As LLMs have become widely available and commoditized, companies that are able to leverage their data for AI applications will emerge as the true differentiators. A typical large B2B organization has three big categories of AI uses.

1. Retrieval-Augmented Generation (RAG) Systems

Since all LLMs are trained on publicly available data, the most common way to utilize the vast amounts of data within an organization is through the use of a strategy called RAG. Using RAG, companies can extract data from their databases and pass that information to the LLMs as context. However, most organizations have data lying in transactional databases, data lakes and warehouses and unstructured data in file systems. This often becomes a bottleneck for building any RAG application. Let's look at a common use case for RAG.

Use Case: Customer Support Agent

- **Description:** A customer support agent can leverage RAG systems to provide accurate and contextually relevant responses to customer queries by retrieving information from support, ticketing and documentation platforms along with and other data sources. However, this requires mixing and matching both structured and unstructured data.

Why SingleStore?

SingleStore enables companies to store structured, semi-structured and vectorized unstructured data in the same database and if the use case demands even in a single table allowing not only easy single-shot retrieval but also milliseconds response times across petabytes of data.

- **SingleStore Features Utilized:**
  - **Vector Index Building:** Efficient retrieval of relevant information based on vector similarity (exact KNN and ANN searches).
  - **Hybrid Vector Index Building and Search:** Combines vector and traditional search techniques (keyword match using Lucene under the hood) to improve retrieval accuracy.
  - **Knowledge Graph Integration:** Stores and queries knowledge graphs using SQL commands, joins, and recursive Common Table Expressions (CTEs).
  - **WASM with UDF:** Ability to create User Defined Functions (UDFs) to run within the database for AI use cases like sentiment analysis etc..
  - **Notebooks on Cloud:** Ability to vectorize data periodically or at data ingestion time and the ability to create Dashboard applications and exposing Python functions as API endpoints.

2. Information Synthesis - Semantic and Lexical Searches

Another common use case within enterprises for AI is information synthesis primarily for decision making and taking actions. However, given that information is usually spread out in structured and unstructured data typically the search misses the analytics augmentation which is critical for any decision making process.

Why SingleStore?

With SingleStore companies can run hybrid searches (both semantic and lexical searches) and augmented the data with rich analytics, for example in the use case below, the queries can define how many times something has been mentioned, or other analytics that could lead to highly specific and personalized results.

Use Case: Legal Document Search

- **Description:** A legal document search application can help lawyers and legal professionals quickly find relevant case laws, statutes, and legal documents using both semantic and lexical search techniques.
- **Features Utilized:**
    - **Java Lucene Integration:** Supports advanced full-text search features like BM25 scoring, fuzzy matching, proximity search, and boosting.
    - **Vector Search:** Allows for semantic searches, retrieving documents based on their semantic similarity to the query.
    - **Hybrid Search:** Ensures more accurate and contextually relevant results by combining vector and traditional search techniques.
    - **Analytics with Search Queries:** Analyzes search queries to improve search performance.
    - **Petabyte-Scale Data Handling:** Ensures most queries are processed in milliseconds.

3. Agentic Use Cases - Building Agents and Agentic Apps

Another use case that a number of organizations have started adopting is taking actions based on AI responses (using RAG and/or information synthesis). A common use case would be an Agent that not only retrieves and serves information to the user but takes action like creation of support ticket, updating information and sending notifications. All of these require data wrangling from different sources making the retrieval slow, expensive and untenable.

Why SingleStore?

SingleStore has connectors with several popular open source libraries like Langchain, Llamaindex, OpenAI etc making it easy to use it for building agents. In addition, the ability to do single-shot retrieval makes the overall action taking extremely fast and efficient that are especially useful for real-time use cases like customer service where there is a massive amount of fast moving data.

Use Case: Real-Time Customer Service Transformation

- **Description:** A real-time customer service agent can autonomously handle customer inquiries, provide personalized responses, and make decisions based on data, improving customer satisfaction and operational efficiency.
- **Features Utilized:**
    - **AI Chat Assistant:** Streamlines interaction and internal queries about user data.
    - **Integration with AI Frameworks:** Integrates with AI frameworks such as Google Vertex AI, AWS Bedrock, IBM's WatsonX platform, and Snowflake's Snowpark services.
    - **WASM with UDF:** Optimizes performance and allows custom function creation for AI tasks.
    - **Notebooks on Cloud:** Facilitates real-time data processing and API endpoint creation.
    - **Petabyte-Scale Data Handling:** Provides low-latency responses for agentic applications.

List of SingleStore AI Features

Vectors:

- **Native Vector Data Type:** SingleStore provides a native vector data type, allowing for efficient storage and querying of vector data. This data type supports various operations such as addition, multiplication, and sorting of vectors.
- **Vector Functions:** SingleStore includes a comprehensive set of vector functions:

    - **DOT_PRODUCT:** Computes the dot product of two vectors.
    - **EUCLIDEAN_DISTANCE:** Calculates the Euclidean distance between vectors.
    - **Cosine Similarity and Distance:** Measures the cosine similarity and distance between vectors.
    - **SCALAR_VECTOR_MUL:** Multiplies a vector by a scalar.
    - **VECTOR_ADD:** Adds two vectors.
    - **VECTOR_ELEMENTS_SUM:** Sums all elements in a vector.
    - **VECTOR_KTH_ELEMENT:** Finds the kth element in a vector.
    - **VECTOR_MUL:** Multiplies two vectors element-wise.
      VECTOR_NUM_ELEMENTS: Counts the number of elements in a vector.
    - **VECTOR_SORT:** Sorts a vector.
    - **VECTOR_SUB:** Subtracts one vector from another.
      Indexed Approximate Nearest Neighbor (ANN) Search: SingleStore supports

ANN search, which is optimized for high-performance vector similarity searches. This allows for fast retrieval of the nearest vectors to a given query vector.

- **Real-Time Vector Search:** New vectors are immediately visible to search, suitable for augmenting transactional apps with vector search.
- **Multiple indices for same vectors** - Ability to create PQ or IVF based indices for the same data to optimize for queries meant for approximation or speed.

Exact Keyword Search:
Inverted Index: An inverted index is a data structure that maps words to their locations in a database table, similar to an index at the back of a book. This allows for efficient text search operations.

- **MATCH Clause:** The MATCH clause is used to perform full-text searches. It returns a relevancy score between 0 and 1, indicating the quality of the match.
- **BM25 Scoring:** BM25 is a ranking function used to score the relevance of documents based on the frequency of query terms. It handles term saturation and document length more effectively than traditional TF-IDF methods.
- **Lucene Integration:** SingleStore's full-text search leverages the Lucene library, a widely-used open-source search engine.
- **Proximity Search:** Allows searching for words that appear within a specified number of words from each other.
- **Boosting:** Increases the relevance of certain terms in the search results using the ^ symbol.
- **JSON Search:** Full-text search can be performed over JSON documents.

WASM (WebAssembly):

- **WASM with UDF (User Defined Functions):** SingleStore supports the use of WebAssembly for creating user-defined functions, allowing for custom processing and extending the database's capabilities with high performance and security.

Notebooks in Cloud:

- **SingleStore Notebooks:** SingleStore provides a built-in notebook feature that allows users to perform live analytics, create interactive dashboards, and facilitate real-time insights and data-driven decision making. Notebooks can be used for tasks such as vectorization of data, running scheduled jobs, and building real-time dashboards. Notebooks can also be used as scheduled lambda functions and can be exposed as API endpoints for execution of code local to the data.

Futures/Roadmap

- **Disk ANN (Approximate Nearest Neighbor) Search:** Designed to improve query performance per dollar (QPS/$) by leveraging disk storage for ANN searches. This feature is particularly beneficial for handling large vector sets that do not fit entirely in memory, providing a cost-effective solution for high-performance vector similarity searches. Disk ANN is expected to be available in mid-2025.

- **Run your models in the cloud -** TBD

#EOM