1. *Why should one use Azure Key Vault when working in the Azure environment? What are the pros and cons? What are the alternatives?*

pros

* separate app security from the app’s code
* keep all credentials in one location so administration is easier

2. *How do you achieve loop functionality within a Azure Data Factory pipeline? Why would you need to use this functionality in a data pipeline?*

how

* add a ForEach activity
* Under Settings -> Items, enter what the activity should iterate over. This could be a list of constants, a list of files in a folder, etc.
* Inside the activity, add activities you would like to be performed for each iteration item. For example, this could be a call to another pipeline that will get performed for each item.

why

* need for repetitive tasks (in our example, we needed it for repetitive loading of initial data across an array of values)

3. *What are expressions in Azure Data Factory? How are they helpful when designing a data pipeline? Please explain with an example.*

Expressions are ways to reference values within the factory that may be useful and can change throughout a pipeline. Instead of hardcoding values, you can use expressions to accommodate multiple possibilities.

An example is “smartfoods\_customers\_@{replace(pipeline().parameters.BatchDt,'-','')}”, which was used to reference a file based on the date that was being pulled in the pipeline.

4. *What are the pros and cons of parametrizing a dataset’s activity in Azure Data Factory?*

pros

* can account for possible values that need to be considered
* keeps number of objects in ADF lower since the dataset can be used in different ways

cons

* The dataset can become too complicated. If not documented well, another designer loses insight on the full use of the dataset.

5. *What are the different supported file formats and compression codecs in Azure Data Factory? When will you use a Parquet file over an ORC file? Why would you choose an AVRO file format over a Parquet file format?*

There are many different file formats such as Avro, Delimited Text, JSON, ORC, Parquet, and XML.

PARQUET can better store nested data than ORC.

Row-based AVRO have better writes than column-based PARQUET.

source: <https://blog.clairvoyantsoft.com/big-data-file-formats-3fb659903271>