**Anton Slizh’s Spark SQL Homework**

Description

**Task 1**

* Fork [jvm](https://git.epam.com/epmc-bdcc/trainings/bd201/m07_sparksql_jvm_azure) or [python](https://git.epam.com/epmc-bdcc/trainings/bd201/m07_sparksql_python_azure) base of the project (please, preserve project naming).
* Data is in Azure ADLS gen2 storage **bd201stacc/m07sparksql**. You can explore it, by connecting **Storage Account** in [Azure Storage Explorer](https://azure.microsoft.com/en-us/features/storage-explorer) if needed, use this SAS URI in Azure
* Copy hotel/weather and expedia data from Azure ADLS gen2 storage into provisioned with terraform Azure ADLS gen2 storage.
* Create Databricks Notebooks (Azure libraries like hadoop-azure and azure-storage are already part of Databricks environment, details are described [here](https://docs.databricks.com/data/data-sources/azure/adls-gen2/index.html)). Use ABFS drivers and OAuth credentials like below:
* Create delta tables based on data in storage account.
* Using Spark SQL calculate and visualize in Databricks Notebooks (for queries use hotel\_id - join key, srch\_ci- checkin, srch\_co - checkout:
  + Top 10 hotels with max absolute temperature difference by month.
  + Top 10 busy (e.g., with the biggest visits count) hotels for each month. If visit dates refer to several months, it should be counted for all affected months.
  + For visits with extended stay (more than 7 days) calculate weather trend (the day temperature difference between last and first day of stay) and average temperature during stay.
* For designed queries analyze execution plan. Map execution plan steps with real query. Specify the most time (resource) consuming part. For each analysis you could create tables with proper structure and partitioning if necessary.
* Deploy Databricks Notebook on cluster, to setup infrastructure use terraform scripts from module. Default resource parameters (specifically memory) will not work because of free tier limitations. You needed to setup memory and cores properly.
* Development and testing is recommended to do locally in your IDE environment with delta delta-core library.
* Store final DataMarts and intermediate data (joined data with all the fields from both datasets) in provisioned with terraform Azure ADLS gen2 storage preserving data partitioning in parquet format in “data” container (it marked with prevent\_destroy=true and will survive terraform destroy).

**Expected results**

* Repository with notebook (with output results), configuration scripts, application sources, execution plan dumps, analysis and etc.
* Upload in task Readme MD file with link on repo, fully documented homework with screenshots and comments.

**DO NOT FORGET TO DELETE UNNECESSARY RESOURCES IN THE CLOUD VIA "TERRAFORM DESTROY", WHEN YOU FINISH WORK WITH THEM!**

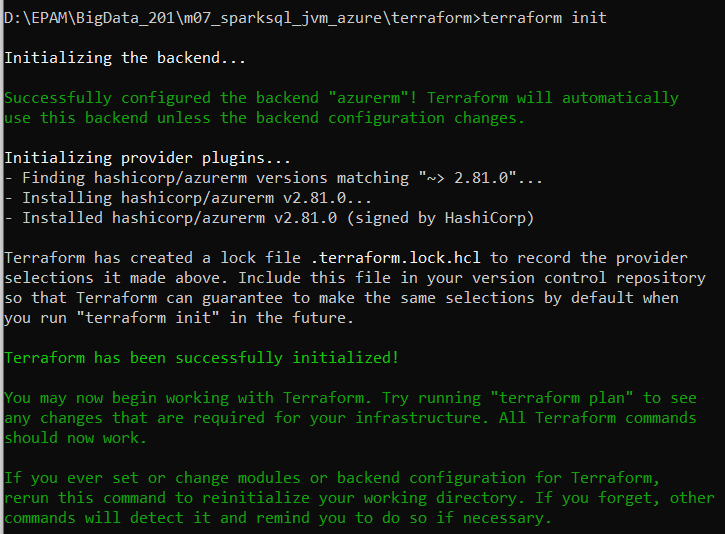
**Evaluation Criteria**

**Task 1**

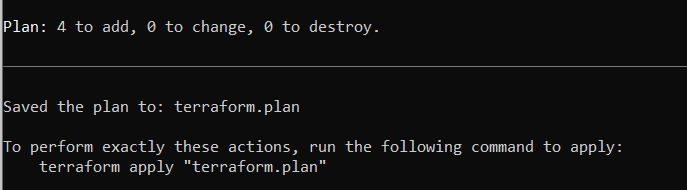
1. Application is working as expected - 45 points.
2. CI/CD with parametrization and credentials management - 15 points.
3. Code quality - 10 points.
4. Execution plans are provided and analyzed - 10 points.
5. Screenshots and comment - 10 points.
6. Code is well-documented - 10 points.

**Deploying infrastructure with terraform:**

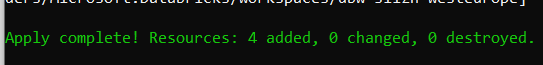
>terraform init



>terraform plan -out terraform.plan



>terraform apply terraform.plan

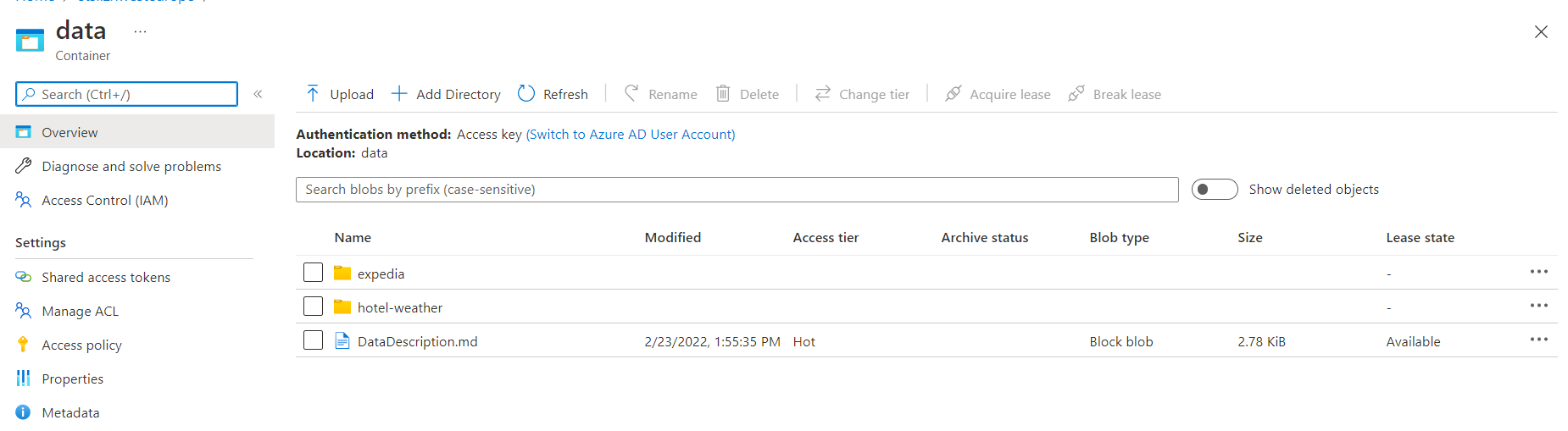


**Copying data from Azure ADLS gen2 storage:**

Generating SAS URI to access to destination storage *(slizhwesteurope).*

Copying the data from source storage to destination storage using Azure Cloud Shell and *azcopy* utility

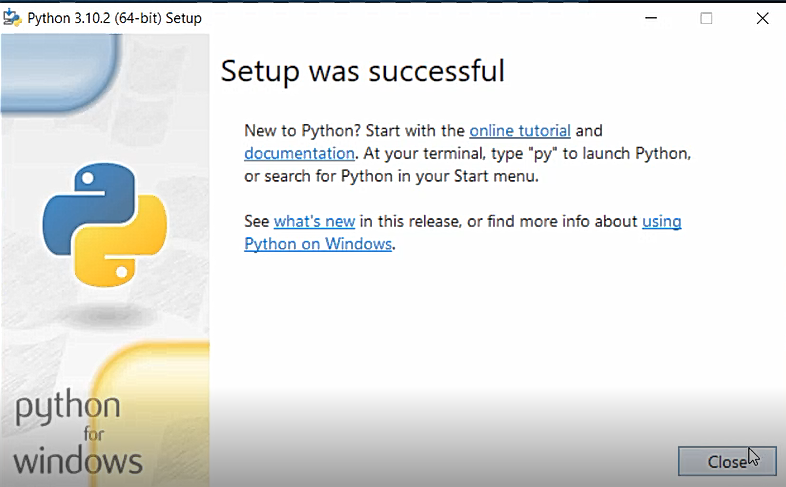
Copied data



**Configuring access to destination Storage Account**

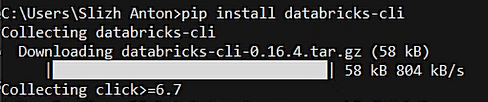
Downloading Python 3.10.2 from <https://www.python.org/ftp/python/3.10.2/python-3.10.2-amd64.exe>

Installing



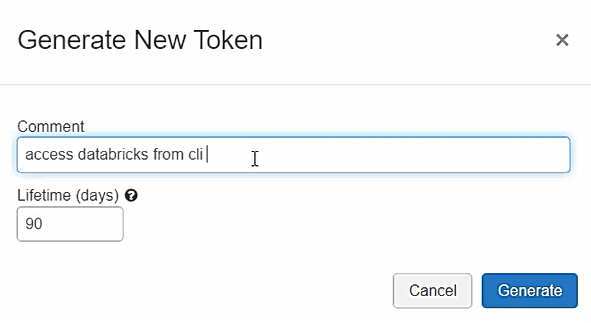
Installing databricks-cli

>pip install databricks-cli



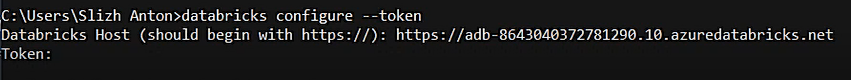


Creating databricks access-token



Authenticating

>databricks configure –token



Clusters list:

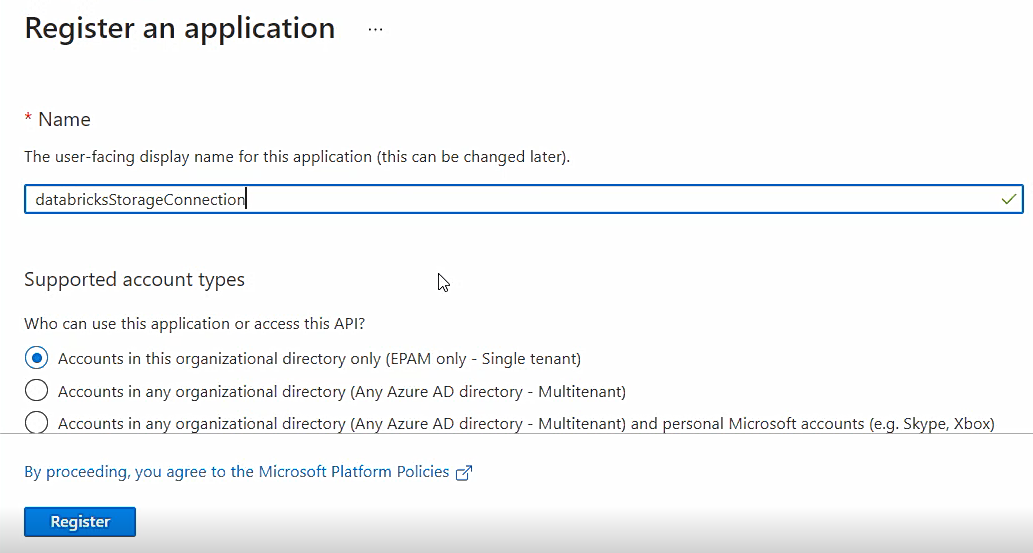
>databricks clusters list



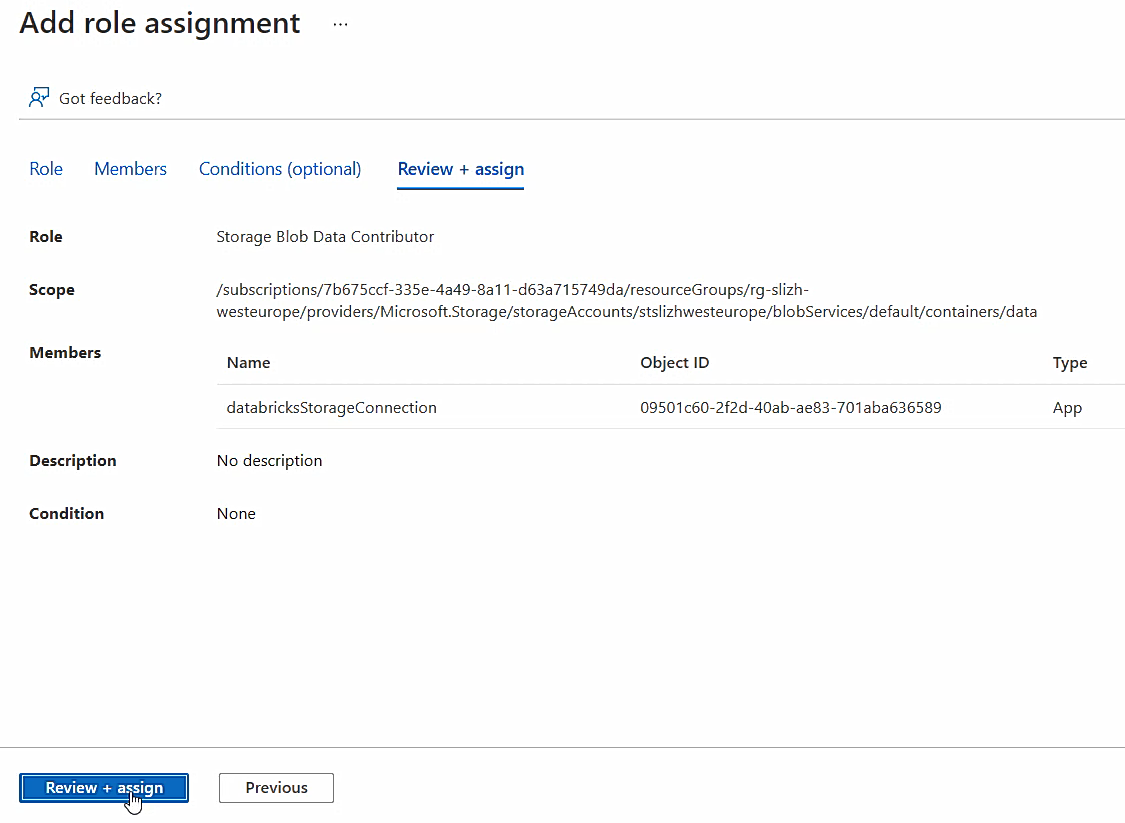
Creating databricks secrets scope

>databricks secrets create-scope –scope slizhscope –initial-manage-principal “users”

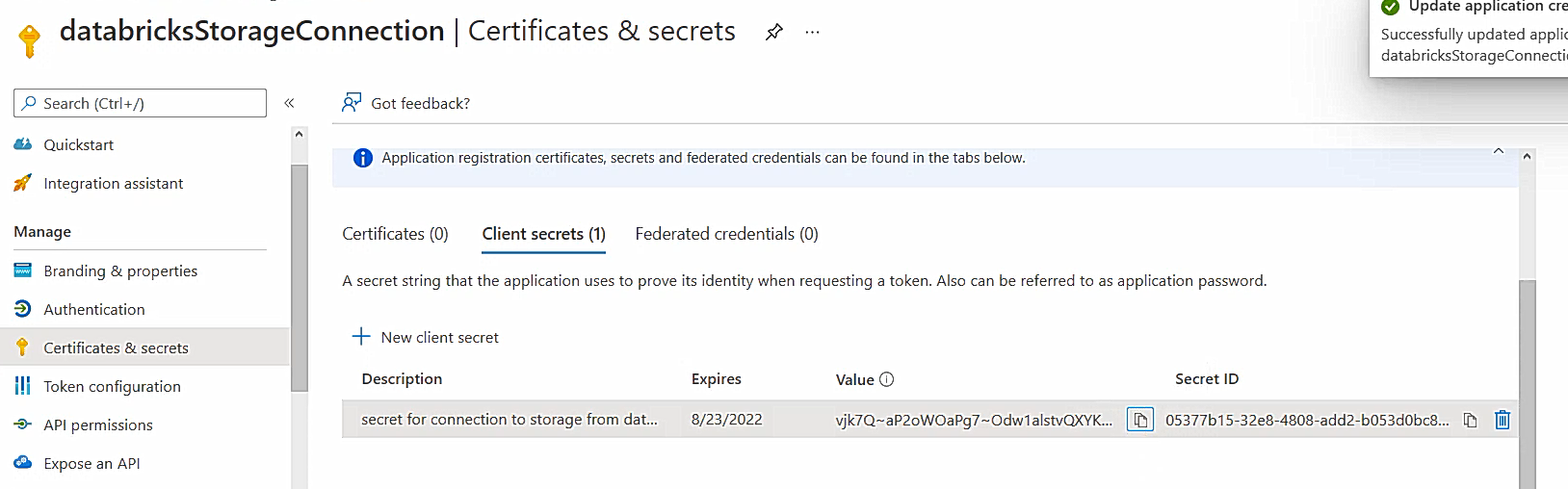
Creating application



Adding role for application

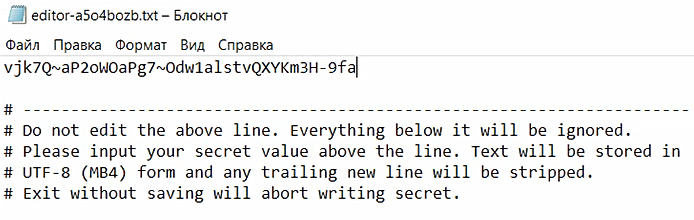


Creating secret for application



Put secret-key into secrets-scope

>databricks secrets put –scope slizhscope –key storagekey

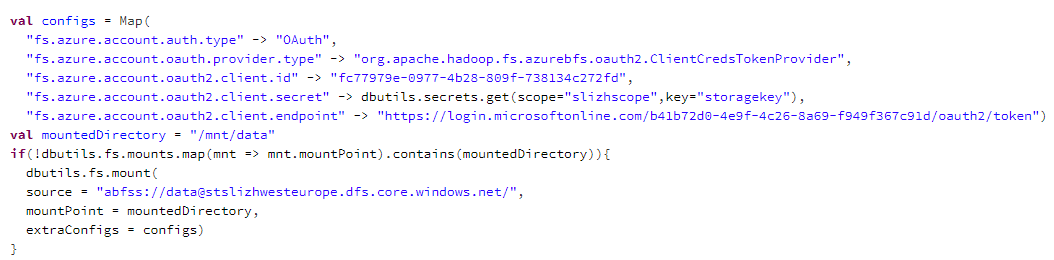


Using *dbutils* to get access to secret key from notebook

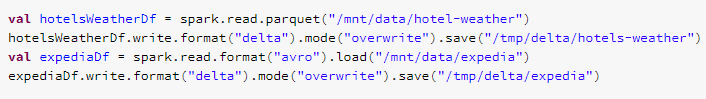


**Creating notebook**

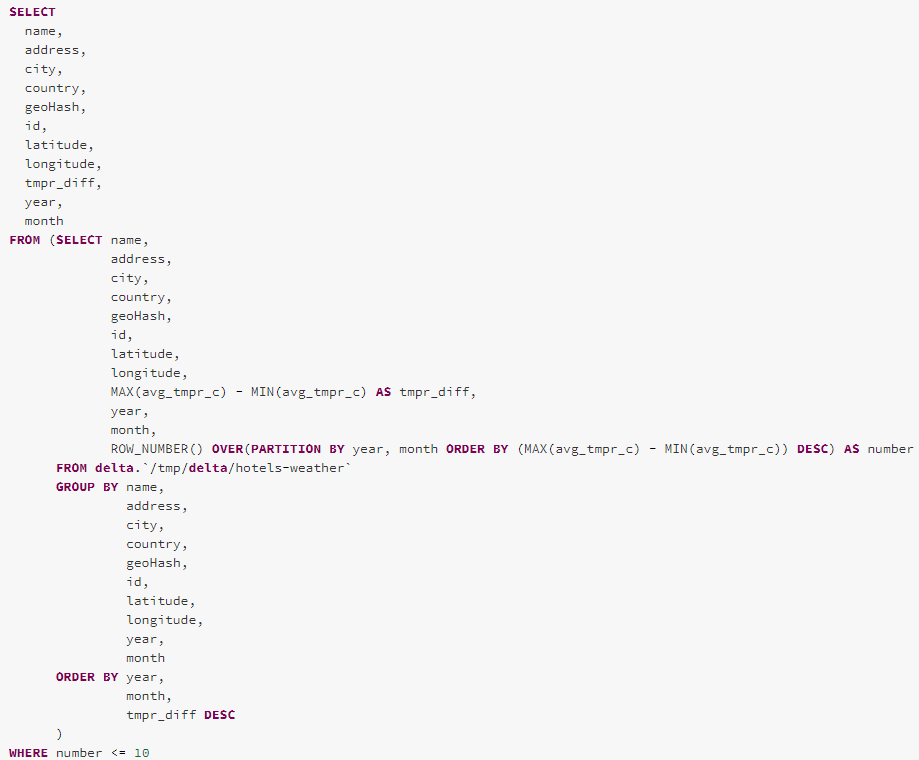
Mounting data-directory



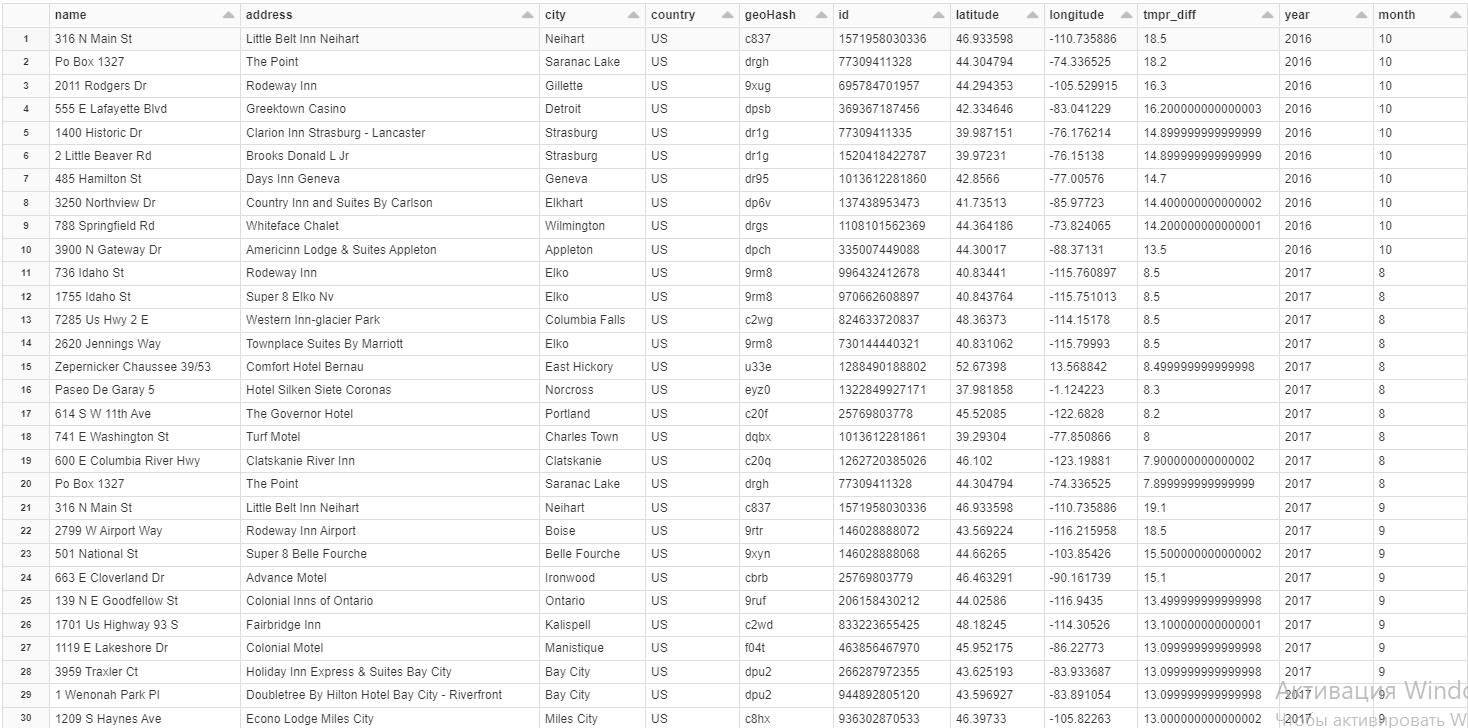
Saving data to delta-tables



First SQL query: Top 10 hotels with max absolute temperature difference by month.



Result:

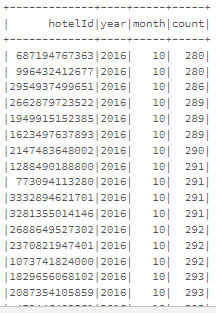


Second SQL query: Top 10 busy (e.g., with the biggest visits count) hotels for each month. If visit dates refer to several months, it should be counted for all affected months.

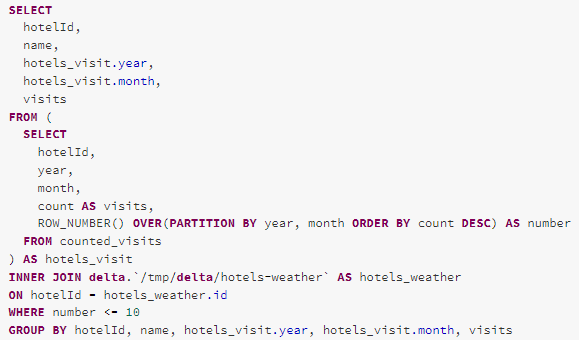
Counting visits for each hotel by month



Result:



Selecting top 10 busy (e.g., with the biggest visits count) hotels for each month.

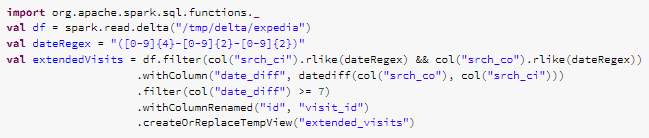


Result:

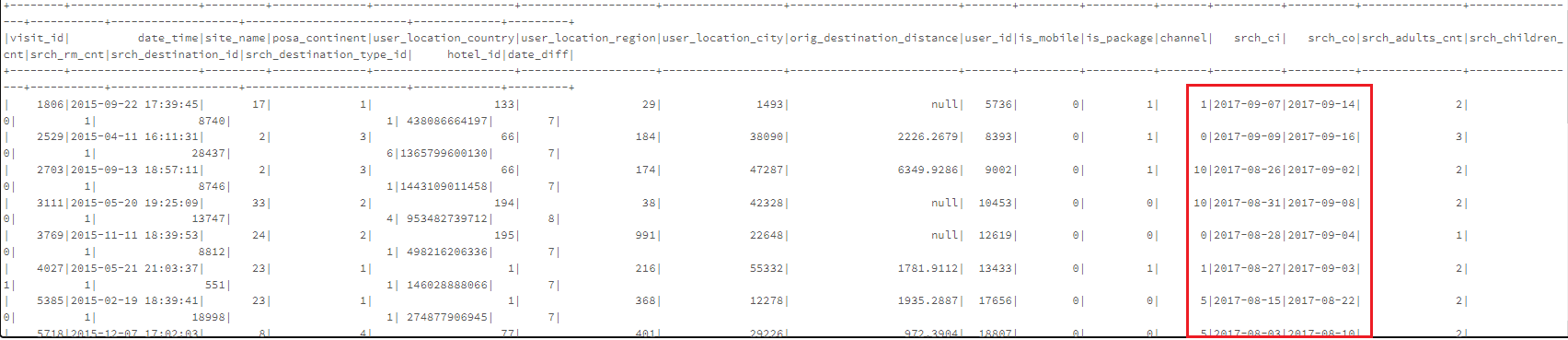


Third SQL query: For visits with extended stay (more than 7 days) calculate weather trend (the day temperature difference between last and first day of stay) and average temperature during stay.

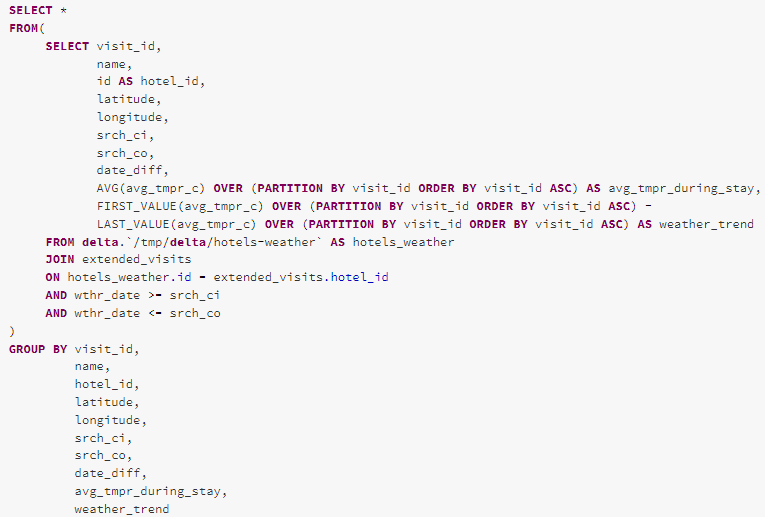
Filter extended (more than 7 days) stays



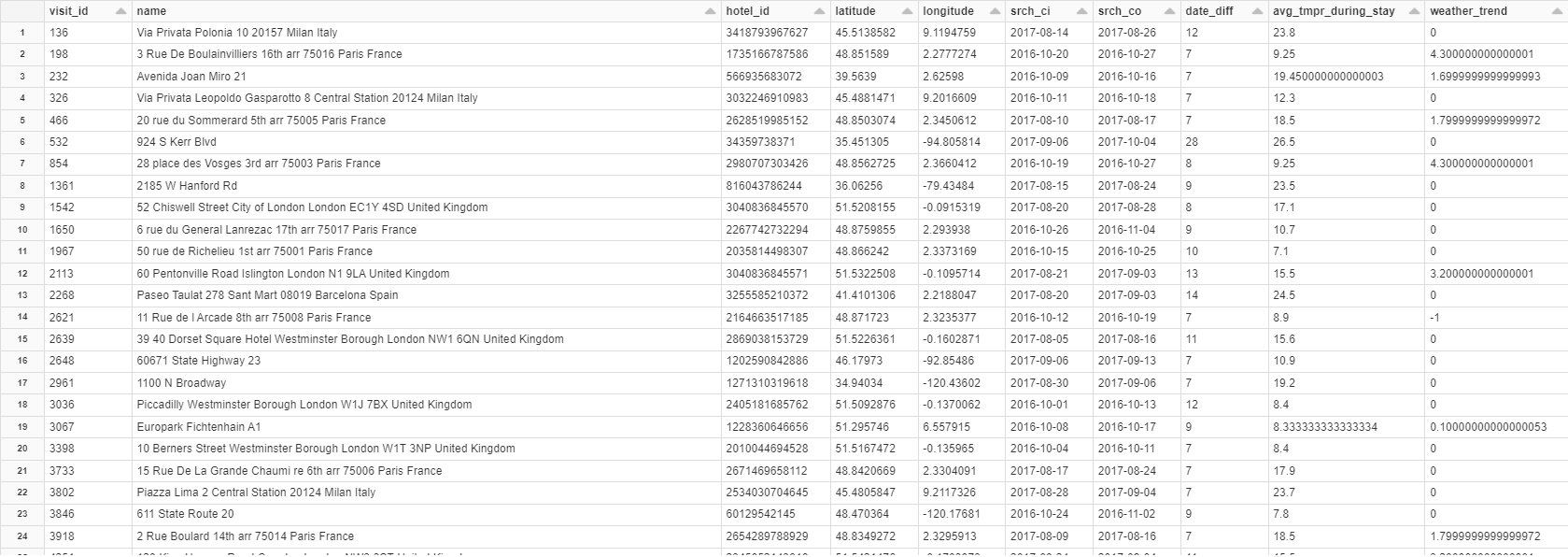
Result



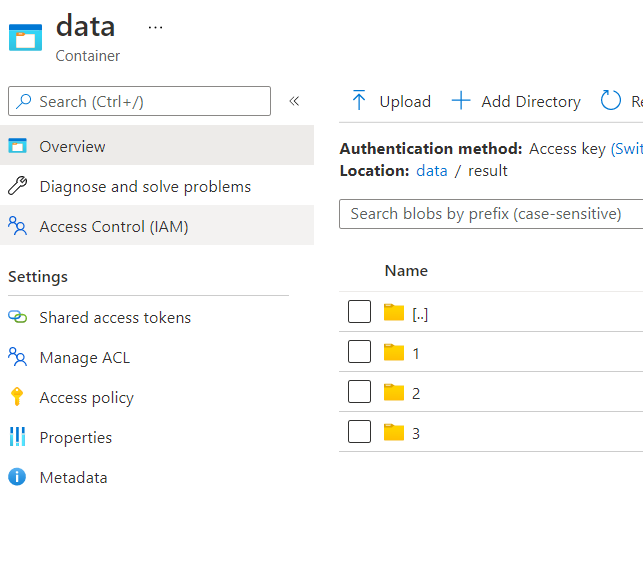
Calculating weather trend (the day temperature difference between last and first day of stay) and average temperature during stay.

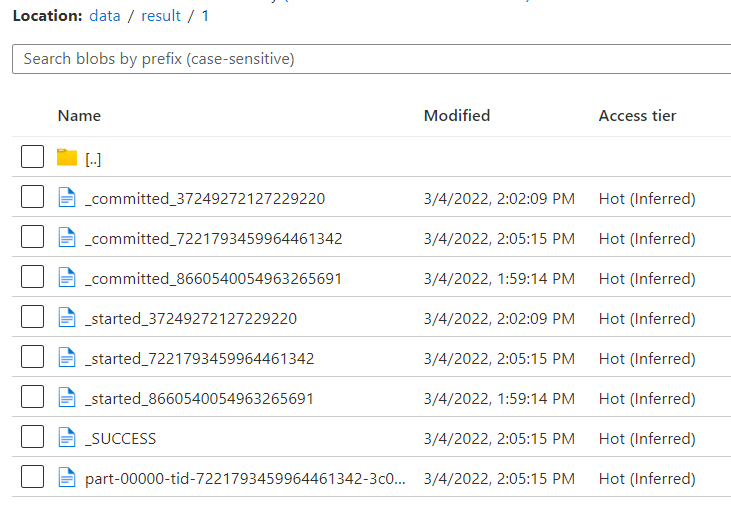


Result



**Saved results:**

****

****

**Execution plans**

Look at screenshots 1-3