**Connect to an SQL database using Azure Key Vault**

**Store Secrets in Azure Key Vault**

Before connecting to the SQL database, securely store credentials in **Azure Key Vault**.

Sql database 🡪 setting 🡪 connection string 🡪 JDBC

jdbc:sqlserver://sqldbshubha.database.windows.net:1433;database=sqldb\_ncpl;user=admin12345@sqldbshubha;password={your\_password\_here};encrypt=true;trustServerCertificate=false;hostNameInCertificate=\*.database.windows.net;loginTimeout=30;

1. **Go to Azure Portal → Key Vault**
2. **Create Secrets** for:
   * sql-url : jdbc:sqlserver://sqldbshubha.database.windows.net:1433;database=sqldb\_ncpl
   * sql-username: admin12345
   * sql-password: Dhrithi@12345

Go to key vault , setting 🡪 objects 🡪 secretes 🡪 create key

URL KEY: A screenshot of a computer

Description automatically generatedusername Key: A screenshot of a computer

Description automatically generated A screenshot of a login box

Description automatically generatedNow lauch Databricks and create Secret scope if not created previously

Now open a notebook and check if the scope is created correctly using

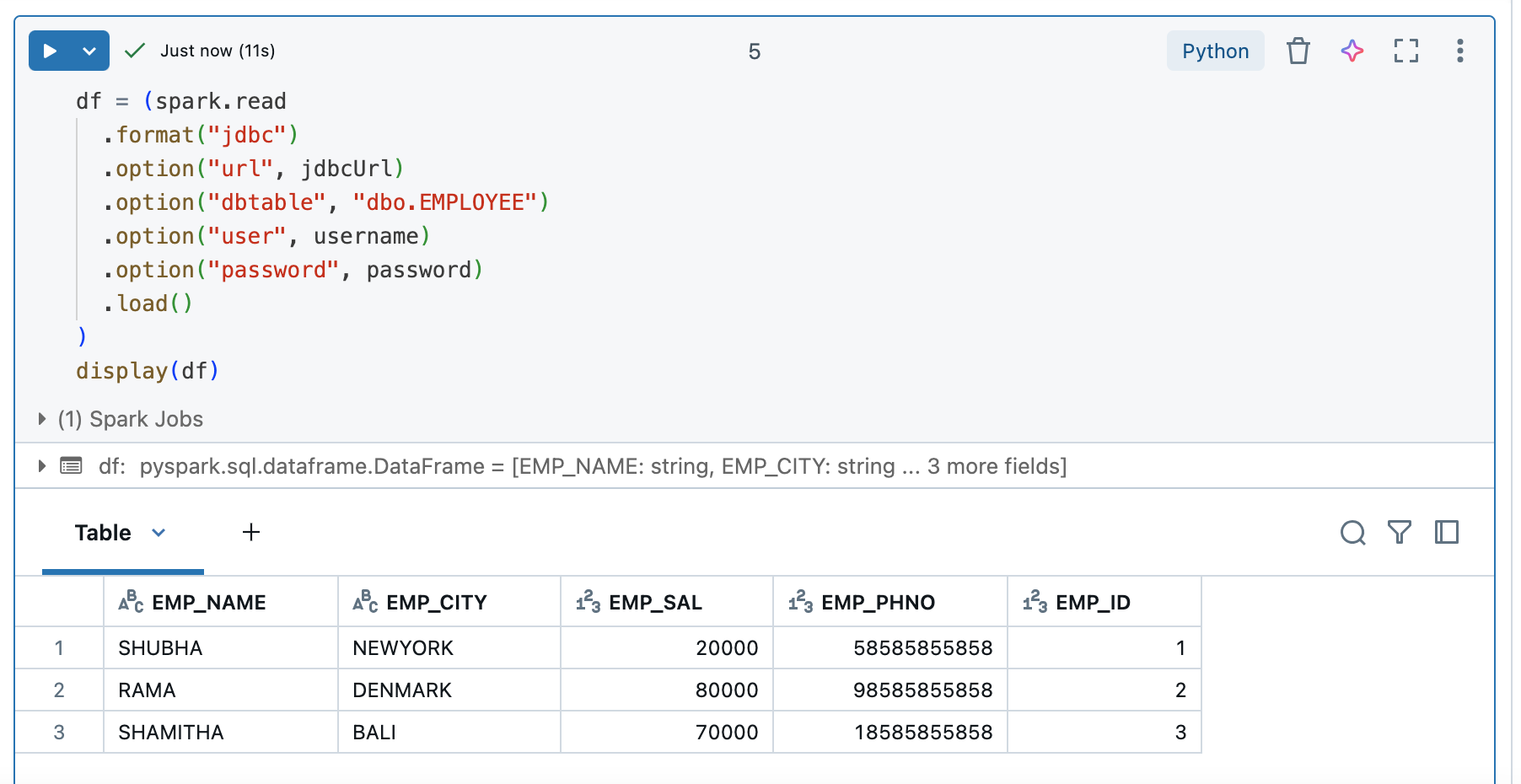
Dbutils.secrets.listScope() command A screenshot of a phone

Description automatically generated A screenshot of a computer

Description automatically generated Now we need to make a connection with SQL using the key which we created

Using code

A screenshot of a computer

Description automatically generated  Instead of creating variables separately we directly give the key values in read code

Like below

df = (spark.read

.format("jdbc")

.option("url", dbutils.secrets.get(scope = "askvconnection", key = "URL"))

.option("dbtable", "SalesLT.Product")

.option("user", dbutils.secrets.get(scope = "askvconnection", key = "Username"))

.option("password", dbutils.secrets.get(scope = "askvconnection", key = "Password"))

.load()

) A screenshot of a computer

Description automatically generated A screenshot of a cell phone

Description automatically generated Table is created in SQL database we can check by using SSMS A screenshot of a computer

Description automatically generated As read and write codes can be used multiple times for different tables, we can use functions to reuse the same code multiple times, we can achieve this by writing the reuseable code in different notebook and use it in any notebook.

We define function suing def key word

READ USING TABLE NAMEA screen shot of a computer code

Description automatically generated READ USING QUERYA screenshot of a computer code

Description automatically generated WRITE INTO SQLA screen shot of a computer code

Description automatically generated Now to use these functions code in another notebook we need to use line command %run/notebook path

%run/ /Workspace/Users/shivashankerbudda@gmail.com /SQL\_connection

Use this code in the notebook where we want to use the reusable code A screenshot of a computer

Description automatically generated Using read\_sql() function in another notebook to read data A screenshot of a computer

Description automatically generated Using read\_query\_sql() function A screenshot of a computer

Description automatically generated