1

THE BIG IDEA

IT'S AMAZING how you can begin something with an idea that turns out to be a non-starter but end up in an incredibly promising place anyway. That's what happened in the early days of Snowflake.

The seeds for the company were developed and planted by Mike Speiser, a managing director at the venture capital firm Sutter Hill Ventures. Back in 2010, he started thinking about the next enterprise technology startup he could help launch.

Sutter Hill Ventures has an unusual business model. The more typical VC strategy is to search for promising startups that have already been formed and invest in them. Sutter Hill's partners do some of that, but they also analyze the tech landscape and try to spot nascent companies that they can incubate within their own offices. They even do initial engineering and design prototyping in-house. Once they've agreed to form a company in a promising area, they recruit technical founders with deep expertise. Then, off they go.

Before Snowflake, Mike was the founding investor for Pure Storage and served as CEO for a little more than one year. The company was a pioneer in replacing disk storage devices in enterprise data centers with faster, more reliable, and more compact solid-state storage devices—so-called flash storage. They based their storage architecture on flash memory chips instead of the traditional mechanical disk.

Even before Mike gave up the CEO role at Pure Storage, he had begun thinking about what he would do next. Because of his vantage point there, he was one of the few people in the world who understood the impact that flash storage could have on the enterprise data center. Computers would be able to access data 100 times faster than before. But traditional databases had been designed to run with disk storage, so he figured there was a huge opportunity to design a database optimized to run on flash.

Mike thinks of himself as being an entrepreneur within a venture capital firm. He's a founding investor and board member at no fewer than ten tech startups, including Observe, Sigma Computing, and Clumio. Earlier in his career he was a co-founder of Epinions.com, the pioneering consumer review website, worked for Symantec, and joined Yahoo through the acquisition of another company he co-founded.

When he's investigating a new business idea, his approach is to talk to a lot of smart people and see what they think of it. "Bayesian learning is at the core of all machine learning, and I'm a human Bayesian," he says. "I develop an idea, and then I go and talk to a lot of people and make my idea better and better. Or, if it turns out to be a bad idea, I drop it."

Over a period of nearly two years, Sutter Hill's engineering recruiting partner, Doug Mohr, introduced Mike to dozens of computer science professors and software architects who were experts in the storage and database fields. Some of them showed interest in his flash database pitch; others did not. None of them gave him a satisfyingly clear explanation for their conclusions.

Finally, Doug introduced Mike to Benoit Dageville, a native of France who was then one of the top software architects at Oracle, the largest database software company in the world. This was in early 2012. Benoit visited Sutter Hill's offices in Palo Alto, Calif., listened to Mike's pitch, and gave him some surprising feedback.

He called the idea "cute."

Mike had lived in France for a summer when he was in college. He had heard the word "cute" used in this way by French people before. It meant "not good," but in a nice sort of way.

Mike doesn't get hung up on being right. He just wants to arrive at the right answer ultimately. So he wasn't offended. He pressed Benoit for an explanation. They spoke for hours at the office and picked up the conversation later over a couple of dinners.

Benoit told him, "You're solving a problem that isn't a problem."

A BETTER IDEA

Benoit talked about how all database systems have two components—storage and computation. The key element of the storage component is the ability of the system to read and write data quickly. This is called Input/Output, or I/O. The computational part is about being able to take lots of disparate pieces of information and process them quickly—turning data into insights.

True, designing databases so they would have faster I/O, as Mike suggested, would be a good thing. There was value in creating a storage-driven approach to making the database perform better. But, Benoit told him, if you wanted to do something really dramatic—what would be a huge boon to businesses—you would focus on the computational side.

At Oracle, Benoit spent a lot of time talking to customers about their computing needs and their frustrations. Two things they complained

about frequently were the cost and complexity of computing. As the amount of data they handled exploded, and as additional types of data were added to the mix, they had to purchase more powerful—and more expensive—computers and software to deal with it.

A company called Teradata had made a mint selling computers and software that crunched data the expensive way. Teradata popularized one of the more important software concepts in the history of computing—the data warehouse, where data of all types could be more effectively collected, managed, and accessed for analysis.

But Oracle customers had told Benoit that they wished there were better and cheaper ways to process data. That, Benoit said, is where Mike should focus. A startup with brilliant technology might be able to double I/O performance in the database, but it might not be able to improve computational performance by a hundredfold.

At the time, cloud computing was taking off. Amazon.com's public cloud offered businesses the opportunity to offload the headaches of purchasing and managing computers and storage devices and of running their own data centers. Instead, customers paid Amazon and other cloud operators to handle these tasks. One of the advantages of cloud computing is that customers aren't constrained by the computer resources they own. Their ability to get computing jobs done on demand is practically limitless.

Mike and Benoit didn't arrive at Snowflake's core concept in a rush. There was no magical "aha!" moment. But, gradually, an idea crystalized: They would create a data warehouse in the cloud that had the potential to massively exceed what Teradata had done in on-premises data centers. They would offer customers a more effective and less expensive method for handling large amounts of diverse data—and there would be zero limits on storage or computing resources.

WHERE OUR FOUNDERS CAME FROM

Benoit was one of the leading experts at designing software to enable powerful clusters of computers to process data rapidly and efficiently. His specialty was parallelization. That's a type of computation where many calculations are carried out simultaneously. Large problems are divided into smaller pieces and farmed out to different microprocessors, then the results are combined. It gets computing tasks done quickly and makes efficient use of computing resources.

He grew up in France. As a teenager, he wasn't interested in computers at first. He owned a motorcycle and loved to ride it and tinker with it. But then a classmate brought a programmable calculator to school one day and showed him how to program it. The calculator asked Benoit for his name. He typed it in. It shot back: "Benoit, you are an ass!" He loved it. He was amazed that you could have a conversation (of sorts) with a computer. He became obsessed with programming. He later created a computer game, which he sold to a French company.

At Pierre and Marie Curie University in Paris. Benoit studied computer science and ultimately got a PhD in the field. His thesis focused on running the SQL data query language on parallel systems, processing massive amounts of data. Right out of university he got what he saw as a chance of a lifetime—to work on a pan-European research project, the European Strategic Program on Research in Information Technology. But in time he learned that he didn't like research so much. He would rather build powerful computing systems than write papers about computer science. So he went to work for Bull Information Systems, which, in the 1980s, was a significant player in Europe's computer industry

The center of gravity for the tech world was far away, in Silicon Valley, and Benoit longed to be in the middle of the action. He got a shot when Bull sent him there in the mid-1990s to work on a project with Oracle. After a few months, he went to work *for* Oracle.

It was a great place for Benoit. He got to work on some of the biggest challenges in data management and data processing. He soon was promoted to the role of architect, which, in the tech world, is a bit like being the wizard Merlin in the Court of King Arthur. You are frequently called upon by colleagues and customers to perform technology magic. Few people really understand what you do, but they respect you for it.

Yet as comfortable and stimulating as Oracle was for Benoit, he occasionally felt the tug of adventures beyond the company's walls. In the late 1990s, during the dot-com era, he was briefly lured away to a high-flying Internet grocery delivery service called Webvan. He was one of many of Oracle's talented computer scientists and engineers who were recruited to the dot-coms—many of which eventually went bust. After his fling with Webvan, Benoit returned to Oracle.

It was in the late 1990s that Benoit and Thierry Cruanes became colleagues and friends. Thierry got his PhD at the same university and in the same program as Benoit. He had grown up in a small farming town in the north of France. His first encounter with a computer came at about age 12 at an agricultural fair, among the tractors, cattle, and sheep. It was an Apple Lisa, the one that came before Macintosh. But, once he saw it work, he was captivated by the idea that computer scientists could build machines that could mimic the human brain. That was magical. Like Benoit before him, Thierry started writing simple programs for a calculator and progressed from there.

Throughout Thierry's career, he found one magical technology after another to learn about and master. He loved the database field because, "When you open a database, as an engineer, you find all of the computer science field in it. It's computer science in a box," Thierry says.

Benoit recalls that he and Thierry first spoke on the phone. It was a long-distance call between France and Silicon Valley. Colleagues at Oracle had approached Benoit. They planned on interviewing a job candidate from France whose accent was too thick for them to understand.

That was Thierry. They pressed Benoit into service to do the first interview. After Oracle hired Thierry, the two became close friends.

A decade later, Benoit was ready to leave Oracle—and he wanted Thierry to go with him. Benoit had been there for sixteen years, and, despite his professional satisfaction, he was frustrated that Oracle was slow to embrace cloud computing. Also, the company was seemingly incapable of scaling its systems to respond to the Big Data revolution—the explosion in the quantity and variety of data that was sweeping the worlds of technology and business.

Oracle's database division at the time was releasing a major upgrade of its software for on-premises computing about every four years. With cloud computing, software could be improved continuously. When Benoit started working at Oracle, engineers would spend about 60 percent of their time developing new products and features, and the rest of the time fixing bugs on existing products. In his last year, engineering was spending 90 percent of its time on bug fixes. That was incredibly frustrating.

Benoit felt that Oracle had become a dinosaur, and he was becoming a little dinosaur inside Oracle. He told Thierry: "We have to leave. We have to do something. We are dying here."

Thierry recalls the difficulty he was having hiring people for his product engineering team. At that time, he was a manager and Benoit was an architect. The brightest young computer scientists fresh out of university didn't want to work for Oracle. They wanted to be part of the Big Data revolution.

THE DREAM OF CREATING A NEW DATABASE

Thierry and Benoit began to meet at lunchtime at a table outside Building 400 at Oracle's headquarters campus in Redwood City, California. It was just off the San Francisco Bay, and typically there was a pleasant

breeze. Benoit drank one cup of coffee after another. They discussed the trends that were sweeping the tech industry—and seemingly leaving them behind. They saw how excited many in the industry were about Hadoop, the open source technology that many in the tech industry were using to manage the explosion of data, but they also saw the limitations of Hadoop. It was too complicated.

Meanwhile, they both started looking for jobs at other tech companies. They had no thought of launching a startup themselves at that point. Yet, amazingly, the first call Benoit got was from Doug Mohr, Mike Speiser's headhunter colleague. He was calling about a chief technologist job at one of Sutter Hill's portfolio companies. That one wasn't a good fit, but a week later, Doug called back and said Mike wanted to talk about something else: the flash database.

A few weeks later, after Benoit and Mike decided that they should focus on building a really big data warehouse in the cloud, Benoit talked to Thierry again. "I told him, there is this crazy guy. He thinks we can build a new database and it would be as big as Oracle."

This was a tremendously exciting time. After all those years of seeing Oracle customers suffer because of the complexity of the software, Benoit relished the idea of giving them a computing service that they wouldn't have to tinker with. And, as a software development guy, he looked forward to being able to constantly improve the software rather than updating it every few years. He also liked the idea of developing a database that could handle all kinds of information, not just the traditional data organized in columns and rows. By starting from a blank page, he and Thierry could build a new kind of database designed for modern computing. "It was a dream database," he says. "We had the idea on what to do, but we had no clue how to do it."

Mike was sizing up the business opportunity. The worldwide database market at the time was about $45 billion. The data warehouse piece by itself was over $10 billion. That segment of the market was growing much faster than the largest part, the relational database, because of the rise of

Big Data and the emergence of artificial intelligence tools with which to perform analysis. So it was a plenty big target for a small company.

Mike believed that these two French guys, accents and all, had the potential to do something that had never been done before. "That takes real vision. A real vision comes from somebody who is an artist. I thought these guys were artists," he says. Also, he liked them because they were not arrogant—unlike a lot of other startup founders he had met. Benoit and Thierry didn't think they were right about everything. They were earnestly searching for the right answers.

SNOWFLAKE IS BORN

At last, they decided to form a company. Mike was to be the interim CEO, and Benoit and Thierry would be the co-founders and chief technologists. Sutter Hill invested $1 million to get things started. Looking back from today, Snowflake's executives believe that one of the keys to the company's success so far is that Benoit and Thierry did not insist on being the business leaders of the company, unlike so many other founders of Silicon Valley startups. That enabled people with business expertise to lead while Benoit and Thierry focused on what they did best, innovation.

In early August of 2012, Benoit and Thierry resigned from Oracle and set up offices in a small apartment Benoit rented in downtown San Mateo. Their first purchase was a large whiteboard. Then they bought computers. Thierry liked Apple's Macs. Benoit didn't. Thierry won that argument. They mounted the whiteboard on a wall in the living room. "We looked at it and said, O.K., now we have to redesign for the cloud," Benoit says.

They spent weeks talking and writing on the whiteboard. They began by listing the issues that corporations faced. These were the problems they wanted their database to solve. It would be in the cloud so there

would be no restraints on capacity and it could run really fast. It had to be able to handle all types of data. It had to be self-service and really simple to use. Business units in a company should not have to ask the IT department to set up the system for them.

Initially, they thought they would build their own cloud rather than running the database on one of the existing public clouds. They sketched out a variety of hardware configurations, including, potentially, using flash storage. Later, they realized that they could get the hardware performance they needed from the big cloud vendors.

Benoit and Thierry were steeped in the knowledge of how to make traditional databases running in conventional data centers. But with this project, they had to rethink and reinvent the database—and how the software interacted with the hardware. "We almost had to forget what we knew," Thierry says.

For the first six months, it was basically two guys in a room working really long hours. Thierry did the cooking. Benoit drank endless cups of coffee.

In those early months, they made two important architectural decisions.

The first was to separate storage from computing. A company would maintain just one copy of all of the data it collected. The data would be placed in the cloud. The data warehouse would map that data and draw records from databases in it as needed. Any number of clusters of computers would be directed to access the same data at the same time.

This architecture had two primary advantages. It meant there was only one version of the truth within a company—not a mishmash of databases that might have outdated or incorrect information in them. Also, it meant that they could store a customer's data on an array of storage devices within a public cloud and charge the customer for storage separately from computing. They would charge mainly for data in motion. Stored data would be highly compressed and very inexpensive.

That encouraged customers to move all of their data to the cloud and to Snowflake's data warehouse.

The second critical design decision they made was to change the way they used computing resources. In conventional scenarios, many software applications and database queries tap into a cluster of computers. It's done that way so all of the users can access computing power concurrently. Nobody has to wait to get started. But the problem comes when too many users are online at the same time. The system slows to a crawl.

Thanks to the cloud, organizations can have access to practically unlimited computing resources. The cloud is "elastic," making it possible to assign a big computing job exclusively to an entire cluster of computers or even more than one cluster. No resource sharing is required. By designing for the cloud, Benoit and Thierry made a database management system that processes queries in seconds rather than minutes or days.

Startup founders have to pick a name for their company early on—sometimes before they're completely sure what they will ultimately make and sell. They know that whatever they choose will probably stick until they sell the company to somebody else or fail. So naming is ultra-important. These guys didn't want a run-of-the-mill name. They wanted something that popped. Mike suggested Snowflake because he, Benoit, and Thierry all enjoyed skiing and being in the mountains. Also, snow comes from the clouds. So Snowflake it would be.

One of their first hires was Marcin Zukowski. He was running a tech company, Vectorwise, which he had started in the Netherlands after he got his PhD from a university there. Its technology was based on work he had done in grad school—a method for improving the performance of databases called vectorized query execution.

Traditionally, databases respond to queries by fetching a single record at a time. With vectorized execution, the system fetches a large number of records in each processing cycle, rather than just one. Think of it this

way: You're a student buying beer for a party. In the liquor store, you don't select one beer at a time from a case and bring it to the cash register; you bring the whole case. As the quantities of data to be analyzed soared, the tech world needed more efficient ways of accessing it. This was one of them.

"We knew we needed this technology and we needed Marcin," says Benoit. "We also knew that the only way to attract a guy like this was to make him a founder, so that's what we did."

Performance was going to be critical for Snowflake. In addition to using vectorized query execution, the founders used another method for making the data warehouse ultra-efficient. It's called micro-partitioning. In databases, indexing and partitioning are common techniques used to allow a query to efficiently retrieve data. Indexes create lookup structures that accelerate record retrieval. Partitioning breaks up a table into manageable chunks to focus the data retrieval. Traditional databases require users to manually and explicitly specify indexing and partitioning strategies. The micro-partitioning technique automatically partitions the data in smaller chunks that can be more efficiently targeted. The result is that indexes are not required to deliver industry-leading performance. This technique enabled the Snowflake data warehouse to efficiently handle huge amounts of data, on the petabyte scale, while at the same time making the system easier to use.

Another critical element was simplicity. Late in his years at Oracle, Benoit had become disenchanted with the company's approach to dealing with complexity. Its software was hard to manage. The onus was on customers, or expensive consultants they hired, to constantly tinker with it. He focused on making Oracle's software more self-managing. His goal was to shift the burden from humans to machines.

This was the beginning of a change of orientation for him. He began to understand the need for what he calls "extreme simplicity." By the time he and Thierry were designing the Snowflake data warehouse, Benoit

had become an evangelist for making computing as simple as possible for the user. It was to become one of the foundational philosophies for Snowflake.

When Amazon first announced its full public cloud service in 2006, Amazon Web Services (AWS), one of its most remarkable attributes was its ease of use. Customers could visit the AWS Web site, select the services they wanted to use, and pay with a credit card. Benoit and Thierry knew that they needed to design a user interface that would be as simple as Amazon's. It would enable anybody in a company (with permission) to tap into the data warehouse as simply as somebody could sign up for AWS or buy a book on Amazon.

But an easy-to-use interface was the least of their challenges. Databases are extremely complex. In the Oracle world, a customer might employ a team of database administrators (DBAs) to keep each major database working smoothly. Database companies designed elaborate command-and-control systems for DBAs to use to tune the system for each task assigned to it and to deal with problems as they emerged.

The rule at Snowflake was there would be none of this. No "knobs" to adjust. The customers would load their data and they would query their data, but that was it. Behind the scenes, the system would manage itself. If there was a failure, the software would instantly replace failing components—and the end user wouldn't even know it happened.

A key element of simplicity was familiarity. Snowflake hoped to lure customers away from Teradata, Oracle, and IBM, so it would make the migration super-easy. For instance, customers would be able to use the SQL query language, as before. All of the significant changes in the technology would operate in the background, invisible to them. "From the outside, Snowflake smells like any other traditional data warehouse. It's only on the inside that we are fundamentally different," says Benoit.

They designed the architecture for the data warehouse and built its principal features based on their deep knowledge of Oracle's technology

and the SQL query language, which was the most popular tool for asking databases for particular bits of information.

THE "DIRTY DOZEN"

By early 2013, the founders had built an engineering team of about a dozen people with the help of Sutter Hill's Doug Mohr. Many of them came from Oracle, but Benoit and Thierry avoided recruiting their friends. "We didn't want them to take the risk based on our friendship," Thierry says. With each new recruit, they had to persuade a person who was enjoying a successful career in another company to leave all of that behind a join a startup that wouldn't have a product to sell for two years. They had to make these strangers believe in the potential of what they were doing as deeply as they believed it themselves.

Those recruiting experiences helped them crystalize their ideas and develop the sales pitch they used when they reached out to investors in their Series B round, in August of 2013. Benoit was worried that they would not be able to raise the money. The early months of Snowflake felt like one of those video games where you play and learn and achieve one level of expertise after another. "To me," Benoit says, "it was incredible that we were still in the game and still continuing." In October of 2014, they raised $20 million in a round led by Redpoint Ventures with Sutter Hill again participating.

The initial product development team—which they nicknamed the "Dirty Dozen"—was a multinational group. Its members hailed from 10 countries. When they gathered at work in the mornings, they would greet each other in their native languages. And they would try greetings in each other's languages. A Russian would try to speak Chinese. A French person would try Urdu. It was a lot of fun. But they also recognized that their strength came in part from their diversity—not just of nationality but of points of view. "It was a culture of everybody's import-

ant. It was a culture of helping each other," recalls Marcin, who was born and grew up in Poland. "We wanted people with lots of self-confidence but very little ego." In fact, recruits who proved to have big egos didn't fit, and they left quickly.

In the early days of the company, a culture quickly took shape that was based on the values of the founders. Those values included a focus on teamwork, a belief in egalitarianism, and the desire to think big and to break from computer industry orthodoxy. Nancy Venezia, who was the office manager and employee No. 8, credits the three technical founders with creating a culture that sustained the company for the first couple of years. "We had titles, but they really didn't mean anything," she recalls. "The founders laid the foundation. We were all in it together."

The early employees were excited to be in on the beginnings of something they thought could get really big someday. They called themselves Snowflakes. At that time the company occupied a small office space above a children's play gym in a little brick building near the railroad station in downtown San Mateo.

The culture of the company helped it cope with the inevitable crises that a startup experiences. A key part of data warehouse is the so-called metadata layer—a system for mapping where your core data is stored. Initially, Snowflake's engineers placed their metadata layer in the open source database management system MySQL. But after they began testing their data warehouse they discovered MySQL was incapable of accommodating the massive scale they needed. What to do? This could kill the company. They searched nearly frantically for a solution. It was all hands on deck. Finally, with the help of Sam Pullara of Sutter Hill, they found another open source database, FoundationDB, which had all of the capabilities they were looking for, and more. "It saved Snowflake. We love it," says Thierry.

As CEO, Mike led by example. One of the founding principles at Snowflake was that everybody should feel free to challenge each other's ideas—even those of the executives. Early on, Mike used an all-hands

meeting to make his point. He encouraged one of the engineers to disagree with him in public. "I want everyone to see that there are no sacred cows," he says. "You treat everybody with respect, but encourage constructive conflict."

As a result of his years of experience with startups, Mike understood the importance of urgency in a very young company. Everything had to be done quickly, especially software coding—building the product. Mike jokingly urged the programmers to "type faster," and that became a mantra throughout the company. Somebody would talk about a challenge. A colleague would say, "Type faster." It was a joke, but the point was that everyone had to contribute and work fast.

At the same time, Mike encouraged employees to feel free to just be themselves. In the workplace, people often adopt a guarded persona. They don't feel free to express their true nature, which, he believes, impedes their ability to do their best work. So he encouraged people to have fun in the office. For instance, as interim CEO, he worked in the office only two to three days a week, so he didn't have a permanent desk. Each time he arrived, he found a new place to sit. The company was growing so fast that at one point there was no desk for him, so they put him on a futon. On the day of the company Halloween party, members of the team propped a straw dummy on the futon and told Mike he had to work in a broom closet. Later, somebody put Mike's name on a Roomba robotic vacuum cleaner and set it loose on the floor in the office, where it banged into walls and chair legs. The Roomba was programmed to say, "Type faster." Mike says: "It was a way to make fun of the VC and management. I loved the Snowflake culture."

Of course, it was not all fun and games. The young company's list of challenges was far longer than the roster of problems it planned on solving for corporate IT customers. Shortly after Benoit and Thierry started whiteboarding in the apartment in San Mateo, Amazon announced that it was going to offer a data warehouse in the cloud called Redshift. At first, the news made Benoit and Thierry heartsick. But, quickly, they

realized that Amazon was placing traditional database technology in the cloud, rather than re-architecting for the cloud like Snowflake. In fact, Amazon's move would validate what they were doing by showing corporations that data warehousing could be done in the cloud.

Another early challenge for Snowflake was the fact that it planned on running its data warehouse *only* in the cloud. There would be no conventional software version that customers could operate on premises, and no "private clouds" where their data would be segregated from the rest. In 2013, most large corporations were not ready to commit to putting their most important data and operations in the public cloud.

REACHING OUT TO CUSTOMERS

Mike and other friends of Snowflake introduced the founders to technical leaders at a number of corporations so they could pitch their concepts to them and get feedback. It was through these conversations that they came to understand how organizations might use their service, and how it might change the way they used data. It became clear that customers would value the ability to have one version of their data that all could tap into—and that their business units and functional organizations would be more willing to share data with one another. Also, since there was no limit on computing and storage, people could get their computing work done quickly, when they wanted to do it.

One of Sutter Hill's core strategies with startups is to not wait until a product is nearly completed to start hiring sales people. They would hire early and send sales people out to make potential customers aware of what was coming but mainly to learn from corporations what they wanted in the product. After every one of these meetings, the Snowflake sales people would send an email to everybody in the company telling what they had learned. In that way, everybody had the same information, unfiltered. This was especially important for the product develop-

ment team. Mike calls this "collective noise cancellation." He says, "Every engineer could bring to bear their own creativity because they actually understand the problem deeply."

The first sales hire was Chris Degnan, employee No. 16. He had been a district or regional sales manager for most of his nine years in sales management, so he didn't have top-management experience. He had spent many of those years at EMC, the leader in computing storage hardware and software. Chris is not one of those sales guys who can sell anything to anybody. "That's not my style. I have to be passionate about what I sell," he says. "I have to believe in it. I believed in Benoit."

Chris's task at Snowflake was to land the first ten or twenty customers with an unusual proposition: If they gave Snowflake guidance to help develop its product, they would get founding-customer discounts. That meant they would be able to use the product for free initially and later would receive the best discount the company offered. In return, they had to be willing to vouch for Snowflake's technology to the press and other customers.

Initially, Chris spoke to potential customers mainly about the ways that Snowflake's data warehouse would be similar to the on-premises technologies they were used to—except it would be faster and cheaper. "We didn't want to freak them out by talking about all the novel features," recalls Benoit. So Chris didn't focus much at first about the truly differentiating aspects of Snowflake's technology.

Chris identified prospects by searching the job-search website Indeed.com for companies that were hiring engineers with experience in Amazon's AWS cloud platform. Then he and an intern searched by zip code on the professional social networking site LinkedIn for people who listed Amazon RedShift, its cloud data warehouse, as an area of expertise. These were breadcrumbs that led to bakeries—places that might need a cloud data warehouse.

Chris also targeted advertising technology companies, which he knew had a tremendous appetite for Big Data analytics.

The strategy worked. The initial handful of customers were willing to put up with the normal glitches associated with an immature technology. One of the early customers was using a $20 million system to do behavioral analysis of online advertising results. Typically, one big analytics job would take about thirty days to complete. When they tried the same job on an early version of Snowflake's data warehouse, it took just six minutes. After Mike learned about this, he said to himself: "Holy shit, we need to hire a lot of sales people. This product will sell itself."

Well, not quite. But the first two years at Snowflake were driven by boundless optimism that its technology could transform the way organizations use data. At its core, this represented a shared belief in the power of innovation. Thierry sums up the belief this way: "It's important to understand what you don't know yet and what you need to know. The creation comes from imagining what is missing. You need to dream things before you can create things that don't exist."

That's true for technology, and, as Snowflake's story shows, it's true for technology companies, as well.