# Directions

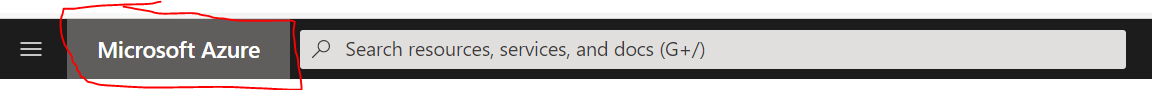
1. Complete the following steps.
2. Screenshot where directed.
3. Submit the screenshots to Blackboard. **Make sure you black out your passwords in the connection string!**

# Overview

You will use Azure to host your Premiere database. Once this is done, you are going to configure SSMS to talk with the Azure database. Finally, you will modify your existing premiere GUI to read from Azure instead of MySQL.

Please note that the Azure screen changes daily. I cannot possibly keep up with the changes in a static document. Do your best to find what I am describing.

# Register with Azure

1. Navigate to azure.microsoft.com
2. Sign in to or create a Microsoft account.
   1. You can use Github account here as well.
3. Create an Azure account by clicking “Start Free.” Debit or Credit card required.
4. Walk through the process to setup the account.
5. **Do not use any of the getting started links.**
6. Logout, and log back in to **portal.azure.com**
7. Click on the Microsoft Azure title in the bar  
   

# SQL Database Creation

1. Find *SQL databases*.
2. Click Create SQL database.
3. Enter a name for your database under database details.
4. Under Server, select create new. Enter server details and create and admin username and password.
   1. Do not use special characters in your username.
   2. If you use an email for the username, you will need to read this document.  
      <https://techcommunity.microsoft.com/t5/azure-database-support-blog/providing-the-server-name-explicitly-in-user-names-for-azure-sql/ba-p/368942>
5. Under Compute + Storage, click Configure database. Click the link that takes you to the Basic database.   
   A picture containing text

   Description automatically generated  
   The Basic option is the best choice, and may cost you around $5.00 per month.
   1. Create the required servers or resource groups
   2. Be **very careful** here. If you choose poorly, it can cost 200 per month.
   3. After the term is finished, you can delete the database to save some money.
   4. ***Remember the username and password you created!!!!!***
6. Click Review + create, then create.

Screenshot your created database

# Create a SQL Server Firewall

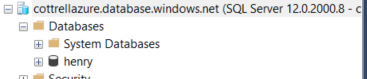
By default, your SQL Server will not let anyone outside of Azure connect to it. We are going to add a firewall setup to solve this problem. You will need the public IP address of your router. If you move to a different Internet, you will need to repeat this step.

1. Visit <https://ipchicken.com> to get your public IP address. Write it down or keep the screen open. You will need the exact sets of 4 digits.
2. Find your database and click on it to open the configuration pane.
3. Find and click *Set Server Firewall*.
4. At the bottom enter a name, and the IP address.
5. Press enter then click Save at the top.

# Connect SSMS to Azure

You can use the database tools in Azure. However, they are clunky and tedious. It is much better to use SSMS as your client. The SSMS to Azure will work just like the SSMS to SQL Server connection. You may encounter an occasional lag.

1. On Azure, wait for the database to be deployed. I will not take too long.
2. Find your database and click on it.
3. Copy the server name. When you hover over it, Azure provides a handy copy to clipboard link.
4. Start SSMS on your computer.
   1. If you already started SSMS and connected to the local SQL Server, click the connect dropdown in Object Explorer. Select Database Engine
5. Paste the server name into the server name.
6. Change authentication to SQL Server Authentication
7. Enter the username and password you created with the server.
8. You will be prompted to sign into your Azure account to make a new Firewall rule. Select “Add my subnet IP address range” and click ok.
   1. Periodically your home IP address may change. If so, then you will need to add another firewall entry to your database to allow you to connect.
9. You now have access to the database that you created via Azure.
10. Use the Premier code to create a new database.

Screenshot the created database in SSMS. Ensure you include the link to Azure, like mine.  
  
A screenshot of a computer

Description automatically generated

# Download the MS SQL JDBC Drivers

You can now use Azure to host the data. This makes your app able to run anywhere. We will use Java Database Connectivity (JDBC) to connect Java to Azure SQL.

For reference, visit this page <https://docs.microsoft.com/en-us/sql/connect/jdbc/connection-url-sample?view=sql-server-ver15>

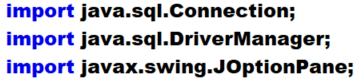
1. Download the JDBC Driver from <https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15>
2. Extract the files to a folder of your choice. Preferably one that will not be deleted in the future.

# Get the Connection String to Azure

You will need the connection string for your Azure database. Conveniently, Azure provides it.

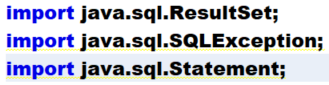
1. Go back to Azure and open your database.
2. Click the link “Show database connection strings”
3. This page shows all the connection strings for a variety of connection possibilities. You want **JDBC**.
   1. Incidentally the JDBC JAR file are available on this page as well.
4. Copy the entire connection string. You will need it in the next step.
5. Paste it into Notepad or another editor
6. **In the connection string is a section {your password here} replace it with the password you created.**
   1. **Do not leave the {}**
7. Copy the modified connection string

# Connect Java to Azure

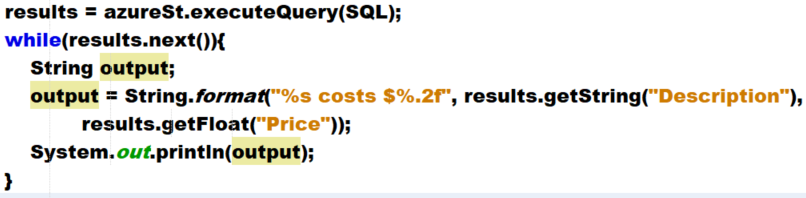
1. Create a new Java project
2. Add the appropriate JAR file to your project. Most likely you are using the JRE8 Java.
3. Add these imports  
   
4. Add this to your main. Paste your **modified** connection string where the code says YourConnectionString, inside of the quotes. Watch the “ and ); on the Connection line  
   
5. Run the program. Move on when it connects successfully.

# Display Data from Azure

The code to retrieve data from Azure is very similar to that of MySQL. It is the same concept. Execute a query and store the result in a ResultSet. Then walk through the result set. Catch errors that occur.

1. Add these imports to your project. Keep the existing ones of course.  
   
2. You can comment out the JOptionPane in main.

Add the following lines ***before*** you close the connection to the database

1. Create a Statement. Recall that this is what runs the code across the bridge to the Azure server.  
   
2. Create a ResultSet and SQL statement  
   
3. Execute the Query and parse the results  
   
4. Recall that results.getX() can take either the name of the field, or the numeric position of the field. It is preferred to use the name over the number.

Screenshot when your code displays the parts.  
A screenshot of a computer program

Description automatically generated

# Your turn!!

1. Make a backup copy of your Premiere GUI assignment. Send it to GitHub or anywhere you feel is safe.
2. Modify your existing Premiere GUI to read from Azure. Include your screenshots that prove that you are reading from Azure rather than MySQL.

Submit proof that you are reading from Azure and not MySQL

# References

* <https://docs.microsoft.com/en-us/sql/connect/jdbc/connection-url-sample?view=sql-server-ver15>
* <https://docs.microsoft.com/en-us/sql/connect/jdbc/using-the-jdbc-driver?view=sql-server-ver15>
* <https://docs.microsoft.com/en-us/sql/connect/jdbc/building-the-connection-url?view=sql-server-ver15#Connectingintegrated>