**AZURE Data Fundamentals Notes**

Over the last few decades, the amount of data that systems, applications, and devices have generated has increased significantly. Data is everywhere. Data is available in different structures and formats. Understanding data and exploring it reveals interesting facts, and helps you gain meaningful insights:

1. **What is data**:Data is a collection of facts such as numbers, descriptions, and observations used in decision making. You can classify data as

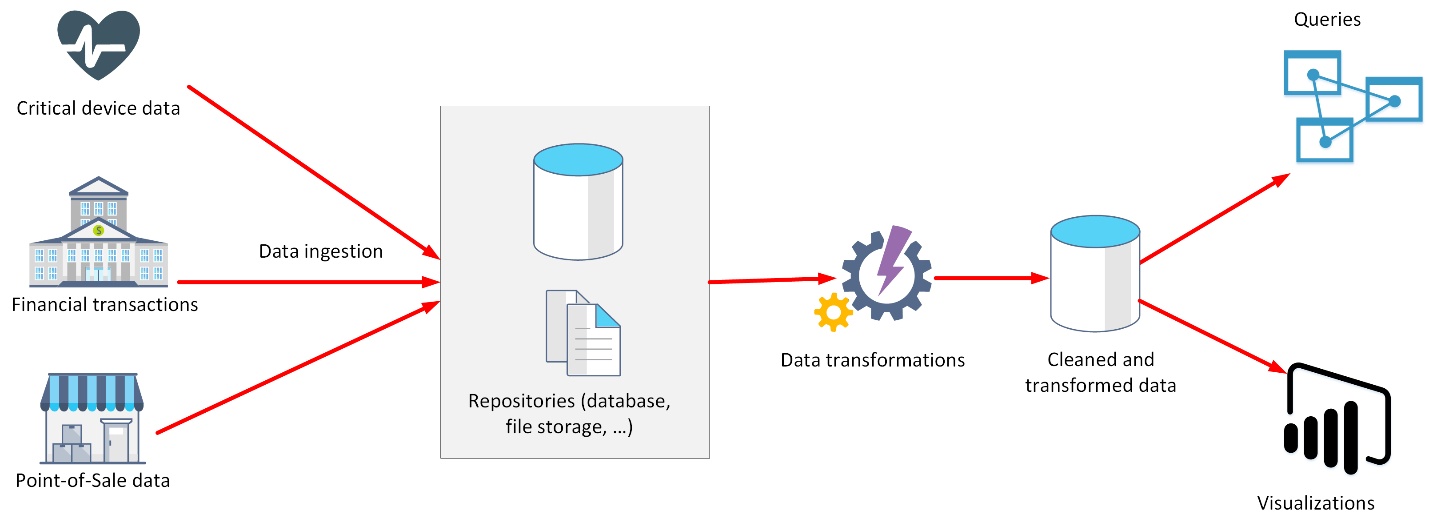
* Structured data, (e.g. relational databases with rows and columns in tables)
* Semi-structured data is information that doesn't reside in a relational database but still has some structure to it e.g. Azure Cosmos DB, Document Databases (JSON, Avro, ORC {Optimized Row Columnar format}, Parquet), key-value stores, graph databases, column family databases
* Unstructured: Videos, Audios, Images.

1. **Data processing solutions**: Data processing solutions often fall into one of two broad categories:

* **Transaction processing systems**: A transactional system is often what most people consider the primary function of business computing. A transactional system records transactions. A transaction could be financial, such as the movement of money between accounts in a banking system, or it might be part of a retail system, tracking payments for goods and services from customers. The work performed by transactional systems is often referred to as Online Transactional Processing (OLTP)

A transactional database must adhere to the ACID (Atomicity, Consistency, Isolation, Durability) properties to ensure that the database remains consistent while processing transactions

* **Analytical systems**: In contrast to systems designed to support OLTP, an analytical system is designed to support business users who need to query data and gain a big picture view of the information held in a database. Most analytical data processing systems need to perform similar tasks: **data ingestion**, **data transformation**, **data querying**, and **data visualization**.



Analytical workloads are typically read-only systems that store vast volumes of historical data or business metrics, such as sales performance and inventory levels. Analytical workloads are used for data analysis and decision making

1. **Types of data and data storage**:

* **Relational databases**:In a relational database, You can define relationships between tables. You model collections of entities from the real world as tables. The main characteristics of a relational database are:
  + All data is tabular. Entities are modeled as tables, each instance of an entity is a row in the table, and each property is defined as a column.
  + All rows in the same table have the same set of columns.
  + A table can contain any number of rows.
  + A primary key uniquely identifies each row in a table. No two rows can share the same primary key.
  + A foreign key references rows in another, related table. For each value in the foreign key column, there should be a row with the same value in the corresponding primary key column in the other table.

Apart from tables, a typical relational database contains other structures that help to optimize data organization, and improve the speed of access:

* + **Index**: An index helps you search for data in a table. When you create an index in a database, you specify a column from the table, and the index contains a copy of this data in a sorted order, with pointers to the corresponding rows in the table. You can create many indexes on a table.However, indexes aren't free. An index might consume additional storage space, and each time you insert, update, or delete data in a table, the indexes for that table must be maintained (result into slower performance)
  + **View**: A view is a virtual table based on the result set of a query. In the simplest case, you can think of a view as a window on specified rows in an underlying table
  + A **stored procedure** is a block of code that runs inside your database. Applications often use stored procedures because they are optimized to run in the database environment, and can access data very quickly.
  + A **linked server** is a connection from one database server to another. SQL Server can use linked servers to run queries on one server that can include data retrieved from other servers; these are known as **distributed queries**.
* **Non-Relational databases**: It enable you to store data in a format that more closely matches the original structure. For example, in a document database.
  + Non-relational databases are either schema free or have relaxed schemas
  + Two entities in the same collection can have a different set of fields rather than a regular set of columns found in a relational table. The lack of a fixed schema means that each entity must be self-describing. Often this is achieved by labeling each field with the name of the data that it represents.
  + Each entity should have a unique key value. The entities in a collection are usually stored in key-value order. Examples: Azure Cosmos DB, Document Databases (JSON, Avro, ORC {Optimized Row Columnar format}, Parquet), key-value stores, graph databases, column family databases
  + Non-relational databases are highly suitable for the following scenarios (IoT and telematics, Retail and marketing, Gaming, Web and mobile applications)
  + When data needs to be ingested very quickly, or the query is unknown and unconstrained, a relational database can be less suitable than a non-relational database.
  + Non-relational data generally falls into two categories; semi-structured and non-structured
    1. **Semi-Structured**: Examples: Azure Cosmos DB, Document Databases (JSON, Avro, ORC{Optimized Row Columnar format}, Parquet), key-value stores, graph databases, column family databases.

A JSON document is enclosed in curly brackets ({ and }). Each field has a name (a label), followed by a colon, and then the value of the field. Fields can contain simple values, or subdocuments (each starting and ending with curly brackets). Fields can also have multiple values, held as arrays and surrounded with square brackets ([ and ]). Literals, or fixed values, in a field are enclosed in quotes, and fields are separated with commas.

You're free to define whatever fields you like. The important point is that the data follows the JSON grammar.

When an application reads a document, it can use a JSON parser to break up the document into its component fields and extract the individual pieces of data.

* + 1. **Un-Structured**: Unstructured data is data that doesn't naturally contain fields. (Examples: video, audio, and other media streams). Each item is an amorphous blob of binary data.

In Azure, you would probably store video and audio data as block blobs in an Azure Storage account

You could also consider files as a form of unstructured data, although in some cases a file might include metadata that indicates what type of file it is (photograph, Word document, Excel spreadsheet, and so on), owner, and other elements that could be stored as fields. However, the main content of the file is unstructured

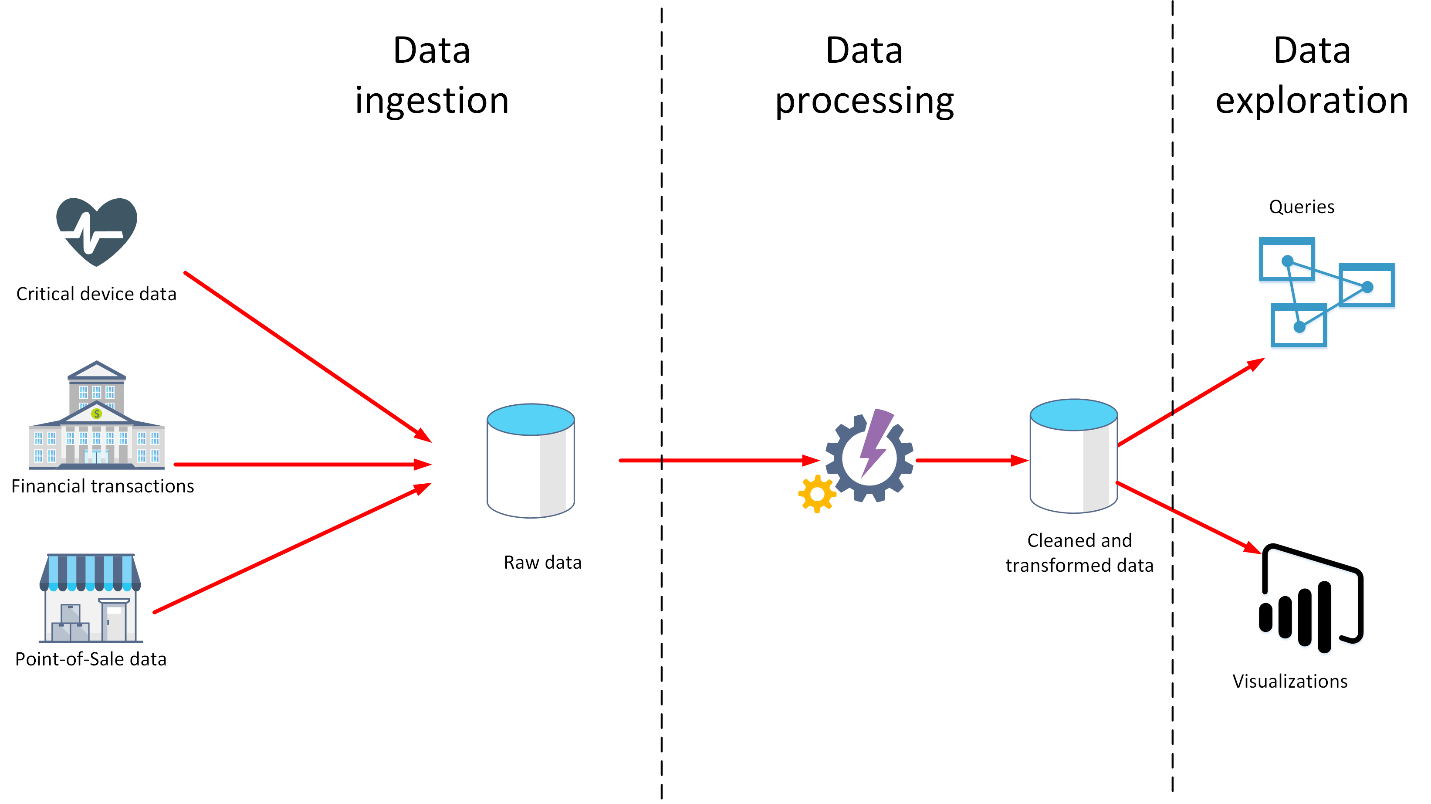
* + **NoSQL** is a loose term that simply means non-relational.
  + Some non-relational databases support a version of SQL adapted for documents rather than tables (examples include Azure Cosmos DB)
  + NoSQL (non-relational) databases generally fall into four categories: key-value stores, document databases, column family databases, and graph databases
    1. **key-value store**: It s the simplest (and often quickest) type of NoSQL database for inserting and querying data.
       - The value is opaque to the database management system. Items are stored in key order.The term opaque means that the database management system just sees the value as an unstructured block. Only the application understands how the data in the value is structured and what fields it contains. The opposite of opaque is transparent. If the data is transparent, the database management system understands how the fields in the data are organized. A relational table is an example of a transparent structure.
       - You can't search on values.
       - An application that retrieves data from a key-value store is responsible for parsing the contents of the values returned.
       - Write operations are restricted to inserts and deletes. If you need to update an item, you must retrieve the item, modify it in memory (in the application), and then write it back to the database, overwriting the original (effectively a delete and an insert).
       - The focus of a key-value store is the ability to read and write data very quickly. Search capabilities are secondary
       - A key-value store is an excellent choice for data ingestion, when a large volume of data arrives as a continual stream and must be stored immediately.
       - **Azure Table storage** is an example of a key-value store. **Cosmos DB also implements a key-value store using the Table API**
    2. **Document Database:** In a document database, each document has a unique ID, but the fields in the documents are transparent to the database management system.
       - Documents could be encoded using other formats such as XML, YAML, JSON, BSON
       - A single document may contain information that would be spread across several relational tables in an RDBMS
       - The fields in documents are exposed to the storage management system, enabling an application to query and filter data by using the values in these fields
       - A document store does not require that all documents have the same structure
       - The focus of a document database is its query capabilities
       - Azure **Cosmos DB implements a document database** approach in its Core (SQL) API
    3. **Column family database**:A column family database organizes data into rows and columns. Examples of this structure include ORC and Parquet files, described in the previous unit.
       - You can think of a column family database as holding tabular data comprising rows and columns, but you can divide the columns into groups known as column-families.
       - Each column family holds a set of columns that are logically related together
       - Many ways of organizing the columns are possible, but you should implement your column-families to optimize the most common queries that your application performs
       - The most widely used column family database management system is **Apache Cassandra**. Azure Cosmos DB supports the column-family approach through the **Cassandra API**
    4. **Graph databases:**It enables you to store entities, but the main focus is on the **relationships** that these entities have with each other.
       - A graph database stores two types of information: nodes that you can think of as instances of entities, and edges, which specify the relationships between nodes
       - Additionally, edges can have a direction indicating the nature of the relationship.
       - The purpose of a graph database is to enable an application to efficiently perform queries that traverse the network of nodes and edges, and to analyze the relationships between entities.
       - For large graphs with lots of entities and relationships, you can perform very complex analyses very quickly, and many graph databases provide a query language that you can use to traverse a network of relationships efficiently.
       - You can often store the same information in a relational database, but the SQL required to query this information might require many expensive recursive join operations and nested subqueries.
       - **Azure Cosmos DB supports graph databases using the Gremlin API**. The Gremlin API is a standard language for creating and querying graphs.

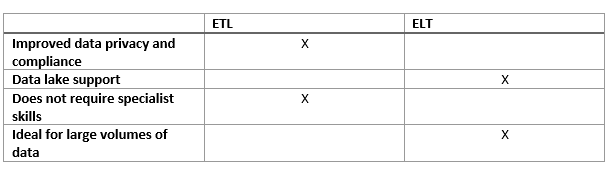
1. **Batch and streaming data**: Data processing is simply the conversion of raw data to meaningful information through a process. Depending on how the data is ingested into your system, you could process each data item as it arrives, or buffer the raw data and process it in groups. Processing data as it arrives is called streaming. Buffering and processing the data in groups is called batch processing
2. **Roles &Responsibilities** in the world of data:
3. **Database Administrators** manage databases, assigning permissions to users, storing backup copies of data and restore data in case of any failures.An Azure database administrator is responsible for the design, implementation, maintenance, and operational aspects of on-premises and cloud-based database solutions built on Azure data services and SQL Server. They're responsible for the overall availability and consistent performance and optimizations of the database solutions
4. **Data Engineers** are vital in working with data, applying data cleaning routines, identifying business rules, and turning data into useful information.A data engineer collaborates with stakeholders to design and implement data-related assets that include data ingestion pipelines, cleansing and transformation activities, and data stores for analytical workloads. They use a wide range of data platform technologies, including relational and nonrelational databases, file stores, and data streams.
5. **Data Analysts** explore and analyze data to create visualizations and charts to enable organizations to make informed decisions. A data analyst enables businesses to maximize the value of their data assets. They're responsible for designing and building scalable models, cleaning and transforming data, and enabling advanced analytics capabilities through reports and visualizations.

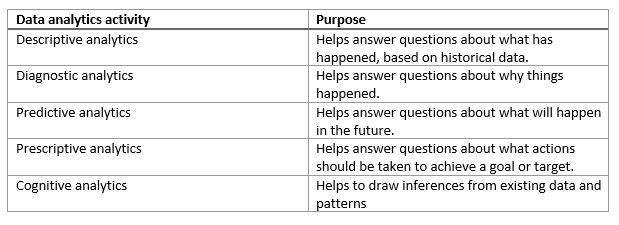
A data analyst processes raw data into relevant insights based on identified business requirements to deliver relevant insights

1. **Data Analytics**:Successful companies make informed decisions to find new opportunities, identify weaknesses, increase efficiency, and improve customer satisfaction. Data analytics is the process of examining raw data to uncover trends, and discover information used to ask and answer questions related to organizational performance.

* A data analyst might take data such as customer spend and look for correlations with other factors such as the weather, regional events, or even the presence (or absence) of incentives such as food and drink.
* Data analytics allows for a thorough look at the data and can lead to a faster diagnosis or treatment plan.



* Data analytics is concerned with taking data and finding meaningful information and inferences from it
* The data a company uses can come from many sources. There could be a mass of historical data to comb through, and fresh data continuing to arrive all the time. This data could be the result of customer purchases, bank transactions, stock price movements, real-time weather data, monitoring devices, or even cameras. In a data analytics solution, you combine this data and construct a data warehouse that you can use to ask (and answer) questions about your business operations
* **Data Ingestion**: Data ingestion is the process of obtaining and importing data for immediate use or storage in a database.
  + The data can arrive as a continuous stream, or it may come in batches, depending on the source.
  + The purpose of the ingestion process is to capture this data and store it.
  + The ingestion process might also perform filtering. For example, ingestion might reject suspicious, corrupt, or duplicated data. Suspicious data might be data arriving from an unexpected source. Corrupt or duplicated data could be due to a device error, transmission failure, or tampering
* **Data Processing or Data Transformation**: The data processing stage occurs after the data has been ingested and collected. Data processing takes the data in its raw form, cleans it, and converts it into a more meaningful format (tables, graphs, documents, and so on). The result is a database of data that you can use to perform queries and generate visualizations
  + **Data cleaning** is a generalized term that encompasses a range of actions, such as removing anomalies, and applying filters and transformations that would be too time-consuming to run during the ingestion stage.
  + **Wrangling** is the process by which you transform and map raw data into a more useful format for analysis. It can involve writing code to capture, filter, clean, combine, and aggregate data from many sources
  + The data processing stage could also generate predictive or other analytical models from the data
  + Data processing can be complex, and may involve automated scripts, and tools such as Azure Databricks, Azure Functions, and Azure Cognitive Services to examine and reformat the data, and generate models
  + A data analyst could use machine learning to help determine future trends based on these models.
  + The data processing mechanism can take two approaches: ELT and ETL.
  + **ETL** stands for Extract, Transform, and Load.
    - It is suitable for systems that only require simple models, with little dependency between items.
    - This type of process is often used for basic data cleaning tasks, deduplicating data, and reformatting the contents of individual fields.
    - ETL can filter data before it's stored. In this way, ETL can help with data privacy and compliance, removing sensitive data before it arrives in your analytical data models.
  + **ELT** is an abbreviation of Extract, Load, and Transform.
    - The data processing engine can take an iterative approach, retrieving and processing the data from storage, before writing the transformed data and models back to storage.
    - ELT is more suitable for constructing complex models that depend on multiple items in the database, often using periodic batch processing
    - ELT is a scalable approach that is suitable for the cloud because it can make use of the extensive processing power available
  + Another more generalized approach is to use Azure Data Factory.
    - Azure Data Factory is a cloud-based data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale.
    - Using Azure Data Factory, you can create and schedule data-driven workflows (called pipelines) that can ingest data from disparate data stores
    - You can build complex ETL processes that transform data visually with data flows, or by using compute services such as Azure HDInsight Hadoop, Azure Databricks, and Azure SQL Database
* **Data Exploration**: A business model can contain an enormous amount of information. The purpose of producing a model such as this is to help you reason over the information it contains, ask questions, and hopefully obtain answers that can help you drive your business forward.
  + **Reporting** is the process of organizing data into informational summaries to monitor how different areas of an organization are performing. Reporting shows you what has happened, while analysis focuses on explaining why it happened and what you can do about it.
  + **Business intelligence**: BI refers to technologies, applications, and practices for the collection, integration, analysis, and presentation of business information. The purpose of business intelligence is to support better decision making. Information is often gathered about other companies in the same industry for comparison. This process of comparison with other companies in the same industry is known as **benchmarking.**
  + **Data visualization**: Data visualization is the graphical representation of information and data. E.g. Power BI. Using Power BI, you can connect to multiple different sources of data, and combine them into a data model. This data model lets you build visuals, and collections of visuals you can share as reports, with other people inside your organization.
    - Data visualization helps you to focus on the meaning of data, rather than looking at the data itself.
    - A good data visualization enables you to quickly spot trends, anomalies, and potential issues.
    - The most common forms of visualizations are:
      * **Bar and column charts**: Bar and column charts enable you to see how a set of variables changes across different categories
      * **Line charts**: Line charts emphasize the overall shape of an entire series of values, usually over time.
      * **Matrix**: A matrix visual is a tabular structure that summarizes data
      * **Key influencers**: A key influencer chart displays the major contributors to a selected result or value. Key influencers are a great choice to help you understand the factors that influence a key metric
      * **Treemap**: Treemaps are charts of colored rectangles, with size representing the relative value of each item. They can be hierarchical, with rectangles nested within the main rectangles
      * **Scatter**: A scatter chart shows the relationship between two numerical values.
      * **Bubblechart** is a scatter chart that replaces data points with bubbles, with the bubble size representing an additional third data dimension
      * **Dot plotchart** is similar to a bubble chart and scatter chart, but can plot categorical data along the X-Axis
      * **Filled map.** If you have geographical data, you can use a filled map to display how a value differs in proportion across a geography or region. You can see relative differences with shading that ranges from light (less-frequent/lower) to dark (more-frequent/more)
* **Data analytics**:Data analytics is concerned with examining, transforming, and arranging data so that you can study it and extract useful information. Data analytics is a discipline that covers the entire range of data management tasks. These tasks **not only include analysis**, but also data collection, organization, storage, and all the tools and techniques used.
  + The term data analytics is a catch-all that covers a range of activities, each with its own focus and goals. You can categorize these activities as **descriptive, diagnostic, predictive, prescriptive, and cognitive analytics**.



* + **Descriptive analytics**:Descriptive analytics helps answer questions about what has happened, based on historical data. Descriptive analytics techniques summarize large datasets to describe outcomes to stakeholders.

By developing KPIs (Key Performance Indicators), these strategies can help track the success or failure of key objectives. Metrics such as return on investment (ROI) are used in many industries. Specialized metrics are developed to track performance in specific industries.

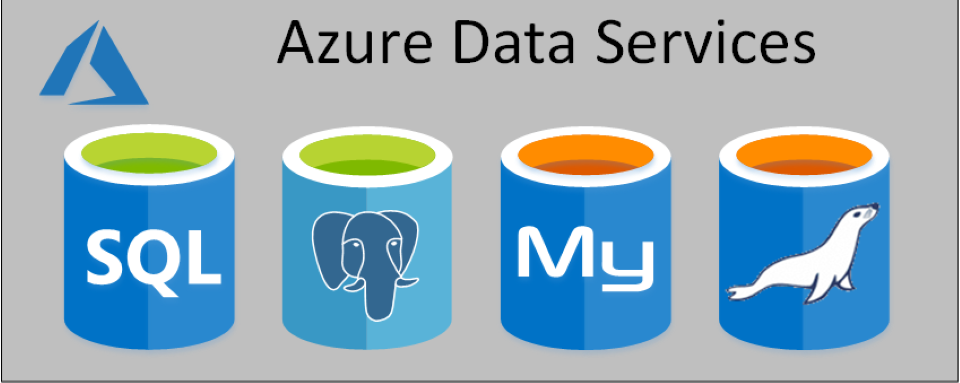
Examples of descriptive analytics include generating reports to provide a view of an organization's sales and financial data

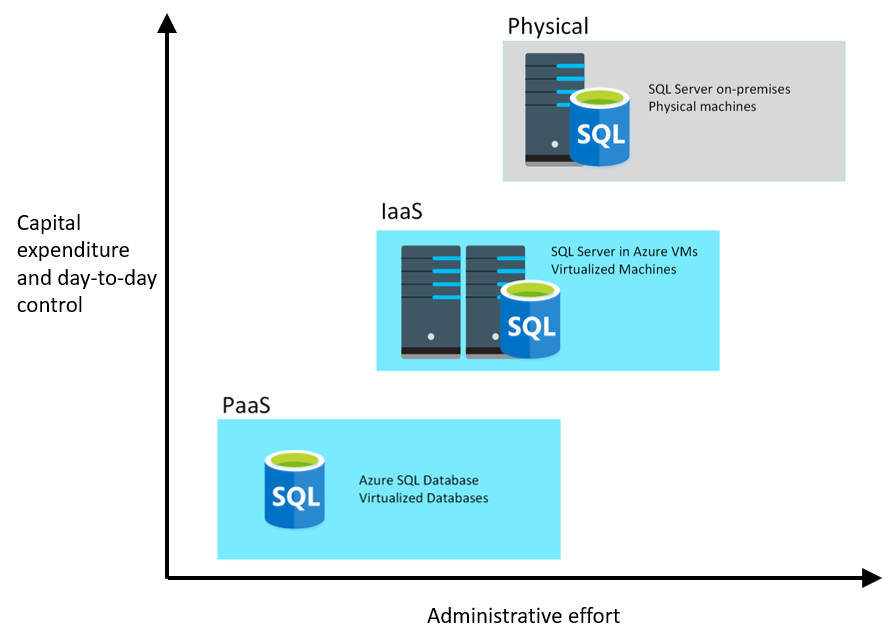
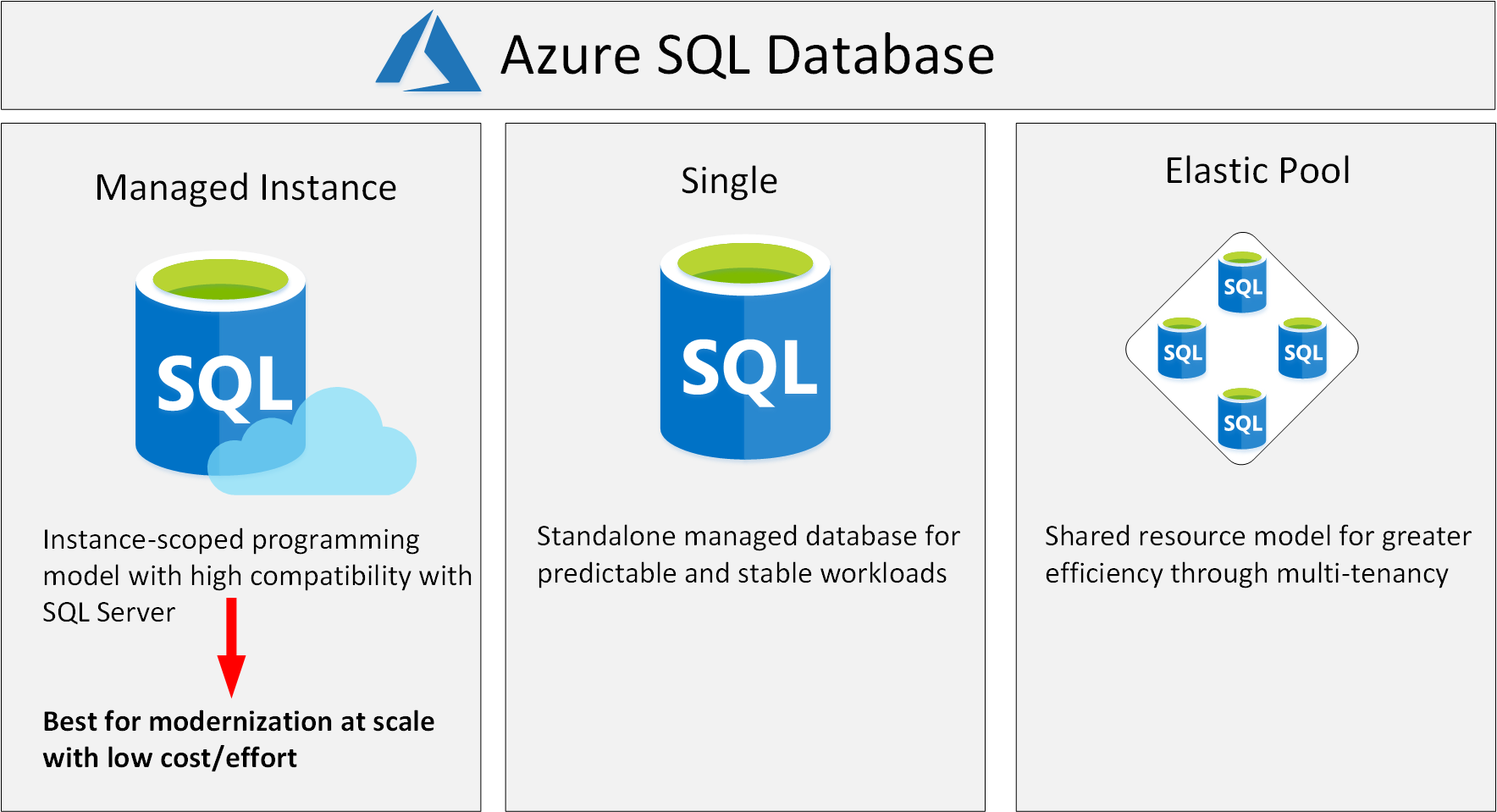
* + **Diagnostic analytics**:Diagnostic analytics helps answer questions about why things happened. Diagnostic analytics techniques supplement more basic descriptive analytics. They take the findings from descriptive analytics and dig deeper to find the cause. This generally occurs in three steps:
    - Identify anomalies in the data. These may be unexpected changes in a metric or a particular market.
    - Collect data that's related to these anomalies.
    - Use statistical techniques to discover relationships and trends that explain these anomalies.
  + **Predictive analytics**:Predictive analytics helps answer questions about what will happen in the future. Predictive analytics techniques use historical data to identify trends and determine if they're likely to recur. Techniques include a variety of statistical and machine learning techniques such as neural networks, decision trees, and regression.
  + **Prescriptive analytics**:Prescriptive analytics helps answer questions about what actions should be taken to achieve a goal or target. By using insights from predictive analytics, data-driven decisions can be made. This technique allows businesses to make informed decisions in the face of uncertainty. Prescriptive analytics techniques rely on machine learning strategies to find patterns in large datasets. By analyzing past decisions and events, the likelihood of different outcomes can be estimated
  + **Cognitive analytics**:Cognitive analytics attempts to draw inferences from existing data and patterns, derive conclusions based on existing knowledge bases, and then add these findings back into the knowledge base for future inferences--a self-learning feedback loop. Cognitive analytics helps you to learn what might happen if circumstances change, and how you might handle these situations.

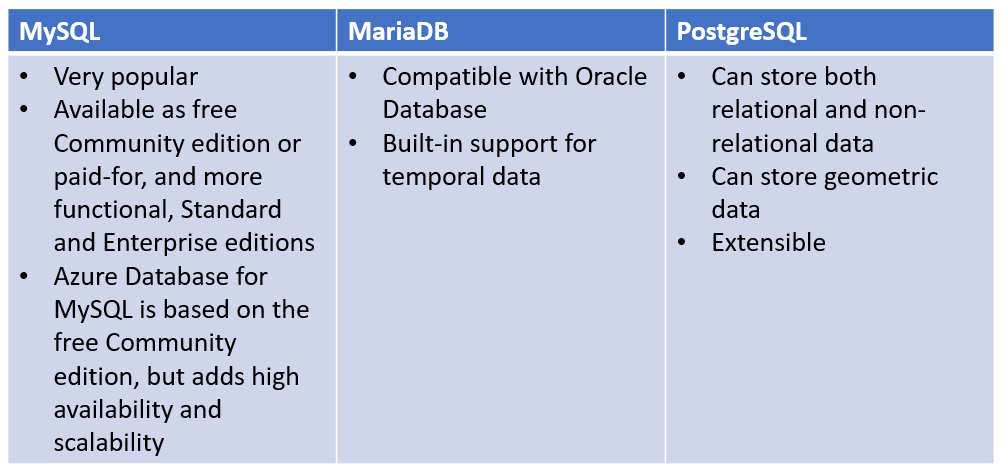
Effective cognitive analytics depends on machine learning algorithms. It uses several NLP (Natural Language Processing) concepts to make sense of previously untapped data sources, such as call center conversation logs and product reviews

Theoretically, by tapping the benefits of massive parallel/distributed computing and the falling costs of data storage and computing power, there's no limit to the cognitive development that these systems can achieve.

1. **Relational Data in Azure**:
   1. **Azure Data Services**: fall into the PaaS category. These **are Azure SQL Database**, **Azure Database for MySQL servers**, **Azure Database for PostgreSQL servers** and **Azure Database for MariaDB servers.**

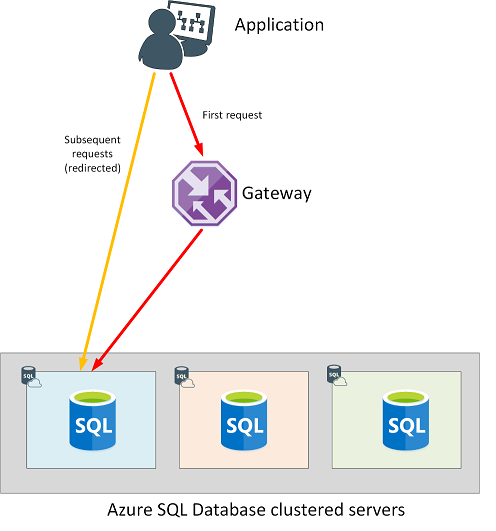


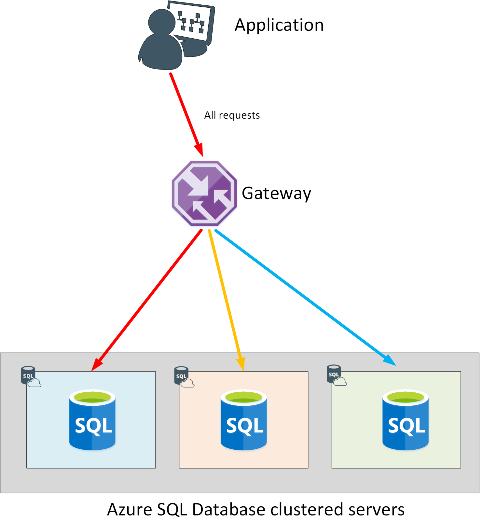
* + The base price of each service covers underlying infrastructure and licensing, together with the administration charges. Additionally, these services are designed to be always on. This means that you can't shut down a database and restart it later.
  + Not all features of a database management system are available in Azure Data Services. Exposing some administrative functions might make the underlying platform vulnerable to misuse, and even open up some security concerns. Therefore, you have no direct control over the platform on which the services run
  + **Azure SQL Database**: Azure SQL Database is a PaaS offering from Microsoft. You create a managed database server in the cloud, and then deploy your databases on this server.
    - A SQL Database server is a logical construct that acts as a central administrative point for multiple single or pooled databases, logins, firewall rules, auditing rules, threat detection policies, and failover groups
    - It is often used in new cloud projects where the application design can accommodate any required changes to your applications
    - Azure SQL Database is available with several options. **Single Database, Elastic Pool, and Managed Instance**.
      * **Single Database**: This option enables you to quickly set up and run a single SQL Server database. You create and run a database server in the cloud, and you access your database through this server. Microsoft manages the server, so all you have to do is configure the database, create your tables, and populate them with your data. You can also specify a serverless configuration. In this configuration, Microsoft creates its own server, which might be shared by a number of databases belonging to other Azure subscribers. Microsoft ensures the privacy of your database. Your database automatically scales and resources are allocated or deallocated as required
      * **Elastic Pool**:This option is similar to Single Database, except that by default multiple databases can share the same resources, such as memory, data storage space, and processing power through multiple-tenancy. The resources are referred to as a pool. You create the pool, and only your databases can use the pool
      * **Azure SQL Database managed instance**:The Single Database and Elastic Pool options restrict some of the administrative features available to SQL Server. Managed instance effectively runs a fully controllable instance of SQL Server in the cloud. You can install multiple databases on the same instance. You have complete control over this instance, much as you would for an on-premises server
        + The Managed instance service automates backups, software patching, database monitoring, and other general tasks, but you have full control over security and resource allocation for your databases
        + Managed instances depend on other Azure services such as Azure Storage for backups, Azure Event Hubs for telemetry, Azure Active Directory for authentication, Azure Key Vault for Transparent Data Encryption (TDE) and a couple of Azure platform services that provide security and supportability features. The managed instances make connections to these services
        + All communications are encrypted and signed using certificates.
        + 
    - **Business benefits**:
      * Azure SQL Database automatically updates and patches the SQL Server software to ensure that you are always running the latest and most secure version of the service
      * Scalability
      * High availability, point-in-time restore, replicated to different regions to provide disaster recovery
      * Advanced threat protection: It continuously monitors your database for suspicious activities, and provides immediate security alerts on potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
      * Auditing: tracks database events and writes them to an audit log in your Azure storage account
      * Encryption: For data in motion, it uses Transport Layer Security. For data at rest, it uses Transparent Data Encryption. For data in use, it uses Always Encrypted
      * linked servers are used to perform distributed queries. However, **neither Single Database nor Elastic Pool** support linked servers
      * **Additional Benefits for Managed Instance**:
        + SQL Database service either performs administrative tasksfor you or greatly simplifies those tasks. Automated tasks include operating system and database management system software installation and patching, dynamic instance resizing and configuration, backups, database replication (including system databases), high availability configuration, and configuration of health and performance monitoring data streams.
        + 100% compatibility with SQL Server Enterprise Edition, running on-premises
        + The SQL Database managed instance deployment option supports traditional SQL Server Database engine logins and logins integrated with Azure Active Directory (AD).
        + SQL Database managed instance supports linked servers,
      * Although some of the other advanced features required by the database might not be available. If you want a complete match, then running SQL Server on a virtual machine may be your only option, but you need to balance the benefits of complete functionality against the administrative and maintenance overhead required
  + **Azure Database for MySQL**:MySQL started life as a simple-to-use open-source database management system
    - It's available in several editions; Community (free), Standard (higher performance, and uses a different technology for storing data), and Enterprise (comprehensive set of tools and features, including enhanced security, availability, and scalability).
    - Azure Database for MySQL is a PaaS implementation of MySQL in the Azure cloud, based on the MySQL Community Edition
    - Azure Database for MySQL provides a global database system that scales up to large databases without the need to manage hardware, network components, virtual servers, software patches, and other underlying components
    - Azure Database for MySQL servers provides monitoring functionality to add alerts, and to view metrics and logs
    - The system uses pay-as-you-go pricing so you only pay for what you use
    - **Benefits**:
      * High availability features built-in.
      * Predictable performance.
      * Easy scaling that responds quickly to demand.
      * Secure data, both at rest and in motion.
      * Automatic backups and point-in-time restore for the last 35 days.
      * Enterprise-level security and compliance with legislation.
  + **Azure Database for MariaDB**:Azure Database for MariaDB (Community Edition) is an implementation of the MariaDB database management system adapted to run in Azure
    - MariaDB is a newer database management system, created by the original developers of MySQL. The database engine has since been rewritten and optimized to improve performance
    - MariaDB offers compatibility with Oracle Database
    - One notable feature of MariaDB is its built-in support for temporal data. A table can hold several versions of data, enabling an application to query the data as it appeared at some point in the past
    - The database is fully managed and controlled by Azure. Once you've provisioned the service and transferred your data, the system requires almost no additional administration
    - **Benefits**:
      * Built-in high availability with no additional cost.
      * Predictable performance, using inclusive pay-as-you-go pricing.
      * Scaling as needed within seconds.
      * Secured protection of sensitive data at rest and in motion.
      * Automatic backups and point-in-time-restore for up to 35 days.
      * Enterprise-grade security and compliance.
  + **Azure Database for PostgreSQL**: It is a PaaS implementation of PostgreSQL in the Azure Cloud.
    - PostgreSQL is a hybrid relational-object database. You can store data in relational tables, but a PostgreSQL database also enables you to store custom data types, with their own non-relational properties
    - The database management system is extensible; you can add code modules to the database, which can be run by queries
    - Another key feature is the ability to store and manipulate geometric data, such as lines, circles, and polygons.
    - PostgreSQL has its own query language called **pgsql**.
    - This service provides the same availability, performance, scaling, security, and administrative benefits as the MySQL service.
    - Some features of on-premises PostgreSQL databases are not available in Azure Database for PostgreSQL. These features are mainly concerned with the extensions that users can add to a database to perform specialized tasks
    - Azure Database for PostgreSQL has two deployment options: Single-server and Hyperscale.
      * **Single-server**: The single-server deployment option for PostgreSQL provides similar benefits as Azure Database for MySQL. You choose from three pricing tiers: Basic, General Purpose, and Memory Optimized. Each tier supports different numbers of CPUs, memory, and storage sizes
      * **Hyperscale (Citus)**: Hyperscale (Citus) is a deployment option that scales queries across multiple server nodes to support large database loads. Your database is split across nodes. Data is split into chunks based on the value of a partition key or sharding key. Consider using this deployment option for the largest database PostgreSQL deployments in the Azure Cloud
    - **Benefits**:
      * highly available service. It contains built-in failure detection and failover mechanisms
      * Users of PostgreSQL will be familiar with the **pgAdmin** tool, which you can use to manage and monitor a PostgreSQL database. You can continue to use this tool to connect to Azure Database for PostgreSQL. However, some server-focused functionality, such as performing server backup and restore, are not available because the server is managed and maintained by Microsoft
      * Azure Database for PostgreSQL servers records information about the queries run against databases on the server, and saves them in a database named **azure\_sys**. You query the **query\_store.qs\_view** view to see this information, and use it to monitor the queries that users are running. This information can prove invaluable if you need to fine-tune the queries performed by your applications.
      * Excellent support for multi-tenant applications, real time operational analytics, and high throughput transactional workloads



* 1. **SQL Server on Azure Virtual Machines**: Migrating from the system running on-premises to an Azure virtual machine is no different than moving the databases from one on-premises server to another.
  + The database runs stored procedures and scripts as part of the database workload. If these stored procedures and scripts depend on features that are restricted by following a PaaS approach, then running SQL Server on your own virtual machines might be a good option
  + The term lift-and-shift refers to the way in which you can move a database directly from an on-premises server to an Azure virtual machine without requiring that you make any changes to it
  + With a virtual machine, you have the full administrative rights over the DBMS and operating system. It's a perfect choice when an organization already has IT resources available to maintain the virtual machines.
  + With a virtual machine, you have the full administrative rights over the DBMS and operating system. It's a perfect choice when an organization already has IT resources available to maintain the virtual machines.
* The Database Migration Service enables you to restore a backup of your on-premises databases directly to databases running in Azure Data Services. You can also **configure replication from an on-premises** database, so that any changes made to data in that database are copied to the database running in Azure Data Services. This strategy enables you to reconfigure users and applications to **connect to the database in the cloud while the on-premises system is still active**; you don't have to shut down the on-premises system while you transfer users to the cloud.

1. **Provisioning and deploying relational database services in Azure**:

* Provisioning is the act of running series of tasks that a service provider, such as Azure SQL Database, performs to create and configure a service
* Azure provides several tools you can use to provision services: The Azure portal, The Azure command-line interface (CLI), Azure PowerShell and Azure Resource Manager templates.
* **postgres is the default management database** created with Azure Database for PostgreSQL. You can create additional databases using the CREATE DATABASE command from psql
* **Configuring relational data services**: After you've provisioned a resource, you'll often need to configure it to meet the needs of your applications and environment. For example, you might need to set up network access, or open a firewall port to enable your applications to connect to the resource.
* The default connectivity for Azure relational data services is to disable access to the world. Configure connectivity to virtual networks and on-premises computers
* Azure SQL Database communicates over port 1433 (PostgreSQL over 5432 and MySQL over 3306). If you're trying to connect from within a corporate network, outbound traffic over port 1433 might not be allowed by your network's firewall. If so, you can't connect to your Azure SQL Database server unless your IT department opens port 1433.
* A firewall rule of 0.0.0.0 enables all Azure services to pass through the server-level firewall rule and attempt to connect to a single or pooled database through the server.
* Private Endpoint uses a private IP address from your virtual network, effectively bringing the service into your virtual network. The service could be an Azure service such as Azure App Service, or your own Private Link Service
* Items that implement network-based ACLs include routers and load balancers. You control traffic flow through these items by defining firewall rules.
* The following steps describe how a connection is established to an Azure SQL database:
  + Clients connect to a gateway that has a public IP address and listens on port 1433.
  + Depending on the effective connection policy, the gateway either redirects traffic to the database cluster, or acts as a proxy for the database cluster.
  + Inside the database cluster, traffic is forwarded to the appropriate Azure SQL database.
* **Connectivity from within Azure**: f you're connecting from within another Azure service, such as a web application running under Azure App Service, your connections have a connection policy of Redirect by default. A policy of Redirect means that after your application establishes a connection to the Azure SQL database through the gateway, all following requests from your application will go directly to the database rather than through the gateway
* **Connectivity from outside of Azure**: If you're connecting from outside Azure, such as an on-premises application, your connections have a connection policy of Proxy by default. A policy of Proxy means the connection is established via the gateway, and all subsequent requests flow through the gateway. Each request could (potentially) be serviced by a different database in the cluster.



* With **Azure Active Directory (AD) authentication**, you can centrally manage the identities of database users and other Microsoft services in one central location
* You control access to resources using Azure RBAC to create role assignments. A role assignment consists of three elements: a security principal, a role definition, and a scope
  + A **security principal** is an object that represents a user, group, service principal, or managed identity that is requesting access to Azure resources
  + A **role definition**, often abbreviated to role, is a collection of permissions. E.g. Owner, Contributor, Reader, User Access Administrator.
  + A **scope** lists the set of resources that the access applies to
* **Advanced data security** implements threat protection and assessment. Threat protection adds security intelligence to your service.
* PostgreSQL also provides the ability to extend the functionality of your database using extensions.After being loaded in the database, extensions function like built-in features.To install a particular extension, run the **CREATE EXTENSION** command from psql tool to load the packaged objects into your database.
* You can replicate data from an Azure Database for PostgreSQL and MySQL server to a read-only server. Azure Database for PostgreSQL supports replication from the master server to up to five replicas. **Read replicas** help to improve the performance and scale of read-intensive workloads. Read workloads can be isolated to the replicas, while write workloads can be directed to the master.A common scenario is to have BI and analytical workloads use read replicas as the data source for reporting
* By default**, SSL connection security** is required and enforced on your Azure Database for MySQL server

1. **Structured Query Language(SQL)**: SQL is used to communicate with a relational database.

* SQL was originally standardized by the American National Standards Institute (ANSI) in 1986, and by the International Organization for Standardization (ISO) in 1987. Since then, the standard has been extended several times as relational database vendors have added new features to their systems. Additionally, most database vendors include their own proprietary extensions that are not part of the standard, which has resulted in a variety of dialects of SQL
* Some popular **dialects of SQL** include:
  + **Transact-SQL (T-SQL):** This version of SQL is used by Microsoft SQL Server and Azure SQL Database.
  + **pgSQL**: This is the dialect, with extensions implemented in PostgreSQL.
  + **PL/SQL:**This is the dialect used by Oracle. PL/SQL stands for Procedural Language/SQL.
* SQL statements are grouped into two main logical groups, and they are:
  + **Data Manipulation Language (DML):**You use DML statements to manipulate the rows in a relational table. SELECT, INSERT-INTO, UPDATE-SET, DELETE with clauses like WHERE, ORDER BY, JOIN, INTO (You specify a table and columns in an INTO clause, and a list of values to be stored in these columns.) and aggregate functions like MIN(), MAX(), AVG(), SUM()
  + **Data Definition Language (DDL)**:You use DDL statements to create, modify, and remove tables and other objects in a database (table, stored procedures, views, and so on). CREATE, ALTER, RENAME and DROP
* SQL doesn't provide are you sure? prompts, so be careful when using DELETE or UPDATE without a WHERE clause because you can lose or modify a lot of data
* The DROP statement is very powerful. When you drop a table, all the rows in that table are lost. Unless you have a backup, you won't be able to retrieve this data
* An empty column in a row is said to have a NULL value.
* E.g.

UPDATE MyTable

SET MyColumn2 = 'contoso'

WHERE MyColumn1 = 3

DELETE FROM MyTable

WHERE MyColumn2 = 'contoso'

INSERT INTO MyTable(MyColumn1, MyColumn2, MyColumn3)

VALUES (99, 'contoso', 'hello')

CREATE TABLE MyTable

(

MyColumn1 INT NOT NULL PRIMARY KEY,

MyColumn2 VARCHAR(50) NOT NULL,

MyColumn3 VARCHAR(10) NULL

);

* You can use any of these **tools to query data** held in **Azure SQL Database**:
  + The query editor in the Azure portal
  + The sqlcmd utility from the command line or the Azure Cloud Shell

(sqlcmd -S <server>.database.windows.net -d <database> -U <username> -P <password>). If the sign-in command succeeds, you'll see a 1> prompt. You can enter SQL commands, then type GO on a line by itself to run them.

* + SQL Server Management Studio
  + Azure Data Studio
  + SQL Server Data Tools in Visual Studio
* **Transact-SQL dialect**:
  + CREATE DATABASE and ALTER DATABASE commands are part of the Transact-SQL dialect, and aren't part of standard SQL.
  + Again, the version of the INSERT statement, with multiple VALUES clauses, is part of the Transact-SQL dialect.
  + The -- characters start a comment in Transact-SQL.
  + The [ and ] characters surround identifiers, such as the name of a table, database, column, or data type.
  + The N character in front of a string indicates that the string uses the Unicode character set.
* You can't create new SQL databases from a connection in Azure Data Studio if you're running SQL Database single database or elastic pools. You can only create new databases in this way if you're using SQL Database managed instance.
* **Tools for PostgreSQL**:
  + pgAdmingraphical user interface
  + psql command-line utility (psql --host=<server-name>.postgres.database.azure.com --username=<admin-user>@<server-name> --dbname=postgres). If your connection is successful, you'll see the prompt postgres=>zure Data Studio
  + Azure Data Studio
* **Tools for MySQL**:
  + MySQL Workbenchgraphical user interface
  + mysql command-line utility
  + Currently there are **no extensions available** for connecting to MySQL from Azure Data Studio

1. **Non-Relational Data in Azure**:
   1. **Azure Table storage**: Azure Table Storage implements the NoSQL key-value model. You create a table using an Azure storage account
   * In an Azure Table Storage table, items are referred to as rows, and fields are known as columns
   * An Azure table enables you to store semi-structured data. All rows in a table must have a key, but apart from that the columns in each row can vary.
   * Unlike traditional relational databases, Azure Table Storage tables have no concept of relationships, stored procedures, secondary indexes, or foreign keys.
   * Data will usually be denormalized, with each row holding the entire data for a logical entity
   * Azure Table Storage provides much faster access to the data because the data is available in a single row, without requiring that you perform joins across relationships.
   * To help ensure fast access, Azure Table Storage splits a table into partitions. Partitioning is a mechanism for grouping related rows, based on a common property or partition key
   * The key in an Azure Table Storage table comprises two elements; the partition key that identifies the partition containing the row (as described above), and a row key that is unique to each row in the same partition
   * Items in the same partition are stored in row key order. If an application adds a new row to a table, Azure ensures that the row is placed in the correct position in the table. This scheme enables an application to quickly perform Point queries that identify a single row, and Range queries that fetch a contiguous block of rows in a partition
   * The columns in a table can hold numeric, string, or binary data up to 64 KB in size. A table can have to 252 columns, apart from the partition and row keys. The maximum row size is 1 MB.
   * **Primary advantages** of using Azure Table Storage tables over other ways of storing data:
     + It's simpler to scale. It takes the same time to insert data in an empty table, or a table with billions of entries. An Azure storage account can hold up to 5 PB of data.
     + A table can hold semi-structured data
     + There's no need to map and maintain the complex relationships typically required by a normalized relational database.
     + Row insertion is fast
     + Data retrieval is fast, if you specify the partition and row keys as query criteria
   * **Disadvantages**:
     + Consistency needs to be given consideration as transactional updates across multiple entities aren't guaranteed
     + There's no referential integrity; any relationships between rows need to be maintained externally to the table
     + It's difficult to filter and sort on non-key data. Queries that search based on non-key fields could result in full table scans
   * Azure Table Storage is an excellent mechanism for:
     + Storing TBs of structured data capable of serving web scale applications. Examples include product catalogs for eCommerce applications, and customer information, where the data can be quickly identified and ordered by a composite key. In the case of a product catalog, the partition key could be the product category (such as footwear), and the row key identifies the specific product in that category (such as climbing boots).
     + Storing datasets that don't require complex joins, foreign keys, or stored procedures, and that can be denormalized for fast access. In an IoT system, you might use Azure Table Storage to capture device sensor data. Each device could have its own partition, and the data could be ordered by the date and time each measurement was captured.
     + Capturing event logging and performance monitoring data. Event log and performance information typically contain data that is structured according to the type of event or performance measure being recorded. The data could be partitioned by event or performance measurement type, and ordered by the date and time it was recorded.
   * Azure Table Storage provides high-availability guarantees in a single region.
   * Azure Table Storage helps to protect your data. You can configure security and role-based access control to ensure that only the people or applications that need to see your data can actually retrieve it.
   1. **Azure Blob storage**: Many applications need to store large, binary data objects, such as images and video streams. Microsoft Azure virtual machines use blob storage for holding virtual machine disk images. These objects can be several hundreds of GB in size. Blob is an acronym for Binary Large OBject.
   * Azure Blob storage is a service that enables you to store massive amounts of unstructured data, or blobs, in the cloud
   * Three different **types of blob**:
     + **Block blobs**. A block blob is handled as a set of blocks. Each block can vary in size, up to 100 MB. A block blob can contain up to 50,000 blocks, giving a maximum size of over 4.7 TB. The block is the smallest amount of data that can be read or written as an individual unit. Block blobs are best used to store discrete, large, binary objects that change infrequently (static data) like images.
     + **Page blobs**. A page blob is organized as a collection of fixed size 512-byte pages. A page blob is optimized to support **random read and write operations**; you can fetch and store data for a single page if necessary. A page blob can hold up to 8 TB of data. Azure uses page blobs to implement **virtual disk storage** for virtual machines.
     + **Append blobs**. An append blob is a block blob optimized to support append operations. You can only add blocks to the end of an append blob; updating or deleting existing blocks isn't supported. Each block can vary in size, up to 4 MB. The maximum size of an append blob is just over 195 GB.

Append blobs are suitable **for storing data that grows in chunks**, such as logs or other archive data

* + Inside an Azure storage account, you create blobs inside containers. A container provides a convenient way of grouping related blobs together, and you can organize blobs in a hierarchy of folders, similar to files in a file system on disk. You control who can read and write blobs inside a container at the container level
  + Blob storage provides three access tiers: Hot, Cool and Archived.
  + You can create lifecycle management policies for blobs in a storage account. A lifecycle management policy can automatically move a blob from Hot to Cool, and then to the Archive tier, as it ages and is used less frequently. A lifecycle management policy can also arrange to delete outdated blobs.
  + Common **uses of Azure Blob Storage** include:
    - Serving images or documents directly to a browser, in the form of a static website. Visit Static website hosting in Azure storage for detailed information.
    - Storing files for distributed access
    - Streaming video and audio
    - Storing data for backup and restore, disaster recovery, and archiving
    - Storing data for analysis by an on-premises or Azure-hosted service
    - Azure Blob storage is also used as the basis for Azure Data Lake storage. You can use Azure Data Lake storage for performing big data analytics
  + **Azure Blob storage Features**:
    - **Versioning**. You can maintain and restore earlier versions of a blob.
    - **Soft delete**. This feature enables you to recover a blob that has been removed or overwritten, by accident or otherwise.
    - **Snapshots**. A snapshot is a read-only version of a blob at a particular point in time.
    - **Change Feed**. The change feed for a blob provides an ordered, read-only, record of the updates made to a blob
  1. **Azure File storage**:Many on-premises systems comprising a network of in-house computers make use of file shares
  + Azure File Storage exposes file shares using the Server Message Block 3.0 (SMB) protocol.
  + You can control access to shares in Azure File Storage using authentication and authorization services available through Azure Active Directory Domain Services.
  + You create Azure File storage in a storage account. Azure File Storage enables you to share up to 100 TB of data in a single storage account. This data can be distributed across any number of file shares in the account. The maximum size of a single file is 1 TB, but you can set quotas to limit the size of each share below this figure. Currently, Azure File Storage supports up to 2000 concurrent connections per shared file.
  + you can upload files to Azure File Storage using the Azure portal, or tools such as the AzCopy utility. You can also use the Azure File Sync service to synchronize locally cached copies of shared files with the data in Azure File Storage
  + Azure File Storage is designed to support:
    - Migrate existing applications to the cloud.
    - Share server data across on-premises and cloud
    - Integrate modern applications with Azure File Storage
    - Simplify hosting High Availability (HA) workload data.
  + Azure aims to provide up to 300 MB/second of throughput for a single Standard file share, but you can increase throughput capacity by creating a Premium file share, for additional cost.
  + All data is encrypted at rest, and you can enable encryption for data in-transit between Azure File Storage and your applications.
  1. **Azure Cosmos DB**:Azure Cosmos DB is a multi-model NoSQL database management system. Cosmos DB manages data as a partitioned set of documents.
  + A document is a collection of fields, identified by a key. The fields in each document can vary, and a field can contain child documents.
  + A document can hold up to 2 MB of data, including small binary objects. If you need to store larger blobs as part of a document, use Azure Blob storage, and add a reference to the blob in the document
  + **The APIs that Cosmos DB currently supports**:
    - **SQL API**: This interface provides a SQL-like query language over documents
    - **Table API**: This interface enables you to use the Azure Table Storage API to store and retrieve documents. The purpose of this interface is to enable you to switch from Table Storage to Cosmos DB without requiring that you modify your existing applications
    - **MongoDB API**: MongoDB is another well-known document database, with its own programmatic interface. You can use the MongoDB API for Cosmos DB to enable a MongoDB application to run unchanged against a Cosmos DB database.
    - **Cassandra API**: Cassandra is a column family database management system. The Cassandra API for Cosmos DB provides a Cassandra-like programmatic interface for Cosmos DB. Cassandra API requests are mapped to Cosmos DB document requests. Primary purpose of the Cassandra API is to enable you to quickly migrate Cassandra databases and applications to Cosmos DB.
    - **Gremlin API**: The Gremlin API implements a graph database interface to Cosmos DB. Data is still held as a set of documents in Cosmos DB, but the Gremlin API enables you to perform graph queries over data.
  + The principal use of the Table, MongoDB, and Cassandra APIs is to support existing applications written using these data stores. If you're building a new application and database, you should use the SQL API or Gremlin API
  + Documents in a Cosmos DB database are organized into containers. The documents in a container are grouped together into partitions. A partition holds a set of documents that share a common partition key.You should select a partition key that collects all related documents together. This approach helps to reduce the amount of I/O (disk reads) that queries might need to perform when retrieving a set of documents for a given entity
  + There's a superficial similarity between a Cosmos DB container and a table in Azure Table storage: in both cases, data is partitioned and documents (rows in a table) are identified by a unique ID within a partition. However, unlike Azure Table storage, documents in a Cosmos DB partition aren't sorted by ID. Instead, Cosmos DB maintains a separate index. This index contains not only the document IDs, but also tracks the value of every other field in each document. This index is created and maintained automatically. This index enables you to perform queries that specify criteria referencing any fields in a container, without incurring the need to scan the entire partition to find that data
  + Each partition can grow up to 10 GB in size.
  + Cosmos DB guarantees less than 10-ms latencies for both reads (indexed) and writes at the 99th percentile, all around the world.This capability enables sustained ingestion of data and fast queries for highly responsive apps.
  + all data in Cosmos DB is encrypted at rest and in motion
  + Cosmos DB is a foundational service in Azure. Cosmos DB has been used by many of Microsoft's products for mission critical applications at global scale, including Skype, Xbox, Microsoft 365, Azure, and many others.
  + Cosmos DB is highly suitable forIoT and telematics, Retail and marketing, Gaming, Web and mobile applications.

1. **Provisioning Non-Relational Data in Azure**:
   1. **Provisioning AZURE Cosmos DB**:Azure Cosmos DB uses the concept of Request Units per second (RU/s) to manage the performance and cost of databases.You can think of a request unit as the amount of computation and I/O resources required to satisfy a simple read request made to the database. Microsoft gives a measure of approximately one RU as the resources required to read a 1-KB document with 10 fields. So a throughput of one RU per second (RU/s) will support an application that reads a single 1-KB document each second. The minimum throughput you can allocate to a database or container is 400 RU/s.If you underprovision (by specifying too few RU/s), Cosmos DB will start throttling performance. Allocating more RU/s increases the costwhether you use resources or not
      * **Create Azure Cosmos DB account**
      * **Create Databaseand Create Container**,using Data Explorer page link in left menu
   * Azure Cosmos DB uses the concept of Request Units per second (RU/s) to manage the performance and cost of databases. You can think of a request unit as the amount of computation and I/O resources required to satisfy a simple read request made to the database. Microsoft gives a measure of approximately one RU as the resources required to read a 1-KB document with 10 fields. So a throughput of one RU per second (RU/s) will support an application that reads a single 1-KB document each second. The minimum throughput you can allocate to a database or container is 400 RU/s.
   * If you underprovision (by specifying too few RU/s), Cosmos DB will start throttling performance. Allocating more RU/s increases the cost whether you use resources or not
   * If you applied the Free Tier Discount to your Cosmos DB account, you get the first 400 RU/s for a single database or container for free. 400 RU/s is enough capacity for most small to moderate databases.
   1. **Provisioning Other non-relational resources in Azure**: Data Lake storage, Blob storage, and File Storage, Tables, Queues.
      * **Create storage account**
      * **Create Blob Container or FileShare**
   * Data Lake storage is only available with a standard storage account, not premium
   * To maintain performance, premium storage accounts only support LRS replication. This is because replication is performed synchronously to maintain data integrity. Replicating data to a distant region can increase latency to the point at which any advantages of using premium storage are lost.
   * If you're **provisioning a Data Lake storage**, you must specify the appropriate configuration settings when you create the storage account. You can't configure Data Lake storage after the storage account has been set up. In the Azure portal, on the Advanced tab of the Create storage account page, in the Data Lake Storage Gen2 section, select Enabled for the Hierarchical namespace option
   * The total size of all files across all file shares in a storage account can't exceed 5120 GB

* **Configuring non-relational data services:** This include general configurations such as**Network, Firewall, Authentication & Authorization, Access Control, Security** like threat protection and assessment, etc. Most of these general configurations aresimilar as that for Relational Azure resources.
  + Access keys provide a rather coarse-grained level of authentication. Additionally, if you need to regenerate an access key (after accidental disclosure, for example), you may need to update all applications that connect using that key.
  + Azure Active Directory (Azure AD) provides superior security and ease of use over access key authorization. Microsoft recommends using Azure AD authorization when possible to minimize potential security vulnerabilities inherent in using access keys
  + **Specific configuration settings**: Apart from the general configuration settings applicable to many services, most services also have specific configuration settings. Example: you may need to configure **replication**, or **database consistency settings** for Azure Cosmos DB
    - **Replication:** The Replicate data globally page enables you to configure replication in more detail. You can replicate to multiple regions, and you select the regions to use. In this way, you can pick the regions that are closest to your consumers, to help minimize the latency of requests made by those consumers.
    - You can also use this page to configure automatic failover to help ensure high availability
    - By default, only the region in which you created the account supports write operations; the replicas are all read-only. However, you can enable multi-region writes. Multi-region writes can cause conflicts though, if applications running in different regions modify the same data. In this case, the most recent write will overwrite changes made earlier when data is replicated, although you can write your own code to apply a different strategy.
    - Replication is asynchronous, so there's likely to be a lag between a change made in one region, and that change becoming visible in other regions.
    - Each replica increases the cost of the Cosmos DB service. For example, if you replicate your account to two regions, your costs will be three times that of a non-replicated account.
    - **Configure database consistency**: Within a single region, Cosmos DB uses a cluster of servers. This approach helps to improve scalability and availability. A copy of all data is held in each server in the cluster
    - Cosmos DB enables you to specify how such inconsistencies should be handled. It provides the following options:
      * **Eventual**: This option is the least consistent. It's based on the situation just described. Changes won't be lost, they'll appear eventually, but they might not appear immediately. Multiple changes could appear out of order.
      * **Consistent Prefix**: This option ensures that changes will appear in order
      * **Session**: If an application makes a number of changes, they'll all be visible to that application, and in order. Other applications may see old data. sometimes known as read your own writes
      * **Bounded Staleness**: There's a lag between writing and then reading the updated data. You specify this staleness.
      * **Strong**:In this case, all writes are only visible to clients after the changes are confirmed as written successfully to all replicas. This option is unavailable if you need to distribute your data across multiple global regions
    - Eventual consistency provides the lowest latency and least consistency. Strong consistency results in the highest latency but also the greatest consistency. You should select a default consistency level that balances the performance and requirements of your applications.
  + Options, such as the account kind and performance tier in **Azure Storage Account Configurations**, are displayed on this page for information only. You can't change them.
  + All data held in an Azure Storage account is automatically encrypted. By default, **encryption** is performed using keys managed and owned by Microsoft. If you prefer, you can provide your own encryption keys (CMK)
  + **Shared access signatures**: You can use shared access signatures (SAS) to grant limited rights to resources in an Azure storage account for a specified time period, without requiring that users are authenticated first.You should only use SAS for data that you intend to make public. The application can then send requests to read or write data using this URL and SAS token.

1. **Manage non-relational data stores in Azure**:

* Perform data operations in Cosmos DB
  + Use Data Explorer in the Azure portal to run ad-hoc queries. You can also use this tool to load data, but you can only load one document at a time. The data load functionality is primarily aimed at uploading a small number of documents (up to 2 MB in total size) for test purposes, rather than importing large quantities of data.
  + Use the [Cosmos DB Data Migration tool](https://docs.microsoft.com/en-us/azure/cosmos-db/import-data) to perform a bulk-load or transfer of data from another data source.
  + Use [Azure Data Factory](https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-cosmos-db) to import data from another source.
  + Write a custom application that imports data using the Cosmos DB [BulkExecutor](https://docs.microsoft.com/en-us/azure/cosmos-db/tutorial-sql-api-dotnet-bulk-import) library. This strategy is beyond the scope of this module.
  + Create your own application that uses the functions available through the [Cosmos DB SQL API client library](https://docs.microsoft.com/en-us/azure/cosmos-db/create-sql-api-dotnet-v4) to store data. This approach is also beyond the scope of this module
* **Load data using the Cosmos DB Data Migration tool**: You can use the Data Migration tool to import data to Azure Cosmos DB from a variety of sources, including:
  + - JSON files
    - MongoDB
    - SQL Server
    - CSV files
    - Azure Table storage
    - Amazon DynamoDB
    - HBase
    - Azure Cosmos containers
  + The Data Migration tool is available as a download from GitHub. The tool guides you through the process of migrating data into a Cosmos DB database. You're prompted for the source of the data (one of the items listed above), and the destination (the Cosmos DB database and container). The tool can either populate an existing container, or create a new one if the specified container doesn't already exist
  + You can also use the Data Migration tool to export data from a Cosmos DB container to a JSON file, either held locally or in Azure Blob storage
* **Query Azure Cosmos DB**: Although Azure Cosmos DB is described as a NoSQL database management system, the SQL API enables you to run SQL-like queries against Cosmos DB databases. These queries use a syntax similar to that of SQL, but there are some differences. This is because the data in a Cosmos DB is structured as documents rather than tables
  + The SQL API returns results in the form of JSON documents.
  + All queries are executed in the context of a single container
  + **FROM clause**: In a relational database query, the FROM clause would contain a table name. In the SQL API, all queries are limited to the scope of a container, so the identifier represents the name of the container.
  + **JOIN Clause**: In the SQL API, you use JOIN clauses to connect fields in a document with fields in a subdocument that is part of the same document. You can't perform joins across different documents.
  + **DISTINCT operator**: use as part of the SELECT clause to eliminate duplicates in the result data
  + **TOP operator**: use to retrieve only the first few rows returned by a query that might otherwise generate a large result set.
  + **BETWEEN operator**: use as part of the WHERE clause to define an inclusive range of values. The condition field BETWEEN a AND b is equivalent to the condition field >= a AND field <= b
  + **IS\_DEFINED operator**: use for detecting whether a specified field exists in a document
  + You can use aggregate functions to summarize data in SELECT queries: SUM, COUNT, MIN, MAX, AVG
  + The SQL API also supports a large number of mathematical, trigonometric, string, array, and spatial functions. Refer [Getting started with SQL queries in Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/sql-api-sql-query).

SELECT p.productname, p.color, p.listprice, p.description, p.images.thumbnail

FROM products p

WHERE p.productcategory.subcategory = "Mountain Bikes"

SELECT DISTINCT c.Address.City

FROM c

WHERE c.Address.State BETWEEN "AK" AND "MD"

SELECT TOP 3 \*

FROM c

ORDER BY c.Name

SELECT \* FROM p

WHERE IS\_DEFINED(p.DateOfBirth)

SELECT COUNT(p.productname)

FROM products p

WHERE p.productcategory.subcategory = "Touring Bikes"

SELECT VALUE COUNT(p.productname)

FROM products p

WHERE p.productcategory.subcategory = "Touring Bikes"

SELECT VALUE SUM(p.quantityinstock)

FROM products p

WHERE p.productcategory.subcategory = "Touring Bikes"

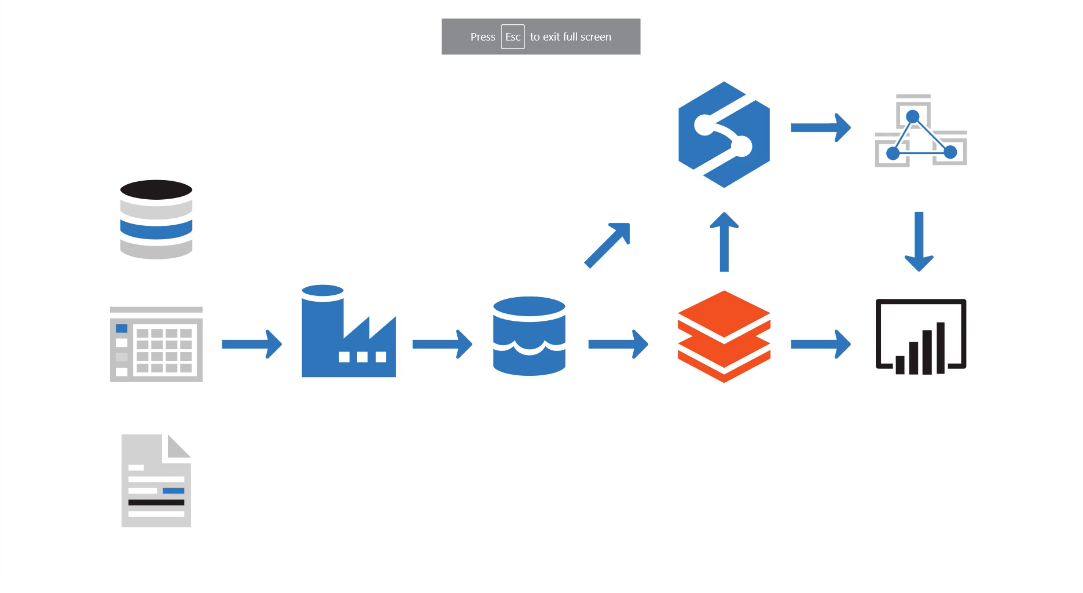
* + You can use the GUI (Azure portal,Azure Storage Explorer), Azure CLI, and Azure PowerShell, AzCopy utility and for managing blobs, blob storage and file storage.
  + **Azure Storage Explorer** is a utility that enables you to manage Azure Storage accounts from your desktop computer. You can download it from the [Azure Storage Explorer](https://azure.microsoft.com/features/storage-explorer/). You can use Storage Explorer to create blob containers and file shares, as well as upload, download and delete files. A version of this utility is also available in the Azure portal, on the Overview page for an Azure Storage account.
  + You can aupload and download individual files to and from Azure File storage manually, by using Storage Explorer, the Azure portal, or **by connecting the file share to your desktop computer** and dragging and dropping files in File Explorer.
  + **AzCopy Utility**: If you need to transfer a significant number of files in and out of Azure File storage, you should use the AzCopy utility. AzCopy is a command-line utility **optimized for transferring large files (and blobs)** between your local computer and Azure File storage. It can detect transfer failures, and restart a failed transfer at the point an error occurred - you don't have to repeat the entire operation
  + **SAS token**: Before you can use AzCopy, you generate a Shared access signature (SAS) token. A SAS token provides controlled, time-limited, anonymous access to services and resources in a storage account; users don't have to provide any additional credentials. SAS tokens are useful in situations where you don't know in advance which users will require access to your resources.
  + The AzCopy command also supports authentication using Azure Active Directory, but this approach requires adding all of your users to Azure Active Directory first.
  + Don't forget to include the copy keyword after the azcopy command. AzCopy supports other operations, such as deleting files and blobs, listing files and blobs, and creating new file shares. Each of these operations has its own keyword.

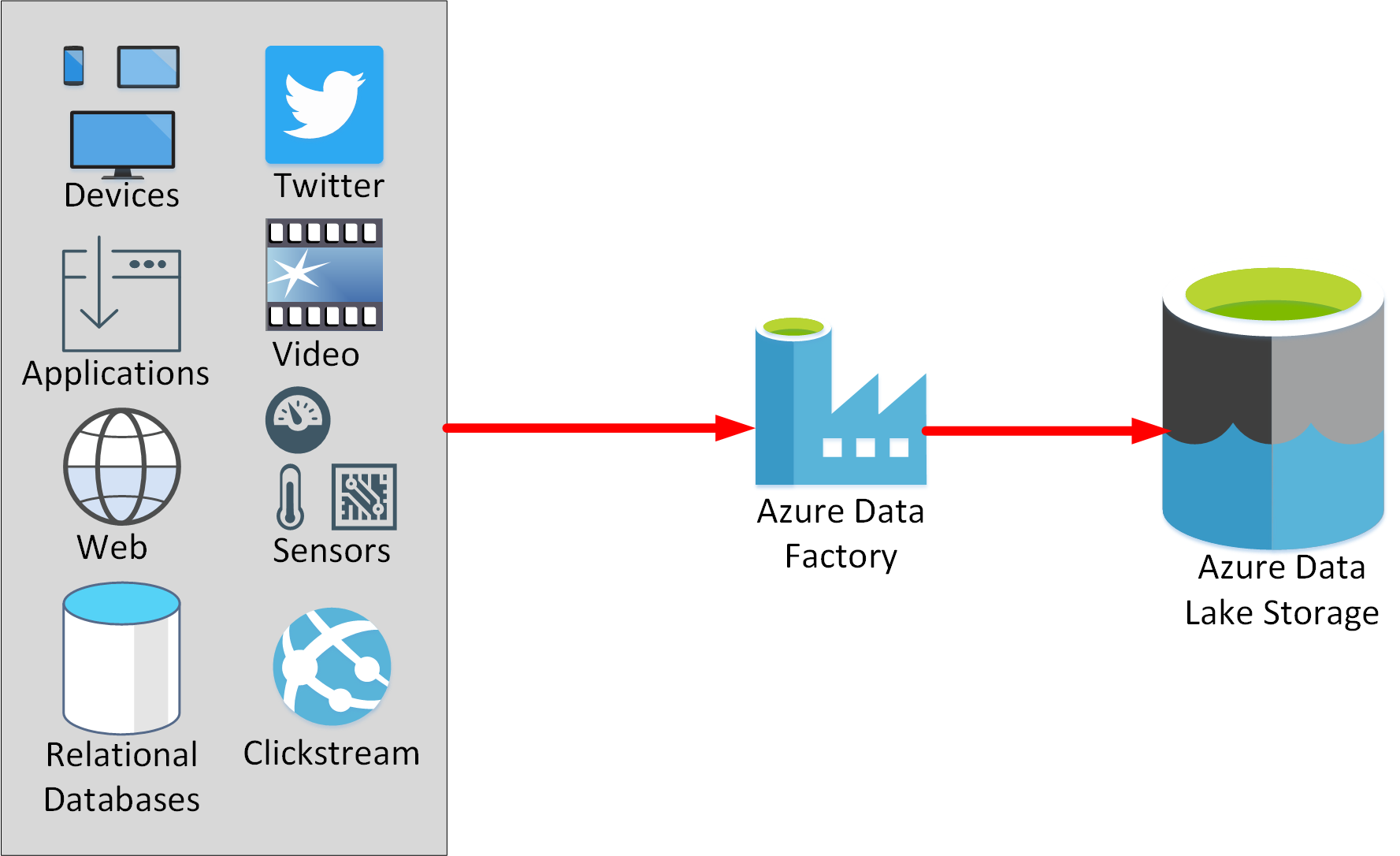
azcopy copy "myfile.txt" https://<storage-account-name>.file.core.windows.net/<file-share-name>/myfile.txt<SAS-token>

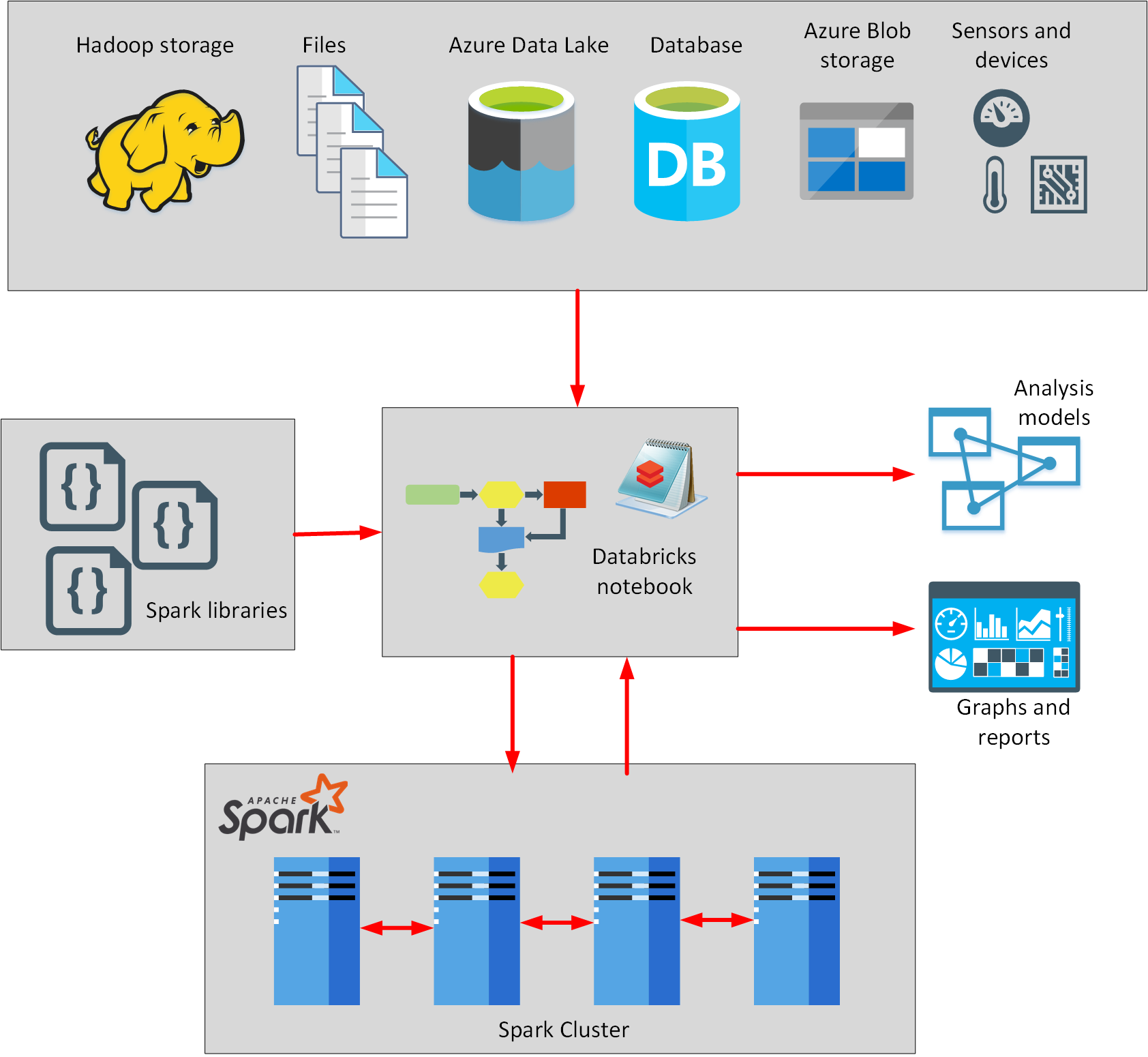
* + You can transfer the entire contents of a local folder to Azure File storage using a similar command. You replace the file name ("myfile.txt") with the name of the folder. If the folder contains subfolders that you want to copy, add the --recursive flag.

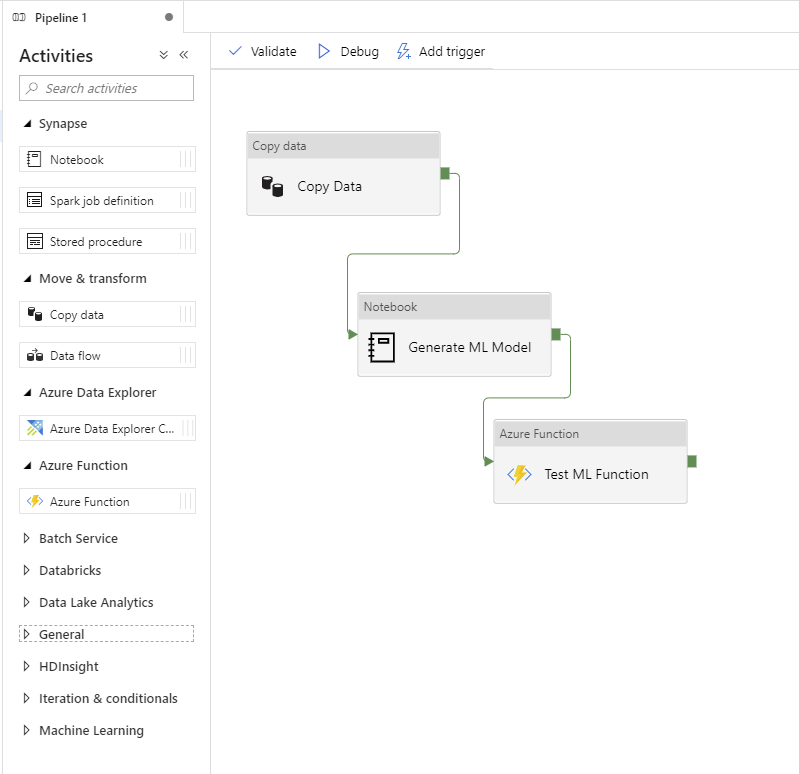
azcopy copy "myfolder" "https://<storage-account-name>.file.core.windows.net/<file-share-name>/myfolder<SAS-token>" –recursive

1. **Examine components of a modern data warehouse**:There's often a plethora of useful information available outside of organizations. This information could be combined with local data to add insights and enrich understanding. By combining all local data with useful external information, it's often possible to gain insights into the data that weren't previously possible.

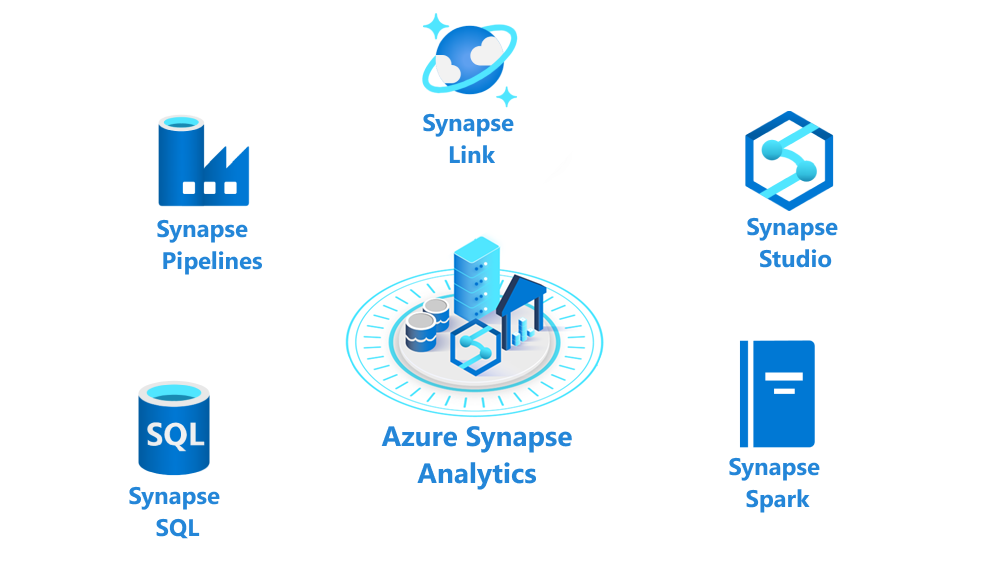
* The process of combining all of the local data sources is known as **datawarehousing.** This data is then used as the source for analysis, reporting, and online analytical processing (OLAP)
* The process of analyzing streaming data and data from the Internet is known as **Big Data analytics**.
* **Azure Synapse Analytics** combines data warehousing with Big Data analytics
* The focus of a data warehouse is to provide answers to complex queries, unlike a traditional relational database, which is focused on transactional performance
* Data warehouses have to handle **big data**. Big data is the term used for large quantities of data collected in escalating volumes, at higher velocities, and in a greater variety of formats than ever before. It can be historical (meaning stored) or real time (meaning streamed from the source). Businesses typically depend on their big data to help make critical business decisions.
* A modern data warehouse might contain a **mixture of relational and non-relational data**, including files, social media streams, and Internet of Things (IoT) sensor data.
* **Azure** provides a collection of services you can use to build a data warehouse solution, including
  1. Azure Data Factory (Data Ingestion from a variety of sources),
  2. Azure Data Lake Storage (Data Storage),
  3. Azure Databricks (Big data processing/transformation orData Cleaning),
  4. Azure Synapse Analytics (process large amounts of dataand Hold Cleaned Data for Business / cooked business information),
  5. Azure Analysis Services (analyze and Reporting).
  6. Power BI to analyze and visualize the data, generating reports, charts, and dashboards
* The up-to-the-second data might be used to help monitor real-time, critical manufacturing processes, where an instant decision is required. (ex. Stock current prices).
* Historical data is equally important, to give a business a more stabilized view of trends in performance (ex. volumes of sales by products across a month, a quarter, or a year, to determine whether to continue producing various items, or whether to increase or decrease production according to seasonal fluctuations.)
* Any modern data warehouse solution must be able to provide access to the streams of raw data, and the cooked business information derived from this data.
  1. **Azure Data Factory**:
  + Azure Data Factory is described as a data **integration** service. The purpose of Azure Data Factory is to retrieve data from one or more data sources, and convert it into a format that you process.
  + Azure Data Factory enables you to extract the interesting data and discard the rest.
  + Azure Data Factory can then write the ingested data to a data store for subsequent processing.
  + You define the work performed by Azure Data Factory as a pipeline of operations. A pipeline can run continuously, as data is received from the various data sources.
  + You can create pipelines using the graphical user interface provided by Microsoft, or by writing your own code
  + Using Azure Data Factory, you can create and schedule data-driven workflows (called pipelines) that can ingest data from the disparate data stores.
  + You can build complex ETL processes that transform data visually with data flows or by using compute services such as Azure HDInsight, Azure Databricks, and Azure SQL Database. You can then publish the transformed data to Azure Synapse Analytics for business intelligence applications to consume.
  1. **Azure Data Lake Storage**:
  + Azure Data Lake is a collection of analytics and storage services that you can combine to implement a big data solution. t comprises three main elements:
    - Data Lake Store
    - Data Lake Analytics
    - HDInsight
  + **Data Lake Store** provides a file system that can store near limitless quantities of data. It uses a hierarchical organization (like the Windows and Linux file systems), but you can hold massive amounts of raw data (blobs) and structured data. It is optimized for analytics workload.
  + Azure Data Lake Store is compatible with the Hadoop Distributed File System (HDFS). You can run Hadoop jobs using Azure HDInsight (see below) that can read and write data in Data Lake Store efficiently.
  + Azure Data Lake Store provides granular security over data, using Access Control Lists. An Access Control List specifies which accounts can access which files and folders in the store.
  + **Azure Data Lake Analytics** is an on-demand analytics job service that you can use to process big data. It provides a framework and set of tools that you use to analyze data held in Microsoft Azure Data Lake Store, and other repositories. You write jobs that contain queries to transform data and extract insights.
  + You define a job using a language called **U-SQL**. This is a hybrid language that takes **features from both SQL and C#,** and provides **declarative and procedural capabilities** that you can use to process data
  + You define a job using a language called U-SQL. This is a hybrid language that takes features from both SQL and C#, and provides declarative and procedural capabilities that you can use to process data
  + A data lake is a **repository** for large quantities of raw data. Because the data is raw and unprocessed, it's very fast to load and update, but the data hasn't been put into a structure suitable for efficient analysis.
  + You can think of a data lake as a staging point for your ingested data, before it's massaged and converted into a format suitable for performing analytics.
  + A data lake holds raw data, but a data warehouse holds structured information.
  + Azure Data Lake Storage combines the hierarchical directory structure and file system semantics of a traditional file system with security and scalability provided by Azure. Azure Data Lake Storage is essentially an extension of Azure Blob storage, organized as a near-infinite file system.
  + It has the following characteristics:
    - Data Lake Storage organizes your files into directories and subdirectories for improved file organization. Blob storage can only mimic a directory structure.
    - Data Lake Storage supports the Portable Operating System Interface (POSIX) file and directory permissions to enable granular Role-Based Access Control (RBAC) on your data.
    - Azure Data Lake Storage is compatible with the Hadoop Distributed File System (HDFS). Hadoop is highly flexible and programmable analysis service, used by many organizations to examine large quantities of data. All Apache Hadoop environments can access data in Azure Data Lake Storage Gen2



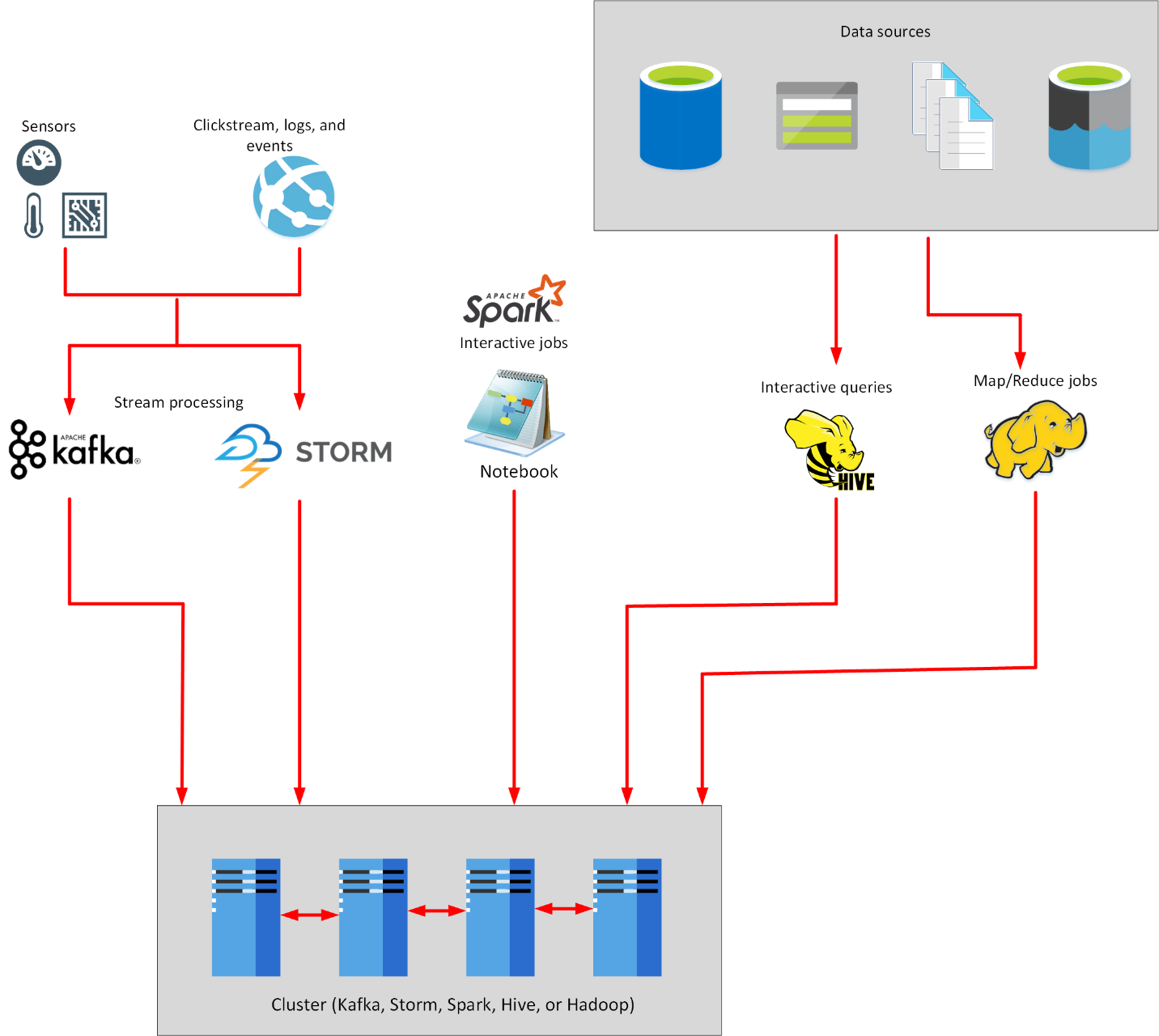
* 1. **Azure Databricks**:
  + Azure Databricks is an analytics platform optimized for the Microsoft Azure cloud services platform
  + Designed with the founders of Apache Spark, Databricks is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.
  + Databricks can process data held in many different types of storage, including Azure Blob storage, Azure Data Lake Store, Hadoop storage, flat files, databases, and data warehouses. Databricks can also process streaming data. Databricks uses an extensible architecture based on drivers.
  + A driver is a piece of code that connects to a specific data source and enables you to read and write that source. A driver is typically provided as part of a library that you can load into the Databricks environment.
  + Azure Databricks is an **Apache Spark environment running on Azure** to provide big data processing, streaming, and **machine learning**.
  + Apache Spark is a parallel-processing engine that supports large-scale analytics. It is a highly efficient data processing engine that can consume and process large amounts of data very quickly.
  + There are a significant number of Spark libraries you can use to perform tasks such as SQL processing, aggregations, and to build and train machine learning models using your data.These libraries include modules for machine learning, statistical analysis, linear and non-linear modeling, predictive analytics, and graphics.
  + Azure Databricks provides a graphical user interface where you can define and test your processing step by step, before submitting it as a set of batch tasks
  + You can create Databricks scripts and query data using languages such as R, Python, and **Scala**. You write your Spark code using notebooks.The first line in the cell is %language. For example, %scala
  + Azure Databricks also supports structured stream processing. In this model, Databricks performs your computations incrementally, and continuously updates the result as streaming data arrives.
  1. **Azure Synapse Analytics**:
  + Azure Synapse Analytics is an analytics engine. It's designed to process large amounts of data very quickly.
  + You can use it to read data from many sources, process this data, generate various analyses and models, and save the results.
  + Azure Synapse Analytics is particularly suitable for ELT approach
  + Using Synapse Analytics, you can **ingest** data from external sources, such as flat files, Azure Data Lake, or other database management systems, and then **transform** and aggregate this data into a format suitable for analytics processing. You can perform complex queries over this data and **generate reports, graphs, and charts.**
  + Azure Synapse Analytics enables you to store the data you have read in and processed locally, within the service. This approach enables you to repeatedly query the same data without the overhead of fetching and converting it each time. You can also use this data as **input to further analytical processing**, using Azure Analysis Services
  + Azure Synapse Analytics leverages a **Massively parallel processing (MPP)** architecture. This architecture includes a **control node** and a **pool of compute nodes**
  + The **Control node** is the brain of the architecture. It's the front end that interacts with all applications. When you submit a processing request, the Control node transforms it into smaller requests that run against distinct subsets of the data in parallel.
  + The **Compute nodes** provide the computational power. The data to be processed is distributed evenly across the nodes.
  + Users and applications send processing requests to the control node. The control node sends the queries to compute nodes, which run the queries over the portion of the data that they each hold. When each node has finished its processing, the results are sent back to the control node where they're combined into an overall result.
  + Azure Synapse is composed of the following elements:
    - **Synapse SQL pool**: This is a collection of servers running Transact-SQL. Transact-SQL is the dialect of SQL used by Azure SQL Database, and Microsoft SQL Server. You write your data processing logic using Transact-SQL.
    - **Synapse Spark pool**: This is a cluster of servers running Apache Spark to process data. You write your data processing logic using one of the four supported languages: Python, Scala, SQL, and C# (via .NET for Apache Spark). Spark pools support Azure Machine Learning through integration with the SparkML and AzureML packages.
    - **Synapse Pipelines**: A Synapse pipeline is a logical grouping of activities that together perform a task. The activities in a pipeline define actions to perform on your data.
      * For example, you might use a copy activity to transform data from a source dataset to a destination dataset.
      * You could include activities that transform the data as it is transferred, or you might combine data from multiple sources together.
      * The pipeline allows you to manage the activities as a set instead of each one individually.
      * Synapse pipelines use the same Data Integration engine used by Azure Data Factory. This gives you the power in Synapse Studio to create pipelines that can connect to over 90 sources from flat files, databases, or online services

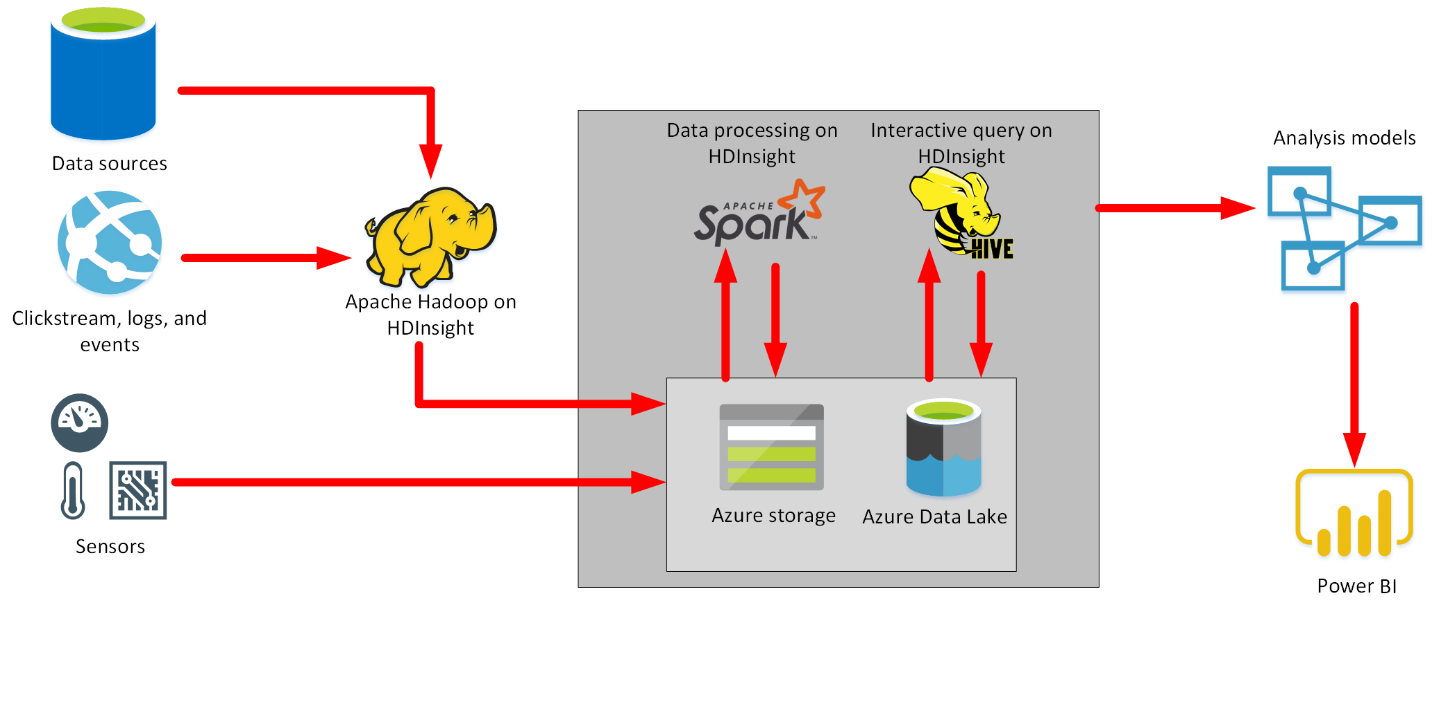


* + - **Synapse Link**: This component allows you to connect to Cosmos DB. You can use it to perform near real-time analytics over the operational data stored in a Cosmos DB database.
      * It is a cloud-native hybrid transactional and analytical processing (HTAP) capability that enables you to run near real-time analytics over operational data stored in Azure Cosmos DB.
      * Synapse link uses a feature of Cosmos DB named Cosmos DB Analytical Store. Cosmos DB Analytical Store contains a copy of the data in a Cosmos DB container, but organized as a column store.
      * Column stores group data by column rather than by row. Column stores are a more optimal format for running analytical workloads that need to aggregate data down a column rather than across a row, such as generating sum totals, averages, maximum or minimum values for a column
      * Cosmos DB automatically keeps the data in its containers synchronized with the copies in the column store.
      * The data doesn't have to go through an ETL (extract, transform, and load) process because the data isn't copied into Synapse Analytics; it remains in the Cosmos DB analytical store.
      * Synapse link has a wide range of uses, including:
        + **Supply chain analytics and forecasting**. You can query operational data directly and use it to build machine learning models. You can use the results generated by these models back into Cosmos DB for near-real-time scoring. You can use these assessments to successively refine the models and generate more accurate forecasts.
        + **Operational reporting**. You can use Synapse Analytics to query operational data using Transact-SQL running in a SQL pool. You can publish the results to dashboards using the support provided to familiar tools such as Microsoft Power BI.
        + **Batch data integration and orchestration**. With supply chains getting more complex, supply chain data platforms need to integrate with a variety of data sources and formats. The Azure Synapse data integration engine allows data engineers to create rich data pipelines without requiring a separate orchestration engine.
        + **Real-time personalization**. You can build engaging ecommerce solutions that allow retailers to generate personalized recommendations and special offers for customers in real time.
        + **IoT maintenance**. Industrial IoT innovations have drastically reduced downtimes of machinery and increased overall efficiency across all fields of industry. One such innovation is predictive maintenance analytics for machinery at the edge of the cloud. The historical operational data from IoT device sensors could be used to train predictive models such as anomaly detectors. These anomaly detectors are then deployed back to the edge for real-time monitoring. Looping back allows for continuous retraining of the predictive models.
    - **Synapse Studio**: This is a web user interface that enables data engineers to access all the Synapse Analytics tools. You can use Synapse Studio to create SQL and Spark pools, define and run pipelines, and configure links to external data sources.
      * With Synapse Studio you can **develop**, test, and debug Spark notebooks and Transact-SQL jobs.
      * You can **monitor** the performance of operations that are currently running, and you can manage the serverless or provisioned resources.
      * All of these capabilities are accessed via the web-native Synapse Studio that allows for model **management**, monitoring, coding, and **security**.



* + Azure Synapse Analytics supports two computational models:
    - **SQL (Transact-SQL) pools**:
      * each compute node uses an Azure SQL Database and Azure Storage to handle a portion of the data.
      * You submit queries in the form of Transact-SQL statements, and Azure Synapse Analytics runs them. However, unlike an ordinary SQL Server database engine, Azure Synapse Analytics can receive data from a wide variety of sources.
      * To do this, Azure Synapse Analytics uses a technology named **PolyBase**. PolyBase is used to make external data look like SQL tables. PolyBase enables you to retrieve data from relational and non-relational sources.
      * You can save the data read in as SQL tables within the Synapse Analytics service.
      * You specify the number of nodes when you create a SQL pool. You can scale the SQL pool manually to add or remove compute nodes as necessary. You can **only scale a SQL pool when it's not running** a Transact-SQL query
      * The data is split into chunks called distributions. A distribution is the basic unit of storage and processing for parallel queries that run on distributed data. Each of the smaller queries runs on one of the data distributions
      * By default, an on-demand SQL pool is created in each Azure Synapse Analytics workspace. You can then provision additional pools, either on-demand or provisioned.
      * On-demand pools only allow you to query data held in external files. If you want to ingest and load the data into Synapse Analytics, you must create your own SQL pool.
      * Use SQL pools in Synapse Analytics for the following scenarios:
        + **Complex reporting**. You can use the full power of Transact-SQL to run complex SQL statements that summarize and aggregate data.
        + **Data ingestion**. PolyBase enables you to retrieve data from many external sources and convert it into a tabular format. You can reformat this data and save it as tables and materialized views in Azure Synapse.
    - **Spark pools**:
      * The nodes are replaced with a Spark cluster.
      * You run Spark jobs comprising code written in Notebooks, in the same way as Azure Databricks. You can write the code for notebook in C#, Python, **Scala, or Spark SQL** (a different dialect of SQL from Transact-SQL)
      * You can save data generated by your notebooks in Azure Storage or Data Lake Storage.
      * **Spark is optimized for in-memory processing**. A Spark job can load and cache data into memory and query it repeatedly. In-memory computing is much faster than disk-based applications, but requires additional memory resources.
      * Autoscaling can occur while processing is active
      * Spark pools enable you to process data held in many formats, such as csv, json, xml, parquet, orc, and avro. Spark can be extended to support many more formats with external data sources
      * Spark pools provide the basic building blocks for performing in-memory cluster computing
      * Spark pools in Azure Synapse are compatible with Azure Storage and Azure Data Lake Storage
      * Spark pools in Synapse Analytics are especially suitable for the following scenarios:
        + **Data Engineering/Data Preparation**. Apache Spark includes many language features to support preparation and processing of large volumes of data so that it can be made more valuable and then consumed by other services within Synapse Analytics. This is enabled through the Spark libraries that support processing and connectivity.
        + **Machine Learning**. Apache Spark comes with MLlib, a machine learning library built on top of Spark that you can use from a Spark pool in Synapse Analytics. Spark pools in Synapse Analytics also include Anaconda, a Python distribution with a variety of packages for data science including machine learning. When combined with built-in support for notebooks, you have an environment for creating machine learning applications.
  + Spark pools and SQL pools can coexist in the same Azure Synapse Analytics instance.
  + Azure Synapse Analytics can consume a lot of resources. If you aren't planning on performing any processing for a while, you can pause the service. This action releases the resources in the pool to other users, and reduces your costs.
  + **Use** Azure Synapse Analytics for:
    - Very high volumes of data (**multi-terabyte to petabyte** sized datasets).
    - Very complex queries and aggregations.
    - Data mining, and data exploration.
    - Complex ETL operations. ETL stands for Extract, Transform, and Load, and refers to the way in which you can retrieve raw data from multiple sources, convert this data into a standard format, and store it.
    - Low to mid concurrency (**128 users or fewer**).
  + Using Apache Spark, and automated pipelines, Synapse Analytics can run parallel processing tasks across massive datasets, and perform big data analytics.
  + Option to analyze operational data in its original location is known as hybrid transactional analytical processing (**HTAP**).
  + Any data stored in Azure Synapse Analytics can be used to build and train models with Azure Machine Learning
  1. **Azure Analysis Services**:
  + Azure Analysis Services enables you **to build tabular models** to support **online analytical processing (OLAP)** queries.
  + You can combine data from multiple sources, including Azure SQL Database, Azure Synapse Analytics, Azure Data Lake store, Azure Cosmos DB, and many others.
  + You use these data sources to build models that incorporate your business knowledge. A **model is essentially a set of queries and expressions** that retrieve data from the various data sources and generate results.
  + Analysis Services includes a **graphical designer** to help you connect data sources together and define queries that **combine, filter, and aggregate data**.
  + Azure Analysis Services has significant functional overlap with Azure Synapse Analytics, but it's more suited **for processing on a smaller scale**
  + **Use** Azure Analysis Services for:
    - Smaller volumes of data (a **few terabytes**).
    - Multiple sources that can be correlated.
    - High read concurrency (**thousands of users**).
    - Detailed analysis, and drilling into data, using functions in Power BI.
    - Rapid dashboard development from tabular data.
  1. **Synapse Analytics and Analysis Services together**:
  + If you have large amounts of ingested data that require preprocessing, you can use Synapse Analytics to read this data and manipulate it into a model that contains business information.
  + The scalability of Synapse Analytics gives it the ability to process and **reduce many terabytes of data down into a smaller**, succinct dataset that summarizes and aggregates much of this data.
  + You can then use **Analysis Services** to perform **detailed interrogation of this information,** and visualize the results of these inquiries with Power BI.
  1. **Azure HDInsight**:
  + Azure HDInsight is a managed analytics service in the cloud. It's based on Apache Hadoop, a collection of open-source tools and utilities that enable you to run processing tasks over large amounts of data. HDInsight uses a clustered model, similar to that of Synapse Analytics
  + Azure HDInsight is a big data processing service, that provides the platform for technologies such as Spark in an Azure environment.
  + HDInsight stores data using Azure Data Lake storage.
  + ou can use HDInsight to analyze data using frameworks such as Hadoop Map/Reduce, Apache Spark, Apache Hive, Apache Kafka, Apache Storm, R, and more.
    - Map/Reduce jobs and Spark jobs are parallelized into a series of subtasks tasks that run on the cluster
    - Apache Hive provides interactive SQL-like facilities for querying, aggregating, and summarizing data
    - Apache Kafka is a clustered streaming service that can ingest data in real time. It's a highly scalable solution that offers publish and subscribe features
    - Apache Storm is a scalable, fault tolerant platform for running real-time data processing applications. Storm is designed for reliability, so that events shouldn't be lost. Storm solutions can also provide guaranteed processing of data, with the ability to replay data that wasn't successfully processed the first time. Storm can interoperate with a variety of event sources, including Azure Event Hubs, Azure IoT Hub, Apache Kafka, and RabbitMQ (a message queuing service). Storm can also write to data stores such as HDFS, Hive, HBase, Redis, and SQL databases.



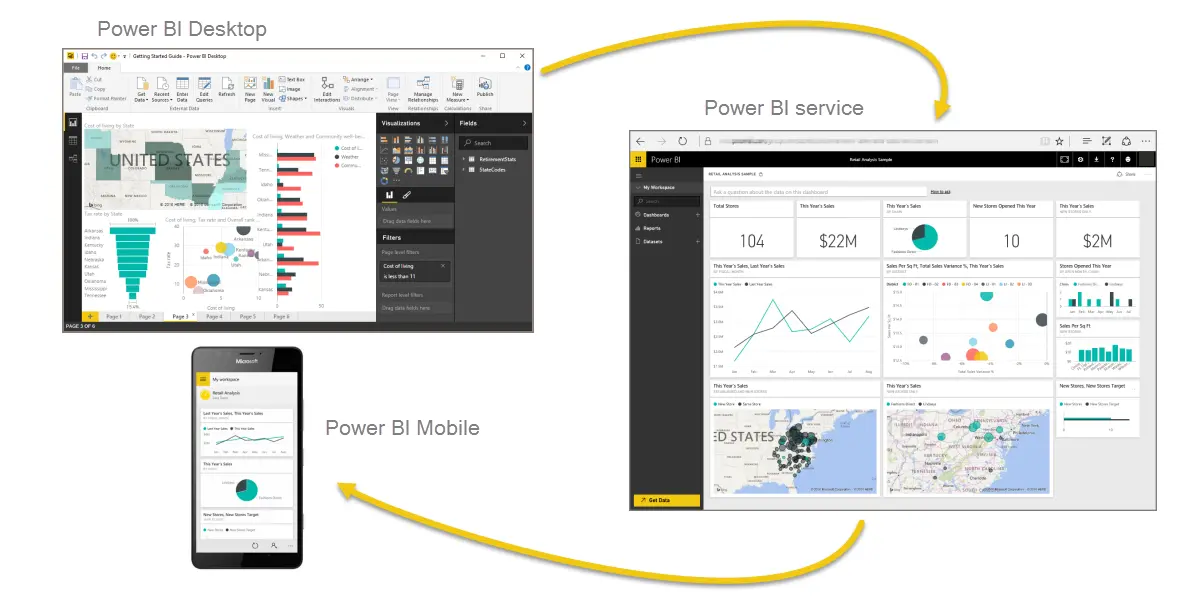
* + HDInsight implements a clustered model that distributes processing across a set of computers. This model is similar to that used by Synapse Analytics, except that the nodes are running the Spark processing engine rather than Azure SQL Database.
  + You can use Azure HDInsight in conjunction with, or instead of, Azure Synapse Analytics
  + Along with Spark, HDInsight supports **clusteredstreaming technologies such as Apache Kafka, Apache Storm and the Apache Hadoop** processing model.
  + The image below shows where you might use the components of HDInsight in a data warehousing solution.
  + **Hadoop** is an open-source framework that breaks large data processing problems down into smaller chunks and distributes them across a cluster of servers, similar to the way in which Synapse Analytics operates.
  + **Hive** is a SQL-like **query facility** that you can use with an HDInsight cluster to examine data held in a variety of formats. You can use it to create, load, and query external tables, in a manner similar to PolyBase for Azure Synapse Analytics

1. **Explore data ingestion in Azure**:Azure provides several services you can use to ingest data. These services can operate with almost any source: **Azure Data Factory**, **PolyBase**, **SQL Server Integration Services (SSIS)**, and **Azure Databricks**.

* **Ingest data using Azure Data Factory**: Azure Data Factory is a data ingestion and transformation service that allows you to load raw data from many different sources, both on-premises and in the cloud. As it ingests the data, Data Factory can clean, transform, and restructure the data, before loading it into a repository such as a data warehouse.
  + Data Factory provides an orchestration engine. **Orchestration** is the process of directing and controlling other services, and connecting them together, to allow data to flow between them. Data Factory uses orchestration to combine and automate sequences of tasks that use different services to perform complex operations.
  + Data Factory moves data from a data source to a destination.
  + Azure Data Factory uses a number of different resources: linked services, datasets, and pipelines.
    - **linked services**:A linked service provides the information needed for Data Factory to connect to a source or destination
    - **datasets**:A dataset in Azure Data Factory represents the data that you want to ingest (input) or store (output)
    - **pipelines**: A pipeline is a logical grouping of activities that together perform a task (unit of work).
      * The activities in a pipeline define actions to perform on your data.
      * Sometimes when ingesting data, the data you're bringing in can have different column names and data types to those required by the output. In these cases, you can use a mapping to transform your data from the input format to the output format.
      * You can run a pipeline manually, or you can arrange for it to be run later using a trigger (schedule a pipeline or when an event occurs such as the arrival/deletion of a file in Azure Data Lake Storage)
* **Ingest data using PolyBase**: PolyBase is a feature of SQL Server and Azure Synapse Analytics that enables you to run Transact-SQL queries that read data from external data sources (like Hadoop, Spark, and Azure Blob Storage, as well as other database management systems such as Cosmos DB, Oracle, Teradata, and MongoDB).
  + Spark is a parallel-processing engine that supports large-scale analytics
  + PolyBase enables you to transfer data from an external data source into a table, as well as in Azure Synapse Analytics or SQL Server
  + You can also run queries that join tables in a SQL database with external data, enabling you to perform analytics that span multiple data stores
  + Azure Data Factory provides PolyBase support for loading data. For instance, Data Factory can directly invoke PolyBase on your behalf if your data is in a PolyBase-compatible data store
  + Azure SQL Database does not support PolyBase.
* **Ingest data using SQL Server Integration Services (SSIS)**: SQL Server Integration Services (SSIS) is a platform for building enterprise-level data integration and data transformations solutions.
  + You can use SSIS to solve complex business problems by copying or downloading files, loading data warehouses, cleaning and mining data, and managing SQL database objects and data.
  + SSIS is part of Microsoft SQL Server
  + SSIS can extract and transform data from a wide variety of sources such as XML data files, flat files, and relational data sources, and then load the data into one or more destinations.
  + A package is an organized collection of connections, control flow elements, data flow elements, event handlers, variables, parameters, and configurations, that you assemble using either the graphical design tools that SQL Server Integration Services provides, or build programmatically.
  + SSIS is an on-premises utility. However, Azure Data factory allows you to run your existing SSIS packages as part of a pipeline in the cloud. This allows you to get started quickly without having to rewrite your existing transformation logic.
  + The SSIS Feature Pack for Azure is an extension that provides components that connect to Azure services, transfer data between Azure and on-premises data sources, and process data stored in Azure. The components in the feature pack support transfer to or from Azure storage, Azure Data Lake, and Azure HDInsight. Using these components, you can perform large-scale processing of ingested data.
* **Ingest data using Azure Databricks**:Azure Databricks is an analytics platform optimized for the Microsoft Azure cloud services platform. Databricks is based on Spark and is integrated with Azure to streamline workflows.
  + It provides an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts
  + Databricks can process data held in many different types of storage, including Azure Blob storage, Azure Data Lake Store, Hadoop storage, flat files, SQL databases, and data warehouses, and Azure services such as Cosmos DB. Databricks can also process streaming data.
  + You write and run Spark code using notebooks. A notebook can contain cells that read data from one or more data sources, process the data, and write the results out to a data store.
  + The scalability of Azure Databricks makes it an ideal platform for performing complex data ingestion and analytics tasks.
  + Azure Data Factory can incorporate Azure Databricks notebooks into a pipeline. A pipeline can pass parameters to a notebook. These parameters can specify which data to read and analyze. The notebook can save the results, which the Azure Data Factory pipeline can use in subsequent activities.

1. **Get started building with Power BI**:Microsoft Power BI is a collection of software services, apps, and connectors that work together to turn your data into interactive insights.

* You can use data from single basic sources, like a Microsoft Excel workbook, or pull in data from multiple databases and cloud sources to create complex datasets and reports.
* Power BI can be as straightforward as you want or as enterprise-ready as your complex global business requires.
* Power BI consists of three main elements—Power BI Desktop, the Power BI service, and Power BI Mobile—which work together to let you create, interact with, share, and consume your data the way you want.



* Basic building blocks in Power BI:
  + Visualizations – Visual representations of data, sometimes just called visuals
  + Datasets – Collections of data that Power BI uses to create visualizations
  + Reports – Collections of visuals from a dataset, spanning one or more pages
  + Dashboards – Single-page collections of visuals built from reports
  + Tiles – Single visualizations on reports or dashboards
* That app, a ready-made collection of visuals and reports, let us easily connect to a software service to populate the app and bring that data to life. App can be installed in just a few clicks.
* We can set up a refresh schedule for our data, so that we know the data will be fresh when we go back to the Power BI service.
* The common flow of work in Power BI:
  + Bring data into Power BI Desktop and create a report.
  + Publish to the Power BI service, where you create new visualizations or build dashboards.
  + Share your dashboards with others, especially people who are on the go.
  + View and interact with shared dashboards and reports in Power BI Mobile apps.