adam Weinstein, you are the CEO at cursor.com. Welcome to Software Engineering Daily.

[00:05:03] AW: Hey, thanks for having me.

[00:05:04] JM: I want to talk about what you're building at Cursor, but I'd like us to start with what you were doing at LinkedIn. So you were at LinkedIn for 3 years. You are working on business development and data products. You had some insights from that experience that led you to eventually working on Cursor. Describe what you were working on at LinkedIn.

[00:05:24] AW: Yeah, sure. Great question. My background is heavily in leading data at early stage companies. I was jokingly acquired in the LinkedIn, they bought a company called Bright where I led data 2014. When I joined the analytics function there, the first thing I was actually assigned to do is to help build out an analytics team in China. So China was an interesting operation, or should I say investment for LinkedIn and that it was built to be heavily autonomous. We didn't necessarily believe that we knew all the answers, and we also, from a legal and security perspective, wanted things to be relatively isolated.

So in doing that, we basically attempted to replicate a lot of the things that we had at headquarters in Mountain View at the time. So as the new guy, the irony was that I didn't have a lot of that understanding. So I spent a fair amount of time going around the business of jokingly said I had – There were over 200 coffee meetings just to try and meet folks that had understandings of various nooks and crannies of the business and x data around the business in order to really document it and then ultimately fly across the world and then teach it to the newly hired team in Beijing.

My role there at the time was as a leader in that analytics function and particularly focused on building out the business in China, but grew overtime into leading various aspects of data around the business. So when I was in BD, I led our developer platform, because it became sort of familiar with the legal ins and outs of data in our company. I ended up doing all the partnerships where teams wanted to use LinkedIn data, profile data, other data part of their product. But never really left the data realm entirely, just sort of worked on various sides of it.

[00:06:50] JM: So as you were getting a survey of LinkedIn basically from the top-down is what it sounds like, although you were able to drill in to specific areas as it was necessary. You were seeing some of the data access issues within a large organization like LinkedIn, and that's a large cutting edge organization. There are much more severe data problems at companies that have a little bit of a legacy infrastructure than a place like LinkedIn. What were the problems with data access that you saw at LinkedIn?

[00:07:24] AW: Yeah. Great question. You're right. We had a phenomenal engineering teams, a lot of teams – We had the luxury to be able to build another organization that may not be a technology company might not have that option. But the challenge really was access in sprawl. So we had roughly 300 folks on the core analytics team when I left. It's called business operations. It was sort of the more commonly used title. But there were roughly a thousand people a day that would write a SQL query inside the company.

So if you think about technical ability of a data user, right? On one end of the spectrum, you've got someone that can pull up a Salesforce report. On the other end, you've probably got a data scientist that can go right something that's machine learning-esk. In between, you've probably got a lot of people that are Excel and SQL savvy. Well, we had a lot of that in between.

So the challenge that we had, and we had data living in Salesforce is a great example. Maybe not so much today in a Microsoft world, but Salesforce your own data. We had a large data warehouse or somebody called it data lake, split between Hadoop and Teradata. Then we had a number of reporting tools that we use around the organization, everything from Tableau, to things that we had built internally that would support our product and a number of things in between, right? There were financial reporting tools. There were HR reporting tools. All of which had some value depending on who you were as a user.

So the challenge became where do you find an answer to a given question if you don't know where to start. If you were a new hire, if you're looking something that was out of your core competency, you could go ask your friendly analyst and they may or may not know. But there was no global repository that could capture all the sort of data work that was going on around the company. That was a challenge that yes we felt and it was difficult mostly given the quantity of data that we had, but I think also the quality aspect became an issue as well, which is if something's called a sales report, which one is accurate out of the 900 you might see in Tableau or in Salesforce? I don't think our needs were as unique as maybe we might have thought they were. They just might have been slightly different shaped.

[00:09:12] JM: Right. These data problems within enterprises, it's like every handoff from one person to another of a problem introduces a new set of problems. For example, there's problems at the bottom of the stack, where a data engineer needs to have access to a certain kind of data or they need to speed up a query of a certain kind of data. Then there's problems at the top of the stack where the VP of marketing just wants to know how a certain advertising campaign did, and that's going to involve this chain of a data analyst talking to a data scientist and a business analyst, and they're all talking to the data engineer and maybe there's data quality people in between. So there's all these different communication points from the top-down question, the VP of marketing saying, "I need to know how my advertising campaign did, because that's going to control the next $5 million of budgeting for marketing, because I need to know which marketing verticals did well. Should I invest more on billboards, or more in targeted ads, or direct mail? Without data, I have no idea how to make decisions." So these problems that manifest in between these different layers of the stack, which of these problems do you see as most acute?

[00:10:32] AW: Yeah. It's funny, you described it very well. Most acute is the – For my personal experience, is the actual sort of data work that's getting done. Whether it's code that's getting written or it's the translation from the question to the data answer. I think that's where a lot of pain gets felt. That's not to say that the actual communication aspect or the telephone game you described isn't also as painful for the business user that's on the receiving end of that. Actually, I think as a product, we ultimately intend to address both, because it's just as valuable to see, "Okay, here's the code that was written to answer a particular question as much as it is the conversation that preceded that actual code."

Ultimately, if you're thinking about business and particularly processes or functions, like that scenario you described about optimizing a marketing campaign, that gets replayed over and over and over again. Even though it may happen once and then go away for six or nine months, it's going to happen again even if it's just a subset of the original question. So there's no area that doesn't have pain associated with it as a data geek I guess you could say. The pain I've experienced most has to deal with how do you deal with the handoff from the business user to the actual code and then back as supposed to the conversation upfront. But to the business and to like creating an institutional knowledge base, there's value in both.

[00:11:49] JM: Then every 18 months per department or per layer of a sack, a new employee is rotating out or rotating in. So you're losing institutional knowledge. This is actually something that your products helps with, which we can get to eventually. But staying on this high-level topic of the org structure and just the workflow patterns that exist in these different enterprises, including companies like LinkedIn. We've seen the workflows change somewhat.

For example, we have seen the rise of the data engineer. We've seen newer querying tools, like Looker, and Periscope data. How have you seen companies change as a result of the increasing volumes of data, data engineers? On the plus side, there's more knowledge of – There's a wider spread of knowledge in terms of people who can actually work with things like HDFS and all the layers on top of that. But how have the volumes of data and the newer tools, how has it changed the org structure in the decision making within these companies?

[00:13:03] AW: Yeah, that's a great question. I think there's a couple of ways to look at it, right? If you go back 10 years or – So first company I sort of jumped to the data with was a company called Exact Target, which is now Salesforce Marketing Cloud. Actually, one of the cofounders at Cursor and I were both there together. He was my engineering counterpart. But the tools that we used then, so it was business objects, we looked at micro strategy, the time to insight as we used to call it. So from the time a request got made to the time the actual answer or the report or dashboard was delivered was in the order of weeks. We tried to keep it down to four weeks, sometimes it was longer, sometimes it was shorter. But it was never, "Hey, I want to go connect the data source, write a query, create a dashboard and then a couple of hours be done." That was very common back in 2000s early.

Then comes along Tableau, and more recently Power BI, and we'll get to look here in a second, and that made it the exact opposite, right? Anybody with enough SQL knowledge or ability to drag and drop some columns and apply some filters could go create a pretty visualization. The challenge with that came that the quality of didn't necessarily match the ease of which you created it. So if you didn't have deep data understanding, if you weren't the data engineer, if you will, if you were just a business analyst or just someone that was just impatient, which is a good thing by the way, and we'll get to that here in a second, you may have missed a couple of things along the way. So you might have created something that might not have ended up being exactly what you sought out to create. You forgot to exclude test users or didn't know how to filter test data or something like this. So you missed out on something.

Now you've had a wave of tools, Looker is a good example of one of them, where it's very balanced. There is some sort of approval or plans to it if you will. I don't know if they want to call it that, but there's some fact checking that's involved before the dashboard gets published to the world, and that's I think a healthy balance compared to maybe the old of the weeks and the new of a couple of hours and somewhere in between.

Where I see this going I guess is that you have companies that will probably have a myriad of tools, some that require extreme rigidity. So if you need to file something with the SEC, you probably need some compliance around it. On the other hand, other questions that might require just something that's directional in nature. So I think teams will have skillsets that match that. So you'll have businesses analysts, or I even call them like citizen data scientist that need to be more and more data fluent. They need to know SQL. I think the death of which is probably great that they're overstated. But they don't necessarily need to be full-fledged data engineers.

On the other hand, you'll have folks that can help assist further down the line if you need to dive deeper into something that are data scientists, or capable of going and diving into some Python or some Scale or some Spark or something like that. Yeah, we've seen both, and I don't think any of them are going away. I think the need to dive in the data is only growing, and that creates fragmentation in terms of the more people using it, the more content that gets created, the more content gets created, the harder to know what the truth is, and that's really an evolution of creation of data. How much is getting created and demand for it?

[00:15:48] JM: When you were at LinkedIn, you built an internal tool or an internal spreadsheet or something that was widely used within the company. What was it that you built?

[00:15:57] AW: Yes. This is a hack days story. A few folks on the team, we had this once a month thing called an "in-day". They'd be focused on various things. Sometimes it was getting out and cleaning at the community. Other times it was maybe you get together as a group of folks and try and build something that might be valuable to the business. After going through the process in China and flying back and forth, one of the guys I worked with, [inaudible 00:16:18], were like, “Okay, we’ve got all these queries. They’re sitting on a hard drive. How do we get this out to the world so A, our inboxes stop going nuts and B, we do something valuable for the business.”

So we built what was literally just a web form application. It had what is the title of the code? What is the code? I think it had maybe a thumbs down, thumbs down, was this still valuable? That was about it. It was not a query editor. It was not collaboration. It was not anything close to this or anything. It was literally just – It could have been an Evernote for it all mattered of the content, but we just couldn’t use Evernote, because security purposes.

So that was it, but we ended up seeing that hundreds of users are on the business started consuming from it. It got shared, and that was the point. Then folks also started contributing to it, which was nice. It was not fully-featured. We weren’t engineers. Hackers might even be the best way to describe it. But it ran on my desktop and people could use it. I don’t believe it’s still running anymore, but I do know it existed after I left, but I think they’ve – It’s gotten stale enough and the contribution count went down because enough folks that were there. Like you said, every 18 months, have rotated or moved on that it probably wasn’t as valuable as it used to be. But that was sort of the light bulb on my head that said, “Hey, there’s a need here.”

[00:17:30] JM: So to be clear, that light bulb was that if I’m somewhere in LinkedIn and issue a query to a database somewhere, it is useful to LinkedIn as a whole if other people within the company are aware of that query and can potentially reuse my query?

[00:17:48] AW: Yeah, exactly. So specifically to that question you asked. It was like, “Okay, we built this little database of queries and hundreds of people on the business were accessing it not even knowing who we were.” That was part A of the light bulb. But you’re right. Part B is that like if you’re taking time to write any amount of code, if you’re an engineer, that code goes into a place that the business can search it and discover it. If you’re an analyst, that code wasn’t going anywhere.

Yeah, you hit the nail on the head, that if I'm taking the time to do work in search of answer to a question, 98% of that is a value to the business. There are certainly times where it may not be. It's like, "Hey, I just want to – Give me 10 rows from this table, because I forget what the column names are," or something like that. Those queries aren't as valuable, but that's still is probably part of a larger answer where you're trying to assemble query to a bigger business question. So most code is something that should be captured by the business and that should be searchable for the next guy like you just said.

[00:18:39] JM: Yeah. I feel like I have worked at places where in my first week of working there somebody shares the wiki page or the Word document that has the list of queries and how to get the database, like how to get access to the database, how to access the database and the right way to get the answer to this query. It's just like there's like a spreadsheet or a Word document or some other non-tool unformed block of text that tells you how to interface with the database and retains the institutional knowledge of the data analysts that have rotated in out of the company.

[00:19:20] AW: That is sad, but true. Yeah, when I got to LinkedIn, we were in like the open source version of – I think it was Wikimedia is what they call it, internally is a wiki or knowledge base and had everything from vacation policies to exactly what you just described. There was a page with probably a thousand queries on it. 980 of which were so out of data, you couldn't even use them. I think it was written in 2008 or 2009. It was 2014, '15 when we were looking at it. It had never been updated. It had some table names and things like that you might make use out of, but yeah, the challenge is to take time out of your day when almost every employee in every organization these days is working to 100% capacity. Their slack time to go sit and think, "Okay, how can I go help the company?" even though that would probably a good thing.

To have something separate like that, it was benevolent of one person to do it at one point, but it rarely gets revisited with any sort of frequency. So if you can have it built into your workflow that, "Hey, you're writing code, it's automatically getting captured, being made searchable." You're doing yourself a favor by helping yourself to find it down the road, but you're also helping your teammates and the business as a whole.

[SPONSOR MESSAGE]

[00:20:34] JM: The Casper Mattress was designed by an in-house team of engineers that spent thousands of hours developing the mattress. As a software engineer, you know what kind of development and dedication it takes to build a great product. The result is an exceptional product. When you put in the amount of work and effort that went into the Casper Mattress, you get something that you'd use and recommend to your friends, and you deserve an exceptional night's rest yourself so that you can continue building great software.

Casper combines supportive memory foams for a sleep surface that's got just the right sink and just the right bounce. Plus, it's breathable design sleeps cool to help you regulate your temperature through the night. Stay cool, people. Stay cool.

Buying a Casper Mattress is completely risk free. Casper offers free delivery and free returns with a 100 night home trial. If you don't love it, they'll pick it up and give you a full refund. Like many of the software services that we have covered on Software Engineering Daily, they are great with refunds. Casper understands the importance of truly sleeping on a mattress before you commit, especially considering that you're going to spend a third of your life on that mattress.

Amazon and Google reviews consistently rank Casper as a favorite mattress, so try it out. Get a good night's rest and up-vote it yourself today. There's a special offer to Software Engineering Daily listeners. Get $50 towards select mattress purchases by visiting casper.com/sedaily and using the code SEDAILY at checkout.

Terms and conditions do apply. You'll get the select mattress purchases if you go to casper.com/sedaily and enter the code SEDAILY at checkout.

Thank you, Casper.

[00:22:40] JM: Is it this a problem that people have seen before and tried to address the problem of, "Okay, I'm writing a query for a database. It would be useful if that query was saved somewhere and if somebody else could use it." This must be a problem that other people have tried to address.

[00:22:56] AW: Yeah, I think there's been a couple of attempts to do things that, say, involved just storing code. Go back to like the company like Mode Analytics, which is a great product. They ended up building out more of a fully functional BI platform. So it's not just code anymore. It's very heavily focused on visualization reporting, and there are other tools out there that allow you just to sort of save snippets of code. I think where we see ourselves going with it is more focused on search. For analyst, search may be focused on finding that code that answers data questions.

For a business user, search is really, "Here's the question I have. Where does that answer live? Or if it's not within a system that I can just click on get access to, who is the analyst that wrote that code and how do I connect with them?" That is a little bit more powerful I think, or a lot more powerful, and that no matter where I live in the business, everybody had data questions of some sort, right? If I'm a sales leader, I need to know stuff about my pipeline, or my quota, or my territory, or where I'm at relative to my peers. If I'm in customer support, similar types of questions. But in finance I've got data question.

How do you capture the conversation and questions that are going on around the business and for that matter, the answers that go with them? On the other end of that problem, an increasing number of places where answers may live. So if you look at what SaaS tooling has done, you've got great tools that solve even incrementally smaller problems. If you think about the number of tools a designer uses today or a data analyst or a finance professional, they've got a tool for sort of every little piece of the job. But what that generates is that solutions or answers or reports that live within each one of those little tools. If we can get visibility in each one of them and most of them have APIs that we can do so with, we can provide a search layer that no one else is I think ever tried to build across all of that.

[00:24:43] JM: But the other element of the timing of your business relative to I don't know the other companies that have tried to build some repository of queries that people throughout the organization have been asking, because I think that's an element of Cursor, and we can talk about it in more detail. But just terms of the context, enterprises are realizing the value of the social component. I think Slack was a real pivotal change for people in this, because if you think about what is kind of generation one of like the social company, it's like maybe internal IM or like yamr or something like that or it's wiki-ing or it's Jira ticketing and that's kind of social, but it's not really leveraging crowd sourcing. It's not leverating social to the same extent that Slack does it. Slack kind of has a feeling of gamification. You feel a little bit rewarded for working within Slack, whereas yamr, I don't know if that was the case. But instant messaging, I certainly don't feel rewarded sending somebody an internal message, instant messenger.

But Slack feels a little bit different. Maybe Asana feels a little bit different. With Cursor, you've also got this social, this crowdsourcing feeling, or this contributor feeling. I think the social component adds something to it.

I guess we could just get into talking about Cursor. When did you start building Cursor and what problem does it attempt to solve?

[00:26:19] AW: Yeah. Great question. I agree on the social piece. Slack wasn't the first to do what they've done, but they did it in a way that got them over the hump compared to others, whether it was the IT hump, or the security hump, or the usage adoption hump.

[00:26:32] JM: The colors, the animation.

[00:26:34] AW: Yup. Yup. It's everything.

[00:26:36] JM: Emojis.

[00:26:37] AW: Yup, you got it. I mean, custom emojis, right? I mean, there was just enough viral there that it got through and was very sticky. We started Cursor officially last August, day of the eclipse actually. I think it was August 21st. The three of us started on that day. So Jason, Pat and I. But the problem we attempt to solve is I think we've discussed a little bit, but it's twofold, right? The beginning, we've focused on the analyst experience. How do you help analysts or folks that work with data across their business? Whether their title is analyst or it's data scientist or it's even data engineers in some cases, right? How do we help them be more productive by not repeating what's already been done before. For that matter, making certain that from a precision standpoint, they're using code that's been blessed or that they know to be accurate. So this motion of like a data dictionary and something that like, "Hey, you've got a definition for a given business term, churn, or sales, or revenue, or booking, or a new business, or a renewal business. How does that actually translate into code and how can you reuse these as building blocks across your sort of data universe?

That's for an analyst or for a data user. Our goal is to build a catalog of that code. But also provide a development that they can use, like you said, it's very social, it's very friendly. It's approachable. Whether you're a SQL expert or you're new to SQL and that can grow with the business, right? Whether it starts on one team and grows from there, or IT says, "Hey, this is your new too and everybody's got to use it, because we want to build this institutional knowledge base. It's flexible in that deployment model." That's the goal from an analyst perspective.

To a business user and sort of where we see the business going is – I think I mentioned earlier, is like how can you be search for all things data? We don't necessarily think we'll get all the answers and we don't want to go there just yet. I think that AI and machine learning is not to a place where you can – Even if you had full visibility to the data, you could actually predict the answer. But if we can say just like a Google search result would, here are 10 places where we think the answer might be, ordering it by relevance. We think we're seeing a much better chance of getting that right.

So to the extent that we have integrations or can connect to any place where answers may live, whether that's a BI platform, like a Tableau or a PowerBI or a Looker or whatever it may be, a micro strategy, etc., or it's a Salesforce report, or it's just an ad hoc query that an analyst wrote six months ago. How do we identify and connect someone asking a question with that answer, or with that source? That's where we see the business really going.

Today we launched the first version of the product in May. It's our free product to the world. It was designed really just to get people's feet wet with this notion of collaboration around queries, and we've continued to add quite a bit of enterprise functionality around that over the last few months. We'll be having what I would call a sort of an enterprise version here available to the world very, very soon.

[00:29:20] JM: Okay, the three word pitch that I heard you say there, collaboration for queries. To my mind, that seems like the best way of explaining it today.

[00:29:29] AW: Agree. Today, the product as it is exists, you hit the nail on the head. I think down the road there's probably search is a part of that three or four-word phrase, but today I'd agree that's where we are.

[00:29:38] JM: What would I be searching for?

[00:29:41] AW: So anything that would be relevant to that code. So we do a bunch of behinds the scenes to sort of abstract away just the terms of a SQL query. So if your SQL query is you've got table names, you've field names, you've got maybe a where clause in there. We also, as part of the product, ask you to give a plain text description of what it is that you're doing. A lot of that goes back to how do we begin to build a corpus of data and attach that to tables and columns, etc., around your business? So that's part of the cataloging process, if you want to call it that, is to be able to understand that.

If you're looking for – As an example, say a churn report in sales. We will behind the scenes go look at – If you search for churn report, we will look at anything that mentions any term related to that and identify what we think is the highest score if you will for a query that you typed. So if it's a snippet of code that's been written and maybe churns in the SQL, we'll find that. If someone gave a plaintext subscription that called something churn and we verify that by looking at the code and other folks have used that same terminology with that same table, or same set of tables, we'll prioritize that. But whatever you would use in any sort of query, right? Whether it's a Google query or just a search in a Google drive, you would be able to use the same thing in Cursor to try and find something.

[00:31:00] JM: It's just really useful, and the analogy that I sometimes think about with this or I was thinking about when I was preparing for this show, is how websites tailor their website layout to be indexed more easily by Google. So you have like a recipe website that will make their internal website index schema in a way that is amenable to Google's crawler so that when Google indexes it and Google looks through it and builds their search rankings, they can do it more effectively, or at least more effectively according to Google. This seems like a kind of tool where you would get people doing the same thing, because I want my queries to be accessible to other people in the organization. If I'm running an organization, I want people to be able to share queries with one another.

So if I could issue a SQL query on the command line or I could go into Cursor, and Cursor can issue the query for me and I'm just typing the same SQL code, if it makes no difference to me whether I'm in a dumb database access tool, like a command line or some free off the shelf tool for processing SQL queries, or if I'm in Cursor and it's like I'm entering Google searches and the search engine is getting smarter, or the database lookup system is getting smarter over time. Of course, I'm going to use the one that is going to get smarter over time.

[00:32:35] AW: And I guess you can say that's what we're banking on, right? Nobody wants to go figure out something from scratch that's already been done before only to realize that at the end or for that matter never realize it, right? There's not a high-switching cost there, right? Which could be a risk to us as well, but it something that there's not great solutions for, especially multi – That are both Mac and PC supported that have a web feel to it, right? We're built in Electron, just like Slack is. So it's got a web look and feel to it. You're right, part of the sort of SEO optimization that you described on the recipe site involves sort of the process of how you build it.

Today, when you go write a query, you literally are opening an editor. You're creating a new tab. You're just typing some code. Maybe if you're diligent about it, you might put a comment above that. So if you happen to come back to it yourself someday, you can at least know what you're doing. That's also something we've built into the product, which is when you go to do something, give us three words or something similar on what you're actually about to do.

Yes, it takes another six seconds of your time to do that, but the hope is that that six seconds, if you trade it in to the community of knowledge that you're able to access, should save you hundreds of times that the next time you go write something and you don't actually have to do it, you can just find somebody else as change a date range and you're done, and so that the social aspect as well as the knowledge aspect, right? That's a lot of what we're banking on.

[00:33:52] JM: Just to make people understand or have another chance of understanding this, give a few examples of how people use Cursor, or maybe you could give a day in the life of a data analyst or a data scientist with and without Cursor, or how it makes a difference.

[00:34:08] AW: Sure. Good question. So imagine you're an analyst attached to the marketing organization. This is real example by the way, real customer. We'll try and leave names out of it for security stuff. You're an analyst in the marketing organization. Your job is you sit in business meetings for some percentage of the day and then you tend to be back at your desk producing working on some sort of analysis. Maybe it came out of that meeting for the rest of the day.

So you'll sit down to your desk, you'll open up a query editor. You'll go look at some data in the data warehouse, or data lake, right? Maybe it's in Hadoop, maybe it's in Teradata, maybe it's in Microsoft SQL Server and you'll start crunching raw data to try and get to the answer to a question. Maybe the question was, like we talked about earlier, a campaign optimization mix. So a company spent a bunch of money, some of it is online, some of it is offline and they're trying to figure out what the ROI looked like. How do they identify what customers or revenue came in that was attached to that campaign? So you're joining spend data to revenue data. You're massaging it to make certain that people that might have been touched in multiple ways are accounted for. Folks that were existing customers that may have been touched again don't get credit to that, because they were already customers before.

Then as you're working through this code, and you probably have several queries that you'll write throughout this process, you're ultimately thinking about what is the undeliverable I want to create? Is it just an Excel spreadsheet that I want to send off on an email? Is it a Tableau dashboard or some sort of BI product? Then who is the audience that I'm trying to build that for?

So that person, they'll finally get the massaged to the right level and they'll get sort of a result set that explains what answers the question that they had. So they'll have input dollars, output revenue broken down by source. Then depending on that output, they'll decide where does it go? Most often, believe it or not, it ends up in an exported Excel spreadsheet that gets attached to an email.

Some small percentage call it 20 or less ends up in a BI platform. Tableau is higher. If you don't have access to a casual or lightweight BI tool like that, you're probably even lower than that. In that point, you'll share that result and that will be at the end of that analysis for that moment. Then what will end up happening is ultimately someone will come back and say, "Okay, can you break it down by week now, because we changed the spend thresholds throughout the time of the campaign," or someone else will want to go revisit that several months later, and their process will probably start at the beginning in a current world. But if you have something like Cursor, you'd be able to go say, "Oh! Hey, Johnny had already attempted this several months back. I could just reuse it. Maybe change the date range, change a couple of parameters and not have to start this from scratch."

I think what the person is doing, you're opening a SQL editor, writing some code, writing some more code, exporting the output, sharing it with some audience. That today is heavily local process, right? The SQL editor lives in your local machine. The data lives essentially in a database, data warehouse, data lake, but the actual work that you do never leaves your computer until maybe it's in that Excel spreadsheet or if you happen to share it somewhere publicly, like a Tableau or Power BI or some BI platform happen to be there.

Yeah, the challenge there is all that work that's kept local. Never could be viewed by anybody else that might need it. It's not in the Excel spreadsheet. We used to do that sometimes at LinkedIn. So people were nice enough to have a code tab. If you get shared the spreadsheet around, you'd include the code with it. But more often than not, that work is lost, and if anybody wants to go revisit it, they have to start from scratch.

[00:37:17] JM: When a company is onboarding with Cursor, do they need to have some concerted effort to get Cursor connected to all the different databases? What is the onboarding process for an enterprise?

[00:37:31] AW: Yeah. Today, if you go – And you could do this today actually. You can go to cursor.com, you could download the product. It's free to use and the version that you see on the website. The process is download it, install it, hook it up to whatever database that you want, just like any other data query tool, SQL query tool. No special effort or serer required. The collaboration layer or the search layer lives in the cloud. It's locked down to your company and you can control how it's locked down. So if it's relocked down to your team or the three users or however you want, you can manage all that much like a Google Drive or Dropbox, the effort to actually get content in the system.

So we'll automatically build a data catalog of any database that you can connect. So if you connect to a Microsoft SQL server, we will build a search layer of all the tables in that system. But if you want to start capturing code that's written, you would have to write it in the product, or we have an API that allow you to import it. If you happen to have a text file of 500 queries, we can actually import those through our API. We'll build a tool to ultimately do that for you. The challenge is that everyone store their things in different formats. So we're trying to work through something that's a little flexible there.

But yeah, I mean the onboarding is very lightweight. It's meant to feel like onboarding in the other sort of social product. If you go download Slack today, you install it, you're the first user, you're the admin, you can control how people get invited and how many challenge there are and who has access to what. But you can start with one. Cursor is very similar. One user can start using it. Then the next time somebody on the team has a question, say, "Okay, go download the tool. Search it. All my stuff is in there." That's mostly how we run the date, which is just this organic process and turnover within companies.

We tend to plant the seed. We'll reach out to someone and say, "Hey, here's what we're doing. Would love to have you come try to product," etc., etc. But typically, the value it's delivering is enough to get people to come use it and let the company grow from there.

[SPONSOR MESSAGE]

[00:39:21] JM: Accenture is hiring software engineers and architects skilled in modern cloud native tech. If you're looking for a job, check out open opportunities at accenture.com/cloud-native-careers. That's accenture.com/cloud-native-careers.

Working with over 90% of the Fortune 100 companies, Accenture is creating innovative, cutting-edge applications for the cloud, and they are the number one integrator for Amazon Web Services, Microsoft Azure, Google Cloud Platform and more. Accenture innovators come from diverse backgrounds and cultures and they work together to solve client's most challenging problems.

Accenture is committed to developing talent, empowering you with leading edge technology and providing exceptional support to shape your future and work with a global collective that's shaping the future of technology.

Accenture's technology academy, established with MIT, is just one example of how they will equip you with the latest tech skills. That's why they've been recognized on Fortune 100's best companies to work for list for 10 consecutive years.

Grow your career while continuously learning, creating and applying new cloud solutions now. Apply for a job at Accenture today by going to accenture.com/cloud-native-careers. That's accenture.com/cloud-native-careers.

[INTERVIEW CONTINUED]

[00:41:00] JM: So the client, the Cursor desktop client, it's an Electron app. It's a desktop app. In practice, is it similar to like just a client side SQL client or a client side Mongo explore? What is it doing to be able to query databases? Are there open source pieces of software you can use to just have this flexible query interface?

[00:41:26] AW: Yeah. I think it certainly has some of the bells and whistles you would expect with a query editor, searching for tables, having a full-fledged editor with type ahead and syntax highlighting and all that kind of stuff. But where it's different and sort or architecturally from an engineering standpoint, I think this is kind of the exiting part. We're an Electron client, but behind that actually sits a Java deamon, because web code can only talk directly to a database, right? So you need some background language there.

That Java daemon is basically a broker between any query that's being executed against the data source. The reason we have a client is because most data source is dealt on to live behind a company's firewall. So there has to be something local to be able to access that. We can't do it directly from our cloud because we're not going to get a firewall exception. That's not likely as a startup.

So that Java daemon, it actually gets the query, it gets the database. But then anytime you're actually searching, so say you want to go see if a query has been ran before and you pull Cursor and you go to the top on the search bar and you type, "Hey, I want to look for this report, or this code." That's actually querying our cloud and any code that's been published there. So we never see data itself. Data remains behind the company's firewall and either between the client and the database, but the actual metadata from that. So the column names, the tables names, the actual code itself. That does get shared securely within your organization as you defined, and that's what is being sort of handled by that, that Java daemon that's behind that electron client. So we don't know how many folks that are doing that that actually have a backend app effectively behind Electron. We think it's a pretty powerful paradigm, because you can do a lot of things behind the scenes there, from crunching, to caching, to even dealing with encryption and stuff like that. We see a lot of opportunity in that architecture.

Actually, if you take it a step further, there are going to be instances where we're going to need to deploy Cursor on-premise just regulated companies that can't use cloud services. Actually, LinkedIn was one of those. In those scenarios, that Java daemon can actually be deployed essentially within the business. You actually could use Cursor within a browser and that Java daemon just lives in one server in a company and it sort of brokers all traffic internally between databases and the actual search layer or the product. Does that make sense?

[00:43:34] JM: It does, yeah. Can you talk more about your infrastructure and the process of building Cursor?

[00:43:41] AW: Yeah, sure. Caveat, I'm the only not engineer in the team, but I'm in infrastructure nerd, so hopefully that helps a little bit. So yeah, the process was rather interesting. I think first debate was what language do we actually use a backend language both from a what's exciting and growing and has the right community around it, to, we're working with data and what restrictions does that impose no us?

So we ended up with Kotlin as a backend language, A, because it compiles to Java bytecode, interoperable with most things Java. Unfortunately, the fact that most databases talk Java are JDBC. We were a little reluctant to go with anything else, just because if the OEM is only going to support their Java derived driver, going with a community driver in any other language. Go is certainly one that was high in the mind, may have put us in some sticky places as we got further down the line with enterprises that had some really unique needs that maybe only the Java driver satisfied.

Kotlin lives in cloud layer, but then that Java daemon I talked behind the Electron app is actually Kotlin, at least in code. Then architecturally, how we designed the app to deal with a lot of security concerns that effectively that Java daemon has to be open only to our app and how do we make certain that there's no man in the middle risk or things that could go between any traffic that we might be generating either to our cloud or to our client such that this is some really sensitive information. Query resolves that might have a private company data. So there's a lot there from an encryption standpoint that we may be even overthought just given all of our backgrounds, me in LinkedIn, Pat at Salesforce, ExactTraget with me, and then Jason at Pandora at the beginning. Now as the team has grown, we all tend to be pretty security minded.

Yeah, I don't know if that helps at a lot. Electron seemed like the natural fit as a client. It allows us to build the code once and deploy it across platforms. Today it's just Mac and PC. Easily could add Linux to the mix and certainly will and potentially everything down the road too. I don't think we'll ever be writing SQL queries on mobile devices. That may be a bit extreme, I mean to have same people use like command lines on their cellphone, it looks painful.

But I do think there are pieces of content within Cursor that might be valuable to be searched from your mobile device down the road. If you're looking for an answer to a question, that search experience, there's no reason why that couldn't live in your mobile device. There's opportunities there that I think will be able to benefit from by using Electron in that regard. Does that answer sort of where you were going?

[00:46:08] JM: Definitely. What have been the biggest engineering challenges from those different domains that you just explained?

[00:46:14] AW: Yeah. I think the one I described earlier around getting an Electron app that has a Java daemon behind it and all the sort of nuances that come with Java as a whole. Does have the JRE installed? Do they have the version of the JRE that you build the product with? Electron has this ability to actually basically pre-fire a service as it starts. It's documented but not well used.

I think we think we're pretty unique or there's not a lot of folks that deal with that, that was certainly one challenge, right? How we bundle that Java app with the Electron client. Then thinking things through that go along with that into a typical enterprise software, like software updates. How do users – How does the product behave when a user may be disconnected. What can they see? What can they not see? What can they do? What can they not do?

For that matter, maybe they're behind the firewall but they don't have internet access. It's not uncommon to have companies that have a production data environment where they can VPN in, but that VPN connection consciously blocks off internet access. There's a lot of networking nuances that we've had to deal with. Some of it which we've already dealt with. Some of which we have yet to deal with. But there are things that we've needed to build in.

Then the whole transport layer I think that I talked about around how what we send back to the cloud and how we do that in such a way that it's only visible to the organization at most, but then it can be locked down by team, by user, however it may be in a flexible way that may or may not be known upfront. So you may run a query today and tomorrow you may change the visibility of that. How do you build a product that can support that? Very similar to a Google Drive or Dropbox, that they can do the same thing. You can create a Google Doc and then share it with the world and then turn that off the next minute, right? But thinking about that in terms of search and how do you build a model around relevance that may change dynamically. It got a little bit more hairy. That's a lot of worry to invest in. If that helps.

[00:48:07] JM: Yeah. Let's talk about the business perspective and the challenges there. So the space of data tools is very crowded, but on the bright side, the enterprises are typically willing to buy multiple tools often times with overlapping concerns as long as there's somebody in the organization that wants it. They say, "Hey, this is 100 buck, or this 1,000 bucks, or this is 50 bucks a month," and generally it just gets approved easily, because any improvement to the productivity of an engineer or a data scientist or a data analyst is fair game. So there's benefit and disadvantages to playing in the kind of data tools space. How do you look at the market and what are the go-to-market challenges and opportunities that you're seeing?

[00:48:53] AW: It's a great question. Part of the time that I spent between LinkedIn and starting Cursor was actually adjusting this very question, right? Because it's a crowded market. Differentiating yourself is probably harder than actually building a valuable product in some ways. What are you doing that's unique that everybody else isn't doing some part of?

So our model has actually been very bottom's up. So I think if you look at other companies out there that have taken the same approach, Tableau and probably Alteryx come to mind, is folks that have been really well with this, which is if you can get an analyst a tool that they can use on their own that will make them more productive, that will make their team more productive, that will maybe even highlight their value to their organization, you've got an in. You've got a start of a sales process.

So that's the way we've approached this. We've gone out and looked at anyone that we think deals with data, right? If you've got SQL on a profile somewhere on the web, or somewhere say that you've got a certification in Microsoft SQL Server or something that might be data related, you used Tableau, you've attended a conference, you give it a speech. We crawled kind of high and wide trying to find anybody that might fit that profile. The fact that we invited them to try the product, to-date that's been very successful.

So we've had roughly 500 companies or users from 500 companies start to use the product and that's growing. A lot of that is on us now to get the product evangelized internally and help that user spread the wealth if you will. But the goal has really been just let the person that's driving to us value, as you described earlier, like the analyst sharing code. Let them be the one that tries it first and then go from there. As supposed to being top-down and having an army sales folks call in and say to a VP or a CIO, "Hey, this is a product you need." Make the user look like a hero and then go from there. That's been, knock on wood, successful to date.

[00:50:43] JM: How do you expect the space to change in the next five years?

[00:50:48] AW: Yeah. It's a good question. The thing that's interesting, and I'm keeping a close eye on, is a lot of the regulatory environment. I think you've had companies that historically have been very slow to make data available to employees, right? I think it's out of a security need, there's huge concern or a huge apprehension around opening things up internally in a business even if it means increasing productivity, the risk that comes along with that is too great.

I'm keeping a close eye on what a GDPR or California's own equivalent of that are going to effect here in a few years, what that will actually do to tooling inside organizations. Beyond that, if you could assume that the tooling will deal with what it needs to, I think what will happen is that they'll be more and more folks in a company that need access to data and to raw data.

The story I kind of tend to tell is that if you go back 20 years and you're graduating college, you used to list Microsoft Office and Excel on your resume as skills that you had. Especially if you are well-skilled or versed in one of those, it was a valuable thing.

If you look at the growth curve of SQL and knowledge of SQL, and this is actually something I did at LinkedIn, a number of people adding in to their profile. It's increasing at a super rapid rate. I think there's roughly 15 million people in the world that knows SQL today. If you see industry surveys, that number is expected to grow somewhere between 40% and 60% over the course of the next 7 years. That's just because you can have as many reports and dash-boarding tools as you want, but a lot of questions still don't have a predefined answer. So you're going to have to dive into the weeds to get there.

I think that tooling that can help encourage that and get people started, and as they might say, like enable a casual or a citizen data scientist to go and be effective, I think will be of great value, and that's certainly the space we hope to play in.

[00:52:35] JM: Okay. Well, Adam, it's been really great talking to you. I've really enjoyed learning about Cursor and your perspective on the data landscape.

[00:52:42] AW: Oh! Thanks for having me. I really enjoyed it.

[END OF INTERVIEW]

[00:52:47] JM: I have learned a ton from QCon. QCon San Francisco takes place this year, November 5th through 9th, 2018, and I will be going for my fourth year in a row. I always love going and seeing the talks, and in between the talks I hang out and eat some mixed nuts, chat with other engineering leadership about the latest talks and stuff that they're seeing, the 50 different stream processing systems that we're seeing, the different databases we're seeing, and QCon is a place where senior software developers and team leaders and managers and senior leaders, they all gather together, and you have fantastic conversations. You have fantastic presentations, and it's extremely high quality. Its technical. A lot of it is also cultural and about management, and you can get $100 by using the code SED100.

QCon is a great learning experience. The presentations this year include 18 editorial tracks with more than 140 speakers from places like Uber, Google, Dropbox, Slack, Twitter. They are curated high quality talks, and you can get $100 off if you use code SED100 at checkout for your ticket. Again, QCon San Francisco takes place November 5th through 9th, 2018, and the conference is the 4th through 7th, November 8th through 9th are the workshops. You can go to qconsf.com to find out more.

Thank you to QCon. If you are going to buy a ticket, please use code SED100, and we really appreciate the sponsorship of QCon.

[END]