File Types and When to Use Them

Bonus: Quick Cheat Sheet

Choosing the right file format optimizes storage, speed, and tool compatibility. We'll explore common file types used in data engineering and their best use cases.

in
Abhishek Agrawal
Azure Data Engineer

CSV (Comma-Separated Values)

Category: Structured

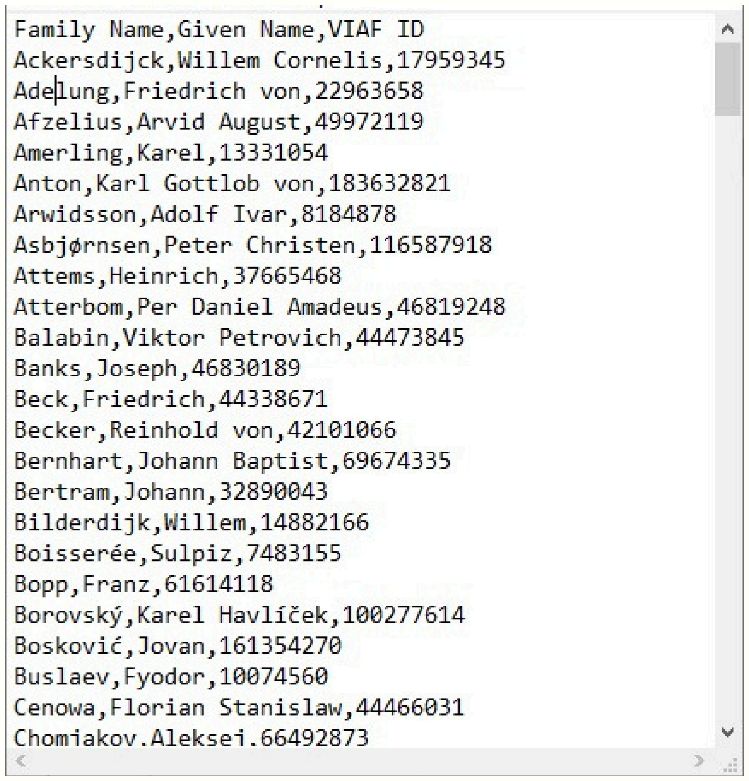
Best Used For: Small to medium-sized tabular data, easy export/import.

Description: A simple text file format where data is organized into rows and columns, separated by commas. It's widely used for data transfer between systems and applications.

When to Use:

- Lightweight datasets.
- Interoperability with many tools and applications.
- Quick data storage or transfer.

Example:



```
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bošković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov,Aleksei,66492873
```

Parquet

Category: Structured

Best Used For: Large-scale analytics, columnar data storage.

Description: A columnar storage file format optimized for big data processing. It provides efficient storage, compression, and faster query performance.

When to Use:

- Big data analytics.
- Data lakes and Hadoop/Spark environments.
- When performance and storage efficiency are critical.

Parquet is not human-readable; it is binary. Use with tools like Hadoop, Spark, or Databricks to read it.

JSON (JavaScript Object Notation)

Category: Semi-Structured

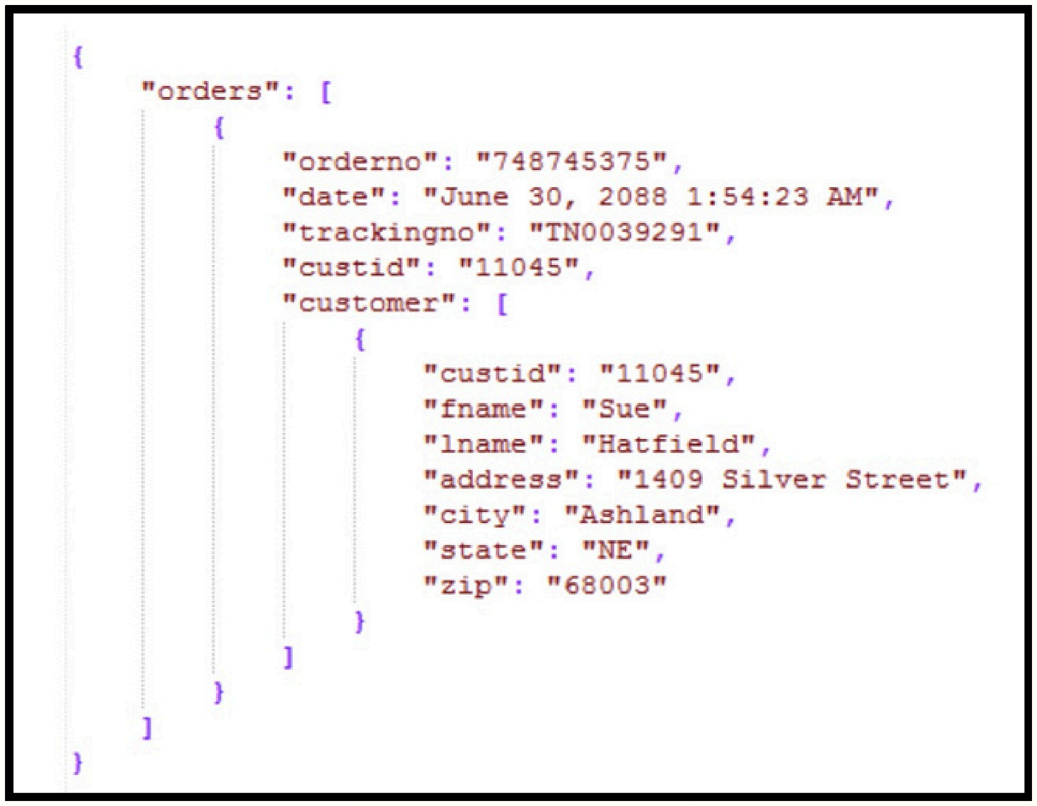
Best Used For: Hierarchical or nested data, data interchange (APIs, web services).

Description: A lightweight, text-based data format that represents data in a key-value pair structure. Often used for web APIs and config files.

When to Use:

- Data interchange between web services.
- Storing configuration files or logs.
- When dealing with nested data.

Example:



```
{
    "orders": [
        {
            "orderno": "748745375",
            "date": "June 30, 2088 1:54:23 AM",
            "trackingno": "TN0039291",
            "custid": "11045",
            "customer": [
                {
                    "custid": "11045",
                    "fname": "Sue",
                    "lname": "Hatfield",
                    "address": "1409 Silver Street",
                    "city": "Ashland",
                    "state": "NE",
                    "zip": "68003"
                }
            ]
        }
    ]
}
```

XML (Extensible Markup Language)

Category: Semi-Structured

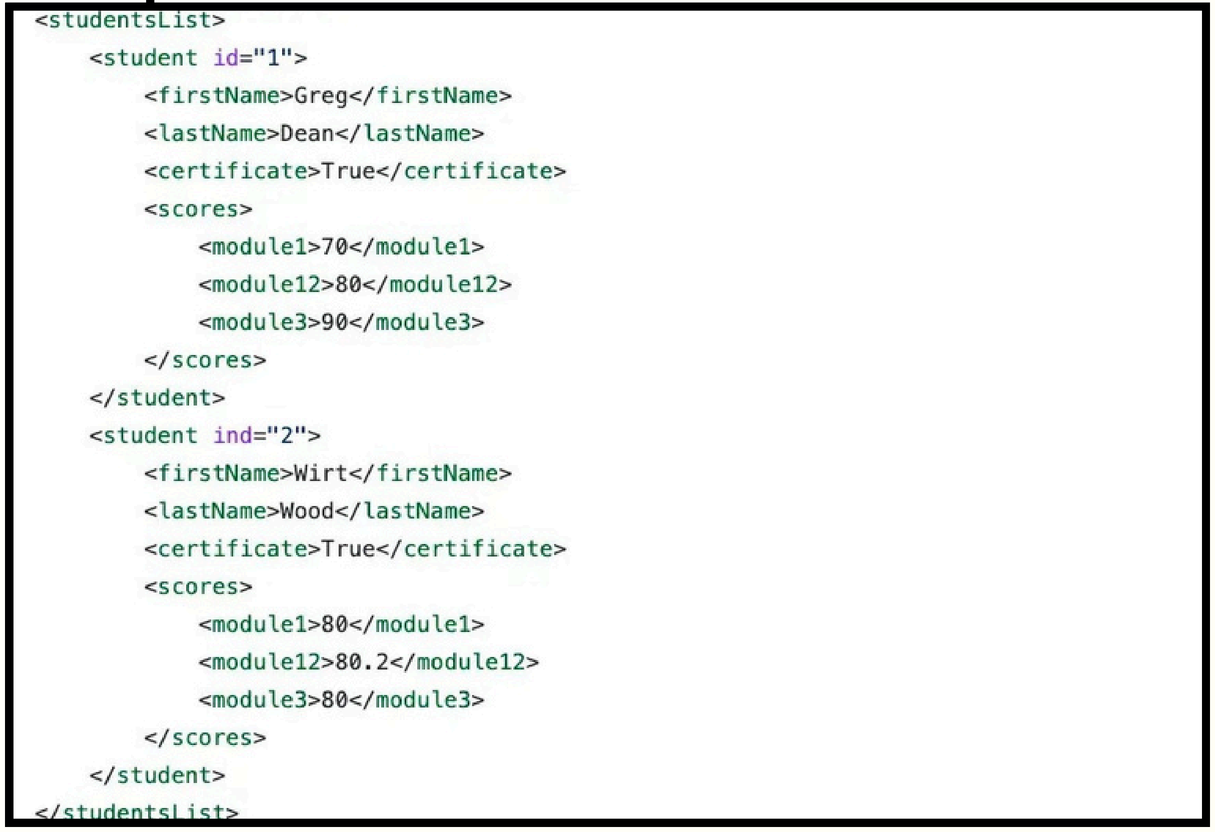
Best Used For: Complex documents, web services with metadata.

Description: A markup language that defines rules for encoding documents in a format that is both human-readable and machine-readable. It is commonly used for document storage and complex data structures.

When to Use:

- Legacy systems.
- Storing metadata-rich documents.
- Complex data with a predefined schema.

Example:



```xml
<studentsList>
    <student id="1">
        <firstName>Greg</firstName>
        <lastName>Dean</lastName>
        <certificate>True</certificate>
        <scores>
            <module1>70</module1>
            <module12>80</module12>
            <module3>90</module3>
        </scores>
    </student>
    <student ind="2">
        <firstName>Wirt</firstName>
        <lastName>Wood</lastName>
        <certificate>True</certificate>
        <scores>
            <module1>80</module1>
            <module12>80.2</module12>
            <module3>80</module3>
        </scores>
    </student>
</studentsList>
```

Avro

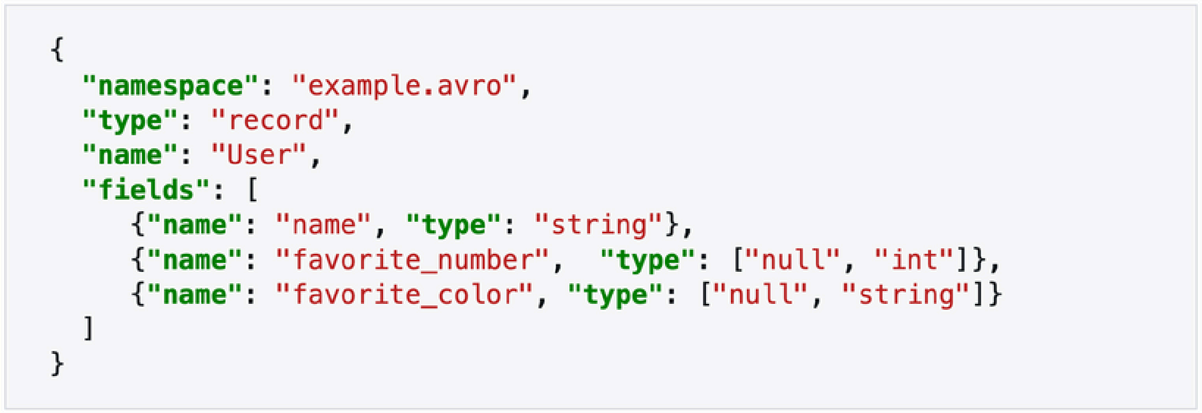
Category: Structured/Semi-Structured

Best Used For: Serialization format for data exchange, schema evolution.

Description: A binary file format that includes schema definition, supporting data serialization and schema evolution. Avro is widely used in big data pipelines.

When to Use:
Data streaming.
When you need efficient serialization with schema.
Interchanging data between systems, especially in big data platforms.

Avro files are binary and contain both data and schema. Ideal for streaming environments like Kafka. Avro schemas are defined using JSON. Schemas are composed of primitive types (null, boolean, int, long, float, double, bytes, and string) and complex types (record, enum, array, map, union, and fixed). Simple schema example:



```
{
  "namespace": "example.avro",
  "type": "record",
  "name": "User",
  "fields": [
     {"name": "name", "type": "string"},
     {"name": "favorite_number",  "type": ["null", "int"]},
     {"name": "favorite_color", "type": ["null", "string"]}
  ]
}
```

ORC (Optimized Row Columnar)

Category: Structured

Best Used For: High-performance analytics, especially in Hadoop.

Description: A columnar format optimized for read-heavy operations, providing high compression and fast read/write access. Commonly used with Hive for big data processing.

When to Use:

- Data lakes or Hadoop environments where performance is a priority.
- Storing large, write-heavy datasets.
- When columnar format is needed for faster analytics.

ORC is not human-readable. It's used in distributed systems for fast access.

Text File

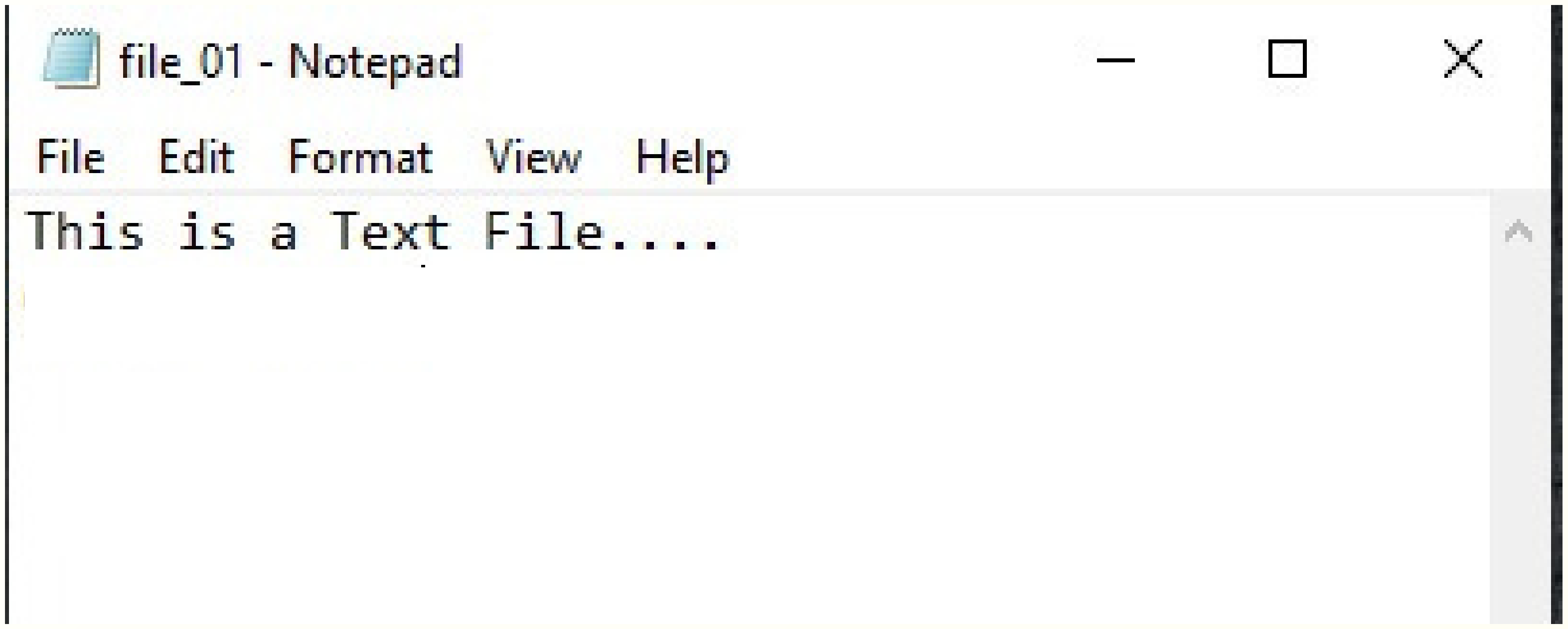
Category: Unstructured

Best Used For: Raw textual data, simple logs, or notes.

Description: A plain text file format used to store raw data. It doesn't enforce any structure, making it easy to write and store data in a human-readable form.

When to Use:
- Storing log files.
- Simple text-based documents.
- When structure is not necessary.

This is a plain text file. It contains raw, unstructured information.



```
This is a Text File....
```

Image/Video Files

Category: Unstructured

Best Used For: Multimedia content such as images, audio, and video.

Description: Non-text data formats used for storing multimedia content. Image files can be JPEG, PNG, while video files can be MP4, AVI.

When to Use:

- Storing and processing multimedia data.
- For media applications, machine learning involving image or video data.

Common formats: JPEG, PNG (Image), MP4, AVI (Video)

Cheat Sheet of File Types

File Type	Category	Best Used For	When to Use
CSV	Structured	Small to medium-sized tabular data.	Quick data transfer, lightweight data storage.
Parquet	Structured	Large-scale analytics, columnar data storage. Used in distributed systems for optimized querying.	Big data processing and efficient storage (Example: Data lakes, cloud storage for large datasets).
JSON	Semi-Structured	Hierarchical or nested data, data interchange (APIs, web services).	APIs, web services, storing complex objects or logs.
XML	Semi-Structured	Complex documents, web services with metadata.	Legacy systems, document storage, metadata-rich documents.
Avro	Structured/Semi-Structured	Efficient serialization with schema, particularly for streaming data in big data environments.	Real-time processing, schema evolution, data streaming.
ORC	Structured	High-performance analytics on big data platforms, like Hive.	Fast queries on large datasets, particularly with Hadoop.
Text	Unstructured	Raw textual data (logs, notes, etc.).	Text processing, logs, simple documents.
Image/Video	Unstructured	Multimedia content such as images and videos.	Storing and processing images, audio, and video data.

Follow for more content like this

Abhishek Agrawal

Azure Data Engineer