SD22317

# Moving to the Cloud for Your Database-Dependent AutoCAD Add-ins

Jerry Winters
VB CAD, Inc.

## Learning Objectives

- Learn the pros and cons of cloud-based database storage
- Learn how to create a new Microsoft Azure Account from scratch
- Learn how to create a new SQL database on the Microsoft Azure Cloud
- Learn how to read from and write to a cloud database from an AutoCAD .NET add-in

## Description

Many AutoCAD software add-ins depend on a database to store relevant information. For years, databases such as Microsoft Access were used to give multiple users access to the same data. At some point, it may have become necessary to move on to a more powerful platform such as SQL Server or Oracle. Now moving to a cloud-based database such as Microsoft Azure SQL database is easy enough to demonstrate the entire process from beginning to end in a 90-minute class. Developers no longer need to wait for their IT department to create a new virtual server and install software before they can begin work on a new project. For literally pennies a day, a developer can create a new cloud-based database that is visible to anyone running their software without worrying about VPNs, firewalls, and an approval process that takes months to complete. This session features AutoCAD, AutoCAD MEP, and AutoCAD Map 3D.
AIA Approved.

## Your AU Expert

Jerry Winters has been writing add-ins for AutoCAD software since 1992, and he lends his experience to developing powerful, useful add-ins to improve accuracy, increase efficiency, and eliminate user error. Although technology continues to advance at a rapid pace, Winters strives to be aware of and proficient with emerging technologies.

## The Pros and cons of cloud-based database storage

### The Pros

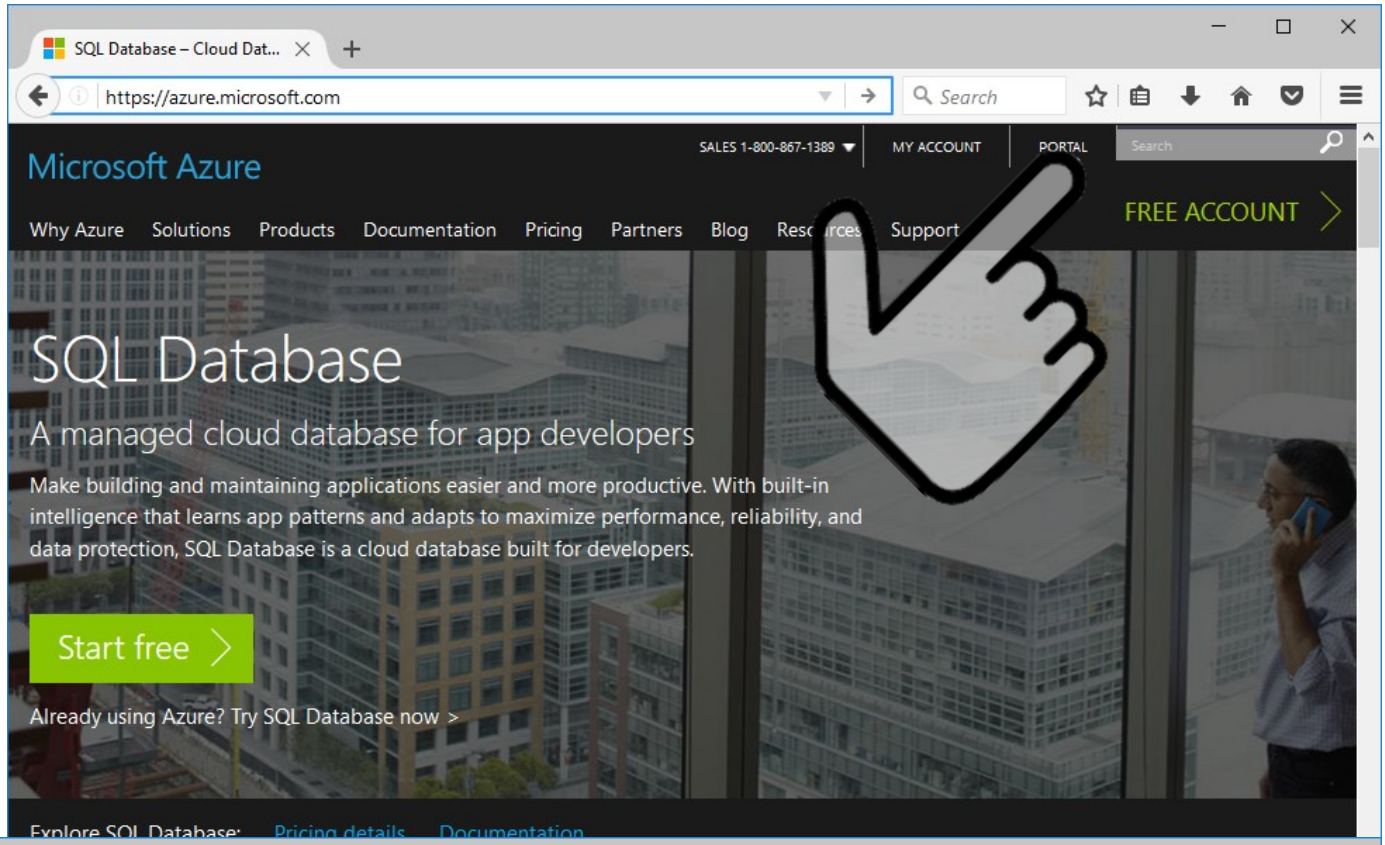
- Inexpensive Startup Costs
- Globally accessible
- SQL Server (Express to Enterprise) compatible SQL and .NET Code
- Bragging Rights
- Can be implemented without IT Department red tape
- Multiple geolocated options

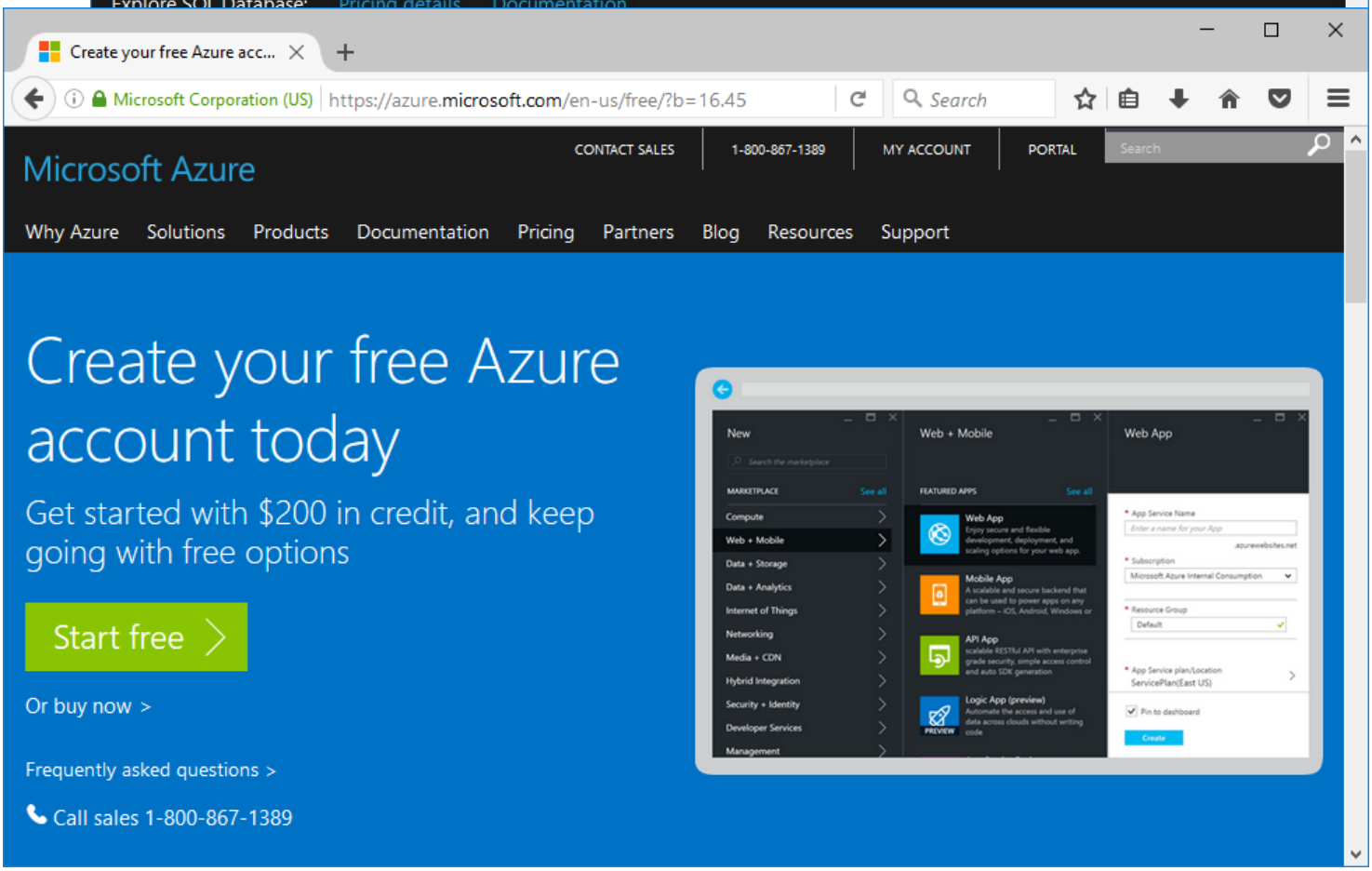
### The Cons

- Surrendering ‘control’
- Performance  - What is it going  to be?

## Creating a SQL Azure Account

Go to *https://azure.microsoft.com* to create a new Azure account. During the account creation process, you will enter a SQL Azure Username and Password. These credentials are the master credentials for access to everything in Azure.



## Creating a new SQL database on the Microsoft Azure Cloud *and*

## How to read from and write to a cloud database from an AutoCAD .NET add-in

The following screen shots demonstrate the steps involved in creating a new database. Part of this process includes the creation of Logins, Usernames, and Permissions.

The Azure Portal and integration with SQL Server Management Studio has been changing at a rapid pace. The screen captures in the following pages are provided as working as of the time of this class.

When a new SQL Azure database is created, one of the primary differences between the different accounts is the number of "DTUs" included in the package. Here is a direct quote from Microsoft's website:

*"A DTU is a blended measure of CPU, memory, and data I/O and transaction log I/O in a ratio determined by an OLTP benchmark workload designed to be typical of real-world OLTP workloads. Doubling the DTUs by increasing the performance level of a database equates to doubling the set of resource available to that database. For example, a Premium P11 database with 1750 DTUs provides 350x more DTU compute power than a Basic database with 5 DTUs."*

*https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers*

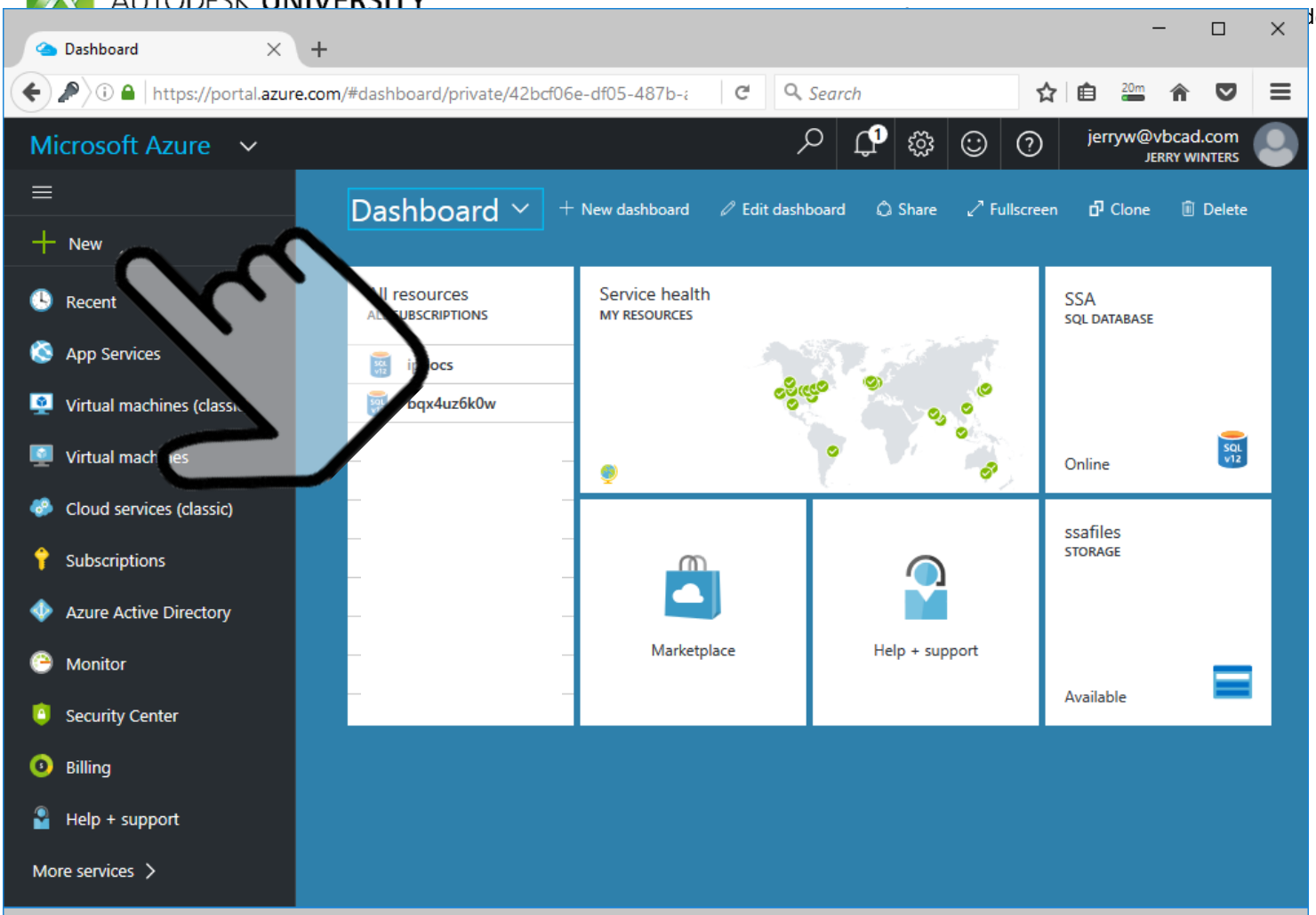
The following graphic shows what each level of DTU provides. Of interest is the number of concurrent logins and sessions.

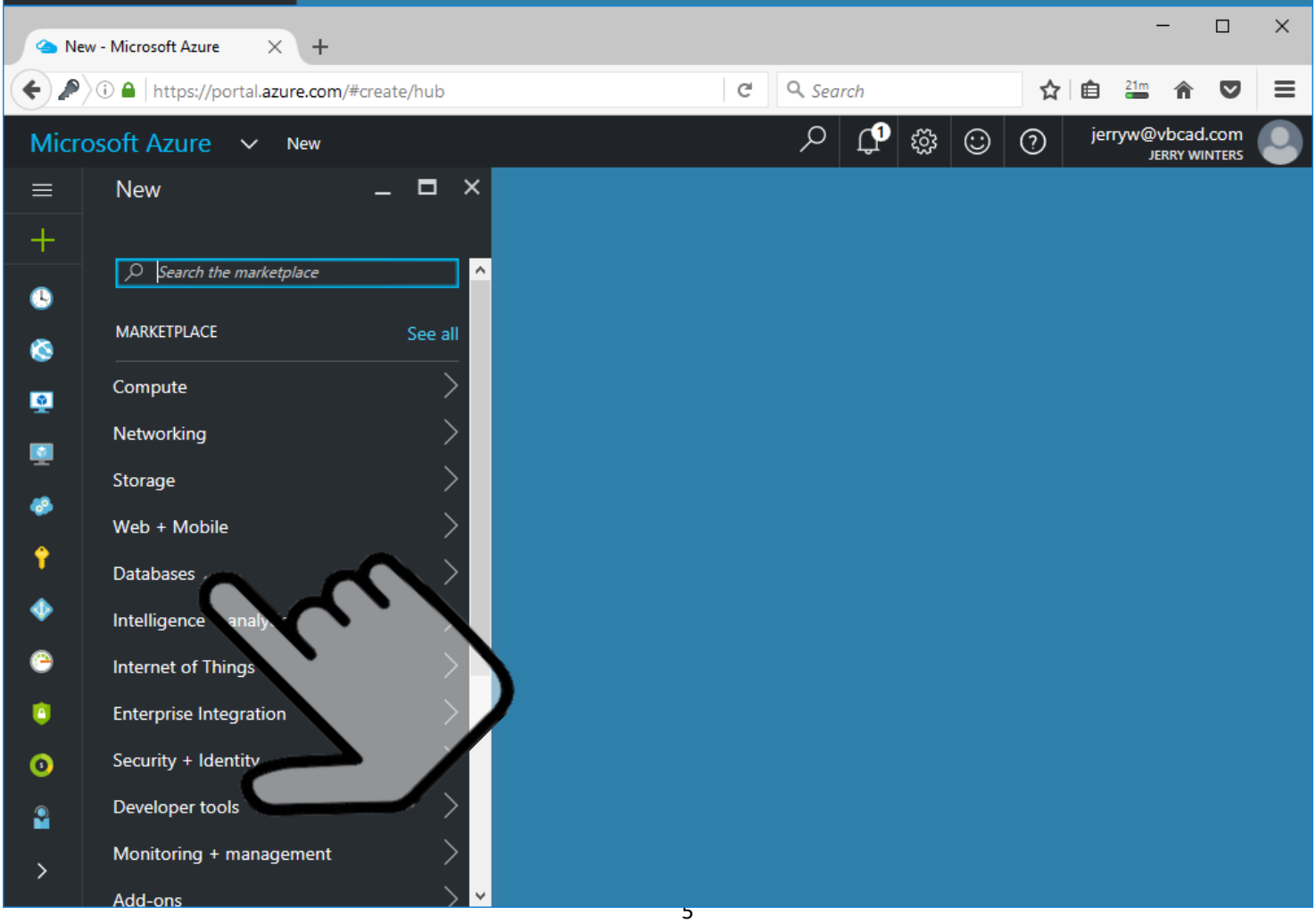
| | Basic | Standard | | | | Premium | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | S0 | S1 | S2 | S3 | P1 | P2 | P4 | P6/P3 | P11 |
| **Maximum database size** | 2 GB | 250 GB | | | | 500 GB | | | | 1 TB |
| **DTUs** | 5 | 10 | 20 | 50 | 100 | 125 | 250 | 500 | 1,000 | 1,750 |
| **Point-in-time restore** | Any point last 7 days | Any point last 14 days | | | | Any point last 35 days | | | | |
| **Disaster recovery** | Geo-Restore, restore to any Azure region | Standard Geo-Replication, offline secondary | | | | Active Geo-Replication, up to 4 online (readable) secondary backups | | | | |
| **Max In-Memory OLTP storage** | NA | NA | NA | NA | NA | 1 GB | 2 GB | 3 GB* | 8 GB | 10 GB* |
| **Max concurrent requests** | 30 | 60 | 90 | 120 | 200 | 200 | 400 | 800 | 1,600 | 2,400 |
| **Max concurrent logins** | 30 | 60 | 90 | 120 | 200 | 200 | 400 | 800 | 1,600 | 2,400 |
| **Max sessions** | 300 | 600 | 900 | 1,200 | 2,400 | 2,400 | 4,800 | 9,600 | 19,200 | 32,000 |

** In-Memory OLTP storage limits will soon adjust to 4 for P4 and 14 for P11.*

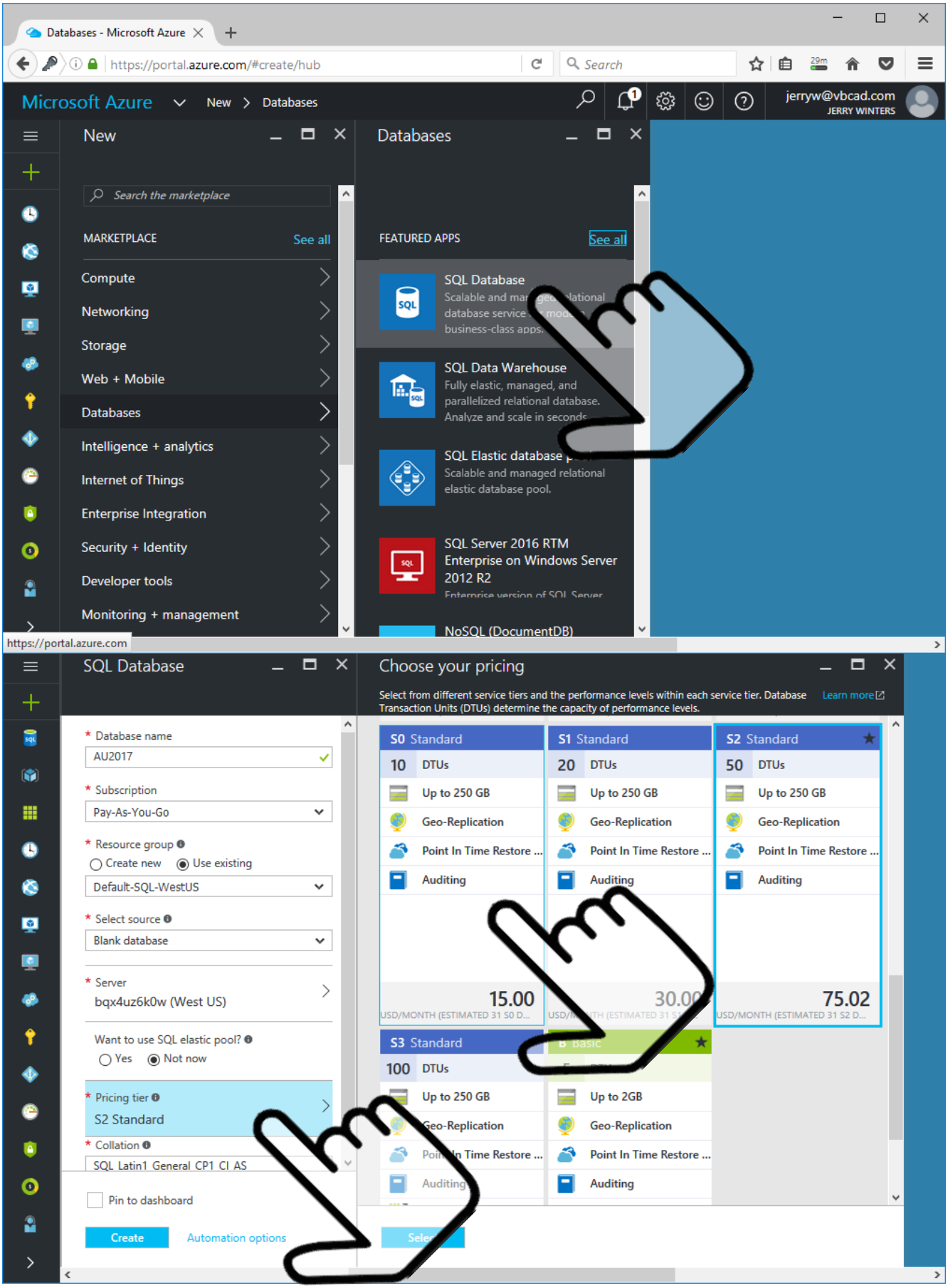
One question I did not have a ready answer for during the class was regarding "Point In Time Restore", what it was and how it works. Here's the answer to those questions:

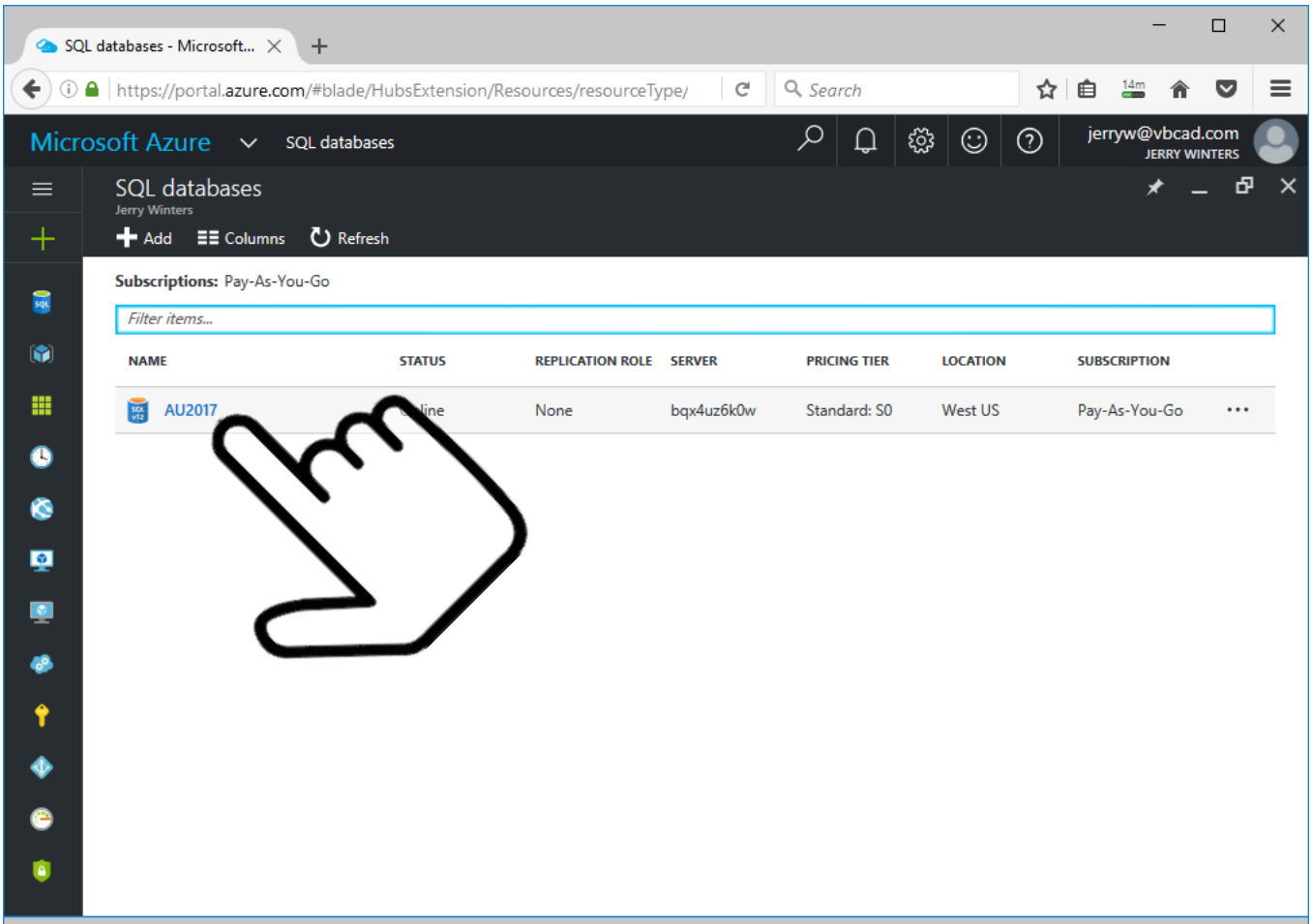
*The Azure SQL Database service protects all databases with an automated backup system. These backups are retained for 7 days for Basic, 14 days for Standard and 35 days for Premium. Point-in-time restore is a self-service capability, allowing customers to restore a Basic, Standard or Premium database from these backups to any point within the retention period. Point-in-time restore always creates a new database. The database backups are taken automatically with no need to opt-in and no additional charges. You only incur additional cost if you use the restore capability. The new database created by restore is charged at normal database rates. Together, the automated backup system and point-in-time restore provide a zero-cost, zero-admin way to protect databases from accidental corruption or deletion, whatever the cause.*

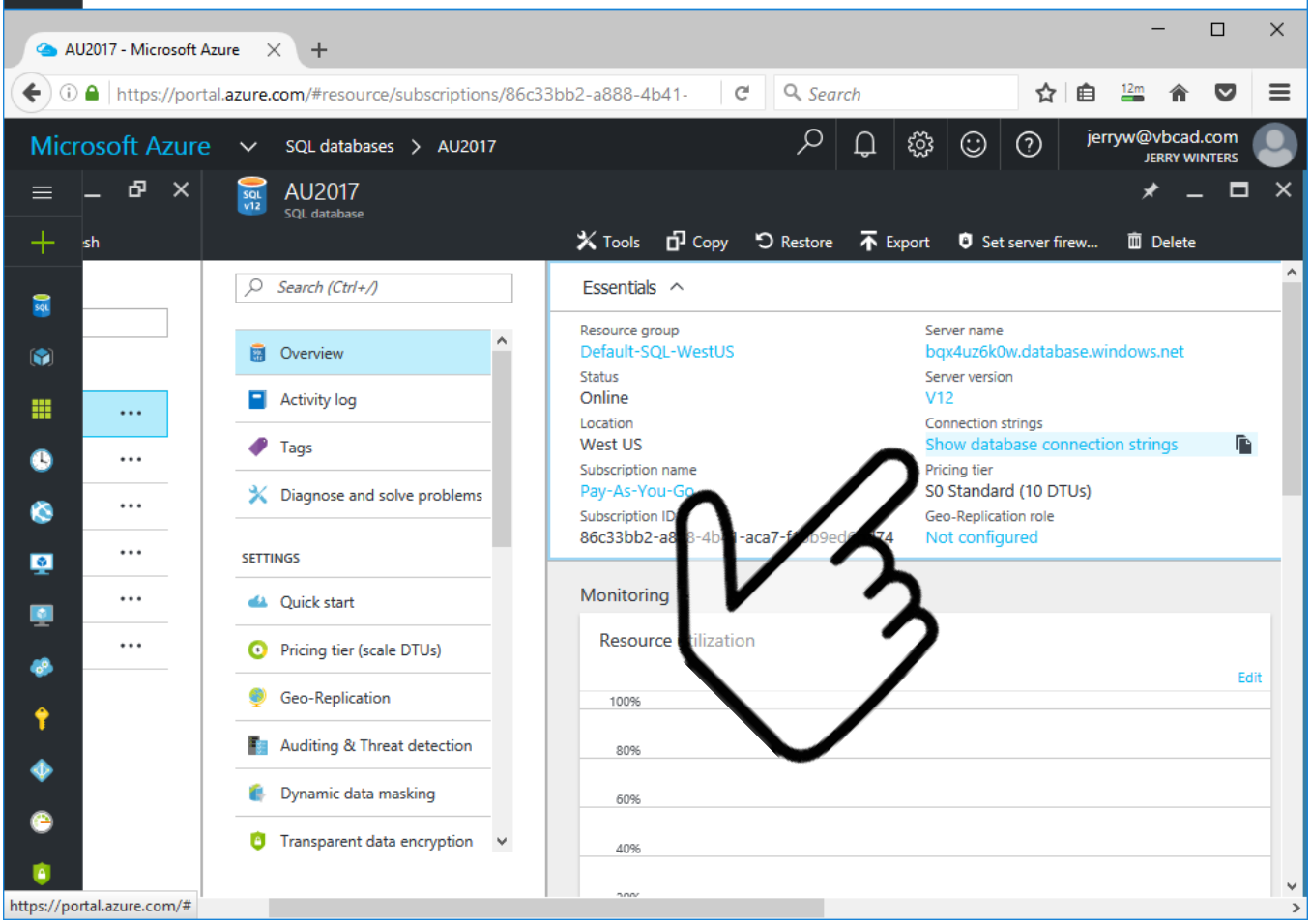

Dashboard
https://portal.azure.com/#dashboard/private/42bcf06e-df05-487b-
Search
Microsoft Azure
jerryw@vbcad.com
JERRY WINTERS
Dashboard
New dashboard
Edit dashboard
Share
Fullscreen
Clone
Delete
New
Recent
App Services
Virtual machines (classi
Virtual mach es
Cloud services (classic)
Subscriptions
Azure Active Directory
Monitor
Security Center
Billing
Help + support
More services
ll resources
UBSCRIPTIONS
ocs
bqx4uz6k0w
Service health
MY RESOURCES
SSA
SQL DATABASE
Online
Marketplace
Help + support
ssafiles
STORAGE
Available

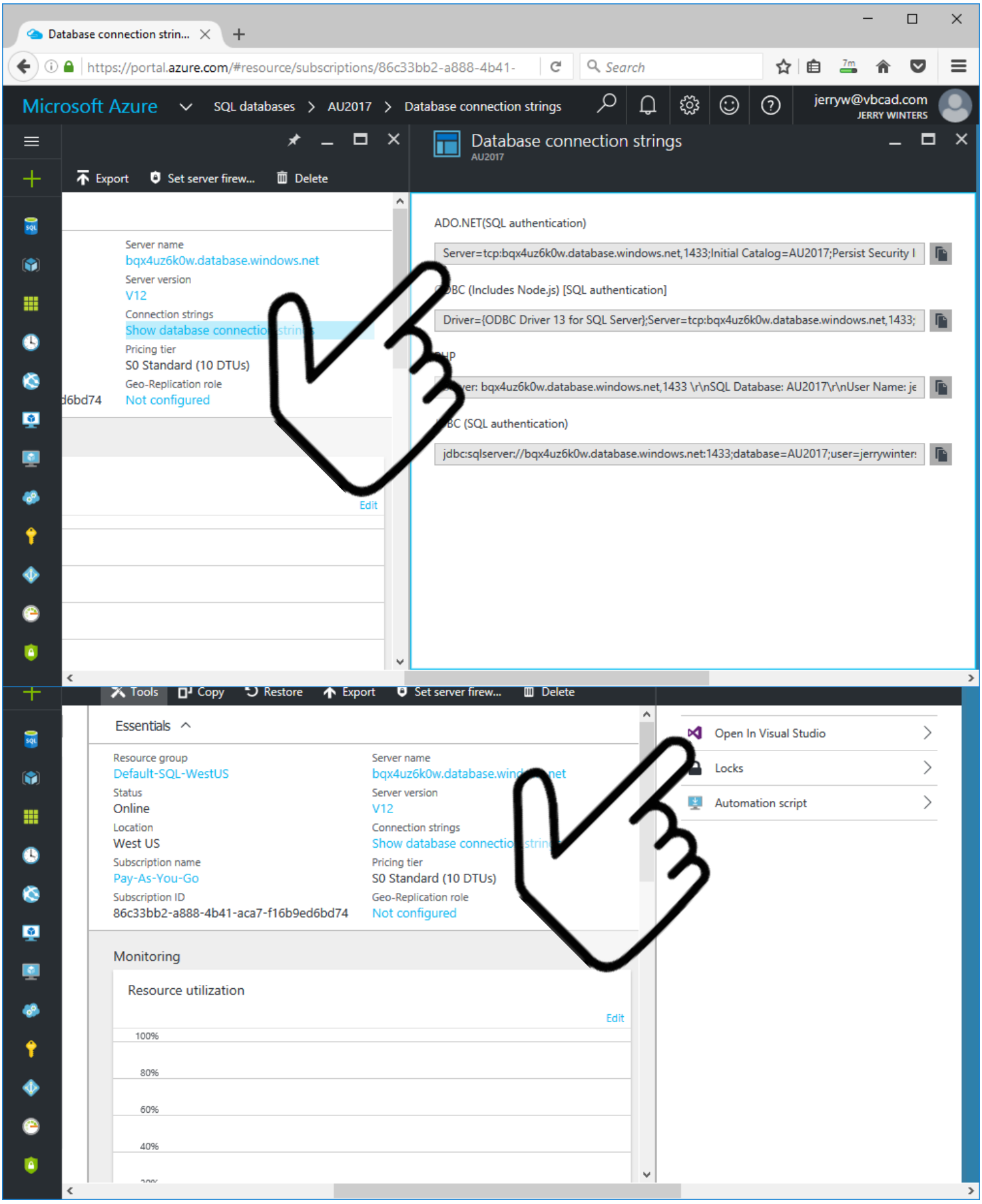

New - Microsoft Azure
https://portal.azure.com/#create/hub
Search
Microsoft Azure
New
jerryw@vbcad.com
JERRY WINTERS
New
Search the marketplace
MARKETPLACE
See all
Compute
Networking
Storage
Web + Mobile
Databases
Intelligence
Internet of Things
Enterprise Integration
Security + Identity
Developer tools
Monitoring + management
Add-ons

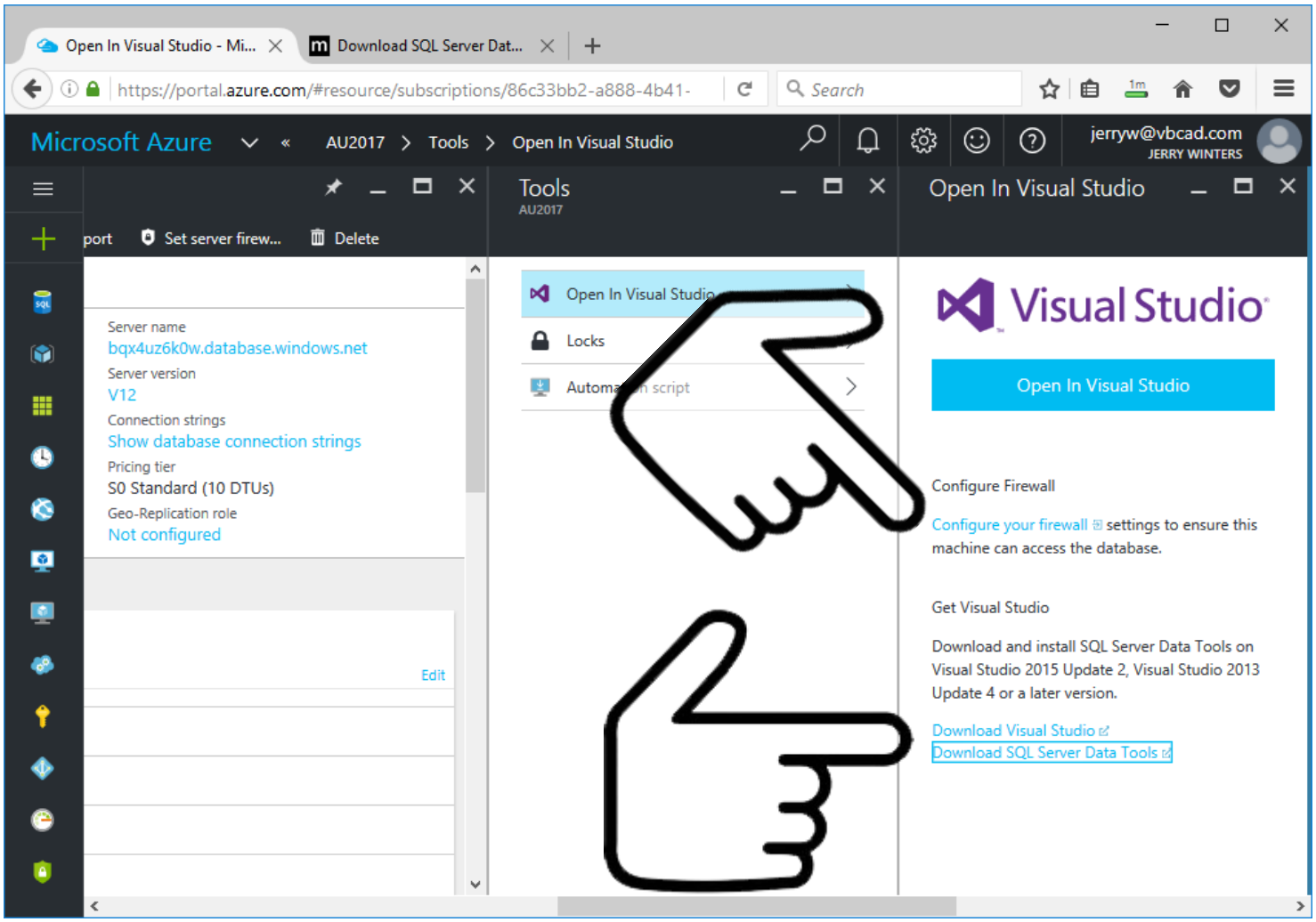
Please note that some of the screen captures shown here indicate that the Database created was named “AU2017” but the name we will be using is “AU2016”.

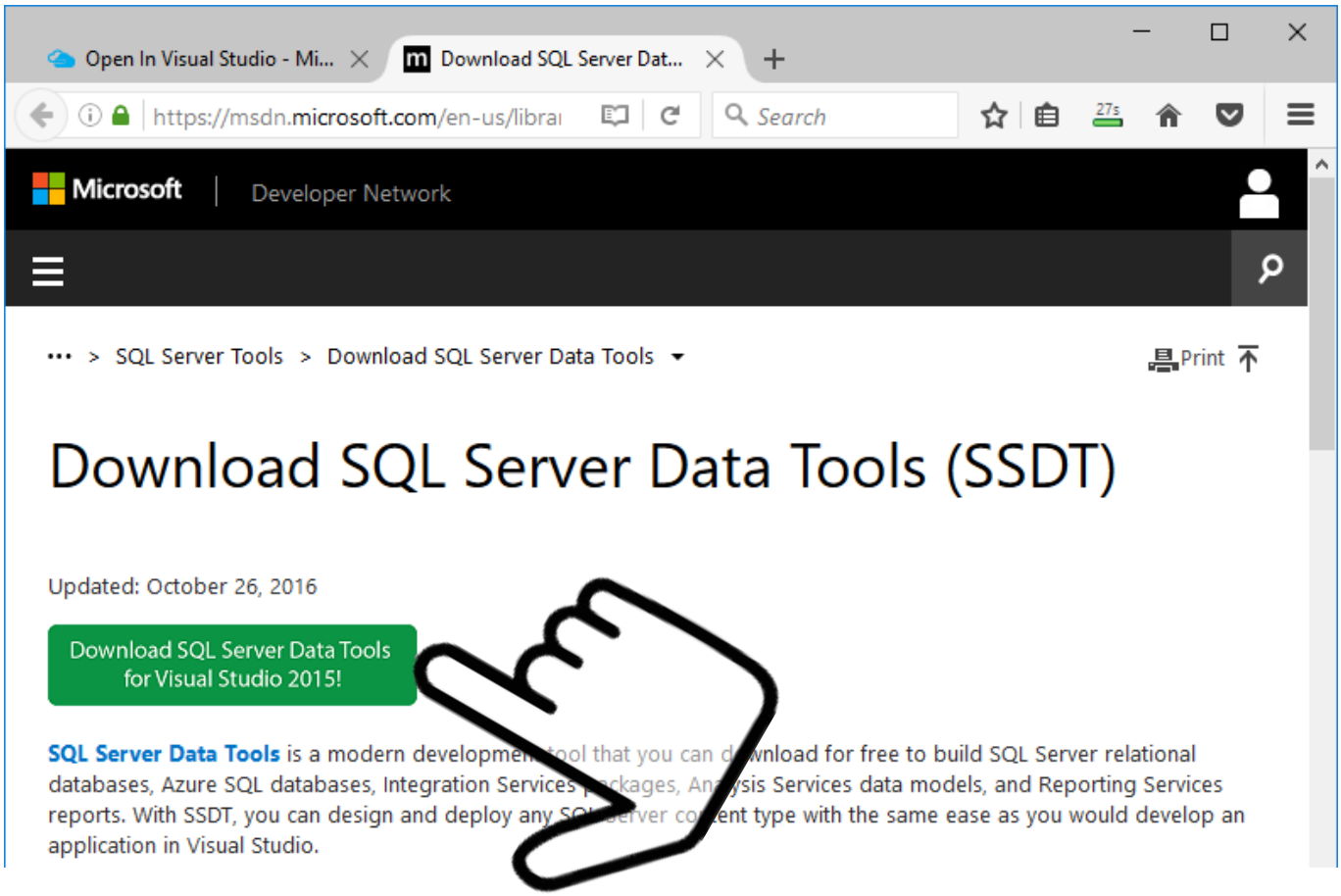



SQL databases - Microsoft...
https://portal.azure.com/#blade/HubsExtension/Resources/resourceType/
Search
Microsoft Azure
SQL databases
jerryw@vbcad.com
JERRY WINTERS
SQL databases
Jerry Winters
Add
Columns
Refresh
Subscriptions: Pay-As-You-Go
Filter items...
NAME
STATUS
REPLICATION ROLE
SERVER
PRICING TIER
LOCATION
SUBSCRIPTION
AU2017
None
bqx4uz6k0w
Standard: S0
West US
Pay-As-You-Go


AU2017 - Microsoft Azure
https://portal.azure.com/#resource/subscriptions/86c33bb2-a888-4b41-
Search
Microsoft Azure
SQL databases > AU2017
jerryw@vbcad.com
JERRY WINTERS
AU2017
SQL database
Tools
Copy
Restore
Export
Set server firew...
Delete
Search (Ctrl+/)
Overview
Activity log
Tags
Diagnose and solve problems
SETTINGS
Quick start
Pricing tier (scale DTUs)
Geo-Replication
Auditing & Threat detection
Dynamic data masking
Transparent data encryption
Essentials
Resource group
Default-SQL-WestUS
Status
Online
Location
West US
Subscription name
Pay-As-You-Go
Server name
bqx4uz6k0w.database.windows.net
Server version
V12
Connection strings
Show database connection strings
Pricing tier
S0 Standard (10 DTUs)
Geo-Replication role
Not configured
Monitoring
Edit
100%
80%
60%
40%
https://portal.azure.com/#

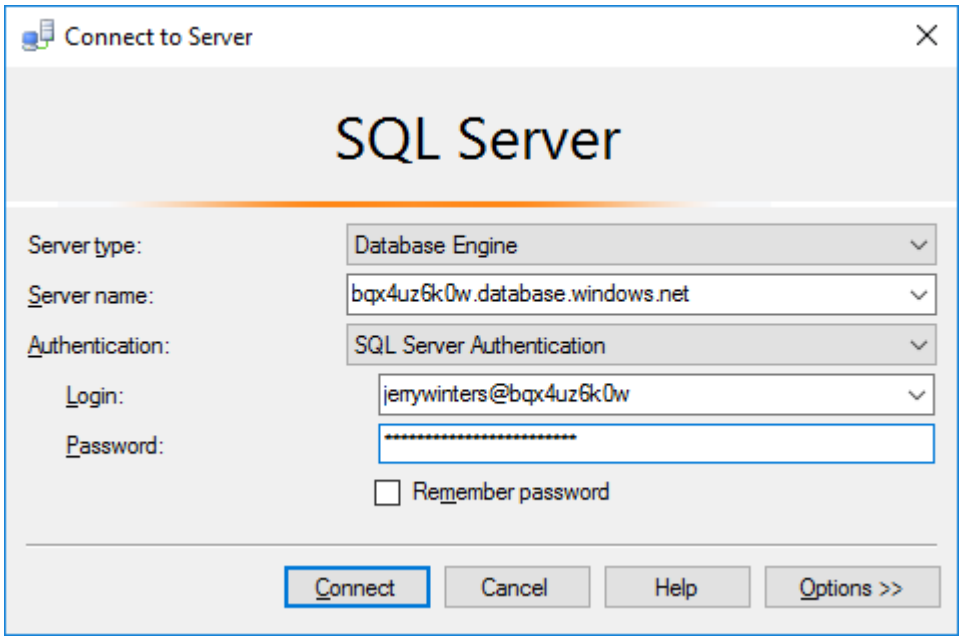

Database connection strin...
https://portal.azure.com/#resource/subscriptions/86c33bb2-a888-4b41-
Search
Microsoft Azure
SQL databases > AU2017 > Database connection strings
jerryw@vbcad.com
JERRY WINTERS
Export
Set server firew...
Delete
Server name
bqx4uz6k0w.database.windows.net
Server version
V12
Connection strings
Show database connectio
Pricing tier
S0 Standard (10 DTUs)
Geo-Replication role
Not configured
d6bd74
Edit
Database connection strings
AU2017
ADO.NET(SQL authentication)
Server=tcp:bqx4uz6k0w.database.windows.net,1433;Initial Catalog=AU2017;Persist Security I
BC (Includes Node.js) [SQL authentication]
Driver={ODBC Driver 13 for SQL Server};Server=tcp:bqx4uz6k0w.database.windows.net,1433;
HP
er: bqx4uz6k0w.database.windows.net,1433 \r\nSQL Database: AU2017\r\nUser Name: je
BC (SQL authentication)
jdbc:sqlserver://bqx4uz6k0w.database.windows.net:1433;database=AU2017;user=jerrywinter
Tools
Copy
Restore
Export
Set server firew...
Delete
Essentials
Resource group
Default-SQL-WestUS
Status
Online
Location
West US
Subscription name
Pay-As-You-Go
Subscription ID
86c33bb2-a888-4b41-aca7-f16b9ed6bd74
Server name
bqx4uz6k0w.database.wind net
Server version
V12
Connection strings
Show database connectio strin
Pricing tier
S0 Standard (10 DTUs)
Geo-Replication role
Not configured
Open In Visual Studio
Locks
Automation script
Monitoring
Resource utilization
Edit
100%
80%
60%
40%

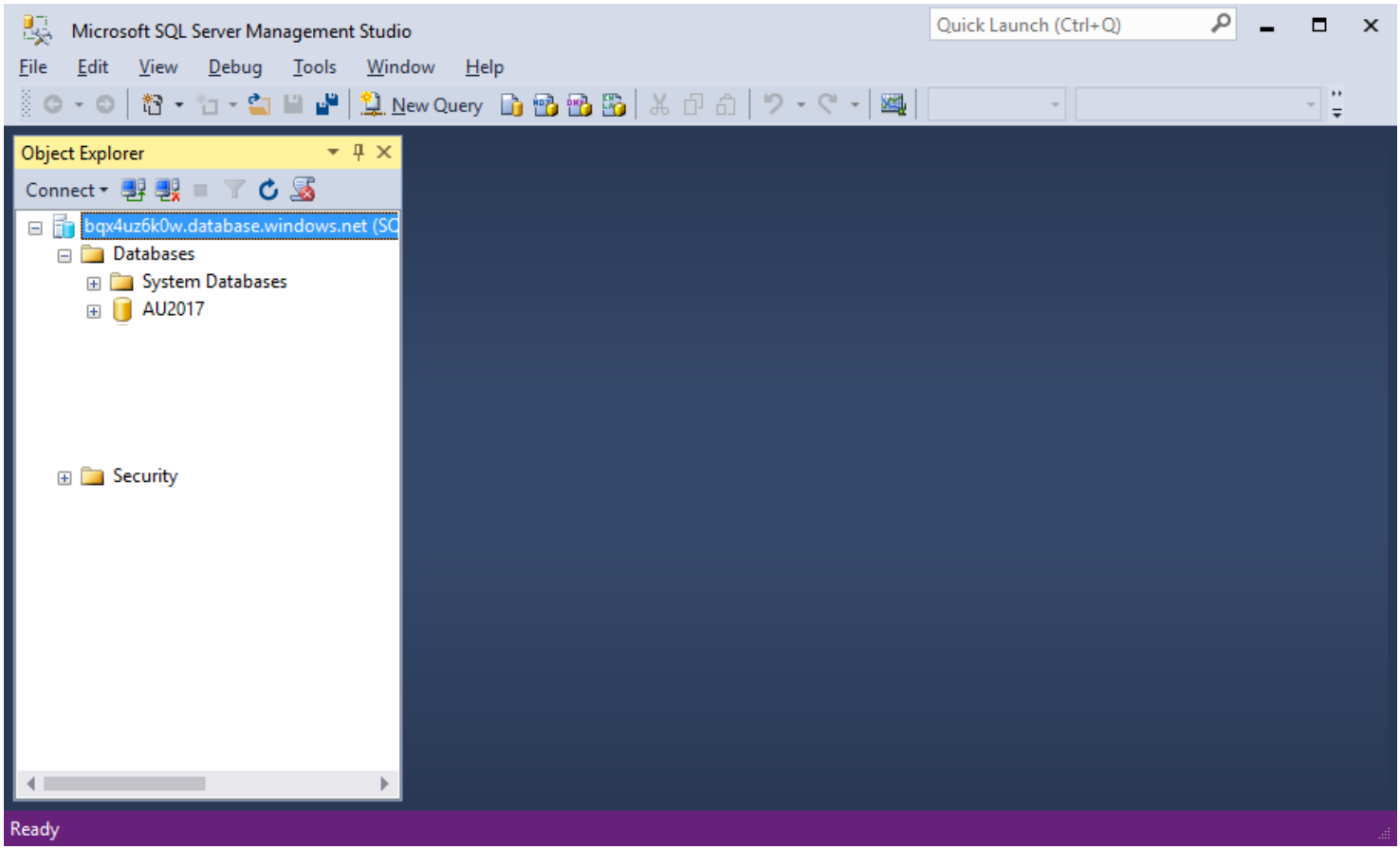

Open In Visual Studio - Mi...
Download SQL Server Dat...
https://portal.azure.com/#resource/subscriptions/86c33bb2-a888-4b41-
Search
Microsoft Azure
AU2017 > Tools > Open In Visual Studio
jerryw@vbcad.com
JERRY WINTERS
Tools
AU2017
Open In Visual Studio
port
Set server firew...
Delete
Server name
bqx4uz6k0w.database.windows.net
Server version
V12
Connection strings
Show database connection strings
Pricing tier
S0 Standard (10 DTUs)
Geo-Replication role
Not configured
Edit
Open In Visual Studio
Locks
Visual Studio
Open In Visual Studio
Configure Firewall
Configure your firewall settings to ensure this machine can access the database.
Get Visual Studio
Download and install SQL Server Data Tools on Visual Studio 2015 Update 2, Visual Studio 2013 Update 4 or a later version.
Download Visual Studio
Download SQL Server Data Tools


Open In Visual Studio - Mi...
Download SQL Server Dat...
https://msdn.microsoft.com/en-us/libra
Search
Microsoft
Developer Network
SQL Server Tools > Download SQL Server Data Tools
Print
Download SQL Server Data Tools (SSDT)
Updated: October 26, 2016
Download SQL Server Data Tools for Visual Studio 2015!
SQL Server Data Tools is a modern development tool that you can download for free to build SQL Server relational databases, Azure SQL databases, Integration Services packages, Analysis Services data models, and Reporting Services reports. With SSDT, you can design and deploy any SQL Server content type with the same ease as you would develop an application in Visual Studio.

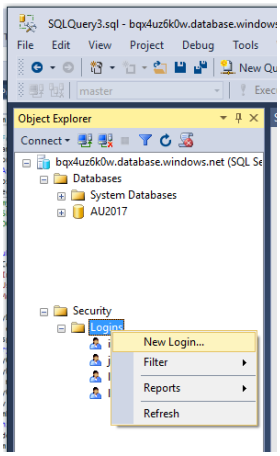
We will use the SQL Server Management Studio program to create Logins, add Users, and create Tables in our SQL Azure Database. Notice the “Server name” entry. This is taken from the “Essentials” screen of the SQL Azure Portal.

*During the AU event, we created a new Table in Visual Studio as well as using SSMS. To see that process, please review the video of this class on the AU website.*



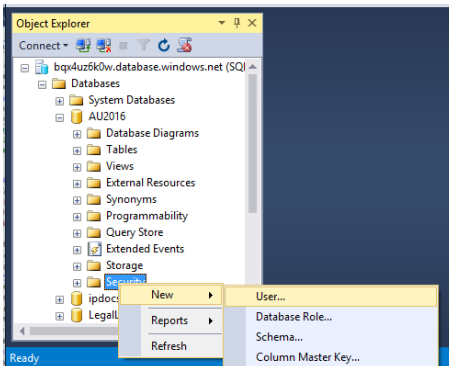



We are going to create 2 Logins now. Once these logins are created, Users can be created based on these Logins.

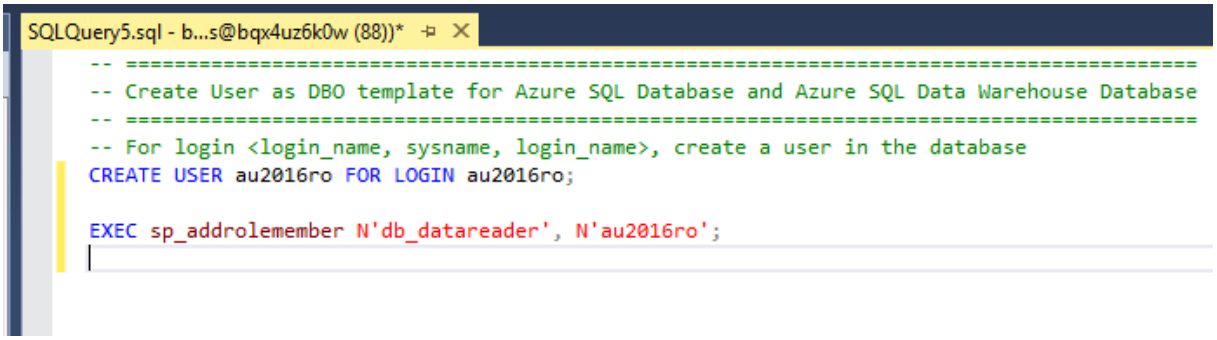



```
-- ================================================================================
-- Create SQL Login template for Azure SQL Database and Azure SQL Data Warehouse Database
-- ================================================================================

CREATE LOGIN au2016ro
    WITH PASSWORD = 'AutoD3$kUniver$ity2016ro'
GO
```

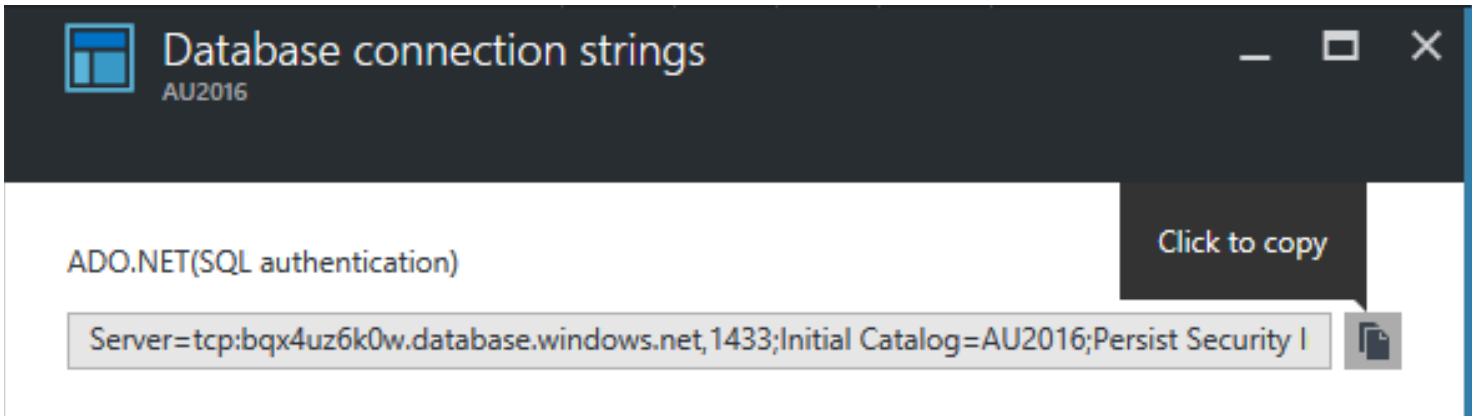
```
-- ================================================================================
-- Create SQL Login template for Azure SQL Database and Azure SQL Data Warehouse Database
-- ================================================================================

CREATE LOGIN au2016rw
    WITH PASSWORD = 'AutoD3$kUniver$ity2016rw'
GO
```




We are now going to create 1 User based on one of the Logins we just created.



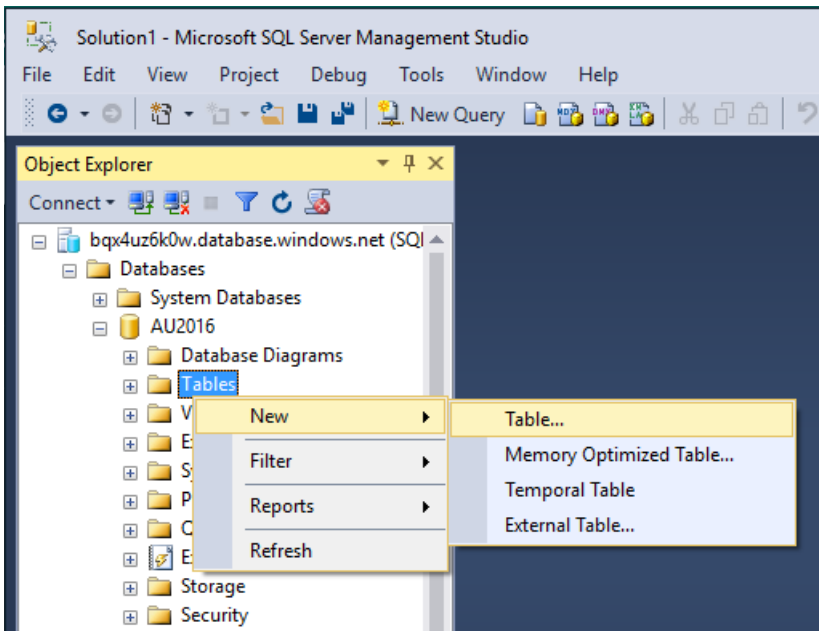
We are going to test our connection credentials by writing a little code. Before we do this we need to get the Connection String from the Azure Portal. When we paste this connection string we need to add the Username and Password.

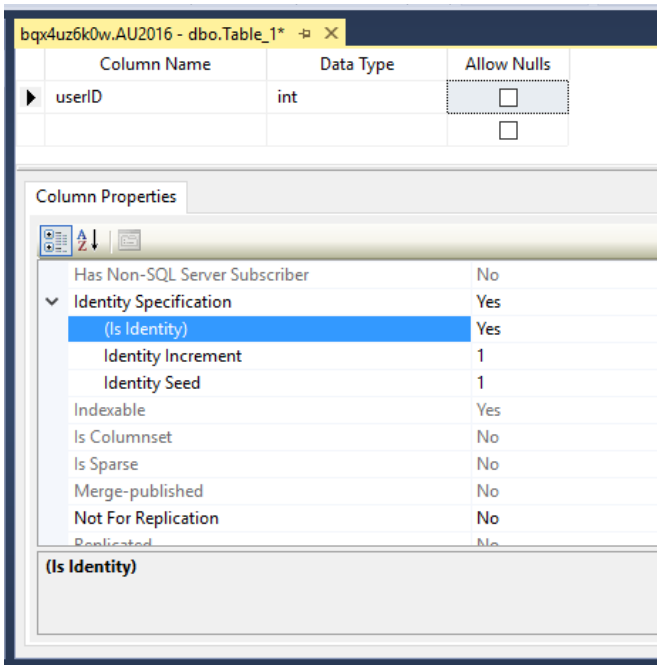


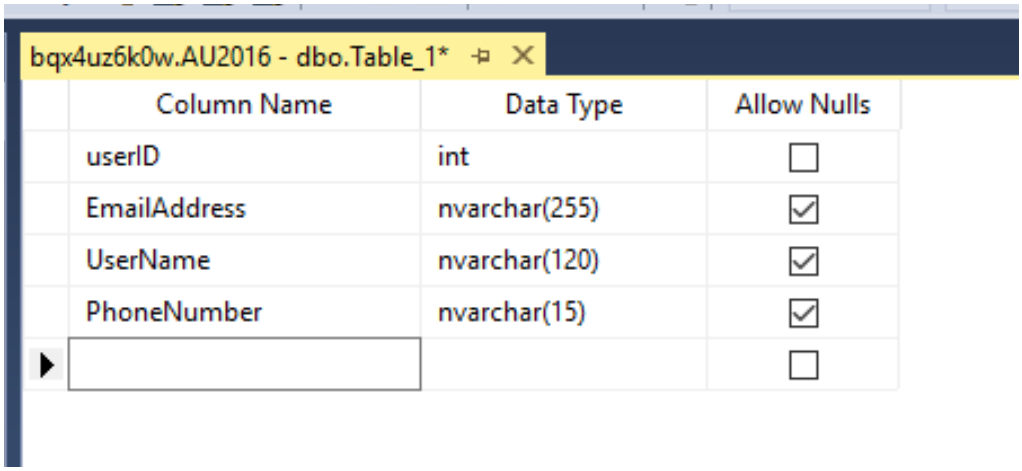

```
0 references
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim myDB As New SqlClient.SqlConnection("Server=tcp:bqx4uz6k0w.database.windows.net,1433;" &
           "Initial Catalog=AU2016;Persist Security Info=False;" &
           "User ID=au2016ro@bqx4uz6k0w.database.windows.net;Password=AutoD3$kUniver$ity2016ro;" &
           "MultipleActiveResultSets=True;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;")
    myDB.Open()
    myDB.Close()
    myDB.Dispose()
End Sub
```

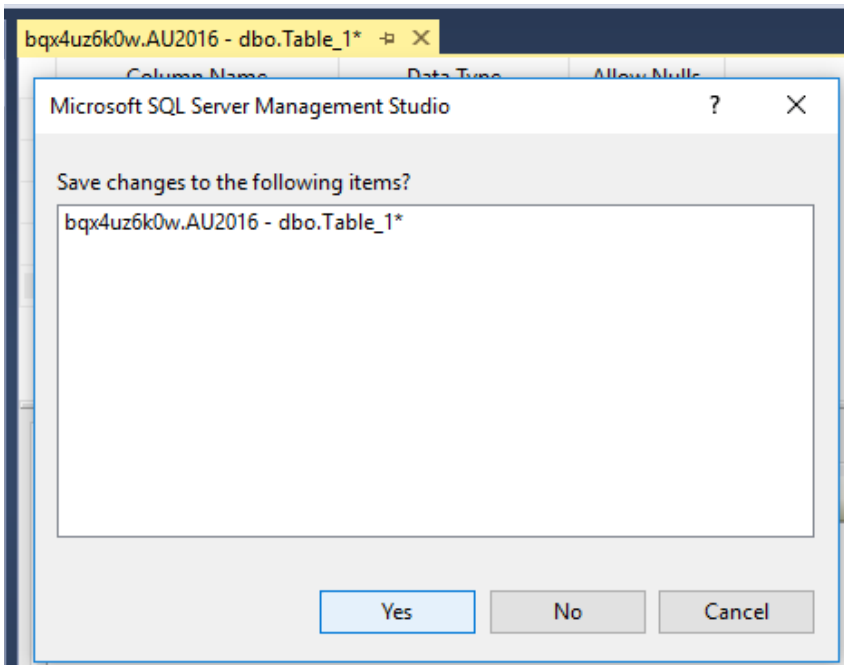
You may notice that this code only opens the Database Connection and then closes it. If this is successful, we are well on our way to reading from and writing to the database tables we create.

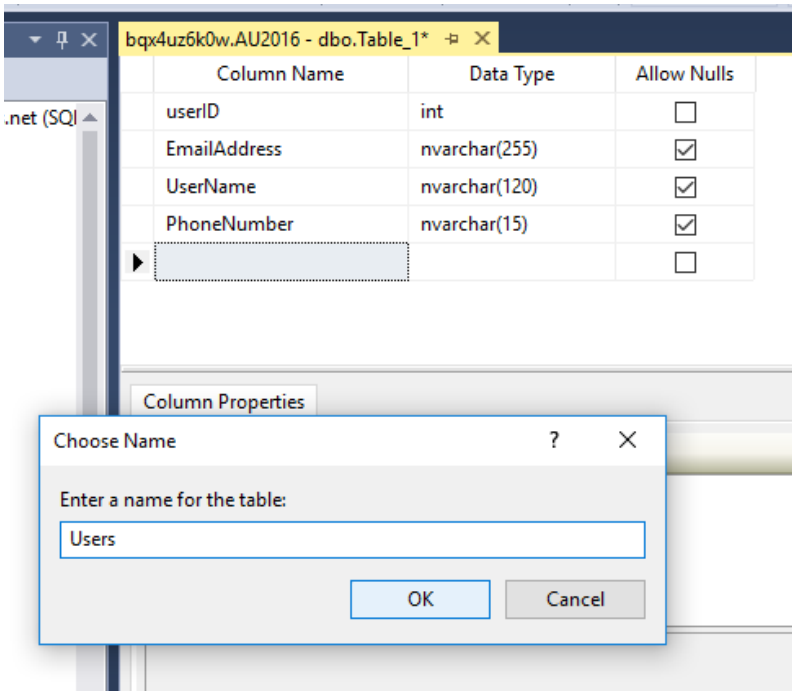
The next thing we are going to do is create a Table so we can look at code to read from and write to the database.



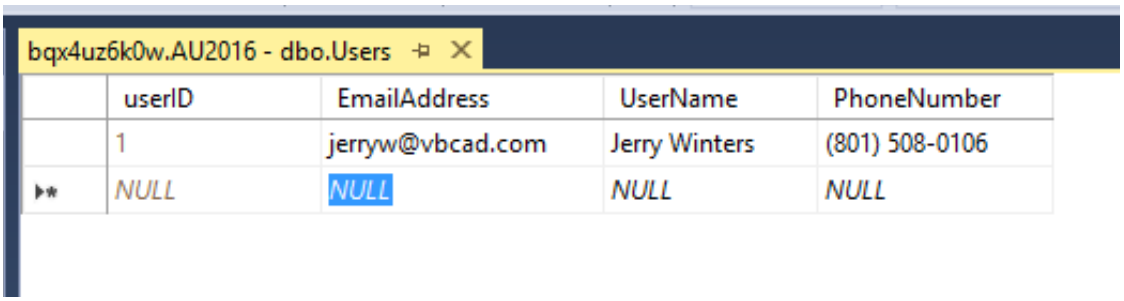

bqx4uz6k0w.AU2016 - dbo.Table_1*
Column Name
Data Type
Allow Nulls
userID
int
Column Properties
Has Non-SQL Server Subscriber
No
Identity Specification
Yes
(Is Identity)
Yes
Identity Increment
1
Identity Seed
1
Indexable
Yes
Is Columnset
No
Is Sparse
No
Merge-published
No
Not For Replication
No
(Is Identity)


bqx4uz6k0w.AU2016 - dbo.Table_1*
Column Name
Data Type
Allow Nulls
userID
int
EmailAddress
nvarchar(255)
UserName
nvarchar(120)
PhoneNumber
nvarchar(15)

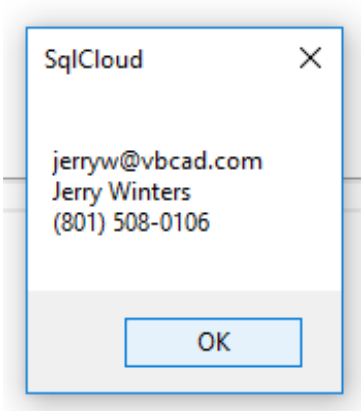

bqx4uz6k0w.AU2016 - dbo.Table_1*
Microsoft SQL Server Management Studio
Save changes to the following items?
bqx4uz6k0w.AU2016 - dbo.Table_1*
Yes
No
Cancel




Now we have a Table named “Users”, we can “Edit” it in SQL Server Management Studio and manually add a new user as shown below.

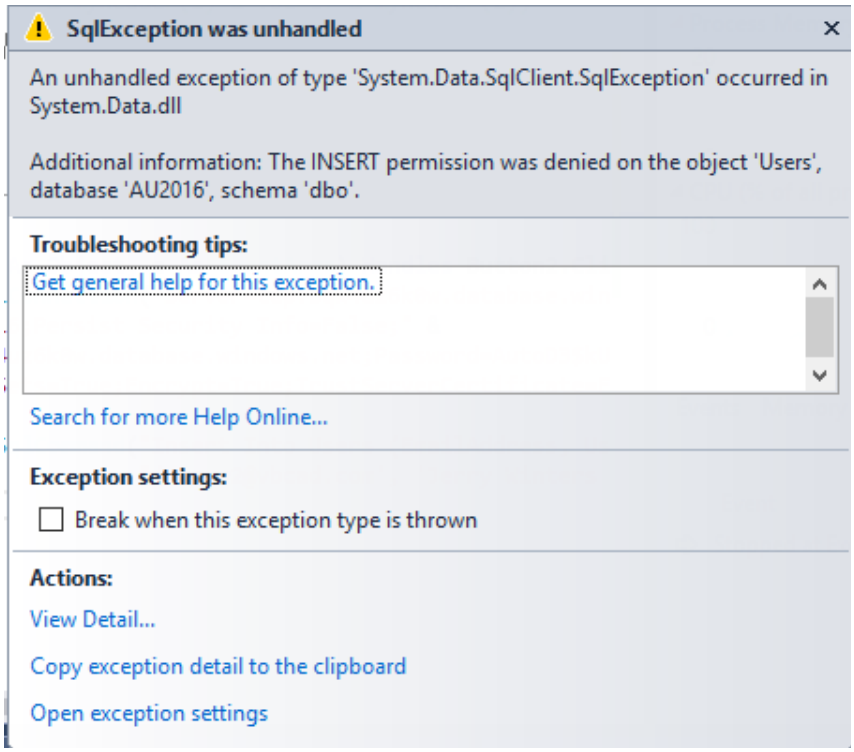



```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim myDB As New SqlClient.SqlConnection("Server=tcp:bqx4uz6k0w.database.windows.net,1433;" &
           "Initial Catalog=AU2016;Persist Security Info=False;" &
           "User ID=au2016ro@bqx4uz6k0w.database.windows.net;Password=AutoD3$kUniver$ity2016ro;" &
           "MultipleActiveResultSets=True;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;")
    myDB.Open()
    Dim myComm As New SqlClient.SqlCommand("Select EmailAddress, UserName, PhoneNumber from Users", myDB)
    Dim myReader As SqlClient.SqlDataReader = myComm.ExecuteReader
    While myReader.Read
        MsgBox(myReader(0) & vbCrLf & myReader(1) & vbCrLf & myReader(2))
    End While
    myReader.Close()
    myComm.Dispose()
    myDB.Close()
    myDB.Dispose()
End Sub
```



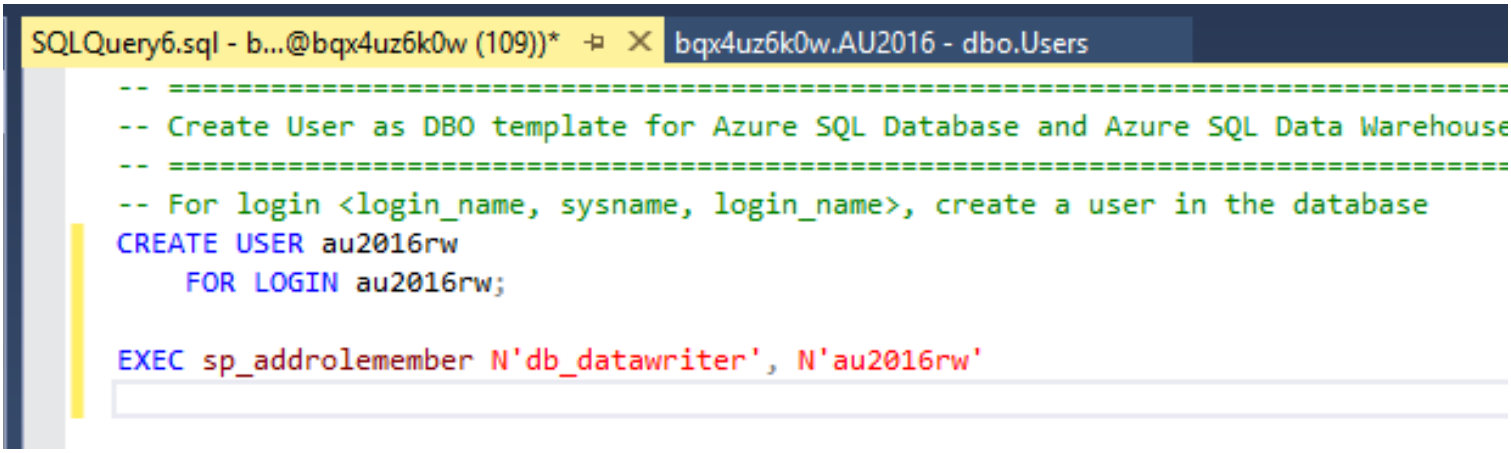

Adding a few lines of code to the code we already had allows us to read from the Database. Please note that we are using the “Read Only” user.

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim myDB As New SqlClient.SqlConnection("Server=tcp:bqx4uz6k0w.database.windows.net,1433;" &
           "Initial Catalog=AU2016;Persist Security Info=False;" &
           "User ID=au2016ro@bqx4uz6k0w.database.windows.net;Password=AutoD3$kUniver$ity2016ro;" &
           "MultipleActiveResultSets=True;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;")
    myDB.Open()
    Dim myComm As New SqlClient.SqlCommand("Insert Into Users (EmailAddress, UserName, PhoneNumber) VALUES ('" &
                                           "jerryw2@vbcad.com', 'Jerry Winters 2', '123-456-7890')", myDB)
    myComm.ExecuteNonQuery()
    myComm.Dispose()
    myDB.Close()
    myDB.Dispose()
End Sub
```



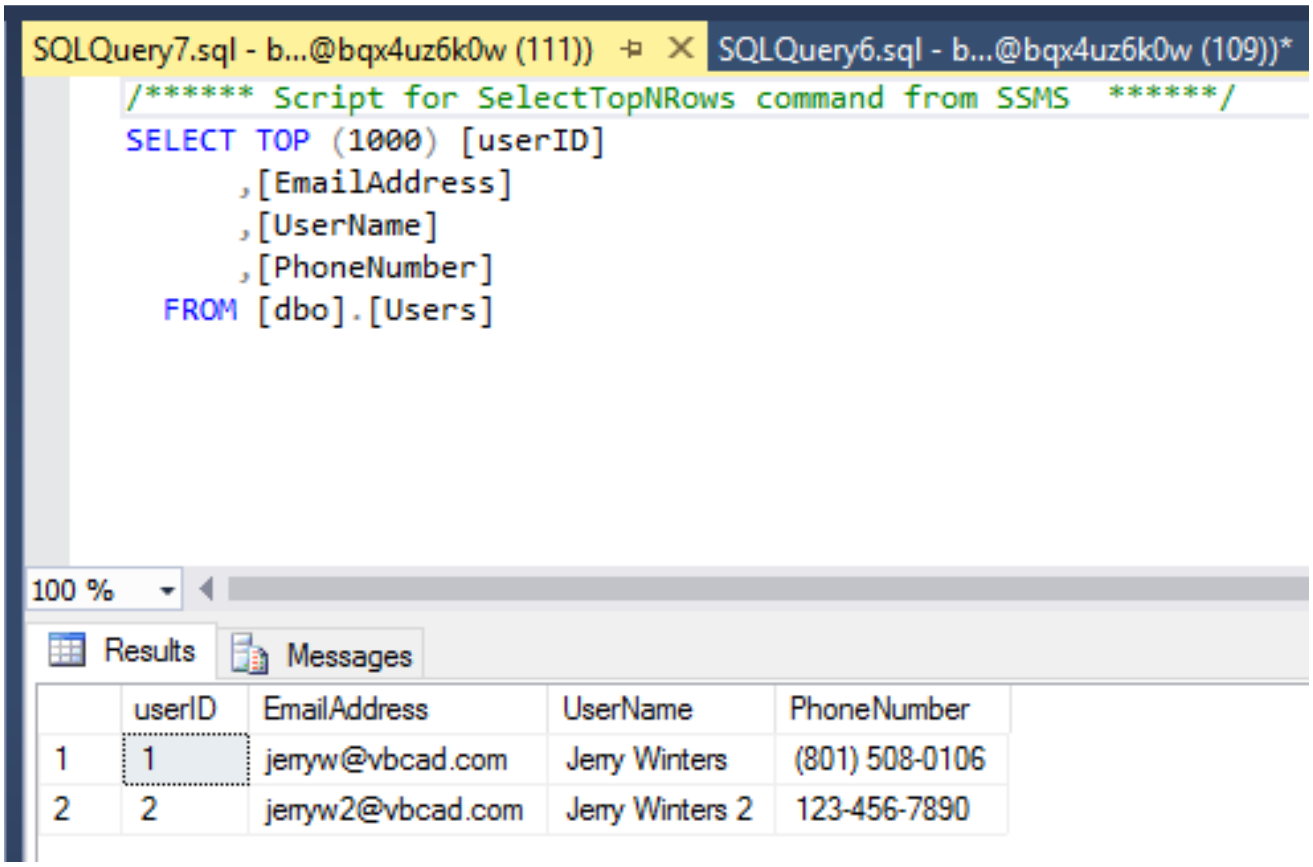

If we attempt to insert a record into our table using the “Read Only” user, we see this error.

- Create a new User based on the “au2016rw” login and add the “db_datawriter” role this new user.



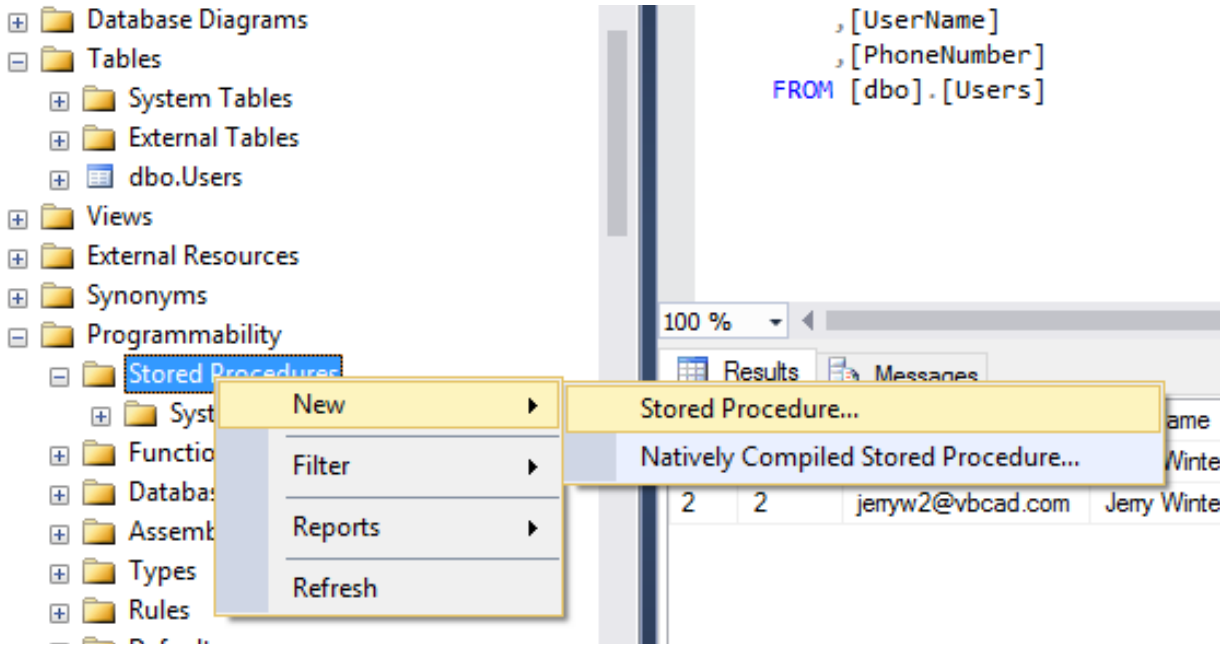
```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim myDB As New SqlClient.SqlConnection("Server=tcp:bqx4uz6k0w.database.windows.net,1433;" &
           "Initial Catalog=AU2016;Persist Security Info=False;" &
           "User ID=au2016rw@bqx4uz6k0w.database.windows.net;Password=AutoD3$kUniver$ity2016rw;" &
           "MultipleActiveResultSets=True;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;")
    myDB.Open()
    Dim myComm As New SqlClient.SqlCommand("Insert Into Users (EmailAddress, UserName, PhoneNumber) VALUES ('" &
                                           "jerryw2@vbcad.com', 'Jerry Winters 2', '123-456-7890')", myDB)
    myComm.ExecuteNonQuery()
    myComm.Dispose()
    myDB.Close()
    myDB.Dispose()
End Sub
```

The code needs to be modified to use the “ReadWrite” user we just created. After doing this, the code will run without error and insert a new record in the table.



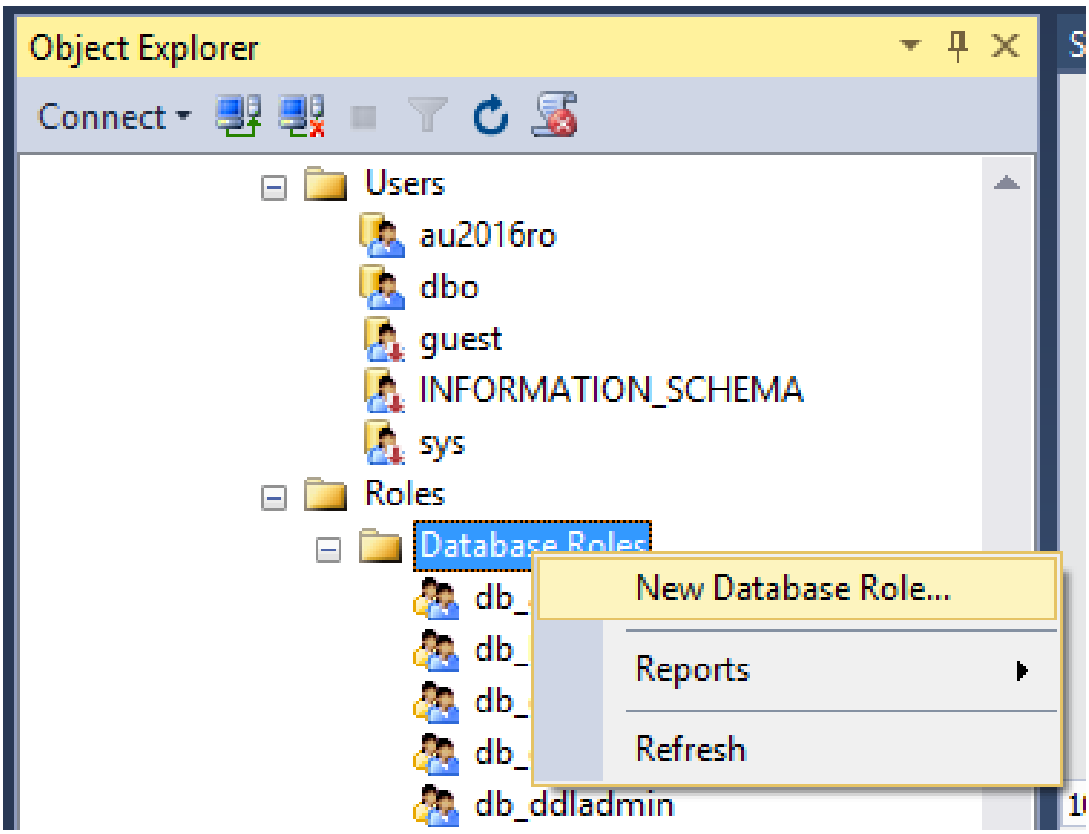

Insert statements work but there are times that using a Stored Procedure is preferred due to performance and security.

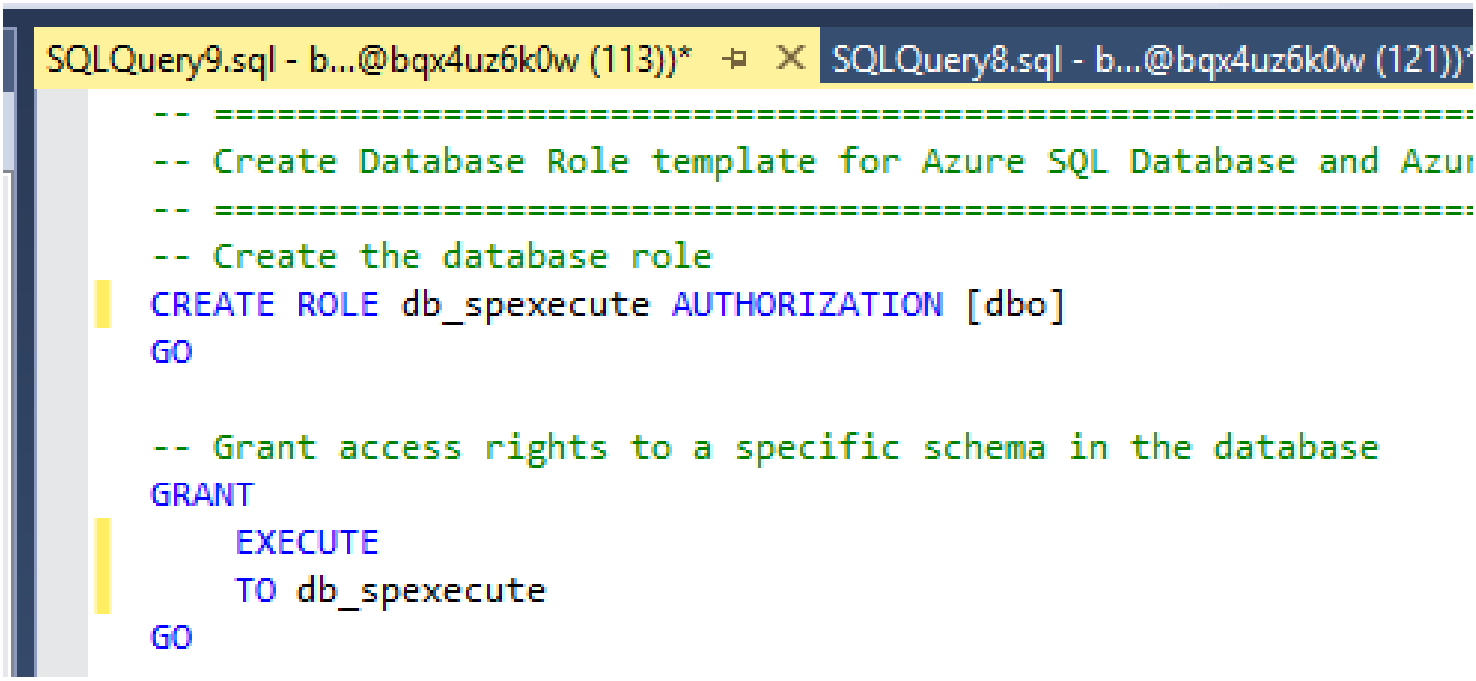
We are going to create a new Stored Procedure that will be used to insert records into the database.




```
SQLQuery8.sql - b...@bqx4uz6k0w (121))*   SQLQuery7.sql - b...@bqx4uz6k0w (111))
-- ============================================================
-- Create Stored Procedure Template for Azure SQL Database and Azure SQ
-- ============================================================
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE AddUser
    -- Add the parameters for the stored procedure here
    @EmailAddress nvarchar(255) = '',
    @UserName nvarchar(120) = '',
    @PhoneNumber nvarchar(15) = ''
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    Insert Into Users (EmailAddress, UserName, PhoneNumber)
    VALUES
    (@EmailAddress, @UserName, @PhoneNumber)
END
GO
```

db_datareader and db_datawriter are roles that can read from and write to the database. However, in order to use a Stored Procedure, we need to create a new Role (db_datareader and db_datawriter are already there for us to use) that has the "Execute" right assigned to it.



A new database role has been created that has Execute rights and now we are going to create a new Login and Username that can make use of this newly created role.

```
-- ============================================================
-- Create SQL Login template for Azure SQL Database and
-- ============================================================

CREATE LOGIN au2016execute
    WITH PASSWORD = 'AutoD3$kUniver$ity2016execute'
GO
```

```
-- ============================================================
-- Create User as DBO template for Azure SQL Database and Az
-- ============================================================
-- For login <login_name, sysname, login_name>, create a use
CREATE USER au2016execute
    FOR LOGIN au2016execute
GO

-- Add user to the database owner role
EXEC sp_addrolemember N'db_spexecute', N'au2016execute'
GO
```

Now let's write some code to make use of the newly created Stored Procedure.

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Dim myDB As New SqlClient.SqlConnection("Server=tcp:bqx4uz6k0w.database.windows.net,1433;" &
            "Initial Catalog=AU2016;Persist Security Info=False;" &
            "User ID=au2016execute@bqx4uz6k0w.database.windows.net;Password=AutoD3$kUniver$ity2016execute;" &
            "MultipleActiveResultSets=True;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;")
    myDB.Open()
    Dim myComm As New SqlClient.SqlCommand("AddUser", myDB)
    myComm.CommandType = CommandType.StoredProcedure
    myComm.Parameters.Add("@EmailAddress", SqlDbType.NVarChar, 255).Value = "jerryw3@vbcad.com"
    myComm.Parameters.Add("@UserName", SqlDbType.NVarChar, 120).Value = "Jerry Winters 3"
    myComm.Parameters.Add("@PhoneNumber", SqlDbType.NVarChar, 15).Value = "(123) 456-7890"
    myComm.ExecuteNonQuery()
    myComm.Dispose()
    myDB.Close()
    myDB.Dispose()
End Sub
```

## Review

SQL Azure Databases give us the ability to rapidly and inexpensively provide access to data globally with a very very very high degree of up-time, built-in replication, and scalability. As we have seen here, in 90 minutes or less we can create a new SQL Azure database and make our data for our AutoCAD Add-ins accessible globally.

I hope this class has been useful and will allow you to begin creating your own SQL Azure databases.

Thanks for your time.

Jerry Winters
(jerryw@vbcad.com)