Part 1: Knowledge Lake

1. Data Curation Component

a) All the examples below are subtasks of information extraction:

- **Named entity recognition (NER)** is used to find and classify named entities in raw data into pre-defined categories such as company names, locations, time expressions.
    - Using NER on a text “Toyota is a car company founded in Japan in early 1900” would return
      {
      ‘Toyota’ : ‘company’,
      ‘Japan’ : ‘country’,
      ‘1900’ : ‘year’
      }
    - resource
- **Keyword extraction** helps us draw some insights from a text or document by identifying the most relevant words and phrases within them.
    - This is used in context summarisation (summary of the meeting/discussion or a blog etc), and preview analysis.
    - resource
- **Sentiment extraction** helps us understand the emotional tone within the text or document, in other words it’s also an insight from the text.
    - As we can classify the text/document based on the level of emotion such as positive, negative and neutral, can be applied to review analysis or customer insights analysis.

b)

- Snowflake Streams & Tasks, Snowpipe can handle new data that has just been added to the tables or even a new PDF file. Its micro-batch approach enables near real-time ingestions as well as real-time analysis. For example, scheduling a task to run at a high frequency allows it to operate the defined pipeline as the new data enters the table.
- Neo4j has a better contextualising capability compared to Snowflake, but does not have as good real-time ingestion as Snowflake. Neo4j’s graph structure allows stored data (nodes, labels and relationships) to be used in real-time analysis.
- Neo4j and Snowflake now have integration in between, so the data in snowflake can be used to bring insights based on GDS within snowflake.

Resource:
https://neo4j.com/blog/neo4j-snowflake-integration/
https://www.youtube.com/watch?v=dN_IZp2W148

c)

1. The pipeline starts with **ingesting data** from sources like Kafka, databases, or files like CSV or JSON. Depending on the source, Snowflake has choices of tools (Kafka Connector, Snowpipe) to make our lives easier. They are both helpful when we have Kafka or a cloud database like AWS as a source of data.
2. As a next step, we have to capture changes in real time by tracking data modifications using **streams**. It's useful to monitor new records in incoming data.
3. Once the data is ingested, **SQL queries** transform the data (cleaning, normalising, and enriching the data.) Then, we **enrich** them with Additional information collected with Wikidata or WordNet for example.
4. Extract entities (e.g., customer names, products, or transactions) from structured data or semi-structured data using simple SQL queries.
5. Using **Snowpipe** the processed data are now in target tables, ensuring that transformed and enriched data is available for querying in near real-time.
6. Once the extracted data is ready to use, we can run complex queries or generate insights with BI tools like Tableau or Power BI.

Snowflake is more effective for entity extraction from sources like structured or semi-structured data due to its scalability, simplicity, and real-time capabilities. It shines in environments focused on data warehousing and large-scale processing. Whereas **Neo4j** is more preferred when we focus on understanding the relationship between entities. It is suited for more connected, graph-based data rather than large-scale structured datasets.

https://billigence.com/blog/data-pipelines-in-snowflake/

2. Data Enrichment Component

a)

- ➢ **WordNet** is used to enrich the data by further extracting synonyms and categorical words (e.g. colour) for each word in a provided document. This approach can improve semantic understanding of the document and result in carrying out more advanced text analysis.
- ➢ **Wikidata** is a huge public knowledge database that can be read and edited by both humans and machines. This is often used to annotate and enrich the extracted data when the data is missing some information or incomplete, or even when we want to add more data on top of them. However, there is an issue: some data we want could be limited or missing on wikidata.

b) Identify and explain existing technologies for data enrichment, with a specific focus on Databricks. Discuss how Databricks' capabilities (e.g., Delta Lake, MLflow, and built-in machine learning libraries) support data enrichment processes, including the integration of external knowledge bases.

- MLflow manages processes that have to do with machine learning. Allowing us to simplify the process of training to deploying models. This is useful as we can use it to enrich existing datasets.
- Delta Lake is a place to store our enriched data as well as manage them efficiently. It allows us to track changes over time so we can make sure the stored data stays organised, accurate and reliable.
- Databricks has built-in libraries that can be used for machine learning - analysis, prediction and even extraction. This can support our data-enrichment tasks.
- As discussed earlier, the external APIs like wikidata can also be used with databricks with ease, which basically means that we can access wiki to enrich our data further to meet the industry requirements etc.

https://medium.com/@ajayverma23/databricks-vs-delta-lake-understanding-the-differences-and-pros-and-cons-e1cd640d1d33

c)

1. **Data Ingestion**: We can start by loading dataset into **Databricks**. We use **Delta Lake** to store and manage this data, this way we can ensure version control and reliability.
2. **Data Enrichment with Machine Learning**: Databricks' **built-in machine learning libraries** or **MLflow** to build and deploy machine learning models (for example, sentiment analysis on customer reviews.) This can identify patterns in the data, like customer behaviour. MLflow is useful for the automation, especially with the model trained being used so as to keep consistent results.
3. **Integration with External Knowledge Base**: we can link enriched data with an external knowledge base, such as **Wikidata**, using an API. Depends on the need, we could use the Wikidata API to fetch additional information to further enrich the dataset. Because Databricks supports API integration and it is easy, this can be useful for enrichment.
4. **Data Storage**: Once the data is enriched, it is stored back in **Delta Lake**, which ensures all changes are tracked and the data is consistent throughout the enrichment process.

A data enrichment process involves enhancing your existing data with additional valuable information from external sources or through analysis. Using Databricks, this process becomes streamlined and scalable. It is important to evaluate the effectiveness of the enrichment process by assessing completeness for the enriched data.

3. Graph Linking Component

a)

Techniques	What it is	Use case
Relationship extraction	One of NLP techniques to derive	This is extremely useful when

	relationships between noun entities in the provided document.	we create a knowledge database with the extracted entities.
Named entity recognition	Find Named entities in the provided document. (more explanation under 1a)	Same as above, this makes the extraction a lot faster hence creation too.
Centrality	Identify the importance of an entity in the knowledge graph.	Can create new labels based on this and form a group of centrals of several communities.
Community detection	Reveals clusters, and evaluates how closely they are tied.	Is used to get an idea of what entities are strongly connected, can be used to get an idea as a starting point.
Similarity	Detect similar entities and connect them.	Find similar entities and get insights.
Knowledge graph embeddings	Represent the knowledge graph into a continuous vector space preserving the semantic meaning and of course entities and relationships.	Is used for relation extraction, clustering, link prediction etc.

https://medium.com/slalom-build/become-a-knowledge-graph-jedi-445d29c59eac

b)

Neo4j is an open-source graph database. It uses cypher language and can visualise graphs when working with highly connected data. Cypher is like SQL, it is used to store, filter, retrieve and visualise the data, and it's designed for everyone to learn and understand with ease without compromising the power and capability of other standard query languages. High performance for graph traversals and real-time querying with features like native graph storage and indexing, supporting its scalability and performance.

Amazon Neptune is advertised by AWS as "A fast, dependable, fully managed graph database service that makes it easy to build and run applications that work with highly connected datasets." A dedicated, high-performance graph database engine at the heart of Neptune is designed to store billions of relationships and query the graph with millisecond latency. We can create searches that effectively traverse densely connected datasets because to Neptune's support for the well-known graph query languages Apache TinkerPop Gremlin and W3C's SPARQL. Network security, fraud detection, knowledge graphs, recommendation engines, and drug discovery are just a few of the graph use cases that Neptune powers.

https://medium.com/@kavithareddy.gade/comparision-of-aws-neptune-vs-neo4j-e9110a827057
https://www.analyticsvidhya.com/blog/2024/08/neo4j-vs-amazon-neptune/

c)

Graph structure allows for powerful analytics while maintaining flexibility for future expansion. Typically, in machine learnings we assume the rows are independent, however, that is not really true as entities always have some relationship with other entities. Graph database is focused on this relationship that connects nouns with verbs for example. Therefore it makes it more natural to apply to our daily lives. Graph can easily accommodate new types of relationships or properties as business needs evolve. It is not just another database, t's a way to see the complete picture, and get actionable insights from them.

4. Security and Governance Component

a)

- **Data Sensitivity and Privacy**: Managing and protecting sensitive information while ensuring compliance with regulations like GDPR is crucial, especially as contextualised data combines multiple sources, increasing privacy risks.
- **Access Control and Role-Based Permissions**: Implementing fine-grained access controls is critical to ensure that only authorised users can access specific contextualised data, which is challenging due to the interconnected nature of the data.
- **Data Governance and Policy Enforcement**: Enforcing governance frameworks and ensuring consistent application of policies for data ownership, access, and use is difficult, especially in a dynamic environment with diverse and constantly evolving data sources.

https://cloudian.com/guides/data-lake/data-lake-security-challenges-and-6-critical-best-practices/

b)

Azure Active Directory (Azure AD) is a cloud-based access management service that is essential for securing Knowledge Lakes. Azure AD comes with many features such as

- Multi-factor authentication (MFA) to enhance security and support protocols like OAuth and OpenID Connect for secure sign-ins.
- Role-based access control (RBAC) to specify which data users can access for their needs.
- Single Sign-On (SSO) simplifies user authentication across many platforms, making it easier to manage secure access to Knowledge Lakes.

Microsoft Defender offers a set of security tools for protecting data in Knowledge Lakes by detecting threats and vulnerabilities. Its features include:

- Threat Detection helps detect anomalies in access or data usage patterns. This can ensure the higher standard of the security of the lake.
- Endpoint Protection secures servers and devices that have access to the lake, and reduces the risk of data breach.
- Automated Responses uses AI to automate responses to certain types of security events, like blocking malicious activities.

While both OKTA and Azure AD offer the same functionalities like SSO and MFA, Azure AD integrates more seamlessly with other Microsoft products, and this a great advantage for businesses with Azure-based knowledge lake. OKTA could be more flexible when it comes to multi-cloud environments, but lacks the seamlessness that Azure AD provides. For data protection, Microsoft Defender is close to Azure, offering stronger integration for securing data flows in knowledge lakes OKTA's Adaptive MFA, which focuses more on identity protection rather than comprehensive data security.

c)

- ➢ All access to the Knowledge Lake should be strictly controlled based on roles and responsibilities, and the implementation can be done with Azure AD. With many layers of security protocols such as RBAC to initially limit the data access to the authorised personnel, and MFA to make sure the access is not easy.

- ➢ Strong identity verification needs to be in place so as to ensure only authorised users can access the Knowledge Lake. SSO can help simplify the management of the users with access to the lake, and incorporating with MFA we can make this more robust. Azure AD makes the access management easy with the HR status i.e. once not employed access should be removed.

- ➢ Data monitoring needs to be implemented to ensure data integrity through proper tracking of data access and modifications. Azure Monitor and Security Centre oversees the activity within the data lake and alerts if there is any suspicious activities. We can also use audit logs to analyse who accessed the data and capture all related metadata to maintain transparency within the lake.

- ➢ Implementing Azure AD ensures compliance with regulatory requirements, maintaining secure access and protecting sensitive information in the Knowledge Lake.

- ➢ These policies ensure strong identity management, consistent monitoring, and tight access control, safeguarding against unauthorised access and ensuring the data's integrity and security.

5. Indexing and Search Component (10 Marks)

a)

Techniques	What is	How its used
Federated Search	Simultaneous search on many data sources, merging results into a single view, enhancing context by combining diverse data.	Research, knowledge lakes
Real-Time Indexing	Indexes data as it is ingested, ensuring immediate searchability and relevance.	Useful for security analytics and real-time monitoring
Faceted Search	Filters search results by specific contextual attributes (e.g., categories, locations) to refine search queries for better results.	E-commerce, customer service data analysis
Distributed Search	Searches across large datasets stored across multiple servers, ensuring the speed to access contextualised, large-scale data.	data lakes
Contextual Search	Uses the semantic relationships to improve search accuracy and relevance.	AI-driven recommendations, personalised search

b)

Google Cloud Search enables federated search across Google Workspace and third-party platforms, in other words, a single query can provide access to many data sources. This lets us search contextualised data more efficiently in the data lake. On top of that, Google Cloud Search comes with an advanced algorithm for users to receive personalised results. Integrating with ML and improves its accuracy based on user interactions.

BigQuery is Google’s fully managed data warehouse, and it helps us query a large dataset very quickly. Thanks to its powerful indexing ability, both structured and semi-structured data can be searched with ease. Google’s ML tools can be used together with BigQuery to allow a wide range of actions such as pattern detection automation on contextualised datasets. Users can run complex queries that combine real-time data with machine learning models to enhance the relevance of the search results.

Elasticsearch is a widely used tool for searching across large, distributed datasets, and has real-time indexing and search capabilities, but it lacks the deep integration with machine learning and analytics that BigQuery offers. Elasticsearch is highly customisable for various use cases but requires more manual setup for federated searches.

Apache Drill is a tool for querying large datasets from sources like NoSQL databases and Hadoop. It’s similar to BigQuery in that it’s great for handling big, diverse datasets. However, Apache Drill is preferred for schema-less, ad-hoc queries and enables more flexibility. Apache

Drill doesn’t have the extensive indexing and machine learning features that Google's tools provide for sentimental analysis.

c)

- **Create and Configure a Cloud SQL Instance:**
    - In the Google Cloud Console, navigate to SQL and create a new Cloud SQL instance.
    - Choose preferred database engine, configure instance details (name, region), and apply security settings needed for the project.
    - Once the instance is created, add datasets to the Cloud SQL instance. CSV uploads or connecting external sources are the main methods.
- **Enable BigQuery Connection API:**
    - In the Cloud Console, head to APIs & Services and enable the BigQuery Connection API. This allows BigQuery to query external datasets stored in Cloud SQL.
- **Connect to Cloud SQL Instance in BigQuery:**
    - In BigQuery, a new external connection can be created to Cloud SQL instance.
    - Choose Cloud SQL as the source and provide the details needed to connect them such as instance name, database type.
- **Check Data in Cloud SQL:**
    - We need to verify that our data is successfully added and available in Cloud SQL for querying
- **Query the External Source:**
    - In BigQuery, we can now use SQL queries to access data from the linked Cloud SQL.
- **Check Available Tables:**
    - Check if it works by running the query in BigQuery. The query will then pull data from the external Cloud SQL, and display it in the BigQuery interface - combining datasets from multiple sources (Cloud SQL + BigQuery datasets). All tables and datasets are now visible in BigQuery. The federated search results will combine data from both Cloud SQL and native BigQuery datasets.

6. Visualization Component (10 Marks)

a)

Techniques	What is	How its used
Graph network	Visualises relationships between nodes, helps exploring contextual connections (relationships)	SNS analysis, Customer churn (event-based analysis)

Scatter plot	Used to identify relationships between variables. Our interest is around how they are related.	Linear model forecast, exploratory data analysis
Heatmap	Uses colour to represent data density or frequency, find patterns across context.	Sentiment analysis.
Bar chart	Compares categories using bars.	Enhances understanding of data based on contextual factors.

b)
D3.js is a JavaScript library that allows developers to create complex visualisations for the web. It offers complete control over visual designs, allowing for extensive customization such as interactive charts, graphs, and maps.

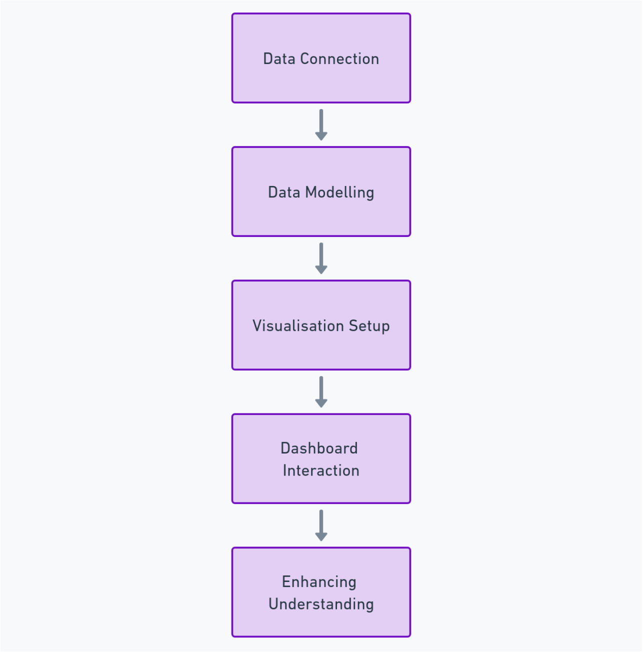
Power BI can transform raw data into visuals like charts and graphs. It's equipped with an user-friendly interface to connect to various data sources. Highly customizable dashboards that are easy to share. Good for both beginners and advanced users.

Power BI's simple drag-and-drop feature links big sets of data with ease, and it allows us to see how different pieces of information connect with each other. Power BI also allows you to add tools to visualise more complicated data. While with D3.js, this is not as easy as how Power BI does, the users are required to be a developer in order to customise further to meet the requirements. This is why Power BI is widely used compared to D3.js.

https://6sense.com/tech/data-visualization/microsoftpowerbi-vs-d3js

c) (4 Marks) Discuss the creation of a data visualization dashboard using Power BI to display insights derived from the contextualized data in your Knowledge Lake.

Explain how the dashboard enhances the understanding of the knowledge graph structure and relationships.



1. We first need to start by connecting data to Power BI. Power BI can be connected directly to the knowledge lake and pull structured and unstructured data from there.
2. Once connected to the lake, Power BI allows you to model the data to represent a knowledge graph structure. Using tables for entities and relationships, you can set up the visualisation like neo4j, but less aesthetic. Not only the graph visualisation, there are more ways to visualise the data with Power BI (**force-directed graphs** and **network charts.**)
3. The dashboards are interactive, so users can take many actions to dive into specific relationships or data points for more investigation. The interactiveness allows users to explore the **knowledge graph structure** and gain a better understanding of the connection between entities.

Dashboards (visualisations) can break down the complicated data into visuals that help us understand the data in our eyes. In this case, contextualised data and entities' linkage etc can be visualised and it is easy to capture the connections. If we are interested in some elements, we can also dig that bit deeper to uncover another truth (hidden patterns.) Thanks to the visualisation, we can also observe clusters and the centrality of them or communities. In summary, the dashboard turns complex relationships into clear, actionable insights, improving decision-making and data understanding.

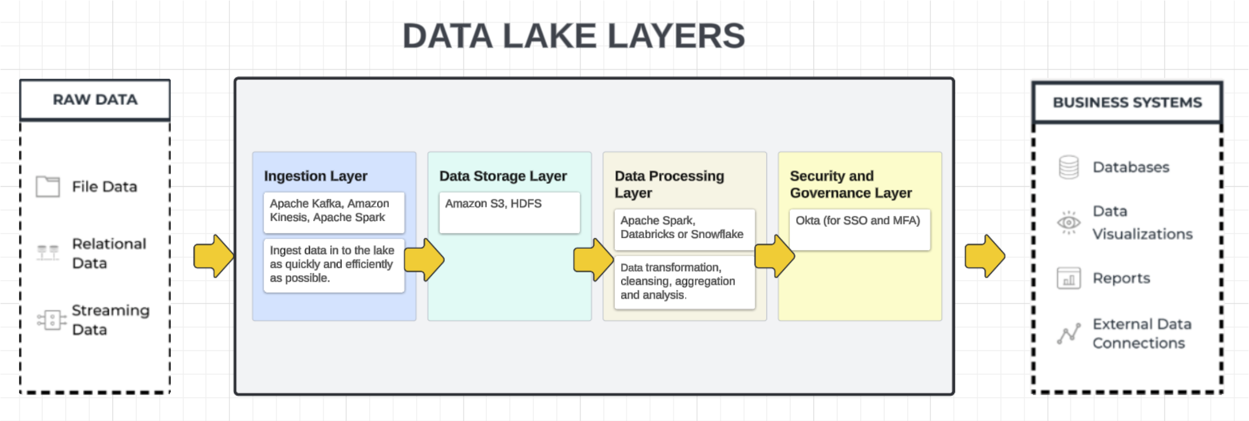
Part 2: Advanced Knowledge Lake Architecture

1. Knowledge Lake Architecture (10 Marks)

a) The Knowledge Lake architecture integrates several key components that work together to manage, enrich, and visualize data effectively. Below is a detailed design of the architecture along with an explanation of the data flow.

Data Flow and Layers in Knowledge Lake:

1. **Data Ingestion**
   - Raw data is collected from various sources (databases, APIs, files) and ingested into the Knowledge Lake.
2. **Data Curation**
   - The ingested data is processed by the tools discussed under Data Curation Component.
   - This layer is critical for collecting, organizing, and maintaining the quality of data.
   - Ensures that data is accurate, consistent, and up-to-date.
3. **Data Enrichment**
   - The curated data is then passed to the Data Enrichment Component.
   - In this layer, we add more context and information to the raw data by using tools like Databricks.
4. **Graph Linking**
   - The enriched data is linked by tools like neo4j, creating relationships between entities.
   - This creates a graph structure that allows for complex queries and insights.
5. **Security and Governance**
   - Throughout the data flow, the Security and Governance Component monitors and ensures tight data protection measures.
   - This layer exists to protect data integrity and confidentiality, and of course the access to the knowledge lake.
6. **Indexing and Search**
   - The enriched and linked data is indexed by the Indexing and Search Component.
   - This enables efficient data retrieval and supports user queries.
7. **Visualization**
   - Finally, the indexed data is made available to be visualised for better understanding.
   - Data is presented in various formats, allowing users to analyse and interpret insights effectively.


DATA LAKE LAYERS
RAW DATA
File Data
Relational Data
Streaming Data
Ingestion Layer
Apache Kafka, Amazon Kinesis, Apache Spark
Ingest data in to the lake as quickly and efficiently as possible.
Data Storage Layer
Amazon S3, HDFS
Data Processing Layer
Apache Spark, Databricks or Snowflake
Data transformation, cleansing, aggregation and analysis.
Security and Governance Layer
Okta (for SSO and MFA)
BUSINESS SYSTEMS
Databases
Data Visualizations
Reports
External Data Connections

(^ Same image used in my previous assignment)

b) It is related to project management, and we are assuming the requirements are gathered and there is architecture document available, so as to skip the stage 1-3. Let's say that the new technologies are selected as we already know which gaps to fill (in this case knowledge extraction and linking technologies.)

1. **Assess the current state**
2. **Define the target state**
3. **Identify the gaps and opportunities**
4. **Develop the roadmap and plan**
5. **Implement the changes**
6. **Monitor and review the results**
7. **Here's what else to consider**

After integrating new technologies, we will have a various ways of dataenhancement, enrichment and graph linking. Before we fully integrate and roll out the new model, we now need to build a prototype and test and develop further - our main focus here is to see its effectiveness. At the same time, we can gather user feedback to assess the performance as well as its effectiveness in solving real-world use cases as per project requirement. Once we prove its effectiveness, we can then train data engineers and analysts to get used to the new tools and technologies. It is important to leave documentation of the new architecture as well as the usecases in order to transfer knowledge and make the training easy for future. To monitor its performance, we need to set some metrics - data quality, extraction accuracy need to be tested. It's important to keep our eyes on the new technologies, but following the steps we can ensure the knowledge lake's improvement is in place successfully.

https://www.linkedin.com/advice/0/how-do-you-update-maintain-enterprise-architecture

2. Integration of External Knowledge Bases (10 Marks)

a)

API Integration allows us to access their data, by sending queries we can retrieve desired data in a structured format like JSON or XML. The retrieved data such as entities, relationships, categories can be ingested into the knowledge lake through pipeline, then store this in the database and link them by common attributes for the sake of enrichment.

RDF Mapping allows us to integrate external knowledge bases into the knowledge lake by linking similar entities with ease.

Batch Download and ETL Pipelines requires us to download the desired sections in bulk first, then use ETL pipelines to incorporate this data into your knowledge lake.

Knowledge Graph Integration uses WikiData/DBpedia's knowledge graph structure to extend the datalake's graph structure. The retrieved data can exist in a graph database like Neo4j, and we link the already existing entities with them so as to enhance the data lake.

b)

- ➢ Enhance data with contextual information from globally recognised knowledge bases.
- ➢ Improve data quality and semantic understanding of datasets.
- ➢ Create richer connections and insights within knowledge lake by leveraging the linked data from external knowledge sources.

Suppose we have a knowledge lake that stores structured data about medical research papers, disease name, authors and contributors. With these information, we can now enhance the data lake further with the help of WikiData.

By integrating WikiData, the knowledge graph in the Knowledge Lake gains richer relationships such as disease histories, cures, hospitals, contributors. Missing information can be filled in from WikiData, providing a more comprehensive view - we cannot 100% fullfill this with just WikiData as some information is always missing. However, contextual data, such as an author's major contributions or co-authorship with others, can now be explored so as to understand the influence within the community. And they are critical when it comes to decision making because this enriched graph can give us a holistic view.

Below are the examples for some methods presented in the previous question:

API Integration: We query WikiData API for detailed information about the authors of the research papers, retrieving additional data like disease history, contributed hospitals, and so many other items useful to enrich the data. We then parse the returned data in a setup ETL pipeline. For each madical research paper's targeted named entities we can map the WikiData entities and enrich the dataset with the additional information. Now the enriched research paper data is stored back in the data lake. Targeted entities are now linked to WikiData entities, and ready for further analysis.

https://agg-shashank.medium.com/an-introduction-to-using-wikidata-apis-a678ee6d2968

RDF Mapping: We first download RDF data from WikiData for authors, diseases, and their publications. RDF comes with graph-like structure so we can use them to map external data into our existing knowledge graph. We parse the RDF data with Neo4j, this way we add new nodes and relationships for mapping. Our knowledge graph now contains external relationships, such as which hospitals, contributors are affiliated with the research paper.

3. Generative AI and Knowledge Lake (20 Marks)

a) Integrating Generative AI into the knowledge lake can automate and boost some processes, including data enrichment, cleaning, visualization, and search abilities. With GenAI, businesses can have more sophisticated data lake that can achieve higher efficiency, improved data quality, and deeper insights, ultimately leading to better decision-making.

- ➢ **Data Enrichment**
    - ○ Generative AI can suggest entities we could link to by analyzing existing data.This way we can automate the process of data exploration, and make further enrichment easy and potentially automated.
- ➢ **Data Cleaning and Validation**
    - ○ AI models can validate data against pre-defined data format, ensuring that only high-quality, reliable data enters the Knowledge Lake. This process can include checking for completeness, accuracy, and relevance.
- ➢ **Contextualisation**
    - ○ Generative AI can add summaries or explanations that help users understand instantaneously, so it helps users to make better decision.
- ➢ **Enhancing ETL in Data Lake**
    - ○ Generative AI can optimise ETL processes by learning from the data it processes, leading to more efficient data extraction, transformation, and loading. This results in improved data quality and operational efficiency.
- ➢ **Data Visualisation**
    - ○ Generative AI can create personalised visualisations, automating the visualisation process and allowing data scientists to focus on analysis rather than design. For example, generative AI can provide the personalised patterns of analytical results’ visuals. AI could suggest what visualisation might be available from the data in the lake.
- ➢ **Semantic Search Enhancement**

  - Generative AI can boost semantic search functionalities, allowing users to query the knowledge lake using natural language. This capability can significantly reduce the time engineers spend on contextualizing and gathering data.

b)

University Matching Recommendation System:
A smart matching system that helps prospective students find their ideal university by analyzing research outputs, faculty expertise, and institutional strengths against student preferences and career goals. This system transforms the traditional university selection process from general ranking-based decisions to personalized research alignment matches. The knowledge lake can be used to:

- Match student research interests with university strengths and faculty expertise
- Analyze university research output, impact, and specializations
- Track faculty research patterns and potential supervisor matches
- Map available resources (facilities, funding, equipment)
- Monitor industry partnerships and career placement rates
- Compare curriculum and training opportunities
- Evaluate research group dynamics and collaboration networks

For example, a Data Science student interested in "AI in Healthcare" might discover that while University A has the highest general ranking, University B actually offers better alignment to the student’s preference.

- Strong healthcare industry partnerships
- Specialized medical AI research groups
- Advanced computing facilities
- Higher healthcare tech industry placement rates
- More relevant potential supervisors

This system can change the way we see universities solely by ranking, students can now choose universities based on their interest and the university’s alignment to it. This system is crucial in today's global education landscape, where international undergrad students often make life-changing decisions and spend fortunes without complete information about which university would best support their specific research interests and career goals.

Clinical Case Note Generation:
Generate comprehensive clinical notes by combining patient-specific data with medical knowledge we discussed in the previous question. This can help doctors to narrow down the disease based on the present symptoms a patient showed. Sometimes, combination of symptoms can cause human errors as there are many reported cases of patients and their cure or death because of the non-explainable, however, this is typically some symptoms hiding a main cause. This is where this data lake can be used:

- Integrate patient symptoms, test results, and vital signs
- Link symptoms to known disease patterns and comorbidities

- Include relevant drug interactions and contraindications
- Add standard treatment protocols and guidelines
- Flag unusual combinations or high-risk factors

Systematic Discovery of Unexplored Research Directions:
A systematic approach to discover unexplored research opportunities by analyzing existing research landscapes through knowledge graphs and pattern detection. This helps researchers identify promising new research directions and avoid redundant work. The system is particularly valuable for PhD students, research institutions, and funding agencies to make informed decisions about research directions.

- Map relationships between research methods, frameworks, and outcomes
- Detect methodological gaps and unexplored variable combinations
- Identify understudied populations or contexts
- Discover opportunities for theoretical integration across fields
- Track abandoned research directions worth revisiting with new technology
- Analyze contradictions requiring resolution
- Predict emerging research trends and their potential impact