**Performing CDC and Initializing a Cassandra Database Container**

The "Activity13.5" folder is opened in Visual Studio Code, containing the files from Activity 13.4.

A screenshot of a computer

Description automatically generated

The container.py code is updated to include the necessary lines for creating a Cassandra database container.

A screen shot of a computer

Description automatically generated

The container.py code is further enhanced with the addition of lines to delete a Cassandra container.

A computer screen with text

Description automatically generated

A new file named cassandradb.py is successfully added to the solution.

A screen shot of a computer

Description automatically generated

The comment importing the Cluster module is removed from the container.py code.

A screen shot of a computer

Description automatically generated

The init\_cassandra() function is copied and added to the container.py program.

A computer screen shot of a code

Description automatically generated

A call to init\_cassandra() is introduced in the container.py program when the **-init** parameter is passed.

A screenshot of a computer

Description automatically generated

The import statement for the cassandradb module is uncommented within the scheduler.py program.

A screen shot of a computer

Description automatically generated

A call to cassandradb.delete() is incorporated within the clearout() function in the scheduler.py program.

A screen shot of a computer

Description automatically generated

The cassandra() function is added to the scheduler.py program, facilitating data transfer with the Cassandra database.

A screen shot of a computer

Description automatically generated

The verify() function in the scheduler.py program is adjusted to retrieve data from the Cassandra database.

A computer screen shot of a program code

Description automatically generated

The process\_data() function in the scheduler.py program is updated with a call to the cassandra() function.

A screen shot of a computer program

Description automatically generated

The container.py program is executed from the command prompt with the **-create** parameter, resulting in successful operation.

A screenshot of a computer program

Description automatically generated

Docker Desktop visually confirms the active state of all four database containers (MySQL, MongoDB, Redis, and Cassandra).

A screenshot of a computer

Description automatically generated

The container.py program is run from the command prompt with the **-init** parameter, initiating the system setup.

A screenshot of a computer

Description automatically generated

The scheduler.py program is executed from the command prompt, showcasing its operational status.

A screenshot of a computer

Description automatically generated

The command prompt window displays the closure of the scheduler.py program window and the subsequent execution of the container.py program with the **-delete** parameter.

A screenshot of a computer program

Description automatically generated

Docker Desktop reflects the removal of all four database containers (MySQL, MongoDB, Redis, and Cassandra).

A screenshot of a computer

Description automatically generated