Healthcare Data and Interoperability

Data types, the HL7 and FHIR standards, data quality, and automated pipelines.

This guide accompanies Module 2 and can be read independently of the lectures. It develops the four themes of the module, the types of healthcare data, the standards that make data exchangeable, the assessment of data quality, and the construction of automated data pipelines, in greater depth than the slides, with worked examples, short pieces of code, and citations to peer-reviewed literature and to the relevant standards.

The unifying message is that reliable automation begins with well-structured, standardised, quality-checked data. The lack of interoperability between health information systems has been shown to reduce the quality of care and to waste resources, which is why standards and data-quality practices are not administrative overhead but preconditions for safe automation [1]. Each section below builds toward the final one, in which the pieces are assembled into a pipeline.

A note on scope: this module concentrates on structured, tabular data and on the FHIR standard, because that is where dependable automation most readily begins. Unstructured data such as free-text notes is introduced, together with the natural-language-processing methods used to handle it, but a full treatment of clinical text mining is beyond the module and is signposted for further study.

Learning outcomes

After working through this guide you will be able to:

- Distinguish structured, unstructured, imaging, and signal data, and the formats each uses.
- Explain the role of natural language processing in extracting value from unstructured clinical text.
- Explain what interoperability is and how HL7 and FHIR support it.
- Read a FHIR resource and explain why its structure aids automation.
- Assess the quality of a dataset along recognised dimensions and express checks as rules.
- Describe the stages of an automated data pipeline and what makes one dependable.
- Relate the module's ideas to a published clinical data pipeline.

1. The kinds of healthcare data

1.1 Four types

Healthcare data is unusually varied, and recognising its type is the first step in judging how easily it can be automated. Four broad types recur. Structured data sits in labelled rows and columns, as laboratory results do, and is the most tractable for automation because a program can read each value directly. Unstructured data, such as free-text clinical notes and discharge summaries, carries its meaning in language and must be processed before its values can be used. Imaging data, such as radiographs and scans, consists of large binary files with their own containers. Signal data, such as electrocardiograms, records continuous measurements over time. Most of this course works with structured data, but recognising the whole landscape prevents you from underestimating a task that involves the harder types.

1.2 Formats you will open

Each type arrives in characteristic formats. Comma-separated values (CSV) files hold plain rows and columns and are a sensible default for structured data. JavaScript Object Notation (JSON) stores nested key-value data and is what many modern web services, including FHIR interfaces, return. DICOM is the standard container for medical images, bundling the picture with information about the patient and the acquisition. HL7 version 2 messages, compact lines of coded text, still move much of the data exchanged between clinical systems. The practical consequence, stressed throughout the module, is that the same logical task can demand very different effort depending on the format the data arrives in: a clean CSV may be read in a single line of code, while the same information trapped in a scanned PDF may require substantial extra work.

1.3 Structured versus unstructured data

The single most useful distinction for automation is between structured and unstructured data. Structured data has labelled, consistent fields, so a computer can read a value directly, rules are easy to write, and automation is straightforward. Unstructured data carries its meaning in language, so before you can act on it you must extract the values, the rules are harder and less reliable, and automation needs additional tooling. A common and valuable first step in a real project is therefore to convert unstructured input into structured data, so that everything downstream becomes easier.

1.4 Unstructured text and natural language processing

Because so much clinical information exists only as free text, natural language processing (NLP) has become an active field within health informatics. A systematic review by Hossain and colleagues, covering more than a hundred studies, found that electronic health records were the most common data source and that the data were primarily unstructured, with tasks including note classification, clinical entity recognition, and information extraction [7]. A methodological introduction by Clay and colleagues walks the clinical researcher through the whole NLP pipeline, from pre-processing to statistical and neural methods, and notes that, despite its promise, relatively few applications have yet entered routine clinical practice [8]. A further systematic review of NLP used to populate clinical registries found that rule-based methods initially dominated before machine-learning approaches gained ground, and cautioned that the diversity of clinical text limits the generalisability of any single method [9]. For this course, the practical lesson is that unstructured data is valuable but costly to automate, and that turning it into reliable structured data is itself a demanding task best approached with realistic expectations.

1.5 Codes and terminologies

Structured records are more valuable still when their values are coded using standard terminologies. Coding systems such as LOINC for laboratory tests and SNOMED CT for clinical concepts give each concept a fixed identifier, so that the same measurement means the same thing across institutions, and automation rules can key off a stable code rather than interpret variable free text. This standardisation is one of the interoperability requirements repeatedly identified in the literature as necessary for exchanging data between heterogeneous systems [1].

Key idea. The type and format of data largely determine the effort automation requires. Prefer structured, coded sources; treat unstructured text as valuable but costly, and expect to invest real effort in turning it into reliable structured data.

2. Interoperability and the HL7 and FHIR standards

2.1 What interoperability means

Interoperability is the ability of different systems to exchange data and to interpret it correctly, so that a value sent by one system means the same to another. The definition has two halves, and the second is the harder one: moving bytes between systems is straightforward, but agreeing on what those bytes mean, their units, their codes, and the identity of the patient they describe, is where the real difficulty lies. Semantic interoperability, in which meaning as well as structure is shared, is the most robust goal.

2.2 The problem standards solve

Without shared standards, connecting systems requires a bespoke translator for every pair, an approach whose cost grows unsustainably as systems are added. A systematic review of interoperability in heterogeneous health information systems identifies HL7, FHIR, CDA, SNOMED CT and related standards as the core requirements for solving this problem, and argues that semantic interoperability is the most robust objective [1]. The answer a standard provides is to agree once on a shared format and vocabulary, so that any conforming system can exchange data with any other, without a new connector each time.

2.3 HL7 and FHIR

Health Level Seven International (HL7) has published healthcare messaging standards for decades. Its version 2 messages remain widely deployed, but the newer Fast Healthcare Interoperability Resources (FHIR) standard has become the leading choice for new work. FHIR represents health data as resources, typically expressed in JSON, and exchanged over the web using ordinary web technologies. In its foundational description, FHIR was introduced as an agile, RESTful approach designed to address the complexity and implementation difficulties of earlier HL7 versions [2]. More recent reviews document its rapid adoption by national health systems and its role in modern digital-health and analytics infrastructure, while noting persisting challenges around implementation consistency, workforce training, and security [3].

2.4 Reading a FHIR resource

FHIR is built from a few simple pieces: a resource is a single thing such as a patient or an observation; a field is a named piece of data inside it; a reference is a link from one resource to another; and a bundle is a collection of resources sent together. A small, simplified observation resource illustrates the structure:

```
{
  "resourceType": "Observation",
  "status": "final",
  "code": { "text": "Glucose" },
  "valueQuantity": {
    "value": 5.4,
    "unit": "mmol/L"
  }
}
```

A FHIR observation in JSON. The resourceType announces what it is; status marks it final; and valueQuantity keeps the value and its unit together.

Reading this resource, the resourceType field tells any system what to expect, the status field marks the result as confirmed, the code names the measurement, and the valueQuantity block keeps the value and its unit together, removing a whole class of unit errors. Because it is plain JSON, it can be read in a few lines of code in any language.

2.5 Why FHIR helps automation

For the purposes of automation, the decisive property of FHIR is predictability. Because every resource is self-describing and follows a consistent structure, with values and their units kept together, a program written for one FHIR source usually works, with little change, for another. This is why the standard features throughout the course and why it underpins the pipeline built in the final lesson of the module.

> **Key idea.** Standards replace an unscalable web of one-off connectors with a shared language. FHIR's consistent, self-describing, web-friendly resources make healthcare data far more amenable to automation than ad hoc formats.

3. Data quality and validation

3.1 Quality as measurable dimensions

Even standardised data can be wrong, and in healthcare a flawed value can propagate into an analysis or a decision. Data quality is best treated not as a vague aspiration but as a set of measurable dimensions. Systematic reviews of electronic health record data quality consistently identify completeness, correctness, concordance, plausibility, and currency as the dimensions most often assessed, with completeness the most common of all [5]. An influential methodological account frames the core domains as accuracy, completeness, consistency, credibility, and timeliness, and recommends concrete appraisal methods including validation with data rules and comparison against manual review [6].

3.2 Common problems

A small number of problems recur. Missing values are empty cells where a value should be, and they quietly break calculations and models. Duplicates are the same record entered more than once, which distorts counts and averages. Unit mismatch means one column holds values in different units, so comparisons and sums become meaningless. Impossible values are readings outside any plausible range, such as a negative age. Each is common, each is dangerous in healthcare, and each can be caught by a simple automated check.

3.3 Expressing checks as code

Because these dimensions are concrete, they can be tested automatically. The literature shows a clear trend toward operationalising and automating data-quality assessment, though it also cautions that current tools remain fragmented and that standardised, theory-backed approaches are still needed [4]. A small validation snippet shows the idea:

```
import pandas as pd
data = pd.read_csv("results.csv")
```

```
missing = data["glucose"].isna().sum()
impossible = (data["glucose"] < 0).sum()

print("missing:", missing,
      "impossible:", impossible)
```

A minimal validation check: two rules, run on every record, producing a clear report. Real checks add units, ranges, and duplicate detection.

The essential discipline is to express quality expectations as explicit rules, completeness, format, range, and consistency checks, and to run them on every load of the data rather than once by hand.

3.4 Reject, flag, or fix

A crucial design decision concerns what to do when a check fails. Options are to reject a clearly invalid record, to flag a borderline value for human review, or to correct it where a safe and explicit rule exists. The safe default in a clinical setting is to flag and report rather than to alter data silently, because an invisible automatic correction can cause more harm than the original error.

Practical rule. Validate on every run, not once. Express each quality expectation as a rule the machine checks automatically, log every failure, and prefer flagging problems for human judgement over silently changing clinical values.

4. Building an automated data pipeline

4.1 What a pipeline is

A data pipeline is an ordered, automated sequence that moves data from a source to a finished output, applying cleaning and transformation along the way, an instance of the extract, transform, load (ETL) pattern familiar from data engineering. In healthcare, pipelines increasingly convert heterogeneous clinical data into standardised forms for research.

4.2 Four stages

A dependable pipeline separates its work into stages, each with a single responsibility: read the source, clean and validate, transform and standardise, and write the result. Keeping the stages separate keeps the pipeline understandable and testable; when something goes wrong, you know which stage to inspect.

4.3 A skeleton

In outline, each stage is a small function, and a short main step calls them in order:

```
import pandas as pd

def read():      return pd.read_csv("in.csv")
def clean(df):   return df.dropna()
def transform(df): return df   # standardise units here
def write(df):   df.to_csv("out.csv", index=False)

write(transform(clean(read())))
```

A pipeline skeleton. Each stage is a separate, testable function; the final line runs them in order.

4.4 What makes a pipeline dependable

Three further properties turn a working script into a system that can be scheduled and trusted. Configuration keeps file names and settings in one place. Logging records what each run did, so that an unattended run can be checked afterwards; a pipeline that silently drops half its rows because of a units problem is dangerous precisely because the loss is invisible without a log. Error handling makes the pipeline fail clearly and safely rather than produce a plausible but wrong output. A guiding safety rule is never to overwrite the source data, so that the pipeline can always be re-run from clean input. A related ordering rule is to standardise units before any aggregation, or the resulting numbers will be quietly wrong.

4.5 A published example

These principles are visible in the research literature. A validated Python-based pipeline that transforms Medicare claims into the OMOP common data model reported minimal data loss and a 99 percent pass rate across more than 1,500 data-quality checks, and its authors describe the result as scalable and reproducible [7]. The example illustrates how validation and pipelines belong together: the checks from the previous section are exactly what give a pipeline's output its credibility.

> **Key idea.** Give each pipeline stage one job; validate before you transform; standardise units before you aggregate; log every run; and never overwrite the source. These habits separate a demonstration from a pipeline that can run unattended.

5. Case studies and worked discussion

The themes of this module are visible in the research literature, and a few sources reward closer reading because they show the ideas working together on real data.

5.1 Interoperability in practice

The systematic review by Torab-Miandoab and colleagues surveys interoperability across heterogeneous health information systems and concludes that HL7, FHIR, CDA, SNOMED CT and related standards are the core requirements, with semantic interoperability, shared meaning rather than merely shared structure, as the most robust goal [1]. The FHIR-specific literature complements this: Bender and Sartipi introduced FHIR as an agile, RESTful response to the complexity of earlier HL7 versions [2], and more recent reviews document its rapid national adoption alongside persisting challenges in implementation consistency, workforce training, and security [3]. Read together, these sources explain both why FHIR has become the modern default and why adopting it well remains demanding.

5.2 The state of data-quality assessment

The data-quality reviews give a sobering picture that motivates the module's emphasis on validation. Lewis and colleagues, extending an earlier review, find that completeness remains the most assessed dimension but that no standard approach to assessment exists, and they call for scalable, flexible guidelines and greater automation [5]. Ozonze and colleagues, focusing on automated tooling, find that automation of quality assessment is gaining traction but remains fragmented and insufficiently grounded in theory [4]. Feder's methodological account provides the practical counterpart, recommending validation with data rules and comparison against manual review as concrete appraisal methods [6]. The combined message is that quality must be assessed deliberately and, increasingly, automatically, because it cannot be assumed.

5.3 A validated pipeline and the cost of unstructured data

Two further strands complete the picture. The OMOP pipeline of Lee and colleagues demonstrates that a Python-based ETL process can transform claims data into a common data model with minimal loss and a 99 percent pass rate across more than 1,500 quality checks, showing how validation and pipelines together produce trustworthy output [10]. Meanwhile, the natural-language-processing reviews temper expectations for unstructured data: Hossain and colleagues document the breadth of NLP tasks on largely unstructured records [7], Clay and colleagues note that few applications have reached routine practice [8], and Liu and colleagues find that the diversity of clinical text limits the generalisability of any single method [9]. The practical lesson is to prefer structured, coded sources, and to budget realistically when unstructured text cannot be avoided.

> **Key idea.** The literature shows structured, standardised, validated data as the foundation of trustworthy automation, and unstructured text as valuable but costly. Design toward the former and approach the latter with realistic expectations.

6. Further reading and practice

The four themes of this module, data types, interoperability, quality, and pipelines, are best consolidated by handling a small dataset end to end. The following exercises are designed to take an afternoon in total and to produce material you can reuse in the module assignment, which asks you to design a FHIR-based pipeline. Use only synthetic or openly published sample data; never use real patient data for practice.

For deeper background, the systematic reviews cited here are accessible entry points: the interoperability review [1] surveys the standards landscape, the data-quality reviews [4], [5], [6] catalogue the dimensions and tools of quality assessment, and the natural-language-processing reviews [7], [8], [9] map the methods used for unstructured text. The official FHIR specification [11] is the authoritative reference for the resources you will meet, and is worth browsing even if you do not read it in full.

- Obtain a small synthetic dataset of results and classify each column as structured, coded, or free text.
- Locate a sample FHIR observation and identify its resourceType, status, value, and unit fields.
- Write four validation rules for your dataset, one each for completeness, format, range, and consistency, and decide for each whether a failure should reject, flag, or fix the record.
- Sketch a four-stage pipeline for your dataset and mark where each validation rule runs.
- Read the abstract of the OMOP pipeline study [10] and note how its authors demonstrate the fidelity of their output.

Completing these exercises will leave you with a validated mental model of a data pipeline and a set of rules you can adapt directly for the assignment.

Key terms

Structured data — Data held in labelled rows and columns, such as laboratory results, that a program can read directly.

Unstructured data — Free-text or other data whose meaning is carried in language and must be extracted before use.

Natural language processing (NLP) — Computational methods for extracting structured information from unstructured clinical text.

Interoperability — The ability of different systems to exchange data and interpret it correctly, so that a value means the same to sender and receiver.

HL7 — Health Level Seven International, and the family of healthcare messaging standards it publishes, including version 2 messages.

FHIR — Fast Healthcare Interoperability Resources, a modern HL7 standard that represents health data as web-friendly resources, usually in JSON.

Data quality dimensions — Measurable properties of data such as completeness, correctness, consistency, plausibility, and timeliness.

Data pipeline (ETL) — An ordered, automated sequence that extracts data from a source, transforms it, and loads a finished result.

Self-check questions

1. For each of the four data types, give a healthcare example and state how easy it is to automate and why.
2. Explain why unstructured text is valuable but costly to automate, and what role NLP plays. Cite one finding from the reviews discussed.
3. Explain the problem that interoperability standards solve, and why building a separate connector for each pair of systems does not scale.
4. Read the FHIR observation on this page. Identify the fields that carry the measurement, its value, and its unit, and explain why keeping value and unit together matters.
5. Choose a dataset you know. Write one check for each of completeness, format, range, and consistency, and say what you would do when each fails.
6. Describe the four stages of a data pipeline and explain why validation belongs inside it rather than as a separate afterthought.
7. Why must a pipeline never overwrite its source data, and why must units be standardised before aggregation?
8. Summarise, with reference to the OMOP example, how validation gives a pipeline's output its credibility.

References

Peer-reviewed sources located via the Consensus academic search tool; standards cited from their issuing bodies. Formatted in IEEE style.

[1] A. Torab-Miandoab, T. Samad-Soltani, A. Jodati, and P. Rezaei-Hachesu, "Interoperability of heterogeneous health information systems: a systematic literature review," BMC Medical Informatics and Decision Making, vol. 23, 2023. [Online]. Available: https://consensus.app/papers/details/77be03c5df5c59389abd5de434b4c220/

[2] D. Bender and K. Sartipi, "HL7 FHIR: an agile and RESTful approach to healthcare information exchange," in Proc. 26th IEEE Int. Symp. Computer-Based Medical Systems (CBMS), 2013. [Online]. Available: https://consensus.app/papers/details/3c8ce6f3d6b756ca86b25d62742d60d7/

[3] A. Hornback et al., "FHIR in focus: enabling biomedical data harmonization for intelligent healthcare systems," IEEE Reviews in Biomedical Engineering, 2025. [Online]. Available: https://consensus.app/papers/details/8efc36fb991658b3a91a66ab65a2b668/

[4] O. Ozonze, P. J. Scott, and A. D. Hopgood, "Automating electronic health record data quality assessment," Journal of Medical Systems, vol. 47, 2023. [Online]. Available: https://consensus.app/papers/details/4b20fb5cfbf4560082a7425e1ec49ac8/

[5] A. Lewis et al., "Electronic health record data quality assessment and tools: a systematic review," Journal of the American Medical Informatics Association (JAMIA), 2023. [Online]. Available: https://consensus.app/papers/details/55e81cd6b9e25b4fbd4f19019aa140b3/

[6] S. L. Feder, "Data quality in electronic health records research: quality domains and assessment methods," Western Journal of Nursing Research, vol. 40, no. 5, 2018. [Online]. Available: https://consensus.app/papers/details/c83f14ccc465570f9939d2c3ed9ef0ab/

[7] E. Hossain et al., "Natural language processing in electronic health records in relation to healthcare decision-making: a systematic review," Computers in Biology and Medicine, 2023. [Online]. Available: https://consensus.app/papers/details/2e821434f8505295abae1541f1a48d45/

[8] B. Clay et al., "Natural language processing techniques applied to the electronic health record in clinical research and practice: an introduction to methodologies," Computers in Biology and Medicine, 2025. [Online]. Available: https://consensus.app/papers/details/94db86f6eb3a5aefb3db0436b77b3e60/

[9] L. Liu et al., "Using natural language processing to extract information from clinical text for populating clinical registries: a systematic review," Journal of the American Medical Informatics Association (JAMIA), 2025. [Online]. Available: https://consensus.app/papers/details/121d0e7cb2fe5b839e01da61422ae90e/

[10] Y. Lee et al., "Harmonizing Medicare claims data with OMOP: a validated ETL pipeline," AMIA Annual Symposium Proceedings, 2024. [Online]. Available: https://consensus.app/papers/details/d0f2047af4255a1e861f0aca1ee2e235/

[11] Health Level Seven International, "HL7 FHIR (Fast Healthcare Interoperability Resources)," standard specification. [Online]. Available: https://www.hl7.org/fhir/