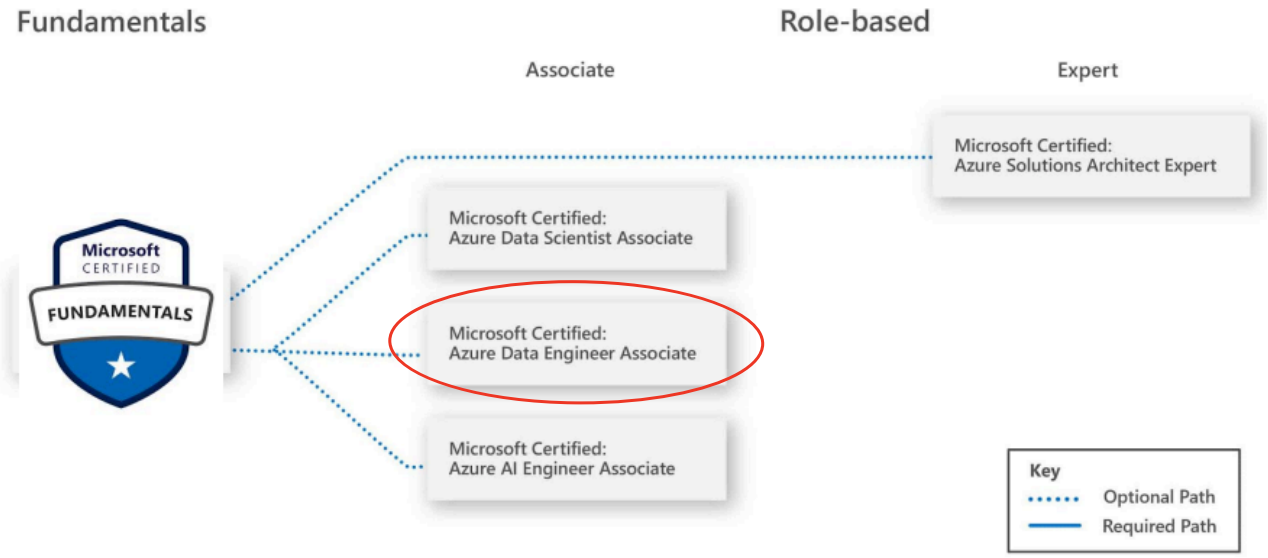
DP 900 - AZURE FUNDAMENTALS



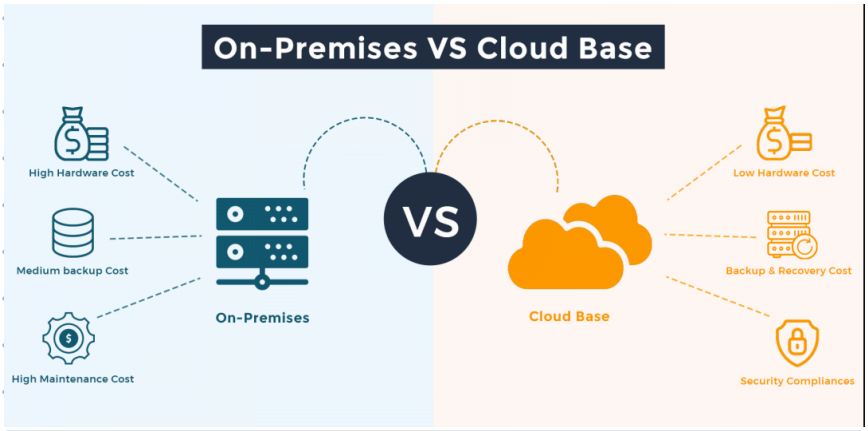
CORE TOPICS;

- Basic Terminology
- Core Data Concepts 15 - 20%
- Relational Data 25 - 35 %
- Non Relational Data 25 - 30 %
- Data Analytics 25 - 30 %

BASIC TERMINOLOGY

- Cloud computing
- Types of cloud
- Scaling Types
- Introduction to Azure
- Azure Services

Cloud Computing means delivering computing services over the internet



Aspect	On-Premise	Cloud
Analogy	Buying an office	Renting an office
Problems	Space, high initial investment, maintenance	No space issues, no infrastructure management
Payment Model	High upfront costs (CapEx)	Pay-as-you-go (OpEx)
Advantages	Full control, no reliance on external providers	Scalability, flexibility, fault tolerance, easy access
Operating Costs	High upfront costs + ongoing maintenance (CapEx)	Subscription-based costs (OpEx)
Example	IBM, internal data centers	AWS, Azure, Google Cloud
Deployment	Hosted within the organization's facilities	Hosted on provider's servers, accessed online
Scalability	Limited by hardware; requires new equipment	Instant scaling on-demand
Maintenance	Managed internally by the organization	Managed by cloud provider (AWS, Azure, GCP)
Flexibility	Fixed capacity, limited dynamic adjustments	Highly flexible, resources scale up/down easily
Security	Greater control, data on-premises	Managed by provider, meets compliance but may raise concerns for sensitive data
Disaster Recovery	Requires internal systems	Built-in disaster recovery and redundancy

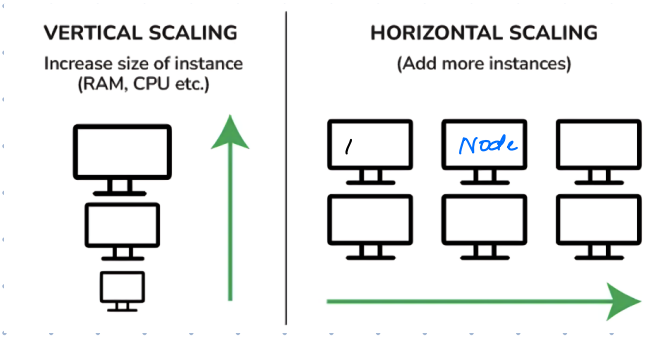
Cloud Benefits

- Scalability
  - processing
  - storage
- Reliability (Fault Tolerance)
- Cost Effective
- Security

Types of Cloud

- Public Cloud - AWS, AZURE, GCP
- private cloud - IBM Cloud, Vmware (One organisation)
- Hybrid Cloud - mix of public + private (AWS) (on premise)

Scaling Types



Vertical - monolythic Architecture

Horizontal - Distributed Architecture

AZURE is Microsoft's cloud computing platform which provides services;

SERVICES;

- Compute = VM's
- Storage = Data lake, Blob Storage
- Databases = SQL DB, Cosmos DB (NO SQL DB)
- Networking = Vnet, Load Balancer
- Data & AI = Data Factory, Synapse Databricks

DATA LAKE vs DW vs DB

Aspect	Messy Data Lakes	Analytics Data Warehouses	Transactions Databases
Data Type	Structured, Semi-structured, Unstructured	Structured (aggregated/processed for analysis) Semi Structured	Structured (relational)
Purpose	Storage of raw data for future processing	Analytical (reporting and decision-making)	Transactional (day-to-day operations)
Data Structure	Raw and unprocessed (can be messy)	Denormalized (optimized for analysis)	Normalized (split into tables)
Speed	High volume storage, but slower queries	Fast querying for complex analytics (Read)	Fast read/write operations
Use Case	Storing massive amounts of raw, diverse data	Aggregated data for reporting and BI	Handling real-time data like customer orders

Note; Azure DW primarily handles structured but can query semi structured Data

- Semi Structured Data is usually stored in Data Lake, accessed using external tables or polybase.

BLOB — Binary Large Object.

Core Data Concepts

- Represent Data
  - Features of Data (Variety)
- options for Data Storage
  - Formats for Data files
  - Types of Databases
- Common Data workloads
  - Transactional (OLTP)
  - Analytical (OLAP)
- Roles and Responsibilities

Types of Analytics

Descriptive — what happened

Summary of existing Data

example; Today's Sales

Diagnostic — why it happened

Deep diving to understand cause

example; Sales by State, Gender

Predictive; what happens in future

Based on past Trends

example; weather prediction

Prescriptive what to do?

Advice on best approach

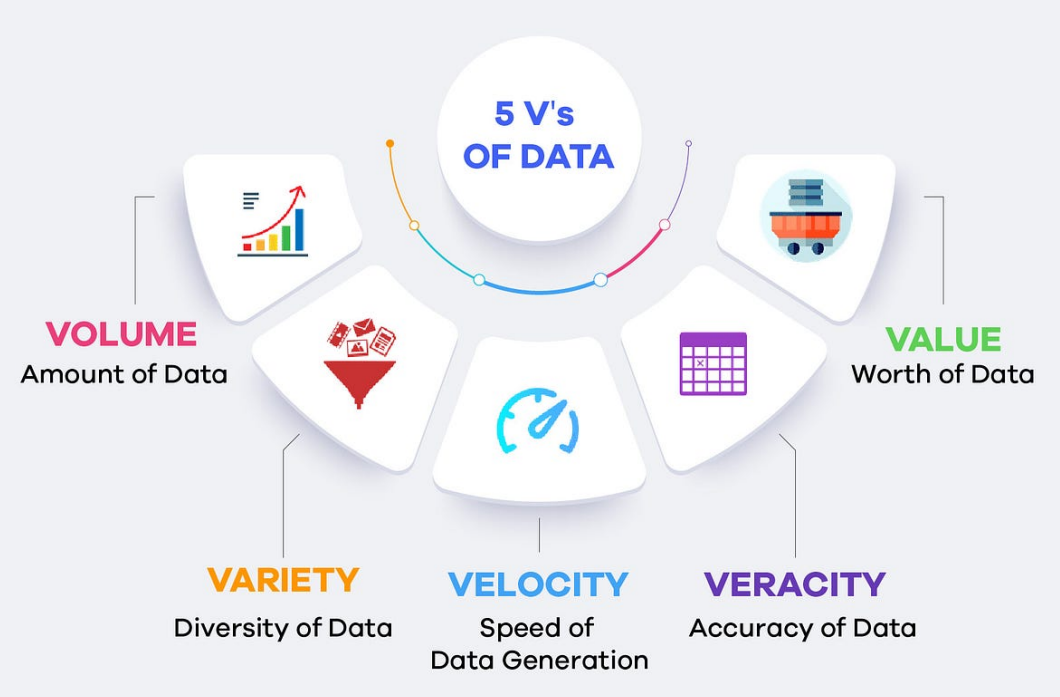
example; Movie Recommendations based on likes of user

Cognitive AI & ML

Learn & Improve over time

example; Self Driving Cars.

DATA – Raw facts, Images etc



1. Volume

The total amount of data generated, stored, and processed, ranging from gigabytes to petabytes and beyond.

Aspect	Definition
Small-scale Data	Manageable datasets that can be processed on a single machine (e.g., local databases, small CSV files).
Large-scale Data	Massive datasets requiring distributed computing (e.g., big data frameworks like Hadoop, Spark).

2. Variety

The different types and formats of data.

Type	Definition
Structured Data	Data organized in fixed formats, such as tables (e.g., SQL databases, spreadsheets).
Unstructured Data	Data without a predefined structure, such as images, videos, or social media posts.
Semi-structured Data	Data with some structure, often stored in formats like JSON or XML files.

3. Velocity

The speed at which data is generated, processed, and analyzed.

Type	Definition
Batch Processing	Data processed in large sets or chunks over a specific time frame (e.g., ETL jobs).
Real-time Data	Data processed immediately as it arrives (e.g., stock prices, live tracking).
Streaming Data	Continuous, real-time data flow from sensors or IoT devices.

→ Data at Rest (Blob)

→ continuously flowing (Event houses)

4. Veracity

The accuracy, quality, and reliability of data, ensuring it is trustworthy for decision-making.

Type	Definition
Accurate Data	Reliable, precise, and error-free data that can be confidently used for analysis.
Biased Data	Data that may contain inaccuracies or distortions due to human or systemic biases.
Noisy Data	Data with irrelevant or misleading information that can obscure meaningful insights.

5. Value

The usefulness of data in generating insights, making informed decisions, and driving business impact.

Type	Definition
Business Insights	Actionable insights derived from data, helping organizations optimize operations and strategy.
Predictive Models	Data-driven algorithms that forecast future trends or outcomes.
Optimization	Using data to enhance efficiency, improve processes, and maximize performance.

Types Of Data (Variety)

📌 Structured vs. Semi-Structured vs. Unstructured Data

Data can be classified into three types based on its format and organization:

Type	Definition	Examples	Storage & Querying
Structured Searchable	Organized in a predefined schema (tables, rows, columns).	SQL Databases (MySQL, PostgreSQL, Azure SQL).	Stored in relational databases, queried with **SQL**.
Semi-Structured	Has some structure but is not strictly tabular. Uses tags or keys.	JSON, XML, CSV, NoSQL (MongoDB, CosmosDB).	Stored in **NoSQL databases, data lakes**, queried with **SQL + special parsers**.
Unstructured	No predefined format, difficult to categorize.	Images, Videos, PDFs, Sensor data, Emails.	Stored in **Blob Storage, Data Lakes**, analyzed using **AI, NLP, and Data Processing tools**.

In relational DB, schema is a structure (Tables, Relations, Constraints) etc.

🔷 Quick Analogy

- **Structured Data** = A well-organized **Excel sheet** with rows & columns.
- **Semi-Structured Data** = A **JSON file** with key-value pairs, flexible but organized.
- **Unstructured Data** = A **folder of random images & videos**, requiring AI for analysis. → AI.

💡 Key Takeaway:

Structured data is easy to query, while **semi-structured and unstructured data need specialized tools like NoSQL, Data Lakes, and AI/ML for analysis.** 🚀

Features Of Data (Variety)

Feature	Structured Data	Semi-Structured Data	Unstructured Data
Format	Fixed schema (tables, rows, columns)	Flexible schema (JSON, XML, CSV)	No predefined structure (videos, images, PDFs)
Storage	Relational databases (SQL, PostgreSQL)	NoSQL databases (MongoDB, Cosmos DB)	Blob storage, data lakes
Querying	SQL-based (Fast & efficient)	Partially structured queries (NoSQL, API-driven)	Requires advanced processing (ML, NLP)
Scalability	Vertical scaling (Add CPU/RAM)	Horizontal scaling (Distributed DBs)	High scalability (Cloud storage, Big Data)
Use Case	Banking, ERP, CRM systems	Web data, IoT logs, emails	Images, videos, audio, social media content

Choose Right Storage

Key factors — Search, Perform, Cost

Frequent Searching	**Structured**
High Performance	**Structured/Semi-Structured**
Cost-Effective	**Unstructured**
Big Data & Analytics	**Semi-Structured/Unstructured**

Azure SQL DB
Azure Ultra Disk
Azure Blob
ADLS, Synapse

OLTP vs OLAP (Relational Workloads)

we have Transactional & Analytical

OLTP — Database

- Day to Day Operations
- Azure SQL DB
  Cosmo DB

OLAP — Warehouse

- Reports
- Deep Analytics
- Azure Synapse
  AAS

Feature	OLTP (Online Transaction Processing) Simple	OLAP (Online Analytical Processing) Historical
Purpose	Real-time transactional processing	Complex queries, reporting, and analytics
Data Type	Structured, operational data	Structured & semi-structured, historical data
Operations	Frequent INSERT, UPDATE, DELETE	Aggregations, complex joins, historical analysis
Performance	Optimized for fast read/write	Optimized for complex queries & analytics
Storage	**Azure SQL Database, Azure Cosmos DB**	**Azure Synapse Analytics, Azure Data Lake Storage (ADLS)**
Normalization	Highly normalized (reduces redundancy)	Denormalized (improves query performance)
Data Size	Smaller (GBs to TBs)	Larger (TBs to PBs)
Example Use Case	Processing e-commerce orders, banking transactions	Sales trend analysis, financial reporting, business intelligence

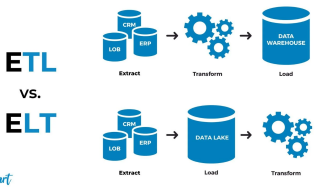
OLTP
↳ Database

OLAP
↳ Data warehouses

Q4 When to use Data Warehouse?

- when queries are long (complex Analytics)
- Data from multiple sources (Structured, Semi)
- when Data need further ETL or ELT

ETL vs ELT



Aspect	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Processing	Data is transformed before loading.	Data is transformed after loading.
Location	Traditional data warehouses with limited compute power.	Modern data lakes or cloud platforms with high compute power.
Data Type	Structured data.	Structured, semi-structured, unstructured data.
Speed	Slower, as transformations occur beforehand.	Faster, as transformations happen after loading.
Use Case	Bank transactions (cleaned before storage).	Social media analysis (raw data stored for future processing).

Common Data Roles & Responsibilities

Database Admin (DBA) - Guardian of DB's

- Manages Databases
- Availability / performance
- Assign permissions
- Manage Data backups
- Handles Restores. (Disaster Recovery)

Data Engineer Build pipelines to ensure enough data is there for Analytics / AI

- Data Migration (without an Error)
- Data Cleaning Routines
- Apply Data Governance (privacy & GDPR) - policies
- Import / Export Data (pipelines)

Data Analyst Analyze the Data.

- Create value from Data
- EDA
- Reports / visualize

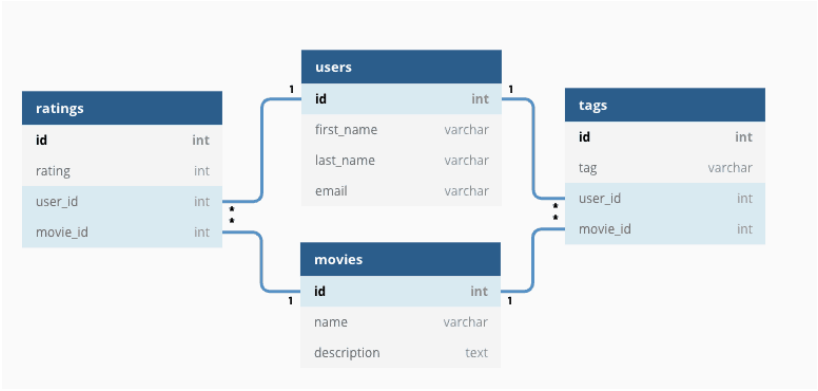
Other Roles

- Data Scientist
- Data Architect
- Application Dev
- Software Engineer

Relational Storage

- Features
- Normalization
- Azure Services
- Azure SQL Database
- Database Security
- Query Tools.

RELATIONAL DATABASE stores data in form of Tables which are related to each other using common Column based on primary & foreign key.



FEATURES;

1. **Primary Key**; Unique, No Null
2. **Foreign Key**; Contains duplicates but references to primary key
3. **Composite key**; Combination of 2 or more keys to form a primary key
3. **Index**; allows for efficient retrieval of Data in SQL Queries
   - primary key is Default Index
   - we can define other indexes.

   example - Put an Index on **Last Name**
4. **View**; are just virtual table based on other Real tables,
   - they do not store data
   - Any changes in View won't change main table

   Privacy No access to underlying Tables

NORMALIZATION; is the process of breaking down large tables into small one's to

- avoid Data Redundancy.
- maintain Data Integrity.
- to avoid Insert, Update, Delete anomalies.

Normal Forms; 1NF, 2NF, 3NF, BCNF etc.

1NF - Atomic values

example

123 ✓		123	124 ✗

2NF - No partial Dependency

example

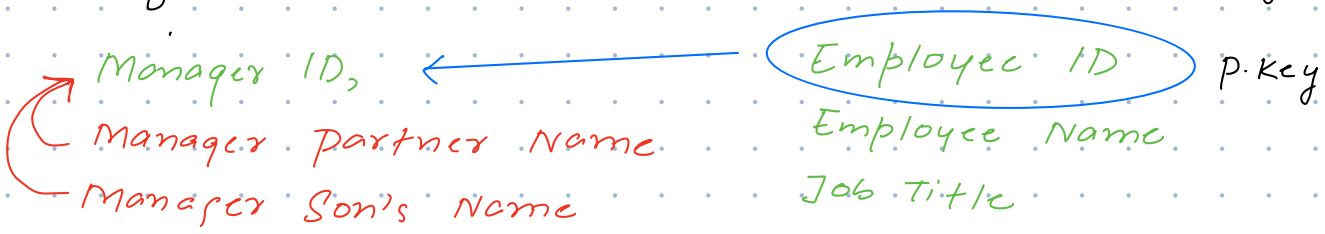
Primary Key (Composite)

here, products are only dependent on half of primary key (partial Dependence)

Solution; Remove any data that is not related to entire primary key

3NF - No Transitive Dependency

All of Data field must be related to P.key



Manager's partner, son is not dependent on Emp ID

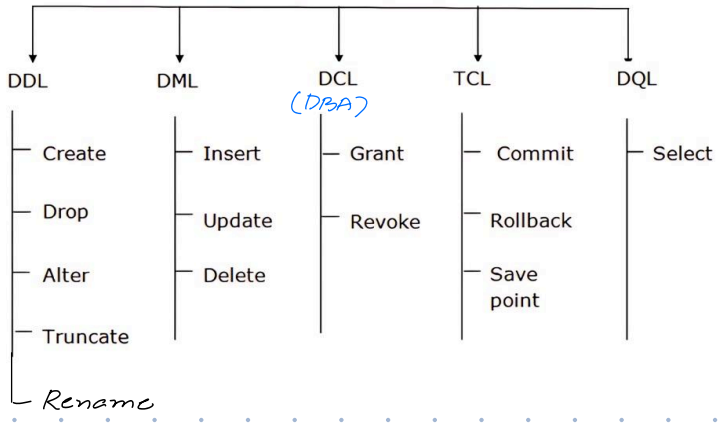
Solution; Remove any data that is not related to P.key

SQL – case insentive Microsoft - TSQL

is used to interact with databases

→ enabled by SQL

SQL COMMANDS



CREATE VIEW

```
CREATE VIEW [MyView] AS
SELECT CustomerID, Sales FROM dbo.Sales
```

CREATE STORED PROCEDURE

```
-- Create Stored Procedure
CREATE PROCEDURE [SELECTCITYP]
@param varchar(20)
AS
SELECT * FROM [SalesLT].[Address]
WHERE CITY = @param
GO

-- Execute Stored Procedure
EXEC SELECTCITYP @param = 'Kashmir'
```

→ Create Parameter

CREATE INDEX – efficient Data Retreival

```
CREATE INDEX IX_CITY
ON SALESLT.ADDRESS (CITY);

SELECT * FROM SALESLT.ADDRESS
WHERE City = 'Kashmir'
```

Note; once index created on city, rather checking 'Kashmir' in all rows, it will have index set for it (which makes data retreival Quick).

we wont create index for all columns because it also takes computation power

Azure Services

Category	Definition	Azure Services Examples
IaaS (Infrastructure as a Service) VM's, Blob	Provides virtualized computing resources over the cloud, giving control over OS, storage, and networking.	- **Azure Virtual Machines** (VMs) - **Azure Blob Storage** - **Azure Virtual Network** - **Azure Load Balancer** - **Azure Kubernetes Service (AKS)** (partially)
PaaS (Platform as a Service) ↙ To Deploy (Azure)	Offers a managed platform for developing, running, and managing applications without dealing with infrastructure.	- **Azure App Services** - **Azure SQL Database** - **Azure Functions** - **Azure Logic Apps** - **Azure Synapse Analytics**
SaaS (Software as a Service) 365, Power BI	Fully managed software applications hosted in the cloud, accessible via the internet.	- **Microsoft 365 (Office 365)** - **Azure DevOps** - **Power BI** - **Microsoft Defender for Cloud** - **Dynamics 365**

Azure Relational DB's

- Azure SQL DB.
  - 5GB – 4TB Storage
  - 2 - 80 v.Cores
  - $5 / month
- SQL Server in VM

↳ Single DB – Allocate resources to Specific DB

↳ Elastic DB – Allocate resources to Group of DB's (pool) → Multiple DB's in Server

ADVANTAGES:

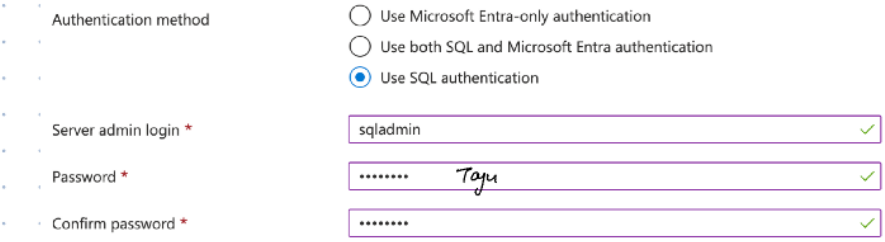
- mostly compatible with SQL Server
- You can scale easily

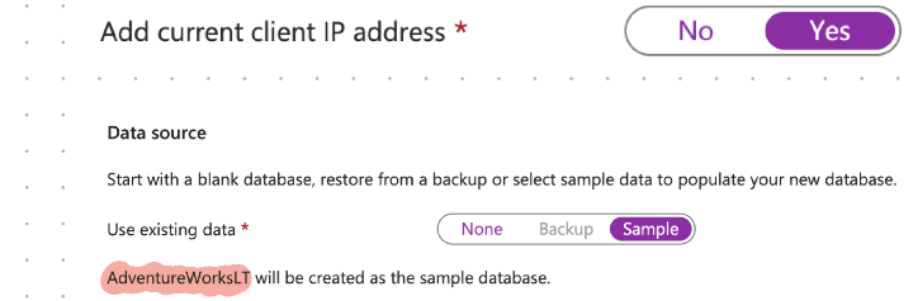
Azure Synapse Analytics: SQL + Spark

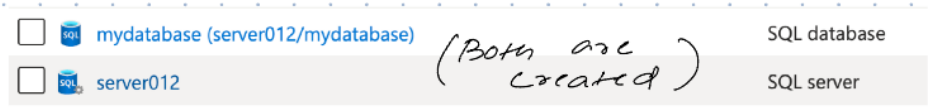
More than a Data warehouse, as it helps with Data warehousing and Big data Analytics as well.

Data warehouse (SQL) + Data Analytics (Spark).

CREATE AZURE SQL DB







(Both are created)

INSTALL AZURE DATA STUDIO

→ New Connection

Server name Use this ↓

server012.database.windows.net

→ Make sure Server is Accessible

- Production is more costly than Dev.
  Development ≈ $5/m
  Production ≈ $500/m
- Vcore → Plan which let's us choose the number of Cores.
- Storage Redundancy
  local → only one location
  Geo Redundant → multiple locations (saved)
- Defender for SQL → $20/month
- Sample Data → Adventure works

ARM - Azure Resource Manager Model

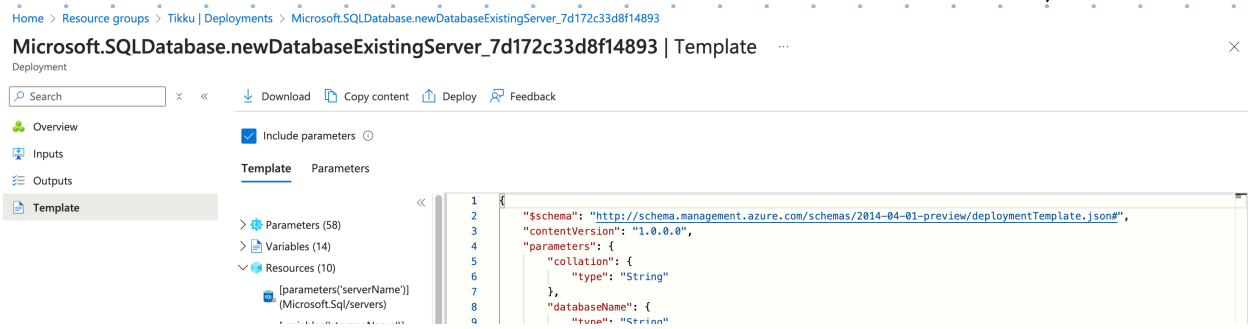
Azure uses Json for data formats in Azure SQL DB,

ARM is used to automate deploying Resources

How does it work?

Resource Group → Automation → ARM Templates

Finally, we can Deploy it again or we can use templates to run it as scripts (change parameters)



Database vs. Server

Aspect	Database 🗄️	Server 🖥️
Definition	Organized collection of data	Hardware/software that hosts and manages resources
Purpose	Stores, retrieves, and manages data	Provides computing power, storage, and services
Dependency	Runs on a server	Can host multiple databases
Example	MySQL, PostgreSQL, SQL Server database	Database server (SQL Server, MySQL Server), Web server (Apache, Nginx)

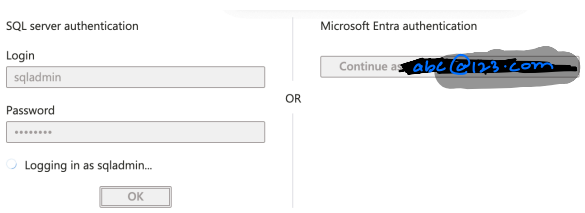
SQL Database Security;

1. Networking; Public / private access

we can customize access based on Firewall Rules

Rule Name → Start Ip address → End Ip address

2. Microsoft Entra; (Active Directory) or Identity & access Security (IAM).



- Secure
- Role Based
- Password less

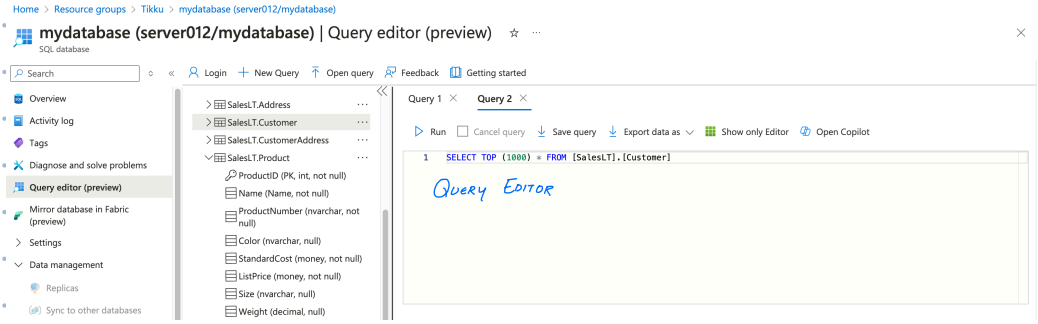
3. Microsoft Defender; is a security solution that protects database from Cyber threats
   - Threat Detection (If someone tries to download all data)

4. Data Encryption; storing data in Unreadable format, even if data is stolen, it makes sure no one can read it, Databases by defaut are encrypted in Azure but we can turn it off.

5. Logging & Auditing; setting up alerts.

RELATIONAL QUERY TOOLS;

QUERY EDITOR



AZURE DATA STUDIO

- Light weight
- Modern
- Cross platform

SSMS OG

- Full SQL Server Admin
- DB management

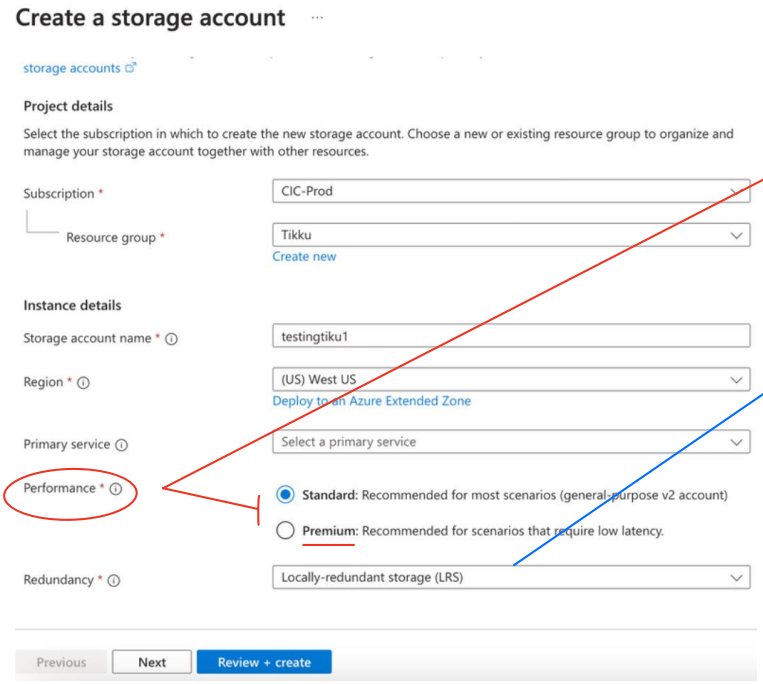
Feature	Azure Data Studio (ADS)	SQL Server Management Studio (SSMS)
Primary Use	Querying, visualization, and cloud integration	Full SQL Server administration & tuning
Best For	Developers, data analysts, Azure users	Database administrators (DBAs)
Platform	Windows, macOS, Linux (Cross Platform)	Windows only
Extensibility	Supports extensions (Jupyter, Git, Python)	Limited to SQL Server features
Performance Tuning	Basic insights	Advanced tuning & Query Store
Cloud Support	Optimized for Azure SQL	Supports Azure but mainly for on-prem SQL Server
UI & Experience	Modern UI, customizable dashboards	Traditional UI, more admin-focused

SQL; SQL SERVER → TSQL
ORACLE / POSTGRE → PLSQL (procedural Language)
↳ The query syntax won't match

NON RELATIONAL STORAGE

- Create Storage Account
- Storage Types
- Non Relational Databases
- Non Relational Data types
- Cosmos DB
  - Create Cosmos DB Account
  - Features of Cosmos DB

Create Storage Account



Faster performance for blobs (SSD)

Redundancy

- Local (LRS) → 3 copies (1 Datacenter)
- Zone (ZRS) → 3 copies (3 Datacenters)
- Geo (GRS) → 6 copies

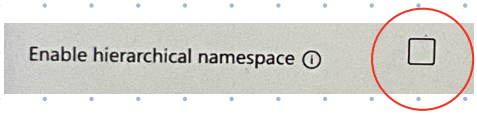
3 in Primary Region
3 in Remote Region

Access Tiers;

HOT - Frequently Used Data

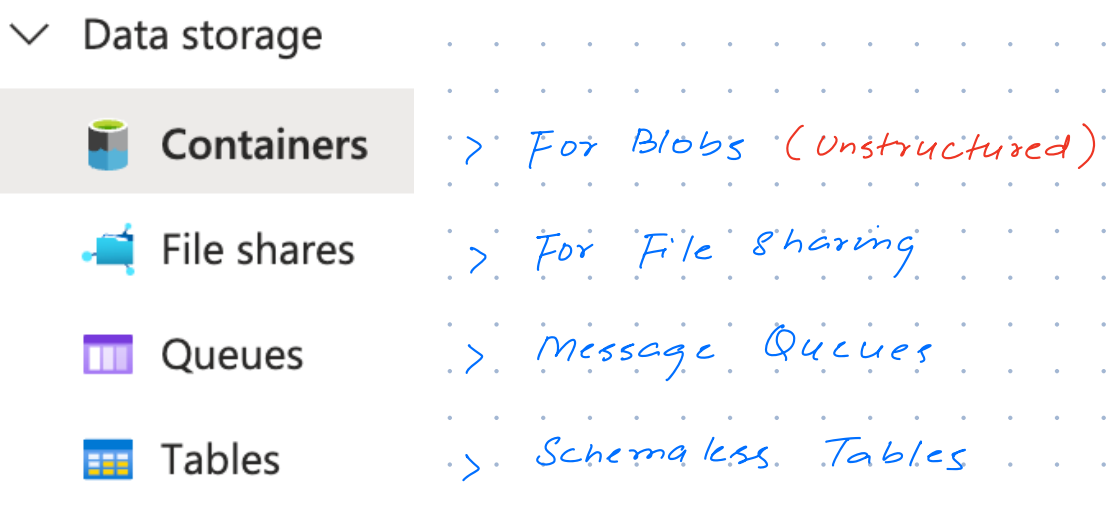
COLD - Rarely Used (Backups) cheap

Gen 2 Storage;



Big Data (Data lake)

Storage Types;



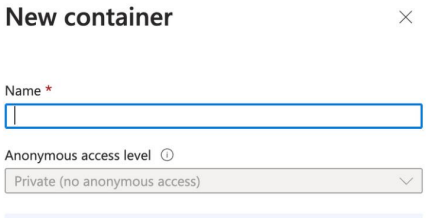
> For Blobs (Unstructured)

> For File sharing

> Message Queues

> Schema less Tables

CREATE CONTAINER BLOB STORAGE (Unstructured Data)



{ we can change access level Not preffered. } Folder type Storage (Container inside Container)

upload Any Type Data inside the container

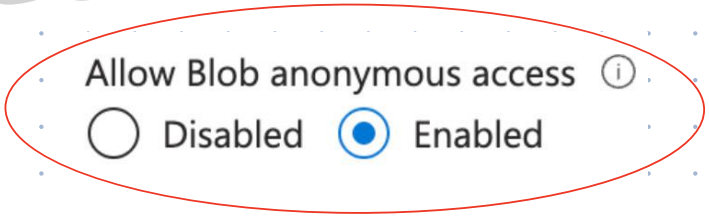
CHANGE ACCESS TIER; HOT, COOL etc.

VIEW EDIT;

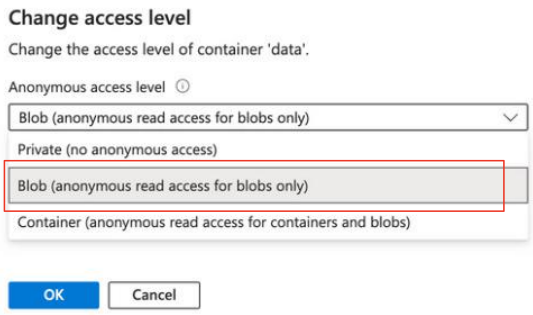
More 2.1MB Blob (File) can be viewed

ALLOW ACCESS TO CONTAINER FILE

Storage Account → Configuration



Container → File → Change Access Level



ACCESS IN PYTHON;

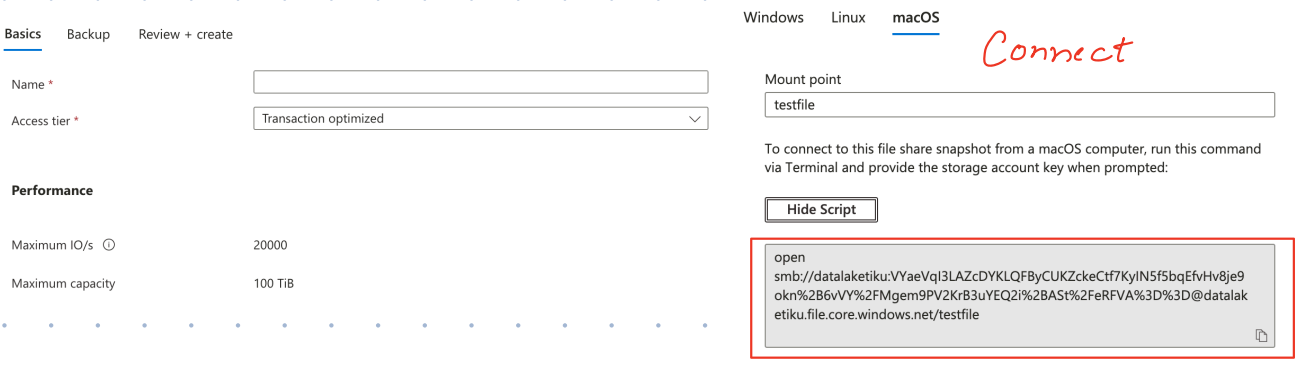
```python
import pandas as pd

url = "https://testingtiku1.blob.core.windows.net/data/ActivityLog-01.csv"
df = pd.read_csv(url)

print(df)
```

FILE SHARES is a cloud based file storage service that allows you to store, share & access files just like a network drive. It works like a shared folder that multiple users or application can access from anywhere.

Max Limit → 100 Tb

Connect → Windows, Mac, Linux

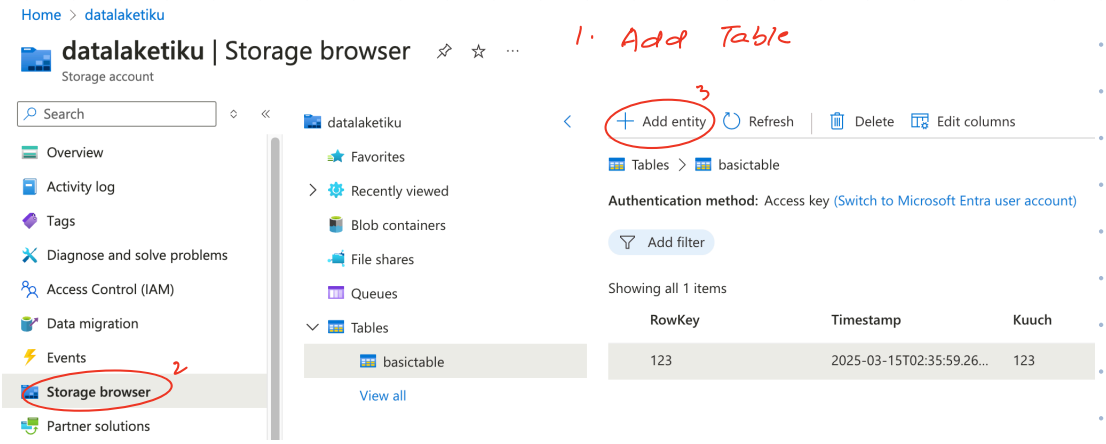
→ Copy Script and Use in Cmd Line



AZURE TABLE STORAGE; Semi Structured Data is not like SQL Database, It is counted as Non Relational Storage. NO Schema.

Add Table → Storage Browser → Add Entity

- NO Relational Element
- Dynamic Schema



NON RELATIONAL DATABASE ; does not used fixed schema or relational tables. (NO SQL) (has a structure)

- Highly Scalable - Big Data (Xml, json)
- Dynamic Schema
- Documents, Key value, Column family, Graph

NON RELATIONAL DATA Types

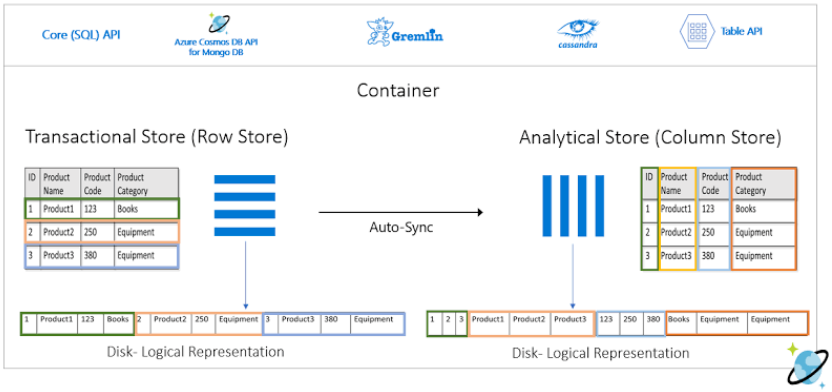
Data Model	Description	Examples
Document Store	Stores data in documents (usually JSON, BSON, or XML)	MongoDB, CouchDB, Azure Cosmos DB (Document API)
Key-Value Store	Uses a simple key-value pair for data retrieval	Redis, Azure Table Storage, Amazon DynamoDB
Column-Family Store	Organizes data into columns rather than rows, ideal for wide datasets	Apache Cassandra, HBase, Azure Cosmos DB (Table API)
Graph Database	Manages data as nodes and edges (relationships)	Neo4j, Azure Cosmos DB (Gremlin API), ArangoDB

DOCUMENT

Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

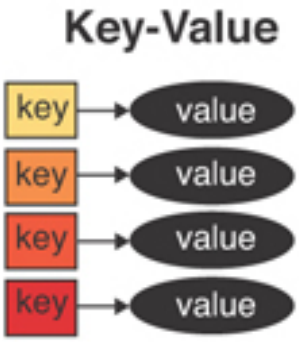
C. DB Core SQL

COLUMN FAMILY



Cosmos DB Cassandra API

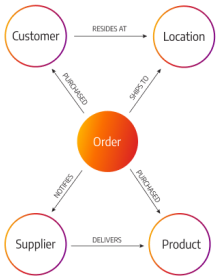
KEY VALUE



Cosmos DB Table API

GRAPH

FB GRAPH DB



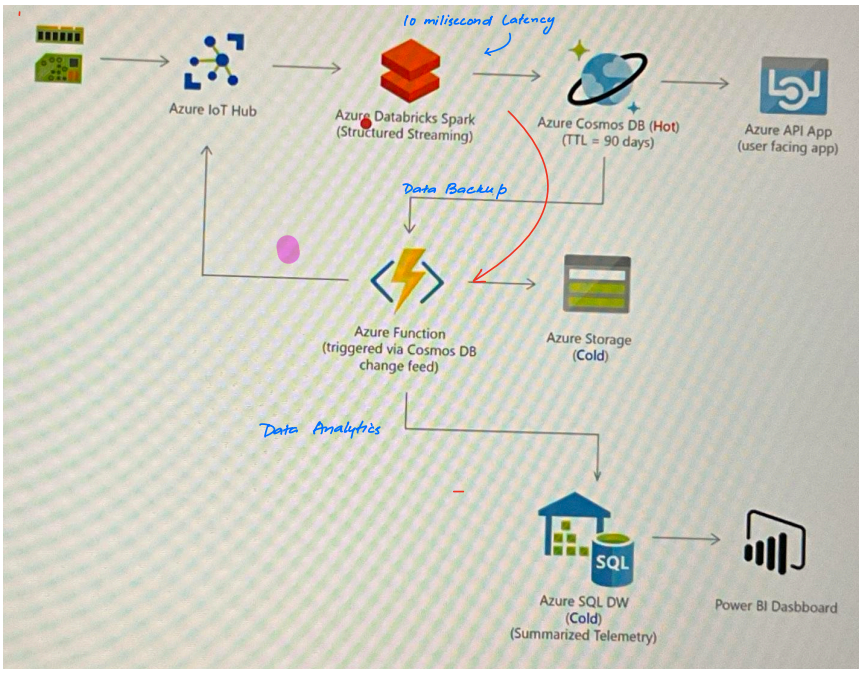
Graph API

OTHERS

- Time Series
- Csv Json in Blob
- Azure Search

Choose Non Relational DB

IoT Device



Non Relational DBs in Azure

Type	Azure Example
Document	Azure Cosmos DB (Core SQL API / MongoDB API)
Key-Value	Azure Table Storage / Azure Cache for Redis
Graph	Azure Cosmos DB (Gremlin API)
Wide Column	Azure Cosmos DB (Cassandra API)

Important Concept

✅ Summary: Why We Don't Classify Images/Videos as a NoSQL Model?

Storage Type	Used for	Example Services
NoSQL Databases	Queryable, structured, or semi-structured data	MongoDB, Cosmos DB, Cassandra, Neo4j
Object Storage (Not NoSQL)	Large-scale binary data (images, videos, PDFs)	Azure Blob Storage, AWS S3, Google Cloud Storage

🚀 **Conclusion: Non-Relational Databases** focus on structured NoSQL models, while images/videos are best stored in **Object Storage** with metadata in a NoSQL or relational database.

Azure Cosmos DB is a no SQL database fully managed by microsoft designed for large data (multimodel)

Low Latency = < 10ms

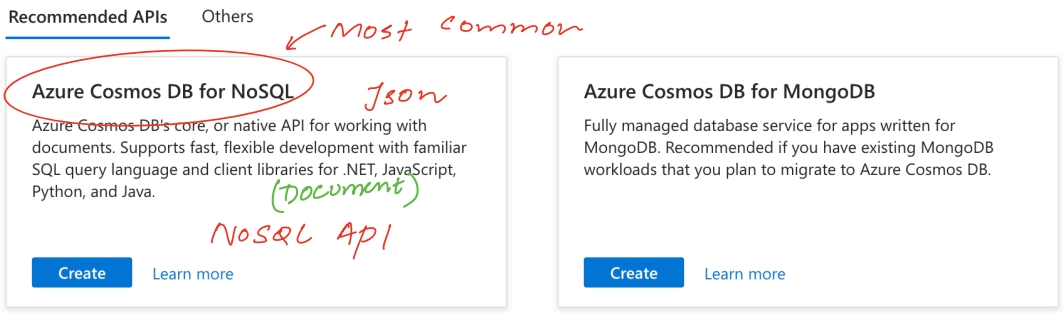
high availability

API

Create Cosmos DB; Flexible database to handle Large amounts of Data.

primary DB → Relational (Azure SQL)

→ Non Relational (Cosmos DB)



Multiple Model Postge Api → Only Relational

Cosmos DB API	Also Known As	Best For	Example Platform
Core (SQL) API	NoSQL API	[redacted] JSON Docs	Web apps, SaaS platforms
MongoDB API	MongoDB-compatible	MongoDB-compatible apps	E-commerce, CMS
Cassandra API	Cassandra-compatible	Wide-column workloads	IoT, Time-series apps
Gremlin API	Graph API	Graph-based queries	Social networks, Recommendation engines
Table API	Azure Table-compatible	Key-value storage	Legacy Azure Table apps

Can I Save Images in Cosmos DB ?

Data	Storage
Image/Video file	Azure Blob Storage
Image/Video metadata	Azure Cosmos DB

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * CIC-Prod

Resource Group * Tikku

Create new

Instance Details

Account Name * firstcosmos

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones ⓘ ○ Enable ◉ Disable

Location * ⓘ (US) East US 2 EUAP

Available locations are determined by your subscription's access and availability zone support (if that i

Click here for more details on how to create a region access request

ⓘ Unsure of your traffic patterns? Explore Azure Cosmos DB's serverless offering if you're just starting out and skip capacity planning by only paying for t

Automatic Scaling

Capacity mode ⓘ

◉ Provisioned throughput
Request units per second (RU/s) are pre-configured and billed hourly, offering guaranteed throug

Pay as you go → ○ Serverless
Ideal for intermittent or unpredictable traffic. Billed only for consumed RU/s, scaling on-demand \

Learn more about capacity mode

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one

Apply Free Tier Discount ◉ Apply ○ Do Not Apply

Limit total account throughput ☑ Limit the total amount of throughput that can be provisioned on this account

ⓘ This limit will prevent unexpected charges related to provisioned throughput. You can update or remove t

Review + create | Previous | Next: Global distribution

Multi region writes; Data can be written from multiple region

Data Encryption *

Microsoft Managed → ◉ Service-managed key

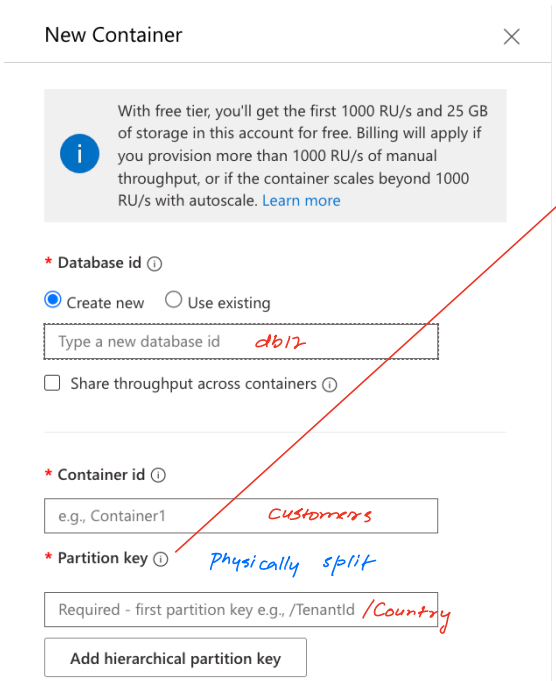
Company Managed → ○ Customer-managed key (CMK)

Query Cosmos DB (NOSQL);

Overview

↓

Create container



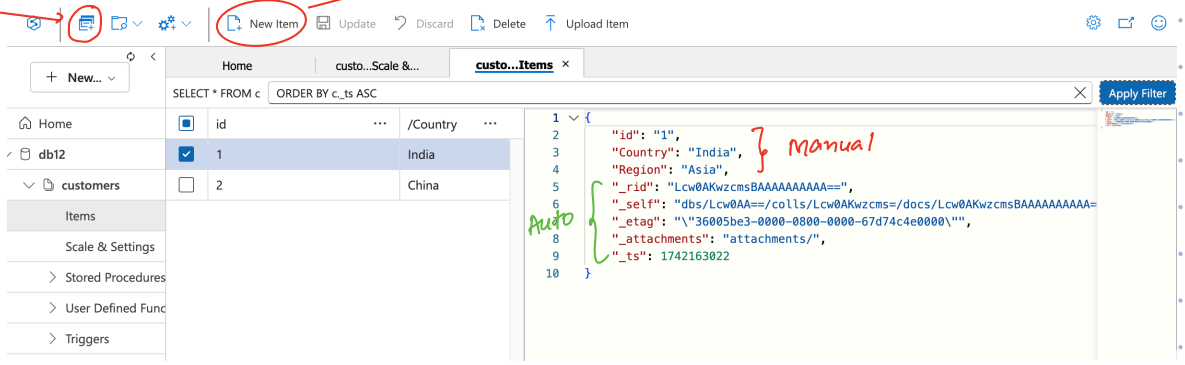
Partition Key

If data gets large, we can use to partition our data based on a column like Country etc.

ADD ITEMS;

SQL Query Editor view

Add New Row



IS SQL inside Cosmos DB case Insensitive?

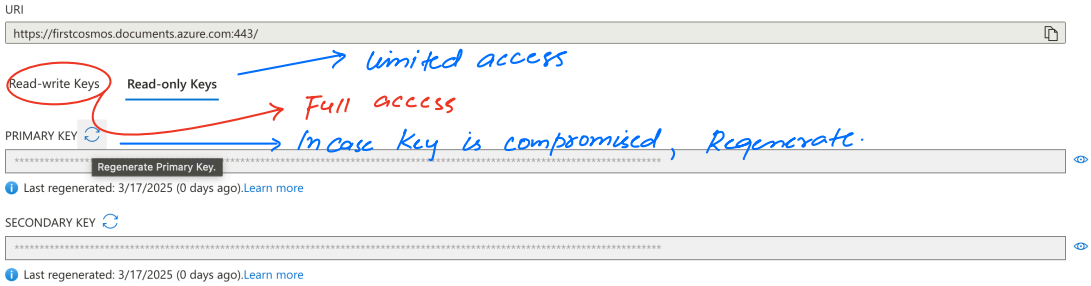
Aspect	Case-sensitive?
Property (Key) names	✅ Yes
Property (String) values	❌ No (by default)
String functions	❌ No (can be made case-sensitive)

ARM TEMPLATES TO MANAGE COSMOS DB

Automation ⟶ Export Template

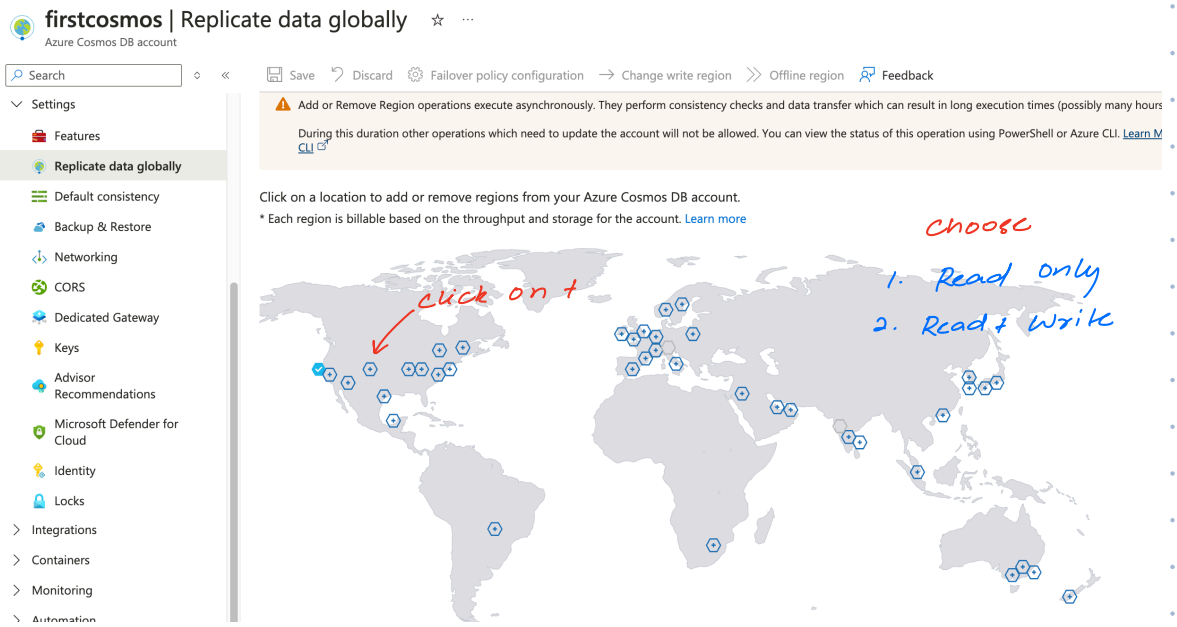
We often use ARM so that we can save these codes in Github because if any change is to be made we can directly make those in code, it is also called (IAC) Infrastructure as code

Cosmos DB Security



Cosmos DB Geo Replication

It is alot more easier to replicate data globally, we can provide read & read+write access, but it 2x's or 3x's the cost



DATA ANALYTICS

- Data workloads
- Data warehouse
- Azure Synapse
- Microsoft Fabric
- ADF
- pipelines - Triggers
- Visualization
  - power BI
  - Types of Reports
  - Components.

Common Data Workloads

Feature	OLTP (Online Transaction Processing) Simple	OLAP (Online Analytical Processing) Historical
Purpose	Real-time transactional processing	Complex queries, reporting, and analytics
Data Type	Structured, operational data (Integrity is there)	Structured & semi-structured, historical data
Operations	Frequent INSERT, UPDATE, DELETE	Aggregations, complex joins, historical analysis
Performance	Optimized for fast read/write	Optimized for complex queries & analytics
Storage	**Azure SQL Database, Azure Cosmos DB** for postgre	**Azure Synapse Analytics, Azure Data Lake Storage (ADLS)** SSAS
Normalization	Highly normalized (reduces redundancy)	Denormalized (improves query performance)
Data Size	Smaller (GBs to TBs)	Larger (TBs to PBs)
Example Use Case	Processing e-commerce orders, banking transactions	Sales trend analysis, financial reporting, business intelligence

General use
Write Heavy
Constanty updates
Az SQL DB / Az Cosmos DB

Complex use
Read Heavy
No updates
Azure Synapse SSAS

Data Warehouse

is a large database where data is from different sources is collected, cleaned and organized so that companies can analyze it and make better decisions

Or

Giant library of data where everything is sorted & ready for reporting / analytics

- Stores historical Data
- OLAP (complex queries & Report)
- Combines data from multi systems

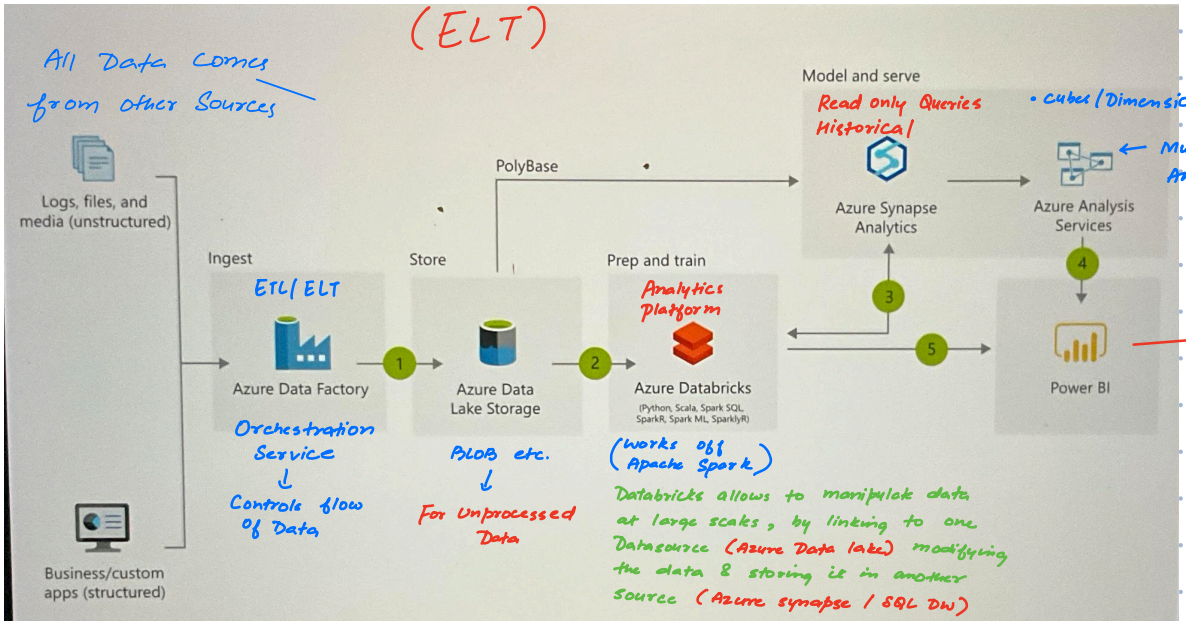
When to use Data Warehouse ?

- When queries are long running & complex
- When Data needs further cleaning ELT or ETL
- (Archiving) when historical data needs to be moved from Day-To-Day Systems
- When we need to integrate data from several sources.

AZURE SYNAPSE = DW + More

Can store both structured + Semi structured unlike Traditional Data warehouses.

COMPONENTS OF MODERN DATAWAREHOUSE ;

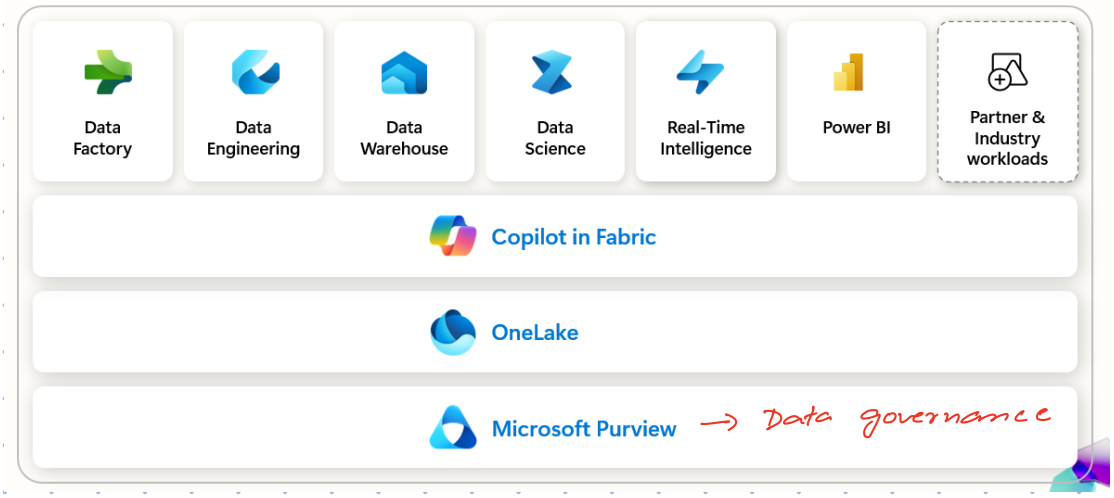


Data from Multiple Sources → ADF

Unstructured Data → Data lake (Blobs) → Databricks

Structured Data → Synapse DW → power BI

MICROSOFT FABRIC ; (SaaS) tool which brings Data factory, One Lake, Synapse power BI all under one roof, No more switching between Synapse, Data Factory etc.



BENIFITS ; • Unified SaaS Model
• End to End Pipelines
• Data Governance & security

↳ Ingestion → Transformation → Analytics

KEY FEATURES ;

One Lake ; A unified data lake that serves as a single storage Layer (onedrive)

Synapse powered Engine combines capabilities from Azure Synapse Analytics for powerful Querying + Data warehousing

power BI & Data Science

Microsoft Purview ; Control center for data security & compliance

How was it before Fabric

Before Fabric (Traditional Azure Setup):

🔄 **You had to use multiple, separate services:** (orchestration)

1. **Azure Data Factory (ADF)** for data ingestion & ETL/ELT
2. **Azure Synapse Analytics** for data warehousing and SQL analytics
3. **Azure Data Lake Storage (ADLS)** for storing large datasets
4. **Azure Databricks** or **Azure Machine Learning** for data science/ML
5. **Power BI** for visualization & reporting
6. **Azure Purview** for data governance

The experience:

- You had to manually **integrate** and **orchestrate** across different services.
- **Switching** between tools like ADF, Synapse, and Power BI was common.
- **Separate billing**, **separate monitoring**, and **more setup overhead**.
- Governance (via Purview) had to be **configured separately**.

After Fabric:

🚀 **Fabric combines them into a single SaaS platform**, so you work in **one unified environment** for:

- Data ingestion
- Data engineering
- Data science/AI
- Real-time analytics
- Business intelligence
- Governance

Before - multiple Tools

After - One Integrated Software

DATA ENGINEER RESPONSIBILITIES

- **provisioning** data storage services;
- **ingesting** streaming and batch data;
- **transforming** data;
- implementing **security** requirements;
- implementing **data retention** policies;
- identifying **performance** bottlenecks; and
- accessing **external data** sources.

AZURE DATA FACTORY

is a data Orchestration and integration service used to build ETL & ELT pipelines for moving & transforming data across hybrid / cloud systems (supporting both Real time & Batch Processing)

OR

It is a pipeline buider that helps you collect, clean and deliver data automatically

WORKING

- Brings Data from external sources
- Data Transformation (ETL / ELT)
- Scheduled Jobs (Automate Transformations)

What is Orchestration in Azure Data Factory?

In ADF, it to refers to controlling how data flows from sources to destination, tasks like

1. Extract Data from sources (SQL, Blob)
2. Transform Data (clean, filter, join)
3. Load into Destination (Azure Synapse, Data Lake)
4. Trigger workflows or schedule based on events
5. Handle failures automatically.

CLOUD VERSION OF SSIS

DATA FACTORY PIPELINES;

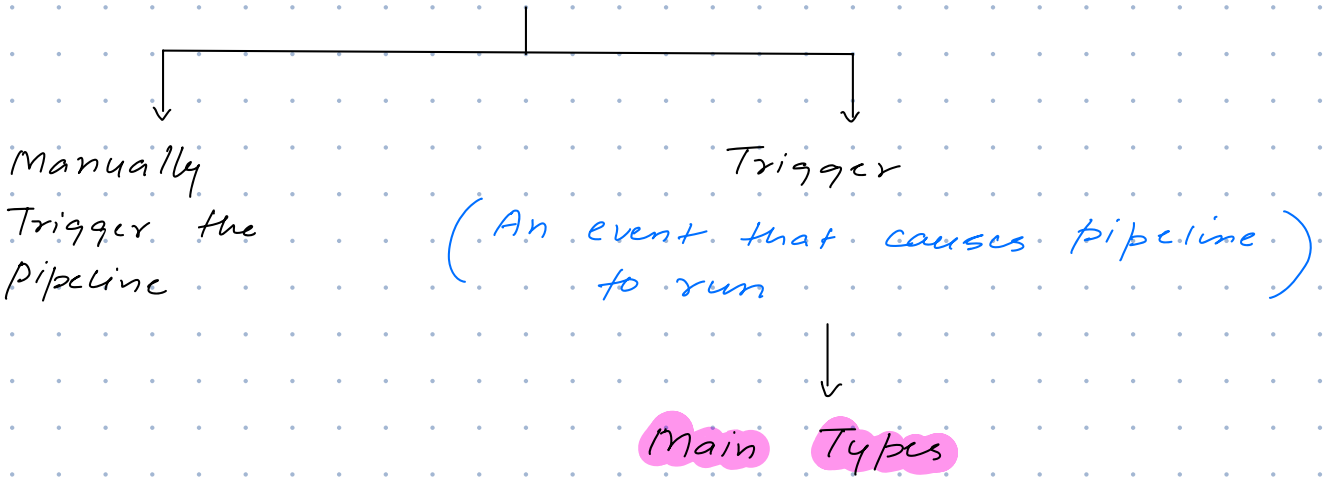
← ADF can have multiple pipelines

is a series of Data related tasks (activities)

- where to get the Data (Extract)
- what to do with Data (Transform)
- where to send Data (Load)

Pipeline Trigger;

Pipeline Run (pipeline Execution)



Trigger Type	Purpose / When to Use
Schedule Trigger	Runs pipelines on a **specific time or recurring schedule** (e.g., every day at 2 AM).

- Every Day @ 8:00pm
- Every hour
- we can set up (Start, End)

Tumblin Window Trigger	Runs pipelines at **regular intervals** (hourly, daily, etc.) and maintains state (for time-based slices of data).

- Can be set to run in the past
- Good when pipeline is time period specific

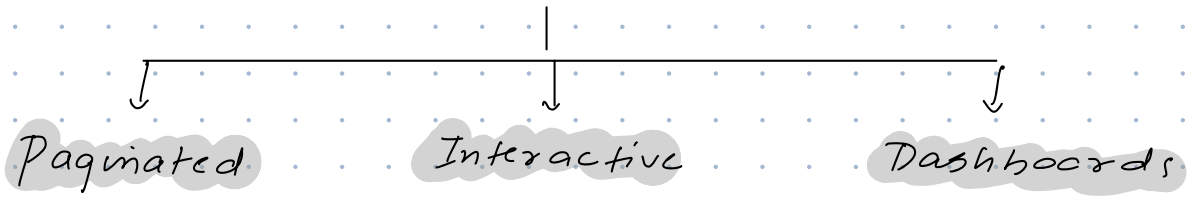
Why use Tumbling Window?

- Ensures **no overlap** between windows (strict time slices).
- Ensures **no missing data** (guarantees each slice is processed).
- Helps in **time-based aggregations** (e.g., hourly sales, hourly logs, etc.).

8:00 - 9:00
9:00 - 10:00
(No Overlap)

VISUALIZATION; Mostly focus on Power BI

Types of Reports



Paginated Reports; Designed to live on page
↳ power BI Report Buider (Download seperately)
→ Finally publish to service

Interactive Report; Designed to be viewed on screen & interacted with
- visuals, slicers etc.
- add remove colum
- Sort etc.

Dashboards; Summarized view of our Reports

POWER BI WORKFLOW;

- Load Data
- Transform (power Query)
- Data Modelling
- Dax (Measure / calculated columns)
- Visualize
- Analyze
- Publish.

Exam Questions

Q1 what are two purchasing models in SQL Database?

DTUs & VCores

Q4 If you set up SQL DB with "no access", which type of user can connect to it?

No one

Q8 compatibility with SQL Server in your own Environment?

SQL Server in VM ⟶ 100%.

SQL managed Instance ≈ 100% (close to 100%.)

Q11 Adding Database (effect on cost)

- Single Database plan - Yes
- Elastic pool - No (we can have upto 100 DBs)

Q14 which of the following Azure Services use the SQL Server Database engine?

- Azure SQL DB
- SQL Server in VM (Not MySQL)
- SQL managed Instance

Q17 Single Table Storage = 5 PB

Q18 Does scaling affect applications using DB?

No, it wont affect @ that time

Q19 If SQL Database is setup @ public endpoint, which type of user can connect to DataBase?

No user, because SQL DB needs to have its firewall configured to allow anyone in.

DW wont be responsible for Transformations

Q11 To implement RLS in DW
Function & Security Policy.

Q4 Sum is not shown in Describe

Q11 date_format not format_date

Q11 innerunique = Deduplicated Matching Rows.

Q11 Telemetry = Data coming from Temp. etc Sensors.

Q11 Use of Shortcuts Cache Enabled.

🚀 **What is Cache for shortcuts enabled in Lakehouse?**

When you **create a shortcut** in a Lakehouse (linking data from another Lakehouse, Warehouse, or storage like ADLS), the data normally **stays at the original source** — it's **not physically copied**.
👉 It reads remotely every time (which can be **slow**).

✅ **If you enable "Cache for shortcuts"**:

- Fabric **copies** (caches) the shortcut data **locally** into your Lakehouse storage.
- So, when you query or process the shortcut table, **it reads from the fast local cache, NOT from the original remote location**.
- **Performance becomes much faster**, especially for **large data**.