

# Azure Fundamentals DP-900

Fundamentals

Associate

Role-based

Expert



Microsoft Certified:  
Azure Data Scientist Associate

Microsoft Certified:  
Azure Data Engineer Associate

Microsoft Certified:  
Azure AI Engineer Associate

Microsoft Certified:  
Azure Solutions Architect Expert

Key  
..... Optional Path  
— Required Path

CORE TOPICS;

Basic Terminology

Core Data Concepts 15 - 20 %

Relational Data 25 - 35 %

Non Relational Data 25 - 30 %

Data Analytics 25 - 30 %



Notes by Tajamul Khan

in @Tajamulkhan

# BASIC TERMINOLOGY

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



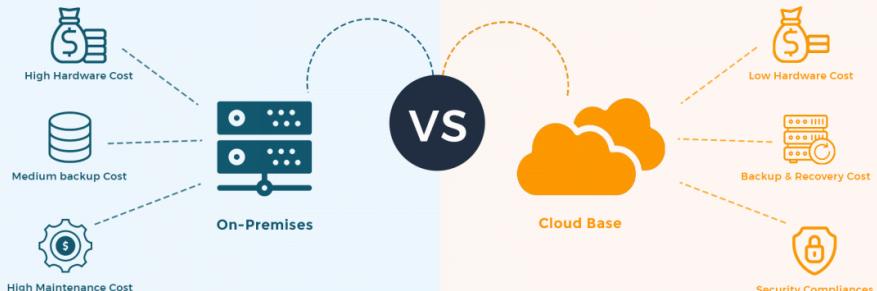
**Notes by Tajamul Khan**

**in @Tajamulkhan**

# CLOUD COMPUTING

means delivering computing services over the internet

## On-Premises VS Cloud Base



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

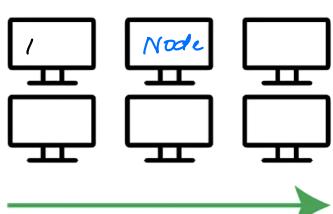
### VERTICAL SCALING

Increase size of instance (RAM, CPU etc.)



### HORIZONTAL SCALING

Add more instances



Vertical - monolithic 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 (NoSQL 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) <small>Semi Structured</small>	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 <small>Read</small>	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 (Relational) 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



Notes by Tajamul Khan  
[in @Tajamulkhan](#)

# 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

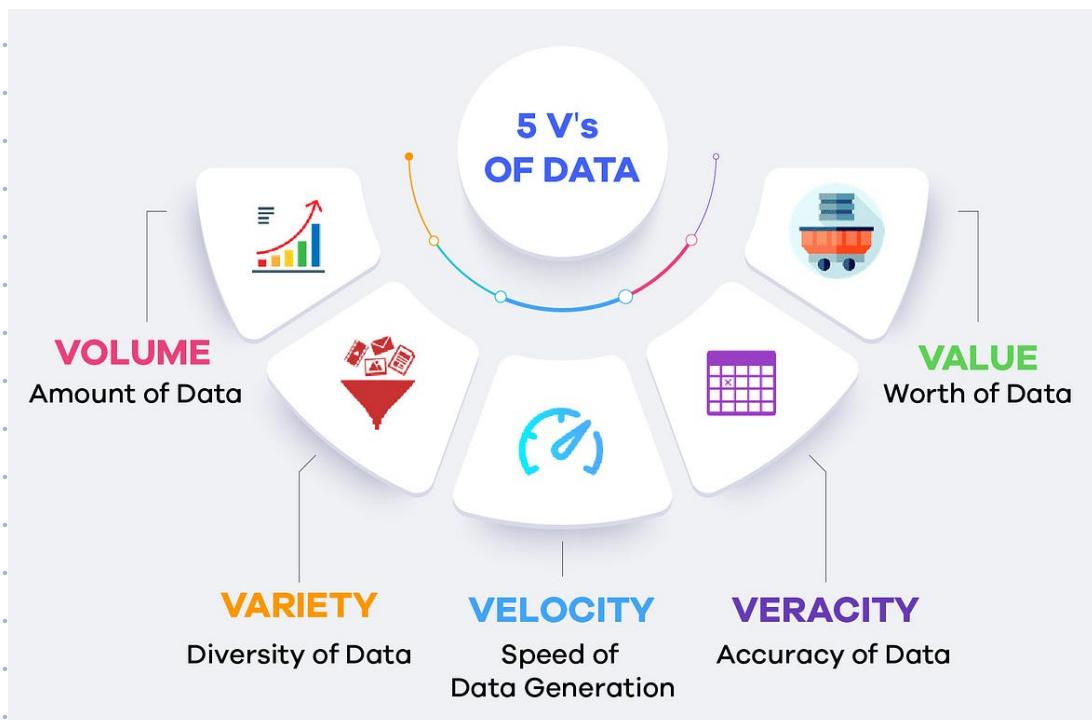
**Cognitive** AI & ML

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).

### 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.

### 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.

### 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.

### 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

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

### ◆ Quick Analogy

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

- **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.

### 💡 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	<u>Fixed schema</u> (tables, rows, columns)	<u>Flexible schema</u> (JSON, XML, CSV)	No predefined structure (videos, images, PDFs)
Storage	Relational databases (SQL, PostgreSQL)	<u>NoSQL databases</u> (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

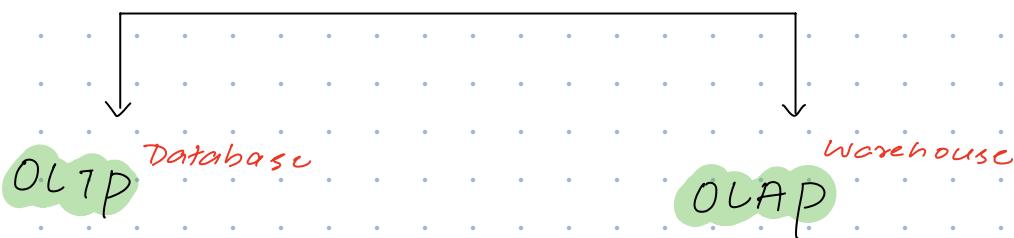
AZURE ULTRA DISK

AZURE BLOB

ADLS, Synapse

# OLTP vs OLAP (Relational / Workloads)

we have Transactional & Analytical



- Day to Day Operations

- Azure SQL DB
- Cosmo DB

- Reports
- Deep Analytics

- Azure Synapse
- AAS

Feature	OLTP (Online Transaction Processing) <i>Simple</i>	OLAP (Online Analytical Processing) <i>Historical</i>
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

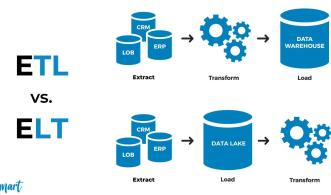
OLTP  
↳ Database

OLAP  
↳ Data warehouses

Q: 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

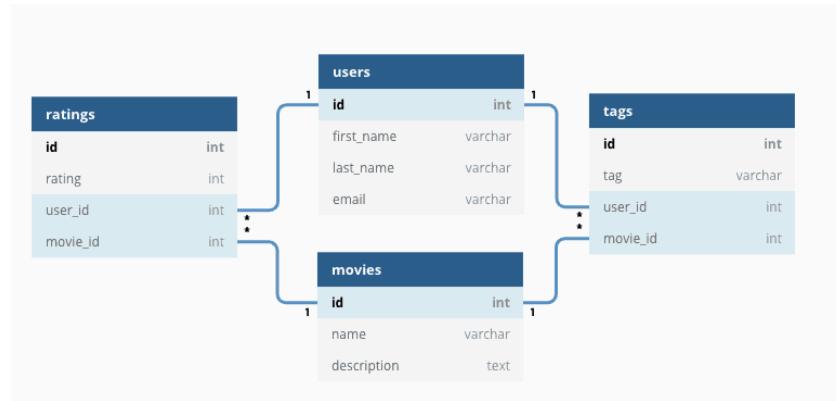
- Features
- Normalization
  - Azure Services
- Azure SQL Database
- Database Security
  - Query Tools.



**Notes by Tajamul Khan**

**in @Tajamulkhann**

**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
4. **Index**; allows for efficient retrieval of data in SQL Queries
  - primary key is default index
  - we can define other indices.

**example -** Put an index on **Last Name**
5. **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.

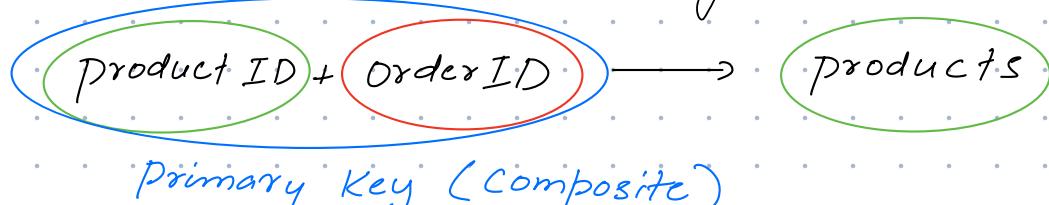
**1NF** - Atomic values

example

123	✓	123	124	✗
-----	---	-----	-----	---

**2NF** - No Partial Dependency

example



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 insensitive

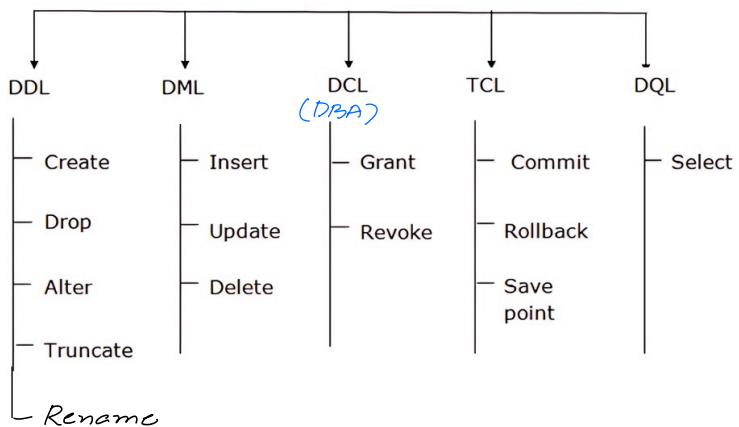
Microsoft - TSQL

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 [SELECTCITY]
@param varchar(20) → Create parameter
AS
SELECT * FROM [SalesLT].[Address]
WHERE CITY = @param
GO

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

## CREATE INDEX

- efficient Data Retrieval

```
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 retrieval quick).

We won't create index for all columns because it also takes computation power

# AZURE SERVICES

Category	Definition	Azure Services Examples
IaaS (Infrastructure as a Service) <i>VMs, Blob</i>	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) <i>To Deploy (Azure)</i>	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) <i>365, Power BI</i>	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 vCores
- \$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 + Data Analytics  
(SQL) (Spark)

# CREATE AZURE SQL DB

Authentication method

Use Microsoft Entra-only authentication  
 Use both SQL and Microsoft Entra authentication  
 Use SQL authentication

Server admin login \*

sqladmin

Password \*

\*\*\*\*\* Taju

Confirm password \*

\*\*\*\*\*

Add current client IP address \*

No Yes

## Data source

Start with a blank database, restore from a backup or select sample data to populate your new database.

Use existing data \*

None Backup Sample

AdventureWorksLT will be created as the sample database.

- mydatabase (server012/mydatabase) SQL database
  - server012 SQL server
- (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  $\approx \$5/m$   
Production  $\approx \$500/m$
- Vcore  $\rightarrow$  Plan which lets us choose the number of cores.
- Storage Redundancy
  - local  $\rightarrow$  only one location
  - Geo Redundant  $\rightarrow$  multiple locations (Saved)
- Defender for SQL  $\rightarrow \$20/\text{month}$
- Sample Data  $\rightarrow$  Adventure Works

# ARM - AZURE RESOURCE MANAGER MODEL

Azure uses JSON for data formats in Azure SQL DB,  
ARM is used to automate deploying Resources

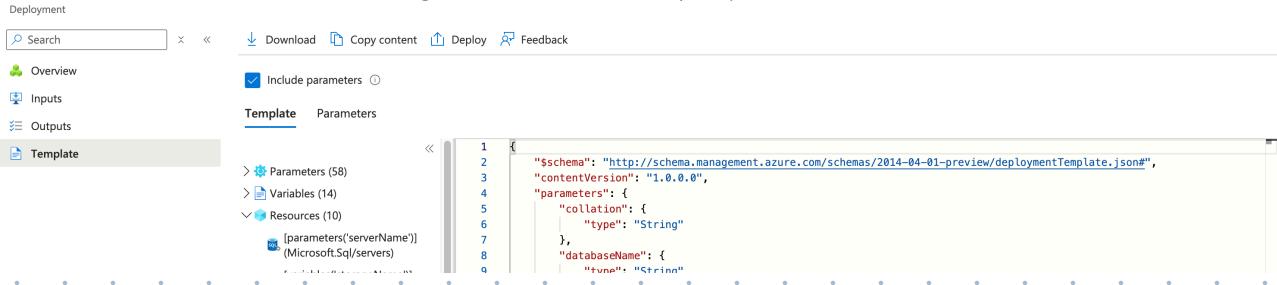
How does it work?

Resource Group → Automation → ARM Templates

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

Home > Resource groups > Tikku | Deployments > Microsoft.SQLDatabase.newDatabaseExistingServer\_7d172c33d8f14893

Microsoft.SQLDatabase.newDatabaseExistingServer\_7d172c33d8f14893 | Template



## 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

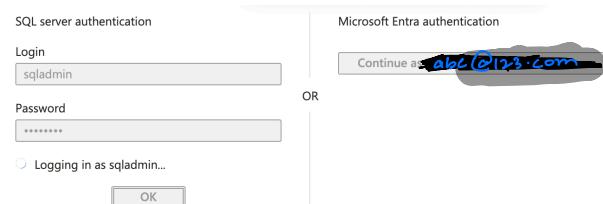
we can customize access based on Firewall Rules

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 default 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 <small>CROSS PLATFORM</small>	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



**Notes by Tajamul Khan**

 **@Tajamulkhann**

# CREATE STORAGE ACCOUNT

## Create a storage account

storage accounts

**Project details**

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription \*

Resource group \*    
[Create new](#)

**Instance details**

Storage account name \*

Region \*    
Deploy to an Azure Extended Zone

Primary service

Performance \*

Standard: Recommended for most scenarios (general purpose v2 account)

Premium: Recommended for scenarios that require low latency.

Redundancy \*

[Previous](#) [Next](#) [Review + create](#)

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;

Enable hierarchical namespace



Big Data (Data Lake)

Storage Types;

## ✓ Data storage



### Containers

> For Blobs (Unstructured)



### File shares

> For File sharing



### Queues

> Message Queues



### Tables

> Schema less Tables

## CREATE CONTAINER

BLOB STORAGE (Unstructured Data)

New container

Name \*

Anonymous access level

{ we can change access level } Folder type storage  
Not preferred. (Container inside Container)

upload Any Type Data inside the container

CHANGE Access TIER; HOT, COOL etc.

VIEW EDIT;

Max 2.1MB Blobs (File) can be viewed

Allow Access To CONTAINER File

Storage Account → Configuration

Allow Blob anonymous access   
 Disabled  Enabled

Containers → File → Change Access Level

Change access level  
Change the access level of container 'data'.

Anonymous access level

Container (anonymous read access for containers and blobs)

Access In 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

Basics    Backup    Review + create

Name \*: testfile

Access tier \*: Transaction optimized

Performance

Maximum IO/s: 20000

Maximum capacity: 100 Tb

Windows    Linux    macOS **Connect**

Mount point: testfile

To connect to this file share snapshot from a macOS computer, run this command via Terminal and provide the storage account key when prompted:

open  
smb://datalaketiku:VYaeVqj3LAZcDYKLQFByCUKZckeCtf7KyIN5f5bqEfVhv8je9okn%2B6vVV%2FMgem9PV2rB3uYEQ2i%2BAST%2FeRFVA%3D%3D@datalaketu.file.core.windows.net/testfile

## 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

Home > datalaketiku

datalaketiku | Storage browser

Storage account

Search

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage browser

Favorites

Recently viewed

Blob containers

File shares

Queues

Tables

basictable

View all

1. Add Table

+ Add entity

Refresh

Delete

Edit columns

Tables > basictable

Authentication method: Access key (Switch to Microsoft Entra user account)

Add filter

Showing all 1 items

RowKey	Timestamp	Kuuch
123	2025-03-15T02:35:59.26...	123

**(Non RELATIONAL DATABASE)**; does not use fixed schema or relational tables. (No SQL) (has a structure) (XML, json)

- Highly Scalable - Big Data
- Dynamic Schema
- Documents, Key Value, Column family, Graph

## Non RELATIONAL DATA Types

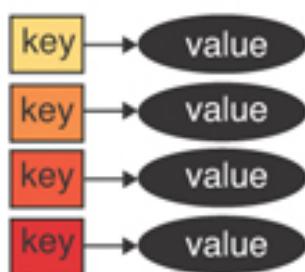
Data Model	Description	Examples
Document Store	Stores data in documents (usually JSON, BSON, or XML)	MongoDB, CouchDB, Azure Cosmos DB (Document API)
Key-Value Store	Uses a simple key-value pair for data retrieval	Redis, Azure Table Storage, Amazon DynamoDB
Column-Family Store	Organizes data into columns 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

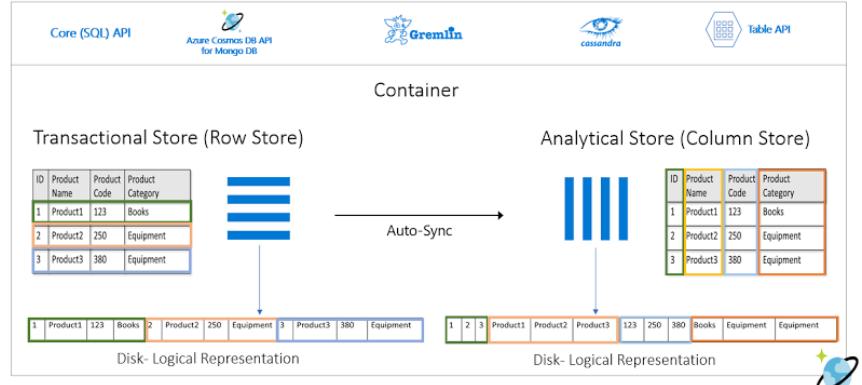
## KEY VALUE



## Key-Value

Cosmos DB Table API

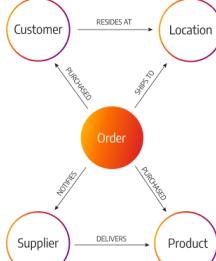
## Column FAMILY



Cosmos DB Cassandra API

## GRAPH

FB GRAPH DB



Graph API

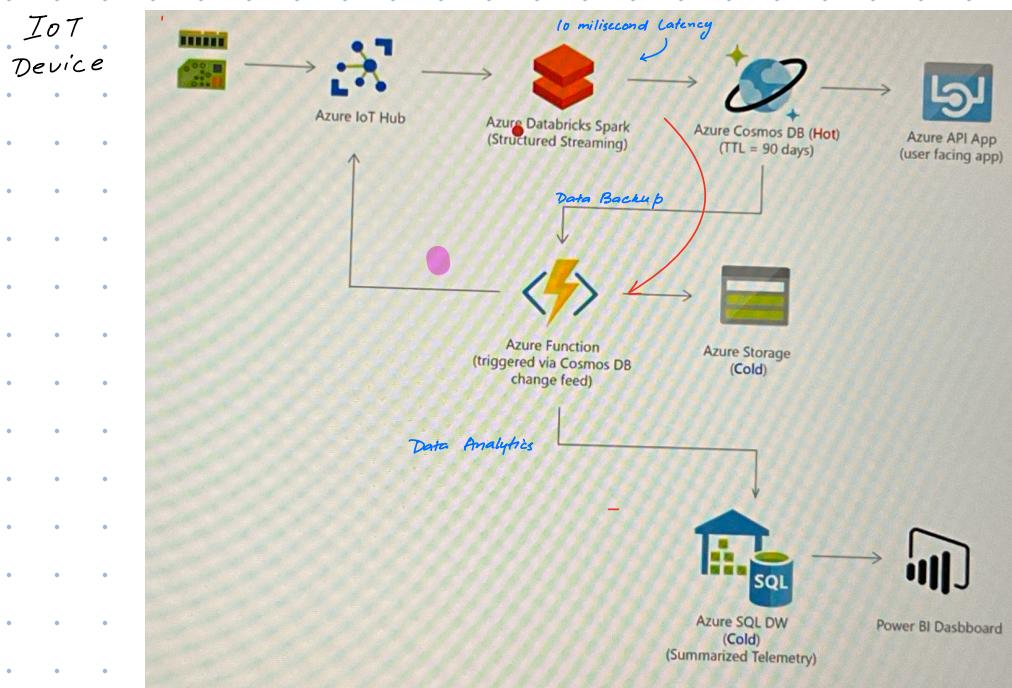
## OTHERS

Time Series

CSV JSON in Blob

Azure Search

# CHOOSE NON RELATIONAL DB



## 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 ( <u>Gremlin API</u> )
Wide Column	Azure Cosmos DB ( <u>Cassandra API</u> )

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 noSQL

database fully managed by Microsoft  
designed for large data (multi model)

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)

Recommended APIs      Others

Most common

**Azure Cosmos DB for NoSQL**      **JSON**  
Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.  
**(document)**  
**NoSQL API**

Create      Learn more

**Azure Cosmos DB for MongoDB**  
Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.

Create      Learn more

MULTIPLE MODEL → Only Relational

Cosmos DB API	Also Known As	Best For	Example Platform
Core (SQL) API	NoSQL API	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.

Enable  Disable

Location \* ⓘ

(US) East US 2 EUAP

Available locations are determined by your subscription's access and availability zone support (if that is available). Click here for more details on how to create a region access request

**Automatic Scaling**

Capacity mode ⓘ

Provisioned throughput

Request units per second (RU/s) are pre-configured and billed hourly, offering guaranteed throughput and low latency.

*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 account.

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

**i** This limit will prevent unexpected charges related to provisioned throughput. You can update or remove the limit at any time.

[Review + create](#)

[Previous](#)

[Next: Global distribution](#)

Multiregion 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

## New Container

With free tier, you'll get the first 1000 RU/s and 25 GB of storage in this account for free. Billing will apply if you provision more than 1000 RU/s of manual throughput, or if the container scales beyond 1000 RU/s with autoscale. [Learn more](#)

\* Database id ⓘ  
 Create new  Use existing  
 Type a new database id **db12**  
 Share throughput across containers ⓘ

\* Container id ⓘ  
 e.g., Container1 **customers**

\* Partition key ⓘ *Physically split*  
 Required - first partition key e.g., /TenantId **/Country**

## 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

	id	/Country
1	1	India
2	2	China

```
SELECT * FROM c ORDER BY c_ts ASC
1 { "id": "1", "Country": "India", "Region": "Asia", "_rid": "Lcw0AKwzcmsBAAAAAAA==", "_self": " dbs/Lcw0AA=/colls/Lcw0AKwzcms=docs/Lcw0AKwzcmsBAAAAAAA=", "_etag": "\"36005be3-0000-0800-0000-67d74c4e0000\"", "_attachments": "attachments/", "_ts": 1742163022 }
```

## 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

URI  
<https://firstcosmos.documents.azure.com:443/>

Read-write Keys      Read-only Keys → limited access

PRIMARY KEY → Full access  
In case key is compromised, Regenerate.

Regenerate Primary Key

Last regenerated: 3/17/2025 (0 days ago). [Learn more](#)

SECONDARY KEY

Last regenerated: 3/17/2025 (0 days ago). [Learn more](#)

# Cosmos DB Geo Replication

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

firstcosmos | Replicate data globally ⚡ ...  
Azure Cosmos DB account

Search Save Discard Failover policy configuration → Change write region >> Offline region Feedback

⚠ Add or Remove Region operations execute asynchronously. They perform consistency checks and data transfer which can result in long execution times (possibly many hours). During this duration other operations which need to update the account will not be allowed. You can view the status of this operation using PowerShell or Azure CLI. [Learn More](#)

Click on a location to add or remove regions from your Azure Cosmos DB account.  
\* Each region is billable based on the throughput and storage for the account. [Learn more](#)

Choose

1. Read Only
2. Read + Write



Features

Replicate data globally

- Default consistency
- Backup & Restore
- Networking
- CORS
- Dedicated Gateway
- Keys
- Advisor Recommendations
- Microsoft Defender for Cloud
- Identity
- Locks
- Integrations
- Containers
- Monitoring
- Automation



# 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) <i>Simple</i>	OLAP (Online Analytical Processing) <i>Historical</i>
Purpose	Real-time transactional processing	Complex queries, reporting, and analytics
Data Type	Structured, operational data <i>(Integrity is there)</i>	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 <i>for Postgre</i>	Azure Synapse Analytics, Azure Data Lake Storage (ADLS) <i>SSAS</i>
Normalization	<u>Highly normalized</u> (reduces redundancy)	<u>Denormalized</u> (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 USC

write heavy

Constantly updates

Az SQL DB/  
Az Cosmos DB

↓  
Complex USC

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

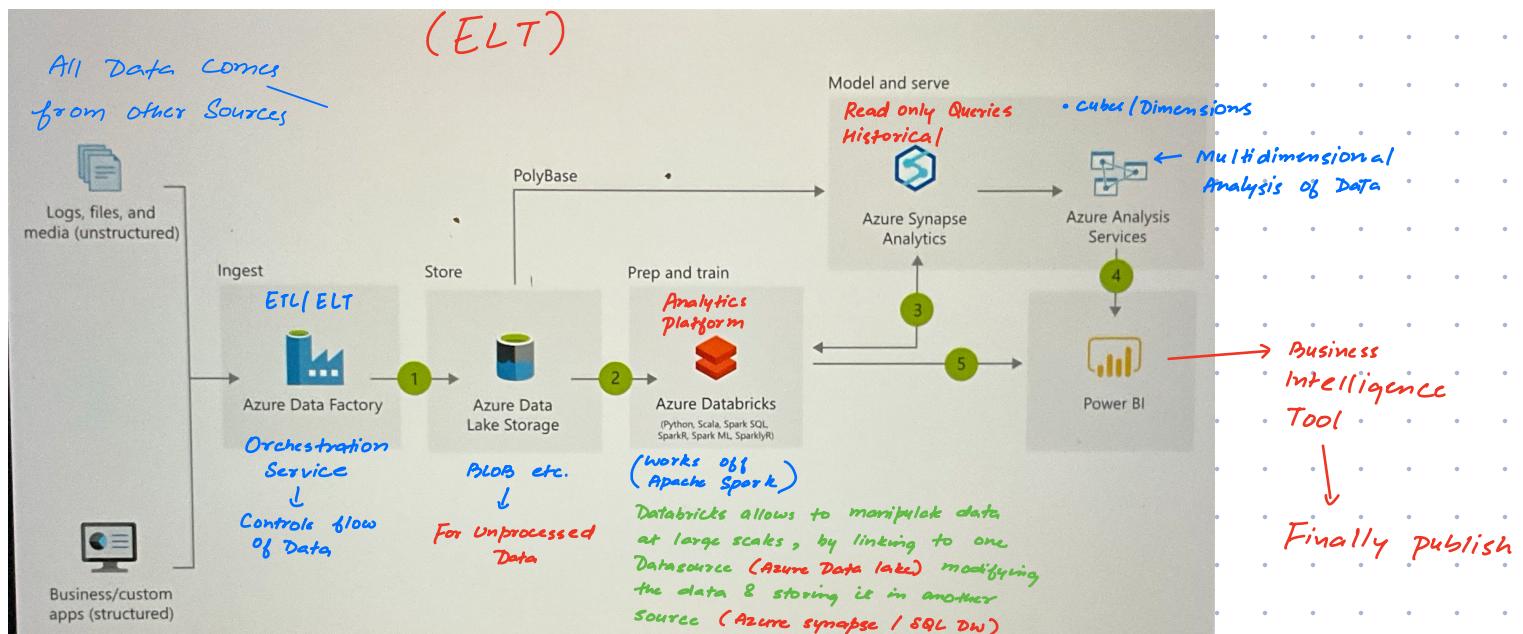
# When to use Data Warehouse?

- 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:

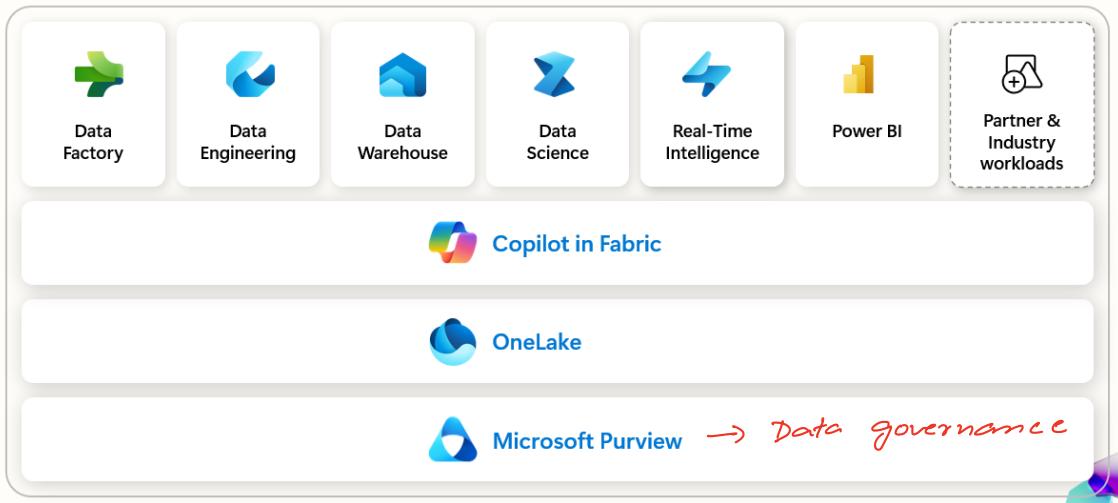


Data from Multiple Sources → ADF

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.



- BENEFITS**;
- 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 builder 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 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

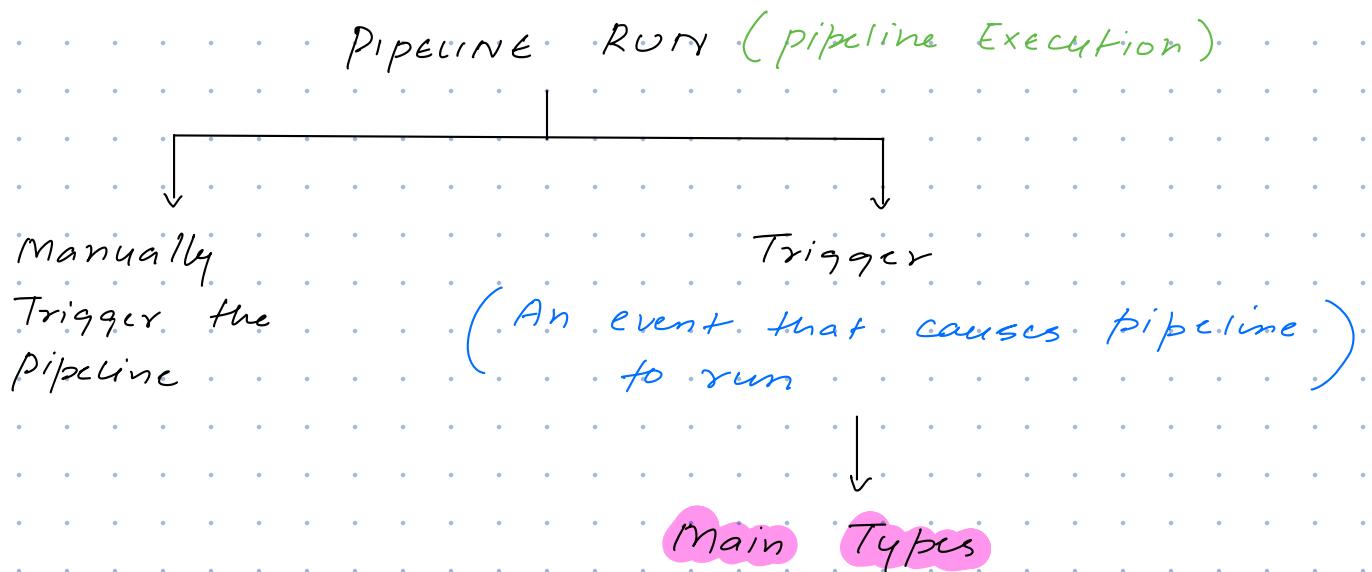
## DATA FACTORY PIPELINES;

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;



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 @ 8:00 PM
- Every hour
- we can set up (start, End)

Tumbling 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?

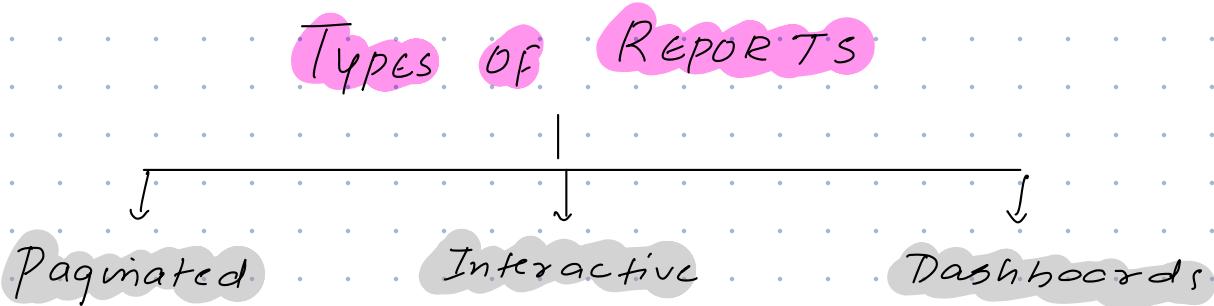
- Ensures no overlap between windows (strict time slices).
- Ensures no missing data (guarantees each slice is processed).
- Helps in time-based aggregations (e.g., hourly sales, hourly logs, etc.).

8:00 - 9:00

9:00 - 10:00

(No Overlap)

# VISUALIZATION; Mostly focus on Power BI



**Paginated Reports;** Designed to live on page

(→ power BI Report Builder (Download separately))

→ Finally publish to service

**Interactive Report;** Designed to be viewed on screen & interacted with

- visuals, slices etc.
- add remove column
- 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

Q<sub>1</sub>, what are two purchasing models in SQL Database?

DTUs & Core

Q<sub>2</sub> If you set up SQL DB with "no access", which type of user can connect to it?

No one

Q<sub>3</sub>, Compatibility with SQL Server in your own Environment?

SQL Server in VM → 100%

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

Q<sub>4</sub>, Adding Database (effect on cost)

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

Q<sub>5</sub>, Which of the following Azure Services use the SQL Server Database engine?

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

Q<sub>6</sub>, Single Table Storage = 5 PB

Q<sub>7</sub>, Does scaling affect applications using DB?

No, it wont affect @ that time

Q<sub>8</sub>, 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

# Azure Fundamentals DP-900



Notes by Tajamul Khan

@Tajamulkhann

Found  
Helpful ?

---

Repost



Follow for more!