ARTIFICIAL INTELLIGENCE SOFTWARE DEVELOPMENT

Week 6 Lecture 1

Dr. Hari M Koduvely

Agenda for Today

- Theory:
  - Fundamentals of Data Engineering *:*
  - Breakout session on Data Model & Data Pipeline for your project

Fundamentals of Data Engineering

Data Engineering refers to building systems for collecting, storing and processing large quantities of user and system generated data.

Data Sources

- An AI system may use data from different sources having different properties
- Understanding sources and properties of data can help in its efficient usage

Data Sources

- User Input Data:
  - Text, Videos, Uploaded files etc.
  - Quite often not in the correct format.
    - E.g. Text instead of numbers, uploaded file with wrong extn etc.
  - Needs more validation checking and processing.
  - Often needs to process the data immediately and quickly

Data Sources

- System Generated Data:
  - System logs, model predictions etc.
  - Usually well formatted, hence requires less checking
  - Large volume of data
    - Important signals can be buried in the noise
    - Storage could be expensive
  - Typically processed in batches
  - System generated data can include user behavior
    - What user clicked, moved, sent etc.
    - Considered as part of user data
    - Subjected to privacy regulations such as GDPR

Data Sources

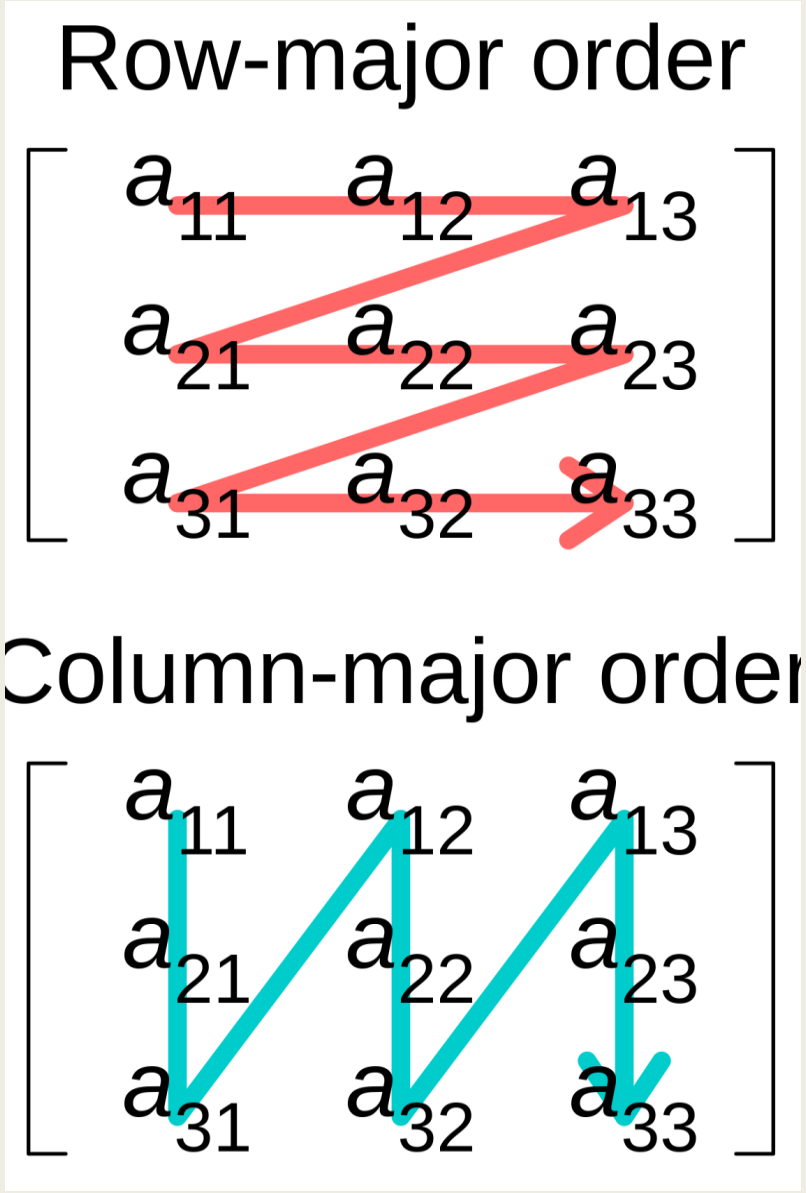
- Internal Databases:
  - Generated by various services and enterprise applications.
  - E. g. Inventory, Customers, Assets
  - Need to be safeguarded properly

Types of Data

- First Party Data: Data collected by a company from their users
- Second Party Data: Data collected by a second company from their customers and sold to first company
- Third Party Data: Data collected from the public domain

Data Formats

- Row-Major Format: Consecutive elements in a row are stored next to each other in memory.
  - E. g. CSV format
  - Useful for Transactional Processing
  - Allow faster data writes
- Column-Major Format: Consecutive elements in a column are stored next to each other in memory.
  - E. g. Parquet format
  - Useful for Analytics Processing (e. g. finding mean value of a field)
  - Allow faster reading of a column
  - Pandas DF is column-major
  - Numpy ndarray is row-major by default but order can be specified



Data Serialization

The process of converting a data structure or object state into a format that can be stored or transmitted and reconstructed later.

Points to consider for deciding serialization format

- ❑ Human readability
- ❑ Access patterns
- ❑ Size of data

Data Serialization

The process of converting a data structure or object state into a format that can be stored or transmitted and reconstructed later.

Major serialization formats:

Format	Binary/Text	Human-readable	Example use cases
JSON	Text	Yes	Everywhere
CSV	*Text*	*Yes*	*Everywhere*
Parquet	*Binary*	*No*	*Hadoop, Amazon Redshift*
Avro	Binary primary	No	Hadoop
Protobuf	Binary primary	No	Google, TensorFlow (TFRecord)
Pickle	Binary	No	Python, PyTorch serialization

Image source: Designing Machine Learning Systems – Chip Guyan

Data Models

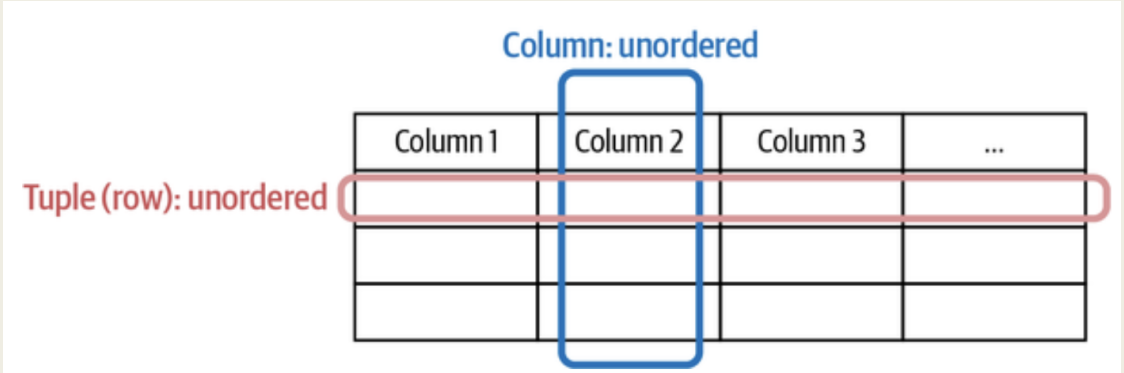
Data models describe how data is represented.

Some important types of Data Models are

- ❑ Relational Model
- ❑ NoSQL Model
  - *Document Model*
  - *Graph Model*

Data Models

- Relational Model
  - One of the oldest and persistent models
  - Invented by Edgar F. Codd in 1970
  - Data is organized into relations
  - Each relation is a set of tuples
  - A relation can be represented as a table



Data Models

- Relational Model
  - If any attribute is changed multiple rows needs to be updated
  - e.g. Change of price
  - Useful to convert this into a normalized form

Title	Author	Format	Publisher	Country	Price
Harry Potter	J.K. Rowling	Paperback	Banana Press	UK	$20
Harry Potter	J.K. Rowling	E-book	Banana Press	UK	$10
Sherlock Holmes	Conan Doyle	Paperback	Guava Press	US	$30
The Hobbit	J.R.R. Tolkien	Paperback	Banana Press	UK	$30
Sherlock Holmes	Conan Doyle	Paperback	Guava Press	US	$15

Data Models

- Relational Model – Normalized Form

Title	Author	Format	Publisher ID	Price
Harry Potter	J.K. Rowling	Paperback	1	$20
Harry Potter	J.K. Rowling	E-book	1	$10
Sherlock Holmes	Conan Doyle	Paperback	2	$30
The Hobbit	J.R.R. Tolkien	Paperback	1	$30
Sherlock Holmes	Conan Doyle	Paperback	2	$15

Publisher ID	Publisher	Country
1	Banana Press	UK
2	Guava Press	US

Data Models

- Relational Model – Normalized Form

Title	Author	Format	Publisher ID	Price
Harry Potter	J.K. Rowling	Paperback	1	$20
Harry Potter	J.K. Rowling	E-book	1	$10
Sherlock Holmes	Conan Doyle	Paperback	2	$30
The Hobbit	J.R.R. Tolkien	Paperback	1	$30
Sherlock Holmes	Conan Doyle	Paperback	2	$15

Publisher ID	Publisher	Country
1	Banana Press	UK
2	Guava Press	US

- In a normalized form data is stored across multiple tables
- Joining multiple tables can be computationally expensive

Data Models

- Relational Model – Query Language
  - SQL – Most popular query language
  - Declarative language

Data Models

- Relational Model – Query Language
  - SQL – Most popular query language
  - Declarative language:
    - you specify the outputs you want, and the computer figures out the steps needed to get you the queried outputs.
  - Imperative language:
    - you specify the steps needed for an action and the computer executes these steps to return the outputs

Data Models

- Question: Name some popular relational databases?

Data Models

- Answer: Some popular relational databases
  - MySQL
  - PostgreSQL
  - Oracle
  - IBM DB2
  - Mycrosoft SQL Server

Data Models

- NoSQL (Not Only SQL) Data Model
  - Relational Data Model demands that data follows a strict schema
  - Schema management can be painful
  - Two main NoSQL Data Models:
    - Document Model
    - Graph Model

Data Models

- Document Type NoSQL Data Model
  - Built around the concept of a "Document"
  - A "Document" is a single continuous string
  - Encoded as JSON, XML or BSON
  - All documents in a database are assumed to be encoded by the same format
  - Each document is represented by a unique key
  - Analogy with relational tables:
    - A single document is similar to a row
    - Collection of documents is similar to a table

Data Models

- Example of Document Type NoSQL Data Model

Example 3-1. Document 1: harry_potter.json

```
{
  "Title": "Harry Potter",
  "Author": "J .K. Rowling",
  "Publisher": "Banana Press",
  "Country": "UK",
  "Sold as": [
    {"Format": "Paperback", "Price": "$20"},
    {"Format": "E-book", "Price": "$10"}
  ]
}
```

Example 3-2. Document 2: sherlock_holmes.json

```
{
  "Title": "Sherlock Holmes",
  "Author": "Conan Doyle",
  "Publisher": "Guava Press",
  "Country": "US",
  "Sold as": [
    {"Format": "Paperback", "Price": "$30"},
    {"Format": "E-book", "Price": "$15"}
  ]
}
```

Example 3-3. Document 3: the_hobbit.json

```
{
  "Title": "The Hobbit",
  "Author": "J.R.R. Tolkien",
  "Publisher": "Banana Press",
  "Country": "UK",
  "Sold as": [
    {"Format": "Paperback", "Price": "$30"},
  ]
}
```

Data Models

- Document Type NoSQL Data Model – Advantages
  - Schema-less
  - Faster creation and read
  - Better locality
- Document Type NoSQL Data Model – Disadvantages
  - Difficult to make analytics computations
  - Consistency limitations
  - Atomicity weakness

Data Models

- Question: Name some popular NoSQL document databases?

Data Models

- Answer: Some popular NoSQL document databases

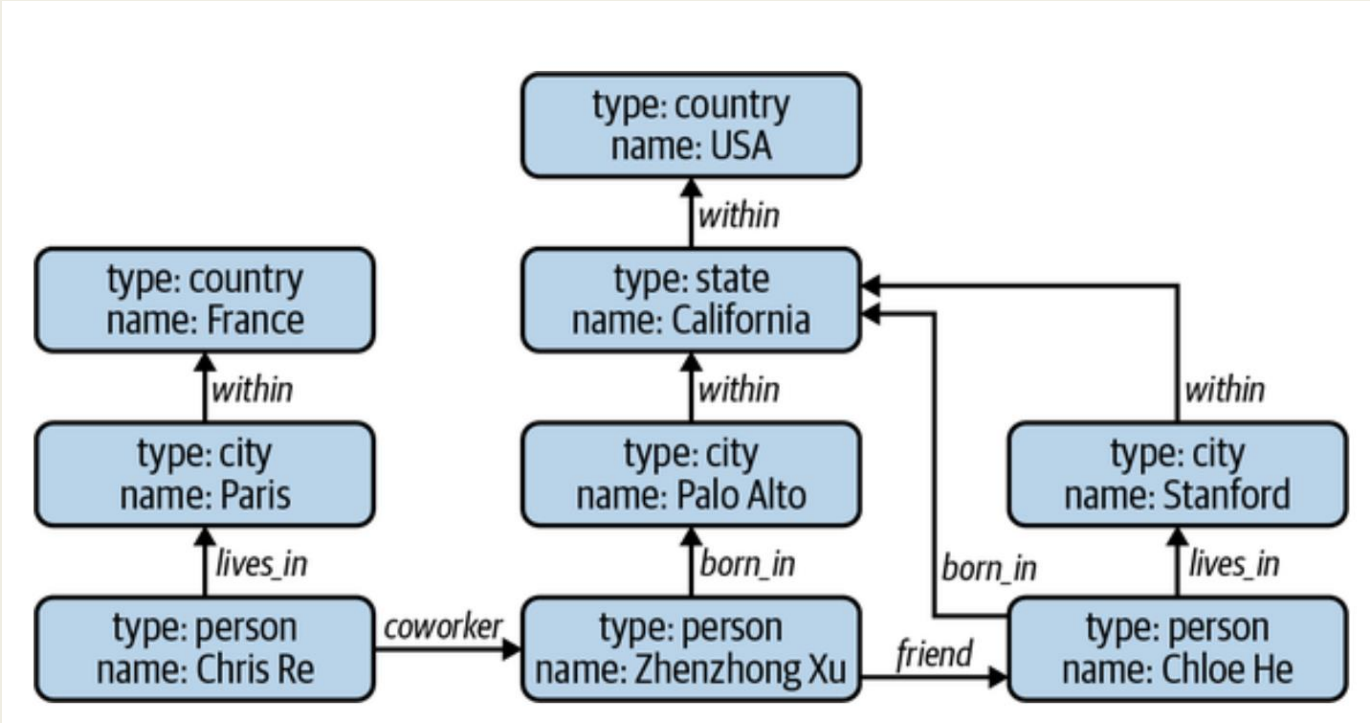
  - MongoDB
  - Amazon DocumentDB
  - CouchDB

Data Models

- Graph Type NoSQL Data Model
  - Built around the concept of a "Graph"
  - A "Graph" consists of Nodes and Edges
  - Edges represent relationship between nodes
  - Importance:
    - Document Data Model: content of each data item
    - Graph Data Model: relationship between each data item

Data Models

- Graph Type NoSQL Data Model – Example



- Graph databases are queried using Graph Query Languages:

- Homework exercise: read further information on GQL from https://en.wikipedia.org/wiki/Graph_Query_Language

Data Models

- Question: Name some popular Graph databases?

Data Models

- Answer: Some popular Graph databases

  - Neo4J
  - OrientDB
  - Amazon Neptune

Data Processing Types

Databases are optimized for two types of workloads

- ❑ Transactional Processing
- ❑ Analytical Processing

Transactional Processing

- Transaction refers to any kind of action:
  - Ordering a Pizza
  - Uploading a YouTube video
  - Booking an Uber Cab
- Transactions are inserted into a DB as they are generated
- Occasionally transactions are updated if some fields are changed
- Also known as Online Transaction Processing or OLTP

Transactional Processing

- Transactional databases are designed to satisfy **ACID** properties:
  - **Atomicity**:
    - To guarantee that all the steps in a transaction are completed successfully as a group.
    - If any step in the transaction fails, all other steps must fail also
    - e. g. If user payment fails, entire Uber booking transaction is cancelled.
  - **Consistency**:
    - To guarantee that all the transactions coming through must follow predefined rules.
    - e. g. Only authorized users can book an Uber cab

Transactional Processing

- Transactional databases are designed to satisfy **ACID** properties:
  - Isolation:
    - To guarantee that two transactions happen at the same time as if they were isolated.
    - e. g. Two passengers cannot book the same Uber cab at the same time.
  - Durability:
    - To guarantee that once a transaction has been committed, it will remain committed even in the case of a system failure.
    - e. g. after you've ordered a ride and even if you lost network connection, you still want your ride to come.

Transactional Processing

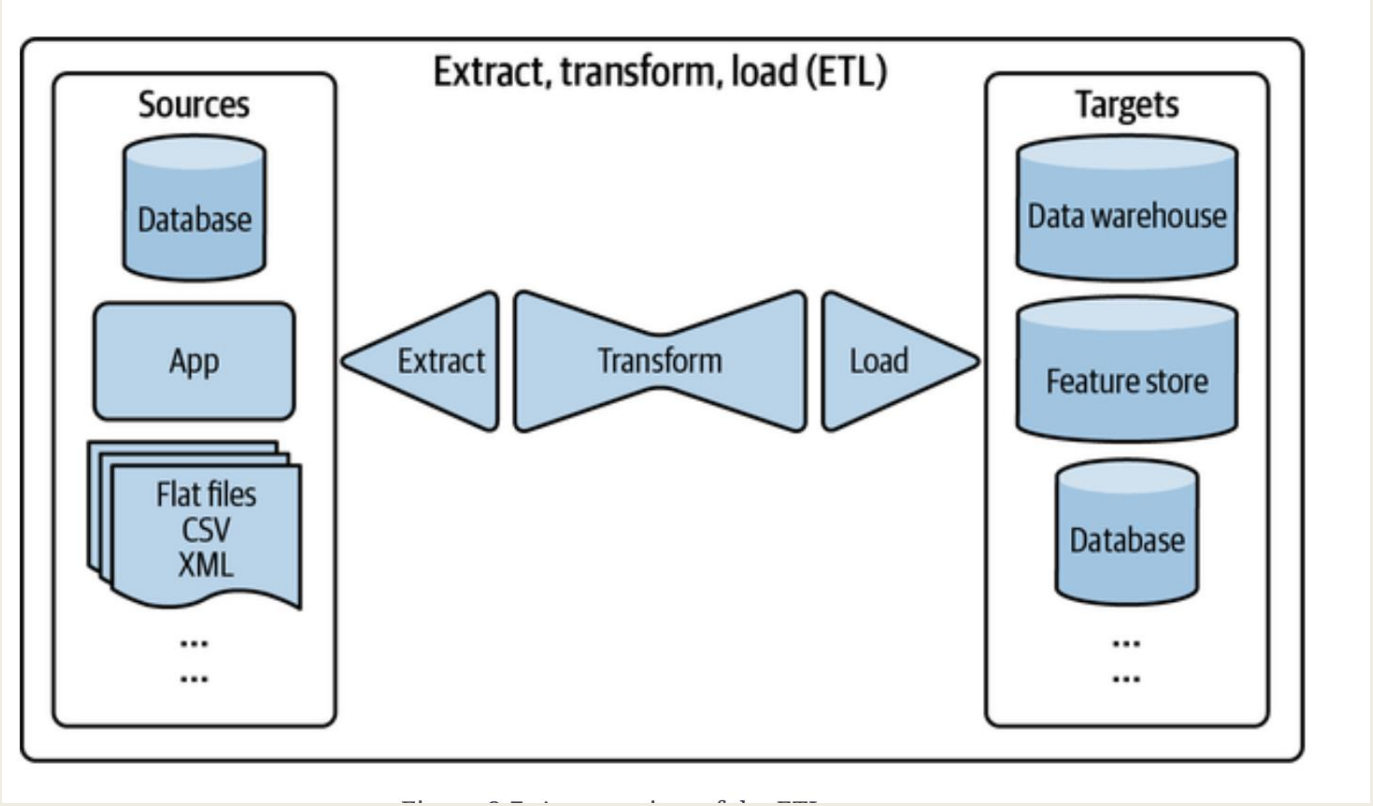
- ACID Requirements can be too restrictive for many use cases
- Instead, BASE requirements are used in many situations
- BASE - **B**asically **A**vailable, **S**oft state, and **E**ventual consistency

Analytical Processing

- These databases are optimized for making aggregations of a column across rows
  - *e. g. To find average price of an Uber ride in a month at a particular city*
- Also known as ***Online Analytics Processing or OLAP***

Extract, Transform and Load (ETL)

- Data is first **E**xtracted from different sources
- It is then **T**ransformed into the desired Schema
- Finally **L**oaded to the target database



Breakout Session

- ❑ Discuss what type of data processing (Transactional or Analytics) is the major type in your project
- ❑ Based on this arrive at what type of database you should be using for your project
- ❑ Also what type of data pipeline your project would be required