1. Using the scenario that was stated in Section 1, list the likely requirements for the database (max 500 words).

The likely requirements for the database that will be used by the research and development department of Big Four Inc. to invest in machine learning research:

1. **Scalability**- The database needs to be scalable since machine learning research creates a lot of data, including datasets, model parameters, and performance indicators. As the research develops, the database needs to be more scalable and adaptable enough to support all types of data types and fields.
2. **Flexibility**: The database needs to be flexible and adaptable so that it can support various types of data kinds, including text, photos, audio, and video as well as structured and unstructured data. The database should be simple to configure and customize to the particular requirements of the research and development department.
3. **Performance**: The database should be able to provide high performance and low latency, for data-intensive machine learning tasks such as training and inference. The database should be able to handle multiple simultaneous queries and transactions without compromising performance.
4. **Trends**: It should be having all of the latest trends that are dealt with by the topic of machine learning like “supervised”, “semi-supervised ”, ”unsupervised”, “meta-learning”, "Neural Networks", "Image Classification", "Computer Vision" etc., and should have the capability of adapting more trends as machine learning is an ever-growing field.
5. **Integration**: The database should be able to integrate with various tools and technologies that are commonly used in machine learning research, such as weka, posit Jupyter notebooks, google colab, Python libraries, and cloud computing platforms. The database should support standard data formats and APIs to facilitate data exchange and collaboration.
6. **Backup**: A highly functional Database should have the option to back up and recover important documents if they are mistakenly deleted during the execution of loss due to some other problems. This is a very necessary feature as it prevents data loss.
7. **Security**: The database needs to be secure to guard against hackers and unauthorized access to critical data and research findings. To prevent data loss, it must have reliable access control features. Data that is extremely valuable should also be encrypted.
8. **Maintenance**: The database must be maintained periodically and checked for any anomalies in the data or the data pipeline also we must check the data sources at all times to get better and correct information.

So, according to the scenario stated these are the requirements for the database which comprises various factors like Scalability, Flexibility, Performance, and Trends which needs to be updated regularly to stay relevant, Integration of various tools and systems, Backup for an emergency, tight Security, and regular maintenance when all of these are combined together they form a good database system.

2. Based on the structure of the parsed data, do you think that a relational database or a document database is most appropriate? Motivate your choice using the concepts you have learned (max 500 words).

After going through the parsed data, I think it would be a better choice to use a document database in place of a relational database

In the parsed data, there are various machine learning concepts and their definitions, tasks, and applications. These are presented in a semi-structured format with headings, subheadings, and bullet points. This kind of data structure is best suited for a document database, where each document can contain fields with relevant metadata and unstructured text for the machine learning concept. Generally, Unstructured or semi-structured data are stored and managed as documents in a document database, a type of NoSQL database that frequently uses JSON or BSON formats.

Additionally, the parsed data contains a combination of textual descriptions, definitions, and examples, which makes it difficult to create explicit links between entities. It's possible that such complicated and varied data won't fit well in a relational database because it depends on pre-set relationships and structured tables. In the document-based database, how we want to use the data dictates how we will design the schema.

Document database has characteristics like-

Scalable-It is horizontally scaled so, more storage, which is required by our database to store upcoming trends

High Availability: Often required to be available continuously and are therefore

- Data is replicated to minimize the likelihood of system failure.
- Can increase response speed as multiple nodes have the same data (write performance can be more difficult though).

Replication: Has a technique to update multiple copies of the same data.

Sharding: It is the process of splitting data over multiple nodes.

Indexing: Document databases typically have more advanced indexing capabilities than relational databases. This allows for fast queries and search operations on large amounts of data, which can be particularly useful for applications that need to perform complex queries or data analysis.

High-Performance Access: Don’t rely on complex queries to access data it uses a key-value structure which makes retrieving data much easier and simpler.

No schema necessary: Does not require data to be structured and can work great with semi-structured and unstructured data.

All of its’ characteristics combined forms the perfect way to represent our database at Big Four Incas it can handle large amounts of data which will be useful in dealing with the ever-growing trends list for machine learning, also as a required document-based database is able to deliver data at a very high frequency. So to conclude document database Big Four Inc. is an approach to deal with this data.

3. Outline the structure of your database. This should include the fields in your data, any relationships that exist etc. You can get a bit creative here, using a simple a diagram (for instance a entity-relationship model if you choose a relational database) or a schematic listing fields if you choose a document based model. Remember to choose fields that will be useful to any downstream CRUD operations you think you'd need.

I have chosen the document database as it aligns with my database needs completely

The structure of my database –

FEILDs	DATA TYPE	DESCRIPTION
filename	Object	Name of the files both article and references
date_published	datetime	Date of the article or reference we have taken
supervised	Object	Contains the value 'yes' or 'no' depending on it's presence
semi-supervised	Object	Contains the value 'yes' or 'no' depending on it's presence
unsupervised	Object	Contains the value 'yes' or 'no' depending on it's presence
meta-learning	Object	Contains the value 'yes' or 'no' depending on it's presence
count_of_refs	Object	It counts the number of references

This completely supports CRUD operations

In this schema, each row represents a single file and contains information about that file's name, publication date, and various attributes related to the file like supervised, unsupervised, semi-supervised, and meta-learning, all of the four fields have values such as 'yes' or 'no' The id field is used as a unique identifier for each document, and the count_of_refs counts the number of references in each file i.e. it is a field of an integer that tracks the number of references used in the file.

This schema has been chosen in such a way that it can support various CRUD operations

- **CREATE:** We can insert new documents containing relevant information anytime, as the year passes we can add new information
- **READ:** We can query all types of information from our documents.
- **Update:** We can update the current documents based on future revelations, like more trends can be found like 'computer vision', and 'big data' or add different fields like – author_names, publication_time, etc.
- **DELETE:** We can delete any document if they are not required any more

It will be in this format when I put it in the database

{'filename': 'article_Machine learning - Wikipedia_2023_3_13',

'date_published': Timestamp('2023-03-13 00:00:00'),

'supervised': 'Yes',

'semi-supervised': 'No',

'unsupervised': 'Yes',

'meta-learning': 'Yes', 'count_of_refs': 92}