

# DATA() { EXPOSED;

DATA EXPOSED SPECIAL

## Around the Clock with Azure SQL and Azure Data Factory

Americas

February 3, 2021

09:00 - 17:00 PT

Asia

February 4, 2021

09:00 - 17:00 SGT

16 Sessions | 2 Ask the Expert Panels | 1 Hackathon



HOSTED BY

Wee Hyong Tok & Anna Hoffman

DATA() {  
EXPOSED;

# Develop with Azure SQL in the Browser

Drew Skwiers-Koballa

# Writing T-SQL Queries

- Visualize data before adding a new dataset to a pipeline
- Ah-hoc reporting needs
- DDL (data definition language) statements for tables, stored procedures, indexes, etc.

The screenshot shows the 'Copy data' task configuration window in Azure Data Factory. The 'Source' tab is active, showing the 'Source dataset' as 'AzureSqlTable1'. Below this, the 'Dataset properties' section is expanded, displaying a table with columns 'NAME', 'VALUE', and 'TYPE'. The 'VALUE' column has a text input field containing 'Value', and the 'TYPE' column is set to 'string'. At the bottom, the 'Use query' section has three radio buttons: 'Table', 'Query' (which is selected), and 'Stored procedure'. Below the 'Query' radio button is a large text area for entering the query.

NAME	VALUE	TYPE
	<input type="text" value="Value"/>	string

Use query: ☒ Table ☐ Query ☐ Stored procedure

Query:

# Writing T-SQL Queries... in a browser!

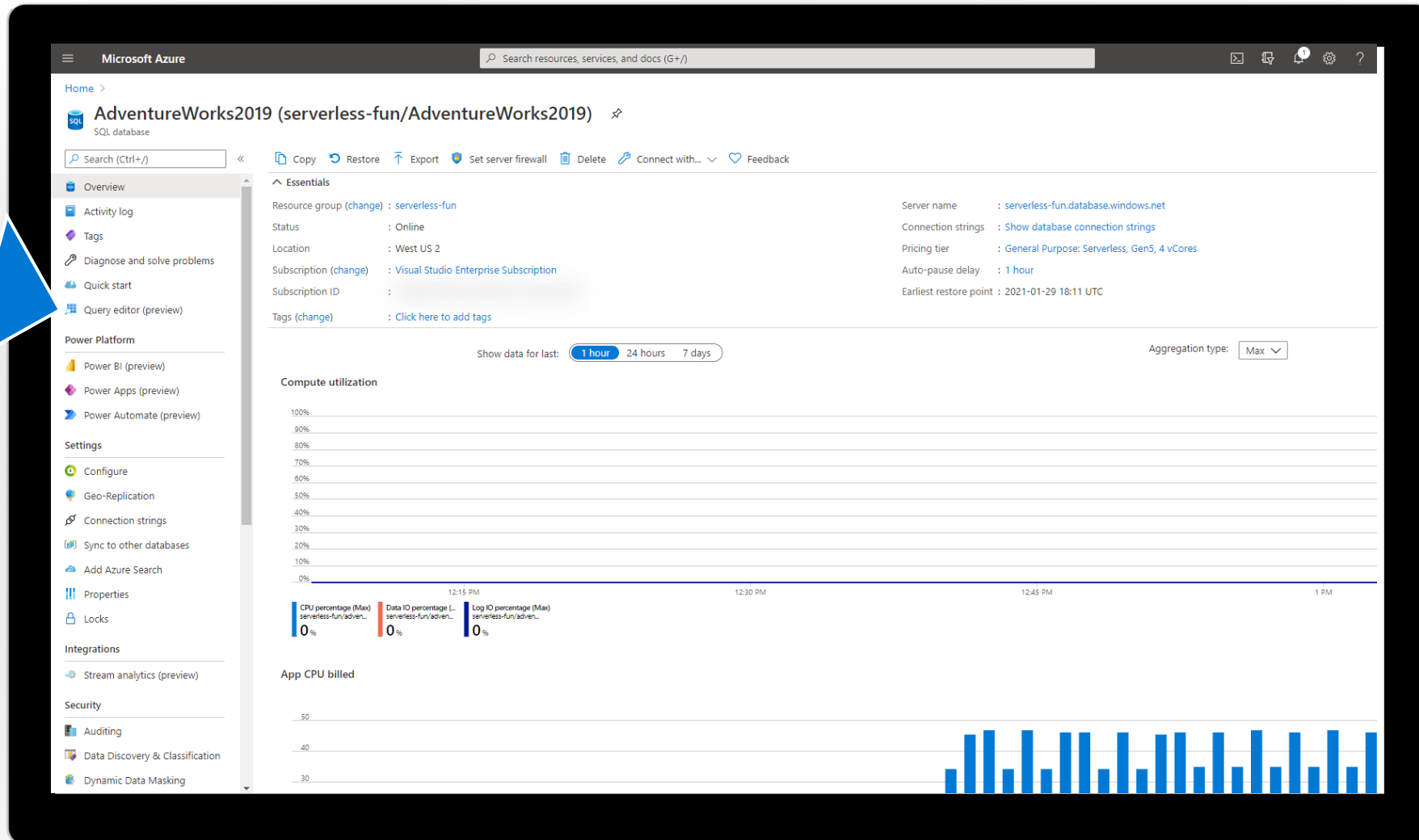
## On the Portal

SQL Database Query Editor can connect and query your Azure SQL database.

## From GitHub in Codespaces beta

The mssql extension for VS Code supports connecting to your Azure SQL, SQL Server, or Synapse SQL instances.

# Azure SQL Query editor in the Portal



The screenshot displays the Azure Portal interface for an AdventureWorks2019 SQL database. The left-hand navigation pane includes sections for Overview, Power Platform, Settings, Integrations, and Security. The 'Query editor (preview)' option is highlighted in the Overview section. A large blue arrow points to this option. The main content area shows the database's Essentials, including resource group, status, location, subscription, and server name. Below this, there are charts for Compute utilization and App CPU billed.

**AdventureWorks2019 (serverless-fun/AdventureWorks2019)**  
SQL database

Search (Ctrl+)

Copy Restore Export Set server firewall Delete Connect with... Feedback

**Essentials**

Resource group (change) : serverless-fun  
Status : Online  
Location : West US 2  
Subscription (change) : Visual Studio Enterprise Subscription  
Subscription ID :  
Tags (change) : Click here to add tags

Server name : serverless-fun.database.windows.net  
Connection strings : Show database connection strings  
Pricing tier : General Purpose: Serverless, Gen5, 4 vCores  
Auto-pause delay : 1 hour  
Earliest restore point : 2021-01-29 18:11 UTC

Show data for last: 1 hour 24 hours 7 days Aggregation type: Max

**Compute utilization**

CPU percentage (Max) serverless-fun/adven... 0%  
Data IO percentage (Max) serverless-fun/adven... 0%  
Log IO percentage (Max) serverless-fun/adven... 0%

**App CPU billed**

# Portal Query editor

The screenshot displays the Portal Query editor interface. On the left, the 'AdventureWorks2019 (sqladmin)' database is selected, showing a list of tables under the 'Tables' folder. A message indicates that the object explorer is limited and full capability requires opening SSDT. The right pane shows a query editor with a query titled 'Query 1'. The query text is 'SELECT TOP 100 \* FROM SalesLT.Customer'. A dropdown menu is open, showing a list of tables from the 'SalesLT' schema, with 'SalesLT.Customer' selected. The query editor includes buttons for 'Run', 'Cancel query', 'Save query', 'Export data as', and 'Show only Editor'.

AdventureWorks2019 (sqladmin)

Showing limited object explorer here. For full capability please open SSDT.

Tables

- > dbo.BuildVersion ...
- > dbo.ErrorLog ...
- > SalesLT.Address ...
- > SalesLT.Customer ...
- > SalesLT.CustomerAddress ...
- > SalesLT.Product ...
- > SalesLT.ProductCategory ...

Query 1

Run Cancel query Save query Export data as Show only Editor

```
1 SELECT TOP 100 *
2 FROM SalesLT.Customer
```

- abc [SalesLT].[Address]
- abc [SalesLT].[CustomerAddress]
- abc [SalesLT].[Customer]
- abc [SalesLT].[ProductCategory]
- abc [SalesLT].[ProductDescription]
- abc [SalesLT].[ProductModelProductDescription]
- abc [SalesLT].[ProductModel]
- abc [SalesLT].[Product]
- abc [SalesLT].[SalesOrderDetail]
- abc [SalesLT].[SalesOrderHeader]
- abc [SalesLT].[vGetAllCategories]
- abc [SalesLT].[vProductAndDescription]

**Write and run queries**

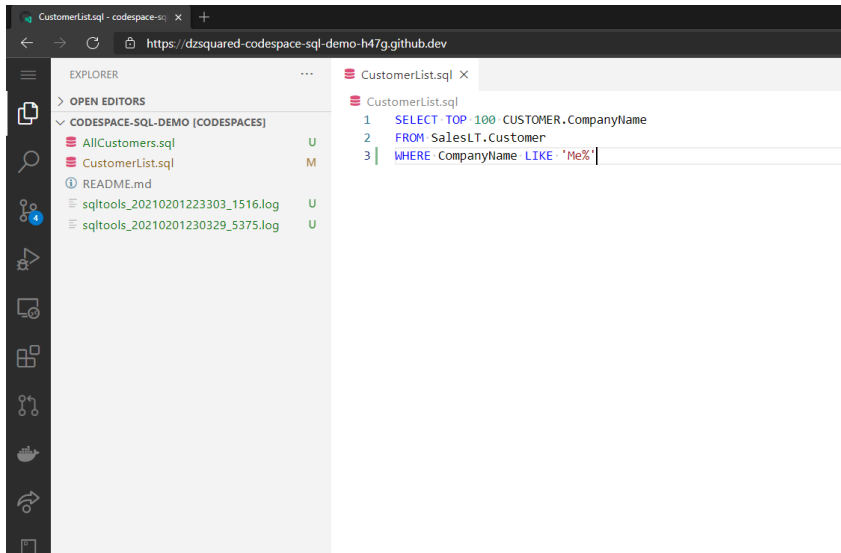
Auto-complete

Open existing query files

# What is a Codespace?

## What it looks like:

VS Code in the browser.



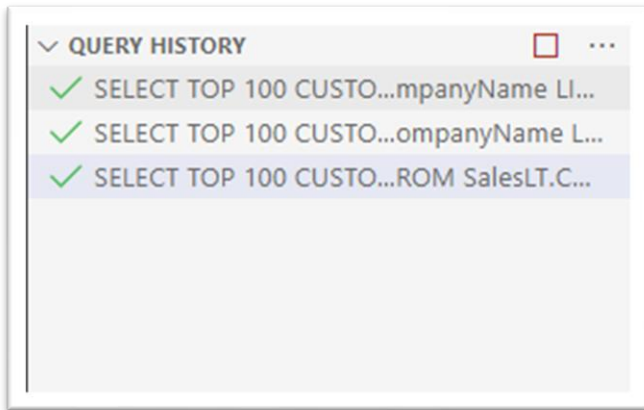
## Behind the scenes:

- A container hosting the VS Code application and extensions
- The environment auto-pauses after a period of inactivity
- Environment can be customized with *.devcontainer* files

# mssql extension features

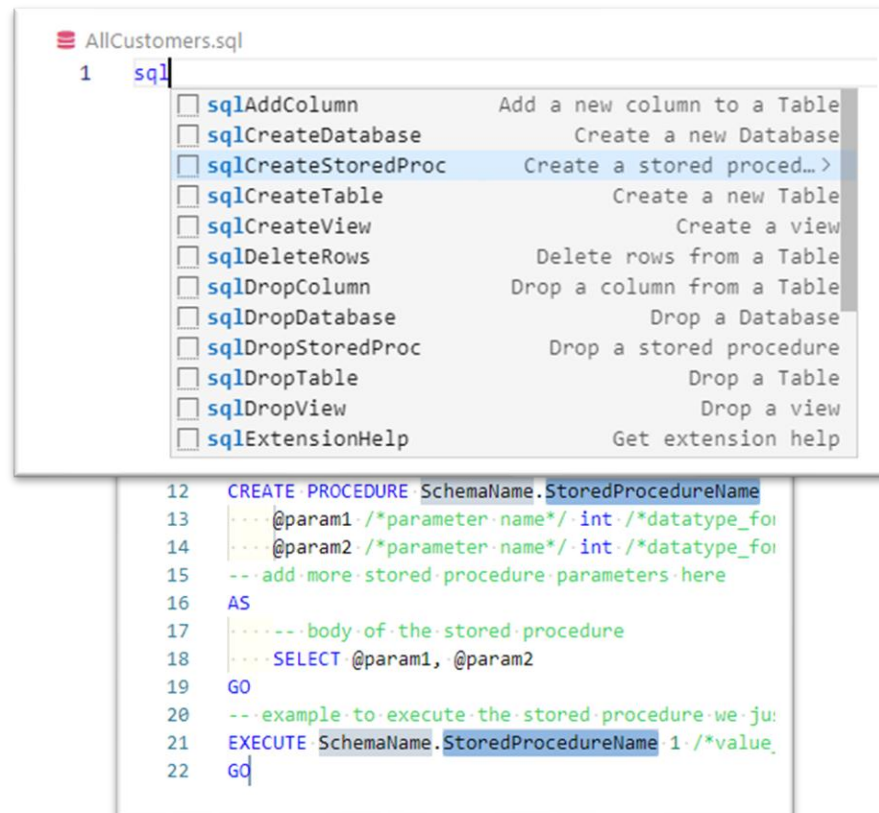
## Query history

- Track recent query history during a session
- Reopen previous queries



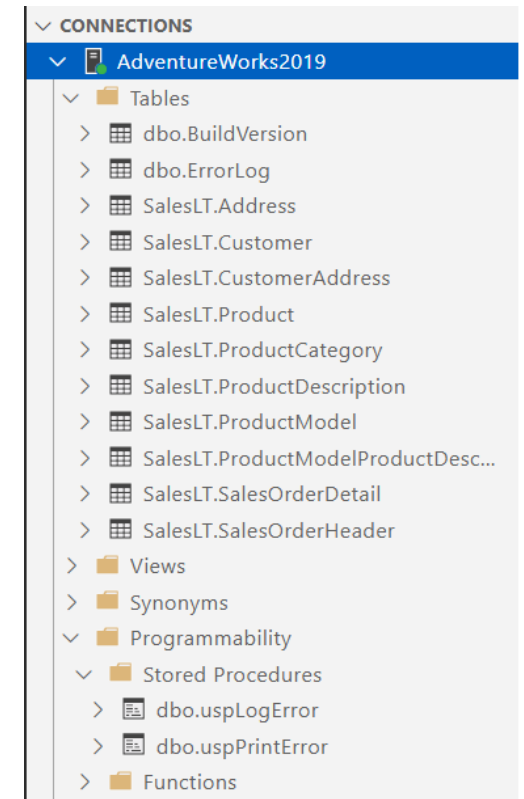
## Snippets

- Save commonly used code templates as VS Code snippets



## Connect to SQL Server and Azure SQL

- Save connections to one or many databases

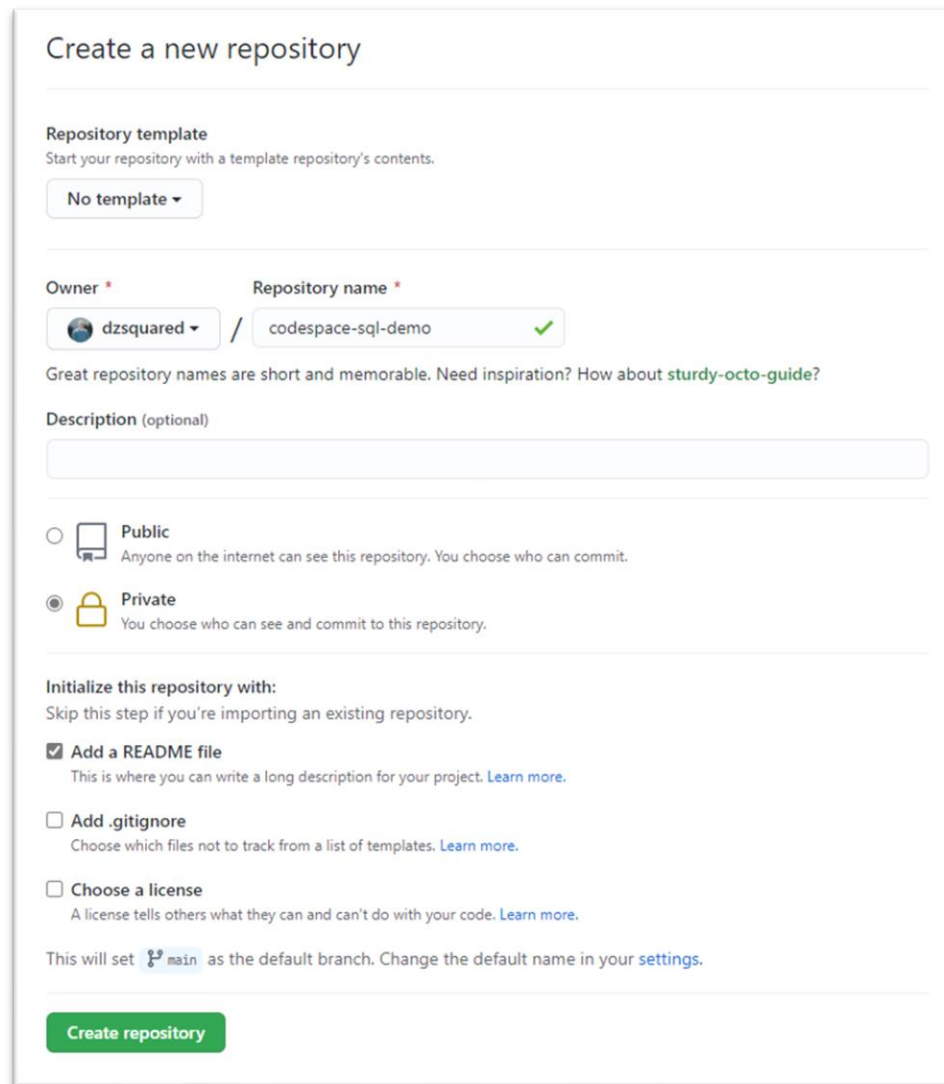




# mssql extension for VS Code and GitHub Codespaces

## Starting from Scratch

1. Create a new repository on GitHub
2. Create new Codespace from GitHub



The screenshot shows the 'Create a new repository' form on GitHub. The 'Repository template' section has a 'No template' dropdown. The 'Owner' is 'dzsquared' and the 'Repository name' is 'codespace-sql-demo', which is marked as valid with a green checkmark. A hint suggests repository names should be short and memorable. The 'Description' field is optional and empty. Under 'Initialize this repository with:', the 'Public' option is selected, and 'Add a README file' is checked. Other options like 'Add .gitignore' and 'Choose a license' are unchecked. A note at the bottom states the default branch will be 'main'. A green 'Create repository' button is at the bottom.

Create a new repository

Repository template  
Start your repository with a template repository's contents.

No template ▾

Owner \* Repository name \*  
dzsquared / codespace-sql-demo ✓

Great repository names are short and memorable. Need inspiration? How about [sturdy-octo-guide](#)?

Description (optional)

☐ Public  
Anyone on the internet can see this repository. You choose who can commit.

☒ Private  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

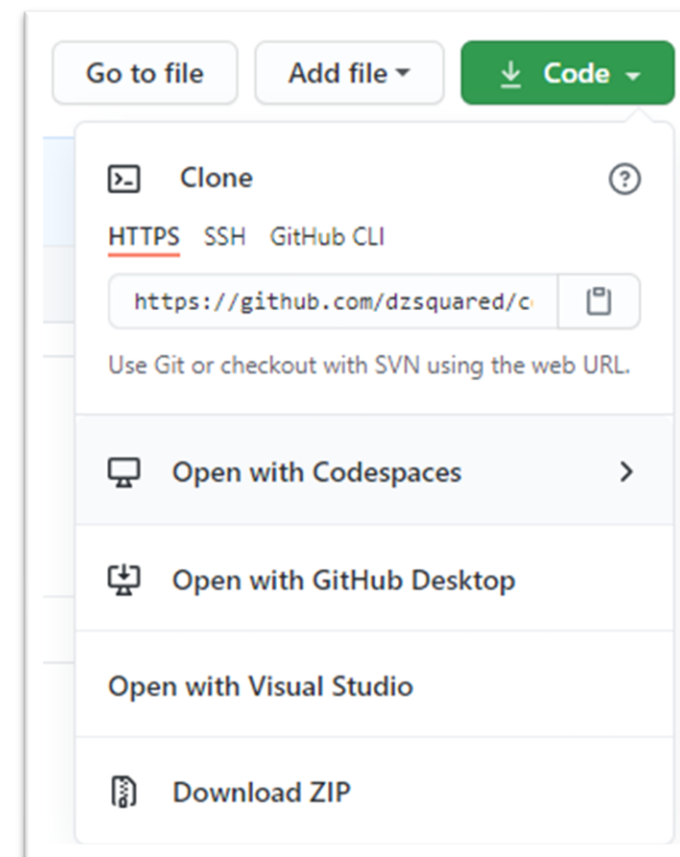
☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

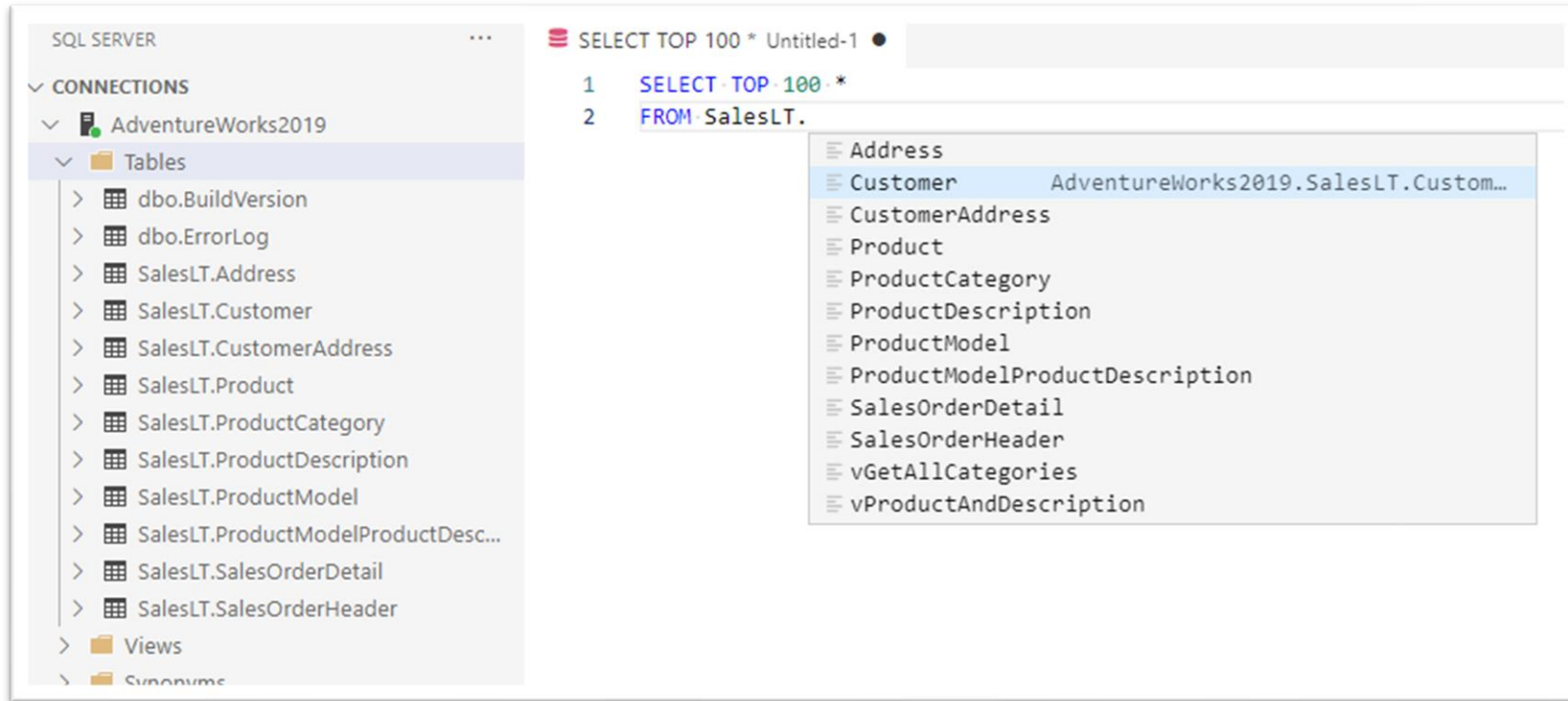
☐ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository



# GitHub Codespace Basics



**Write and run queries**

Intellisense

Organize queries in source control

**+ all the mssql extension features**

# Recap

**Quick!** Query editor for Azure SQL database in the Portal

<https://docs.microsoft.com/azure/azure-sql/database/connect-query-portal>

**Cloud Environment:** GitHub Codespace with mssql extension for VS Code

<https://github.com/features/codespaces>

<https://github.com/microsoft/vscode-mssql>

DATA() {  
EXPOSED;

Learn with us!

View our on-demand playlist:  
[aka.ms/azuresqlandadf](https://aka.ms/azuresqlandadf)

@AzureSQL  
@AzDataFactory



# Thank you!