

[VIDEO PLAYBACK]

- I'm going to make a lot of hand gestures. I'm going to do this with my right hand and I'm going to do this with my left. I'm going to adjust my glasses and then I'm going to ask you all a question. By show of hands, how many of you have all been asked a question before? OK, great. I'm seeing some hands, and again, I have nothing here.

Now, I'm going to react to that and act like I'm telling you a personal anecdote. Something to break the tension. Something to endear myself a little bit. Something kind of embarrassing and you guys are going to make an aw sound.

- Aw.

- It's true. It really happened. And now I'm going to bring it to a broader point. I'm going to really back in. I'm going to make it intellectual. I'm going to bring it to this man right here. Now, what this man did was important, I'm sure. But I for one have no idea who he is. I simply Google imaged the word, "scientist." And now, you see, I'd like it to seem like I'm making points, building an argument, inspiring you to change your life. When in reality this is just me buying time.

Now, if you don't believe me, let's take a look at the numbers. This is a real thing that's happening right now. The number of talks that I'm giving is one. Interesting facts imparted thus far in said talk, well that's

going to be a 0. My height in inches is 70.5, note the 0.5. Two times six equals 12, and then interestingly enough, six times two also equals 12. That's math. 352 is a three digit number. One, two, three, four, five, and then almost immediately following that, we get six, seven, 8, 9, 10.

Now, to add more filler here, I'm to give you a couple more numbers to consider. 18, 237, 5,601, 2.6 million, 4, 4, 24. Staggering. These are real numbers, all of them. And to follow that up, let's take a look at some graphs. Now, if you take a look at this pie chart, what you're going to see is that the majority far exceeds the minority. Everybody see that? Cool isn't it?

And let's take a look at this bar graph because it shows similarly irrelevant data. Now, I'm doing this because I'd like to make it seem like I've done my homework. If you were, say, watching this on YouTube with the sound off, you might think huh, OK, this guy knows what he's talking about. But I don't. I'm floundering, panicking, I've got nothing. I'm a total and utter phony. But you know what? I was offered a Ted Talk, and dammit, I'm going to see it through.

Now, if you take a look behind me, these are just words paired with vaguely thought provoking stock photos. I'm going to point at them like I'm making use both of my time as well as your time, but in reality, I don't know what half of the mean. And now as these continue, I'm just going to start saying gibberish. Wagawa, Gaba Gaba, turkey, mouth in a mouth, chip, trip, my dog skip. Rip it and dip it, Richard. I'm an itty bitty baby bopper and I'm hungry in my tum-tum. Brad Pitt, Uma Thurman, names, things, words, words, and more things.

And see, it feels like it might make sense, doesn't it? Like maybe, just maybe, I'm building to some sort of satisfying conclusion. I mean, I'm gesticulating as though I am. I'm pacing, I'm growing in intensity, I'm taking off my glasses, which by the way, are just frames. I wore them to look smart even though my vision is perfect.

And now I'm going to slow things down a little bit. I'm going to change the tone. I'm going to make it seem like I'm building to a moment. And what if I was. Amazing, isn't it? What can you do? Life's a roller coaster. You know, if there's one thing you take away from my talk, I'd like you to think about what you heard at the beginning, and I'd like you to think about what you hear now because it was nothing and it's still nothing. Think about that. Or don't, that's fine. And now I'm going to stop talking. Thank you.

[APPLAUSE]

[END PLAYBACK]

JERRY WINTERS:

OK. So that is nothing. Better stop that before we keep going here. And so, now that you know what nothing is, hopefully the time we spend together, you'll leave saying you felt like you heard something. For those of you who didn't see the full thing, this was a Ted Talk about five minutes long, talking about how to look or sound smart in your Ted Talk presentations.

The whole thing was about how people sometimes do all sorts of things that make it seem like you're actually getting something, when in reality, you're not getting anything of value. So hopefully, once again, you'll feel like you actually got something during our time together.

My name is Jerry Winters. My title now is grandpa. I had my first grandchild about a week ago and we're thrilled with that. Thank you. I didn't do the work. It was the mother that did the work. She carried the baby. She birthed the baby. She did a lot of work here.

And so this is my PowerPoint presentation here. This is the title screen and I have one more. This is it. This is the PowerPoint presentation. I don't like PowerPoint because as you saw in the previous talks, sometimes PowerPoint is used as a substitution for actual content and actual discussion.

So this is it. This is all you get for the PowerPoint, but hopefully the content that we show will actually be worth your while here. This class is all about-- let me go to my PowerPoint presentation again, here. This class is all about moving to the cloud for your database dependent AutoCAD add-ins.

Now, you may be thinking to yourself, it's actually database not database. I don't know. I use both interchangeably. I hope it means the same thing. But this is about taking an application that you've written in AutoCAD, that maybe uses a database of some sort, or a database if that's what your preference is, and taking that database and moving it to the cloud.

So what is the cloud? Any thoughts on that. What is the cloud? 10 years ago I would say what is the cloud and you'd start looking up. Right? The cloud, the cloud. We pretty much know what the cloud is, right? The cloud is a bunch of computers, really. So the cloud was invented by Al Gore, just previous to inventing the internet, and he actually invented the internet so that he could implement the cloud. That's where the cloud came from. So we have him to thank for that. We're grateful for that.

So the cloud is basically a bunch of computers and these computers are placed in different data centers or data centers. And they're here, they're everywhere, all around the world and that's what makes up the cloud. And so with the cloud, data in the cloud, oftentimes, is in multiple locations. Multiple storage locations so that depending on where you are you have a closer connection to the cloud.

Is anyone here from Europe? Anyone here from Asia? Anyone here from other places? If you go to a web site-- you're from Europe? Where? Sweden, wonderful. So if you go to Sweden and you try to hit a web site here in the States, is there any speed difference for you? Do you notice that? OK, and maybe it's changing a little bit just because speeds are faster. OK. I remember the time when you wanted to get on the internet you'd hear this

[SCREECHING CRASHING SOUND]

--sound. Right? You'd hear this crazy sound because your 14.4 and 4 K modem is dialing up, right? And then we got 56K. Whoa, how fast was that? That was amazing. And now we have people who make a web page and they put a 10 gigabyte photo of themselves-- yes, I said gigabyte, not megabyte. I understand the difference. They put this photo of themselves on the home page, and today we can load it up, and it comes up relatively quickly. Whereas-- well, a five gigabyte photo might be pushing a little bit but-- so yes, so speeds are faster and better in that sense.

So in the cloud you have the same data, in some cases, put in multiple locations. OK. So let's talk about the cloud and databases and let's get to-- it's just not showing up up there. I must have closed it. Uno momento por favor. What do I want? I want [INAUDIBLE] There we go.

All right. So we're going to talk today about four primary things. First of all, the pros and cons of cloud based database storage. So what are the pros and cons of cloud based database storage? I've listed a few here. We'll talk a little bit about more things here. So the first thing I have here is inexpensive startup costs. If you want to get an instance of SQL Server going, you've got-- let's pretend that you have nothing. You're starting from scratch. OK?

You've got to start by getting a machine. Right? A server of some sort. How much is that going to cost you? \$12,000. OK, you've got a better budget than I do. \$12,000, that's good. OK, so we've got a \$12,000 machine now, a server. Then we've got to get the actual database going. A seed or a license of SQL Server, which is going to cost you how much? \$800, SQL Server.

[? So we ?] got an operating system, right? OK. But SQL server, depending on what, you're looking at \$1,000, \$2,000, \$3,000, \$4,000, \$10,000, \$20,000, \$100,000, depending on what you're getting. Licensing with SQL Server has kind of been crazy. It used to be

bought a license of SQL Server and you had SQL Server, and it's not really that way anymore. OK.

Before all that, the startup costs is convincing someone you have to do it because you're getting the prices for your machine and your software, and you have to convince about 10 people that you have to do it. But before you can convince those 10 people you have to do it, you first have to go through this thing called the IT department.

Now, if any of you are from the IT department, well, this isn't a negative reflection on you. I was going to say it's not a negative reflection, but it is because those of you that are not in the IT department, you know how difficult it is to work with people like him. OK, because you want it. You see a need for it and you want it, and so you've got to convince the IT department. And what does the IT department say?

What do you say when someone tells you, hey I need this software? You say-- what was that? For what? Right. And now, we start the questions. For what? Why do you want it? Why isn't this other software good enough?

AUDIENCE:

I want to try something and they've got to have a meeting to find out what our capacity is going to be using in 10 years from now. So we buy the right server now.

JERRY WINTERS:

OK, right. Or with the virtual machines and that you already have the server. Right? And it's a question. OK, well, I've got cores now that you're saying you want, and I have to determine now, am I going to allow you to use these cores? Am I going to allow you to use this RAM, let alone the storage space? I've got to add storage space my virtual server? No way. That's just a crazy thing.

So that's a cost associated with using SQL Server and the benefit of the cloud is-- IT department, plug your ears-- you can bypass the IT department. OK? Yes.

AUDIENCE: [INAUDIBLE]

JERRY WINTERS: What was that? Sorry, what was that?

AUDIENCE: [INAUDIBLE] bringing legal into.

JERRY WINTERS: No, I'm bringing legal into it. Right. You can usually bypass the IT department because the IT department still has an iron clad grip on this thing called the firewall. Right? OK. So there is one element there that usually is not an issue.

OK, usually not an issue. But if you're a developer, and hey, I just want to try this out. You can do that because you're not dealing with now \$12,000 for this, \$8,000 for that, red tape, meetings, having a special session of Congress to get an instance of SQL Server just because you want to try something.

As we'll see in a minute, it's basically a matter of I've got \$5, I want to try something. I'm going to withgo my coffee this morning. I'll still have my coffee in the afternoon. I'm going to withgo my coffee this morning because I want to try something. So number one, start up costs, inexpensive. Number two, globally accessible.

Some of you have talked about being in the field or having multiple offices and that. If you have an instance of SQL Server running in a machine in your office and you want to make it available to the world, what's the IT department going to say? No. Right? Are they going to say something other than no? No.

You can try to blackmail. You can hold crazy pictures over their head of things that happen at Autodesk university, and they're still going to say no because when you open up a pipeline to a local server in your system, you're opening up a lot of problems, potentially.

Now, I can appreciate that. I can understand that. I can accept that kind of a no answer. But making it globally accessible, it is, by its nature, globally accessible. Now you can close that down on the cloud as well. You can say I only want IP addresses in this range to be able

to access this. So the fact that it can be globally accessible, and normally is globally accessible, does not mean it has to be globally accessible. It can be pared down a little bit. OK.

SQL server, Express to Enterprise compatible SQL and .NET code. So if you're going to move to the cloud or you want to move to the cloud, you don't have to rewrite 10 million lines of code. There's something called a connection string, and you change the connection string, and you run the program, and it works.

AUDIENCE:

Even if it's not written in .NET?

JERRY WINTERS:

Even if it's not written in .NET. Well, what's it written in?

AUDIENCE:

VBA.

JERRY WINTERS:

OK, yes, VBA, same thing. Yeah, VBA, same thing. I was going to say list-- I don't know if list could take a connection string, it might work. I've never tried that. I don't know. It works with list? OK. It's a connection string. The connection string is saying, this is the driver to use, this is the location of the database, whether it be a URL, a URN type thing, or a IP address, and then credentials, username, password, that kind of thing.

All right, number four, bragging rights. OK hey, I'm on the cloud. Check me out, I'm on the cloud. Yeah, I just have a fully replicated SQL Azure database I put on the cloud. It's accessible now to all the countries in the world. And the cool thing is with that is that you can be a-- don't raise your hand, OK, don't raise your hand. You can be a 34-year-old guy living in the basement of your parents' house and you could actually still do this. Or gal, although it's probably more prevalent with guys. OK.

It can be implemented without IT department red tape. Now, I'm not saying bypass the IT department every time, or all the time, or even ever necessarily. But the fact that you can actually implement this and get it going without having to go through a bunch of them, is nice.

Multiple geolocated locations and options. So when it comes to cloud stuff, you can basically say, I want to have my server on the West Coast. I don't have my server on the East Coast. I want to have my server wherever you want to have your server.

Now, why is that important? I've got a friend slash client that is in Alaska, and in Alaska they have connectivity issues in some of the areas that they live in. We joke about 56K bandwidth, they dream about 56K bandwidth. OK. So sometimes that might be an important thing. I might be able to set one up in Sweden that I might be fine accessing from Utah, but my friend in Alaska might have a different experience.

Are there cons? The primary con to me is surrendering control. I like a box sitting there. I like to know there's a drive in that machine, be it a spinning drive or an SSD. I like knowing that it's right there. I like being able to take a thumb drive and stick it in that machine and copy my files down to create a backup like that. I like that. OK. I couldn't even tell where the machine is that this stuff is. I couldn't.

I could tell you where some are in Utah, but that's the NSA building and I don't know if they'd want us to knock on their door. But I couldn't tell you where that is. So surrendering control, to me, is one of the more difficult things. Hey, I want to backup my database. No problem. Boom, here's your backup. Hey, I want to backup my database. OK, I can get that to you, but it's going to take a little time, especially for a larger database on the cloud.

All right, and the last one here, performance. What's it going to be? I know that I can go to the Azure portal and I can create a new database. And we're going to do that today in this class, but what is it actually running on? Are there little hamsters running on a little wheel to get this thing going? Because I know that I can build a machine, and I've done this. I can build a crazy very high performance machine, and I know what I've got.

I know that if I query a database that has let's say 20 million records in it, I know that if the performance isn't the way I want it to be, I've got a machine there and I can put some more stuff in it. Right? I can get a different processor with more cores. I can get an additional processor. I can boost up the RAM. I can do things to improve performance with that particular machine, but in the cloud I don't know what it's running on. And I can probably pay more money to get a little bit better performance, but I still don't know what I'm going to deal with.

Now, I've got two cons listed here. If anyone's aware of any more cons, they probably are out there, you're welcome to bring them up. But those, to me, are the two primary cons to moving to the cloud. And for some of you that might be a deal breaker. I need to have control. I need to be able to grab onto this thing. Now, the downside of that is this.

When I go into a client's place or the software development, one of the questions they ask is security. What are we going to do about security? And that's a legitimate, valid question. Having a box that's there, and this is why I bring it up to them. Yes, we'll deal with security. Yes, we'll take care of that. Yes, your people should not be sharing usernames and passwords. That would be a good thing, to not share those things.

But I bet if I came to your office at 7:00 in the morning, I could walk in the front door, walk to their cubicle, unplug their keyboard and their mouse, unplug their monitor and walk out the door with their computer. And that's probably true in a lot of offices.

Yes, there are some that require some kind of a key code to get in and that kind of thing. That's great, but in a lot of offices-- and it's interesting because the people that are most concerned about security on the software side, are usually the ones where you could walk through the door at 7 o'clock in the morning, and walk into someone's cubicle and unplug their computer, and walk away.

Or worse, walk in their front door in the morning-- and don't try this at home, please. Do not try this at home. But walk in their office at 7 o'clock in the morning and go to their server room, undo a couple of screws and pull out a rack. OK? And yet they're paranoid about security on the software side.

So one of the benefits-- I didn't put that up there-- is the fact that there is not a box where someone could potentially do that. So alarm systems, things like that are great, but not having a physical box is also a benefit. All right. Any questions about that? Yes?

AUDIENCE:

If you're storing it on the cloud, how are updates handled? If the cloud is handling all of the database updates, and what if there's a conflict? It's always current, is that correct?

JERRY WINTERS:

Great question. So the question was, if it's on the cloud, how are updates handled? So if you have a physical machine, once in a while you'll see something that pops up, and based on your settings it will say either a restart was required-- right, that kind of thing can happen. Or updates are available, that kind of thing.

So when it's on the cloud, in this respect here-- OK, when we're talking about the SQL Azure platform, updates are seamless. You don't see them happen. It's just there and it's continuously running. And the uptime is one of those 99.9999% uptime kinds of things.

Now, on the cons-- let me add one more thing to the con, OK? We have these people out there-- well, you've heard of Wiki leaks, right? We have people out there that are hacking things and trying to hack things. And one of these days I imagine something may happen, or something may or will get hacked on the cloud. That's another one of those things.

If your organization is being hacked, I can pull a CAT5 cable and I'm good. I can recover from that. If the cloud is being hacked or something's going on there, with the rest of the world-- so we're in good company. But with the rest of the world, we're basically sitting

there and just kind of waiting for things to resolve. So on the security side, that's one of the other issues. But updates are handled seamlessly. All we're dealing with here is data that's up there. Yeah?

AUDIENCE:

So the cloud doesn't have its own backup?

JERRY WINTERS:

So backups, when you use the cloud, your database is normally replicated to multiple locations so that if one machine went down, the other machine would basically instantaneously be up. So that's what the uptime is the 99.9999 type percent.

And the reason they say that is because they'll say, well, no one's perfect. Something could happen. They're not saying it happened. Or they'll say because we had an issue with this server, it went down and it took one second for this other one to be up and available. But the infrastructure was built entirely around uptime. Yes?

AUDIENCE:

I think that's important or is a good question is that Microsoft 365 and all of these copied platforms, none of them backup. They replicate. So replication and backup, definitely not the same.

JERRY WINTERS:

Yeah, they're not the same thing because a backup you say, I want to go back to the data as it was yesterday or the week before that. Right? Replication is replication, which means drop table users-- what does that do? It drops table users. Now this is an Autodesk conference, and Autodesk has this wonderful command called undo and the cloud hasn't got that command yet.

So Control Z, you can hit it all you want. It's not going to back out of what had just happened. So you can backup your Cloud database, and I recommend that you do it. You can back them up, but one thing to keep in mind is the fact that-- as was mentioned, thank you-- that it's replicated and so, it's not the same as backup. You can't back them up. You can get copies, basically, of that backup. And I recommend you do that. All right?

Good. Any other questions about pros and cons of cloud stuff? All

right. I think you mentioned we're moving to the cloud. Is that what you said? What does that mean to you? We're moving to the cloud.

AUDIENCE:

[INAUDIBLE]

JERRY WINTERS:

OK. So you had a physical server in the building, now you're moving to the cloud, which was still your storage, your file shares, that kind of thing is moving to the cloud. OK, very good. All right, and then I've got a class on Thursday about using the storage in the cloud for Azure. So if that's of interest to you, you might want to attend that class as well. And that's on Thursday. OK? All right.

Let's move on down here. Actually, just move on up here. How to create a new Microsoft Azure account from scratch. All right. So if you go on the web, the world wide web-- which one do I want here? I want this one here. And you Google create SQL Azure account. Just Google that. The top thing that comes up here is Azure SQL.

Now let me just make a quick note here. Microsoft has SQL Azure. It has its cloud database. This is not the only cloud database platform out there. Oracle has their thing. Amazon has their thing. There are different platforms that are out there. Today, we're focusing on the SQL Azure. All right?

You'll notice this little link up here. It says free account up here and it says start free here. So to create a new Azure account, basically, you just follow the links. When you create a new account, it says right now at least, you get a \$200 credit. That's not going to be a gift card they send to you, by the way.

That would be cool if it was because I could just create lots of email accounts. Hey, I'm a new guy. So all you do is you create this little account. It starts the \$200 credit and then from there you basically just jump into creating databases, which is going to be our next step. All right. Very, very simple. Very straightforward.

Once you log into your Azure account, you're going to see something

like this. All right, if I click on the plus symbol up here, this is my account I've logged into here. All right, what do you want to have a new what? A new compute? A new networking? A new storage? Web or mobile, databases, internet of things. Sounds like a Dr. Seuss thing, doesn't it?

OK, we want a new database. I'm going to click on database. What kind of database do you want? I want a SQL database, S-Q-L. Next question. What's your database name? Now, I know it's not AU 2017, but I'm going to create an AU 2017 because I have an AU 2016 already. OK. And resource group. West US. Blank database. Do you want to use a sample? What kind of database do you want? Let's do a blank database.

Server. Where do you want this thing to be? Well, I'll stay here with this west US. Pricing. Let's take a look at pricing here. When you start a new SQL database, you want to have a SQL database, basically if you're just trying something out, then depending on what you're dealing with, this first tier here, this basic might be good enough for you. This tells us we can have up to 2 gigabytes of database. That's your database size. OK? Now how big is 2 gigabytes? It's not as big as it used to be. Right

I have an 8 megabyte flash thumb drive back in my house. I didn't bring it with me here. But 8 megabytes used to be considered pretty cool. Before that, we had a 100 megabyte zip drive. Who remembers those? Anyone remember those? OK. Before that we had-- what did we have before that? We had the 1.44 megabyte floppy drive, which is why the 100 megabyte zip drive was so monumental. Before that we had the wider, bigger 5 and 1/4. That was a real floppy drive. That's where the word floppy came from.

For those of you who have no idea what I'm talking about, sorry. Some of you are getting it. So 2 gigabytes can store a lot of stuff. And remember, this is not pictures and video we're talking about. It's just data, unless you're using blobs to store pictures and video. In which

case, 2 gigabytes won't go very far, but this is just data. And you'll notice for \$5 a month I can have up to 2 gigabytes of data storage. OK?

AUDIENCE:

If you have a server, do you not have to purchase a server?

JERRY WINTERS:

No. \$5 a month. You've got a database on the cloud that you can program against and your clients can use, or your people off site can use for \$5 a month.

AUDIENCE:

[INAUDIBLE] I thought through this you'd have to have a server to put that database on.

JERRY WINTERS:

Nope. Not the case. You're buying the server with \$5 a month. OK? Good question. What was that?

AUDIENCE:

[INAUDIBLE]

JERRY WINTERS:

Yeah, it's running on a server. There's a machine out there. But frankly, the machine that's out there, I'm not getting my own server. There's 200 million other people that have a \$5 month account here. Which goes back to one of the cons, which is performance. I have no idea what I'm going to get. But for \$5 a month I can get a database on the cloud that I can now program against so my clients can use. Not bad.

They've got this thing called DTUs here. I can get 5 DTUs. If you want to impress people just DTUs, don't try to explain what it means because you won't be able to explain it because I can't explain it. Basically, it's basically a computational value that talks about how many processing computational things that you're getting with that. So if you want it--

AUDIENCE:

[INAUDIBLE] how ever much this grows, they're guaranteeing a certain amount of performance?

JERRY WINTERS:

Right. Yeah. So if you have an option of 5 DTUs versus 100 DTUs, you might say well, 100 DTUs is better. And maybe you need that

processing time. Maybe you don't need the space necessarily, but you need the processing time. So this is kind of the way it works here. Now, you'll notice here that even at \$5 a month you have georeplication.

And we talked about that a little bit. It basically means that my database is going to be the data and my database is going to be in multiple locations, so that if one data center gets hit by a bomb or the power goes out, then the database is not going to go down because of that. OK? So that's \$5 a month.

All right. Now you'll notice that's a basic. You've got the standard, here, S2, S3. It's all about price and what you're getting in that, generally speaking, if you look at the comparisons here, you're looking at, basically, how much storage you need. You've got up to two gigabytes and then we jump to 250 gigabytes. Big difference there. Even at the S0 level here, at \$15 a month I get 250 gigabytes, which is a lot of data. It's a lot of data. All right?

I know we have a tendency to think in hard drive. You're thinking to yourself, I can get a one terabyte drive for whatever. Well, one terabytes a lot of data. Now, maybe some of you need that, but most of us don't need a terabyte of database storage here. So notice the primary difference here is the DTUs. How much processing time, processing power you're getting behind them.

AUDIENCE:

Can you change the plan?

JERRY WINTERS:

Yes.

AUDIENCE:

So if you start out on \$5 a month and it works really well, and all of a sudden everybody wants to be on it--

JERRY WINTERS:

Yeah, you can change your plans. You can bump it up. Yeah, otherwise what are you dealing with? You're like OK, I started with \$5 a month and I created this cool thing called Facebook, but a lot of people need it now and so what I really need is this, over here, where

I'm spending \$16,000 a month.

So you know Microsoft's not going to say, sorry, stick with your \$5 month plan. There's your there's your terabyte database, there. There's your 4,000 DTUs. Talk about bragging rights. That would be really cool. All right. Well, let's just create a S0. I can spend \$15 a month for. Yes?

AUDIENCE:

[INAUDIBLE] I can't be sure I'm reading it right. Point in time and auditing. What are those?

JERRY WINTERS:

That's a great question. The point in time-- what was that?

AUDIENCE:

[INAUDIBLE] your backup button.

JERRY WINTERS:

Yeah. Well, point in time restore. I haven't looked at that to be honest with you. Let's see what it says here. I'm clicking. That feature must not be enabled. Great question. I'll look that up and give you a detailed description in the handout that you'll be able to download today.

OK, it sounds to me like point in time restore, you can create backups and to me, a point in time restore would mean that you could actually say, I created a backup, I want to go to that point. I haven't had to use it. It's one of those things that if you don't have to use it-- there's a spare tire in the trunk. I have no idea what it's there for. I don't know why it's there, but that's most like what it is.

Auditing, I can do that is. You'll see a page here that basically tells you things about uptime and what servers are being used for the replication, and how much is being used. That's the auditing. It's basically how much is being used. So if you create this amazing application, and you put it out there, and you're thinking it's awesome, and no one uses it, you'll know.

AUDIENCE:

Point in time restore is like a VM snapshot.

JERRY WINTERS:

Yes, but VM-- yes, on the data side of things. Remember, you're not getting a machine. You're just getting database storage here. Yes.

AUDIENCE: And you're paying for the storage. You don't pay based on the number of users that are connecting to?

JERRY WINTERS: That's the DTUs. So to a degree you are paying for that. So you'll notice if I come back-- I'm still here, OK. So we have \$15 a month will get me 10 DTUs. Notice the only difference between S0 and S1 is the number of DTUs. Right? And that's not going to let me bring up a good description of that either.

So 10 DTUs, 20 DTUs, 50 DTUs, 100 DTUs. Pretty much, they're all the same thing except for how much is being used. And Microsoft will let you know if there's a problem. They'll let you know, hey you're kind of exceeding your stuff here. We need to bump you up to the next level plan.

AUDIENCE: Is there any calculation between the number of users compared to DTU?

JERRY WINTERS: It's not the number of users. It's the amount of bandwidth that's used to access it and what you're pulling down. Did you have a question?

AUDIENCE: [INAUDIBLE] answer this. It's not based on number of users.

JERRY WINTERS: No, but the thing you want to understand is that you don't want to create-- when you have a local SQL Server database, you might find it useful to give permissions to people on your Active Directory network or whatever. When you go to the cloud, you don't want to create a new login for every single user that might be using it. You want to take care of that through your software and things like that. And we'll go through the process of creating users, and permissions, and that kind of stuff while we're here. Yes?

AUDIENCE: If you stop the subscription, your files will be deleted [INAUDIBLE]

JERRY WINTERS: Yes, if you stop the subscription, the data goes away. Now, my guess is-- I haven't tried this-- but my guess is if I stop my subscription today and I come back to them tomorrow and say oops, which anyone try

that AutoCAD command, oops? Great command. OK, if I say oops, my guess is that they'll come back and they'll say yeah, we can get your data back for you. My guess is it doesn't go away instantaneously.

But if you're under investigation for handling classified information, deleting your account-- it might take more than that to get rid of the data itself. Haven't tried it though. Haven't tried it. Haven't googled it, but my guess is it's going to stick around a little bit. And the reason is because Microsoft understands-- and the same would be with other cloud operators.

They understand that when you're putting data out that, you're counting on it being there. I've had clients come back to me and say, you know 10 years ago you did something for me, and I usually have a copy of it still somewhere. So my guess is that they're not going to just dump it because the credit card was declined or because you say shut it down. Great question.

AUDIENCE:

What is the understanding about the ownership of the data?

JERRY WINTERS:

Ownership of the data? The understanding is that you own the data. They're hosting the data. That's the understanding. It gets back to that pro con for control though. Right? I like control. I like to have my data. I know it's my data. I know where it is. On the cloud, it's your data still. You've seen the posts that come up on Facebook? I hereby do not give Facebook permission to-- thinking that that's going to somehow mean that Facebook no longer can use their photos on that. Well, they will never claim ownership of the data, but they're hosting the data.

AUDIENCE:

Do they have a responsibility to protect that data from other people?

JERRY WINTERS:

Do they have a responsibility? A liability, yes. A responsibility, it's in their best interest. Yeah, it's in their best interest. There's a license agreement, I'm sure, somewhere. When you sign up for this it talked about that. I'm sure I agreed to their licensing agreement at some point in time. What did it say? Accept. That's what it said. It basically said, if you want to use SQL Azure, you click the Accept button and

that's what I did.

So yeah, if you haven't signed up already and you site up, it might be interesting to read that. But obviously they understand that their company's reputation, their butt's on the line for the data that's there, which is why replication-- what does replication mean? It means it's in two places at the same time. That's what replication means. Right?

Over here they can say georeplication, and that might mean bare minimum, 2 at the same time. If I had to guess, they might say at least 2 at the same time. Or they probably have it in 10 places at the same time because the last thing they want or need is someone saying, I trusted you. I gave you my data, and what did you do with it? So great question.

All right. So create a database. Let's create a database. I'm going to get this one, right here, and I'm going to click on the Select button. And I'm going to click on Create. And you'll notice down in the lower left hand corner it said validating, validation successful. OK, now if I come up here and hit the Refresh button, we might see a database show up here that says AU 2017.

Come on database. Come on database. Refresh database. Refresh database. And this is intentional. This is to show you how long it takes for that database to be created. I wish I had a graph, or a chart, or something like that to show you something. This might be more instructive. What was that?

AUDIENCE:

[INAUDIBLE]

JERRY WINTERS:

It's actually just my account. So hopefully it won't come back and say declined. If it does, I still have my one that I already created that's up there. There we go and here it is. Now, that took-- and you can go back to the video to verify this, how long that took. That took a certain amount of time, right? It took about as long as it takes to take that CD and stick it in the drive, or download the ISO or whatever, and run the installation. So that's how long it took to run them. Didn't take a lot of

time.

So there's my monitoring there. This is going on. So now I have this database and it's called AU 2017. And what's my activity log look like here? Status exceeded just now. That's what just happened. OK, and go back to my overview, and let's see here. Monitoring, it's not being used. My DTU percentage is 0% right now. All right. A couple of things here that are important to understand and see, the server name is up here. You need to know the server name and we'll see why in just a minute. OK. You'll notice when I put my cursor over there, it says that I can actually click on this little thing here. Come on. I may not be able to do it. Always right-click. If you can get your mouse over there without it going away. Don't ask me why they do this. I'm almost there.

[YELLING]

Oh, I got it. I can click the copy to the clipboard. That's what that's for. OK? I'm going to click now and then forever hold my peace. All right, next thing we have here is the server version. So this will let you know what's happening with your server. And then we have show database connection strings. This is one of the more important things, right here OK, here are my connection strings. I'm using the adu.net stuff because I'm going to be developing here in vb.net I've got odbc, php, jdbc as well. So different connection strings are there that I can copy as well.

So I come over here and same thing. I can come over here and copy that, but I'm not going to do that because it was so hard to copy the previous one. I don't want to lose that. All right, so I can copy that to the database. Next thing I want to talk about is tools. Click on Tools, here, and here are my tools.

You'll notice there's something here called open in Visual Studio. If I click on that, lets minimize this, I've got a few things here. Open in Visual Studio. I've got configure your firewall to make sure your machine can access the database. Now, this is the firewall of the

machine that you're on. OK? This is not the IT department's firewall.
All right?

We've got download Visual Studio, if you don't already have it, and SQL Server Data Tools. I'm going to click on Download SQL Server Data Tools here for a second. Some people look at my presentation or say, well, why don't you have your handouts earlier? Why don't you things? And the reason is because last night I was still working on this. Why is that? Because things change. In the cloud things change rapidly. They change quickly.

You'll notice the date of the release on here is-- where is it? Dated October 26, 2016. Is that fairly recent? Fairly recent, right? Fairly recent. Autodesk wanted to have stuff up like three months ago. Now, the reason this is important is because, especially on the cloud side of things with SQL, they're changing things all the time, and they're actually making them better. It's good change, OK? Better change stuff.

All right? It used to be if you wanted to create a table in Azure, you do it through the portal. Now you don't even use a portal. Then you had the Visual Studio tools stuff. So go to the tool section, you'll want to download the tools here for Visual Studio 2015 because once you do that, what you can do is you can come over here and you can click on this open Visual Studio button, right there. Sure. And it opens up Visual Studio for me.

Now, it used to be this was one of the better ways to access it. Using the Microsoft SQL Management Studio is much better way now to do this. OK? And so I recommend you download that. Get the latest and greatest on that which also, by the way, is very recent. So it's coming here saying, OK, I already have your server name in here. I already know we're using SQL Server authentication. Here's your username. What's your password?

So I want to make sure we're not going to get this on the screen for

the recording here. All right, database name is AU 2017. I'm going to click connect and here it is. There's my database. Now, if you use Visual Studio, these should look fairly familiar to you for SQL Server. From this point on, we've done the hard work with the cloud. From this point on, pretty much everything is similar to, or equivalent to, or equal to SQL server.

So what's my next step here? Create some tables, maybe. I'm going to try this here instead of Visual Studio, and then we'll do it in the SQL Server thingy whopper here. So let's come to tables. Right-click Add New Table. Let's just see what happens here. We've got the design surface. Doesn't that sound cool, design surface? The design surface is loading.

Now, I've got my SQL statement down here. So if I'm really smart, smarter than I am, then I can use this and just put everything in here. If it gives me a decent interface up here, I may choose to use that. How are we doing on time here? About halfway through.

AUDIENCE:

What was the password that you entered before? Is that password that you set for-- I didn't see you set any password.

JERRY WINTERS:

Can you believe she just asked me what my password is? What was the password? Great question. I understand what you're asking. Great question. So when you create your Azure account, there's going to be a global administrator type username and password, and that username and password is used when you log in and that kind of thing. Si that is what that was. That was the admin password that I put in for that account.

OK, now I'm going straight right now to creating tables, which if I'm just developing an application, that's what I'm going to do. I'm not going to go straight to creating users, and permissions, and things like that. I'm going to go straight to the tables. I want to get my data in there. Right? I want to design this thing. So we'll get to adding users in a little bit here, but what we're doing right now is just creating a new table,

basically, in Azure. Good question, though. Thank you.

AUDIENCE:

[INAUDIBLE] access database, can you just import it?

JERRY WINTERS:

If you have an access database, can be important it? Probably. We're not going to show any third party tools up here, but there are third party tools that you can use and probably even through-- SQL server has its own tools for importing data and it uses this SQL Server Management system, here. So my guess is you could do that. I've not tried that myself. My guess is you could do that though. Great question.

All right, let's go ahead and create a data here. I'm going to create a-- I'm going to call this a user ID. It's an int. We're going to come over to the properties over here, and I apologize for the size of things here. Let's see here. We're going to go down to-- is this identity? Yeah, this is identity. So this is equivalent to an auto incrementing ID in access, for those of you who haven't used SQL Server before.

So the first time this comes it might have a value of one, and then the next time I create a record it might have a value of two. The next time I create a record, I'm going to have a value of three. OK, that's the way that works. All right, and they we'll have a-- what did I put in the handout here? Give me one second here.

All right. Email address. All right, that's going to be a-- then we've got a user name. We give that to 220

[NOTIFICATION BING]

Hey, backup your [? file. ?] Yeah, not going to happen. Phone number. OK, so this is one thing I've noticed. In the Visual Studio interface, here, is you've got to be careful because as you type, sometimes it's not as responsive. I'm glad this happened because I wanted to show that. If I type here, it's now showing up here, but you saw kind of what happened there where it didn't really catch on what I was doing very quickly.

OK, now inside SQL-- inside of Visual Studio, I'm sorry, it's a little bit different than inside of the SQL Management Studio. We'll see that in just a second here. So I've just created this, and when I'm in SQL Server, I create my stuff, I hit the close button. Do you want to save it? Yes, give the table a name and we're going to go. Inside of Visual Studio it's a little bit different. You'll notice there's an update button way up here at the top, right up here is an update button. If I click this update button right now, I will indeed have a table created inside of my new Azure SQL database. But what's its name going to be?

Its name is going to be table. How's that for a name for a table? That's what it is, right? OK, so that's probably not a good thing to call this thing. I'm going to call this Users. So in Visual Studio, this is how it happens. I give it a name. I come over here and click on my Update button. I hit my Update button and now it's going to go out and generate a [? script, ?] and now I'm going to click on the Update Database button.

And you'll notice down here, it's thinking about it. It's thinking about it. It's really working hard because you can tell because the thing is going around really fast, the little circle thing there. It's thinking about it. Still thinking about it. Elapsed time, it even counts down the elapsed time here. 44 seconds, 45 seconds, 46 seconds. Still thinking about it. Still thinking about-- this is exciting stuff. But it's more than nothing. Remember that. It's more than nothing. You saw something today. It's still thinking about it.

There we go. We've got the green check-mark. I just created, now, a table. Now, if I gave someone my super secret username and password, they could now go in and use that password to do things to the table. They could insert records, delete, query, that kind of thing. All right, so that's the Visual Studio way of doing it. There's another way to do it, which is kind of my preferred way, which is using this.

I'm just going to go back to my AU 2016 one I created previously, and let's do a-- I don't want a new database. I want a new table. Let's do

the same thing here. New table, and for those of you that have used SQL Server, this should look a little bit familiar here. So we'll go user ID, that's going to be an int. Come down here. Grab my identity specification. Put that at yes. Then we'll go here. What do I have, email address?

Now we have a username. If any of you have needed more than 120 characters for a username, I'm sorry for you. All right, there we go. Hit the close button. Yes, save it. The name is-- OK, I'll call this Users2. We'll say OK to that. There we go. Just for kicks, I am going to get reconnected up here. Let's try that again.

I got disconnected because I hibernated the computer, here. So give me a second to bring this back up here. There we go. All right. So just like SQL Server, but we're on the cloud here. OK. I want to edit those records. Actually, I want to delete those records.

[NOTIFICATION BING]

Delete four rows? Yes. OK. All right. And there's my Users2 table I just created there. All right, let's talk about creating users here. For SQL in the Azure side, you need two things, basically, in order to access a table. You need a login and you need a user, and they're not the same things. A login is an Azure thing, the user is more of a database thing. OK? So we're going to show you how to do both of those things here.

We come down to logins. Let's create a new log in here and you'll notice that my GUI goes away when we're dealing with logins. So what we need to do is create a user, a login called-- and just for kicks, I'll call this AU 2017. With password. Card This is where it gets a little tricky. They want a good password.

So 1, 1, 1, 1 is not going to cut it. 1, 2, 3, 4 is not going to cut it. So I'm going to put in a password here that's just kind of a play on things here. Autodesk universe city 2017. [INAUDIBLE] good, OK. Let me try something here. I'm just going to copy that and just to show you what

happens when you don't do that. How about that? Anyone use that for a password? Invalid syntax. It doesn't like that. We'll call it password1. We'll be really sneaky here. Thank you.

Password validation failed. The password does not meet Paul's requirements because it is not complex enough. So that's what happens when you get a password that isn't complex enough. All right, I'll come back up here. Let me get my apostrophes in place here. And we'll hit Execute. Command completed successfully. Now I can log into my database using that password and username, but accessing the tables is a little bit different. So I need to actually create a user now.

So now I'm going to create a user. I'll come up here to my-- so that was the security of my server. OK? Now I'm going to go to the security of my database. New user.

AUDIENCE: The properties of the user doesn't show up [INAUDIBLE]

JERRY WINTERS: I don't follow you, I'm sorry.

AUDIENCE: [INAUDIBLE]

JERRY WINTERS: Which one?

AUDIENCE: The database security.

JERRY WINTERS: Database security.

AUDIENCE: [INAUDIBLE]

JERRY WINTERS: Down here?

AUDIENCE: [INAUDIBLE] the properties of [INAUDIBLE]

JERRY WINTERS: Click on a user, OK. How about this one here? Yeah, it's all gone.

AUDIENCE: So it changes when you [INAUDIBLE]

JERRY WINTERS: Yes. This is a little bit different, yeah. There are some things that are a little bit different between SQL Server and Azure. That's one of them.

Yeah, it's not there. All right, so let's come up here. So create user. So the user we're going to create is called-- we'll call this AU 2017 User. For login, AU 2017. And then it's going to pull this off of here. And then we're going to give this person-- we're going to leave it DB owner for this. So we're going to give someone the DB Owner role for user, AU 2017 User.

AUDIENCE:

Does this have automatic pull down when you type your syntax into [INAUDIBLE] How do you know whether-- what are the other options besides owner?

JERRY WINTERS:

Great question. The question was basically, I'm starting out here in Azure here, and it's now really giving me information on what my options are as far as the roles. So there's a couple of things you can do. One, you can use Google, of course. Right? Another option is if we come over here, back to security. What do I want? Where did things go here?

AUDIENCE:

[INAUDIBLE] Visual Studio. Would that matter?

JERRY WINTERS:

Oh, I'm sorry. Thank you. I was like, whoa, things are suddenly different. All right. All right, so back to-- that's logins. Tables, not tables, security. New database role. Is this going to give me much? It's not going to give me much. All right, we're back to Google. So if we look up here, SQL Server database rolls, and just click here. We'll see the different roles here.

AUDIENCE:

So it doesn't have the [INAUDIBLE].

JERRY WINTERS:

Yeah, it's not really helping you go in and create these things. OK? There's two primary roles-- well three, we're going to talk about today. So one is the DB Data Writer, the other one is the DB Data Reader, and the other one is the DB Execute. And I'm not seeing it up there, but there's probably more down here. So yeah, it would be nice if they kind of put more stuff in there for you. And where's execute? Execute, one found. That does not help me. All right, well we're going to go through it today anyway.

So there's our role we're going to add there, just someone as a DB Owner. So hit the Execute button here, and command completed successfully. Always a good sign. So now we have a login and we user called AU 2017 User AU 2017. So that's what we would use now to access this database in our connection string. OK. Very good.

Let's talk about, now, how do we actually access this database here. We want to get to our-- here we go. So here's the code that's going to be used. Now, I'm going to be accessing this AU 2016 database, which is what was created prior to--

[APPLAUSE]

Thank you. Thank you very much. Prior to this class, I create the AU 2016 database. Now, you'll notice a couple of things here. First of all, the user ID is the username that you created here, and then the at and the server name. That's why I tried to copy that to the clipboard, if you remember that step. So it's a username that you create, at, and the server name.

We've got a password, here. There is a password that was created for that. I'm going to set multiple active result sets to true. It's not necessary in this specific instance, but that's OK. OK, here's our server, there. So all this came out of that connection string thing. So if I come back to my Azure, here-- that's not the one.

[CLAPPING]

Boy, they are excited about this class. Let me go back to my tools here. Where's tools? There go, no. Click on my database here, and my connection strings, here. If I click on that, I click on my copy. Then I can come over here to SQL and I can just paste it in here. So that gets you everything that you need, except for the username. It doesn't put that in here for you, and the password, it does not put that in for you. OK, so you'll need to put your own username and password in here.

Aside from that, I changed multiple active result sets from false to true. What that is is you create a single connection. OK. So here's my connection. You open the connection, then you have a command. And in this case, I want to see the different records in the database in a message box. So I'm going to use a data reader as well. The reason multiple active result sets is handy is because if I don't have that set to true, I can only have one command going at the time. Well, one reader, really.

So I want to have multiple readers, multiple commands going at the same time. I set that to true and I can do that, otherwise, I have to open another connection to the database. And the connection is where the time is taken up. It takes time to connect the database, and a lot of time, as we'll see in a second here. OK, so there's my basic code here. Open the database.

Create the instance of the database here. Open the database. There's my command. Select email address, username, phone number from users, and then I create a reader to read that data, and then I start cycling through those. For each record I show a message box. Then I close the reader, I close the command, I close the connection and dispose of the connection.

Now, I've never tried this. If you had your own local SQL Server and you opened connections to it a lot, and you don't close the connections in code, you could end up with this thing called a pooling error, pooling timeout. OK? So I've never tried this on the cloud. I've never tried to create that problem, but that could possibly happen here as well. So it's always important to make sure that in your code, that you're specifically closing and disposing of that connection.

All right, we'll hit F5 on the keyboard. And we're going to start our program here. And it's very exciting. As you can see, I clicked on button one, and what happens? Nothing. Why? I just finished deleting all the records. That's why nothing happened. So let's get something in there. Let's close this down, here.

Now, you'll notice that I'm just reading data here. So I'm using my-- that was my user ID. I'm using my read only user ID, and all this stuff will be in the handout, I just didn't want to go through all those steps. I'm using my read only. My data reader is the permission that was given to that, the DB Data Reader. OK? Read only. Why is that good? That's good because they can't mess up the data. Why is that bad? That's bad because the data reader can read anything.

So when I go into an office sometimes to do some consulting work, they'll say I've got this database. I'm not quite sure what's in it. That's OK. Do you have Microsoft Excel? I do. OK, do you have a username and password? I do. I can go on with the username and password and basically read their entire database. The entire structure as well as the data put into Excel, then we sit down, we look at it, and figure out what's happening. And a data reader can do that.

Now, if I want to write data, I need more than the data reader. I need a data what? Data writer, right? So I have my data writer. So I have RO for read only, RW for read write. I'm old. Sorry about that. I'm a grandpa. I can do it. OK, so here's my data writer code. It should look very similar. I open my connection, but instead of having a select statement, I have an insert into. So I'm going to give it an email address, username, and phone number. There are my values. I execute the non query.

Basically, a non query is I don't want information back from a database. I just want to execute this query. OK. I'm just going to add a little bit of code here just for kicks here because I can. I'm going to get the identity. Dim user ID as integer equals mycom, execute scaler. Execute scaler. What is scaler? I want one value back from the database. Just one. One value back.

And what am I going to get back? I'm going to get back the identity, which is the unique identifier of the record I just put in. Do you remember when I created that record or the table, I said I want the

user ID to be identity specification. What that does for me is it auto increments the number. That's nice, but it also allows me to use a select identity and it knows, even if 10,000 people are using the program at the same time, it knows who I am, and it knows my program just entered this one record, and it's going to give me back the unique ID of the one I just put in. OK.

So, we'll give that back, and then we'll show that in the message box just for fun. I added two string there for the people that do C# because they get nervous when I don't put two string on there. Here we go. Let's hit F5 on the keyboard. We'll compile a great program. We're going to hit button two here and-- seriously?

We're going to pretend that didn't happen. OK? Select identity-- do I have that in there right? OK. This is the command type. I don't want the command type. I want the command string. So just to close things out here, I'm going to do a right-click down here, set next statement. In other words, let's just kind of go over that, like I'm trying to smooth it over, OK. Hit F11, F11, F11, F11. OK we executed or exited gracefully there.

Let's try changing our, not our command type, we want our command text. I have a feeling it's going to work better. Button two, click it. Hey, look at that. So there's my ID of six. I say OK to that. Now, if I click on button one, what's going to happen? There's my email, my username, my phone number.

So previously, we were using the read only. Now we're using the read write user. Now, read write-- oh, I got two of them there. OK, that's all right. So my read write user can do things such as insert records, modify data in records. Is that a good thing? Yes, because that means it happens sometimes. Is that a bad thing? Yes, because if you can add a record, you can also delete a record. Right? And is deleting records a good thing? Sometimes.

OK. Once again, you don't have an oops command here. No undo. So

that's not a bad thing. So this is the DB Data Writer, is the permission that the read write username here has. DB Data Writer. The other bad thing about this is something called SQL injection. Has anyone heard that term before? OK, let's pretend for a moment that I have my desktop application I'm developing here, I also have a website application I'm using. And someone brings up my website and it says, enter your username. And I say, my username is droptableusers. Let's say I put that in parentheses even, OK?

So my code comes in. My code says, OK, we're going to execute. Insert into this, and this, and this, and this, and SQL Azure sees droptableusers. And what does SQL Server say? Do you have permission to do this? Yes, you do. OK. All right. And so what happens is someone is able to take-- I was asking for a username, and instead of a username, they put a SQL statement in there. And so they actually executed that. So that's the general security stuff. This is not specific to Azure. This is applicable to all database platforms and things like that.

So to prevent SQL injection-- how are we doing on time, by the way? We OK? OK, got 20 minutes. Good. To prevent that, what we do is we create this other user type that can actually create-- they can execute. So this code is going to use my AU 2016 Execute. This is the DB Execute, I think is the name of the permissions on that.

Now why do I want to do an execute? Because SQL Server in the cloud, SQL Azure, can use this thing called programmability and these stored procedures. So I'm going to bring this up here. A stored procedure is something that-- let me see if I can get the font a little bit bigger up there. I might be able to do that. What was that? Control what? Wheel? I don't have a wheel. I just have a pad here.

Let's see if this works as well. This might work. OK, control plus? All right, thank you. OK, I've got the whole thing on there. Good. So a stored procedure is a way to say, OK, I want to provide you some parameters, and then I want you to do something with those

parameters. In this case, I want to insert a new record based on the parameters I put in. I want to answer that into users and there's my statement.

I'm using these parameters called email address, username, and phone number. They're mapped up here to the same data types and sizes as my database can handle.

AUDIENCE:

How does that prevent SQL injection?

JERRY WINTERS:

Because stored procedures, it evaluates what's being put in and it won't execute. It allows the string in, but it sees it as a value instead of as an executable SQL statement.

AUDIENCE:

[INAUDIBLE]

JERRY WINTERS:

Right. Yeah. OK, so there's my stored procedure. So what I'm going to do now is instead of using, in my code here, the insert statement looked a lot like that up here. Right? Insert into users. I want email address, username, phone number. I'm doing the same thing, but now I'm doing it with a stored procedure. And we saw a moment ago how I said the command type. I wanted a command text.

Now, I want the command type to be a stored procedure, and all I'm saying is I've got a parameter called email address, and here's the type, here's the size, and there's the value I want to put into it. Just to make sure we're not doing any smoke and mirrors here, we'll change that to four, change that to four.

So now, this username I've created is called AU 2016 Execute, can only execute stored procedures. I could have 50 tables there and they have no way to basically get at those tables. All they could do is execute the stored procedures I've created. So you might want to create a user for non-employee people that you want to give access to. You can take these stored procedures and basically make a set of stored procedures just for them, if you want to go to that level.

So let me just run this quickly here. Just for kicks we'll step through the

code on this one. Here we go, F11. OK, open. OK, that didn't take that long, but if you have to open and close, and open and close a lot, it takes a long time. OK. My command is called add user. It's a stored procedure. My email address is this. My username is this. My phone number is this. Execute it, dispose, close the database, dispose the database, done.

Now when I come up here, I should be able to see the ones that were already there. I've got two, four, two. Oops, sorry. So we saw the four wouldn't show up there. So that's using stored procedures. Now, stored procedures can be used for lots of different things.

So we've got stored procedures, we also have functions you can use. I'm not going to go into a big explanation of the difference between stored procedures and functions. There are some things you can do in one that you can't do in another. There are different types of functions as well.

If you remember that scaler, when we executed the scaler to get back to the user ID, scalars are single things. You can create scalar functions or table valued functions. So this is starting to get into SQL Server types of things. Right? But you can do these things and this is all in the cloud. Now, how much time do we have left? Like 14 minutes I'm seeing there? Any questions about what you saw so far, or you've seen so far? Any other-- yes?

AUDIENCE:

Actually I have [INAUDIBLE] introduce conditioning at a very high level, or whether or not it has permission to read [INAUDIBLE] If you're interested in introducing a permission on a table or even an attribute bubble, and you're not working with an API, [INAUDIBLE] maintains those permissions. How do you incorporate that into a [INAUDIBLE] where you're executing SQL [INAUDIBLE] desktop application?

JERRY WINTERS:

So the question is, if we want someone to have access to one table but not another table, how could we do that? Great question. Let's see what we have here. That's my stored procedure, here. Table. There is

a way somewhere, I believe, to set permissions on a table by table basis. I'm not seeing it in the GUI though.

So you can give user permissions on a table by table basis. You can also give permissions on-- it might just have to be done in a SQL statement here, though. You can do the same thing with functions [INAUDIBLE] which stored procedures they can use and can't use. Another option there. But you're saying you want to do it at the database level, not in your code, right?

AUDIENCE:

Well I'm just [INAUDIBLE] in terms of [INAUDIBLE] if you're distributing applications to users and you realize you need to make a change in the logic behind [INAUDIBLE] do they need to download [INAUDIBLE] are you able to manage those permissions [INAUDIBLE]

JERRY WINTERS:

So the question is, I've got a change in my business logic, if you will, if we can use that kind of term. And if I want to do that, do I have to distribute a new application, or can I do that at the database level? So it can be done on the database level, which is one of the reasons why people like to use the stored procedures in that, which is because a lot of business logic can be used or implemented in the stored procedure.

If you look at a large type system, we'll call it, that uses SQL Server and you open up the stored procedures, you'll see them used a lot. OK? A lot a lot. So that gives you the benefit of being able to modify your business logic inside of, in this case, SQL Azure. But it also allows you to something else.

If I just do a SQL statement that says select ABC from table users, it will run, but SQL Server sees that text come in, It? Has to compile it into a format that it can understand, and then it executes it. So one of the benefits of stored procedures is they're actually already-- I'll use the term pre-compiled, for lack of a better term-- on the machine. So it actually is faster. It's not just better on the permission side of things, the security side of things, it actually executes faster than a standard SQL statement. Yes, sir?

AUDIENCE:

In this scenario [INAUDIBLE]

JERRY WINTERS:

So say that again, now.

AUDIENCE:

Your Azure is separate from your AD. So how does users--

JERRY WINTERS:

What we're saying here is when you're on the Azure, you're using a SQL username and password. You're not using Active Directory. Is that what you're saying? OK, so when you have a local system, you might use Active Directory to be able to determine who's coming to your database, and do some things based on that.

With Azure, you don't have that in Azure as an authentication method to connect to Azure. However, your application still sees who it is. So at the application level, your application can see Active Directory, and you can authenticate. With the application you could actually say, put in your password. Right? And they could put in their password, and you could authenticate against Active Directory to see who they are, make sure they are who they say they are, and then you can handle some things like that, perhaps, at the application level.

But you're correct, that you don't have that in Azure. So you're going to have to have some mechanism, obviously, and that would be in your connection string as to who's logging in. Right? So remember that the login just gives you access to the database. It doesn't give you access to the data. It just gives you access to the database.

So if you have an application you're having people login, then their username and password is going to be stored inside of Azure as well, and you can authenticate against that table, and that will determine what they can or can't do in the application. Does that make sense?

So your application comes up. You say, OK, what's your super application A username and password? They put that in. You use a stored procedure against SQL Server here, saying here's my username I got, here's the password I got, execute. What can they do?

And so that gets into kind of another discussion of how do you implement that, but you can have an implementation level. You could go to groups. You could have a table for groups. I've got an admins group. I've got a read only group. I've got a users group, a power users group. I've got a supervisor group. Right?

So your user table then says OK, user A is a member of these three groups. That kind of thing. So you could do that. But in terms of business logic changes, you can do it at the Azure level using stored procedures. It would be a good way to do that. Views, not so much, just because it's straight data on the views, but you have views as well in here. So if you've got three tables and you want to kind of pull them together in one view to query against that view, you can do that instead of getting into the inner join, outer join, and that stuff. Great question. Yes?

AUDIENCE:

You may have already answered this, but with creating usernames and passwords, they seem kind of hardcoded where if you have to change it, you're changing the code. You don't store those in a table?

JERRY WINTERS:

So the question was, we hardcode the username and the password. In this example, just to make it easier to see run, that's what we did. So let me switch back here to Visual Studio, here. So I've got my user ID right here, my password right here. That's not the best practice if you have a big application you're sending out. OK. There are multiple ways to handle this. Some way, somehow, your application has to have a way to authenticate against your Azure login. Right? And so, that information has to be somewhere.

Now, you've got options here. Right? You could create an XML file, or way back when we called it any files, that you could actually read from when you start up. And if you decide you need to change something, then you change it on your Azure side of things, and then when they try to log in, it doesn't work because you've changed the credentials on Azure. Right? And they say I can't log in. It says my passwords are

bad or whatever. Then you get them a new connection string or a new username and password.

You can take this and hash it. Using some kind of an algorithm, you could hash it, encrypt it, and put it out as a file. It can be called `UsernamesAndPasswords.txt`, but it could be a hash binary type file to confuse people. So a lot of different things you can do there, but some way, somehow, it has to be able to get that username and password in there. Compiling it is nice because the user can't go in and oh, look at that, they've got something there. I'm going to write my own application now using those same credentials.

So you compile it into a DLL. Some people won't push as hard to get it. You can use something like `Dotfuscator` to take your DLL and make it even crazier so it's more difficult to decompile, to get at that. If you're really concerned about security though, you're going to be dealing with an encryption of some sort somewhere in the local system, or on the network for that matter.

Networks are nice because if you have it on the network they have to be from the network, now you're back to Active Directory options, as far as who can read and write from that particular thing. So someone can be an authenticated user, they can be logged into your network, but if they don't have permissions to read and write from that or read from that directory, they try to use their application and they can't even get to that username and password through Active Directory. All right, other questions? Anyone?

Well, this is best session so far, Autodesk University. OK. Good. All right, well, as I mentioned, I'll have a handout for everyone to download. It'll be up there in a few hours. I'll incorporate some of the things that we talked about today, including what real time, in time, whatever that thing was there on the Azure database portal there. Hopefully, this was good information for you. Hopefully it was helpful for you. I'll be doing another class about storage.

So this is all about SQL. It's about querying data. My class on Thursday is about creating your own Dropbox, if you will. So if you have an application and people need to be able to use templates, you could give them an FTP site, you could give them a Dropbox share, you could [INAUDIBLE] Google Docs. You could do those things or you could have your own little SQL Azure-- not Azure SQL-- your own Azure storage area and that they'd have access to through your application. It's really pretty cool stuff. All right. So, thank you so much for coming and I hope you have a great AU this year.

[APPLAUSE]