Bonus Questions (Conceptual)

1. Why is data cleaning important in real-time data processing?

   Data cleaning is important because real-time systems depend on fast and accurate decisions. If the incoming data has errors, duplicates, or missing values, the whole pipeline can produce wrong results or even fail. Clean data ensures that insights, dashboards, or machine learning models get consistent and reliable inputs. In simple terms, data cleaning makes sure we are “feeding good data to get good outcomes” without delays or interruptions.

2. What are pipeline artifacts and how are they used in DevOps workflows?

   Pipeline artifacts are basically the outputs generated when a pipeline runs, such as files, datasets, reports, or application builds. Instead of recreating these outputs every time, DevOps saves them as artifacts so that later stages (like testing, deployment, or release) can use the exact same version. This helps teams maintain consistency, avoid errors caused by mismatched files, and ensure that what gets deployed in production is exactly what was tested earlier.

3. How would you modify the pipeline to store the cleaned data into Azure Blob Storage?

   To store the cleaned sales data in Azure Blob Storage, we would extend the pipeline by adding an upload step after the cleaning process. This step uses Azure CLI or a storage upload task to push the clean_sales_data.csv file into a chosen blob container in an Azure Storage Account. By doing this, the cleaned dataset is not only available as an artifact inside DevOps but also stored in the cloud, making it accessible for analytics, reporting, or integration with other Azure services like Power BI or Databricks.

Example code:

```yaml
# Upload cleaned CSV to Azure Blob Storage
- task: AzureCLI@2
  inputs:
    azureSubscription: 'MyServiceConnection'   # Service connection in DevOps
    scriptType: 'bash'
    scriptLocation: 'inlineScript'
    inlineScript: |
      az storage blob upload \
        --account-name <yourStorageAccount> \
        --container-name <yourContainer> \
        --name clean_sales_data.csv \
        --file data/clean_sales_data.csv \
        --auth-mode login
```