XML

XML is a markup language, To define a syntax for encoding documents which are both humans and machines readable.

It is standards file format for exchange between applications.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<book>
 <name>A Song of Ice and Fire</name>
 <author>George R. R. Martin</author>
 <language>English</language>
 <genre>Epic fantasy</genre>
</book>
```

```
<students>
 <student>
  <name>Rick Grimes</name>
  <age>35</age>
  <subject>Maths</subject>
  <gender>Male</gender>
 </student>
 <student>
  <name>Daryl Dixon </name>
  <age>33</age>
  <subject>Science</subject>
  <gender>Male</gender>
 </student>
 <student>
  <name>Maggie</name>
  <age>36</age>
  <subject>Arts</subject>
  <gender>Female</gender>
 </student>
</students>
```

Performance:

- The redundant nature of the syntax causes higher storage and transportation cost when the volume of data is large.
- Not splitable since XML has an opening tag at the beginning and a closing tag at the end. You cannot start processing at any point in the middle of those tags.

JSON

- JavaScript Object Notation is a schema-less, text-based representation of structured data that is based on key-value pairs and ordered lists.
- It is used for storing and exchanging the data between computers, more commonly used in network communication
- JSON is language independent used by any programming language
- JSON allows you to create a hierarchical structure of your data.

JSON represents data in two ways:

- **Object**: a collection of name-value (or key-value) pairs. An object is defined within left ({) and right (}) braces. Each name-value pair begins with the name, followed by a colon, followed by the value. Name-value pairs are comma separated.
- **Array**: an ordered collection of values. An array is defined within left ([) and right (]) brackets. Items in the array are comma separated.

Example:

```
{
        "name":     "John",
        "salary":    56000,
        "married":   true
}
```

Json document with information about books

```
{
  "book":[
    {
      "id":"444",
      "language":"C",
      "edition":"First",
      "author":"Dennis Ritchie "
    },
    {
      "id":"555",
```

```
      "language":"C++",
      "edition":"second",
      "author":" Bjarne Stroustrup "
    }
  ]
}
```

Json Applications:

- It is widely used for JavaScript-based application, which includes browser extension and websites.
- You can transmit data between the server and web application using JSON.
- Web services and Restful APIs use the JSON format to get public data.
- JSON is a widely used file format for NoSQL databases such as MongoDB, Couchbase and Azure Cosmos DB.
- JSON is naturally the **raw data for "source of truth**", which is always needed,

Performance:

- JSON is lightweight text-based format in comparison to XML
- JSON as a simple but not so efficient format is very accessible
- JSON is **naturally the raw data** for "source of truth used for data **from web API and NoSQL databases**
- It is supported by all major big data query engines, such as Apache Hive and SparkSQL which can directly query JSON files

AVRO Format

- **Avro** is an row based, open-***source*** **schema** specification for data serialization that provides serialization and data exchange services for Apache Hadoop.
- Language-agnostic
- Rich data structure
- The Data is stored in a binary format making it compact and efficient.
- **Schema definition** is stored in JSON format making it easy to read and interpret.
- Supports Schema Evolution

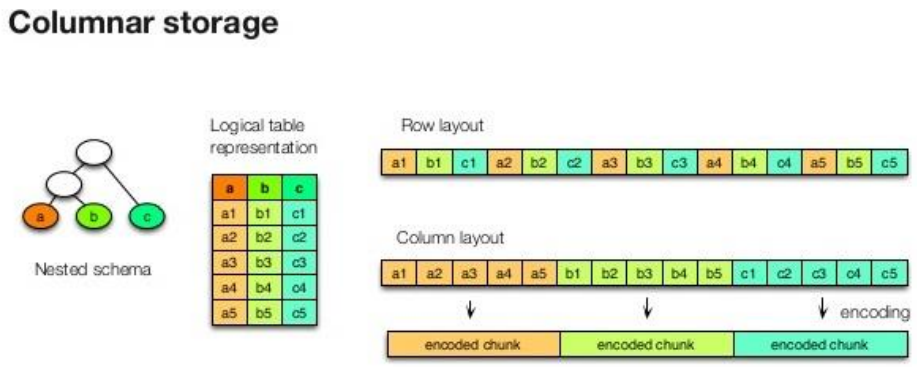
Application:

- **Avro** format is preferred for loading data lake landing because downstream systems can easily retrieve table schemas from files, and any source schema changes can be easily handled.
- Consider **Avro** file format in cases where your I/O patterns are more write heavy, or the query patterns favour retrieving multiple rows of records in their entirety.
- Works well with Event Hub or Kafka that write multiple events/messages in succession.

PARQUET

Parquet

- Columanar storage, used by Hadoop systems, such as Pig, Spark, and Hive
- Supports efficient data compression and encoding schemes that can lower data storage costs.
- It is cross platform and language independent format.
- It stores Metadata
- Services such as Azure Synapse Analytics, Azure Databricks and Azure Data Factory have native functionality that take advantage of Parquet file formats.



Performance:

- Unlike row-based file formats like CSV, Parquet is optimized for performance.
- You can focus only on the relevant data very quickly. Moreover, the amount of data scanned will be way smaller and will result in less I/O usage

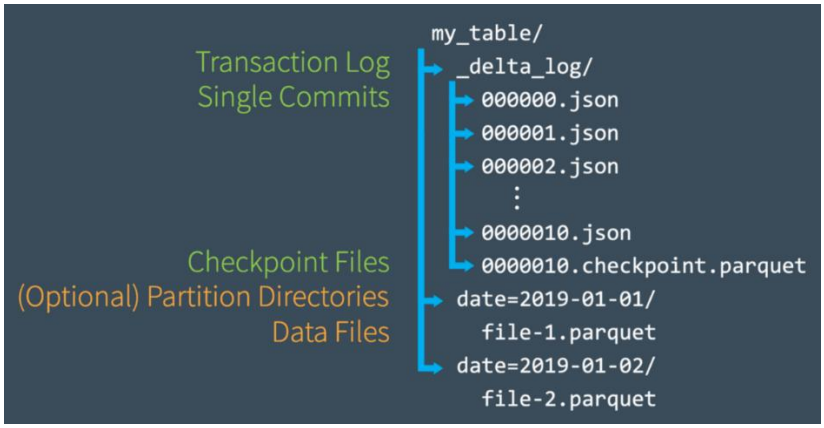
Note: *Consider Parquet when the I/O patterns are more read heavy or when the query patterns are focused on a subset of columns in the records.*

AVRO vs. PARQUET

1. AVRO is a row-based storage format, whereas PARQUET is a columnar-based storage format.
2. PARQUET is much better for analytical querying, i.e., reads and querying are much more efficient than writing.
3. Writing operations in AVRO are better than in PARQUET.
4. AVRO is much matured than PARQUET when it comes to schema evolution. PARQUET only supports schema append, whereas AVRO supports a much-featured schema evolution, i.e., adding or modifying columns.
5. PARQUET is ideal for querying a subset of columns in a multi-column table. AVRO is ideal in the case of ETL operations, where we need to query all the columns.

DELTA

- Delta is a data format based on Apache Parquet.
- It's an open source project (https://github.com/delta-io/delta), delivered with Databricks runtimes and it's the default table format from runtimes 8.0 onwards.
- You can use Delta format through notebooks and applications executed in Databricks with various APIs (Python, Scala, SQL etc.) and also with Databricks SQL.
- Delta is made of many components:
    - Parquet data files organized or not as partitions
    - Json files as transaction log
    - Checkpoint file



All of this is built on top of your data lake which can be hosted on AWS S3, Microsoft Azure DataLake, or Google Storage service.

- Delta is, like Parquet, a columnar oriented format. So, it's best fitted for analytic workloads.
- With Delta transaction log files, it provides ACID transactions and isolation level to Spark.