Microsoft

Working with Hive in HDInsight

Contents

Contents

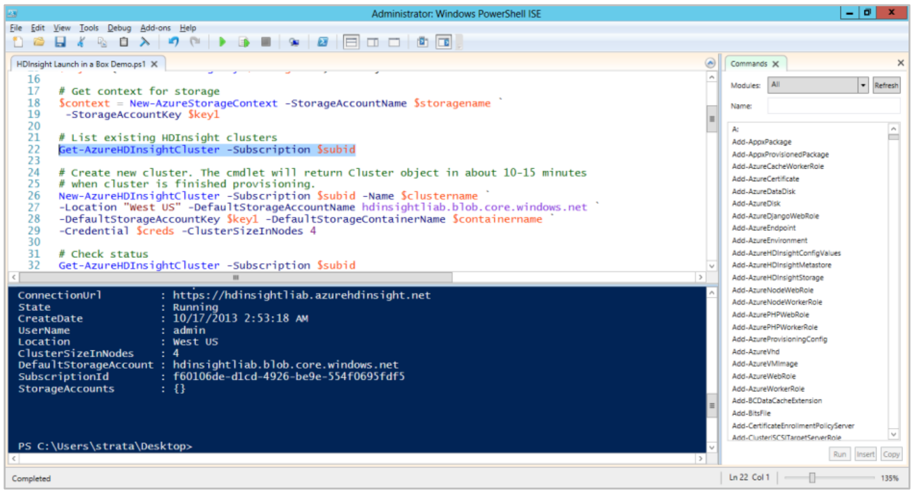
Introduction 3
Creating an Azure storage account 5
Creating an HDInsight Cluster 7
Loading MapReduce data into Azure Storage 11
Running Hive commands as a job.......................... 14
Running Hive commands using Invoke-Hive....... 17
Using the Hive console with HDInsight.................. 24
Terms of Use 27

Introduction

In this hands-on lab, you will setup a Microsoft Azure storage account along with an HDInsight Hadoop cluster. You will then upload a data file that was processed using a map reduce job that was created using the TR19 Exploring MapReduce in HDInsight using .NET lab. Then, you will use Hive to query the data on your HDInsight cluster help you learn the essentials of Hive.

The hands-on-lab requires a Microsoft Azure account.

During TechReady, please raise your hand and ask a Technical Learning Guide for a pre-configured Azure trial account.



```powershell

# Get context for storage
$context = New-AzureStorageContext -StorageAccountName $storagename `
 -StorageAccountKey $key1

# List existing HDInsight clusters
Get-AzureHDInsightCluster -Subscription $subid

# Create new cluster. The cmdlet will return Cluster object in about 10-15 minutes
# when cluster is finished provisioning.
New-AzureHDInsightCluster -Subscription $subid -Name $clustername `
-Location "West US" -DefaultStorageAccountName hdinsightliab.blob.core.windows.net `
-DefaultStorageAccountKey $key1 -DefaultStorageContainerName $containername `
-Credential $creds -ClusterSizeInNodes 4

# Check status
Get-AzureHDInsightCluster -Subscription $subid
```

```
ConnectionUrl         : https://hdinsightliab.azurehdinsight.net
State                 : Running
CreateDate            : 10/17/2013 2:53:18 AM
UserName              : admin
Location              : West US
ClusterSizeInNodes    : 4
DefaultStorageAccount : hdinsightliab.blob.core.windows.net
SubscriptionId        : f60106de-d1cd-4926-be9e-554f0695fdf5
StorageAccounts       : {}


PS C:\Users\strata\Desktop>
```

First Time Setup Instructions

Configure the PowerShell environment as follows:

1. Create a directory called C:\data.
2. Go to the D:\TR19HiveLab folder. Then right click on the file "TR19 Hive Lab.ps1" and choose the Edit command to launch the PowerShell script in the Windows PowerShell ISE.
3. Log into your Microsoft Azure Management portal with your Live ID at https://manage.windowsazure.com. Choose the option to "Keep me signed in".
4. Select the command `Get-AzurePublishSettingsFile` in the editor and press F8 to execute the selection. This will launch a new tab in IE and then prompt you to open or save the .publishsettings file. Select the Save option to copy the file into the c:\data folder.
5. Edit the `Import-AzurePublishSettingsFile` command in the editor include the full file path of the .publishsettings file in your C;\data folder. For example: `Import-AzurePublishSettingsFile "C:\data\Greg_AzurePlatform_Subscription-10-22-2013-credentials.publishsettings"` and then press F8 to execute the command. This completes the one time setup for PowerShell so that it can talk with Microsoft Azure.
6. Verify the subscription by executing the following command: `Get-AzureSubscription –Default`

NOTE: If you completed the TR19 Exploring MapReduce in HDInsight using .NET lab and kept your HDInsight cluster, storage account and student container, you can skip to section Loading MapReduce data into Azure Storage.

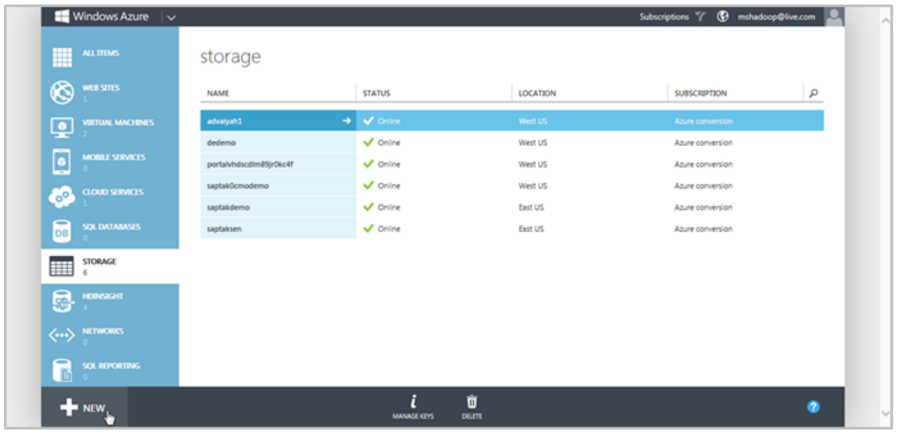
Creating an Azure storage account

In this task, you will create a storage account and a container within the account you will use with the HDInsight cluster.

NOTE: Skip this step if you are using a shared account

1. Log into the Microsoft Azure management portal at https://manage.windowsazure.com/.

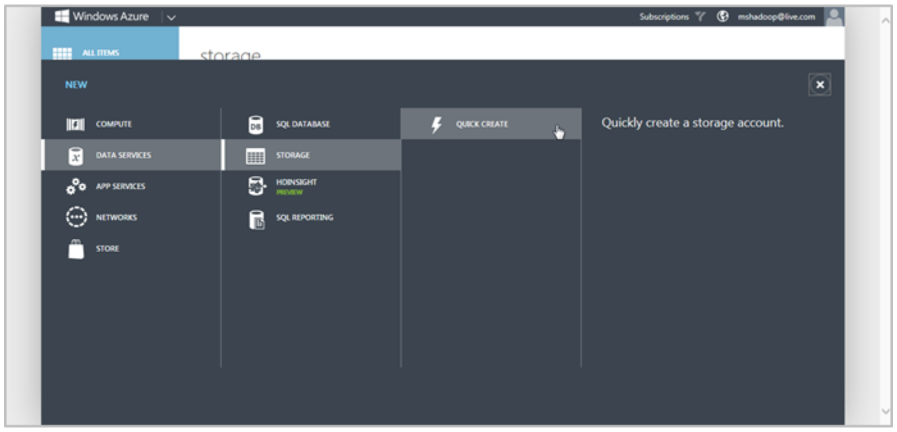
2. Select the STORAGE page and click NEW in the bottom left corner.



storage

NAME	STATUS	LOCATION	SUBSCRIPTION
advaiyah1	✓ Online	West US	Azure conversion
dedemo	✓ Online	West US	Azure conversion
portalvhdscdlm89jr0kc4f	✓ Online	West US	Azure conversion
saptak0cmodemo	✓ Online	West US	Azure conversion
saptakdemo	✓ Online	East US	Azure conversion
saptaksen	✓ Online	East US	Azure conversion

3. Click **QUICK CREATE**.



4. For the URL, you need to provide a unique storage account name that is from 3-15 characters. In this example, the storage account name will be the same as the HDInsight cluster name. You will need to replace this name with one that is unique for your training session. All of the exercises use the name: <storage_account_name>. You can search and replace this document <storage_account_name> with the one you choose.
   Complete the required fields:

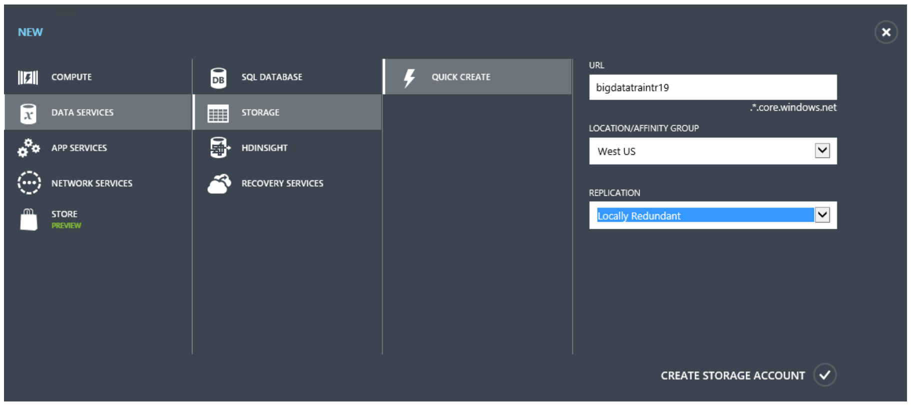
   a. URL: <storage_account_name>

b. Location: **West US**.

   Note, there are many to choose from, but you must select a location that supports HDInsight. These include West US, East US and North Europe at this time.

c. GEO-REPLICATION: Deselect it.

d. Click CREATE STORAGE ACCOUNT to complete the operation.



URL
bigdatatraintr19
.*.core.windows.net
LOCATION/AFFINITY GROUP
West US
REPLICATION
Locally Redundant

This operation can take between 1 to 5 minutes to complete.

Creating an HDInsight Cluster

In this section, you will create an HDInsight cluster that aligns with the storage account you created in the previous exercise.

Key Points

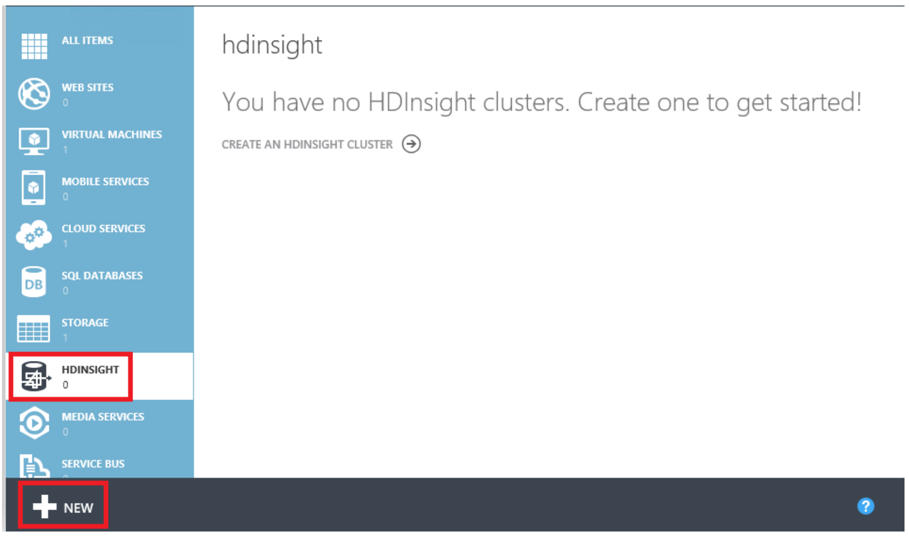
The Azure Management Portal provides a user interface for creating a basic HDInsight cluster.

Demonstration

In this first demo, you will create a new HDInsight cluster. This step takes about 10 minutes to complete.

1. Navigate to the management portal – https://manage.windowsazure.com.

2. Select the HDINSIGHT page and click on the NEW button in the lower left corner.



You have no HDInsight clusters. Create one to get started!

3. You have two options: QUICK CREATE or CUSTOM CREATE. As of June 10, 2014, the default version created with QUICK CREATE is 3.0.
   NOTE: Do not select the 3.1 preview edition as there are some incompatibilities with the PowerShell cmdlets.

   Select CUSTOM CREATE with the following parameters:

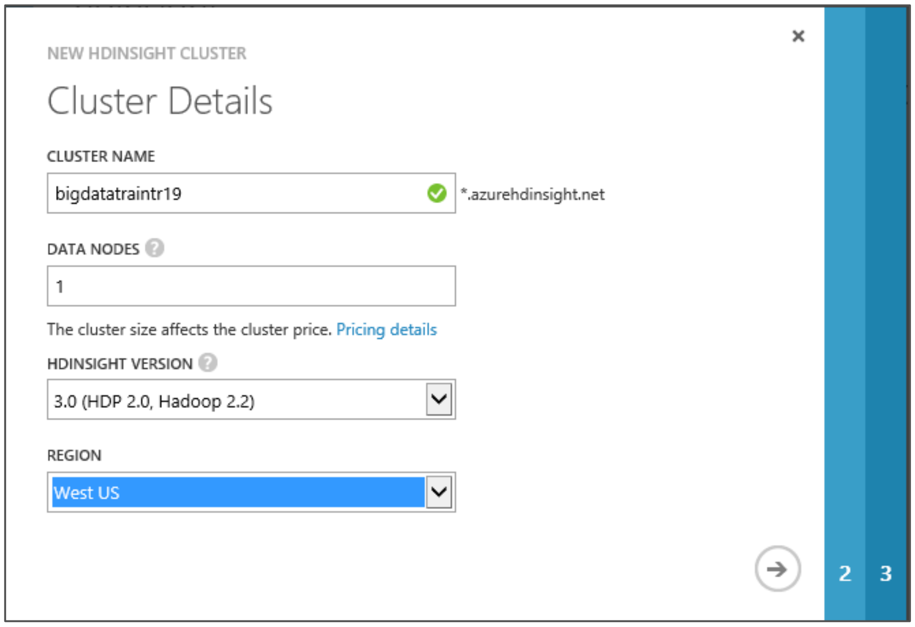
   a. CLUSTER NAME: <cluster_name>

   b. DATA NODES: 1

   c. HDINSIGHT VERSION: 3.0 (HDP 2.0, Hadoop 2.2)

d. REGION: West US (needs to match your storage account)

e. Click next page arrow.



NEW HDINSIGHT CLUSTER

Cluster Details

CLUSTER NAME: bigdatatraintr19 *.azurehdinsight.net

DATA NODES: 1

The cluster size affects the cluster price. Pricing details

HDINSIGHT VERSION: 3.0 (HDP 2.0, Hadoop 2.2)

REGION: West US

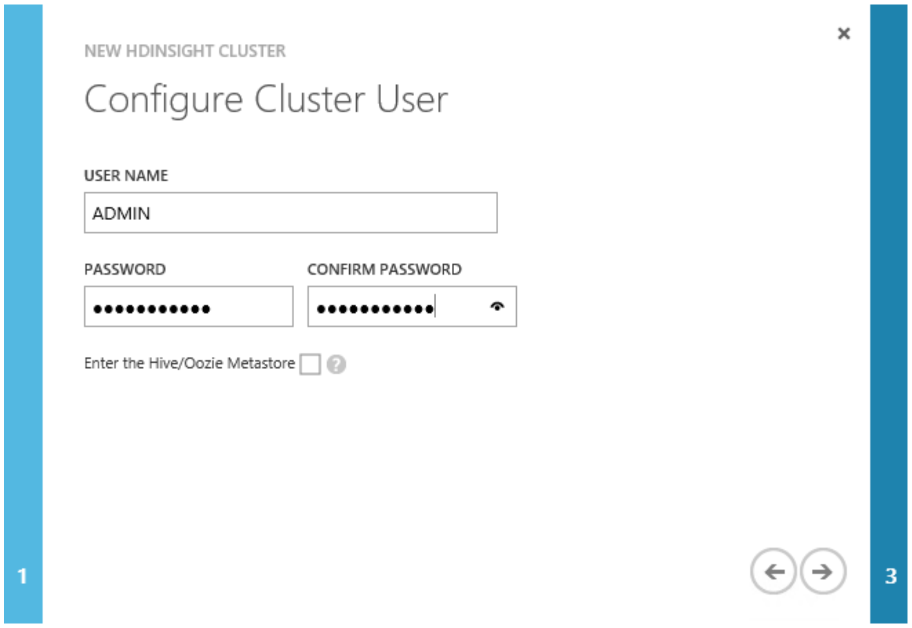
4. For the Configure Cluster User page, enter in the following values:

    a. USER NAME: ADMIN

    b. PASSWORD: Pass@word12

    c. CONFIRM PASSWORD: Pass@word12

    d. Click the next page arrow.


NEW HDINSIGHT CLUSTER

Configure Cluster User

USER NAME: ADMIN

PASSWORD: ●●●●●●●●●●●

CONFIRM PASSWORD: ●●●●●●●●●●●

Enter the Hive/Oozie Metastore ☐

5. For the Storage Account page, enter in the following:

    a. STORAGE ACCOUNT: Use Existing Storage

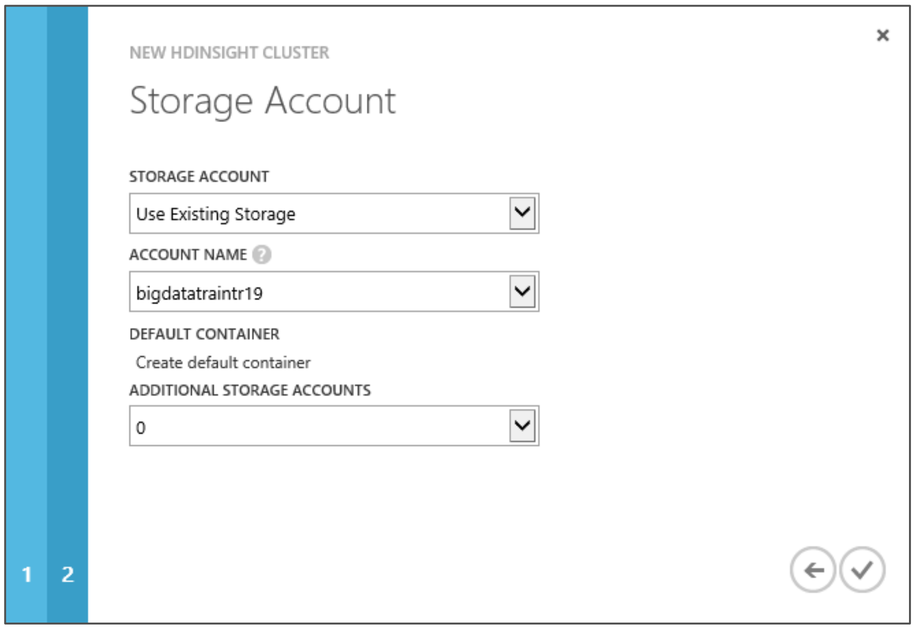
       NOTE: If you don't see your storage account available for selection, your default location may be not be the same as West US. In which case, use the CUSTOM CREATE option.

    b. ACCOUNT NAME: select your account name from the region.

    c. DEFAULT CONTAINER: Create default container

    d. ADDITIONAL STORAGE ACCOUNTS: 0 (this is used to link multiple storage accounts to your cluster)

    e. Click the completion check box.


NEW HDINSIGHT CLUSTER

Storage Account

STORAGE ACCOUNT: Use Existing Storage

ACCOUNT NAME: bigdatatraintr19

DEFAULT CONTAINER: Create default container

ADDITIONAL STORAGE ACCOUNTS: 0

This operation takes about 10 minutes to complete

6. Review the progress by clicking on the moving bar chart to see details. When the operation is compete, the status for the cluster shows as Running.

Summary

Using the Azure management portal is a convenient way to create a cluster. The CUSTOM CREATE operation offers more flexibility.

Loading MapReduce data into Azure Storage

In this exercise, you upload the MapReduce part-00000 file that was initially created during the TR19 Exploring MapReduce in HDInsight using .NET lab. A copy of the file is located in the D:\TR19HiveLab directory.

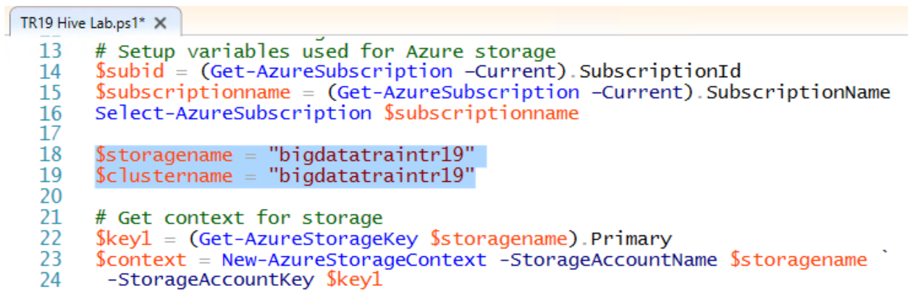
Key Points

- The Azure Storage cmdlets are the recommended way for working with data files with your storage account because you can create and use pseudo directories within a container.

Demonstration

1. Go to your open "TR19 Hive Lab.ps1" using the PowerShell ISE and scroll down to line 18.

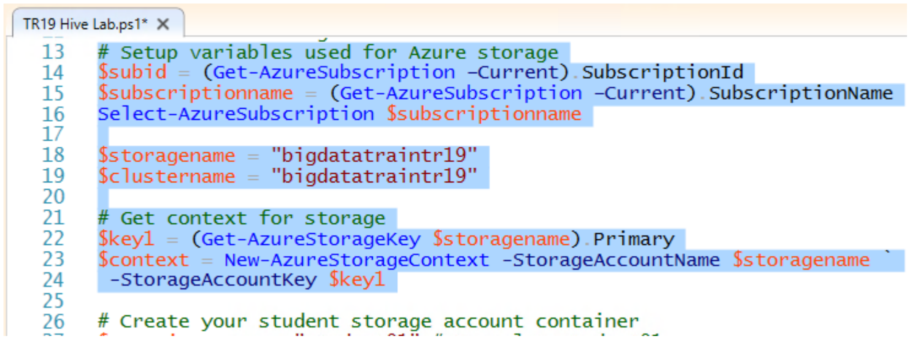
2. Change the line `$storagename = "<storage_account_name>"` to match your storage account name. Then change the line `$clustername = "<cluster_name>"` to match your cluster name.



TR19 Hive Lab.ps1*

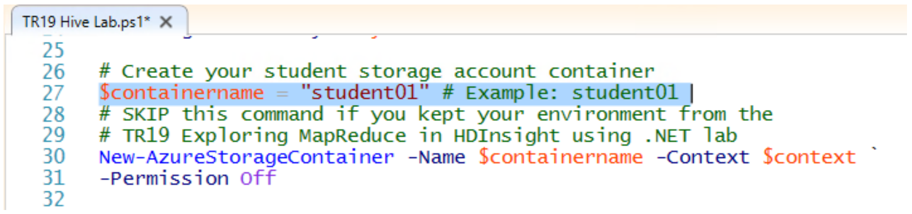
```
# Setup variables used for Azure storage
$subid = (Get-AzureSubscription -Current).SubscriptionId
$subscriptionname = (Get-AzureSubscription -Current).SubscriptionName
Select-AzureSubscription $subscriptionname

$storagename = "bigdatatraintr19"
$clustername = "bigdatatraintr19"

# Get context for storage
$key1 = (Get-AzureStorageKey $storagename).Primary
$context = New-AzureStorageContext -StorageAccountName $storagename `
 -StorageAccountKey $key1
```

3. Select the following lines of code in the editor and press F8 to setup the storage variables and set content for the storage account.



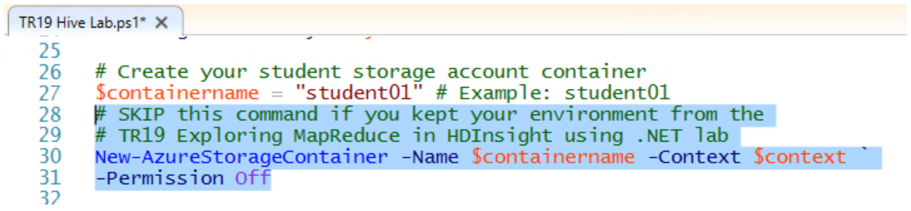
```
# Setup variables used for Azure storage
$subid = (Get-AzureSubscription -Current).SubscriptionId
$subscriptionname = (Get-AzureSubscription -Current).SubscriptionName
Select-AzureSubscription $subscriptionname

$storagename = "bigdatatraintr19"
$clustername = "bigdatatraintr19"

# Get context for storage
$key1 = (Get-AzureStorageKey $storagename).Primary
$context = New-AzureStorageContext -StorageAccountName $storagename `
 -StorageAccountKey $key1

# Create your student storage account container
```

4. Locate the line - `$containername = "<student_id>"` and replace the student number with your assigned number or just use `"student01"`.



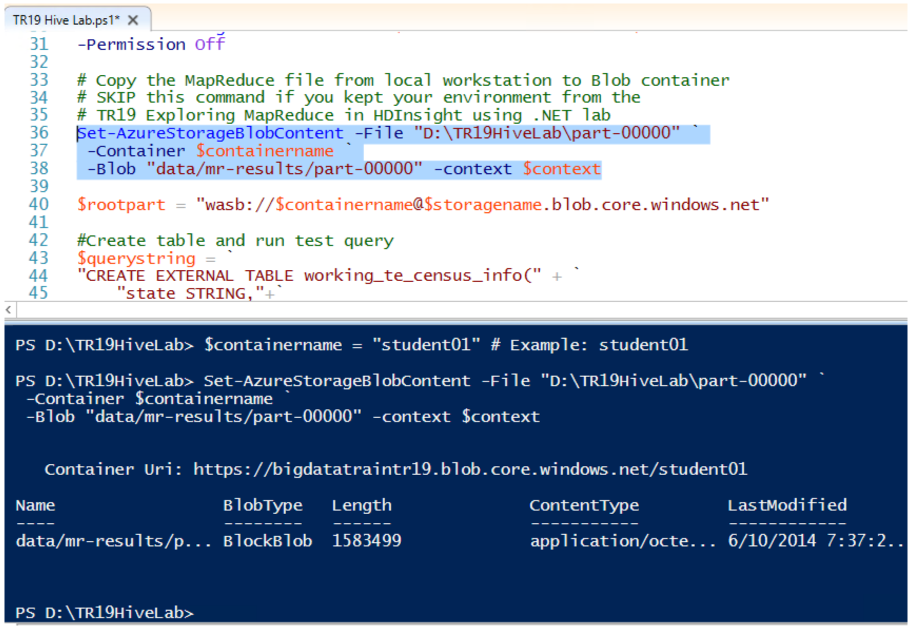
```

# Create your student storage account container
$containername = "student01" # Example: student01 |
# SKIP this command if you kept your environment from the
# TR19 Exploring MapReduce in HDInsight using .NET lab
New-AzureStorageContainer -Name $containername -Context $context `
-Permission Off

```

5. Create the container for the census files by selecting the following lines of code and pressing F8 to execute them.
NOTE: Skip this step if you retained your HDInsight cluster environment from the TR19 Exploring MapReduce in HDInsight using .NET lab.



```

# Create your student storage account container
$containername = "student01" # Example: student01
# SKIP this command if you kept your environment from the
# TR19 Exploring MapReduce in HDInsight using .NET lab
New-AzureStorageContainer -Name $containername -Context $context `
-Permission Off

```

6. Select the following lines of code to copy the MapReduce part-00000 file to your student container.
NOTE: Skip this step if you retained your HDInsight cluster environment from the TR19 Exploring MapReduce in HDInsight using .NET lab.



```
TR19 Hive Lab.ps1*
-Permission Off

# Copy the MapReduce file from local workstation to Blob container
# SKIP this command if you kept your environment from the
# TR19 Exploring MapReduce in HDInsight using .NET lab
Set-AzureStorageBlobContent -File "D:\TR19HiveLab\part-00000" `
 -Container $containername `
 -Blob "data/mr-results/part-00000" -context $context

$rootpart = "wasb://$containername@$storagename.blob.core.windows.net"

#Create table and run test query
$querystring = `
"CREATE EXTERNAL TABLE working_te_census_info(" + `
    "state STRING,"+`
```

```
PS D:\TR19HiveLab> $containername = "student01" # Example: student01

PS D:\TR19HiveLab> Set-AzureStorageBlobContent -File "D:\TR19HiveLab\part-00000" `
 -Container $containername `
 -Blob "data/mr-results/part-00000" -context $context


   Container Uri: https://bigdatatraintr19.blob.core.windows.net/student01

Name                 BlobType  Length              ContentType        LastModified
----                 --------  ------              -----------        ------------
data/mr-results/p... BlockBlob 1583499             application/octe... 6/10/2014 7:37:2..


PS D:\TR19HiveLab>
```

Summary

You now have the environment prepared to run Hive queries.

Running Hive commands as a job

In this exercise, you will learn the essentials how to execute HiveQL queries against HDInsight using the PowerShell interfaces. You will learn the similarities of HiveQL with SQL Server Transact-SQL.

Initial setup for this exercise

Go to the Windows PowerShell ISE with TR19 Hive Lab.ps1 already opened.

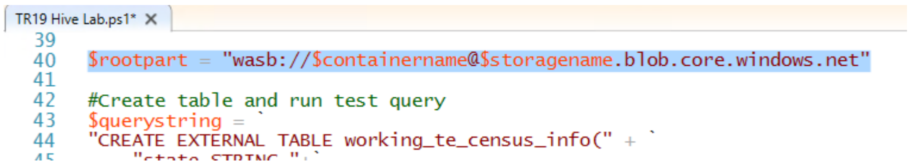
There are several ways to run Hive commands within PowerShell. In this first example, you will learn how to run a set of HiveQL commands as a job.

In this exercise, you will create a Hive table and perform a select statement against the table using the following sequence:

- Create a $querystring with the commands. Each command uses a semi-colon end of line delimiter.
- Create the Hive Job Definition.
- Start the HDInsight job
- Wait for the results
- View the job output.

Running Hive statements as a job

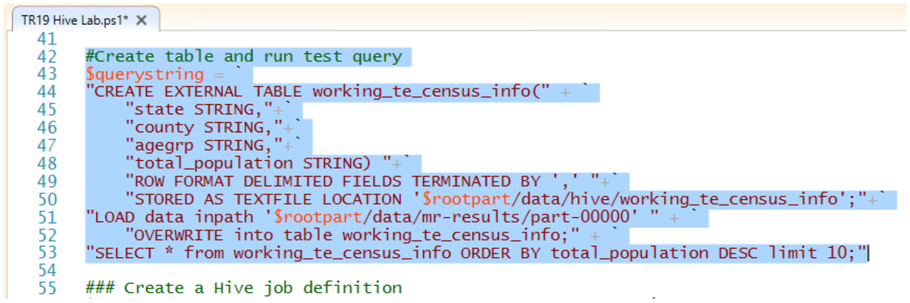
1. Select the following command to tell Hive where the data resides on Azure blob storage. By default, Hive will expect to locate files in the default container for the HDInsight cluster.



TR19 Hive Lab.ps1*

```

$rootpart = "wasb://$containername@$storagename.blob.core.windows.net"

#Create table and run test query
$querystring = `
"CREATE EXTERNAL TABLE working_te_census_info(" + `
```

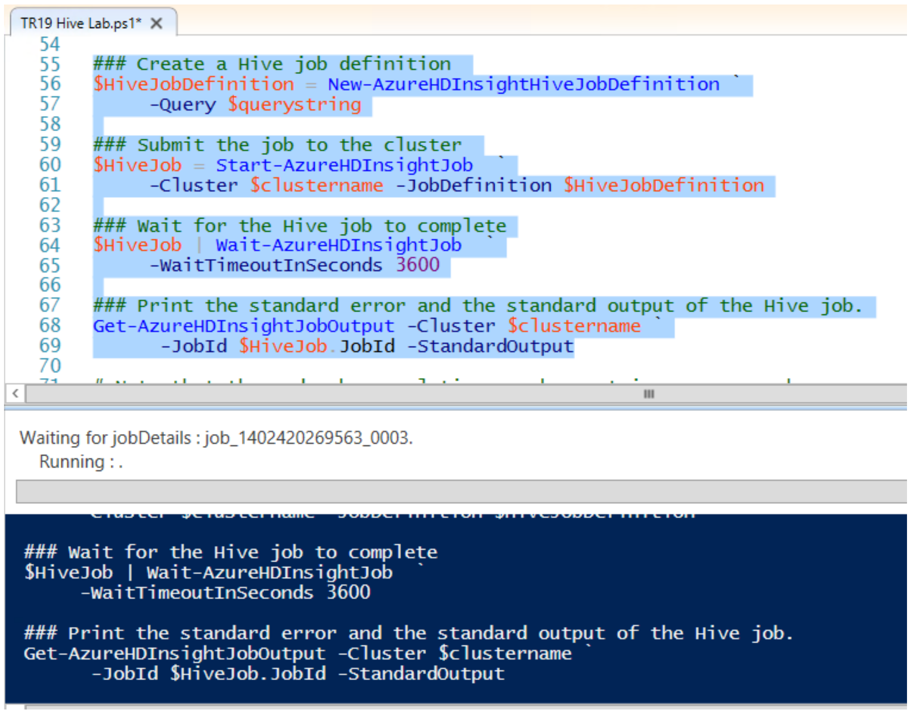
2. Select the following command and press F8 to create the query variable that will be used to create the Hive table based on the map-reduce results **part-00000** file.



```
#Create table and run test query
$querystring = `
"CREATE EXTERNAL TABLE working_te_census_info(" + `
    "state STRING,"+`
    "county STRING,"+`
    "agegrp STRING,"+`
    "total_population STRING) "+`
    "ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' "+`
    "STORED AS TEXTFILE LOCATION '$rootpart/data/hive/working_te_census_info';"+`
"LOAD data inpath '$rootpart/data/mr-results/part-00000' " + `
    "OVERWRITE into table working_te_census_info;" + `
"SELECT * from working_te_census_info ORDER BY total_population DESC limit 10;"

### Create a Hive job definition
```

In reviewing the syntax, you will see that the table was defined with the EXTERNAL option. This tells Hive to keep the data if the table is dropped. You'll also see that the data is stored as a TEXTFILE with comma separated fields. You can submit multiple HiveQL statements as long as they are separated by a semi-colon. Notice that the STRING data type does not include a length.

3. Select the following four commands in the script and press F8 to submit the Hive job and return the results.



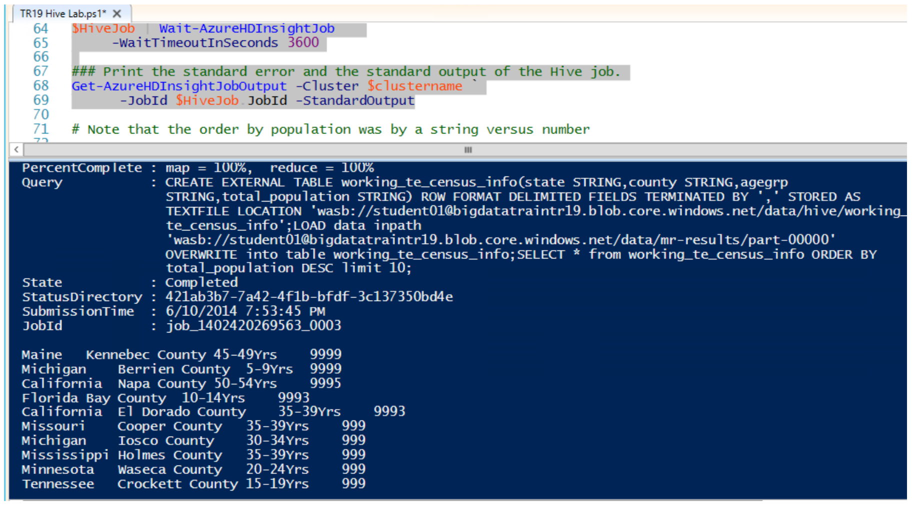
```
### Create a Hive job definition
$HiveJobDefinition = New-AzureHDInsightHiveJobDefinition `
     -Query $querystring

### Submit the job to the cluster
$HiveJob = Start-AzureHDInsightJob `
     -Cluster $clustername -JobDefinition $HiveJobDefinition

### Wait for the Hive job to complete
$HiveJob | Wait-AzureHDInsightJob `
     -WaitTimeoutInSeconds 3600

### Print the standard error and the standard output of the Hive job.
Get-AzureHDInsightJobOutput -Cluster $clustername `
      -JobId $HiveJob.JobId -StandardOutput
```

```
Waiting for jobDetails : job_1402420269563_0003.
  Running : .
```

```
### Wait for the Hive job to complete
$HiveJob | Wait-AzureHDInsightJob `
     -WaitTimeoutInSeconds 3600

### Print the standard error and the standard output of the Hive job.
Get-AzureHDInsightJobOutput -Cluster $clustername `
      -JobId $HiveJob.JobId -StandardOutput
```

You could have piped the commands together similar to the way you executed the streaming map-reduce job in the prior exercise.

4. Scroll through the console results to see the output.



```
$HiveJob | Wait-AzureHDInsightJob
    -WaitTimeoutInSeconds 3600

### Print the standard error and the standard output of the Hive job.
Get-AzureHDInsightJobOutput -Cluster $clustername `
      -JobId $HiveJob JobId -StandardOutput

# Note that the order by population was by a string versus number
```

```
PercentComplete : map = 100%,  reduce = 100%
Query           : CREATE EXTERNAL TABLE working_te_census_info(state STRING,county STRING,agegrp
                  STRING,total_population STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
                  TEXTFILE LOCATION 'wasb://student01@bigdatatraintr19.blob.core.windows.net/data/hive/working_
                  te_census_info';LOAD data inpath
                  'wasb://student01@bigdatatraintr19.blob.core.windows.net/data/mr-results/part-00000'
                  OVERWRITE into table working_te_census_info;SELECT * from working_te_census_info ORDER BY
                  total_population DESC limit 10;
State           : Completed
StatusDirectory : 421ab3b7-7a42-4f1b-bfdf-3c137350bd4e
SubmissionTime  : 6/10/2014 7:53:45 PM
JobId           : job_1402420269563_0003

Maine   Kennebec County 45-49Yrs    9999
Michigan    Berrien County  5-9Yrs  9999
California  Napa County 50-54Yrs    9995
Florida Bay County  10-14Yrs    9993
California  El Dorado County    35-39Yrs    9993
Missouri    Cooper County   35-39Yrs    999
Michigan    Iosco County    30-34Yrs    999
Mississippi Holmes County   35-39Yrs    999
Minnesota   Waseca County   20-24Yrs    999
Tennessee   Crockett County 15-19Yrs    999
```

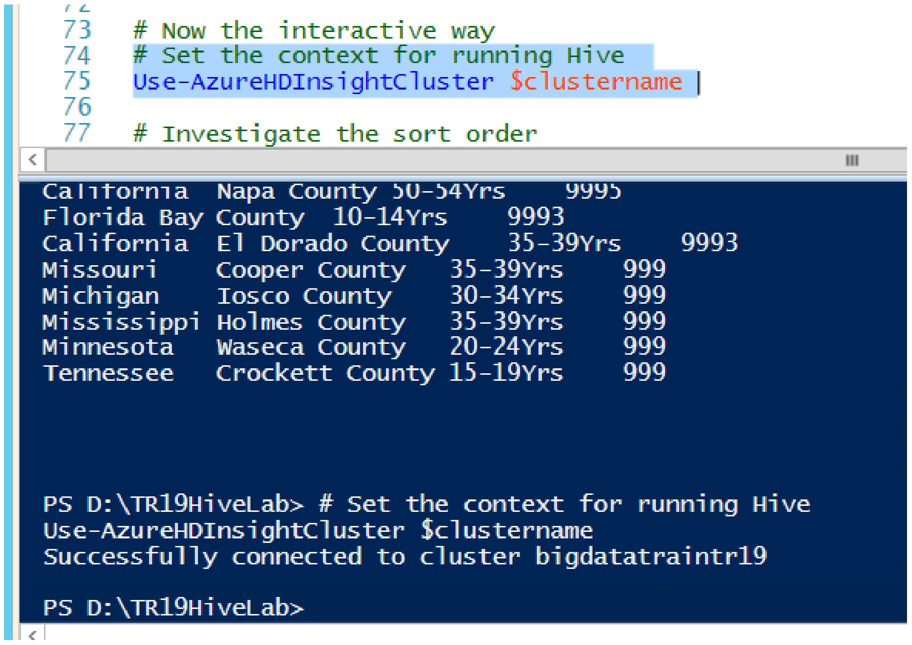
Running Hive statements as a job like this allows you to fire off a long running query and then view the results later by skipping the Wait-AzureHDInsightJob command.

Running Hive commands using Invoke-Hive

The Invoke-Hive cmdlet is a shortcut for defining a hive job, submitting the job and then waiting for the results.

Using Invoke-Hive

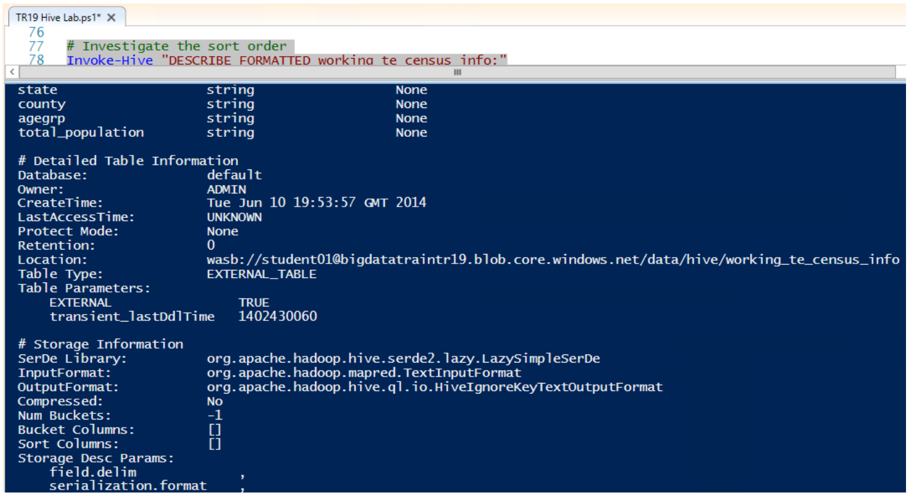
1. To use the Invoke-Hive, you need to first set the execution context to your HDInsight cluster if you have more than one cluster under your subscription. Select the following command and press **F8** to set the context.



```
# Now the interactive way
# Set the context for running Hive
Use-AzureHDInsightCluster $clustername

# Investigate the sort order
```

```
California  Napa County 50-54Yrs     9995
Florida Bay County  10-14Yrs     9993
California  El Dorado County     35-39Yrs     9993
Missouri    Cooper County    35-39Yrs     999
Michigan    Iosco County     30-34Yrs     999
Mississippi Holmes County    35-39Yrs     999
Minnesota   Waseca County    20-24Yrs     999
Tennessee   Crockett County 15-19Yrs     999



PS D:\TR19HiveLab> # Set the context for running Hive
Use-AzureHDInsightCluster $clustername
Successfully connected to cluster bigdatatraintr19

PS D:\TR19HiveLab>
```

2. To view the definition of a Hive table, you can use the DESCRIBE command. Select the following command and press **F8** to see the table definition for the *working_te_census_table*.



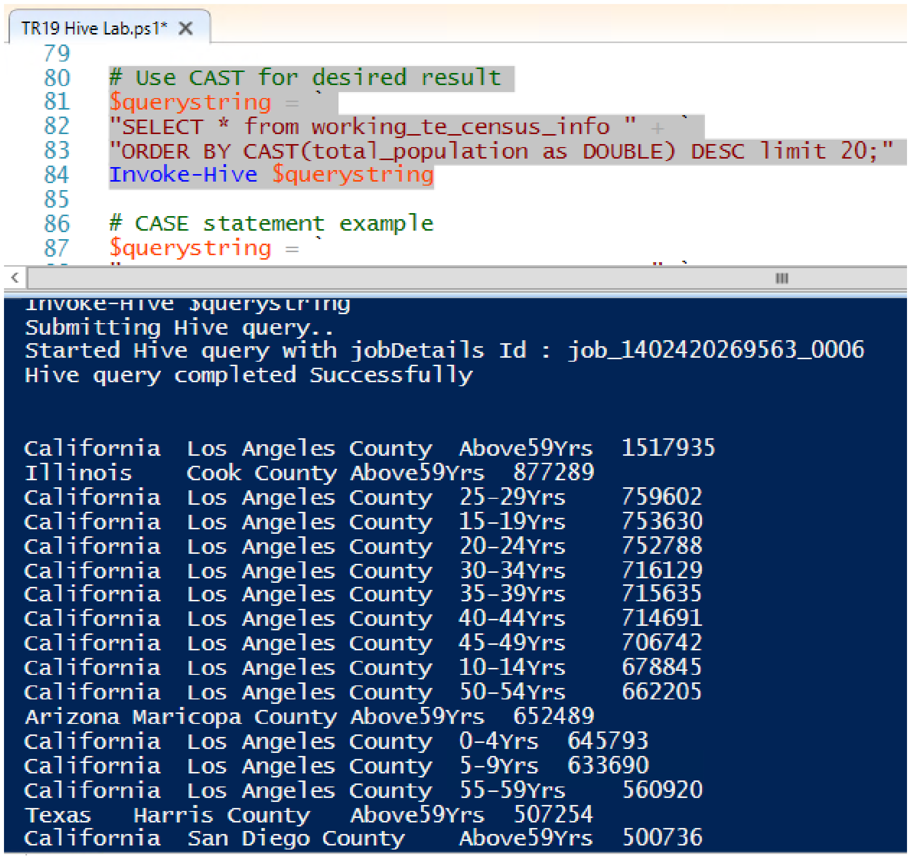
TR19 Hive Lab.ps1*

```

# Investigate the sort order
Invoke-Hive "DESCRIBE FORMATTED working_te_census_info;"
```

```
state                   string                  None
county                  string                  None
agegrp                  string                  None
total_population        string                  None

# Detailed Table Information
Database:               default
Owner:                  ADMIN
CreateTime:             Tue Jun 10 19:53:57 GMT 2014
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               wasb://student01@bigdatatraintr19.blob.core.windows.net/data/hive/working_te_census_info
Table Type:             EXTERNAL_TABLE
Table Parameters:
    EXTERNAL                TRUE
    transient_lastDdlTime   1402430060

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
    field.delim             ,
    serialization.format    ,
```

Notice the total_population is a string. This is why the sort order for the first query did not make sense initially. Scroll through the other table information to see how the table is stored.

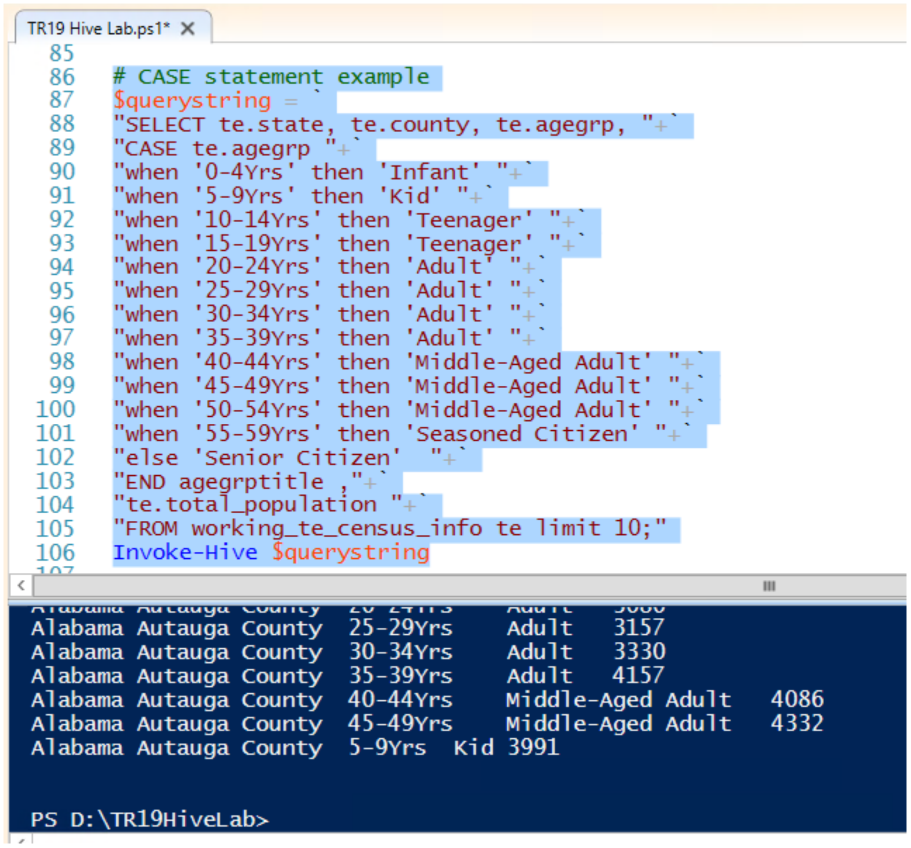
3. HiveQL syntax includes the CAST() function for converting data types. Select the following commands and press F8 to show the correct results.



```
# Use CAST for desired result
$querystring = `
"SELECT * from working_te_census_info " + `
"ORDER BY CAST(total_population as DOUBLE) DESC limit 20;"
Invoke-Hive $querystring

# CASE statement example
$querystring = `
```

```
Invoke-Hive $querystring
Submitting Hive query..
Started Hive query with jobDetails Id : job_1402420269563_0006
Hive query completed Successfully


California  Los Angeles County  Above59Yrs  1517935
Illinois    Cook County Above59Yrs  877289
California  Los Angeles County  25-29Yrs    759602
California  Los Angeles County  15-19Yrs    753630
California  Los Angeles County  20-24Yrs    752788
California  Los Angeles County  30-34Yrs    716129
California  Los Angeles County  35-39Yrs    715635
California  Los Angeles County  40-44Yrs    714691
California  Los Angeles County  45-49Yrs    706742
California  Los Angeles County  10-14Yrs    678845
California  Los Angeles County  50-54Yrs    662205
Arizona Maricopa County Above59Yrs  652489
California  Los Angeles County  0-4Yrs  645793
California  Los Angeles County  5-9Yrs  633690
California  Los Angeles County  55-59Yrs    560920
Texas   Harris County   Above59Yrs  507254
California  San Diego County    Above59Yrs  500736
```

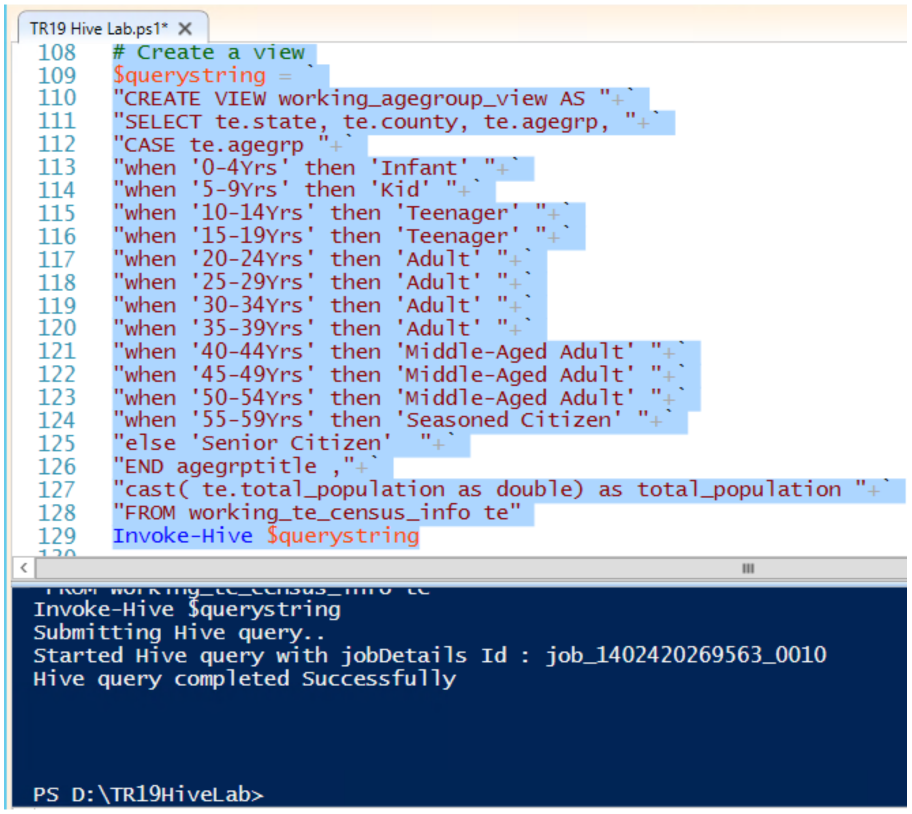
4. HiveQL syntax includes the CASE statement. Select the following commands and press F8 to see how the CASE statement can be used to provide additional grouping of age categories.



```
# CASE statement example
$querystring = `
"SELECT te.state, te.county, te.agegrp, "+`
"CASE te.agegrp "+`
"when '0-4Yrs' then 'Infant' "+`
"when '5-9Yrs' then 'Kid' "+`
"when '10-14Yrs' then 'Teenager' "+`
"when '15-19Yrs' then 'Teenager' "+`
"when '20-24Yrs' then 'Adult' "+`
"when '25-29Yrs' then 'Adult' "+`
"when '30-34Yrs' then 'Adult' "+`
"when '35-39Yrs' then 'Adult' "+`
"when '40-44Yrs' then 'Middle-Aged Adult' "+`
"when '45-49Yrs' then 'Middle-Aged Adult' "+`
"when '50-54Yrs' then 'Middle-Aged Adult' "+`
"when '55-59Yrs' then 'Seasoned Citizen' "+`
"else 'Senior Citizen'  "+`
"END agegrptitle ,"+`
"te.total_population "+`
"FROM working_te_census_info te limit 10;"
Invoke-Hive $querystring
```

```
Alabama Autauga County  25-29Yrs     Adult   3157
Alabama Autauga County  30-34Yrs     Adult   3330
Alabama Autauga County  35-39Yrs     Adult   4157
Alabama Autauga County  40-44Yrs     Middle-Aged Adult    4086
Alabama Autauga County  45-49Yrs     Middle-Aged Adult    4332
Alabama Autauga County  5-9Yrs  Kid 3991


PS D:\TR19HiveLab>
```

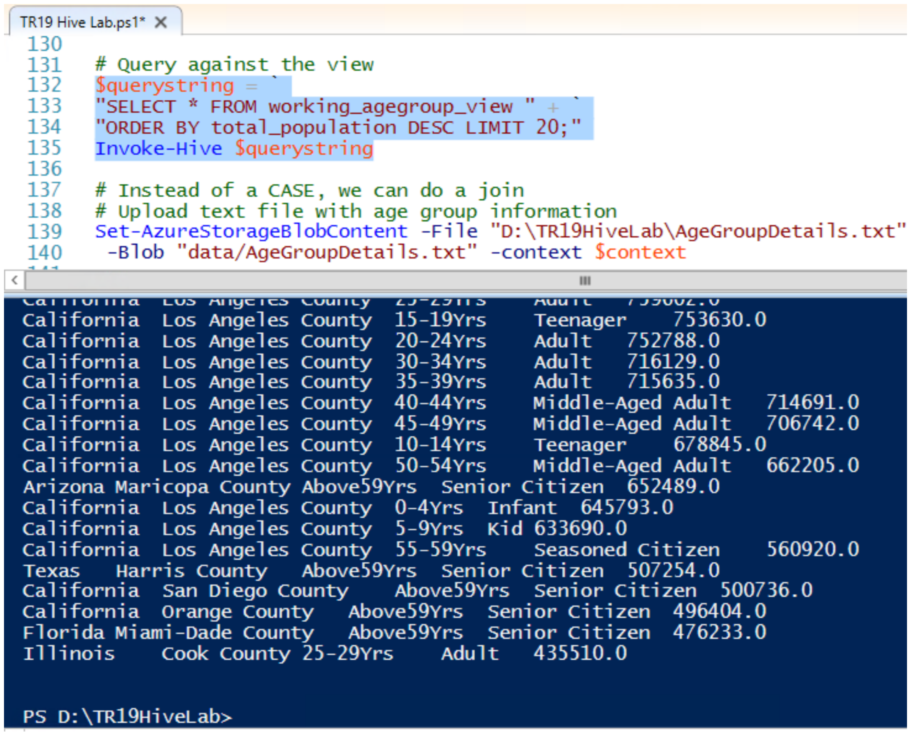
5. To abstract queries, HiveQL includes the CREATE VIEW statement. To create a view based on the prior query that also includes casting the total_population column as a number, select the following commands and press F8.



```
TR19 Hive Lab.ps1*
# Create a view
$querystring = `
"CREATE VIEW working_agegroup_view AS "+`
"SELECT te.state, te.county, te.agegrp, "+`
"CASE te.agegrp "+`
"when '0-4Yrs' then 'Infant' "+`
"when '5-9Yrs' then 'Kid' "+`
"when '10-14Yrs' then 'Teenager' "+`
"when '15-19Yrs' then 'Teenager' "+`
"when '20-24Yrs' then 'Adult' "+`
"when '25-29Yrs' then 'Adult' "+`
"when '30-34Yrs' then 'Adult' "+`
"when '35-39Yrs' then 'Adult' "+`
"when '40-44Yrs' then 'Middle-Aged Adult' "+`
"when '45-49Yrs' then 'Middle-Aged Adult' "+`
"when '50-54Yrs' then 'Middle-Aged Adult' "+`
"when '55-59Yrs' then 'Seasoned Citizen' "+`
"else 'Senior Citizen'  "+`
"END agegrptitle ,"+`
"cast( te.total_population as double) as total_population "+`
"FROM working_te_census_info te"
Invoke-Hive $querystring
```

```
Invoke-Hive $querystring
Submitting Hive query..
Started Hive query with jobDetails Id : job_1402420269563_0010
Hive query completed Successfully



PS D:\TR19HiveLab>
```

6. Creating a view does not improve performance, but it does make repeated queries easier. To run a view, select the following commands and press F8.

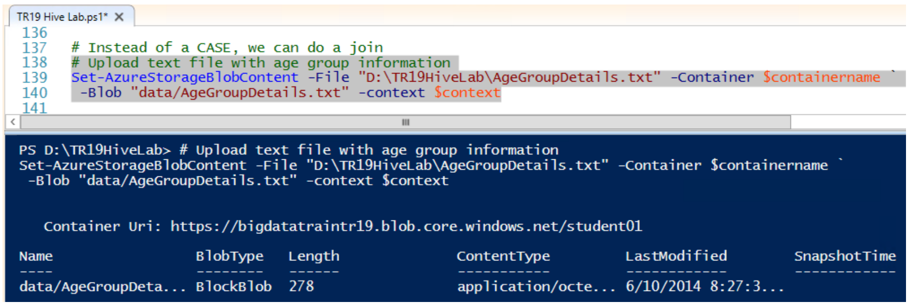


```
TR19 Hive Lab.ps1*

# Query against the view
$querystring = `
"SELECT * FROM working_agegroup_view " + `
"ORDER BY total_population DESC LIMIT 20;"
Invoke-Hive $querystring

# Instead of a CASE, we can do a join
# Upload text file with age group information
Set-AzureStorageBlobContent -File "D:\TR19HiveLab\AgeGroupDetails.txt"
 -Blob "data/AgeGroupDetails.txt" -context $context
```

```
California  Los Angeles County  15-19Yrs    Teenager    753630.0
California  Los Angeles County  20-24Yrs    Adult   752788.0
California  Los Angeles County  30-34Yrs    Adult   716129.0
California  Los Angeles County  35-39Yrs    Adult   715635.0
California  Los Angeles County  40-44Yrs    Middle-Aged Adult   714691.0
California  Los Angeles County  45-49Yrs    Middle-Aged Adult   706742.0
California  Los Angeles County  10-14Yrs    Teenager    678845.0
California  Los Angeles County  50-54Yrs    Middle-Aged Adult   662205.0
Arizona Maricopa County Above59Yrs  Senior Citizen  652489.0
California  Los Angeles County  0-4Yrs  Infant  645793.0
California  Los Angeles County  5-9Yrs  Kid 633690.0
California  Los Angeles County  55-59Yrs    Seasoned Citizen    560920.0
Texas   Harris County   Above59Yrs  Senior Citizen  507254.0
California  San Diego County    Above59Yrs  Senior Citizen  500736.0
California  Orange County   Above59Yrs  Senior Citizen  496404.0
Florida Miami-Dade County   Above59Yrs  Senior Citizen  476233.0
Illinois    Cook County 25-29Yrs    Adult   435510.0

PS D:\TR19HiveLab>
```

Performing a join query

1. From a database design perspective, rather than using a CASE statement, it might be better to create a lookup table and then use the HiveQL join syntax. To upload the data for the age group lookup, select the following command and press **F8**.

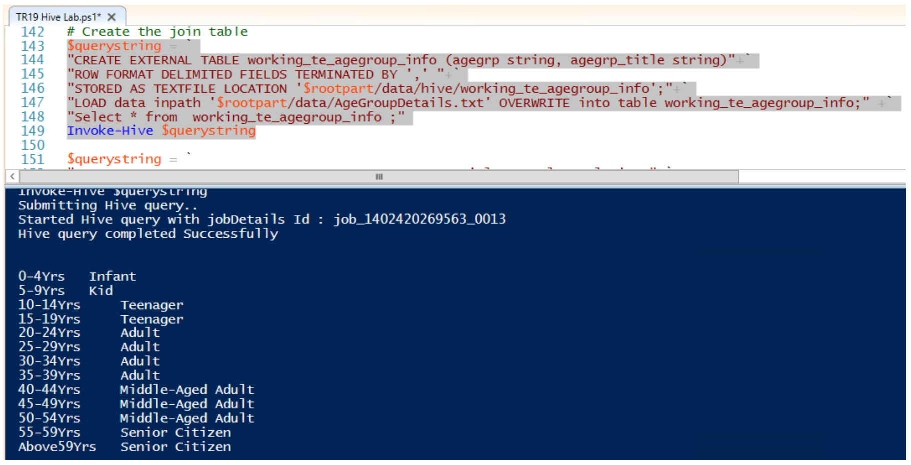


TR19 Hive Lab.ps1*

```

# Instead of a CASE, we can do a join
# Upload text file with age group information
Set-AzureStorageBlobContent -File "D:\TR19HiveLab\AgeGroupDetails.txt" -Container $containername `
 -Blob "data/AgeGroupDetails.txt" -context $context

```

```
PS D:\TR19HiveLab> # Upload text file with age group information
Set-AzureStorageBlobContent -File "D:\TR19HiveLab\AgeGroupDetails.txt" -Container $containername `
 -Blob "data/AgeGroupDetails.txt" -context $context


   Container Uri: https://bigdatatraintr19.blob.core.windows.net/student01

Name                BlobType   Length            ContentType         LastModified        SnapshotTime
----                --------   ------            -----------         ------------        ------------
data/AgeGroupDeta... BlockBlob  278               application/octe... 6/10/2014 8:27:3...
```

2. To create the lookup table and see the results in Hive, select the following commands and press F8.

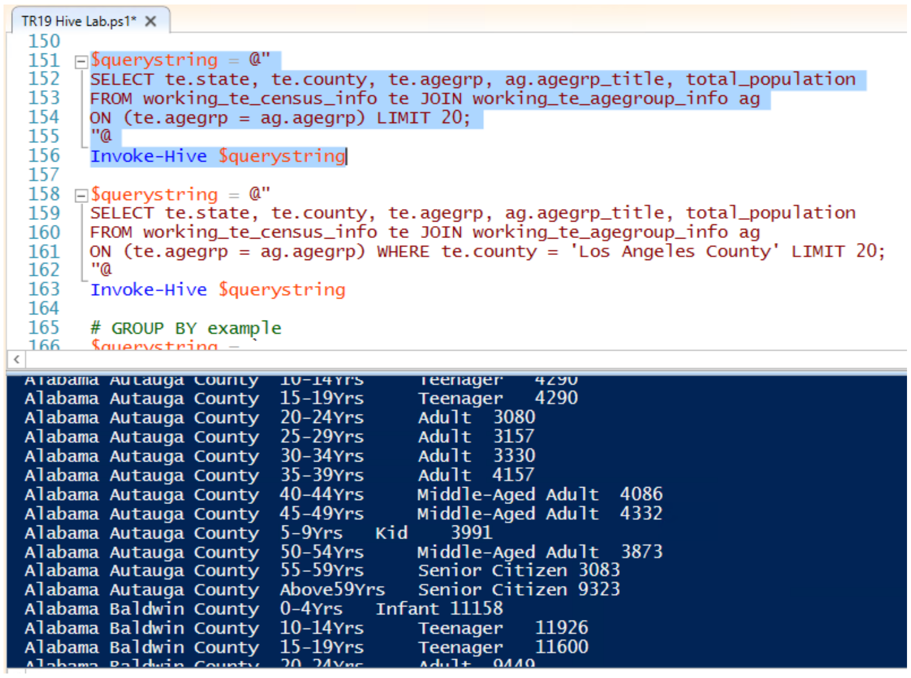


TR19 Hive Lab.ps1*

```
# Create the join table
$querystring = `
"CREATE EXTERNAL TABLE working_te_agegroup_info (agegrp string, agegrp_title string)"+`
"ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' "+`
"STORED AS TEXTFILE LOCATION '$rootpart/data/hive/working_te_agegroup_info';"+`
"LOAD data inpath '$rootpart/data/AgeGroupDetails.txt' OVERWRITE into table working_te_agegroup_info;" +`
"Select * from  working_te_agegroup_info ;"
Invoke-Hive $querystring

$querystring = `
```

```
Invoke-Hive $querystring
Submitting Hive query..
Started Hive query with jobDetails Id : job_1402420269563_0013
Hive query completed Successfully


0-4Yrs   Infant
5-9Yrs   Kid
10-14Yrs     Teenager
15-19Yrs     Teenager
20-24Yrs     Adult
25-29Yrs     Adult
30-34Yrs     Adult
35-39Yrs     Adult
40-44Yrs     Middle-Aged Adult
45-49Yrs     Middle-Aged Adult
50-54Yrs     Middle-Aged Adult
55-59Yrs     Senior Citizen
Above59Yrs   Senior Citizen
```

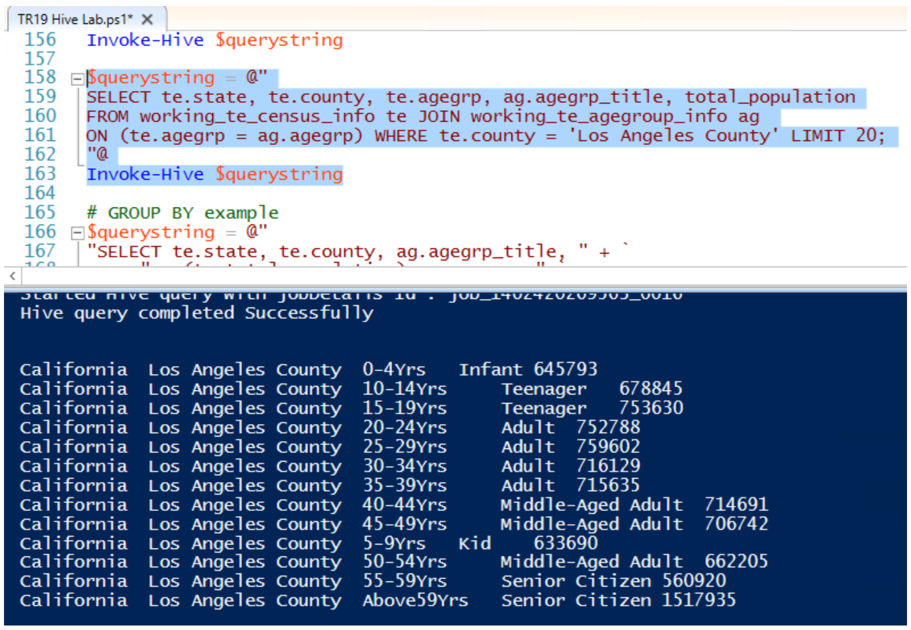
3. To run a select statement with a join, select the following commands and press F8. Notice the different PowerShell string syntax you can use to avoid string concatenation errors.



```
$querystring = @"
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) LIMIT 20;
"@
Invoke-Hive $querystring

$querystring = @"
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) WHERE te.county = 'Los Angeles County' LIMIT 20;
"@
Invoke-Hive $querystring

# GROUP BY example
```

```
Alabama Autauga County 15-19Yrs     Teenager   4290
Alabama Autauga County 20-24Yrs     Adult  3080
Alabama Autauga County 25-29Yrs     Adult  3157
Alabama Autauga County 30-34Yrs     Adult  3330
Alabama Autauga County 35-39Yrs     Adult  4157
Alabama Autauga County 40-44Yrs     Middle-Aged Adult  4086
Alabama Autauga County 45-49Yrs     Middle-Aged Adult  4332
Alabama Autauga County 5-9Yrs   Kid    3991
Alabama Autauga County 50-54Yrs     Middle-Aged Adult  3873
Alabama Autauga County 55-59Yrs     Senior Citizen 3083
Alabama Autauga County Above59Yrs   Senior Citizen 9323
Alabama Baldwin County 0-4Yrs   Infant 11158
Alabama Baldwin County 10-14Yrs     Teenager   11926
Alabama Baldwin County 15-19Yrs     Teenager   11600
```

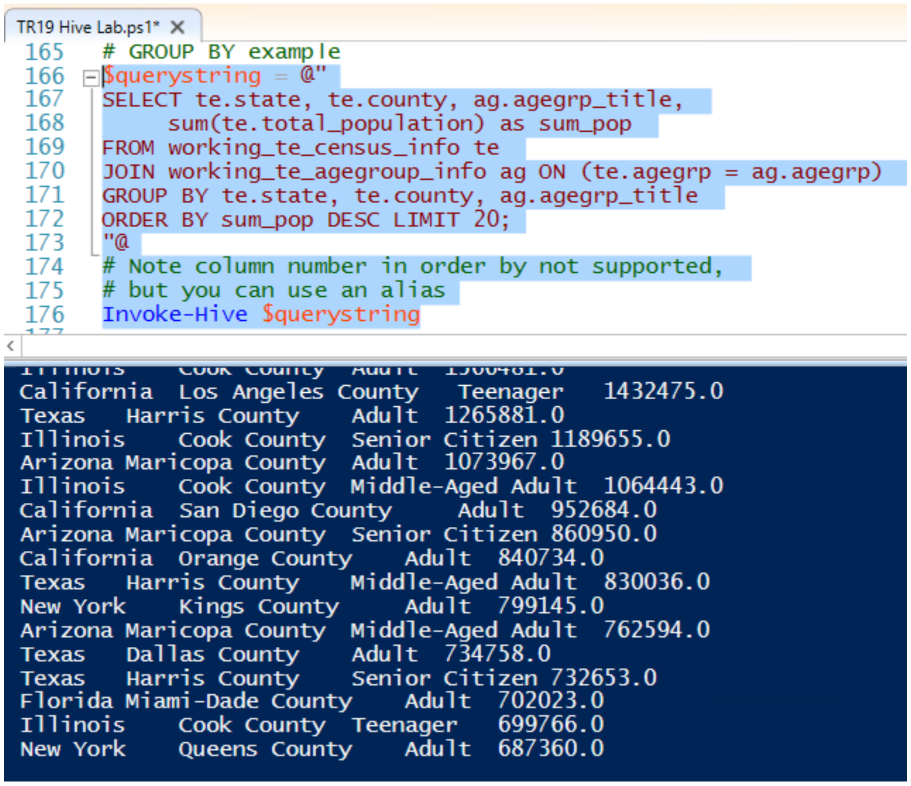
4. To run a query with a combination where and join clause, select the following commands and press **F8**.



```
Invoke-Hive $querystring

$querystring = @"
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) WHERE te.county = 'Los Angeles County' LIMIT 20;
"@
Invoke-Hive $querystring

# GROUP BY example
$querystring = @"
"SELECT te.state, te.county, ag.agegrp_title, " + `
```

```
Hive query completed Successfully


California  Los Angeles County  0-4Yrs    Infant 645793
California  Los Angeles County  10-14Yrs      Teenager   678845
California  Los Angeles County  15-19Yrs      Teenager   753630
California  Los Angeles County  20-24Yrs      Adult  752788
California  Los Angeles County  25-29Yrs      Adult  759602
California  Los Angeles County  30-34Yrs      Adult  716129
California  Los Angeles County  35-39Yrs      Adult  715635
California  Los Angeles County  40-44Yrs      Middle-Aged Adult  714691
California  Los Angeles County  45-49Yrs      Middle-Aged Adult  706742
California  Los Angeles County  5-9Yrs    Kid    633690
California  Los Angeles County  50-54Yrs      Middle-Aged Adult  662205
California  Los Angeles County  55-59Yrs      Senior Citizen 560920
California  Los Angeles County  Above59Yrs    Senior Citizen 1517935
```

5. The GROUP BY clause is similar to Transact-SQL as well. There is a slight change in how ORDER BY works. To run a group by query with a join, select the following commands and press **F8**.


TR19 Hive Lab.ps1*

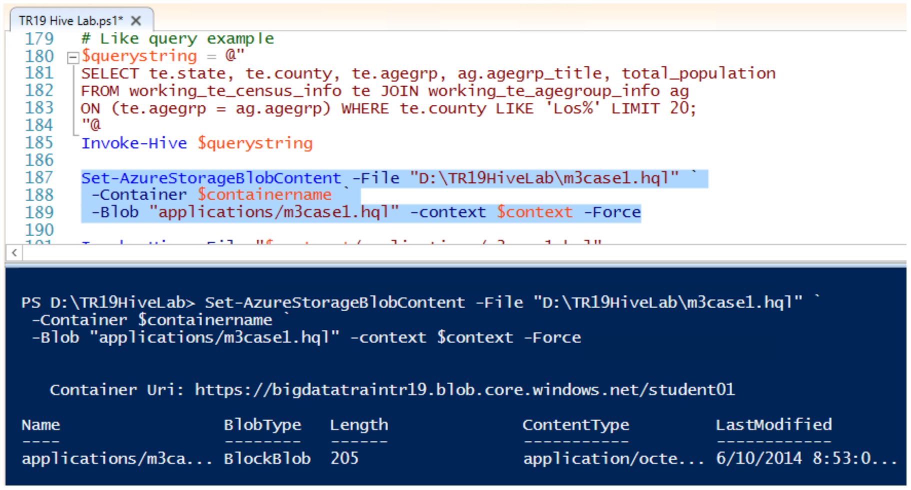
```
# GROUP BY example
$querystring = @"
SELECT te.state, te.county, ag.agegrp_title,
     sum(te.total_population) as sum_pop
FROM working_te_census_info te
JOIN working_te_agegroup_info ag ON (te.agegrp = ag.agegrp)
GROUP BY te.state, te.county, ag.agegrp_title
ORDER BY sum_pop DESC LIMIT 20;
"@
# Note column number in order by not supported,
# but you can use an alias
Invoke-Hive $querystring
```

```
California  Los Angeles County   Teenager   1432475.0
Texas   Harris County    Adult  1265881.0
Illinois    Cook County  Senior Citizen 1189655.0
Arizona Maricopa County  Adult  1073967.0
Illinois    Cook County  Middle-Aged Adult  1064443.0
California  San Diego County     Adult  952684.0
Arizona Maricopa County  Senior Citizen 860950.0
California  Orange County    Adult  840734.0
Texas   Harris County    Middle-Aged Adult  830036.0
New York    Kings County     Adult  799145.0
Arizona Maricopa County  Middle-Aged Adult  762594.0
Texas   Dallas County    Adult  734758.0
Texas   Harris County    Senior Citizen 732653.0
Florida Miami-Dade County    Adult  702023.0
Illinois    Cook County  Teenager   699766.0
New York    Queens County    Adult  687360.0
```

Running a script of HiveQL statements

HDInsight allows you to run Hive commands that are in a text file which is stored in a storage container. In this exercise, you will learn how to run HiveQL statements using the Invoke-Hive command. For larger jobs, you will want to use the New-AzureHDInsightHiveJobDefinition cmdlet with the –File parameter to schedule the job and check back later with the results.

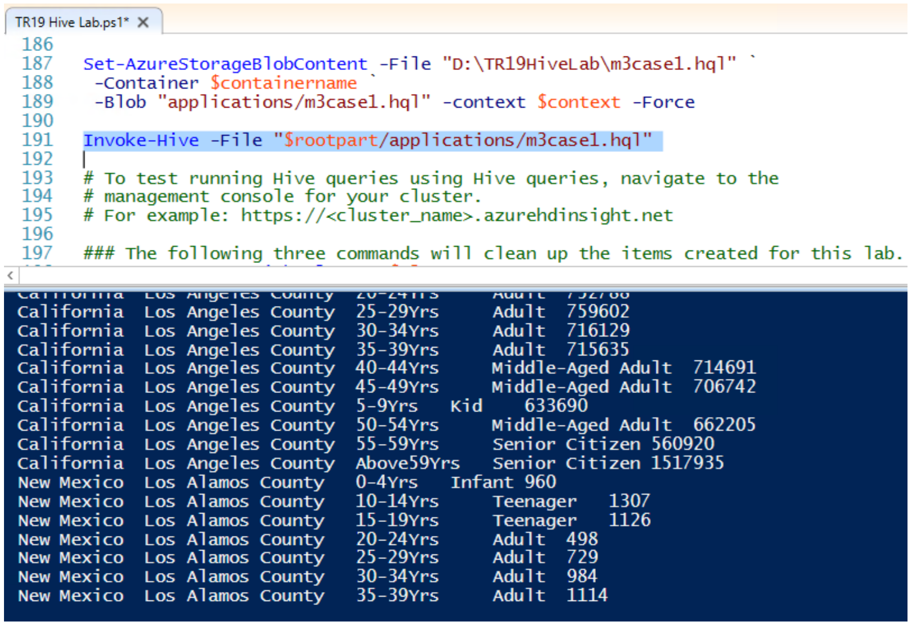
1. There are some cases where you must run your HiveQL statements using a file. The D:\TR19HiveLab\m3case1.hql file contains the query shown for the Like query example. To upload the file to Azure storage so that you can execute it, select the following command and press **F8**.



```
TR19 Hive Lab.ps1*
# Like query example
$querystring = @"
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) WHERE te.county LIKE 'Los%' LIMIT 20;
"@
Invoke-Hive $querystring

Set-AzureStorageBlobContent -File "D:\TR19HiveLab\m3case1.hql" `
 -Container $containername `
 -Blob "applications/m3case1.hql" -context $context -Force
```

```
PS D:\TR19HiveLab> Set-AzureStorageBlobContent -File "D:\TR19HiveLab\m3case1.hql" `
 -Container $containername `
 -Blob "applications/m3case1.hql" -context $context -Force


   Container Uri: https://bigdatatraintr19.blob.core.windows.net/student01

Name                 BlobType  Length          ContentType        LastModified
----                 --------  ------          -----------        ------------
applications/m3ca... BlockBlob 205             application/octe... 6/10/2014 8:53:0...
```

2. To run the file with the HiveQL statement, select the following command and press **F8**.



```
TR19 Hive Lab.ps1*

Set-AzureStorageBlobContent -File "D:\TR19HiveLab\m3case1.hql" `
 -Container $containername `
 -Blob "applications/m3case1.hql" -context $context -Force

Invoke-Hive -File "$rootpart/applications/m3case1.hql"

# To test running Hive queries using Hive queries, navigate to the
# management console for your cluster.
# For example: https://<cluster_name>.azurehdinsight.net

### The following three commands will clean up the items created for this lab.
```

```
California  Los Angeles County  25-29Yrs     Adult  759602
California  Los Angeles County  30-34Yrs     Adult  716129
California  Los Angeles County  35-39Yrs     Adult  715635
California  Los Angeles County  40-44Yrs     Middle-Aged Adult  714691
California  Los Angeles County  45-49Yrs     Middle-Aged Adult  706742
California  Los Angeles County  5-9Yrs   Kid     633690
California  Los Angeles County  50-54Yrs     Middle-Aged Adult  662205
California  Los Angeles County  55-59Yrs     Senior Citizen 560920
California  Los Angeles County  Above59Yrs   Senior Citizen 1517935
New Mexico  Los Alamos County  0-4Yrs    Infant 960
New Mexico  Los Alamos County  10-14Yrs     Teenager    1307
New Mexico  Los Alamos County  15-19Yrs     Teenager    1126
New Mexico  Los Alamos County  20-24Yrs     Adult  498
New Mexico  Los Alamos County  25-29Yrs     Adult  729
New Mexico  Los Alamos County  30-34Yrs     Adult  984
New Mexico  Los Alamos County  35-39Yrs     Adult  1114
```

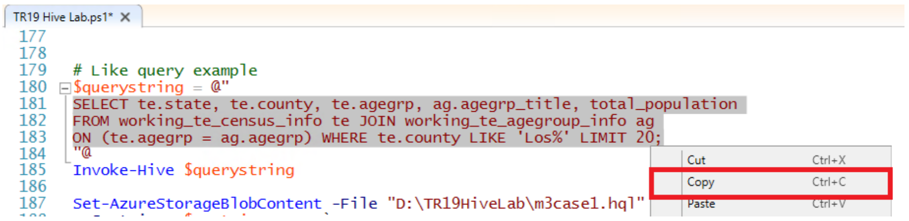
Using the Hive console with HDInsight

Starting in June 2014, the HDInsight team brought back the ability of running Hive queries through the Manage Cluster web interface. The console is a useful way of testing out Hive queries before automating execution of them using PowerShell. In this exercise, you will run a query using the console.

1. Navigate to https://<your_cluster>.azurehdinsight.net.

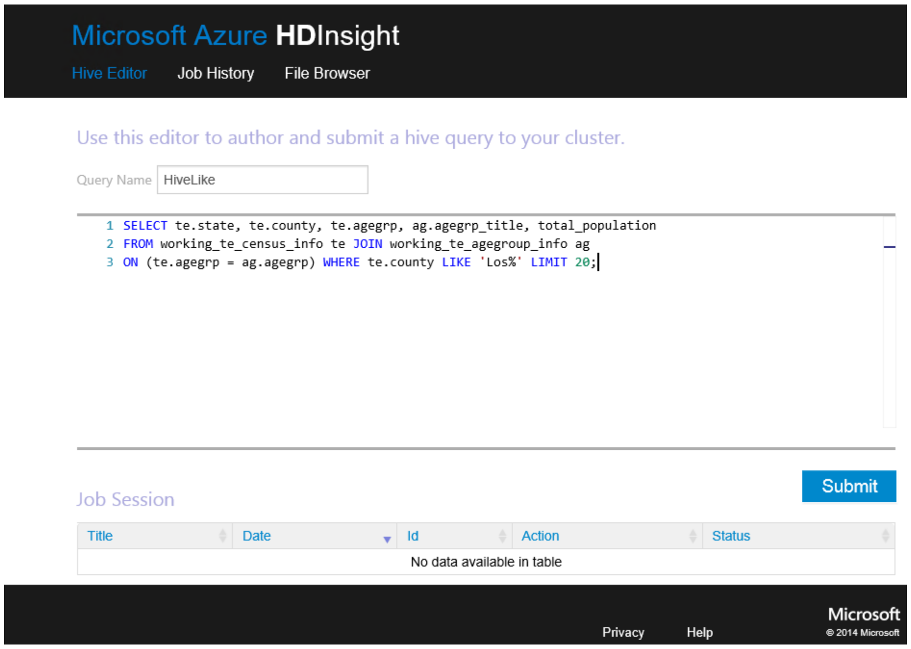
2. Enter in your user name Admin and password Pass@word12 in the credentials dialog.

3. Select the text for the Like query in the TR19 Hive Lab.ps1 shown below and copy the text to the clipboard.



```
# Like query example
$querystring = @"
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) WHERE te.county LIKE 'Los%' LIMIT 20;
"@
Invoke-Hive $querystring

Set-AzureStorageBlobContent -File "D:\TR19HiveLab\m3case1.hql"
```

4. Go back to the browser and enter in a **Query Name** such as **HiveLike** and then paste the HiveQL statement into the editor region.



Use this editor to author and submit a hive query to your cluster.

Query Name: HiveLike

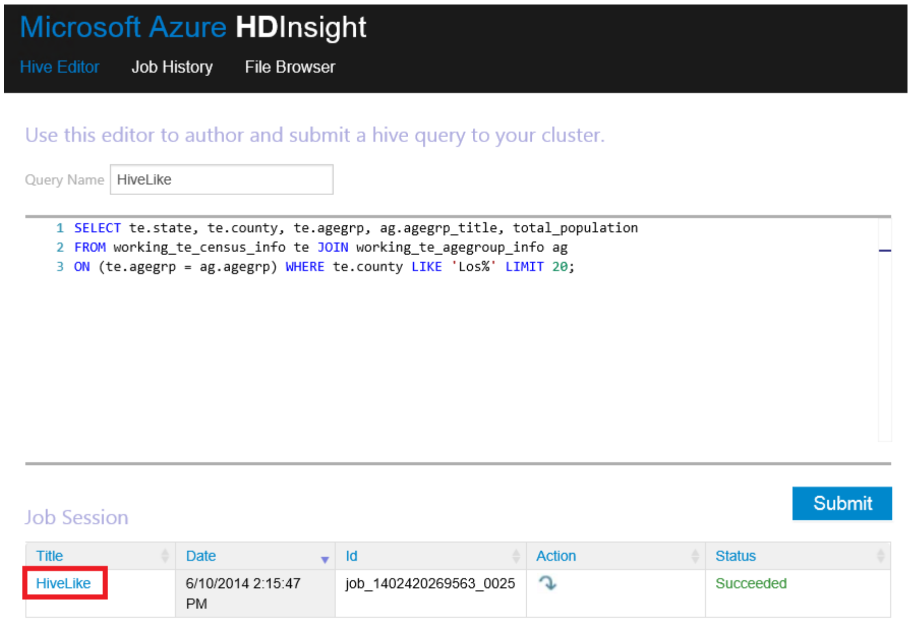
```
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) WHERE te.county LIKE 'Los%' LIMIT 20;
```

Job Session

Title	Date	Id	Action	Status
No data available in table				

5. Click **Submit** to run the query. HDInsight displays the progress of the query in the **Status** column.

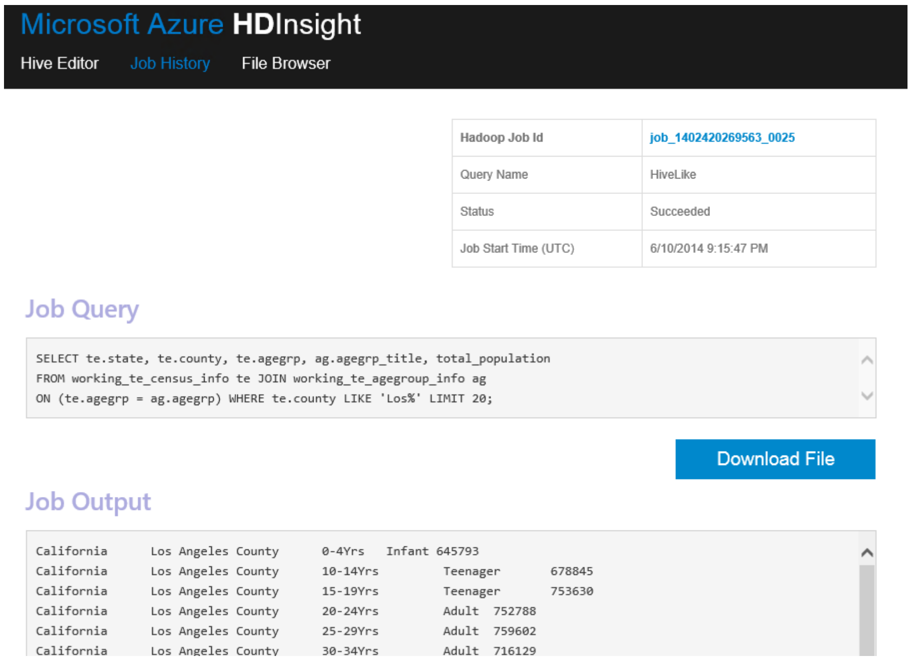
6. Once the **Status** value shows **Succeeded**, click on the Title link for the query to view the results.


Microsoft Azure HDInsight
Hive Editor
Job History
File Browser
Use this editor to author and submit a hive query to your cluster.
Query Name
HiveLike
Submit
Job Session

```
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) WHERE te.county LIKE 'Los%' LIMIT 20;
```

Title	Date	Id	Action	Status
HiveLike	6/10/2014 2:15:47 PM	job_1402420269563_0025		Succeeded

7. HDInsight displays a status page as shown below that lists the Job Query, Job Output and Job Log.



Hadoop Job Id	job_1402420269563_0025
Query Name	HiveLike
Status	Succeeded
Job Start Time (UTC)	6/10/2014 9:15:47 PM

```
SELECT te.state, te.county, te.agegrp, ag.agegrp_title, total_population
FROM working_te_census_info te JOIN working_te_agegroup_info ag
ON (te.agegrp = ag.agegrp) WHERE te.county LIKE 'Los%' LIMIT 20;
```

```
California      Los Angeles County      0-4Yrs   Infant 645793
California      Los Angeles County      10-14Yrs        Teenager        678845
California      Los Angeles County      15-19Yrs        Teenager        753630
California      Los Angeles County      20-24Yrs        Adult  752788
California      Los Angeles County      25-29Yrs        Adult  759602
California      Los Angeles County      30-34Yrs        Adult  716129
```

Summary

In this set of exercises, you learned the different ways of running HiveQL statements and the similarities of HiveQL to Transact-SQL. For a complete HiveQL language reference, go to https://cwiki.apache.org/confluence/display/Hive/LanguageManual.

Terms of Use

DISCLAIMER

This lab contains only a portion of new features and enhancements in Microsoft SQL Server 2014. The exercises are built on the General Availability release as of October 25, 2013 and updated with the April 2014 release that includes HDInsight version 3.0. Some of the features might change in future releases of the product. In this lab, you will learn about some, but not all, new features.