CLDV6211 Cloud Development A

POE Part 2

Kundai Wonder Makuzha ST10251297

2025

Contents

[LINKS 1](#_Toc198072757)

[THEORY 1](#_Toc198072758)

[QUESTION 1 1](#_Toc198072759)

[QUESTION2 3](#_Toc198072760)

[References 4](#_Toc198072761)

# LINKS

eventeasebookingsystem1-frb7ghczg2dnebge.southafricanorth-01.azurewebsites.net

https://github.com/MakuzhaKundai/CloudDevelopmentProject.git

# THEORY

# QUESTION 1

**1 . Explain how Azure's Cognitive Search differs from traditional search engines and discuss potential use cases where Cognitive Search would offer a clear advantage. What limitations does it have, and how can they be mitigated?**

As explained earlier, Azure Cognitive Search goes beyond keyword matching through the inclusion of AI-driven features. This unlocks a variety of use cases where it has a distinct benefit over other search engines:

Potential Use Cases for Azure Cognitive Search:

E-commerce Product Search: If a customer searches for "comfortable red running shoes," an ordinary search might return only exact matches. Cognitive Search, understanding intention and characteristics, might return red running shoes with characteristics like "cushioned," "lightweight," or "breathable," even though those exact words might not have been typed into the search phrase.

Knowledge Mining from Unstructured Data: The majority of organisations have vast amounts of unstructured data like scanned documents, PDFS, images, and videos. Cognitive Search can leverage OCR to pull out text, analyse images for their content, and apply NLP techniques to identify key phrases, entities, and sentiment. This allows the user to search on this data effectively, yielding rich insights that would be impossible with legacy search. For example, a pharmaceutical firm may scour research papers, patents, and clinical trial documentation for potential drug interactions or side effects.

Customer Support Sites: Customers often have trouble finding information in FAQS or knowledge bases via plain keyword queries. Cognitive Search can understand the nuances of what they ask, even if the query is spelt wrong or not very formal and return articles or solutions. Sentiment analysis can also score high-priority topics.

Media and Entertainment Content Discovery: Scrolling through vast repositories of videos, music, or images feels very natural with Cognitive Search. Users can search for visual content (e.g., "display scenes with a blue car"), words uttered in a video, or music genre and mood.

Legal and Compliance Discovery (eDiscovery): In litigation, quick retrieval of relevant documents in massive collections is crucial. Cognitive Search can find key entities, subjects, and relationships from legal documents and thus accelerate the discovery and accuracy process.

Limitations of Azure Cognitive Search and Mitigation Strategies:

* While extremely capable, Azure Cognitive Search does have some limitations:
* Cost: The more complex the indexing is, and the higher the query rate, then the more expensive it can become. Processing of cognitive abilities also adds to the cost.
* Mitigation: Optimize the indexing plans, make use of the correct service tiers, and wisely select cognitive abilities needed. Batch processing for non-real-time enrichment tasks is an option.
* Complexity of Configuration: The configuration of index schema, cognitive skills pipeline, and scoring profiles can be complicated and requires an adequate understanding of the concepts concerned.
* Mitigation: Leverage Azure portal tools and SDKs, use the supplied documentation and samples, and consider hiring Azure specialists for initial setup and fine-tuning.
* Performance with Extremely Large Datasets: Highly scalable, but performance may be lost with extremely large and complex datasets unless properly architected and optimized.
* Mitigation: Use techniques like sharding, partitioning, and indexing judiciously. Modify query structure and leverage caching mechanisms.
* Accuracy of Cognitive Capabilities: Accuracy of cognitive capabilities driven by AI (like entity recognition or sentiment analysis) is not always hundred percent and depends on the nature and quality of the data.
* Mitigation: Fine-tune cognitive capabilities where feasible, combine multiple capabilities together for greater accuracy, and incorporate human review procedures for significant information.
* Language Support for Cognitive Capabilities: Not all cognitive capabilities support all languages to the same extent.
* Mitigation: Notice the language support for the skills being utilized and potentially use language-specific pipelines or translations. (Anon., n.d.)

# QUESTION2

**2. Why is database normalisation important in cloud-based database design? Discuss the impact of both normalized and denormalised structures on performance and scalability in a cloud environment like Azure.**

**As previously mentioned, database normalization plays a significant role in data integrity and reducing redundancy, which translates to several benefits in the cloud: cost savings due to reduced storage, simplified data management, and a solid foundation for scalability.** (Azure, n.d.)

**Impact of Normalised Structures on Performance and Scalability in Azure**

**Performance: Databases with high normalisation will often require more joins to access all data, while in a cloud platform like Azure SQL Database or Azure Database for Postgresql, though hardware underneath is powerful, too many joins can nevertheless adversely affect query performance, especially for complex queries or high-transaction workloads. Properly executed indexing techniques, query optimisation techniques, and the use of read replicas will keep such performance issues at bay.**

**Scalability: Normalized structures may be better in terms of write scalability because updates are localized to a specific table, reducing the number of contention possibilities. However, read scalability might need to be addressed with techniques like read replicas to distribute read traffic across multiple instances. Sharding or partitioning methods might also be necessary for very big, highly normalized databases in order to distribute data and workload across multiple database instances.**

**Impact of DenorDenormalisationmalized Structures on Performance and Scaling in Azure:**

**Performance: Denormalization, or data duplication or data consolidation into fewer tables, can significantly improve read performance by performing fewer or no joins to retrieve the data needed. It can be very helpful for read-intensive workloads common in many cloud applications. In Azure Cosmos DB, a NoSQL database for instance, putting related data within a single document is an example of common denormalisation to reduce read latency.**

**Scalability: Denormalisation may improve reading scalability by keeping join operations on spread data to a minimum across servers but reduce write scalability. Updates can become potentially updating multiple places in data and raise the risk of inconsistencies and become performance bottlenecks for extremely heavy write usage. Furthermore, data redundancy results in higher costs of storage. In Azure Cosmos DB, although it has good horizontal write scalability, proper schema design is necessary to avoid "hot partitions" because of skewed data distribution because of denormalization.**

**Things to consider in Azure:**

**In a cloud environment like Azure, the choice between normalization and denormalization is not necessarily always an easy one. Azure has several database offerings with varying strengths and capabilities:**

**Relational Databases (Azure SQL Database, Azure Database for PostgreSQL/MySQL): These are appropriate for applications that require strong consistency and high-level transactions. While normalization is typically in order for transactional systems, there can be some denormalization considered for some read-heavy scenarios to improve performance, but this must be handled judiciously.**

**NoSQL Databases (Azure Cosmos DB, Azure Cache for Redis): These services favor less strict schemas and denormalization in order to offer high availability, horizontal scalability, and low latency for specific data access patterns. Azure Cosmos DB, by virtue of supporting multiple data models (document, key-value, graph, column-family), provides tailored denormalization strategies. Azure Cache for Redis, being a key-value store, has an inherent denormalized representation for caching frequently used data.**

**In conclusion the optimal database design in Azure, or indeed any cloud platform, typically involves a pragmatic approach. Although normalization provides valuable benefits in terms of data integrity and manageability, denormalization can be used strategically to optimize read performance for specific use cases. The key is to understand the application requirements, data access patterns, and characteristics of the chosen cloud database service to obtain the right balance between normalization and denormalization for optimal performance, scalability, and cost-effectiveness**

# References

Anon., n.d. *Azure Cognitive Search Documentation: The official Microsoft documentation is the most comprehensive resource..* [Online]   
Available at: https://learn.microsoft.com/en-us/azure/search/

Azure, n.d. *Best Practices for Azure Database for PostgreSQL:.* [Online]   
Available at: https://www.google.com/search?q=https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-best-practices

Vanier, E., n.d. *Database Optimization Articles, SQL Database.* [Online]   
Available at: https://ericvanier.com/performance-optimization-in-mariadb-key-considerations/