Big Data Management

Exam-Focused Notes: Week 1

Lectures 1-3: Introduction, Knowledge Graphs & Semantic Web Technologies

Complete Beginner's Guide

Exam Preparation 2026

What This Document Covers

- ✓ Complete coverage of Lectures 1-3
- ✓ Simplified explanations for beginners
- ✓ Real-world examples and diagrams
- ✓ Exam-style questions and answers
- ✓ Key formulas and concepts highlighted
- ✓ Step-by-step worked examples

Contents

1 Lecture 1: Introduction to Big Data Management

1.1 The Basics: Data, Information, and Knowledge

Core Concept - Understanding the Hierarchy

Think of these three terms like cooking:

- **Data** = Individual ingredients (flour, eggs, sugar)
- **Information** = Recipe with measurements (2 cups flour, 3 eggs)
- **Knowledge** = Knowing how to bake (experience, technique)

1.1.1 Detailed Definitions

Term	Definition & Example
Data	**Definition:** Raw, unorganized facts and figures without context **Examples:** • Temperature: 25°C • Numbers: 42, 100, 5.7 • Text: “Apple”, “Dubai”, “Monday” • These mean nothing alone!
Information	**Definition:** Data that has been processed and organized to have meaning **Examples:** • “The temperature in Dubai today is 25°C” • “We sold 42 units in January” • “Apple stock price: $100” • Now we can understand the context!
Knowledge	**Definition:** Information combined with experience, rules, and understanding **Examples:** • “25°C in Dubai is pleasant, so more people will go out” • “Sales are down from last month, we need to run a promotion” • “Apple stock is rising, it might be a good time to invest” • This helps us make decisions!

Table 1: Data, Information, and Knowledge Hierarchy

1.1.2 The DIKW Pyramid

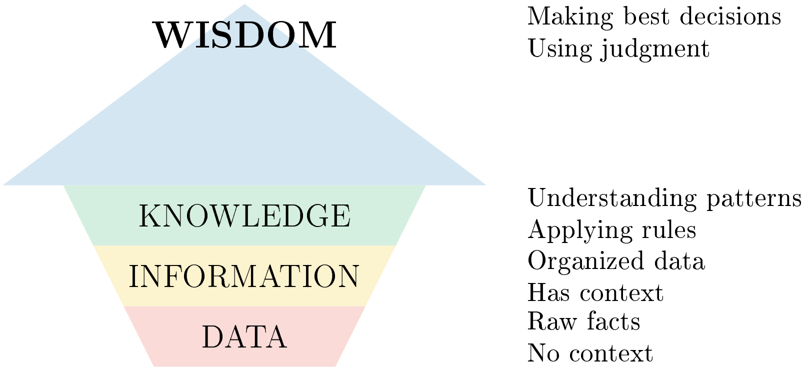


Figure 1: DIKW Pyramid (Ackoff 1989) - Each level builds on the one below

Real-World Example: Online Shopping

Data: User clicked "iPhone 15", time stamp: 10:30 AM, IP address: 192.168.1.1
Information: "John from Dubai browsed iPhone 15 at 10:30 AM on Monday"
Knowledge: "Users who browse iPhones on Monday mornings often purchase accessories. We should show them cases and chargers."
Wisdom: "Based on past patterns, send John a 10% discount on accessories within 2 hours for maximum conversion."

1.2 What is Big Data?

Exam-Important Definition

Big Data is data that is so large, fast, or varied that traditional databases cannot handle it efficiently.
Two Official Definitions:

1. **Laney (2001):** "High-volume, velocity, and variety information assets demanding cost-effective, innovative processing for enhanced insight."
2. **Manyika (2011):** "High volume, velocity, and/or variety information requiring new processing forms for enhanced decision making."

1.2.1 The 5 V's of Big Data (EXAM CRITICAL)

V	Meaning	Simple Example
VOLUME	**Size of data** How much data exists? Think: Terabytes, Petabytes, Exabytes	YouTube: 500 hours of video uploaded *every minute* Facebook: 4 petabytes of data *per day* Your phone: A few gigabytes
VELOCITY	**Speed of data** How fast is data created and processed? Think: Real-time	Twitter: 6,000 tweets per second Stock market: Prices change every millisecond Traffic sensors: Data every second

V	Meaning	Simple Example
VARIETY	**Types of data** Is it structured, semi-structured, or unstructured? Think: Different formats	**Structured:** Excel tables, databases **Semi-structured:** JSON, XML files **Unstructured:** Videos, images, tweets
VERACITY	**Quality/Trustworthiness** Can we trust this data? Think: Accuracy	Is this Twitter account real or a bot? Are sensor readings accurate? Is this review fake?
VALUE	**Usefulness** Can we extract insights? Think: Business value	Can we predict customer behavior? Can we prevent diseases? Can we optimize delivery routes?

Table 2: The 5 V's of Big Data - Know These for the Exam!

Exam-Style Question & Answer

Question: A hospital collects patient heart rate data every second from 10,000 patients. The data includes text notes from doctors, X-ray images, and structured records. However, 5% of sensors malfunction. Identify which of the 5 V's apply and explain why.

Answer:

- **Volume:** $10,000 \times 86,400 = 864$ million readings per day
- **Velocity:** Data collected every second = real-time processing needed
- **Variety:** Multiple types: structured (records), semi-structured (notes), unstructured (images)
- **Veracity:** 5% sensor malfunction = data quality issues
- **Value:** Can predict heart attacks, improve treatment (high value)

1.3 Data Science vs Big Data Management

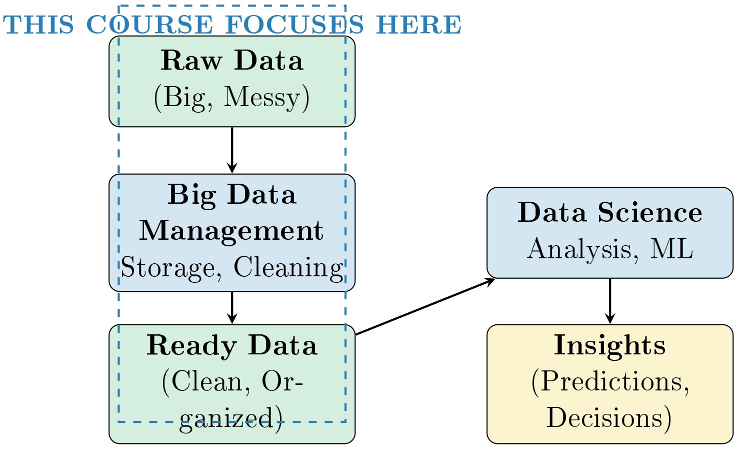


Figure 2: Big Data Management vs Data Science Pipeline

Key Difference - Remember This!

- **Big Data Management** = Preparing and storing data (like organizing a library)
- **Data Science** = Analyzing data and making predictions (like reading books and writing reports)

This course focuses on Big Data Management: How to store, organize, and retrieve big data efficiently.

1.4 Big Data Statistics (Exam Facts)

Memorize These Numbers for Exam!

- **2010:** 2 Zettabytes of data existed worldwide
- **2016:** 18 Zettabytes (9× growth in 6 years!)
- **2021:** 74 Zettabytes
- **2030:** Predicted 2,500 Zettabytes
- **2035:** Predicted 19,200 Zettabytes

What's a Zettabyte?

$$\begin{aligned}1\text{ Zettabyte} &= 1,000\text{ Exabytes}\\&= 1,000,000\text{ Petabytes}\\&= 1,000,000,000\text{ Terabytes}\\&= 1,000,000,000,000\text{ Gigabytes}\end{aligned}$$

Real comparison: 1 Zettabyte = watching HD movies for 36 million years non-stop!

2 Lecture 2: Semantic Web & Knowledge Graphs

2.1 Understanding the Problem: Web of Documents

The Fundamental Problem

The original web (created by Tim Berners-Lee in 1989) was designed for **humans**, not machines.
Problem: Machines can display documents but cannot *understand* them.

Simple Example: Google Search

Human sees: “Radu Mihailescu works at Heriot-Watt University”
Human understands:

- Radu Mihailescu is a person
- Heriot-Watt is a university
- “works at” shows employment relationship

Machine sees: Just text: “Radu Mihailescu works at Heriot-Watt University”
Machine understands: Nothing! It’s just letters.
Solution: Add *metadata* (data about data) to make it machine-readable!

2.2 Solution: Adding Metadata

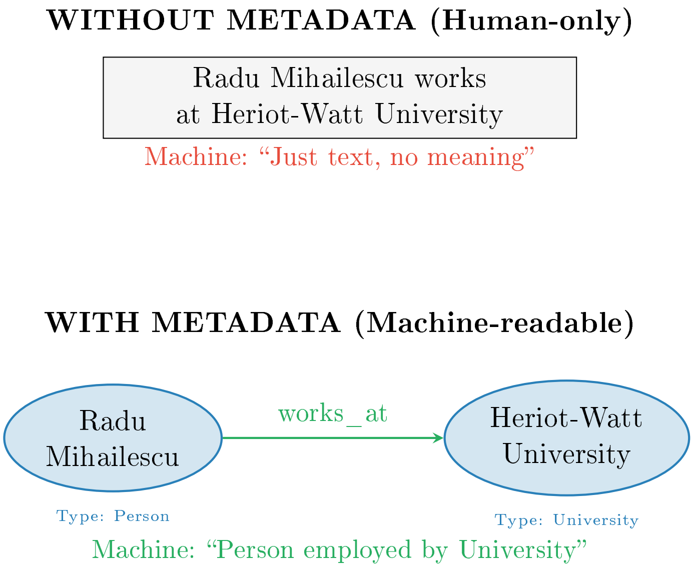

WITHOUT METADATA (Human-only)
Radu Mihailescu works at Heriot-Watt University
Machine: “Just text, no meaning”
WITH METADATA (Machine-readable)
Radu Mihailescu
works_at
Heriot-Watt University
Type: Person
Type: University
Machine: “Person employed by University”

Figure 3: Human-Readable vs Machine-Readable Data

2.3 What is the Semantic Web?

Core Definition - Know This!

Semantic Web = A "web of data" that machines can understand and process
Key Idea: Add meaning (semantics) to web data so computers can:

- Understand relationships between things
- Answer complex questions automatically
- Make intelligent decisions
- Connect information from different sources

Created by: Tim Berners-Lee (same person who invented the original web!)

2.3.1 Traditional Web vs Semantic Web

Traditional Web (Syntactic)	**Semantic Web**
Web of Documents	**Web of Data**
HTML pages linked together	Data entities linked together
Designed for humans to read	Designed for machines to process
Example: Wikipedia article about Bob Dylan	**Example:** Structured data: Bob Dylan (type: Person) born (relationship) 1941 (date)
Search finds matching text	Search understands meaning and relationships
Google shows "pages about Bob Dylan"	Google shows "Bob Dylan is a musician, born in 1941, won Nobel Prize"

Table 3: Syntactic Web vs Semantic Web Comparison

2.4 What is a Knowledge Graph?

Definition - Exam Critical

A **Knowledge Graph** is a knowledge base structured as a **graph**.
Components:

- **Nodes (Vertices):** Represent entities (people, places, things)
- **Edges:** Represent relationships between entities
- **Labels:** Describe what the relationship means

Key Property: It's a **directed labeled graph** (arrows point in a direction and have names)

2.4.1 Simple Knowledge Graph Example

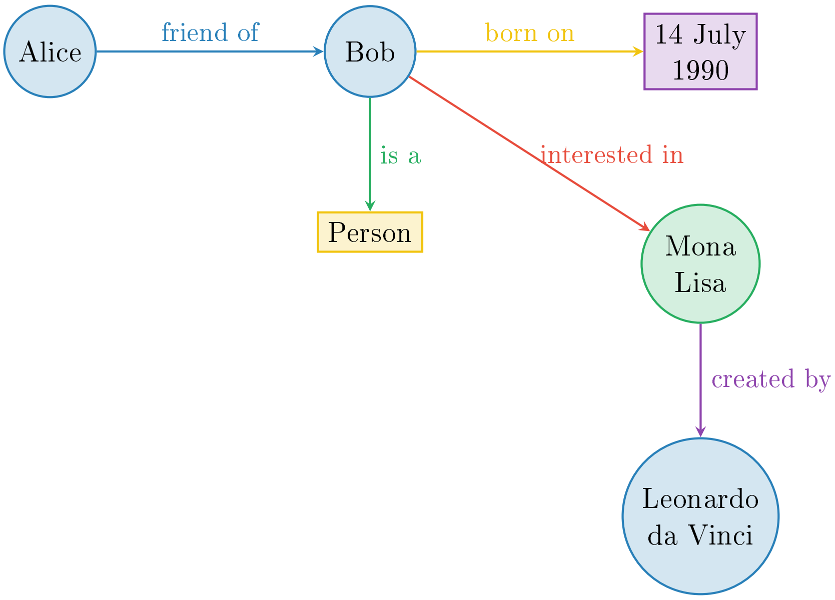


Figure 4: Simple Knowledge Graph - Each arrow is a fact!

Reading the Knowledge Graph

From the graph above, we can read facts:

1. Alice is a friend of Bob (relationship between two people)
2. Bob is a Person (type classification)
3. Bob was born on 14 July 1990 (attribute/property)
4. Bob is interested in Mona Lisa (interest relationship)
5. Mona Lisa was created by Leonardo da Vinci (creation relationship)

Power: A machine can now automatically answer questions like:

- "Who is Alice's friend?" → Bob
- "When was Bob born?" → 14 July 1990
- "Who created the Mona Lisa?" → Leonardo da Vinci

2.5 Real-World Knowledge Graphs

Company/Project	Knowledge Graph Name	Use Case
Google	Google Knowledge Graph	Search results: Shows info panels with facts about people, places, things
Facebook	Facebook Graph	Understands social connections, suggests friends, targeted ads

Company/Project	Knowledge Graph Name	Use Case
Amazon	Amazon Product Graph	Product recommendations: "Customers who bought X also bought Y"
Microsoft	Microsoft Satori	Powers Bing search and intelligent services
Wikidata	Wikidata Knowledge Base	Free, public database with 71 million facts
DBpedia	DBpedia Knowledge Graph	Structured data extracted from Wikipedia

Table 4: Major Knowledge Graphs and Their Applications

Google Knowledge Graph Example

Try this: Search "Albert Einstein" on Google
What you see:

- Picture of Einstein
- Born: March 14, 1879, Germany
- Died: April 18, 1955, USA
- Education: University of Zurich
- Known for: Theory of Relativity, $E = mc^2$
- Nobel Prize: 1921

Behind the scenes: This information comes from Google's Knowledge Graph automatically connecting facts from millions of sources!

2.6 Linked Data Principles (EXAM IMPORTANT)

Tim Berners-Lee's 4 Principles of Linked Data

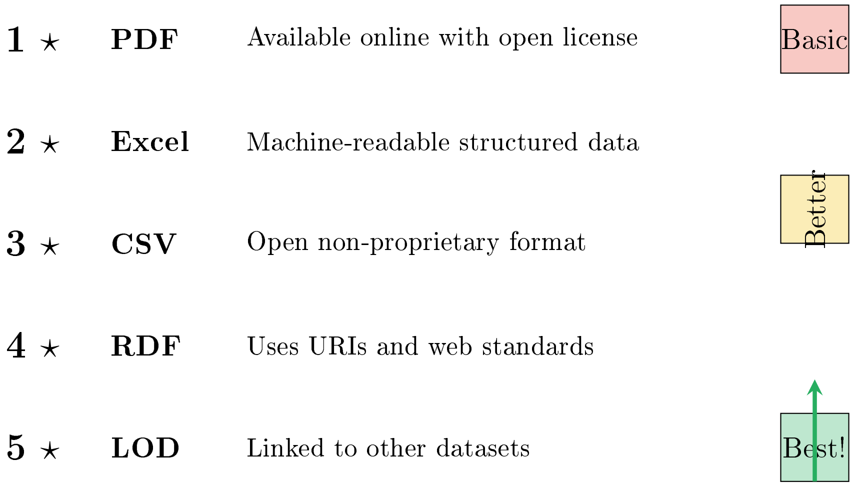
1. **Use IRIs to identify things**
   Give everything a unique web address (like a URL)
   Example: `http://dbpedia.org/resource/Dubai`

2. **Use HTTP IRIs**
   Make addresses accessible over the web
   Anyone can look up the IRI and get information

3. **Provide useful information**
   When someone looks up an IRI, return data in standard formats (RDF)
   HTML for humans, RDF for machines

4. **Link to other data**
   Connect your data to other datasets
   Example: Link "Dubai" to "United Arab Emirates"

2.7 The 5-Star Linked Open Data System (EXAM CRITICAL)



1 ⋆ **PDF** Available online with open license
2 ⋆ **Excel** Machine-readable structured data
3 ⋆ **CSV** Open non-proprietary format
4 ⋆ **RDF** Uses URIs and web standards
5 ⋆ **LOD** Linked to other datasets
Basic
Better
Best!

Figure 5: 5-Star Linked Open Data Deployment Scheme (Tim Berners-Lee)

Stars	Requirements	Example
⋆	Data available online with open license	A PDF government report you can download for free
⋆⋆	Machine-readable structured data	An Excel spreadsheet with organized columns and rows
⋆⋆⋆	Non-proprietary open format	A CSV file (can open without Microsoft Office)
⋆⋆⋆⋆	Uses web standards (IRIs, RDF)	RDF data with HTTP URIs - machines can understand relationships
⋆⋆⋆⋆⋆	Linked to other datasets	Your data connects to Wikidata, DBpedia, etc. - full semantic web!

Table 5: 5-Star System Explained Simply

Real Example: City Budget Data

1-Star: Dubai city budget as PDF on government website
↓ *Good: Available, but hard for machines to process*
2-Star: Same budget as Excel file
↓ *Better: Can import into programs, but needs Microsoft Excel*
3-Star: Budget as CSV file
↓ *Better: Anyone can open it with any software*
4-Star: Budget as RDF with URIs like `budget:2024/transportation`
↓ *Much better: Machines understand structure and meaning*
5-Star: Budget links to other open data (population, maps, services)
Best: Can automatically analyze "cost per citizen" by combining datasets

2.8 Types of Data

Type	Definition
Open Data (OD)	Data available for reuse *free of charge* Not necessarily linked or in good format Example: Government datasets you can download
Linked Data (LD)	Data *connected* to other data using relationships May or may not be free Example: Your database links to Wikipedia entries
Linked Open Data (LOD)	Data that is BOTH: • Free to use (Open) • Connected to other data (Linked) Example: Wikidata, DBpedia
Linked Closed Data (LCD)	Connected data but NOT free Requires license or payment Example: Commercial business intelligence databases

Table 6: Types of Data - Know the Differences!

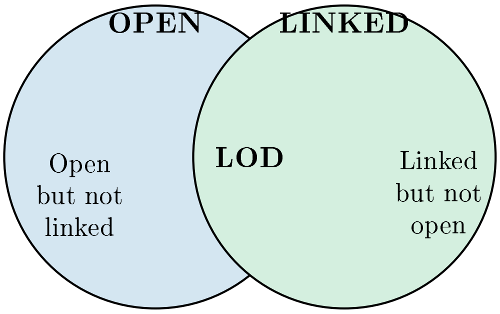


Figure 6: Relationship Between Open Data and Linked Data

2.9 Knowledge Graph Construction (2-Step Process)

Building a Knowledge Graph - The Process

Step 1: Named Entity Recognition (NER)

- Identify entities (things) in text
- Disambiguate (is “Apple” the fruit or the company?)
- Ensure consistency

Step 2: Relationship Extraction

- Find connections between entities
- Label the relationships
- Build the graph

Step-by-Step Example

Input Text: “John teaches Big Data Management at Heriot-Watt University in Edinburgh and Dubai.”

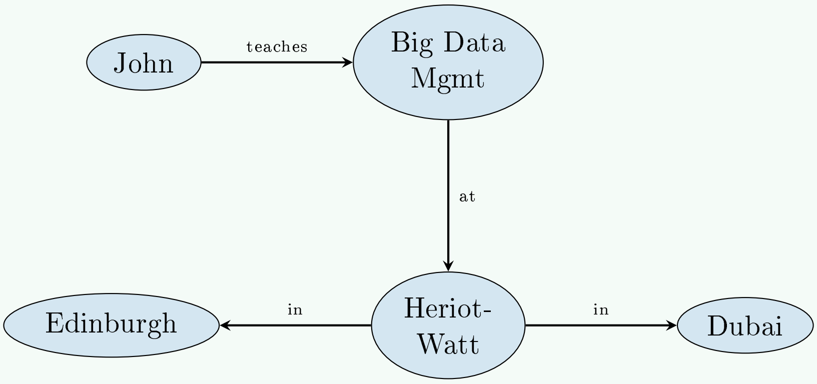
Step 1 - Identify Entities:

- John → Person
- Big Data Management → Course
- Heriot-Watt University → University
- Edinburgh → City
- Dubai → City

Step 2 - Extract Relationships:

- John *teaches* Big Data Management
- John *is-a* Lecturer
- Big Data Management *taught-at* Heriot-Watt University
- Heriot-Watt University *located-in* Edinburgh
- Heriot-Watt University *located-in* Dubai

Result: Knowledge Graph



3 Lecture 3: Semantic Web Technologies

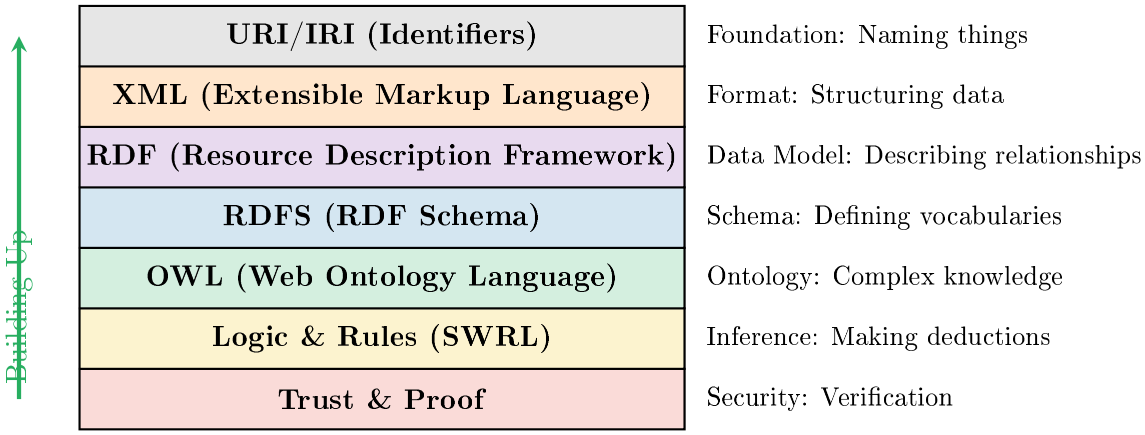
3.1 The Semantic Web Technology Stack

Understanding the Stack

Think of the Semantic Web stack like building a house:

- **Foundation:** URI/IRI (addresses for things)
- **Floor:** XML (format for data)
- **Walls:** RDF (basic data model)
- **Rooms:** RDFS (vocabulary/schema)
- **Furniture:** OWL (advanced ontologies)
- **Intelligence:** Logic & Rules
- **Security:** Trust & Proof layers

Each layer builds on the one below it!



URI/IRI (Identifiers)
Foundation: Naming things
XML (Extensible Markup Language)
Format: Structuring data
RDF (Resource Description Framework)
Data Model: Describing relationships
RDFS (RDF Schema)
Schema: Defining vocabularies
OWL (Web Ontology Language)
Ontology: Complex knowledge
Logic & Rules (SWRL)
Inference: Making deductions
Trust & Proof
Security: Verification
Building Up

Figure 7: Semantic Web Technology Stack - Layers Build on Each Other

3.2 IRIs: Naming Things Globally

What is an IRI?

IRI = **I**nternationalized **R**esource **I**dentifier
Simple Explanation: A unique global "address" for anything (like a URL but more general)
Why needed? To avoid naming conflicts and ensure global uniqueness

The Naming Problem

Scenario: Two developers create entities called "Developer"
Without IRIs:

- Business team: "Developer" = Person who develops land

- Software team: "Developer" = Person who writes code
- Collision! Which one do we mean?

With IRIs:

- Business: `http://myonto.hw.ac.uk/business#developer`
- Software: `http://myonto.hw.ac.uk/software#developer`
- No collision! Each has unique identifier

3.2.1 IRI Structure

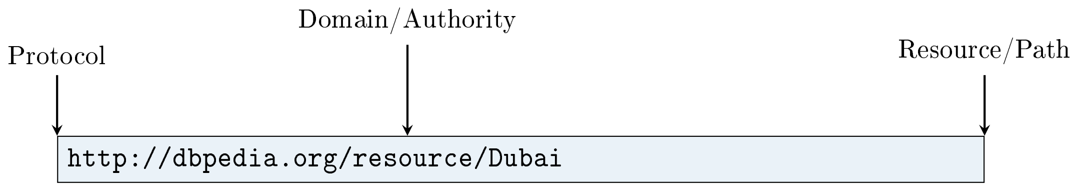


Figure 8: Anatomy of an IRI

3.3 Namespaces (EXAM IMPORTANT)

What are Namespaces?

Namespace = An abbreviation for a long IRI prefix
Why use them? To make writing IRIs shorter and more readable
Format: `PREFIX name: <full_IRI>`

3.3.1 Common Namespaces (Memorize These!)

`Prefix`	**Full IRI**	**Purpose**
`rdf:`	http://www.w3.org/.../rdf-syntax-ns#	Core RDF vocabulary (type, Property, etc.)
`rdfs:`	http://www.w3.org/.../rdf-schema#	RDF Schema terms (Class, subClassOf, label, etc.)
`owl:`	http://www.w3.org/.../owl#	OWL ontology language (Class, ObjectProperty, etc.)
`xsd:`	http://www.w3.org/.../XMLSchema#	Data types (integer, string, date, boolean, etc.)
`foaf:`	http://xmlns.com/foaf/0.1/	Friend-of-a-Friend: people & social networks
`dcterms:`	http://purl.org/dc/terms/	Dublin Core: document metadata (title, creator, date)
`schema:`	http://schema.org/	Schema.org vocabulary (used by Google)

Table 7: Common Namespaces - Know These for Exam!

Using Namespaces

Without namespace:

```
<http://xmlns.com/foaf/0.1/Person>
<http://xmlns.com/foaf/0.1/name>
<http://xmlns.com/foaf/0.1/knows>
```

With namespace:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

foaf:Person
foaf:name
foaf:knows
```

Much shorter and easier to read!

3.4 RDF: Resource Description Framework (CORE CONCEPT)

RDF - The Most Important Technology

RDF = **R**esource **D**escription **F**ramework
What is it? A standard way to describe data as **triples**
Triple Structure: (Subject, Predicate, Object)
Think of it like: A simple sentence with 3 parts

- Subject = The thing we're talking about
- Predicate = The relationship or property
- Object = The value or related thing

3.4.1 Understanding RDF Triples

Simple Triple Example

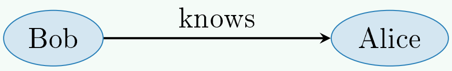
Statement: "Bob knows Alice"
RDF Triple:

Subject	**Predicate**	**Object**
Bob	knows	Alice

Formal RDF:

```
PREFIX ex: <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

ex:bob foaf:knows ex:alice .
```

Visual:



3.4.2 Types of Objects in RDF

Type	Description	Example
Resource (IRI)	Another entity with an IRI Links to other things	`ex:alice` `dbpedia:Dubai`
Literal (Value)	A simple value: text, number, date Cannot be a subject	“Alice” 25 “1990-07-04”
Typed Literal	A literal with a specific data type	“25”^^xsd:integer “1990-07-04”^^xsd:date

Table 8: Types of Objects in RDF Triples

Important Rule!

Literals can ONLY be objects, never subjects!
Valid:

```
ex:bob foaf:age "25"^^xsd:integer .
```

INVALID:

```
"25"^^xsd:integer foaf:age ex:bob .
```

3.4.3 Complete RDF Example with Multiple Triples

Building a Small Knowledge Base

Facts:

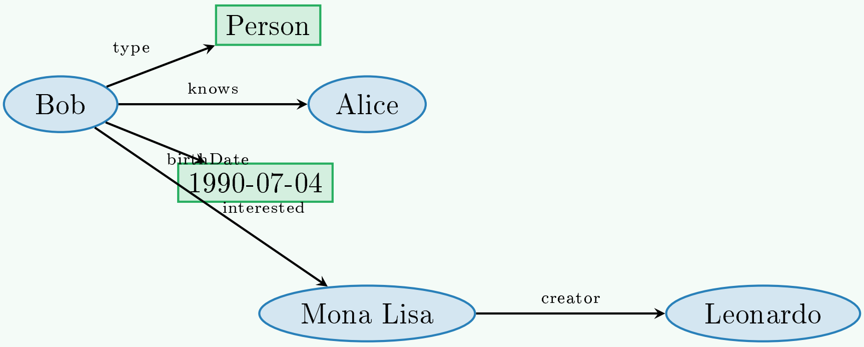
1. Bob is a Person
2. Bob knows Alice
3. Bob was born on July 4, 1990
4. Bob is interested in the Mona Lisa
5. The Mona Lisa was created by Leonardo da Vinci

RDF Triples (Turtle syntax):

```
PREFIX ex: <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX schema: <http://schema.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dbr: <http://dbpedia.org/resource/>

# Fact 1
ex:bob rdf:type foaf:Person .

# Fact 2
ex:bob foaf:knows ex:alice .

# Fact 3
ex:bob schema:birthDate "1990-07-04"^^xsd:date .

# Fact 4
ex:bob foaf:topic_interest dbr:Mona_Lisa .
```

```
# Fact 5
dbr:Mona_Lisa dcterms:creator dbr:Leonardo_da_Vinci .
```

Graph Visualization:



3.5 RDF Serialization Formats (Know the Differences!)

What is Serialization?

Serialization = Converting the abstract RDF graph into a file format that can be saved and shared

Key Point: Same graph, different file formats (like saving a document as .docx, .pdf, or .txt)

3.5.1 Format Comparison Table

Format	Pros	Cons
RDF/XML	• Original standard • Compatible with XML tools • Good for automated processing	• Very verbose • Hard for humans to read • Not intuitive
N-Triples	• Simplest format • One triple per line • Easy to parse • Good for large imports	• No abbreviations • Very repetitive • Large file sizes • No namespace prefixes
Turtle	• **Most human-readable** • Uses namespace prefixes • Compact syntax • **Best for learning!**	• Cannot represent named graphs • Need to understand syntax
N-Quads	• Extends N-Triples • Supports named graphs • Good for large databases	• Very verbose • Harder to read
TriG	• Extends Turtle • Supports named graphs • Human-readable	• More complex syntax • Less tool support
JSON-LD	• JSON format (web-friendly) • Easy for developers • Google recommends it	• More complex for beginners • Different structure

Format	Pros	Cons
RDF-a	• Embeds in HTML • Both humans & machines read • Good for web pages	• Mixes data with presentation • Harder to maintain

Table 9: RDF Serialization Formats Comparison

3.5.2 Same Data, Different Formats

Format Examples: "Bob knows Alice"

1. N-Triples (Most Basic):

```
<http://example.org/bob> <http://xmlns.com/foaf/0.1/knows> <http://example.org/alice> .
```

2. Turtle (Human-Readable):

```
PREFIX ex: <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

ex:bob foaf:knows ex:alice .
```

3. RDF/XML (XML Format):

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <rdf:Description rdf:about="http://example.org/bob">
    <foaf:knows rdf:resource="http://example.org/alice"/>
  </rdf:Description>
</rdf:RDF>
```

4. JSON-LD (Web-Friendly):

```
{
  "@context": "http://xmlns.com/foaf/0.1/",
  "@id": "http://example.org/bob",
  "knows": {"@id": "http://example.org/alice"}
}
```

All four represent the SAME triple!

3.6 Turtle Syntax Shortcuts (Learn These!)

Turtle Makes Life Easier

Turtle has special shortcuts to make RDF more readable and less repetitive.

3.6.1 Shortcut 1: Same Subject (Semicolon)

Using Semicolons

Long Way (Repetitive):

```
ex:bob foaf:name "Bob" .
ex:bob foaf:age 30 .
ex:bob foaf:knows ex:alice .
```

Short Way (Using ;):

```
ex:bob foaf:name "Bob" ;
       foaf:age 30 ;
       foaf:knows ex:alice .
```

Rule: Semicolon (;) means "same subject, different predicate"

3.6.2 Shortcut 2: Same Subject and Predicate (Comma)

Using Commas

Long Way:

```
ex:bob foaf:knows ex:alice .
ex:bob foaf:knows ex:charlie .
ex:bob foaf:knows ex:diana .
```

Short Way (Using ,):

```
ex:bob foaf:knows ex:alice ,
                  ex:charlie ,
                  ex:diana .
```

Rule: Comma (,) means "same subject and predicate, different object"

3.6.3 Shortcut 3: Type Abbreviation

The "a" Shortcut

Long Way:

```
ex:bob rdf:type foaf:Person .
```

Short Way:

```
ex:bob a foaf:Person .
```

Rule: "a" is shorthand for `rdf:type`

3.7 Advanced RDF Concepts

3.7.1 Reification (Statements About Statements)

What is Reification?

Problem: How do we make statements ABOUT other statements?
Example: "Bob says that Alice lives in Dubai" (Bob is the source of the claim)
Solution: Create a resource that represents the statement itself

Reification Example

Original Statement: "Alice lives in Dubai"
Meta-Statement: "Bob created this statement on Jan 1, 2024"
Reification Pattern:

```
# The statement itself
:statement1 rdf:type rdf:Statement ;
            rdf:subject :alice ;
            rdf:predicate :livesIn ;
            rdf:object :dubai .

# Metadata about the statement
:statement1 dc:creator :bob ;
            dc:date "2024-01-01"^^xsd:date .
```

Problem: This creates 4 triples for 1 statement! (Verbose)

3.7.2 Named Graphs (Better Alternative)

Named Graphs - Modern Solution

Idea: Group related triples and give the group a name (IRI)
Advantage: Simpler than reification, widely supported
Format: Use TriG or N-Quads

Named Graph Example

TriG Syntax:

```
# Graph created by Bob
:bobsGraph {
    :alice :livesIn :dubai .
    :alice :age 25 .
}

# Metadata about the graph
:bobsGraph dc:creator :bob ;
           dc:date "2024-01-01"^^xsd:date .
```

Benefit: Easy to track the source of information!

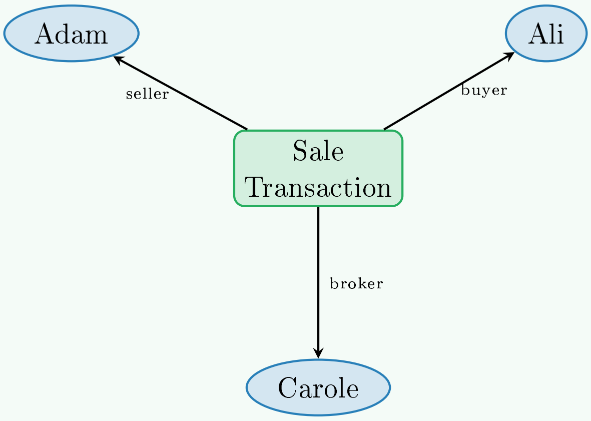
3.7.3 N-ary Relationships

The N-ary Problem

Challenge: RDF only supports *binary* relationships (2 entities)
Question: How do we represent relationships with 3+ participants?
Example: "Adam sold a car to Ali, and Carole was the broker"

N-ary Relationship Solution

Pattern: Create an intermediate "event" or "transaction" node
Graph:



RDF Code:

```
:saleTransaction a :Sale ;
                 :seller :adam ;
                 :buyer :ali ;
                 :broker :carole .
```

Key Insight: The transaction is now a resource that connects all participants!

3.8 Merging Knowledge Graphs

The Power of IRIs

Amazing Feature: If two graphs use the SAME IRI, they automatically merge!
Example:

- Graph A: `dbr:Mona_Lisa dcterms:creator dbr:Leonardo`
- Graph B: `dbr:Mona_Lisa dcterms:title “Mona Lisa”`
- **Merged:** Both facts now connected to the same Mona Lisa!

This is how the Linked Open Data cloud connects billions of facts!

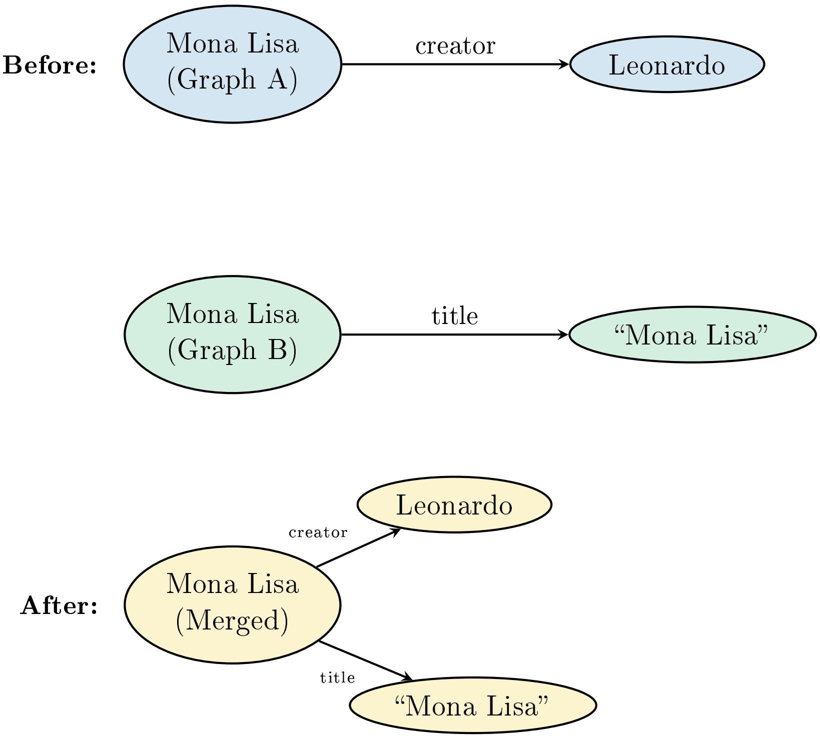


Figure 9: Automatic Graph Merging Using Same IRIs

3.9 Inference and Reasoning

What is Inference?

Inference = Automatically deriving new facts from existing ones using logical rules
Simple Example:

- **Fact 1:** Socrates is a human
- **Fact 2:** All humans are mortal
- **Inference:** Socrates is mortal (derived automatically!)

3.9.1 Open World Assumption vs Closed World Assumption

Closed World Assumption (CWA)	**Open World Assumption (OWA)**
Used by: Traditional SQL databases	**Used by:** Semantic Web, RDF, OWL
Rule: "If it's not in the database, it's FALSE"	**Rule:** "If it's not in the database, it's UNKNOWN"
Assumes complete knowledge	Assumes incomplete knowledge
Example: Database: "Socrates is mortal" Query: "Is John mortal?" Answer: FALSE (not in database)	**Example:** Database: "Socrates is mortal" Query: "Is John mortal?" Answer: UNKNOWN (not stated)
Good for: Closed systems with complete data	Good for: Open web where data is incomplete

Table 10: Closed World vs Open World Assumptions

OWA Example for Exam

Given Facts:

- Socrates is a human
- All humans are mortal

Questions:

1. Is Socrates mortal? → TRUE (can be inferred)
2. Is John mortal? → UNKNOWN (we don't know if John is human)
3. Is ChatGPT mortal? → UNKNOWN (we don't know what ChatGPT is)

Key Point: Semantic Web uses OWA because the web is always incomplete!

4 Exam Preparation Section

4.1 Key Terms Summary

Term	One-Sentence Definition
Semantic Web	A web of data with meaning that machines can understand and process
Knowledge Graph	A graph-based knowledge base with entities (nodes) and relationships (edges)
RDF	Resource Description Framework - a standard for representing data as triples
Triple	A statement with 3 parts: (Subject, Predicate, Object)
IRI	Internationalized Resource Identifier - a unique global address for resources
Namespace	An abbreviation for a long IRI prefix
Ontology	A formal model of a domain defining concepts and their relationships
Linked Data	Data connected to other data through relationships and IRIs
Open Data	Data available online for free reuse under an open license
LOD	Linked Open Data - data that is both linked AND open
Turtle	A human-readable RDF serialization format
Literal	A simple value (text, number, date) that can only be an object in RDF
OWA	Open World Assumption - unknown information is treated as unknown, not false
CWA	Closed World Assumption - unknown information is treated as false
Inference	Automatically deriving new facts from existing ones using rules

Table 11: Key Terms for Exam

4.2 Common Exam Question Types

4.2.1 Type 1: Drawing Knowledge Graphs

Exam Question Example 1

Question: Draw a knowledge graph for the following statements:

- John is a lecturer
- John teaches Big Data Management
- Big Data Management is a course at Heriot-Watt University
- Big Data Management is taught in Edinburgh and Dubai

Answer:
Step 1 - Identify Entities:

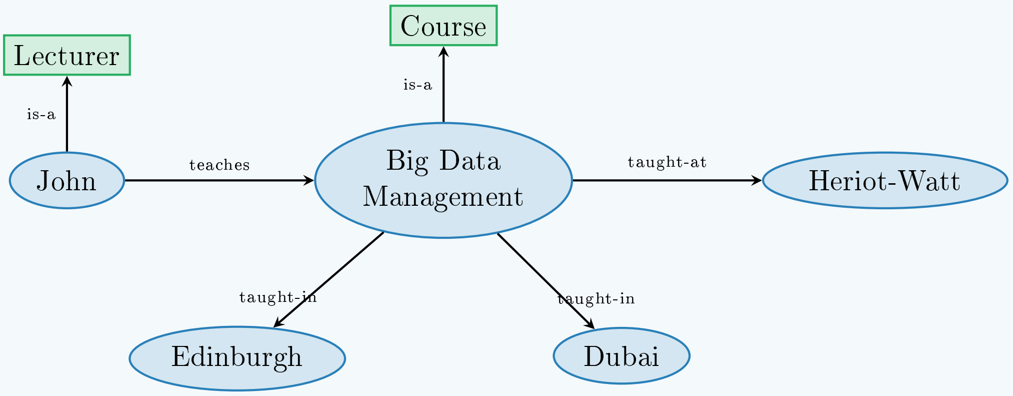
- John (Person/Lecturer)
- Big Data Management (Course)

- Heriot-Watt University (University)
- Edinburgh (City)
- Dubai (City)
- Lecturer (Type/Class)
- Course (Type/Class)

Step 2 - Identify Relationships:

- John *is-a* Lecturer
- John *teaches* Big Data Management
- Big Data Management *is-a* Course
- Big Data Management *taught-at* Heriot-Watt University
- Big Data Management *taught-in* Edinburgh
- Big Data Management *taught-in* Dubai

Step 3 - Draw Graph:



4.2.2 Type 2: Writing RDF Triples

Exam Question Example 2

Question: Write RDF triples in Turtle syntax for the graph from Question 1.
Answer:

```
@prefix ex: <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

ex:John rdf:type ex:Lecturer .
ex:John ex:teaches ex:BigDataManagement .
ex:BigDataManagement rdf:type ex:Course .
ex:BigDataManagement ex:taughtAt ex:HeriotWattUniversity .
ex:BigDataManagement ex:taughtIn ex:Edinburgh .
ex:BigDataManagement ex:taughtIn ex:Dubai .
```

4.3 Practice Questions

Practice Makes Perfect

Work through these questions to prepare for your exam!

1. Explain the difference between data, information, and knowledge with examples.
2. What are the 5 V's of Big Data? Explain each with an example.
3. Draw a knowledge graph representing your family relationships.
4. Write RDF triples in Turtle format for: "Paris is the capital of France, which is in Europe."
5. Explain the difference between Open World Assumption and Closed World Assumption.

5 Conclusion

This document has covered the fundamental concepts from Lectures 1-3 of Big Data Management. Make sure you understand:

- The Data-Information-Knowledge hierarchy
- The 5 V's of Big Data
- Knowledge graphs and their structure
- RDF triples and Turtle syntax
- Semantic Web technologies
- Inference and reasoning

Good luck with your exam preparation!