03 File Formats and Intro to Avro

An Evolution of data Comma Separated Values (CSV)

- First CSV:

rownum	column1	column2	column3	column4	column5	column6
row1	John	Doe	25	John.doe	true	OK
row2	Mary	Poppins	sixty	Mary.pop	yes	OK
row3	Tom	Cruise	45	Tom.Cru		

An Evolution of data Comma Separated Values (CSV)

- Advantages:
  - Easy to parse
  - Easy to read
  - Easy to make sense of

- Disadvantages:
  - The data types of elements has to be inferred and is not a guarantee
  - Parsing becomes tricky when data contains commas
  - Column names may or may not be there

An Evolution of data Relational tables definitions

- Relational table definitions add types:

```
CREATE TABLE distributors (
    did     integer PRIMARY KEY,
    name    varchar(40)
);
```

- Advantages:
  - Data is fully typed
  - Data fits in a table
- Disadvantages:
  - Data has to be flat
  - Data is stored in a database, and data definition will be different for each database

An Evolution of data JSON (JavaScript Object Notation)

- JSON format can be shared across the network!

```
{
        "id": "0001",
        "type": "donut",
        "name": "Cake",
        "image":
                {
                        "url": "images/0001.jpg",
                        "width": 200,
                        "height": 200
                },
        "thumbnail":
                {
                        "url": "images/thumbnails/0001.jpg",
                        "width": 32,
                        "height": 32
                }
}
```

An Evolution of data JSON (JavaScript Object Notation)

- JSON format

- Advantages:
  - Data can take any form (arrays, nested elements)
  - JSON is a widely accepted format on the web
  - JSON can be read by pretty much any language
  - JSON can be easily shared over a network
- Disadvantages:
  - Data has no schema enforcing
  - JSON Objects can be quite big in size because of repeated keys

An Evolution of data AVRO

- Avro is defined by a schema (schema is written in JSON)

- To get started, you can view Avro as JSON with a schema attached to it

```
{
    "type": "record",
    "name": "userInfo",
    "namespace": "my.example",
    "fields": [{
            "name": "username",
            "type": "string",
            "default": "NONE"
        },
        {
            "name": "age",
            "type": "int",
            "default": -1
        },
        {
            "name": "address",
            "type": {
                "type": "record",
                "name": "mailing_address",
                "fields": [{
                        "name": "street",
                        "type": "string",
                        "default": "NONE"
                    },

                    {
                        "name": "city",
                        "type": "string",
                        "default": "NONE"
                    }
                ]
            },
            "default": {}
        }
    ]
}
```

An Evolution of data AVRO

- Advantages:
  - Data is fully typed
  - Data is compressed automatically (less CPU usage)
  - Schema (defined using JSON) comes along with the data
  - Documentation is embedded in the schema
  - Data can be read across any language
  - Schema can evolve over time, in a safe manner (schema evolution)

- Disadvantages:
  - Avro support for some languages may be lacking (but the main ones is fine)
  - Can't "print" the data without using the avro tools (because it's compressed and serialised)

Suggested Readings:

1. A Detailed Introduction to Avro Data Format: Schema Example
2. What is Apache Avro?: A Guide to the Big Data File Format | Airbyte | Airbyte