2025

**NAME: MASEO JUNIOR**

**SURNAME: MANDIWANA**

**STUDENT NO.: 10367488**

**MODULE: CLOUD DEVELOPMENT 1A**

**MODULE CODE: CLDV6211**

**ASSESSMENT 2**

## Table of Contents

1. **Differences Between Azure Cognitive Search and Traditional Search Engines**

- **AI-Powered Content Understanding:**
  Traditional search engines primarily rely on keyword matching and exact text queries. Azure Cognitive Search enhances this by integrating AI capabilities such as natural language processing (NLP), image recognition, and entity extraction. This means it can understand the context and meaning behind queries and documents, not just match keywords (Microsoft, 2021).

- **Hybrid Search Capabilities:**
  Azure Cognitive Search supports both keyword-based and vector-based (semantic) search. This allows it to find results based on semantic similarity, even when the exact words don't match. Traditional engines usually lack this semantic understanding (Boué, 2023).

- **Data Enrichment and Indexing:**
  Azure Cognitive Search can ingest and enrich unstructured data (PDFs, images, scanned documents) using built-in cognitive skills (OCR, language detection, sentiment analysis). Traditional engines typically require pre-processed, structured data (Chukwura, 2023).

- **Cloud-Native Scalability:**
  Azure Cognitive Search is a fully managed cloud service that automatically scales to handle large volumes of data and queries, without requiring manual infrastructure management (Microsoft Cognitive Services Research, n.d.).

**Use Cases Where Azure Cognitive Search Offers Clear Advantages**

- **Enterprise Knowledge Mining:**
  Extracting insights from large repositories of unstructured documents (legal contracts, research papers, customer feedback) where semantic understanding improves search relevance.

- **E-Commerce Search:**
  Enabling faceted and personalized search experiences that understand user intent, such as filtering products by attributes and interpreting natural language queries.

- **Healthcare and Compliance:**
  Searching sensitive medical records with built-in AI to identify and redact protected health information (PHI) while maintaining searchability.

- **Conversational AI and RAG (Retrieval-Augmented Generation):**
  Enhancing chatbots and virtual assistants by grounding responses in up-to-date, domain-specific data indicated by Cognitive Search.

**Azure Cognitive Search faces several limitations that can be mitigated through strategic approaches:**

**Latency in complex AI processing** often arises from resource-intensive operations like partial term searches or high skip values. To address this, using pre-built cognitive skills instead of custom models reduces computational overhead, while query caching accelerates frequent requests by storing results.

For **cost management at scale**, proactive monitoring via Azure Monitor combined with auto-scaling ensures optimal resource allocation, preventing overprovisioning. Selecting appropriate pricing tiers and adding partitions during high demand further enhances cost-efficiency by parallelizing workloads.

The **complexity of AI skill integration** is streamlined through Azure's pre-configured templates and SDKs, which simplify deployment and reduce development time. For instance, semantic search configurations improve relevance without requiring custom code, particularly for structured prose.

Finally, **limited support for certain data types** is resolved by preprocessing unsupported formats (e.g., raw text extraction from images) before ingestion or leveraging Azure Data Factory for ETL pipelines, ensuring compatibility with Cognitive Search's indexing capabilities. These strategies collectively enhance performance while maintaining scalability and cost control.

## 2. Importance of Database Normalization in Cloud-Based Database Design and Impact on Performance and Scalability in Azure

### Why Database Normalization Is Important in Cloud Environments

- **Data Integrity and Consistency:**
  Normalization reduces data redundancy and ensures that each piece of information is stored only once, minimizing anomalies during inserts, updates, or deletes (Microsoft Cognitive Services Research, n.d.).
- **Cost Efficiency:**
  Cloud storage costs can add up quickly. Normalized databases consume less storage by avoiding duplicate data, reducing overall cloud storage expenses (Bhadja & Bhattacharjee, 2024).
- **Simplified Maintenance:**
  Changes to data structures or business rules are easier to implement when data is logically organized, which is especially important in dynamic cloud environments.
- **Improved Transactional Performance:**
  Normalized tables with smaller rows and fewer redundant columns improve write

performance, critical for OLTP (online transaction processing) workloads in Azure SQL Database or Cosmos DB (Bhadja & Bhattacharjee, 2024).

**Normalized and denormalized database structures affect performance and scalability in fundamentally different ways, each with distinct trade-offs suited for different use cases.**

**Query Performance:**
Normalized databases organize data into multiple related tables to minimize redundancy. As a result, queries often require multiple JOIN operations to retrieve related data, which can slow down read performance, especially for complex queries. In contrast, denormalized databases consolidate related data into fewer tables or documents, reducing or eliminating the need for joins. This simplification leads to faster read queries, making denormalization ideal for read-heavy applications such as reporting or dashboards (Couchbase, 2025; Pure Storage, n.d.).

**Write Performance:**
Normalized structures typically offer faster write operations because data is stored in smaller, focused tables with minimal duplication. Updates and inserts affect fewer records, reducing overhead. Denormalization introduces redundancy by duplicating data across tables or documents, so write operations must update multiple copies of the same data. This increases the complexity and cost of writes, making them slower and more error-prone if not carefully managed (CodiLime, 2024; HarperDB, 2024).

**Storage Costs:**
Normalization reduces storage requirements by eliminating duplicate data, leading to more efficient use of disk space. Denormalization increases storage needs due to data duplication and precomputed aggregates stored alongside base data, which can significantly raise storage costs, especially at scale (CelerData, 2024.

**Scalability:**
Normalized databases are easier to scale horizontally in distributed systems, such as partitioning data by keys in NoSQL databases like Azure Cosmos DB. This is because normalized data can be cleanly partitioned without duplicating large amounts of data. Denormalized data, particularly in document stores, can be limited by document size constraints and the complexity of managing redundant data across partitions.

However, denormalization can improve scalability in read-heavy workloads by reducing query complexity and load on the database engine (Couchbase, 2025; HarperDB, 2024).

**Complexity:**
Normalized schemas tend to be more complex, requiring careful design of relationships and queries involving multiple joins. This complexity can increase development and maintenance effort but ensures data integrity and consistency. Denormalized schemas simplify query logic by reducing joins and flattening data structures but increase the risk of data anomalies, inconsistencies, and more complex update logic (Pure Storage, n.d.; CodiLime, 2024).

In conclusion normalized databases prioritize data integrity, efficient storage, and write performance but may suffer slower read queries due to joins. Denormalized databases optimize read performance and query simplicity by duplicating data, increasing storage costs, and complicating writes. The choice depends on workload patterns: transactional systems benefit from normalization, while analytical and read-heavy systems often gain from denormalization (CelerData, 2024; Couchbase, 2025).

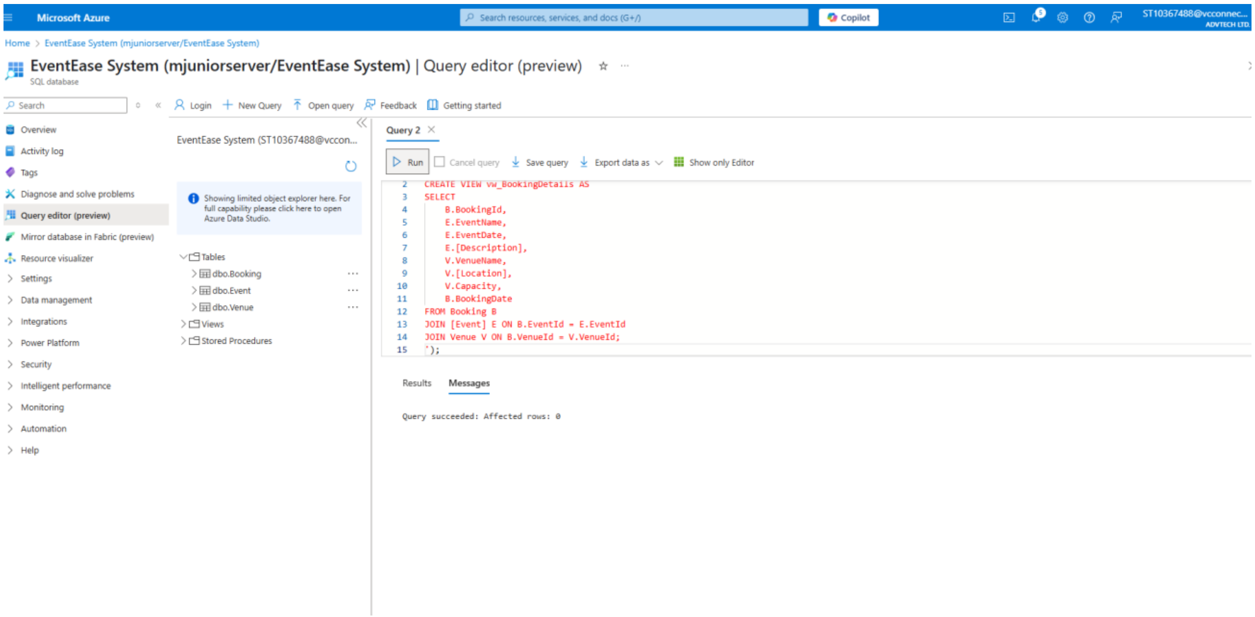
### Specific Considerations in Azure

- **Normalized Databases (e.g., Azure SQL Database):**
  Ideal for transactional systems requiring ACID compliance, such as banking or inventory management. Supports efficient scaling and cost control by minimizing data duplication (Microsoft Cognitive Services Research, n.d.).
- **Denormalized Databases (e.g., Azure Synapse Analytics, Cosmos DB):**
  Suited for read-heavy analytical workloads where query speed is critical, such as reporting dashboards or big data analytics. Reduces join operations but increases storage and update complexity (Bhadja & Bhattacharjee, 2024).

- **Hybrid Approaches:**
  Use normalized schemas for transactional data, and denormalize selectively for analytics or caching layers. For example, materialized views in Azure SQL can pre-aggregate data, balancing performance and integrity (Microsoft, 2021).

### Best Practices

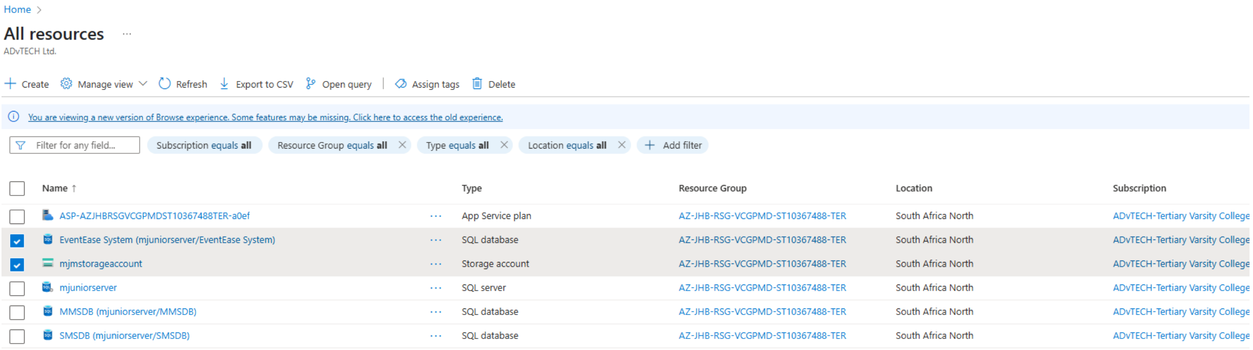
- Start with normalization to ensure data integrity and cost efficiency.
- Profile workloads using Azure Monitor to identify read-heavy queries that may benefit from denormalization.
- Use Azure Synapse or materialized views for analytical workloads requiring denormalized structures.
- Leverage caching (e.g., Azure Redis Cache) to improve performance without compromising normalization.


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
ST10367488@vcconnec...
ADVTECH LTD.
Home > EventEase System (mjuniorserver/EventEase System)

EventEase System (mjuniorserver/EventEase System) | Query editor (preview)

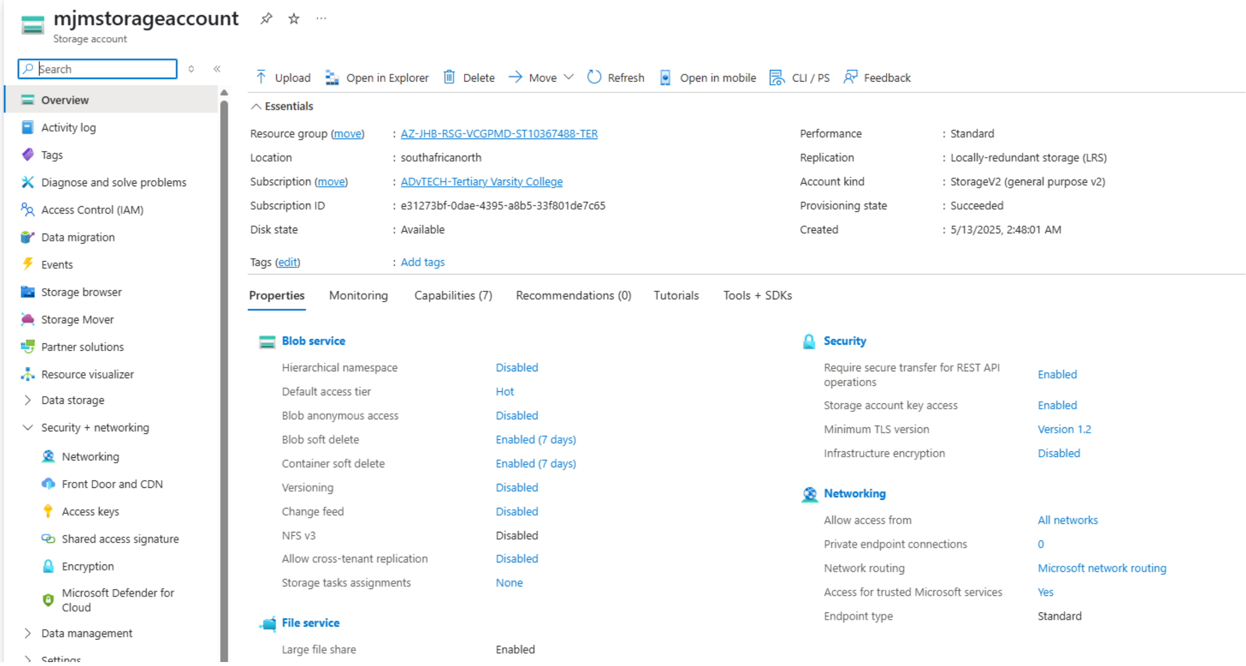
```
CREATE VIEW vw_BookingDetails AS
SELECT
    B.BookingId,
    E.EventName,
    E.EventDate,
    E.[Description],
    V.VenueName,
    V.[Location],
    V.Capacity,
    B.BookingDate
FROM Booking B
JOIN [Event] E ON B.EventId = E.EventId
JOIN Venue V ON B.VenueId = V.VenueId;
');
```

```
Query succeeded: Affected rows: 0
```



All resources

| □ | Name ↑ | Type | Resource Group | Location | Subscription |
|---|---|---|---|---|---|
| □ | ASP-AZJHBRSGVCGPMDST10367488TER-a0ef | App Service plan | AZ-JHB-RSG-VCGPMD-ST10367488-TER | South Africa North | ADvTECH-Tertiary Varsity College |
| ☑ | EventEase System (mjuniorserver/EventEase System) | SQL database | AZ-JHB-RSG-VCGPMD-ST10367488-TER | South Africa North | ADvTECH-Tertiary Varsity College |
| ☑ | mjmstorageaccount | Storage account | AZ-JHB-RSG-VCGPMD-ST10367488-TER | South Africa North | ADvTECH-Tertiary Varsity College |
| □ | mjuniorserver | SQL server | AZ-JHB-RSG-VCGPMD-ST10367488-TER | South Africa North | ADvTECH-Tertiary Varsity College |
| □ | MMSDB (mjuniorserver/MMSDB) | SQL database | AZ-JHB-RSG-VCGPMD-ST10367488-TER | South Africa North | ADvTECH-Tertiary Varsity College |
| □ | SMSDB (mjuniorserver/SMSDB) | SQL database | AZ-JHB-RSG-VCGPMD-ST10367488-TER | South Africa North | ADvTECH-Tertiary Varsity College |



| | |
|---|---|
| Resource group (move) | : AZ-JHB-RSG-VCGPMD-ST10367488-TER |
| Location | : southafricanorth |
| Subscription (move) | : ADvTECH-Tertiary Varsity College |
| Subscription ID | : e31273bf-0dae-4395-a8b5-33f801de7c65 |
| Disk state | : Available |
| Tags (edit) | : Add tags |

| | |
|---|---|
| Performance | : Standard |
| Replication | : Locally-redundant storage (LRS) |
| Account kind | : StorageV2 (general purpose v2) |
| Provisioning state | : Succeeded |
| Created | : 5/13/2025, 2:48:01 AM |

**Blob service**

| | |
|---|---|
| Hierarchical namespace | Disabled |
| Default access tier | Hot |
| Blob anonymous access | Disabled |
| Blob soft delete | Enabled (7 days) |
| Container soft delete | Enabled (7 days) |
| Versioning | Disabled |
| Change feed | Disabled |
| NFS v3 | Disabled |
| Allow cross-tenant replication | Disabled |
| Storage tasks assignments | None |

**File service**

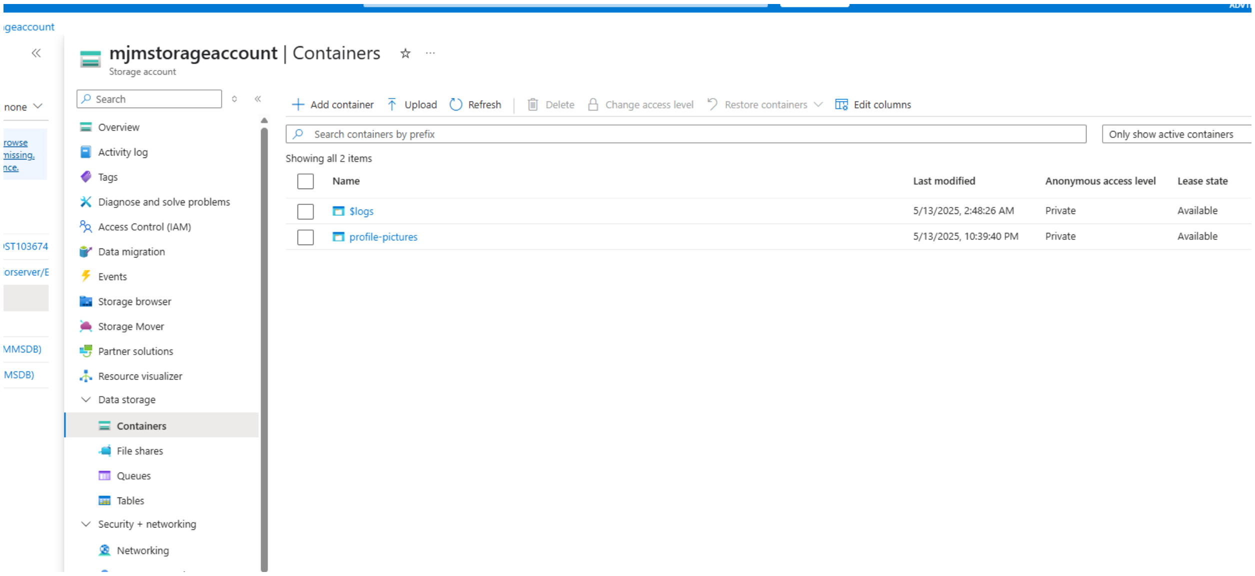
| | |
|---|---|
| Large file share | Enabled |

**Security**

| | |
|---|---|
| Require secure transfer for REST API operations | Enabled |
| Storage account key access | Enabled |
| Minimum TLS version | Version 1.2 |
| Infrastructure encryption | Disabled |

**Networking**

| | |
|---|---|
| Allow access from | All networks |
| Private endpoint connections | 0 |
| Network routing | Microsoft network routing |
| Access for trusted Microsoft services | Yes |
| Endpoint type | Standard |



Showing all 2 items

| | Name | Last modified | Anonymous access level | Lease state |
|---|---|---|---|---|
| □ | $logs | 5/13/2025, 2:48:26 AM | Private | Available |
| □ | profile-pictures | 5/13/2025, 10:39:40 PM | Private | Available |

## Reference List

Boué, L. (2023) ‘Improving search relevance of Azure Cognitive Search by Bayesian optimization’, *arXiv*. Available at: https://arxiv.org/abs/2312.08021 (Accessed: 10 May 2025).

Bhadja, B. and Bhattacharjee, S. (2024) ‘Azure AI Services: A Comprehensive Research Paper’, *International Journal of Novel Research and Development*. Available at: https://www.ijnrd.org/papers/IJNRD2403346.pdf (Accessed: 10 May 2025).

Chukwura, I. (2023) ‘Exploring Azure Cognitive Search’s Power’, *C# Corner*, 2 September. Available at: https://www.c-sharpcorner.com/article/exploring-azure-cognitive-searchs-power/ (Accessed: 10 May 2025).

Microsoft (2021) ‘Research talk: The science behind semantic search: How AI from Microsoft Bing is powering Azure Cognitive Search’, *Microsoft*. Available at: https://www.microsoft.com/en-us/research/video/research-talk-semantic-search-science-how-microsoft-bing-ai-is-powering-azure-cognitive-search/ (Accessed: 11 May 2025).

Microsoft Cognitive Services Research (n.d.) *Azure Cognitive Services Research: Publications*. Available at: https://www.microsoft.com/en-us/research/group/cognitive-services-research/publications/ (Accessed: 11 May 2025).

CelerData (2024) ‘Normalization vs Denormalization: The Trade-offs You Need to Know’. Available at: https://celerdata.com/glossary/normalization-vs-denormalization-the-trade-offs-you-need-to-know (Accessed: 11 May 2025).

CodiLime (2024) ‘Normalization vs. Denormalization in Databases’, *CodiLime Blog*. Available at: https://codilime.com/blog/normalization-vs-denormalization-in-databases/ (Accessed: 11 May 2025).

Couchbase (2025) ‘Data Normalization vs. Denormalization Comparison’, *Couchbase Blog*. Available at: https://www.couchbase.com/blog/normalization-vs-denormalization/ (Accessed: 13 May 2025).

HarperDB (2024) ‘Normalized vs. Denormalized Lookup Speed Comparison’, *HarperDB*. Available at: https://www.harperdb.io/resources/normalized-vs-denormalized-lookup-speed-comparison (Accessed: 13 May 2025).

Pure Storage (n.d.) ‘Denormalized vs. Normalized Data’, *Pure Storage Blog*. Available at: https://blog.purestorage.com/purely-educational/denormalized-vs-normalized-data/ (Accessed: 13 May 2025).