Setting Up a Relational Database in the Cloud

Contents

datasciencedojo
unleash the data scientist in you

Exercise 1: Provisioning an Azure SQL Database

In the SaaS era of cloud computing, databases are now commoditized. Cloud-based databases are manage free and elastically scaled for maximum efficiency. This frees up DBA staff and allows companies to refocus on the core of the business.

1. Visit the Azure Management portal:
   a. https://manage.windowsazure.com
2. Provisioning an Azure SQL DB
   a. Within the Azure Management portal, select **new**.
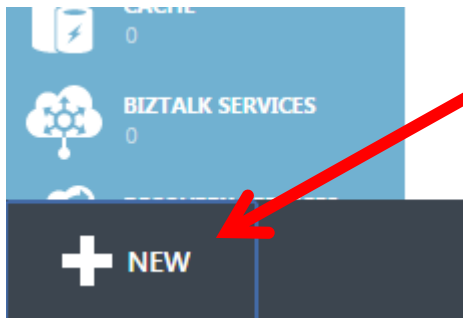
   b. **New > Data Services > SQL Database > Custom Create**
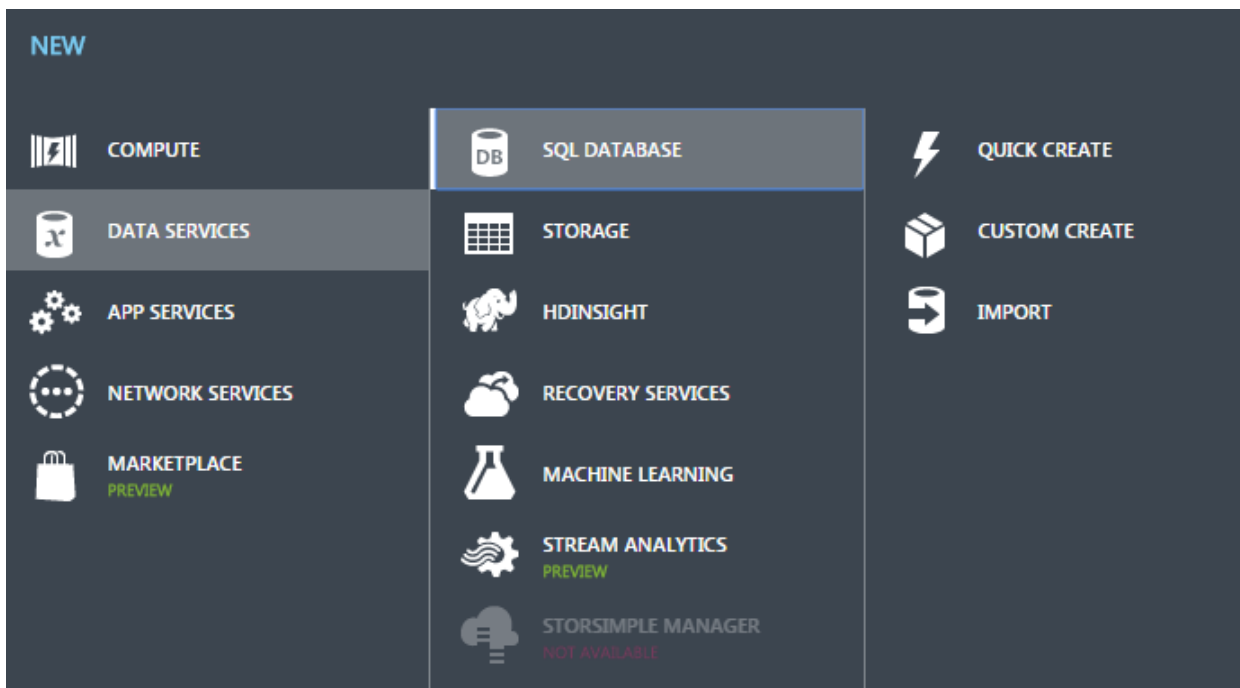
3. **Set Database Specifications**

NEW SQL DATABASE - CUSTOM CREATE

Specify database settings

NAME

SUBSCRIPTION

Free Trial (5576a6d9-c6fe-45a3-a834-9fc58b7c ▾

SERVICE TIERS   RETIRED TIERS

BASIC **STANDARD** PREMIUM   WEB   BUSINESS

PERFORMANCE LEVEL

S0 (10 DTUs)   ▾

MAX SIZE

250 GB

COLLATION

SQL_Latin1_General_CP1_CI_AS   ▾

SERVER

Choose a server   ▾

a. **Name**: Name the database. A database is a collection of tables. A server may have multiple databases on it.
b. **Subscription**: The subscription that the database will be charged to.
c. **Server Tiers**: The quality of the server. Choose **basic tier** for this example. Visit the following link for a comparison chart of the 3 tiers. Do not choose web or business tiers; they will be phased out in the near future. http://azure.microsoft.com/en-us/pricing/details/sql-database/
d. **Performance Level:** Select "**Basic (5 DTUs)**" for this lab. Specify the workload level of the database. Visit the link in server tiers for a full comparison chart of each performance level.
e. **Collation**: Leave this on default (**SQL_Latin1_General_CP1_CI_AS**). This is the character encoding within the table data fields. For additional information, visit: https://msdn.microsoft.com/en-us/library/azure/ee336245.aspx#sscs
f. **Server**: Choose **new server**. This server will host all your databases. A server name will be automatically assigned.

datasciencedojo
unleash the data scientist in you

4. **Server Credentials**

CREATE SERVER

SQL database server settings

LOGIN NAME

LOGIN PASSWORD

CONFIRM PASSWORD

REGION

South Central US ▼

☑ ALLOW WINDOWS AZURE SERVICES TO ACCESS THE SERVER. ?

☑ ENABLE LATEST SQL DATABASE UPDATE (V12) ?

a. **Login Name**: The login name of the server. All the databases in the server will require this login credential.
b. **Password**: Password of the server.
c. **Region**: For the purposes of this lab, choose **West Europe**. This is the region in which the database will be hosted. Be sure to keep the database server in the same region as your servers, clients, or other data pipeline artifacts.
d. **Allow Windows Azure Services to Access the Server**: Check this box. This will allow other Azure Services such as Stream Analytics to link with the database easily.
e. **Enable Latest SQL Database Update (v12):** Checked.

5. Confirm the existence of the server in the Azure Management Portal. The database will take ~2 minutes to provision.

ALL ITEMS

WEB APPS
0

VIRTUAL MACHINES
0

MOBILE SERVICES
0

CLOUD SERVICES
0

SQL DATABASES
3

STORAGE
3
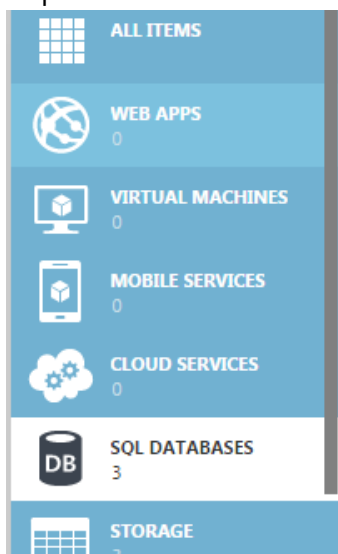
sql databases

DATABASES   SERVERS   DELETED DATABASES

NAME	STATUS	REPLICATION
dojodb	✔ Online	None
TollDataDB2	✔ Online	None
tempdb	˙ ˙ Creating	None

datasciencedojo
unleash the data scientist in you

# Exercise 2: Configuring DB Firewall and Access Rights

1. Add your current computer IP address to the firewall rules.
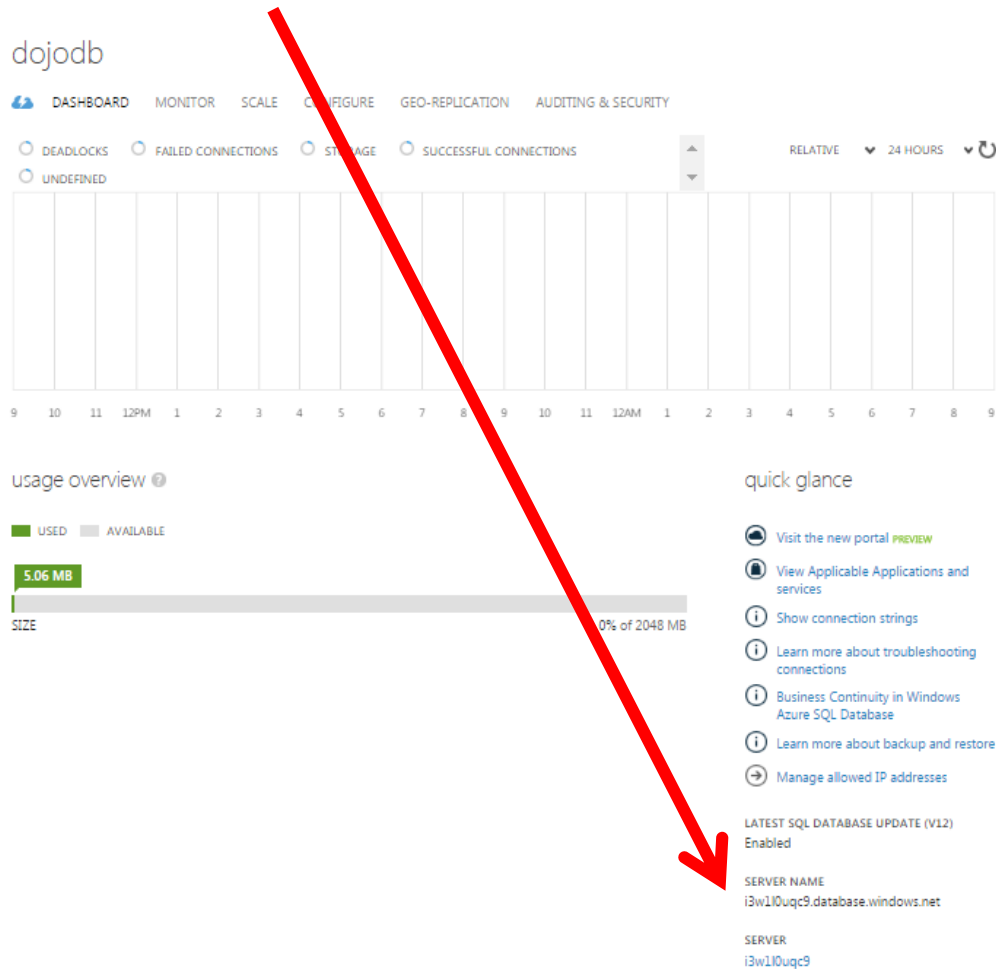   a. Azure Manangement Portal > SQL Databases > YourDB
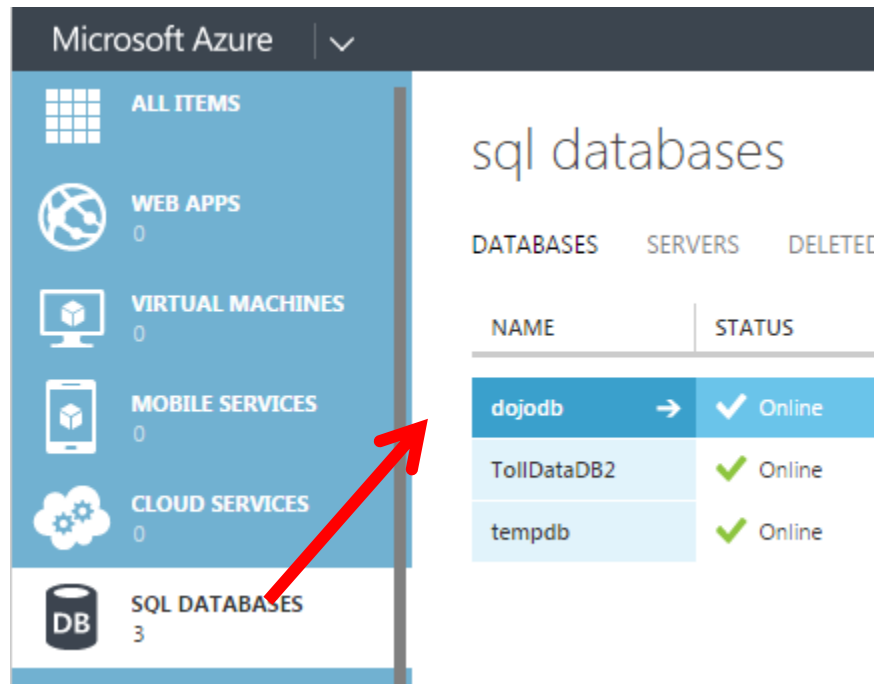
   b. Select "Set up Windows Azure firewall rules for this IP address" to automatically add the IP address of the current device to the database's firewall rules.
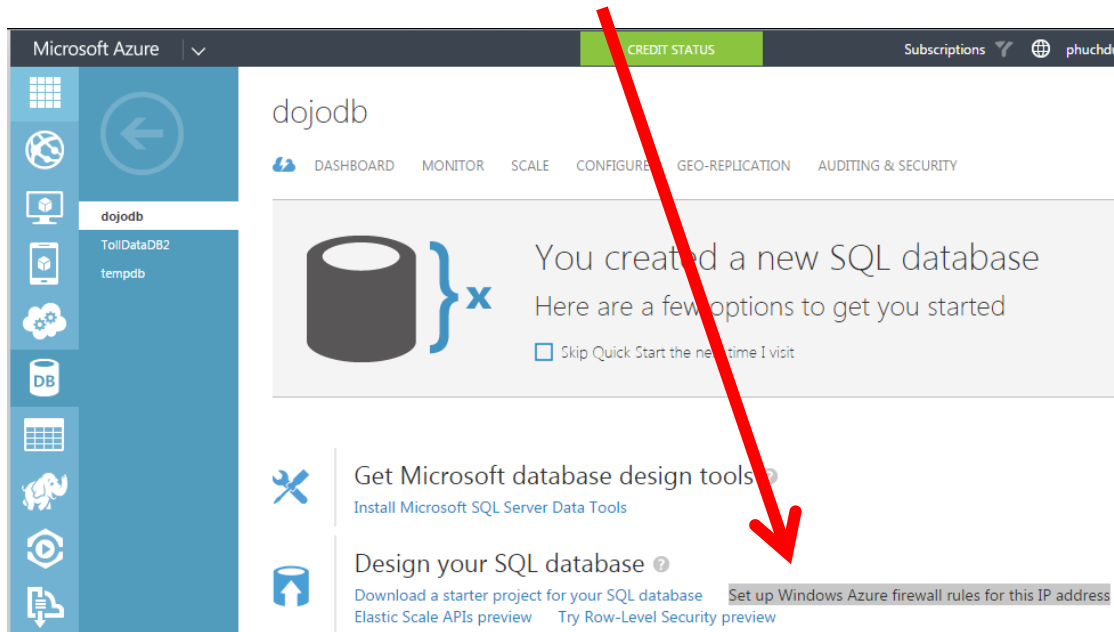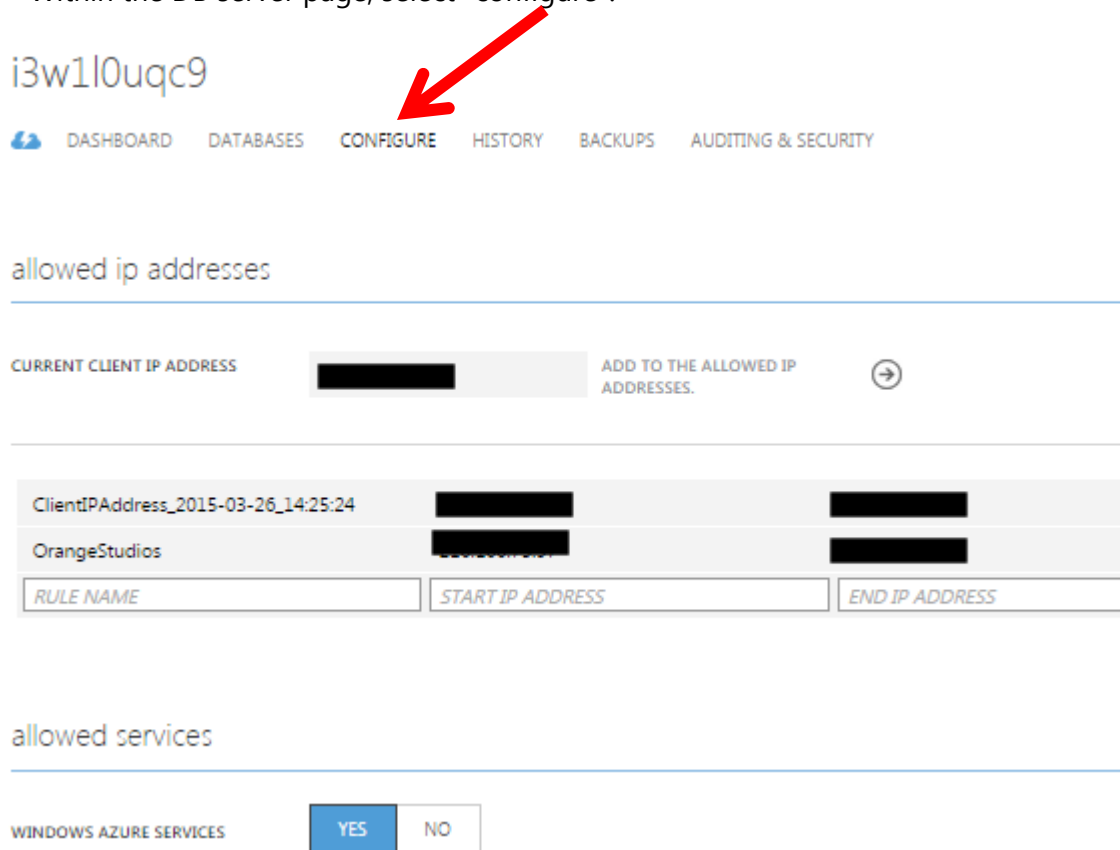
2. View firewall rules:
   a. Select "Dashboard" from the DB view.
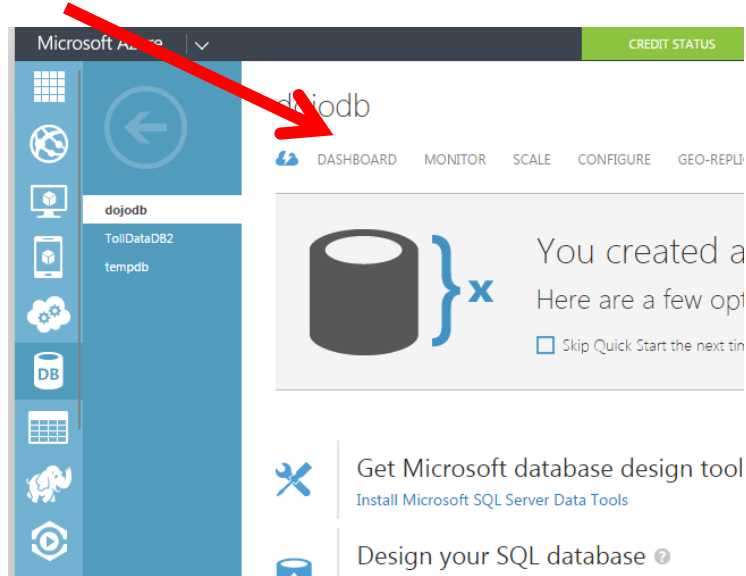
   b. Click on the DB's server name within the dashboard.

c. Within the DB server page, select "configure".

i3w1l0uqc9

DASHBOARD   DATABASES   CONFIGURE   HISTORY   BACKUPS   AUDITING & SECURITY

allowed ip addresses

| CURRENT CLIENT IP ADDRESS | ████████ | ADD TO THE ALLOWED IP ADDRESSES. | → |

ClientIPAddress_2015-03-26_14:25:24	████████	████████
OrangeStudios	████████	████████
RULE NAME	START IP ADDRESS	END IP ADDRESS

allowed services

| WINDOWS AZURE SERVICES | YES | NO |

d. This screen allows users to manually add IP addresses, in case the company or router has an external IP as well.

e. Be sure that the "Windows Azure Services" is enabled with "YES".

datasciencedojo
unleash the data scientist in you

Exercise 3: Loading Data Tables into the Database

There are many methods to connect to the database. Some include:

- Azure PowerShell
- Azure Machine Learning Studio
- MS Access
- MS Excel
- Programmatically (sqlcmd, ASP.net, ODBC, PHP, JDBC, ADO.NET)
- SQL Server Management Studio (SSMS)
- Visual Studio

This exercise will focus on connecting to an Azure SQL DB using Azure Machine Learning Studio.

1. Log into Azure ML and create a new experiment.
   a. https://studio.azureml.net
2. The datasets.
   a. A database is just a collection of tables which may reference each other in a "key" and index fashion.
   b. Drag in the following three datasets: **IMDB Movie Titles, Movie Ratings, Movie Tweets**
3. Writing to Azure SQL DB: The Azure ML writer module
   a. Drag in a writer module and connect it to "IMDB Movie Titles"
   b. The Writer Module will dynamically create schemas and specify the correct datatypes for each of the columns. No coding or SQL necessary.
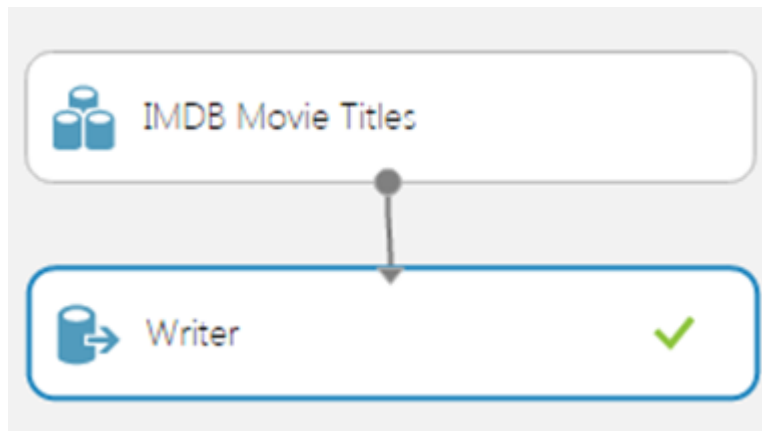
c. Writer Credentials:
   i. **Please specify data destination:** Azure SQL Database
   ii. **Database server name:** <YourServerName>.database.windows.net
      (1) Within your Azure Management Portal, select SQL Databases > YourDB
      (2) Find the "Server" address under "Connect to your Database"
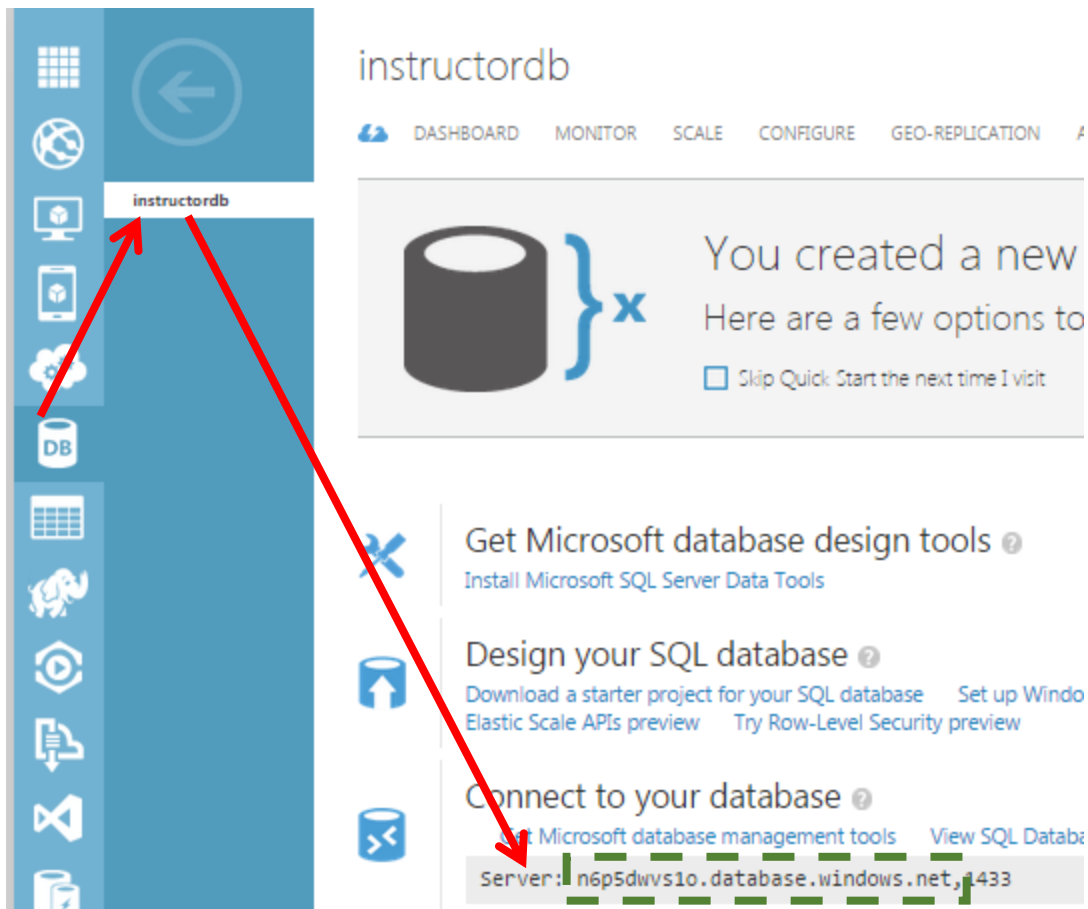
   iii. **Database name:** the name of your database. In the example above, the database is called instructordb.
   iv. **Server user account name:** the server's user name that was set by the user during provisioning.
   v. **Server user account password:** the server's login password that was set by the user during provisioning.
   vi. **Accept any server certificate (insecure):** unchecked
   vii. **Comma separated list of columns to be saved:** these are the columns to be saved. List the column names in order, separated by commas. In the example above, the dataset "IMDB Movie Titles" has the following columns to be saved: '**Movie ID, Movie Name**'
   viii. **Data table name:** this is what the table will be called. Call this table '**movies'**.
   ix. **Comma separated list of data table columns:** this is how the columns will be provisioned within the database table. This is a chance to rename the columns. In this example, fill in "**filmid, moviename"**. Spaces in column names will break the database. Also, its good style not to include capitalizations within column names as it will improve the efficiency when writing queries.
   x. **Number of rows written per SQL Azure operation:** 50

datasciencedojo
unleash the data scientist in you

4. Run the experiment to write to the database. This will take ~4 minutes to write 16,000 rows to the database.

datasciencedojo
unleash the data scientist in you

Exercise 4: Reading from Azure SQL Database

The reader module allows users to query the database.

1. Create a new experiment.
2. Drag in a reader module.
3. Reader module credentials:
   a. **Data source**: Azure SQL Database
   b. **Database server name**: <YourServerName>.database.windows.net
   c. **Database query**: the following query returns the first 10 rows of the table.

```
select top 10 *
from movies;
```

Final result:

4. Run the experiment, and visualize the output.

datasciencedojo
unleash the data scientist in you

Exercise 5: Forming a Relational Database

Right now, the database has a single table, which is just a database table. For this database to truly be a relational database, other tables that reference each other in relationships must also be present. In this exercise, we will write additional tables to the database, "movie ratings" and "movie tweets", which will take 7 and 10 minutes to write to the database respectively. To increase the write times, consider sampling the datasets first using the sampling module.

1. Write the dataset called '**Movie Ratings'** to the database. 227472 rows by 4 columns.
   a. Comma separated list of columns to be saved: '**UserId, MovieId, Rating, Timestamp**'
   b. Data Table Name: '**ratings'**
   c. Comma separated list of data table columns: '**userid, filmid, rating, timestamp**'
   d. Write time: ~7 minutes
2. Write the dataset called '**Movie Tweets**' to the database. 170285 rows by 8 columns.
   a. Comma separated list of columns to be saved: '**Scraping Time, Tweet ID, User ID, Movie ID, Rating, Retweet Count, Favorite Count, Time Zone'**
   b. Data Table Name: '**tweets'**
   c. Comma separated list of data table columns: '**scrapetime, tweetid, userid, filmid, rating, retweetcount, favoritecount, timezone'**
   d. Write time: ~10 minutes
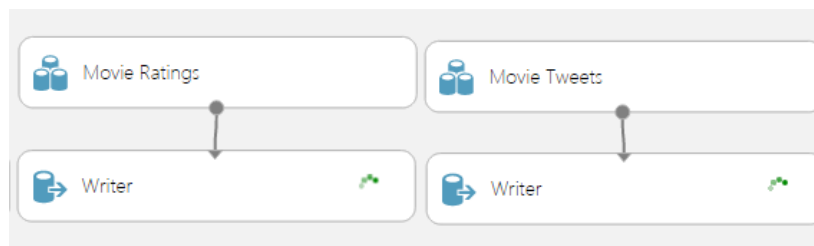
3. View the database
   a. Create a new experiment.
   b. Drag in a reader module.
      i. The following query will list all the tables inside the database. Be sure that the table 'movies', 'ratings', and 'tweets' are in the database.

```
SELECT * FROM
information_schema.tables
```

ii.   Run the experiment and visualize the output.

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
instructordb	sys	database_firewall_rules	VIEW
instructordb	dbo	iris	BASE TABLE
instructordb	dbo	city	BASE TABLE
instructordb	dbo	movies	BASE TABLE
instructordb	dbo	ratings	BASE TABLE
instructordb	dbo	tweets	BASE TABLE

c.   Drag in a second reader module.
i.   Probe the tweets table.

```sql
SELECT TOP 10 * FROM tweets;
```

id	scrapetime	tweetid	userid	filmid	rating	retweetcount	favoritecount	timezone
1	1391030896	307139025897152000	296041028	444778	8	0	0	Bangkok
2	1391030896	307139922186366000	239615275	1411238	6	0	0	Eastern Time (US & Canada)
3	1391030895	307142713369849000	36937267	1496422	7	0	0	Belgrade
4	1391030895	307143340607041000	45826125	118799	5	0	0	Budapest
5	1391030894	307144290214215000	45826125	338013	4	0	0	Budapest
6	1391030894	307144402382491000	27662592	1003052	8	0	0	Prague
7	1391030893	307144992542031000	45826125	64665	8	0	0	Budapest
8	1391030893	307145230707220000	206144706	884328	7	0	0	Riyadh
9	1391030892	307145590041624000	114473716	1559547	9	0	0	Taipei
10	1391030891	307149804595716000	124670134	949731	5	0	0	Unknown

datasciencedojo
unleash the data scientist in you

d. Drag in a third reader module.
   i. Probe the ratings table.

```sql
SELECT TOP 10 * FROM ratings;
```

id	userid	filmid	rating	timestamp
1	1	68646	10	1381620027
2	1	113277	10	1379466669
3	2	454876	8	1394818630
4	2	790636	7	1389963947
5	2	816711	8	1379963769
6	2	1091191	7	1391173869
7	2	1322269	7	1391529691
8	2	1433811	8	1380453043
9	2	1454468	8	1387016442
10	2	1535109	8	1386350135

Final output:

4. Perform a join on the 3 tables
   a. Drag in another reader module and try to join the tables.

b. Below is a sample code on joining two tables together.

```sql
select
    m.filmname as Film,
    t.userid as UserID,
    t.rating as UserRating,
    t.timezone as Region
from tweets as t
join movies as m
    on t.filmid = m.filmid
```

rows   columns
338404   4

view as

Film	UserID	UserRating	Region
Mah nakorn (2004)	296041028	8	Bangkok
Mah nakorn (2004)	296041028	8	Bangkok
No Strings Attached (2011)	239615275	6	Eastern Time (US & Canada)
No Strings Attached (2011)	239615275	6	Eastern Time (US & Canada)
The Paperboy (2012)	36937267	7	Belgrade
The Paperboy (2012)	36937267	7	Belgrade
La vita ? bella (1997)	45826125	5	Budapest
La vita ? bella (1997)	45826125	5	Budapest
Eternal Sunshine of the Spotless Mind (2004)	45826125	4	Budapest
Eternal Sunshine of the Spotless Mind (2004)	45826125	4	Budapest

datasciencedojo
unleash the data scientist in you