

DP-900

# Azure Data Fundamentals



# Getting Started

In **28**  
Minutes



Azure Database MySQL



SQL Database



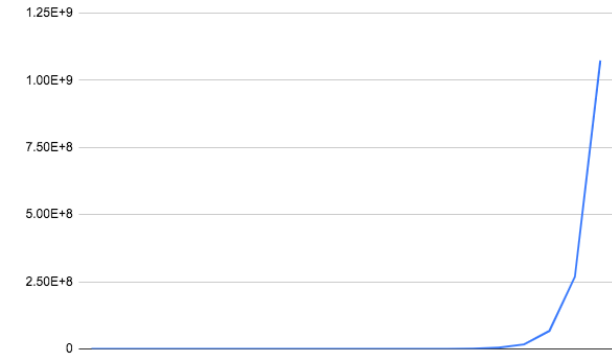
Cosmos DB



Synapse Analytics

- Azure has *200+ services*. Exam expects you to understand *40+ services*.
- Exam *tests* your **decision making abilities**:
  - Which data format will you use in which situation?
  - Which Azure data store will you use in which situation?
- This course is **designed** to help you *make these choices*
- **Our Goal** : Help you get certified and start your cloud journey with Azure

- Data is the "oil of the 21st Century Digital Economy"
- Amount of data generated **increasing exponentially**
  - Mobile devices, IOT devices, application metrics etc
  - **Variety of**
    - **Data formats:** Structured, Semi Structured and Unstructured
    - **Data store options:** Relational databases, NoSQL databases, Analytical databases, Object/Block/File storage ...
- Store data efficiently and gain intelligence
- **Goal of the course:** Help you choose specific data format and Azure data store for your use case
  - We will start with 10,000 feet overview of cloud:
    - Regions, Zones and IaaS/PaaS/SaaS
  - After that, play with different data formats and data storage options in Azure



# IaaS vs PaaS vs SaaS

# Azure Virtual Machines

In **28**  
Minutes

- In corporate data centers, data stores are deployed on physical servers
- Where do you deploy data stores in the cloud?
  - Rent virtual servers
  - **Virtual Machines** - Virtual servers in Azure
  - **Azure Virtual Machines** - Provision & Manage Virtual Machines



VM

# Problem with using VMs for Databases

- You need to take care of:
  - OS installation & upgrades
  - Database installation & upgrades
  - Availability (create a standby database)
  - Durability (take regular backups)
  - Scaling compute & storage



VM

# Managed Services

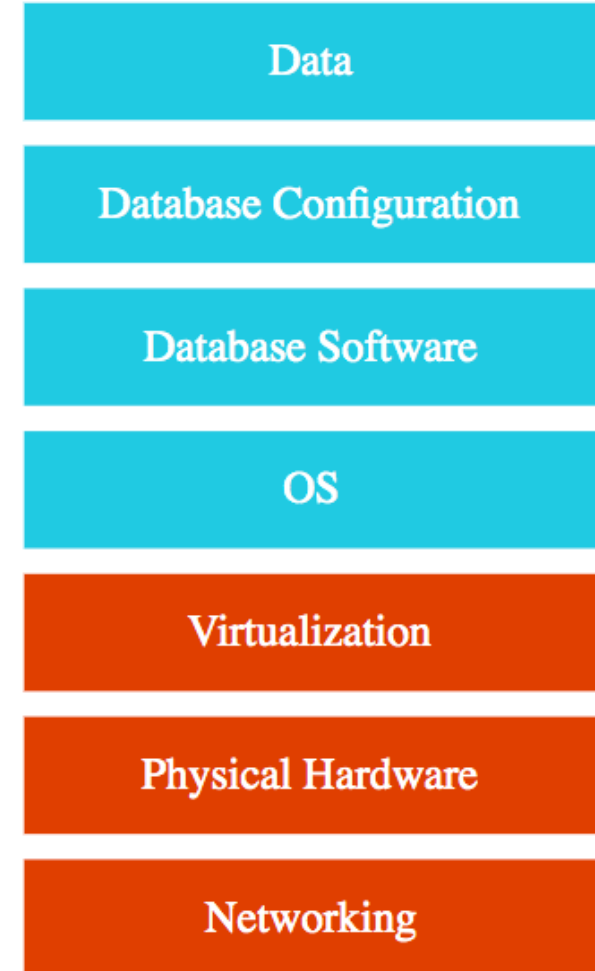
- Do you want to continue **running databases in the cloud, the same way you run them in your data center?**
- **OR are there OTHER approaches?**
- **Let's understand some terminology** used with cloud services:
  - **IaaS** (Infrastructure as a Service)
  - **PaaS** (Platform as a Service)
  - **SaaS** (Software as a Service)
- Let's get on a quick **journey** to understand these!





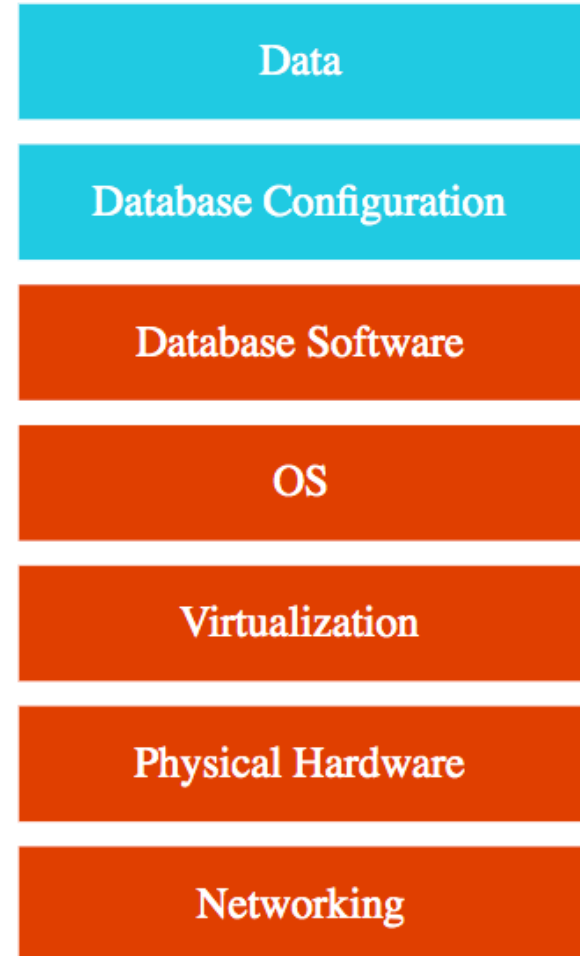
# IaaS (Infrastructure as a Service)

- Use **only** infrastructure from cloud provider
- **Example:** Running SQL Server on a VM
- Cloud Provider is responsible for:
  - Virtualization, Hardware and Networking
- You are responsible for:
  - OS upgrades and patches
  - Database software and upgrades
  - Database Configuration (Tables, Indexes, Views etc)
  - Data
  - Scaling of compute & storage, Availability and Durability



# PaaS (Platform as a Service)

- Use a platform provided by cloud
- **Cloud provider** is responsible for:
  - Virtualization, Hardware and Networking
  - OS upgrades and patches
  - Database software and upgrades
  - Scaling, Availability, Durability etc..
- **You** are responsible for:
  - Database Configuration (Tables, Views, Indexes, ...)
  - Data
- Examples: Azure SQL Database, Azure Cosmos DB and a lot more ...
- **You** will NOT have access to OS and Database software (most of the times!)



# SaaS (Software as a Service)



- **Centrally hosted software** (mostly on the cloud)
  - Offered on a subscription basis (pay-as-you-go)
  - Examples:
    - Email, calendaring & office tools (such as Outlook 365, Microsoft Office 365, Gmail, Google Docs)
    - Customer relationship management (CRM), enterprise resource planning (ERP) and document management tools
- **Cloud provider** is responsible for:
  - OS (incl. upgrades and patches)
  - Application Runtime
  - Auto scaling, Availability & Load balancing etc..
  - Application code and/or
  - Application Configuration (How much memory? How many instances? ..)
- **Customer** is responsible for:
  - Configuring the software!

# Azure Cloud Service Categories - Scenarios

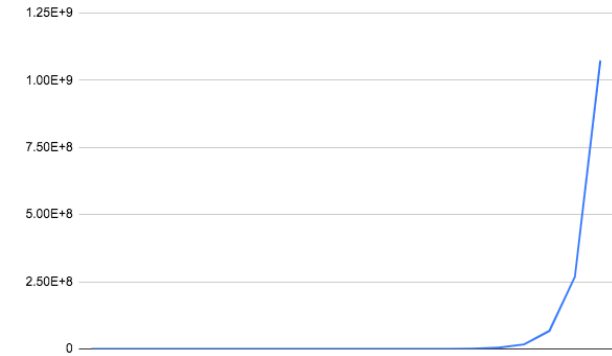
| Scenario  | Solution |
|---|----------|
| IaaS or PaaS or SaaS: Deploy a Database in Virtual Machines                     | IaaS     |
| IaaS or PaaS or SaaS: Using Gmail   | SaaS     |
| IaaS or PaaS or SaaS: Using Azure SQL Database to create a database             | PaaS     |
| True or False: Customer is responsible for OS updates when using PaaS           | False    |
| True or False: Customer is responsible for Availability when using PaaS         | False    |
| True or False: In PaaS, customer has access to VM instances                     | False    |
| True or False: In PaaS, customer can customize OS and install custom software   | False    |
| True or False: In PaaS, customer can configure hardware needs (memory, cpu etc) | True     |

# Data Formats & Data Stores

## 10,000 Feet Overview

# Data Formats & Data Stores

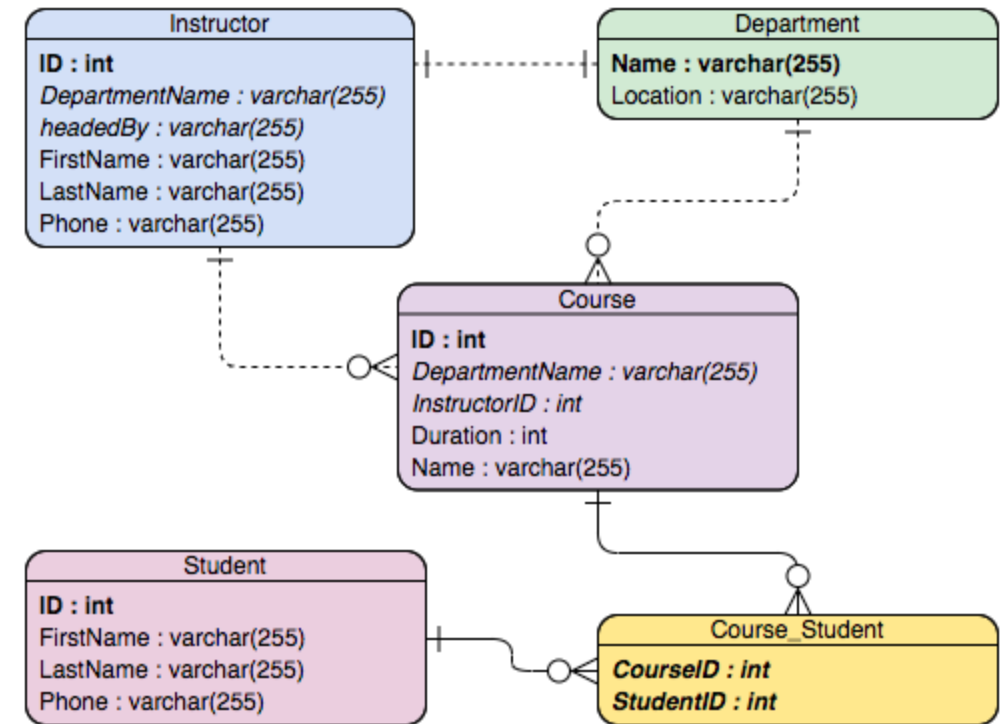
- Data is the "oil of the 21st Century Digital Economy"
- Amount of data generated increasing exponentially
- **Data formats:**
  - Structured: Tables, Rows and Columns (Relational)
  - Semi Structured: Key-Value, Document (JSON), Graph, etc
  - Unstructured: Video, Audio, Image, Text files, Binary files ...
- **Data stores:**
  - Relational databases
  - NoSQL databases
  - Analytical databases
  - Object/Block/File storage



# Structured Data - Relational Databases

- Data stored in Tables - Rows & Columns
- Predefined schema - Tables, Relationships and Constraints
- Define indexes - Query efficiently on all columns
- Used for
  - OLTP (Online Transaction Processing) use cases and
  - OLAP (Online Analytics Processing) use cases

| ID | DepartmentName   | Name                        | Duration | InstructorID |
|----|------------------|-----------------------------|----------|--------------|
| 1  | Computer Science | Algorithms                  | 8        | 2            |
| 2  | Computer Science | Data Structures             | 6        | 4            |
| 3  | Computer Science | Operating Systems           | 5        | 4            |
| 4  | Computer Science | Database Management Systems | 20       | 2            |



# Relational Database - OLTP (Online Transaction Processing)

- Applications where large number of users make large number (millions) of transactions
  - Transaction - small, discrete, unit of work
    - Example: Transfer money from your account to your friend's account
  - Heavy writes and moderate reads
  - Quick processing expected
- **Use cases:** Most traditional applications - banking, e-commerce, ..
- **Popular databases:** MySQL, Oracle, SQL Server etc
- **Some Azure Managed Services:**
  - **Azure SQL Database:** Managed Microsoft SQL Server
  - **Azure Database for MySQL:** Managed MySQL
  - **Azure Database for PostgreSQL:** Managed PostgreSQL



SQL Database



Azure Database  
PostgreSQL



# Relational Database - OLAP (Online Analytics Processing) <sup>In 28 Minutes</sup>

- Applications allowing users to **analyze petabytes of data**
  - **Examples:** Reporting applications, Data warehouses, Business intelligence applications, Analytics systems
  - Data is consolidated from multiple (typically transactional) databases
  - **Sample application** : Decide insurance premiums analyzing data from last hundred years
- Azure Managed Service: **Azure Synapse Analytics**
  - **Petabyte-scale** distributed data ware house
  - Unified experience for developing end-to-end analytics solutions
    - Data integration + Data warehouse + Data analytics
  - Run complex queries across petabytes of data
  - Earlier called Azure SQL Data Warehouse

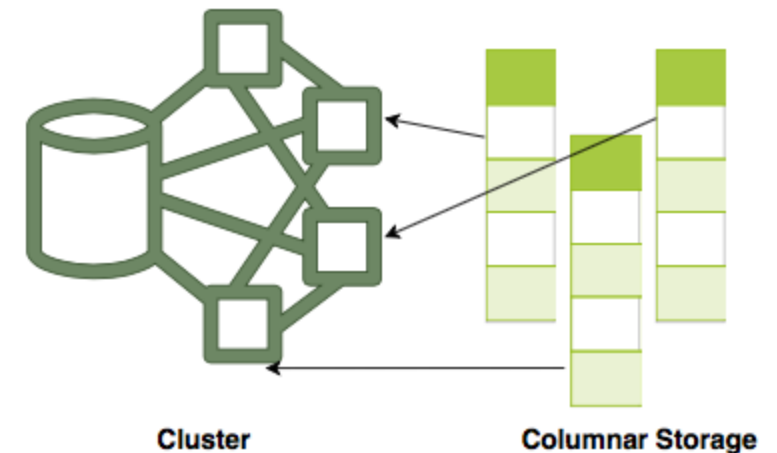
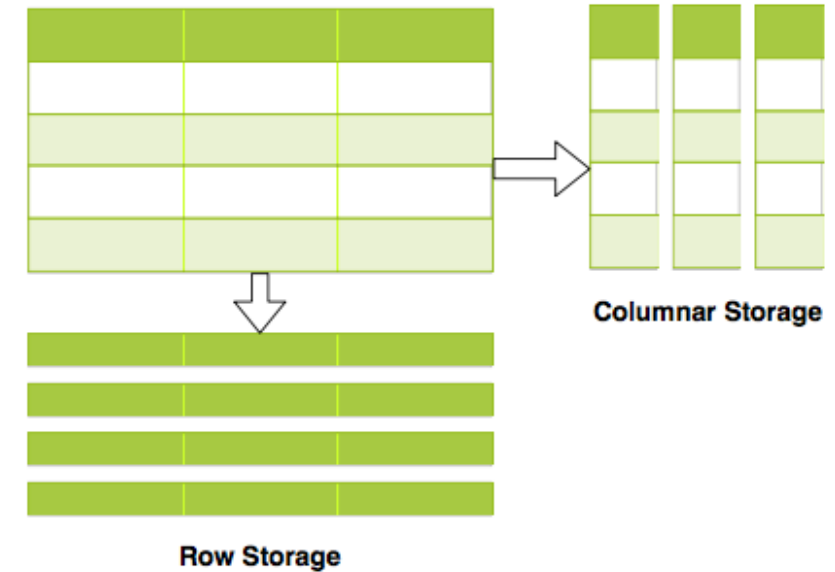


Synapse Analytics

# Relational Databases - OLAP vs OLTP

In 28  
Minutes

- OLAP and OLTP use similar data structures
- BUT very different approach in how data is stored
- **OLTP databases** use row storage
  - Each table row is stored together
  - Efficient for processing small transactions
- **OLAP databases** use columnar storage
  - Each table column is stored together
  - **High compression** - store petabytes of data efficiently
  - **Distribute data** - one table in multiple cluster nodes
  - **Execute single query across multiple nodes** - Complex queries can be executed efficiently





# Semi Structured Data

- Data has **some structure BUT not very strict**
- Semi Structured Data is stored in NoSQL databases
  - NoSQL = not only SQL
  - Flexible schema
    - Structure data **the way your application needs it**
    - Let the structure evolve with time
  - Horizontally scale to petabytes of data with millions of TPS
- **Managed Service: Azure Cosmos DB**
- **Types of Semi Structured Data:**
  - Document
  - Key Value
  - Graph
  - Column Family

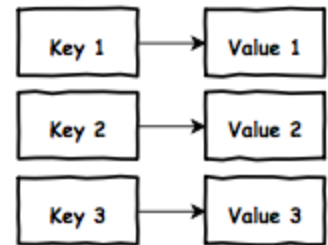
```
{
  "customerId": "999999999",
  "firstName": "Ranga",
  "lastName": "Ranga",
  "address": {
    "number": "505",
    "street": "Main Street",
    "city": "Hyderabad",
    "state": "Telangana"
  },
  "socialProfiles": [
    {
      "name": "twitter",
      "username": "@in28minutes"
    },
    {
      "name": "linkedin",
      "username": "rangaraokaranam"
    }
  ]
}
```

# Semi Structured Data - 1 - Document

- Data stored as **collection of documents**
  - Typically **JSON** (Javascript Object Notation)
    - Be careful with formatting (name/value pairs, commas etc)
    - address - Child Object - { }
    - socialProfiles - Array - [ ]
  - Documents are retrieved by unique id (called the key)
    - Typically, you can define additional indexes
  - Documents don't need to have the same structure
    - No strict schema defined on database
    - Apps should handle variations (application defined schema)
  - Typically, information in one document would be stored in multiple tables, if you were using a relational database
- **Use cases:** Product Catalog, Profile, Shopping Cart etc
- **Managed Service:** Azure Cosmos DB SQL API & MongoDB API

```
{
  "customerId": "999999999",
  "firstName": "Ranga",
  "lastName": "Ranga",
  "address": {
    "number": "505",
    "street": "Main Street",
    "city": "Hyderabad",
    "state": "Telangana"
  },
  "socialProfiles": [
    {
      "name": "twitter",
      "username": "@in28minutes"
    },
    {
      "name": "linkedin",
      "username": "rangaraokaranam"
    }
  ]
}
```

# Semi Structured Data - 2 - Key-Value



Key Value Database

| userId ⓘ ▲ | session  |
|------------|--|
| user1      | { "name": "Jane", "previousAction" : "someAction1" } |
| user2      | { "name": "Doe", "previousAction" : "someAction2" }  |
| user3      | { "name": "Doe", "previousAction" : "someAction3" }  |

- Similar to a **HashMap**
  - **Key** - Unique identifier to retrieve a specific value
  - **Value** - Number or a String or a complex object, like a JSON file
  - Supports simple lookups - query by keys
    - NOT optimized for query by values
    - Typically, no other indexes allowed
- **Use cases:** Session Store, Caching Data
- **Managed Services:** Azure Cosmos DB Table API, Azure Table Storage

# Semi Structured Data - 3 - Graph



- Social media applications have data with complex relationships
- How do you store such data?
  - As a graph in **Graph** Databases
  - Used to store data with **complex relationships**
- Contains nodes and edges (relationships)
- **Use cases:** People and relationships, Organizational charts, Fraud Detection
- **Managed Service:** Azure Cosmos DB Gremlin API

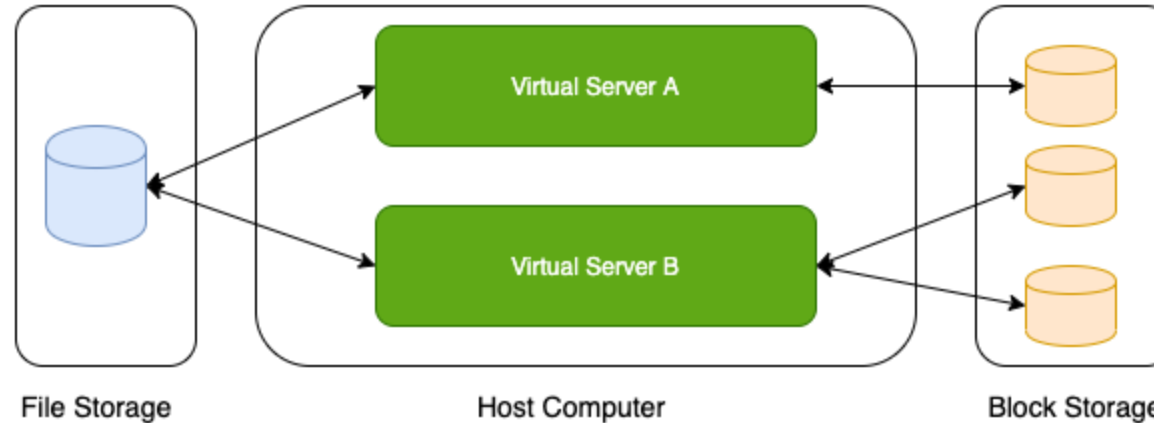
# Semi Structured Data - 4 - Column Family

| Rowid | Column Family 1 |      |      | Column Family 2 |      |      | Column Family 3 |      |      |
|-------|-----------------|------|------|-----------------|------|------|-----------------|------|------|
|       | col1            | col2 | col3 | col1            | col2 | col3 | col1            | col2 | col3 |
| 1     |                 |      |      |                 |      |      |                 |      |      |
| 2     |                 |      |      |                 |      |      |                 |      |      |
| 3     |                 |      |      |                 |      |      |                 |      |      |

- Data organized into **rows and columns**
- Can appear similar to a relational database
- **IMPORTANT FEATURE:** Columns are divided into groups called column-family
  - Rows can be sparse (does NOT need to have value for every column)
- **Use cases:** IOT streams and real time analytics, financial data - transaction histories, stock prices etc
- **Managed Service:** Azure Cosmos DB Cassandra API



# Unstructured Data



- Data which does not have any structure (Audio files, Video files, Binary files)
  - What is the type of storage of your hard disk?
    - **Block Storage** (Azure Managed Service: Azure Disks)
  - You've created a file share to share a set of files with your colleagues in a enterprise. What type of storage are you using?
    - **File Storage** (Azure Managed Service: Azure Files)
  - You want to be able to upload/download objects using a REST API without mounting them onto your VM. What type of storage are you using?
    - **Object Storage** (Azure Managed Service: Azure Blob Storage)

•

# Relational vs Non Relational Data - Quick Overview

In **28**  
Minutes

- **Relational Data (Structured Data)**

- **OLTP:** SQL Server on Azure VMs, Azure SQL Database (or Azure SQL Managed Instance), Azure Database for PostgreSQL, MariaDB, MySQL
- **OLAP:** Azure Synapse Analytics

- **Non Relational Data (Semi Structured/Unstructured Data)**

- **Semi Structured - Document (JSON)**
  - Azure Cosmos DB SQL API and Cosmos DB MongoDB API
- **Semi Structured - Key-Value**
  - Azure Cosmos DB Table API, Azure Table Storage
- **Semi Structured - Column-Family**
  - Azure Cosmos DB Cassandra API
- **Semi Structured - Graph**
  - Azure Cosmos DB Gremlin API
- **Unstructured Data**
  - Block Storage (Azure Disks), File Storage (Azure Files), Object Storage (Azure Blob Storage)



Cosmos DB



SQL Database



Azure Database MySQL



Azure Storage

# Databases - Scenarios

| Scenario   | Solution  |
|--|---|
| A start up with quickly evolving schema for storing documents                | Azure Cosmos DB SQL API and Cosmos DB MongoDB API   |
| Transactional local database processing thousands of transactions per second | Azure SQL Database and other relational databases.. |
| Store complex relationships between transactions to identify fraud           | Azure Cosmos DB Gremlin API                         |
| Database for analytics processing of petabytes of structured data            | Azure Synapse Analytics                             |
| File share between multiple VMs  | Azure Files   |
| Storing profile images uploaded by your users                                | Azure Blob Storage                                  |

# Relational Databases

# Relational Databases

In **28**  
Minutes

- **Structured Data** - Tables, Rows and Columns
- **Structured Query Language (SQL)** for retrieving and managing data
- Recommended when **strong transactional consistency guarantees** are needed
- Database schema is mandatory
- **Azure Managed Services:**
  - Azure SQL Database
  - Azure SQL Managed Instance
  - Azure Database for PostgreSQL
  - Azure Database for MySQL
  - Azure Database for MariaDB



SQL Database



Azure Database MySQL



Azure Database  
PostgreSQL

# Azure SQL Database

In **28**  
Minutes

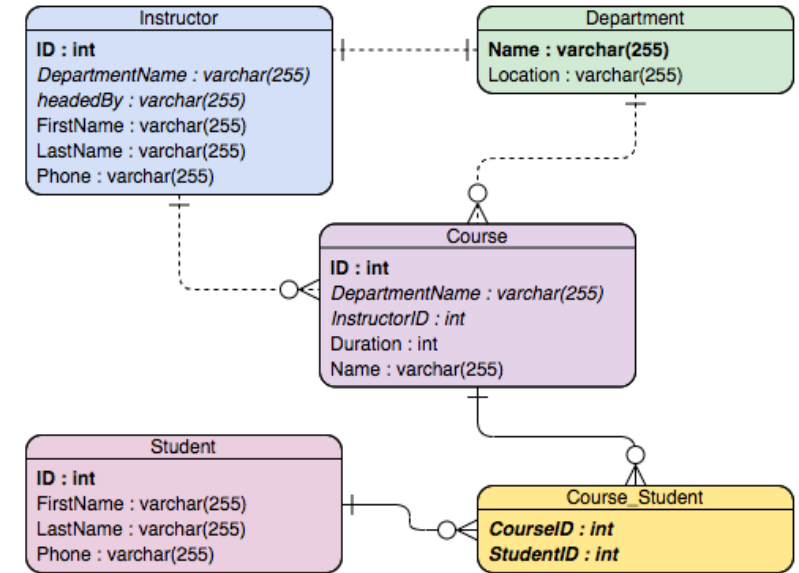
- **Fully Managed Service** for Microsoft SQL Server
- 99.99% availability
- **Built-in** high availability, automatic updates and backups
- Flexible and responsive serverless compute
- Hyperscale (up to 100 TB) storage
- **Transparent data encryption(TDE)** - Data is automatically encrypted at rest
- **Authentication:** SQL Server authentication or Active Directory (and MFA)



SQL Database

# Relational Databases - Tables and Relationships

- Relational Databases are modeled using **Tables and Relationships**
  - A Course has an Instructor
  - A Course belongs to a Department
- **Table:** Table contains columns and rows
  - All rows in a table have same set of columns
  - Relationship between tables is established using Primary Key and Foreign Key
    - **Primary Key:** Uniquely identifies a row in a table
    - **Foreign Key:** Provides a link between data in two tables



# Structured Query Language



- **SQL:** Language used to perform operations on relational databases
  - **Data Definition Language (DDL):** Create and modify structure of database objects
    - Create: Create a database or its constituent objects (Table, View, Index etc)
    - Drop: Delete objects (Table, View, Index) from database
    - Alter: Alter structure of the database
  - **Data Query Language (DQL):** Perform queries on the data
    - Example: `SELECT * from Course, SELECT Count(*) from Course`
  - **Data Manipulation Language (DML):** Insert, update or delete data
    - Example: `insert into Course values (1, 'AZ-900', 1);`
    - Example: `Update Course Set title='AZ-900 Azure Fundamentals' where id=1`
    - Example: `Delete from Course where id=1`
  - **Data Control Language (DCL):** Manage permissions and other controls
    - Example: Grant and revoke user access - `GRANT SELECT ON course TO user1`
  - **Transaction Control Language (TCL):** Control transactions within a database
    - `Commit` - commits a transaction
    - `Rollback` - rolls back a transaction (used in case of an error)



```
CREATE CLUSTERED INDEX INDEX_NAME on TABLE (COLUMN_NAME);
```

```
CREATE NONCLUSTERED INDEX INDEX_NAME on TABLE (COLUMN_NAME);
```

- Allows **efficient data retrieval** from a database
- **Combination** of one or more columns
- **Remember:** An index is automatically created with the primary key
- **Remember:** A table can have more than one index
- **Two Types of Indexes:**
  - **Clustered:** Data in table is stored in the order of the index key values
    - Remember: Only one clustered index per table ( Why? - data rows can only be sorted in one way)
  - **Non-clustered** indexes: Index stored separately with pointers to the data rows

```
create view all_courses_with_students
as
  select course_id, student_id, first_name, last_name, title
  from Course_Student, Student, Course
  where Course_Student.student_id = Student.id and
         Course_Student.course_id=Course.id;
```

- **View:** Virtual table mapped to a query
- Can be used just like a table in SQL queries
- **Use cases:** Add calculated columns, join multiple tables, filter unnecessary columns

# Normalization



Database

- **Goals** in designing relational databases:
  - High Data Integrity
  - Minimum Data Redundancy (or Duplication)
- **How do achieve these goals?**
  - **Database Normalization:** "Process of restructuring a relational database to reduce data redundancy and improve data integrity"
    - **First Normal Form (1NF):** Single(atomic) valued columns
      - Violation Example: A column named address
    - **Second Normal Form (2NF):** Eliminate redundant data
    - **Third Normal Form (3NF):** Move columns not directly dependent on primary key
      - (REMEMBER) There are other normal forms (4NF, 5NF, ...) but 3NF is considered good enough for most relational data
- **Advantages of Normalization**
  - Avoid same data being duplicated in multiple tables
  - Reduce disk space wastage
  - Avoid data inconsistencies

# Normalization example

## Unnormalized - Enrollment Details

|   | student_first_name | student_last_name | title        | instructor_first_name | instructor_last_name |
|---|--------------------|-------------------|--------------|-----------------------|----------------------|
| 1 | Ranga              | K                 | AZ-900       | in28minutes           | cloud                |
| 2 | Ranga              | K                 | DP-900       | in28minutes           | cloud                |
| 3 | Sathish            | M                 | AZ-900       | in28minutes           | cloud                |
| 4 | Sathish            | M                 | DP-900       | in28minutes           | cloud                |
| 5 | Ramesh             | S                 | AZ-900       | in28minutes           | cloud                |
| 6 | Ramesh             | S                 | Google Cloud | in28minutes           | cloud                |

## Normalized - Student

| id | first_name | last_name |
|----|------------|-----------|
| 1  | Ranga      | K         |
| 2  | Sathish    | M         |
| 3  | Ramesh     | S         |

## Normalized - Instructor

| id | first_name  | last_name |
|----|-------------|-----------|
| 1  | in28minutes | cloud     |

# Normalization example - 2

## Normalized - Course

| id | title        | instructor_id |
|----|--------------|---------------|
| 1  | AZ-900       | 1             |
| 2  | DP-900       | 1             |
| 3  | Google Cloud | 1             |

## Normalized - Course\_Student

| id | course_id | student_id |
|----|-----------|------------|
| 1  | 1         | 1          |
| 2  | 2         | 1          |
| 3  | 1         | 2          |
| 4  | 2         | 2          |
| 5  | 1         | 3          |
| 6  | 3         | 3          |

# Transactions



Database

- **Transaction:** Sequence of operations that need to be atomic
  - All operations are successful (commit) OR NONE are successful (rollback)
  - **Example:** Transfer \$10 from Account A to B
    - Operation 1: Reduce \$10 from Account A
    - Operation 2: Add \$10 to Account B
    - If Operation 1 is successful and Operation 2 fails - Inconsistent state
      - You don't want that!
- **Properties: ACID (Atomicity, Consistency, Isolation, Durability)**
  - **Atomicity:** Each transaction is atomic (either succeeds completely, or fails completely)
  - **Consistency:** Database must be consistent before and after the transaction
  - **Isolation:** Multiple Transactions occur independently
  - **Durability:** Once a transaction is committed, it remains committed even if there are system failures (a power outage, for example)
- **Remember:** Supported in all Relational Databases

# Azure SQL Database - Purchase Models

In **28**  
Minutes



SQL Database

- **vCore-based:** Choose between provisioned or serverless compute
  - OPTIONAL: Hyperscale (Autoscale storage)
  - Higher compute, memory, I/O, and storage limits
  - Supports BYOL
  - **Serverless Compute:** Database is paused during inactive periods
    - You are only billed for storage during inactive periods
    - If there is any activity, database is automatically resumed
- **DTU-based:** Bundled compute and storage packages
  - Balanced allocation of CPU, memory and IO
  - Assign DTUs (relative - Double DTU => Double resources)
  - Recommended when you want to keep things simple
    - You CANNOT scale CPU, memory and IO independently
  - Use DTUs for **small and medium databases** (< few hundred DTUs)

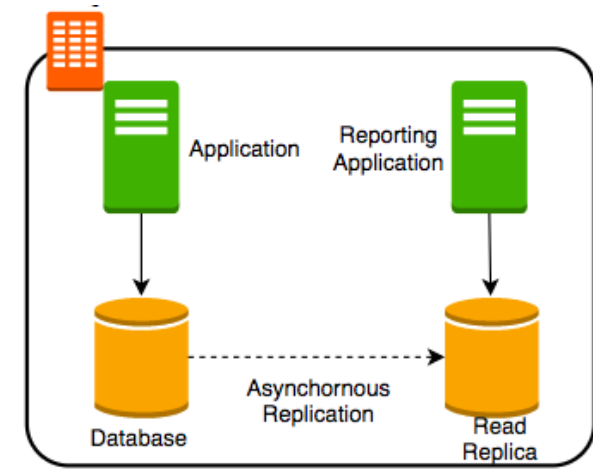
# Azure SQL Database - Important Features

| Feature                | Description   |
|------------------------|---|
| <b>Single database</b> | Great fit for modern, cloud-born applications<br>Fully managed database with predictable performance<br>Hyperscale storage (up to 100TB)<br>Serverless compute        |
| <b>Elastic pool</b>    | Cost-effective solution for multiple databases with variable usage patterns<br>Manage multiple databases within a fixed budget  |
| <b>Database server</b> | Database servers are used to manage groups of single databases and elastic pools.<br>Things configured at Database server level: Access management, Backup management |



# Azure SQL Database - Remember

- **Prerequisites** to connect and query from Azure SQL database:
  - **1:** Connection Security: Database should allow connection from your IP address
  - **2:** User should be created in the database
  - **3:** User should have grants (permissions) to perform queries - Select, Insert etc.
- Use **BYOL** to reduce license costs
- Use read-only replicas (Read scale-out) for offloading read-only query workloads



# Azure SQL managed instance

In **28**  
Minutes



- Another **Fully Managed Service** for Microsoft SQL Server
- What's New: **Near 100% SQL Server feature compatibility**
- Recommended when migrating on premise SQL Servers to Azure
- **Azure SQL managed instance features NOT in Azure SQL Database**
  - Cross-database queries (and transactions) within a single SQL Server instance
  - Database Mail
  - Built in SQL Server Agent
    - Service to execute scheduled administrative tasks - jobs in SQL Server
  - Native virtual network support
- Supports **only vCore-based** purchasing model
- (Remember) SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS), Polybase: NOT supported by both Azure SQL Database and SQL Managed Instance

# SQL Server in Azure - Summary

| Service                              | Description   |
|--------------------------------------|---|
| SQL Server on Azure Virtual Machines | Provides full administrative control over the SQL Server instance and underlying OS for migration to Azure  |
| Azure SQL Database                   | Fully Managed Service for Microsoft SQL Server.<br>Recommended for cloud-born applications  |
| Azure SQL managed instance           | Full (Near 100%) SQL Server access and feature compatibility<br>Recommended for migrating on-premise SQL Server databases<br>Azure SQL managed instance ONLY features: Cross-database queries, Database Mail Support, SQL Server Agent etc. |

# Azure database for MySQL

In 28  
Minutes

- Fully managed, scalable MySQL database
- Supports 5.6, 5.7 and 8.0 community editions of MySQL
- 99.99% availability
  - Choose single zone or zone redundant high availability
- Automatic updates and backups
- **Alternative:** Azure Database for MariaDB
  - MariaDB: community-developed, commercially supported fork of MySQL



Azure Database MySQL

# Azure Database for PostgreSQL

In **28**  
Minutes

- Fully managed, intelligent and scalable PostgreSQL
- 99.99% availability
  - Choose single zone or zone redundant high availability
- Automatic updates and backups
- **Single Server and Hyperscale Options**
  - Hyperscale: Scale to hundreds of nodes and execute queries across multiple nodes



Azure Database  
PostgreSQL

# Relational Data - Scenarios

| Scenario   | Solution                                       |
|--|--|
| You are migrating a Microsoft SQL Server database to cloud. You want full access to OS and Microsoft SQL Server installation.  | SQL Server on VM                               |
| You are migrating a Microsoft SQL Server database to cloud. You do NOT need full access to OS and Microsoft SQL Server installation. However, you need access to Database Mail and SQL Server Agent. | Azure SQL Managed Instance                     |
| You want create a new managed Microsoft SQL Server database in cloud   | Azure SQL Database, Azure SQL Managed Instance |
| Which category of SQL is this? <b>GRANT SELECT ON course TO user1</b>  | Data Control Language (DCL)                    |

# Relational Data - Scenarios - 2

| Scenario  | Solution   |
|---|--|
| Which category of SQL is this? <b>create table course</b><br>( . . . )                                  | Data Definition Language (DDL)                                 |
| Your queries on a relational databases are slow. What is the first thing that you would consider doing? | Check if there is an index                                     |
| Your colleague asked you to normalize your tables. What should be your goals?                           | High Data Integrity & Minimum Data Redundancy (or Duplication) |
| How can you offload read-only workloads from Azure SQL database?  | Read-only replicas (Read scale-out)                            |

# Azure Cosmos DB



# Relational vs Non Relational Data - Quick Overview

In **28**  
Minutes

- **Relational Data (Structured Data)**

- **OLTP:** Azure SQL Database, Azure SQL Managed Instance, SQL Server on Azure VMs, Azure Database for PostgreSQL, MariaDB, MySQL
- **OLAP:** Azure Synapse Analytics

- **Non Relational Data (Semi Structured/Unstructured Data)**

- **Semi Structured - Document (JSON)**
  - Azure Cosmos DB SQL API and Cosmos DB MongoDB API
- **Semi Structured - Key-Value**
  - Azure Cosmos DB Table API, Azure Table Storage
- **Semi Structured - Column-Family**
  - Azure Cosmos DB Cassandra API
- **Semi Structured - Graph**
  - Azure Cosmos DB Gremlin API
- **Unstructured Data**
  - Block Storage (Azure Disks), File Storage (Azure Files), Object Storage (Azure Blob Storage)



Cosmos DB



SQL Database



Azure Database MySQL



Azure Storage

# Azure Cosmos DB

In **28**  
Minutes

- Fully managed NoSQL database service
- **Global database:** Automatically replicates data across multiple Azure regions
  - Single-digit millisecond response times
  - 99.999% availability
  - Automatic scaling (serverless) - Storage and Compute
  - Multi-region writes
  - Data distribution to any Azure region with the click of a button
    - Your app doesn't need to be paused or redeployed to add or remove a region
- **Structure:** Azure Cosmos account(s) > database(s) > container(s) > item(s)



Cosmos DB

# Azure Cosmos DB APIs

In **28**  
Minutes

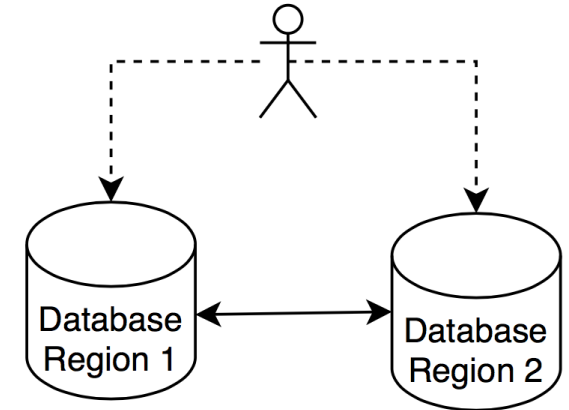
- **Core(SQL)**: SQL based API for working with documents
- **MongoDB**: Document with MongoDB API
  - Move existing MongoDB workloads
- **Table**: Key Value
  - Ideal for moving existing Azure Table storage workloads
- **Gremlin**: Graph
  - Store complex relationships between data
- **Cassandra**: Column Family
- **REMEMBER**: You need a separate Cosmos DB account for each type of API



Cosmos DB

# Azure Cosmos DB - What is Different?

- Single-digit millisecond response times even if you scale to petabytes of data with millions of TPS
  - Horizontal scalability
- One thing I love about Azure Cosmos DB: **Flexibility**
  - Structure data **the way your application needs it**
    - Let the structure evolve with time
  - Provides a variety of consistency levels
    - Strong, Bounded staleness, Session, Consistent prefix, Eventual
  - If you are familiar with SQL but want to still use document database use SQL API
  - Options for key-value, column-family and graph databases

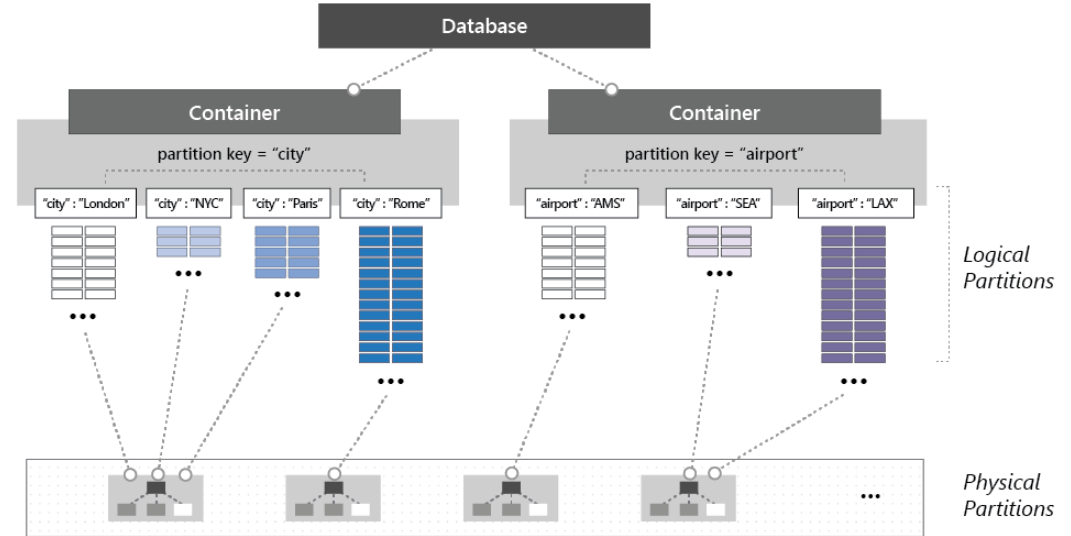


# Cosmos DB - Structure

| Entity    | SQL       | Cassandra | MongoDB    | Gremlin      | Table |
|-----------|-----------|-----------|------------|--------------|-------|
| Database  | Database  | Keyspace  | Database   | Database     | NA    |
| Container | Container | Table     | Collection | Graph        | Table |
| Item      | Item      | Row       | Document   | Node or edge | Item  |

# Cosmos DB - Logical and Physical Partitions

- Each container is **horizontally partitioned** in an Azure region
  - ALSO distributed to all Azure regions associated with the Cosmos DB account
- Items in a container **divided into logical partitions** based on the partition key
- Cosmos DB take care of categorizing logical partitions into physical partitions
- Ensures high availability and durability



(<https://docs.microsoft.com>)

# Cosmos DB - Provisioned throughput vs Serverless

| Factor                    | Provisioned throughput                                     | Serverless   |
|---------------------------|--|--|
| Description               | Provision throughput in Request Units per second           | No need to provision capacity. Auto scales to meet request load. |
| What are you billed for?  | RUs provisioned per hour (usage does NOT matter) + Storage | per-hour RUs consumed + Storage                                  |
| When to use?              | Continuous predictable traffic                             | Intermittent, unpredictable traffic                              |
| Multi Regions             | Yes  | No - only in 1 Azure region                                      |
| Max storage per container | No limit   | 50 GB  |
| Performance               | < 10 ms latency for point-reads and writes                 | < 10 ms latency for point-reads and < 30 ms for writes           |

# Azure Cosmos DB - Scenarios

| Scenario  | Solution   |
|---|--|
| How can you increase storage associated with Azure Cosmos DB?                 | Automatic scaling (serverless)                                 |
| What is the high level structure of storing data in Azure Cosmos DB?          | Azure Cosmos account(s) > database(s) > container(s) > item(s) |
| How are items in a container divided into logical partitions?                 | Using partition key  |
| You want to store data for a social networking app with complex relationships | Gremlin API  |
| You want SQL based API for working with documents                             | Core(SQL) API  |
| You want to move existing MongoDB workloads to Azure                          | MongoDB API  |



# Azure Storage

# Relational vs Non Relational Data - Quick Overview

In **28**  
Minutes

- **Relational Data (Structured Data)**

- **OLTP:** Azure SQL Database, Azure SQL Managed Instance, SQL Server on Azure VMs, Azure Database for PostgreSQL, MariaDB, MySQL
- **OLAP:** Azure Synapse Analytics

- **Non Relational Data (Semi Structured/Unstructured Data)**

- **Semi Structured - Document (JSON)**
  - Azure Cosmos DB SQL API and Cosmos DB MongoDB API
- **Semi Structured - Key-Value**
  - Azure Cosmos DB Table API, Azure Table Storage
- **Semi Structured - Column-Family**
  - Azure Cosmos DB Cassandra API
- **Semi Structured - Graph**
  - Azure Cosmos DB Gremlin API
- **Unstructured Data**
  - Block Storage (Azure Disks), File Storage (Azure Files), Object Storage (Azure Blob Storage)



Cosmos DB



SQL Database



Azure Database MySQL



Azure Storage

# Azure Storage

- Managed Cloud Storage Solution
  - Highly available, durable and massively scalable (upto few PetaBytes)
- Core Storage Services:
  - **Azure Disks:** Block storage (hard disks) for Azure VMs
  - **Azure Files:** File shares for cloud and on-premises
  - **Azure Blobs:** Object store for text and binary data
  - **Azure Queues:** Decouple applications using messaging
  - **Azure Tables:** NoSQL store (Very Basic)
    - Prefer Azure Cosmos DB for NoSQL
- (PRE-REQUISITE) Storage Account is needed for Azure Files, Azure Blobs, Azure Queues and Azure Tables



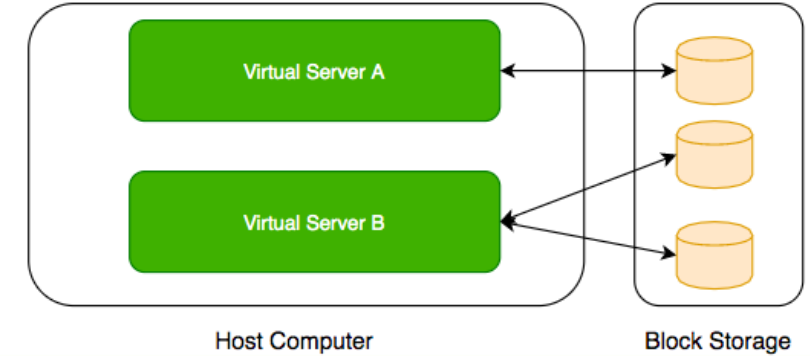
Azure Storage

# Azure Storage - Data Redundancy

| Option                            | Redundancy  | Discussion                              |
|-----------------------------------|---|---|
| Locally redundant storage (LRS)   | Three synchronous copies in same data center                              | Least expensive and least availability  |
| Zone-redundant storage (ZRS)      | Three synchronous copies in three AZs in the primary region               |   |
| Geo-redundant storage (GRS)       | LRS + Asynchronous copy to secondary region (three more copies using LRS) |   |
| Geo-zone-redundant storage (GZRS) | ZRS + Asynchronous copy to secondary region (three more copies using LRS) | Most expensive and highest availability |

# Block Storage

- Use case: Hard-disks attached to your computers
- Typically, ONE Block Storage device can be connected to ONE virtual server
- HOWEVER, you can connect multiple different block storage devices to one virtual server



# Azure Disks Storage

In **28**  
Minutes

- **Disk storage: Disks for Azure VMs**

- **Types:**

- **Standard HDD:** Recommended for Backup, non-critical, infrequent access
    - **Standard SSD:** Recommended for Web servers, lightly used enterprise applications and dev/test environments
    - **Premium SSD disks:** Recommended for production and performance sensitive workloads
    - **Ultra disks (SSD):** Recommended for IO-intensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads

- Premium and Ultra provide very high availability

- **Managed vs Unmanaged Disks:**

- **Managed Disks are easy to use:**

- Azure handles storage
    - High fault tolerance and availability

- **Unmanaged Disks are old and tricky (Avoid them if you can)**

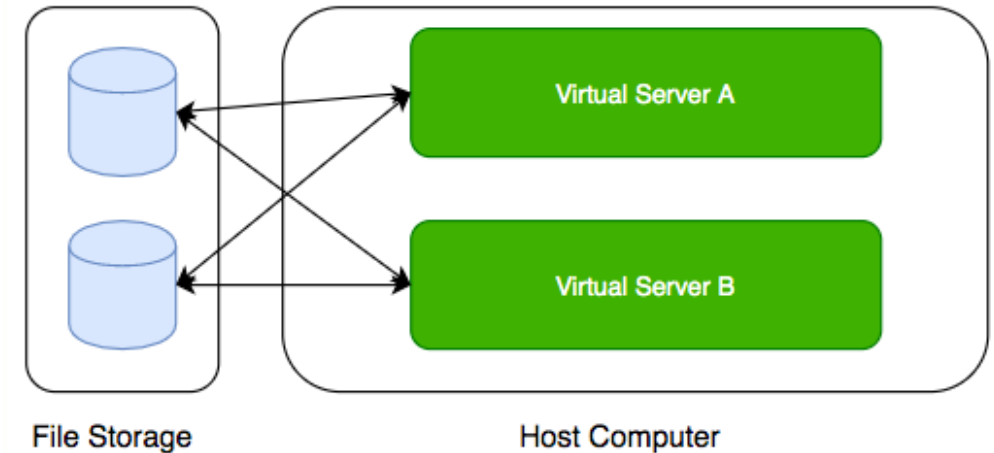
- You need to manage storage and storage account
    - Disks stored in Containers (NOT Docker containers. Completely unrelated.)



Azure Storage

# Azure Files

- Media workflows need huge shared storage for things like video editing
- Enterprise users need a quick way to share files in a secure & organized way
- **Azure Files:**
  - Managed File Shares
  - Connect from multiple devices concurrently:
    - From cloud or on-premises
    - From different OS: Windows, Linux, and macOS
  - Supports Server Message Block (SMB) and Network File System (NFS) protocols
  - Usecase: Shared files between multiple VMs (example: configuration files)



# Azure Blob Storage

In **28**  
Minutes

- **Azure Blob Storage:** Object storage in Azure
- **Structure:** Storage Account > Container(s) > Blob(s)
- Store massive volumes of unstructured data
  - **Store all file types** - text, binary, backup & archives:
    - Media files and archives, Application packages and logs
    - Backups of your databases or storage devices
- **Three Types of Blobs**
  - Block Blobs: Store text or binary files (videos, archives etc)
  - Append Blobs: Store log files (Ideal for append operations)
  - Page Blobs: Foundation for Azure IaaS Disks (512-byte pages up to 8 TB)
- **Azure Data Lake Storage Gen2:** Azure Blob Storage Enhanced
  - Designed for enterprise big data analytics (exabytes, hierarchical)
  - Low-cost, tiered storage, with high availability/disaster recovery



Azure Storage



# Azure Blob Storage - Access Tiers

In **28**  
Minutes

- **Different kinds of data** can be stored in Blob Storage
  - Media files, website static content
  - Backups of your databases or storage devices
  - Long term archives
- Huge variations in **access patterns**
- Can I pay a cheaper price for objects I access less frequently?
  - **Access tiers**
    - **Hot:** Store frequently accessed data
    - **Cool:** Infrequently accessed data stored for min. 30 days
    - **Archive:** Rarely accessed data stored for min. 180 days
      - Lowest storage cost BUT Highest access cost
      - Access latency: In hours
      - To access: **Rehydrate** (Change access tier to hot or cool) OR
        - Copy to another blob with access tier hot or cool
    - You can **change access tiers** of an object **at any point in time**



Azure Storage

# Azure Storage - Remember

In **28**  
Minutes

- **Azure Queues:** Decouple applications using messaging
- **Azure Tables:** NoSQL store (Very Basic)
  - A key/value store
  - Store and retrieve values by key
  - Supports simple query, insert, and delete operations
  - Cosmos DB Table API is recommended as key/value store for newer usecases (supports multi-master in multiple regions)
  - Azure Tables only supports read replicas in other regions
    - **GRS or GZRS:** Data in secondary region is generally NOT available for read or write access
      - Available for read or write only in case of failover to the secondary region
    - To enable round the clock read access:
      - Use read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS)



Azure Storage

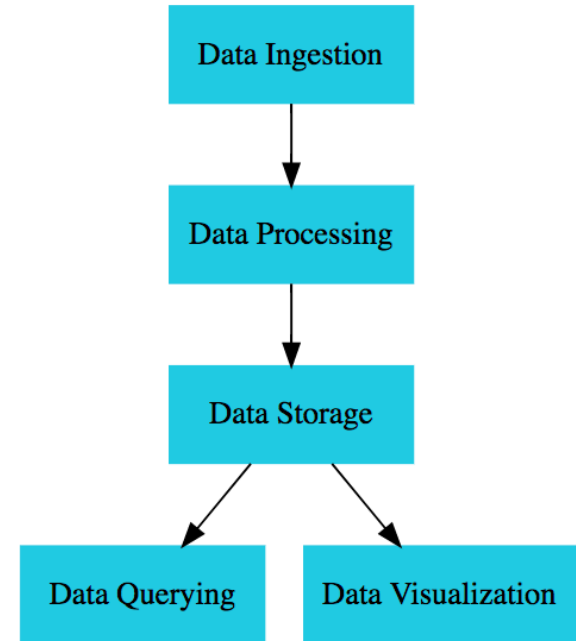
# Azure Storage - Scenarios

| Scenario  | Solution                          |
|---|-----------------------------------|
| What is needed before storing data to Azure Files, Azure Blobs, Azure Queues and Azure Tables?  | Storage Account                   |
| You have a Storage Account and you are making use of Azure Blob Storage. You want to create a new file share. Is it mandatory to create a new Storage Account?                | No                                |
| You want highest availability for data in your Storage Account  | Geo-zone-redundant storage (GZRS) |
| Which service supports Server Message Block (SMB) and Network File System (NFS) protocols?  | Azure Files                       |
| You are not planning to access your data in Azure Blob storage for a few years. You can wait for a few hours when you need to access the data. How can you reduce your costs? | Move data to Archive tier         |

# Data Analytics

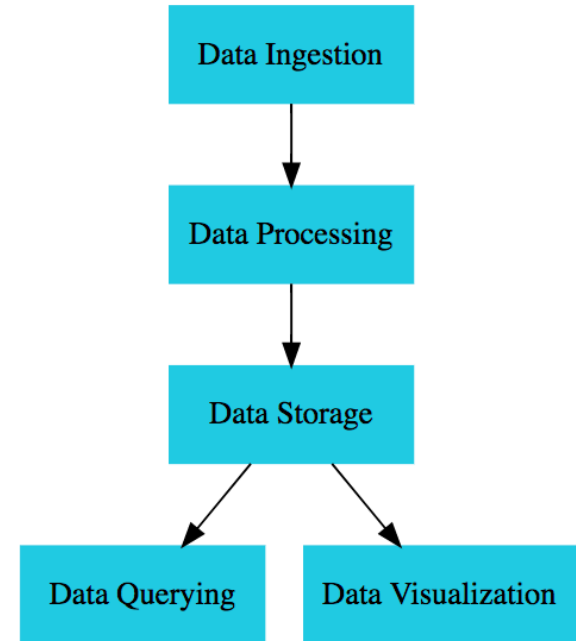
# Data Analytics

- **Goal:** Convert raw data to intelligence
  - Uncover trends and discover meaningful information
  - Find new opportunities and identify weaknesses
  - Increase efficiency and improve customer satisfaction
  - Make appropriate business decisions
- Raw data can be from different sources:
  - Customer purchases, bank transactions, stock prices, weather data, monitoring devices etc
- **Approach:** Ingest => Process => Store (data warehouse or a data lake) => Analyze
- Ex: Decide future sales using past customer behavior
- Ex: Faster diagnosis & treatment using patient history



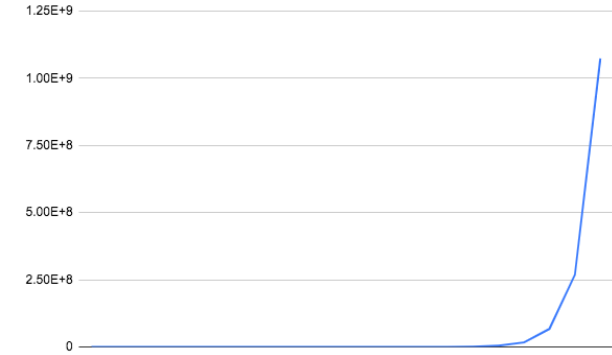
# Data Analytics Work Flow

- **Data Ingestion:** Capture raw data
  - From various sources (stream or batch)
    - Example: Weather data, sales records, user actions - websites ..
- **Data Processing:** Process data
  - Raw data is not suitable for querying
    - Clean (remove duplicates), filter (remove anomalies) and/or aggregate data
    - Transform data to required format (Transformation)
- **Data Storage:** Store to data warehouse or data lake
- **Data Querying:** Run queries to analyze data
- **Data Visualization:** Create visualizations to make it easier to understand data and make better decisions
  - Create dashboards, charts and reports (capture trends)
  - Help business spot trends, outliers, and hidden patterns in data



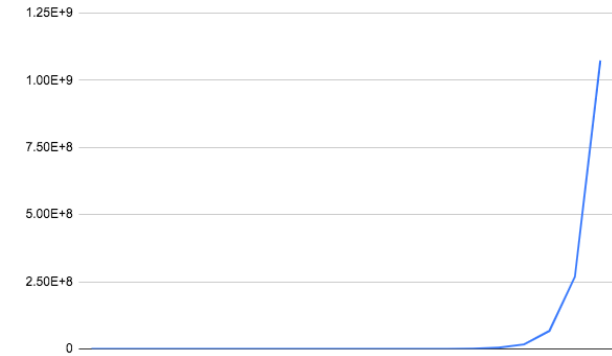
# Data Analysis Categories

- **Descriptive analytics:** What's happening?
  - Based on historical/current data
  - Monitor status (of KPIs) and generate alerts
  - Example: Generating reports (current vs planned)
- **Diagnostic analytics:** Why is something happening?
  - Take findings from descriptive analytics and dig deeper
  - Example: Why did sales increase last month?
  - Example: Why are sales low in Netherlands?
- **Predictive analytics:** What will happen?
  - Predict probability based on historical data
  - Mitigate risk and identify opportunities
  - Example: What will be the future demand?
  - Example: Calculate probability of something happening in future



# Data Analysis Categories - 2

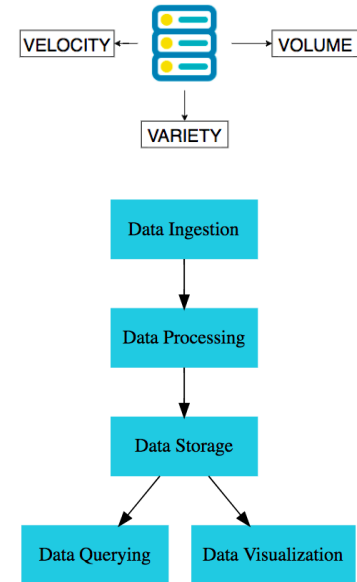
- **Prescriptive analytics:** What actions should we take?
  - Use insights from predictive analytics and make data-driven informed decisions
  - Still in early stages
  - Example: What can I do to increase probability of this course being successful in future?
- **Cognitive analytics:** Make analytic tools to think like humans
  - Combine traditional analytics techniques with AI and ML features
  - Examples: Speech to text (transcription or subtitles), text to speech, Video Analysis, Image Analysis, Semantic Analysis of Text (Analyze reviews)



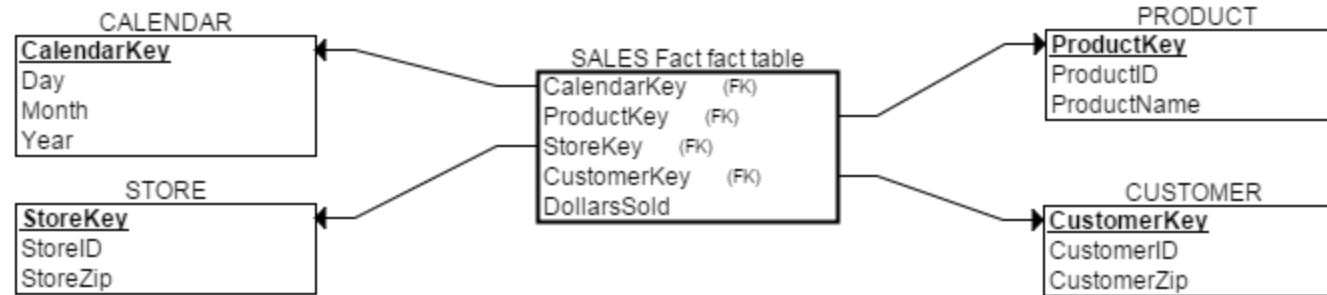


# Big Data - Terminology and Evolution

- **3Vs of Big Data**
  - **Volume:** Terabytes to Petabytes to Exabytes
  - **Variety:** Structured, Semi structured, Unstructured
  - **Velocity:** Batch, Streaming ..
- **Terminology: Data warehouse vs Data lake**
  - **Data warehouse:** PBs of Storage + Compute (Typically)
    - Data stored in a format ready for specific analysis! (processed data)
    - Examples: Teradata, BigQuery(GCP), Redshift(AWS), Azure Synapse Analytics
    - Typically uses specialized hardware
  - **Data lake:** Typically retains all raw data (compressed)
    - Typically object storage is used as data lake
    - Amazon S3, Google Cloud Storage, Azure Data Lake Storage Gen2 etc..
    - Flexibility while saving cost
    - Perform ad-hoc analysis on demand
    - Analytics & intelligence services (even data warehouses) can directly read from data lake
    - Azure Synapse Analytics, BigQuery(GCP) etc..



# Data warehouse Best Practice - De-normalized Star Schema



- How do you structure data for quick analysis in a data warehouse?
  - Option: **Star Schema**
  - Modeling approach most widely used by relational data warehouses
- **Each Table classified as "Dimension" or "Fact":**
  - **Fact** tables: Quantitative data - Data generated in a transactional system (typically)
    - Contains observations or events (sales orders, stock balances, temperatures ...)
  - **Dimension** tables: Contain descriptive attributes related to fact data
    - Example: Product, Customer, Store, Calendar
- **Advantage:** Star schemas are de-normalized and are easier to query

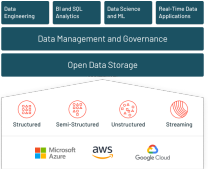
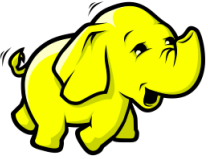
# Data Analytics: 3 Azure Specific Services

- **Azure Synapse Analytics:** End-to-end analytics solutions
  - Data integration + Enterprise data warehouse + Data analytics
  - Create SQL and Spark pools to analyze data
- **Azure Data Factory: Fully managed serverless** service to build complex data pipelines
  - Extract-transform-load (ETL), extract-load-transform (ELT) and data integration
- **Power BI:** Create visualization around data
  - Unify data and create BI reports & dashboards

# Big Data - Hadoop, Spark and Databricks

In **28**  
Minutes

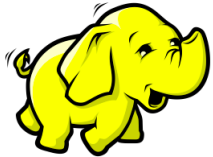
- Hadoop based approaches:
  - **Apache Hadoop:** Create datasets with variety of data. Get intelligence.
    - Runs on commodity servers with attached storage (Large clusters - thousands of nodes)
    - **Hadoop Distributed File System (HDFS):** Primary data storage
    - **MapReduce:** Write Java, Python, .. apps to process data
      - Enables massive parallelization
    - **HIVE:** Query using SQL
    - **Apache Spark:** How about processing in-memory?
      - Really fast: Can be up to 100 times faster than MapReduce (if you make sufficient memory available)
      - Supports Java, Python, R, SQL and Scala programming languages
      - Run data analytics, data processing and machine learning workloads
      - Has become very popular and is offered as a separate service in most cloud platforms!
  - **Databricks:** Web-based platform for working with Spark
    - Centralized platform for machine learning, streaming analytics and business intelligence workloads
    - Founded by the creators of Apache Spark
    - **Automated cluster management**



# Hadoop and Spark in Azure

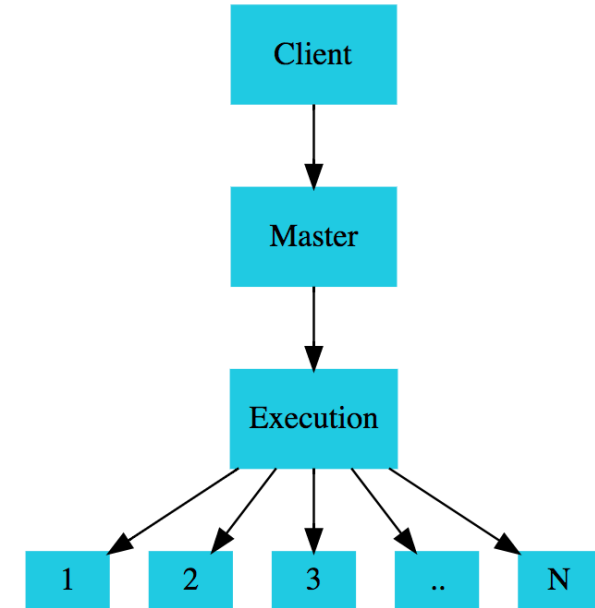
In **28**  
Minutes

- **Azure HDInsight:** Managed Apache Hadoop Azure service
  - Process big data with Hadoop, Spark
- **Azure Databricks:** Managed Apache Spark service
  - Premium Spark offering
  - Focused only on running Apache Spark workloads
  - Can consume data from Azure SQL Database, Event hubs, Cosmos DB
- **Other Azure Spark Integrations:**
  - **Azure Synapse Analytics:** Can run Spark jobs using "Apache Spark for Azure Synapse"
  - **Azure Data Factory:** Run pipelines involving Azure services like Azure HDInsight, Azure Databricks

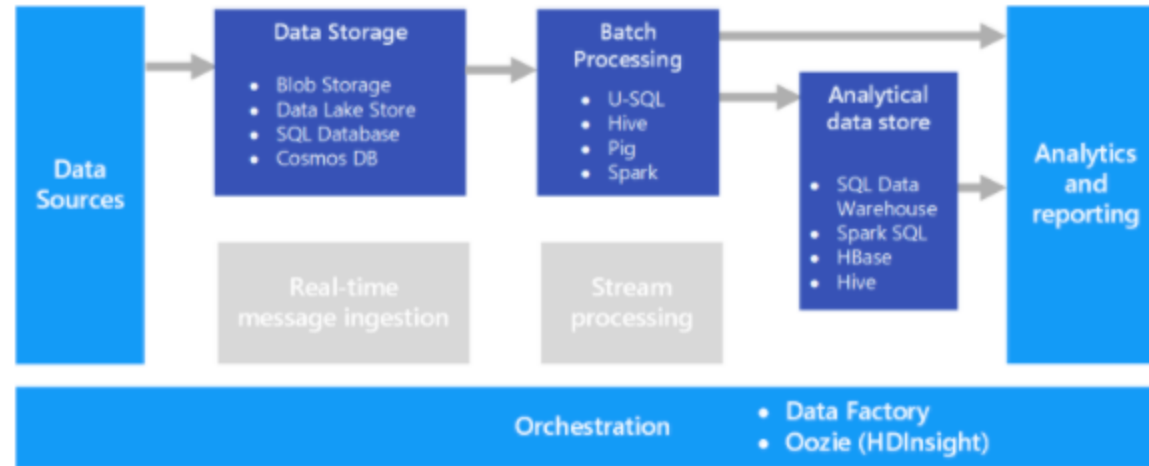


# Massive Parallel Processing (MPP)

- Split processing across multiple compute nodes
- Typically separate storage and compute
  - Use Data lake as storage (for example)
  - Scale compute on demand
- Examples: Spark, Azure Synapse Analytics
  - Some services run Spark in serverless mode!



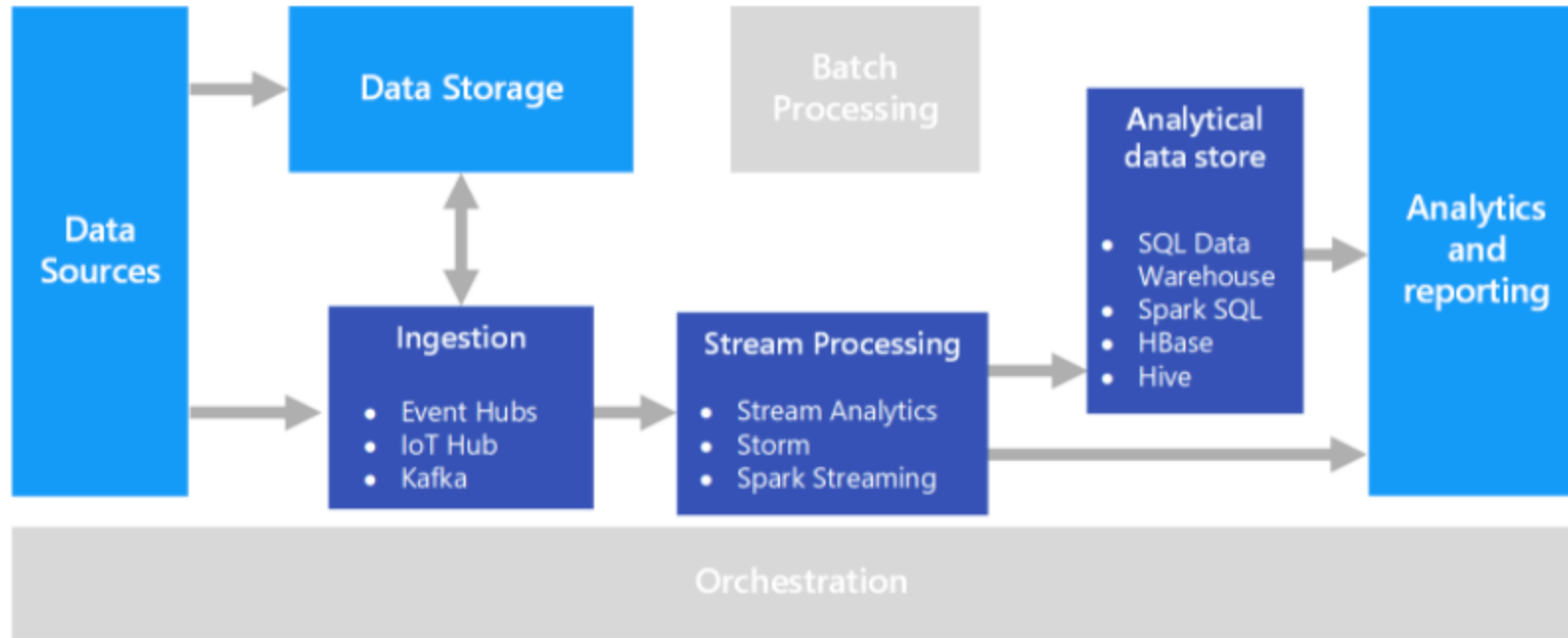
# Batch Pipelines



(<https://docs.microsoft.com>)

- **Batch Processing:** Buffering and processing data in groups
  - Define condition - how often to run? (every 6 hours or after 10K records)
  - **Advantages:** Process huge volumes of data during off-peak hours (overnight, for example)
    - Typically takes longer to run (minutes to hours to days)
  - **Example:** Read from storage (Azure Data Lake Store), process, and write to Relational Database or NoSQL Database or Data warehouse

# Streaming Pipelines



(<https://docs.microsoft.com>)

- **Streaming Processing:** Real-time data processing
  - Processing data as it arrives (in seconds or milliseconds)
  - Examples: Stock Market Data, Telemetry from IOT Devices, User action metrics from websites



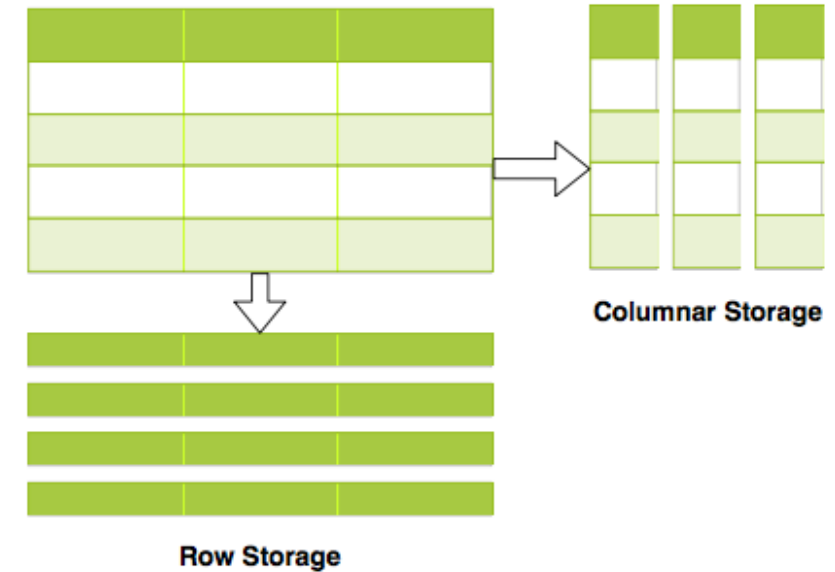
# Stream vs Batch Processing

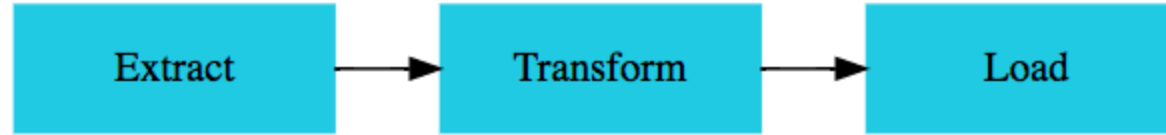
| Feature     | Batch   | Streaming   |
|-------------|---|---|
| Time Period | Process data in batches - all data from few hours to few days to few months | Process most recent data (last 30 seconds, for example).                                |
| Data Size   | Process large datasets efficiently  | Process individual records or micro batches containing a few records                    |
| Latency     | High - Typically few hours  | Low - Typically few seconds or milliseconds   |
| Usecase     | Use for performing complex storage or analysis                              | Used for storing individual records, simple aggregation or rolling average calculations |

# Apache Parquet

In 28  
Minutes

- Open source **columnar storage format**
- **High compression** because of columnar storage
- **Efficient storage** for big data workloads
- Introduced by the Apache Hadoop ecosystem
- **Supported by most big data platforms:**
  - Azure Data Factory supports Parquet for both read and write (Source and Sink)
  - Azure Data Lake Storage / Azure Blob Storage - Store data in Parquet format
  - Azure Synapse Analytics can be used to store tabular representation of data in Parquet format



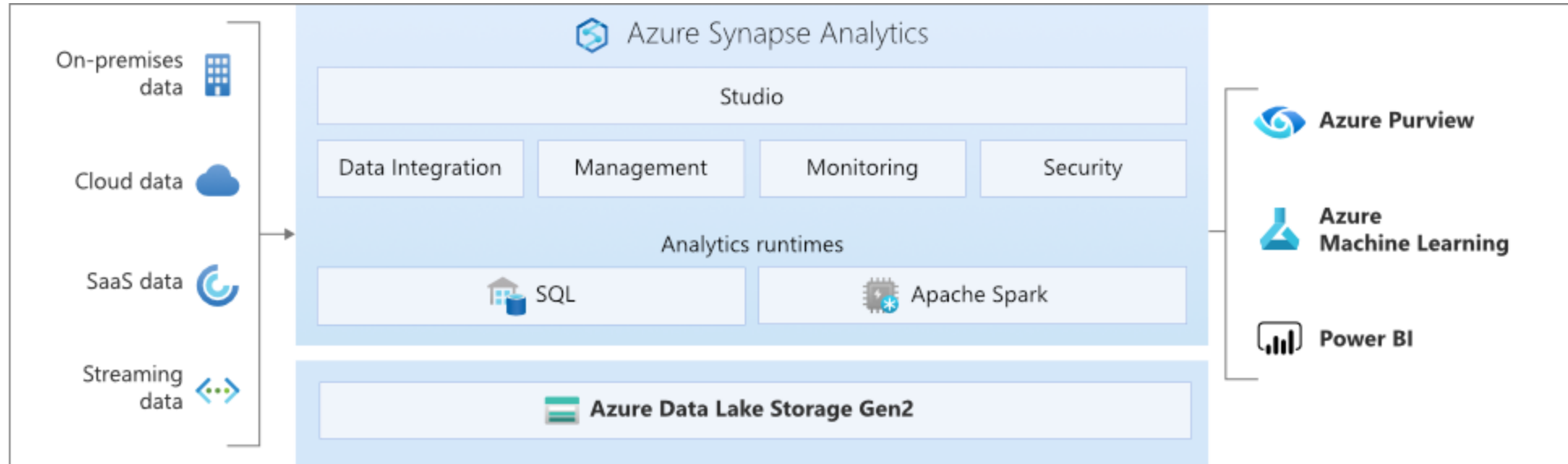


- **ETL (Extract, Transform, and Load):** Retrieve data, process and store it
- Data can be from multiple sources
- Recommended for simple processing:
  - Basic data cleaning tasks, de-duplicating data, formatting data
  - Example: Ensure data privacy and compliance
    - Removing sensitive data before it reaches analytical data models
- Can run each of the phases in parallel
  - While extract is going on, you can transform data which is already loaded



- **ELT (Extract, Load, and Transform):** Data is stored before it is transformed
- Uses an iterative approach (multiple steps) to process data in target system
- Needs a powerful target datastore:
  - Target datastore should be able to perform transformations
- Advantage: Does NOT use a separate transformation engine
- Typical target data stores: Hadoop cluster (using Hive or Spark), Azure Synapse Analytics
  - Enables use of massively parallel processing (MPP) capabilities of data stores

# Azure Synapse Analytics

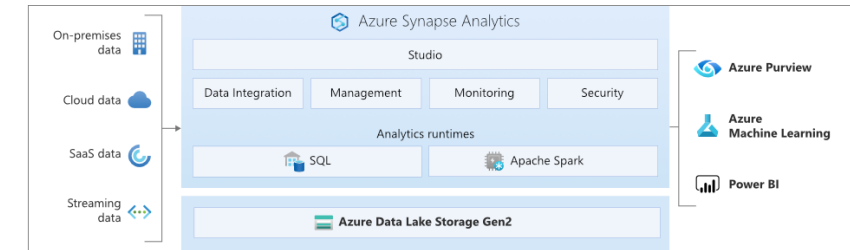


(<https://docs.microsoft.com>)

- Develop **end-to-end analytics** solutions
  - Data integration + Enterprise data warehouse + Data analytics
  - SQL technologies + Spark technologies + Pipelines
  - Full integration with Power BI, Cosmos DB, and Azure ML

# Azure Synapse Analytics - Workflow

- In a workspace, create pipelines for:
  - **Data Ingestion:**
    - Ingest data from 90+ data sources (Cosmos DB, AWS, GCP..)
    - Stream data into SQL tables
  - **Data Storage:** Datasets - Azure Storage, Azure Data Lake Storage
    - Formats: Parquet, CSV, JSON ..
  - **Data Processing:** Mix & match SQL and Spark
    - **SQL pool:** SQL Database supporting distributed T-SQL queries
      - Two consumption models: dedicated and serverless
      - Recommended for complex reporting & data ingestion using Polybase
      - SQL Pool can be paused to reduce compute costs
    - **Apache Spark pools:** Run Spark based workloads
      - 1: Create Spark data analysis notebooks OR
      - 2: Run batch Spark jobs (jar files)
      - Recommended for data preparation and ML



(<https://docs.microsoft.com>)

# Azure Data Factory

In **28**  
Minutes



Data Factory

- **Fully managed serverless** service to build complex data pipelines:
  - Extract-transform-load (ETL), extract-load-transform (ELT) and data integration
    - **90 built-in** connectors
    - Ingest data from:
      - **Big Data sources** like Amazon Redshift, Google BigQuery
      - **Enterprise data warehouses** like Oracle Exadata, Teradata
      - All Azure data services
  - **Build data flows** to transform data
    - Integrate with services like Azure HDInsight, Azure Databricks, Azure Synapse Analytics for data processing
  - Move SQL Server Integration Services (SSIS) packages to cloud
- CI/CD support with Azure Devops

# Demo - Azure Data Factory and Synapse Analytics

- Create a Data Lake Storage Account Gen2
- Create a SQL Server Database
- Task: Extract data from SQL Server to CSV file



# Azure Data Lake Storage (Gen2)

In **28**  
Minutes

- Blob storage + Hierarchical directory structure
- Configure permissions(RBAC) at file and directory level
- Fully compatible with Hadoop Distributed File System (HDFS)
  - Apache Hadoop workloads can directly access data in Azure Data Lake Storage
- **Three main elements:**
  - **Data Lake Store:** Azure Data Factory, Azure Databricks, Azure HDInsight, Azure Data Lake Analytics, and Azure Stream Analytics can read directly
  - **Data Lake Analytics:** Run analytics jobs using U-SQL
  - **HDInsight:** Run Hadoop jobs



Data Lake Storage

# Azure Data Factory - Components

In **28**  
Minutes



Data Factory

- **Pipeline:** Logical group of activities that can be scheduled
  - You can chain activities in a pipeline
  - You can run activities sequentially or in parallel
  - A pipeline can execute other pipelines
- **Activity:** Represents a step in a pipeline (an action to be performed)
  - **Copy Activity:** Copy data from one store to another store
    - Example: Copy CSV from Blob Storage to a Table in SQL Database
  - **Three types** of activities: Data movement, Data transformation, Control activities
- **Data Flow:** Create and manage data transformation logic
  - Build reusable library of data transformation routines
  - Executes logic on a Spark cluster:
    - You don't need to manage the cluster (it is spun up and down automatically as needed)
- **Control flow:** Orchestrate pipeline activity based on output of another pipeline activity

# Azure Data Factory - Components - 2

In 28  
Minutes

- **Linked Service:** Used to connect to an external source
  - Connect to different sources like Azure Storage Blob, SQL Databases etc
- **Dataset:** Representation of data structures within data stores
- **Integration Runtime:** Compute infrastructure used by Azure Data Factory allowing you to perform
- **Triggers:** Trigger pipeline at a specific times



Data Factory

- **Power BI: Unify data and create BI reports & dashboards**
  - Integrates with all Azure analytics services
    - Azure Synapse Analytics to Azure Data Lake Storage
  - **Power BI Components**
    - **Power BI Service:** Online SaaS (Software as a Service) service
      - Power BI online - [app.powerbi.com](https://app.powerbi.com)
      - Create/share reports and dashboards
    - **Power BI Desktop:** Windows desktop application to create and share reports
      - More data sources, Complex modeling and transformations
    - **Power BI Report Builder:** Standalone tool to author paginated reports
    - **Power BI Mobile Apps:** Apps for Windows, iOS, and Android devices
  - **Typical Power BI Workflow:**
    - **1:** Create a report with Power BI Service/Desktop (or paginated report with Power BI Report Builder)
    - **2:** Share it to the Power BI service
    - **3:** View and interact with report (and create dashboards) using Power BI service
      - Reports can also be accessed from Power BI mobile



Power BI

# Power BI Dashboard

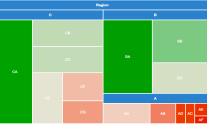
- **Workspace:** Container for dashboards, reports, workbooks & datasets
  - **Dataset:** Collection of data
    - Can be a file(Excel, CSV etc) or a database
      - Azure SQL Database, Azure Synapse Analytics, Azure HDInsight, ..
    - Each dataset can be used in multiple reports
  - **Report:** One or more pages of visualizations
    - Highly interactive and highly customizable
    - All data for a report comes from a single dataset
    - A report can be used in multiple dashboards
  - **Paginated Reports:** Create pixel perfect multi page reports for printing & archiving (PDF/Word)
    - Create in "Power BI Report Builder" and publish to use in Power BI service
  - **Dashboard:** Single page - visualizations from one or more reports
    - Technically a canvas with multiple tiles
    - Monitor the most important information at one glance and dig deeper, if needed
      - You can select a tile and go to a report page to dig deeper



Power BI

# Visualization Options

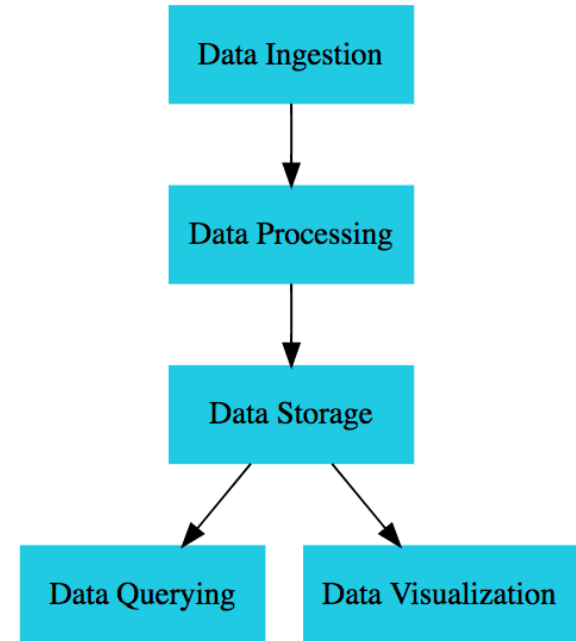
- **Bar and column charts:** Most basic of charts
- **Line Charts:** Emphasize shape of a series of values over time
- **Pie Charts:** Displays division of total into different categories
- **Matrix:** Summarize data in a tabular structure
- **Treemap:** Charts of colored rectangles
- **Scatter:** Shows relationship between two numerical values
  - **Bubble chart:** Replace data points with bubbles
    - Bubble size represents a 3rd dimension
- **Filled map:** Show on a Map
- **Cards:** Single number
- **Link:** *Reference for all visualization options in Power BI*



| Category            | Sales Amount | Units   |
|---------------------|--------------|---------|
| 1) Home Appliances  | \$1,362,741  | 41,124  |
| 2) Pet and Value    | \$6,155,111  | 44,589  |
| 3) Home Appliances  | \$1,362,741  | 41,124  |
| 4) Pet and Value    | \$6,155,111  | 44,589  |
| 5) Home Appliances  | \$1,362,741  | 41,124  |
| 6) Pet and Value    | \$6,155,111  | 44,589  |
| 7) Home Appliances  | \$1,362,741  | 41,124  |
| 8) Pet and Value    | \$6,155,111  | 44,589  |
| 9) Home Appliances  | \$1,362,741  | 41,124  |
| 10) Pet and Value   | \$6,155,111  | 44,589  |
| 11) Home Appliances | \$1,362,741  | 41,124  |
| 12) Pet and Value   | \$6,155,111  | 44,589  |
| 13) Home Appliances | \$1,362,741  | 41,124  |
| 14) Pet and Value   | \$6,155,111  | 44,589  |
| 15) Home Appliances | \$1,362,741  | 41,124  |
| 16) Pet and Value   | \$6,155,111  | 44,589  |
| 17) Home Appliances | \$1,362,741  | 41,124  |
| 18) Pet and Value   | \$6,155,111  | 44,589  |
| 19) Home Appliances | \$1,362,741  | 41,124  |
| 20) Pet and Value   | \$6,155,111  | 44,589  |
| Total               | \$40,645,973 | 116,609 |

# Data Analytics Work Flow - Data Ingestion

- **Data Ingestion: Capture raw data**
  - **Azure Data Factory:** data ingestion and transformation service
    - Ingest streaming and batch data
    - Data from on-premises and cloud
    - PolyBase: Run T-SQL queries on external data sources
      - PolyBase makes external data sources appear like tables
    - SQL Server Integration Services (SSIS): on-premises tool data integration and data transformation solution that is part of Microsoft SQL Server
      - Run existing SSIS packages as part of Azure Data Factory pipeline
  - **Spark:** Ingest streaming data
  - **IOT Hub:** Managed message hub for IoT devices
  - **Event Hub:** Big data streaming platform and event ingestion service

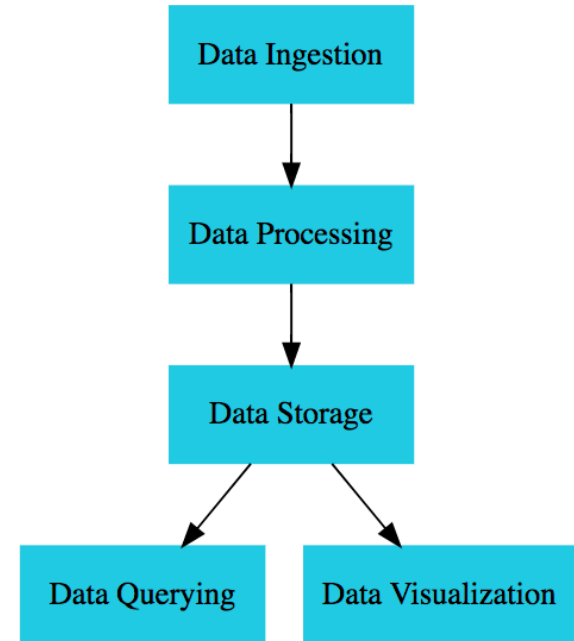


# Data Analytics Work Flow - Data Processing and Storage

In 28  
Minutes

- **Data Processing and Storage:**

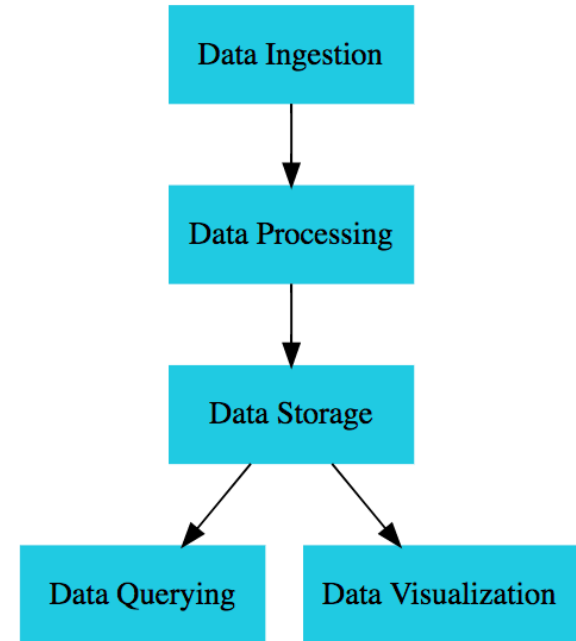
- **Azure Data Lake Storage Gen2:** Data lake storage
- **Azure Synapse Analytics:** Data processing can be done using:
  - 1: T-SQL - Query using SQL from databases, files, and Azure Data Lake storage
  - 2: Spark - Write and run Spark jobs using C#, Scala, Python, SQL etc
- **Azure Databricks:** Process data from Azure Blob storage, Azure Data Lake Store, Hadoop storage, flat files, databases, and data warehouses
  - Handle streaming data
- **Azure HDInsight:** Storage - Azure Data Lake storage
  - Analyze data using Hadoop Map/Reduce, Apache Spark, Apache Hive (SQL)
- **Azure Data Factory:** Build pipelines and data-driven workflows
  - Ingest data from relational and non-relational systems





# Data Analytics Work Flow - Querying and Visualization

- **Data Querying:** Run queries to analyze data
  - Recommended Services: Azure Synapse Analytics, Hive (SQL)
- **Data Visualization:** Create dashboards, charts and reports
  - Recommended Services: Power BI



# Data Analytics - Scenarios

| Scenario   | Solution               |
|--|------------------------|
| Decide Data Analysis Category: You have generated a report showing current status vs planned | Descriptive analytics  |
| Decide Data Analysis Category: Why did sales increase last month?                            | Diagnostic analytics   |
| Decide Data Analysis Category: Semantic Analysis of Text (Analyze reviews)                   | Cognitive analytics    |
| You want to move your Hadoop workloads to the cloud  | Azure HDInsight        |
| Stream or Batch: Processing after you received 10K records                                   | Batch                  |
| You want to move SQL Server Integration Services (SSIS) packages to cloud                    | Use Azure Data Factory |
| Categorize activity: Copy data from one store to another store                               | Data movement          |

# Data Analytics - Scenarios - 2

| Scenario  | Solution                |
|---|-------------------------|
| Categorize activity: Orchestrate pipeline activity based on output of another pipeline activity | Control flow            |
| You want to connect to an external source from Data Factory                                     | Linked Service          |
| Compute infrastructure used by Azure Data Factory   | Integration Runtime     |
| Standalone tool to author paginated reports   | Power BI Report Builder |
| What is Online SaaS Power BI called?  | Power BI Service        |
| Which of these represents "One or more pages of visualizations in Power BI"?                    | Report                  |
| Which of these represents "Single page visualizations from one or more reports"?                | Dashboard               |

# Other Important Azure Concepts

| Tool                                | Description   |
|-------------------------------------|---|
| Azure Data Studio                   | Cross-platform (Windows, Mac, linux) db tool with Intellisense, code snippets and source control<br>Run SQL queries. Save results in different formats - text, JSON, Excel<br>Supports SQL Server, Azure SQL Database, Azure Synapse Analytics..<br>Notebooks: Create and share documents with text, images and SQL query results<br>Support to create and restore backup from SQL Database |
| SQL Server Management Studio (SSMS) | Graphical tool for managing SQL Server and Azure Databases<br>Query, design, and manage your databases and data warehouses<br>Supports configuration, management and administration tasks<br>Suitable for SQL Server, SQL Database, Azure Synapse Analytics   |
| SQL Server Data Tools (SSDT)        | Build SQL Server and Azure SQL relational databases, Analysis Services (AS) data models, Integration Services (IS) packages, and Reporting Services (RS) reports  |
| sqlcmd                              | Run SQL scripts and procedures from command line<br>Supports SQL Server, Azure SQL Database, Azure SQL MI, Azure Synapse Analytics  |

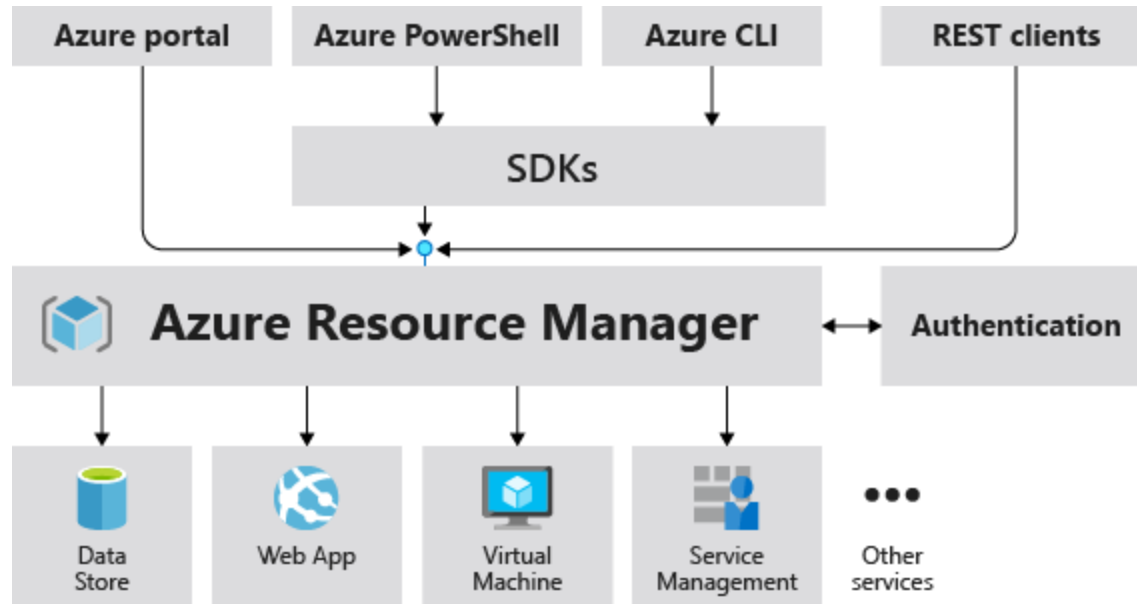
| Role                   | Description  |
|------------------------|--|
| Database Administrator | <b>Role:</b> Install, upgrade, control (authorization, availability, durability, performance optimization, backups, disaster recovery, compliance with licensing) of data servers<br><b>Tools:</b> Azure Data Studio, SQL Server Management Studio (SSMS)  |
| Data Engineers         | <b>Responsible for</b> data architecture, data acquisition, data ingestion, data processing (transformation, cleansing and pipelines) and data storage (design, build and test) for analytical workloads<br>Responsible for build, test, monitoring, performance optimization of data pipelines<br>Responsible for improving data reliability, efficiency, and quality<br><b>Tools:</b> Azure Data Studio, Azure HDInsight, Azure Databricks, Azure Data Factory, Azure Cosmos DB, Azure Storage ...<br>Programming Languages - HiveQL, R, or Python |
| Data Analyst           | <b>Responsible for</b> getting intelligence from data through integration of data(from multiple sources), dashboards, reports, visualizations (charts, graphs, ..) and pattern identification (from huge volumes of data)<br><b>Tools:</b> Microsoft Excel, Power BI...  |

# Pricing calculator

- Estimate the costs for Azure services
- Example Services that you can estimate costs for:
  - Virtual Machines
  - Storage Accounts
  - Azure SQL Database
  - Azure Cosmos DB
  - ...
- Ideal place to explore and learn important factors about different Azure services



# Azure Resource Manager



(<https://docs.microsoft.com/>)

- Deployment and management service for Azure
- All actions to any resource in Azure **go through ARM**
  - Irrespective of where you are performing it from
    - Azure portal OR Powershell OR CLI or ARM template or ...



# Azure Resource Manager (ARM) templates



- **Lets consider an example:**
  - I would want to create an Azure SQL Database
  - I would want to create an Azure Data Lake Storage Gen2
  - I would want to create an Azure Data Factory Workspace
- **AND I would want to create 4 environments**
  - Dev, QA, Stage and Production!
- **Azure Resource Manager (ARM) templates** can help you do all these with a simple (actually NOT so simple) script!

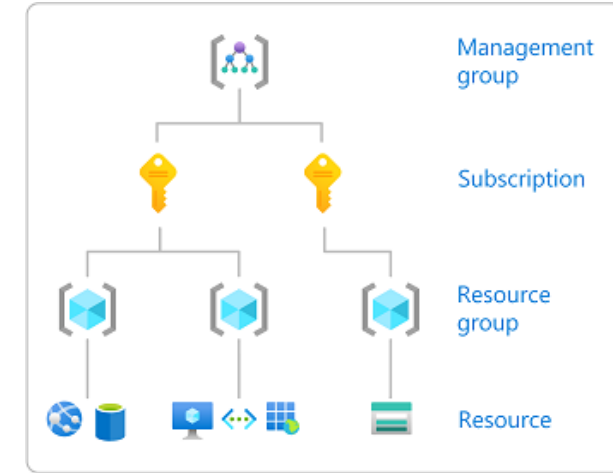
# Azure Portal, PowerShell, CLI, Cloud Shell

| Tool              | Details  |
|-------------------|--|
| Azure Portal      | Web-based user interface. Great to get started BUT NO automation possible.<br>Runs in all modern desktop and tablet browsers   |
| Azure PowerShell  | Execute cmdlets (sequence of commands) and create scripts (PowerShell script)<br>Recommended for teams familiar with Windows administration<br>Cross-platform (Windows, Linux, and macOS)  |
| Azure CLI         | Similar to Azure PowerShell BUT uses a different syntax (Bash Scripts)<br>Recommended for teams familiar with Linux administration (and Bash Scripts)<br>Cross-platform (Windows, Linux, and macOS)                                      |
| Azure Cloud Shell | Free Browser based interactive shell (Access from Azure Portal)<br>Common Azure tools pre-installed and configured to use with your account<br>Supports both PowerShell and CLI (bash)<br>Runs in all modern desktop and tablet browsers |

# Azure Resource Hierarchy

In 28  
Minutes

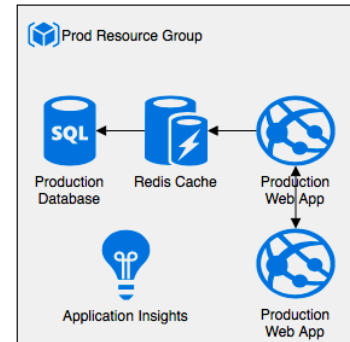
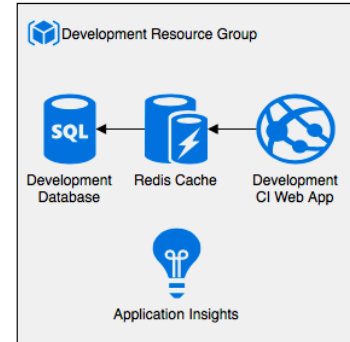
- **Hierarchy:** Management Group(s) > Subscription (s) > Resource Group (s) > Resources
  - **Resources:** VMs, Storage, Databases
  - **Resource groups:** Organize resources by grouping them into Resource groups
  - **Subscriptions:** Manage costs for resources provisioned for different teams or different projects or different business units
  - **Management groups:** Centralized management for access, policy, and compliance across multiple subscriptions
- **Remember:**
  - No hierarchy in resource groups BUT management groups can have a hierarchy



(<https://docs.microsoft.com/>)

# Resource Groups

- **Resource Group:** Logical container for resources
  - Associated with a single subscription
  - Can have multiple resources
    - (REMEMBER) A resource can be associated with one and only one resource group
  - Can have resources from multiple regions
  - Deleting it deletes all resources under it
- Tags assigned to resource group are not automatically applied to resources
  - HOWEVER, Permissions/Roles assigned to user at the resource group level are inherited by all resources in the group
- Resource Groups (like Management Groups) are free



# Subscriptions

In 28  
Minutes

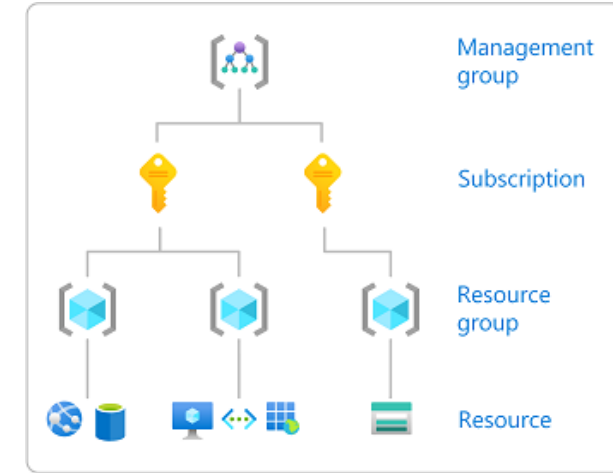
- You need a Subscription to create resources in Azure
  - Subscription links Azure Account to its resources
- An Azure Account can have multiple subscriptions and multiple account administrators
- **When do you create a new subscription?**
  - I want to manage different access-management policies for different environments:
    - Create different subscriptions for different environments
    - Manage distinct Azure subscription policies for each environment
  - I want to manage costs across different departments of an organization:
    - Create different subscriptions for different departments
    - Create separate billing reports and invoices for each subscription (or department) and manage costs
  - I'm exceeding the limits available per subscription
    - Example: VMs per subscription - 25,000 per region



Subscriptions

# Management Groups

- Allows you to manage access, policies, and compliance across multiple subscriptions
  - Group subscriptions into **Management Groups**
  - All subscriptions & resources under a Management Group inherit all constraints applied to it
- (REMEMBER) You can create a hierarchy of management groups
- (REMEMBER) All subscriptions in a management group should be associated with the same Azure AD tenant



(<https://docs.microsoft.com/>)

# Quick Review

# Azure Storage - Quick Review

| Service             | Description  |
|---------------------|--|
| Azure Disk storage  | Store disks attached to VMs.   |
| Azure Blob storage  | Store unstructured data - video files, database archives etc.                                |
| Azure File storage  | Create file shares or file servers in the cloud  |
| Azure Queue storage | Decouple applications using a queue (asynchronous communication)                             |
| Azure Table storage | Store structure data using NoSQL approach (NON-relational). Schemaless. Key/attribute store. |



# Azure Databases - Quick Review

| Service                              | Description   |
|--------------------------------------|---|
| SQL Server on Azure Virtual Machines | Provides full administrative control over the SQL Server instance and underlying OS for migration to Azure  |
| Azure SQL Database                   | Fully Managed Service for Microsoft SQL Server.<br>Recommended for cloud-born applications  |
| Azure SQL managed instance           | Full (Near 100%) SQL Server access and feature compatibility<br>Recommended for migrating on-premise SQL Server databases<br>Azure SQL managed instance ONLY features: Cross-database queries, Database Mail Support, SQL Server Agent etc. |
| Azure Database for MySQL             | Fully managed MySQL database  |
| Azure Database for PostgreSQL        | Fully managed PostgreSQL database   |

# Azure Databases - Quick Review - 2

| Service                          | Description  |
|----------------------------------|--|
| Azure Cosmos DB                  | NoSQL database. Globally distributed.<br>Core(SQL), MongoDB, Table, Gremlin and Cassandra APIs |
| Azure Cache for Redis            | Managed service for Redis (high-throughput, low-latency data caching)                          |
| Azure Database Migration Service | Migrate databases to the cloud   |

# Azure Analytics Services - Quick Review

| Service                 | Description   |
|-------------------------|---|
| Azure Data Lake Storage | Data lake built on Azure Blob Storage   |
| Azure Databricks        | Managed Apache Spark  |
| Azure HDInsight         | Managed Apache Hadoop   |
| Azure Synapse Analytics | End-to-end analytics solutions<br>Data integration + Enterprise data warehouse + Data analytics |
| Azure Data Factory      | Data Integration. Fully managed serverless service to build complex data pipelines.             |
| Power BI                | Create visualization around data - reports & dashboards   |
| Event Hubs              | Receive telemetry from millions of devices  |

# Get Ready

# Certification Exam

In **28**  
Minutes



- Certification Home Page
  - <https://docs.microsoft.com/en-gb/learn/certifications/exams/dp-900>
- Different Types of Multiple Choice Questions
  - **Type 1** : Single Answer - 2/3/4 options and 1 right answer
  - **Type 2** : Multiple Answer - 5 options and 2 right answers
- **No penalty** for wrong answers
  - Feel free to guess if you do not know the answer
- **40-60** questions and **65** minutes
- Result immediately shown after exam completion
- Email with detailed scores (a couple of days later)

# Certification Exam - My Recommendations

In **28**  
Minutes

- Read the **entire question**
  - Identify the key parts of the question
- Read all answers at least once
- If you do NOT know the answer, **eliminate** wrong answers first
- **Mark questions for future consideration** and review them before final submission



**You are all set!**

# Let's clap for you!

- You have a lot of patience! Congratulations
- You have put your **best foot forward** to get Microsoft Certification - DP-900: Microsoft Azure Data Fundamentals
- Make sure you prepare well and
- **Good Luck!**





# Do Not Forget!

- **Recommend the course to your friends!**
  - Do not forget to review!
- **Your Success = My Success**
  - Share your success story with me on LinkedIn (Ranga Karanam)
  - Share your success story and lessons learnt in Q&A with other learners!

In **28**  
Minutes



Google Cloud



# What next?

[\*https://github.com/in28minutes/learn\*](https://github.com/in28minutes/learn)

- Learn Other Cloud Platforms:
  - Gartner predicts a multi cloud world soon
  - Get certified on AWS, Azure and Google Cloud
- Learn DevOps (Containers and Container Orchestration)
- Learn Full Stack Development

