# Choosing an Azure data store

Microsoft

## Choose a data store that fits your app, your skills, and your needs

Data can come from many sources—external services, or users, or as log data (to name just a few). These data sets have extremely varied characteristics and processing requirements.

A single data store may not be the best approach. Instead, store different types of data in different data stores, with consideration for the workload or usage pattern. The models here help to understand the choices available on Azure

### REASONS TO USE MICROSOFT AZURE

- **Global reach:** datacenters in 42 countries
- **Resilience:** Azure is an intelligent, self-monitoring, self-healing platform
- **Industry compliances:** large portfolio includes ISO 27001 (information security standard), HIPAA (health information privacy), and SOC 3 (service organization control). Region-specific compliances include the EU-US Privacy Shield and China DJCP
- **Developer focus:** Azure provides "plumbing" to be more efficient. Features such as autoscaling, authentication, and authorization are easy to implement
- **Support for open frameworks:** Write your applications in JavaScript and deploy them to Web Apps. Or write in Django, Java, PHP, or .NET
- **Mobile access:** Monitor Azure resources and performance from mobile devices

### INGEST, STORE AND PROCESS ANY DATA

- Structured, unstructured, streaming or static data
- High throughput, analytical, complex event processing or batch workloads
- Data of any size from kilobytes to petabytes
- Dynamically scales to match your business priorities and manage cost

### IN THE CLOUD, OR ON-PREMISES LEGACY WORKLOADS

- Industry leading products support modernizing legacy applications to Azure
- Best-of-breed managed data services for any type of workload in Azure
- Hybrid deployments across on-premises and cloud

### SIMPLE, FAMILIAR, AND CONSISTENT TOOLING

- Managed and supported open source tools and Microsoft services for all functions
- Designed for seamless interaction between tools and services
- Strong partner ecosystem to integrate and extend the solution

## Points to consider

### Data formats

Common types include images, XML, JSON objects, search indexes, and flat files


- **Structured data**: highly organized information—use relational *(model 1)*, data warehouse *(model 2)*, graph *(model 4)*, search engine *(model 7)*, or object *(model 8)*
- **Semi-structured data**: partially organized information *(Parquet, XML, JSON etc.)*—use key/value *(model 3)*, graph *(model 4)*, columnar *(model 5)*, document *(model 6)*, search engine *(model 7)* or object *(model 8)*
- **Unstructured data**—information with no discernible organization or apparent relationships then use object (model 8)

Many data stores allow the ingestion and transformation of different data formats (e.g. Azure SQL Data Warehouse uses Polybase to ingest and transform semi-structured to structured data type)

### Data characteristics

Data characteristics are the attributes that extend beyond the basic data formats being processed

An essential characteristic is the **workload** type, which affects your final computing requirements:

- For **batch workloads**—large collections of records generated over time—use relational *(model 1)*, data warehouse *(model 2)*, graph *(model 4)*, columnar *(model 5)*, search engine *(model 7)*, or object *(model 8)*
- For **transactional workloads**—smaller records resulting from event triggers—use relational *(model 1)*, key/value *(model 3)*, graph *(model 4)*, document *(model 6)*, or object *(model 8)*

Understanding the type of workload across your data will influence the decision. The initial size and estimated growth of a data set will help determine a solution and how you address some of these challenges could span multiple data stores. Some workloads allow for data to be archived or partitioned across multiple data stores, where others might require that all the data is always highly available and accessible.

### Data relationships

Defined as a common reference between two data sets that joins two or more data sets together

- **Relational data**: important relationships exist between sets of data. Use relational *(model 1)*, data warehouse *(model 2)*, graph *(model 4)*, or columnar *(model 5)*
- **Non-relational data**: no apparent relationships exist between self-contained collections. Use graph *(model 4)*, document *(model 6)*, or object *(model 8)*

Hidden relationships can sometimes be derived for relational and non-relational data sets.

Some data stores can combine disparate data sets that live in various data stores to enrich an existing data set and derive additional insights.t

### Data consistency

Data can exist in more than one place and data consistency refers to the usability of data; data should be constant in time for all instances of an application of that data

- **ACID transactions** ensure that upon completion of a transaction all instances of a data record reflect the change, and any failures to adhere to the properties of that record are not implemented: use relational (*model 1*), data warehouse (*model 2*), or columnar *(model 5)*
- **Eventual consistency** is associated with high write-data workloads—consistency is a trade-off with high performance, and is enforced asynchronously. Use key/value (*model 3*), document (*model 6*), or search engine (*model 7*)

All data stores in Azure ensure consistency of data at some level.

Different models for consistency are characterized by the speed and accuracy with which a change propagates across all copies of a record

### Schema flexibility

A schema refers to the organization and structure of a database or of the data itself

- **Schema on write**—validating a schema when inserting data is usually associated with verifying all rows before a transaction is committed. Changes to a schema must be applied to all tables. Use relational *(model 1)*, data warehouse *(model 2)* graph *(model 4)*, or columnar *(model 5)*
- **Schema on read**—flexible, since any data can be stored. Validation occurs when data is read, and checks ensure that the results conform to the application. Records are ignored where a specified attribute is missing or does not conform. Use key/value *(model 3)*, graph *(model 4)*, or document *(model 6)*

### Concurrency

Concurrency is the ability for multiple processes to access or change shared data at the same time

- High concurrency – associated with highly transactional workloads – use relational *(model 1)*, key/value *(model 3)*, or document (*model 6*)
- Low concurrency – associated with large analytical and batch workloads – use data warehouse (*model 2*), columnar *(model 5)*, or object (*model 8*)

Long running executions can lock data causing a bottleneck affecting concurrency of a system

Some data models handle concurrency at the service level to minimize the impact on performance. Others work at the data level, restricting data access during a transaction

Some systems are optimized for high concurrent-write workloads, while others are designed for high read-concurrency or low-concurrency/large-batch workloads.

### Reliability, replication + availability

Customers and other services may require the data to be available in certain regions within strict time limits

Selecting a product could depend on your ability to mitigate failures in strict time frames. Select a service that satisfies required fault tolerances

- All Azure data services are covered by comprehensive performance and reliability service level agreements. Search for "azure SLAs" or go to https://azure.microsoft.com/support/legal/sla/
- Not all the data services are available in all Azure regions—determine which services support your workload in your required region. Search for "azure regions" or go to https://azure.microsoft.com/regions/

### Performance and scalability

Every workload has a different performance characteristic: some workloads require high throughput writing, and others require large volumes of batch processing for analytics functions

- **Vertical scaling** of a single logical service for high performance throughput—use relational (*model 1*), key/value (*model 3*), columnar *(model 5), or* document (*model 6*)
- **Horizontal scaling** adds compute or storage nodes for linear and predictable performance improvement—use data warehouse (*model 2*), or object (*model 8*)
- **Sharding** is the distribution of a single data set across multiple storage units, to allow for high computational concurrency—use relational (*model 1*)

Some data stores scale automatically, as the workload demands; others offer zero downtime between scaling operations.

Keep applications and data as close together as possible. Access patterns may be associated with specific time zones, so replicating and keeping those databases in sync affects performance

### Data movement and lifecycle

The stages of the data lifecycle are: data capture, data maintenance, data synthesis, data publication, and data purge

**Data capture**: each data store has mechanisms to ingest data, and in some cases rely on other Azure services

**Data maintenance**: refers to the actual storage of the data, this can span multiple services

**Data synthesis**: the creation of new data objects from existing data to create new insights or drive business value

**Data publication**: how data is presented to any downstream application

**Data purge**: the archiving or deletion of data that has outlived its value to downstream applications

Use Azure Data Factory to orchestrate data movement

### Security + network requirements

Azure has the most compliance certifications and offers the most comprehensive layers of protection when moving data to the cloud.


**Protect data**: transparent data encryption, Always Encrypted and Virtual Network Service Endpoints ensure that your data is secure, at rest or in transit

**Monitor activity**: all Azure data services have monitoring features that produce various kinds of logs. The resulting security logs can be shared and integrated into other monitoring tools

**Control access**: integration into Azure Active Directory offers single sign-on experience with familiar auditing and access control functionality

To see all certifications, search for "azure security center" or go to https://azure.microsoft.com/services/security-center/

### Management + Cost

We recommend that you use a managed service, unless you require specific IaaS-hosted features

**Portability**: the complexity around migrating and transforming data from its existing environment to the cloud

**Cost**: the total cost of ownership to deploy a solution and its dependencies

**Cost effectiveness**: A migration is rarely a single service migration—multiple services can be deployed for each phase of your data's lifecycle

### DevOps

Every developer has experience and preferences when making a choice on new or existing products


The Azure platform fully embraces open source technologies which can help simplify a deployment to Azure for data and applications

Your preferences for a specific programming language, operating system or processes plays a role in how you store and interact with data

Customers will often choose technologies they are comfortable with, and with which they have expertise

## Core data store models + data analytics

### Relational databases

- Square, 2-D, rows by columns
- SQL preferred
- ACID transactional consistency
- Schema defined and enforced
- Normalized data

**AZURE PRODUCTS**

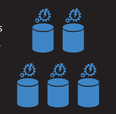
- Azure SQL Database
- Azure Database for MySQL
- Azure Database for PostgreSQL

- *Human capital management*
- *CRM planning*
- *Enterprise resource planning*
- *Inventory management*
- *Reporting database*
- *Accounting*

**MODEL 1**

### Data warehouses

- Separated storage and compute for highly parallelized workloads across multiple servers
- Distributed compute and storage for massive parallelized processing
- Optimized for: large scale data warehousing, data aggregation, batch workloads
- Denormalized data



**AZURE PRODUCTS**

- Azure SQL Data Warehouse
- Hive LLAP in Azure HDInsight

- *Enterprise data warehousing for analytics*
- *Reports*
- *Semantic data models and dashboards*
- *Data mart consolidation*

**MODEL 2**

### Key/value stores

- Data associated with unique key
- Simple query, insert, delete operations
- Schema interpreted by the app—values are blobs
- Optimized for applications
- Highly scalable—can easily distribute across nodes
- Ideal for volatile, semi-structured data

**AZURE PRODUCTS**

- Azure Redis Cache
- Azure Table Storage
- Azure Cosmos DB (Table API)

- *Data caching*
- *Session management*
- *Dictionaries*
- *Product recommendation and ad serving*
- *User preference and profile management*

**MODEL 3**

### Graph databases

- Stores data as nodes (entities) and edges (relationships)
- Edges or relationships are first class citizens in the database
- Allows you to practically model and query data
- Intuitive and extremely efficient while generating insights from highly connected, complex data

**AZURE PRODUCTS**

- Azure Cosmos DB (Graph API)
- Azure SQL Database

- *Product catalog*
- *User accounts*
- *Bill of materials*
- *Personalization*
- *Content management*
- *Operations data*
- *Inventory management*
- *Transaction history*
- *File/blob indexing*

**MODEL 4**

### Data analytics engines *

Analyzing big data spread across multiple data stores requires an analytics engine that can work with all Azure data stores. The engine:

- Scales compute and storage independently
- Features high computational throughput across persisted storage
- Can adopt characteristics of other data stores (relational databases, document stores etc.)
- Enables Machine Learning and AI at scale

**AZURE PRODUCTS**

- Azure Databricks
- Azure HDInsight
- SQL Data Warehouse
- Azure Data Lake Analytics
- Azure Analysis Services
- Azure Batch

- *Modern data warehouse*
- *Big data batch processing (ETL/ELT)*
- *Advanced analytics on big data*
- *Real-time analytics*
- *Personalization/recommendations*
- *Social media analytics*
- *Fraud detection*
- *Telemetry analysis*

** An analytics engine can exist apart from any store, and can work with multiple store models.*

**MODEL 5**

### Columnar databases

- Similar to relational databases—data is stored in columns instead of rows
- Schema defined and enforced
- Usually highly compressed—uses less disk space
- Optimized for columnar operations—MIN, MAX, SUM, COUNT, AVG

**AZURE PRODUCTS**

- Azure SQL Database
- Azure SQL Data Warehouse
- Azure Database for MariaDB

- *Historical data analysis*
- *Data warehousing*
- *Business intelligence*

**MODEL 6**

### Document databases

- Similar to key/value stores—named fields and data known as documents
- Can contain compound elements (lists or child collections)
- Fields are visible to storage system which enables filtering by values
- Documents are free form structures
- Update a record without rewriting document

**AZURE PRODUCTS**

- Azure Cosmos DB (MongoDB API)

- *Product catalog*
- *User accounts*
- *Bill of materials*
- *Personalization*
- *Content management*
- *Operations data*
- *Inventory management*
- *Transaction history*
- *File/blob indexing*

**MODEL 7**

### Search engines

- Search across multiple data sources and services
- Indexes and stores massive volumes
- Near real time access to indexes
- Can run across existing databases
- Indexing performed by a pull or push (triggered by external application)

**AZURE PRODUCTS**

- Azure Search
- Azure Data Lake Analytics

- *Product catalogs*
- *Site search*
- *Logging*
- *Analytics*
- *Shopping sites*

**MODEL 8**

### Object stores

- Optimized for large binary objects
- Objects are composed of the stored data, metadata, and unique ID
- Designed to support large files and provide large amounts of total storage
- Can replicate a given blob across multiple server nodes for fast parallel reads

**AZURE PRODUCTS**

- Azure Storage (Archive, Blob, Disk, File, Queue, Table)
- Azure Data Lake Store

- *Images, videos, office documents, PDFs*
- *CSS, scripts, CSV*
- *Static HTML, JSON*
- *Log and audit files*
- *Database backups*

**MODEL 8**

For more information: https://docs.microsoft.com/en-us/azure/architecture/guide/



Version 1.0 Created by the Azure Poster Team Email: azureposter@microsoft.com