DP900 NOTES

I. DATA CONCEPTS

1. Azure Core Data Related Services

- **Azure Storage Accounts**: umbrella for various storage (tables, files, blob)
- **Azure Blob storage**: data stored as objects instead of files (multiples machines – unstructured data)
- **Azure Tables:** key/value NoSQL data store (for simpler project)
- **Azure Files:** managed file-shared NFS or SMB
- **Azure Storage Explorer**: App used to explore data inside *Azure Storage Accounts*
- **Azure Synapse Analytics**: Data warehouse and unified analytics platform
- **CosmoDB**: fully managed NoSQL db service -> various NoSQL engines (Tables, Doc, Key/value, Graph…)
- **Azure Data Lake Store (Gen2)**: centralized data repo for big data (bob storage used for large amount of data)
- **Azure Data Analytics:** Big Data as a Service (BDaS) write U-SQL to return data from Azure Data Lake
- **Azure Data Box:** Import or export TB of data via hard drive -> mail it into Azure datacenters
- **SQL Server on Azure VM:** while migrating via lift & shift, you want to bring ur existing license and need OS access along with control of VM.
- **SQL Managed Instances:** managed MS SQL server> broad adaptably when migrating to Azure
- **Azure SQL:** Fully managed MS SQL db
- **Azure Db for <Open-Source>:** managed relational db for Azure (eg. MySQL, PSQL, MariaDB)
- **Azure Cache for REDIS:** In-Memory data-store for returning data extremely vast & volatile
- **Microsoft Office 365 Share Point:** shared file system for organizations. (company owns all the files & fine grain role-based access-controls)
- **Azure Databricks:** Third-party provider specialize in Apache Spark -> fast ELT jobs/ML/Stream
- **Microsoft Power BI:** Business/8534 Intelligence tool used to create dashboards & interactive reports to empower business decisions
- **HDInsights:** fully managed Hadoop System, can run many open sources Big Data engines -> used for data transformations for Streaming, ETL/ELT.
- **Azure Data Studio:** IDE like VSCODE, designed around data related tasks (cross platform similar to SSIS but broader data workloads)
- **Azure Data Factory:** Managed ETL/ELT pipeline builder. Easy build transformation pipelines via a web-interface
- **SQL Server Integration Services (SSIS):** stand-alone Windows app to prepare data for SQL workloads via transformation pipelines

2. Types of Cloud Computing

- **« SaaS » - Software as a Service (for Customers)**

Product that is run and managed by the service provider – **Don't worry about how the service is maintained. It just works and remains available**. (eg. PowerBI & Office 365)

- **« PaaS » - Platform as a Service (for Devs)**

Focus on the deployment and management of your apps. **Don't worry about provisioning, configuring or understanding the hardware or OS**. (eg. HDInsights, Managed SQL, Azure SQL, CosmoDB)

- **« IaaS » - Infrastructure as a Service (for Admins)**

The basic building blocks for cloud IT. Provides access to networking features, computers and data storage space. **Don't worry about IT staff, data centers and hardware**.

3. Azure Data Related Roles

- **Database Administrator: configure and maintains a database (eg. Azure Data services/SQL)**

Responsibilities	**Common Tools**
- Database management - Manage security, granting user access - Backups - Monitors Performance	- Azure Data Studio - SQL Server Management Studio - Azure Portal - Azure CLI

- **Data Engineer: Design & implement data tasks** related to the transfer and storage of **Big Data**

Responsibilities	**Common Tools**
- Database pipelines & process - Data ingestion storage - Prepare data analytics - Prepare data for analytical processing	- Azure Synapse Studio - SQL - Azure CLI

- **Data Analyst: Analyze business data** to reveal important information

Responsibilities	**Common Tools**
- Provides insights into the data - Visual reporting - Modeling data for analysis - Combines data for Viz & Analysis	- Power BI Desktop - Power BI Portal - Power BI services - Power BI report builder

4. Database Administrator Common tools

- **Azure Data Studio**

Connect to Azure SQL, Azure SQL data warehouse, Postgres SQL & SQL Server (BD clusters, on-premises)
- Various libraries and extensions along with automation tools
- Graphical interface for managing on-premises and cloud-based data services
- Runs on Windows, macOS, Linux
- Possibly a replacement for SSMS (still lacks some features of SSMS)

- **SQL Server Management Studio (SSMS)**

- Automation tooling for running SQL commands or common db operations
- Graphical interface for managing on-premises and cloud-based data services
- Runs on Windows
- More mature than Azure Data Studio

- **Azure Portal and CLI**

- Manage SQL db configurations. Eg. Create, deleting, resizing, number of cores
- Manage and provision other Azure Data Services
- Automate the creating, updating or modifying resources via Azure Resource Manager templates (IaC)

5. Data Engineering Common tools

- **Azure Synapse Studio**
  Azure portal integrated to manage Azure Synapse, data ingestion (Azure data factory), management of azure synapse assets (SQL Pools/Spark Pool)

- **Knowledge SQL** – Create databases, tables, views, etc

- **Azure CLI** – Support operations SQL cmd to connect to Microsoft server Azure SQL data & run a talk queries/command.

- **HDInsights** – Streaming data via Apache Kafka or Apache Spark. Applying ELT jobs via HIVE, PIG, A Spark

- **Azure Databricks** – Using Apache Spark to create ELT or streaming jobs to data ware house/lakes

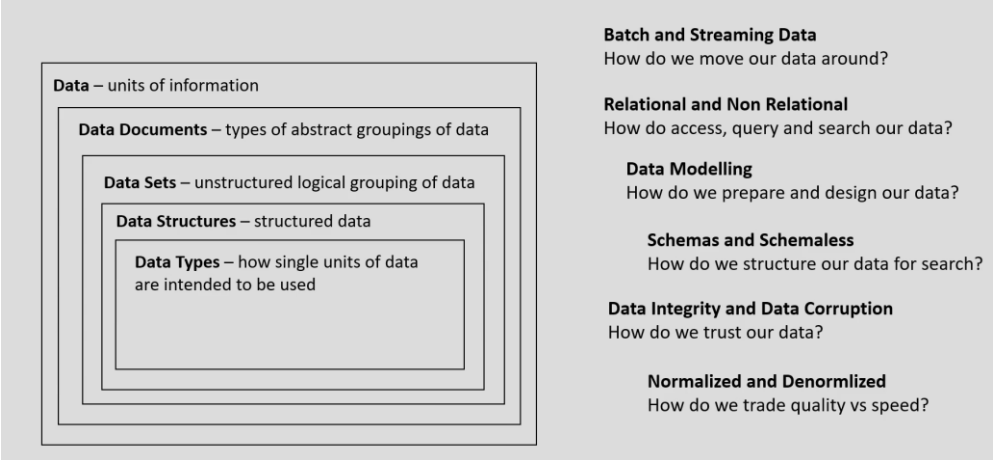
6. Data Analyst Common tools

- **Power BI Desktop**
  A stand-alone application for data visualization. You can do data modelling, connect to many data sources and create interactive reports

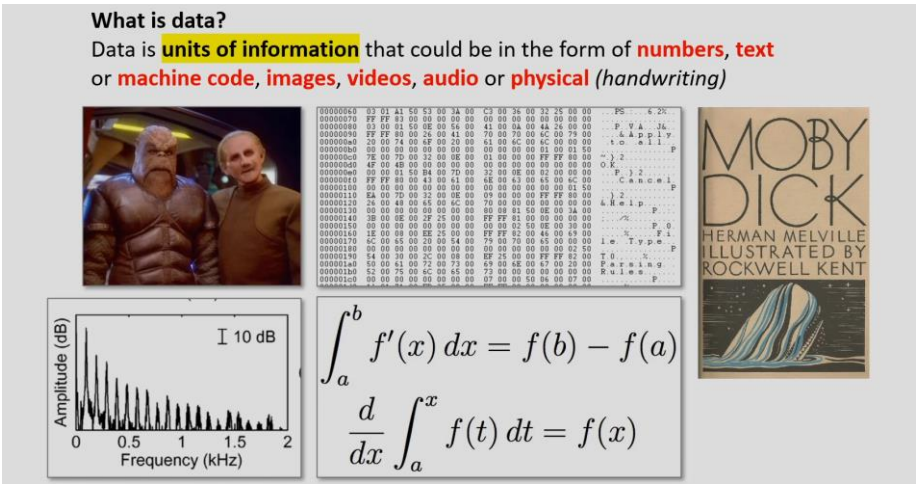
- **Power BI Portal/Power BI Services** – A web UI for creating interactive dashboards

- **Power BI Report Builder** – Create paginated reports (printable reports)

7. Data Overview



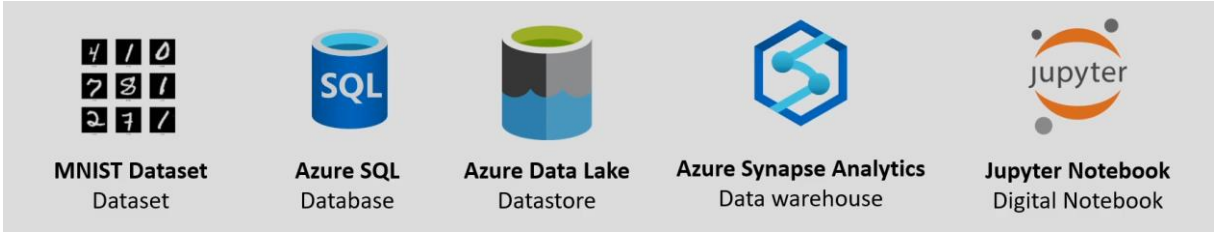
8. Introduction to data



9. Data Documents

A data document defines the **collective form in which data exists**.
Here's the common types of data documents:

- **Datasets** – a logical grouping of data
- **Databases** – structured data that can be quickly access and searched
- **Datastores** – unstructured or semi-structured data to housing data
- **Data Warehouses** – structured or semi-structured data for creating reports and analytics
- **Notebooks** – data that is arranged in pages, designed for easy consumption



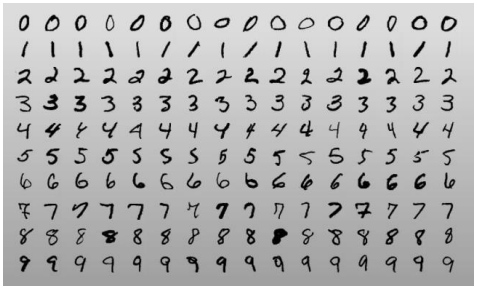
10. Data sets

A data set is **a logical grouping of unit of data** that generally are closely related and/or share the same data structure.

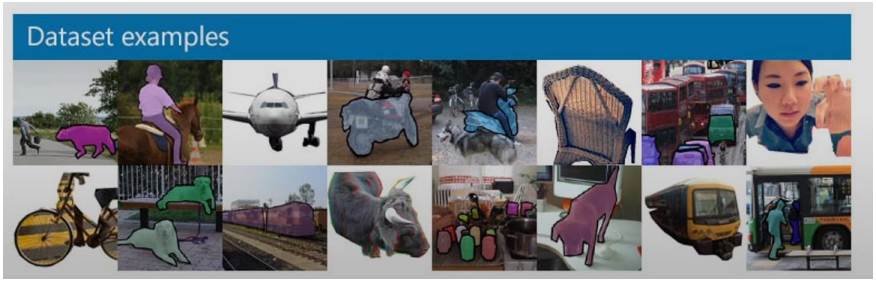
There are **publicly available data sets** that are used in the **learning of statistics, data analytics, machine learning**.

Examples of some datasets

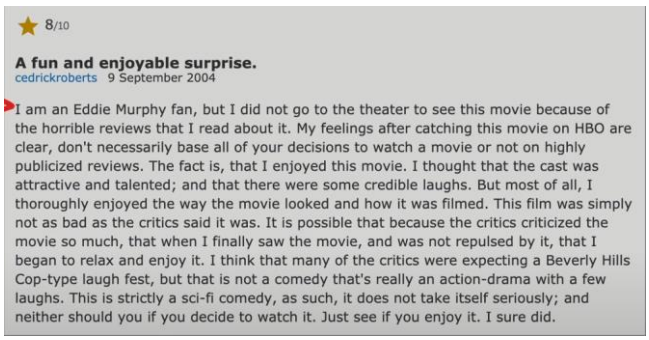
- **MNIST dataset**: Images of **handwritten digits** used to test classification, clustering, and image processing algorithm. *Commonly used when learning how to build computer vision ML models to translate handwriting into digital text.*



- **Common Objects In Context (*COCO*) dataset**: A dataset which contains many common images using a JSON file (coco format) that identify objects or segments within an image. *This dataset features: Object segmentation, Recognition in context, Superpixel stuff segmentation, 329K Images (>200K labeled), 0.5 million object instances, 79 object categories, 90 stuff categories, 4 captions per image, 249,000 people with key points.*



- **IMD Reviews Dataset:** A **movie review** dataset with 25 000 highly polar movie reviews for training, and 25 000 for testing. Could be useful to determine customer sentiment analysis.



8/10

A fun and enjoyable surprise.

cedrickroberts 9 September 2004

I am an Eddie Murphy fan, but I did not go to the theater to see this movie because of the horrible reviews that I read about it. My feelings after catching this movie on HBO are clear, don't necessarily base all of your decisions to watch a movie or not on highly publicized reviews. The fact is, that I enjoyed this movie. I thought that the cast was attractive and talented; and that there were some credible laughs. But most of all, I thoroughly enjoyed the way the movie looked and how it was filmed. This film was simply not as bad as the critics said it was. It is possible that because the critics criticized the movie so much, that when I finally saw the movie, and was not repulsed by it, that I began to relax and enjoy it. I think that many of the critics were expecting a Beverly Hills Cop-type laugh fest, but that is not a comedy that's really an action-drama with a few laughs. This is strictly a sci-fi comedy, as such, it does not take itself seriously; and neither should you if you decide to watch it. Just see if you enjoy it. I sure did.

11. Data types

A data type is **a single unit of data** that tells a compiler or interpreter (computer program) **how data is intended to be used**.

The **variety** of data types will **greatly vary based on the computer program**.

Numeric Data Types: A data type involving mathematical **numbers.**

- **Integer** – a whole number, (could be negative or positive): -100, 7, 11, 219185354…
- **Float** – a number that has a decimal e.g: 1.5, 0.0, -10.24, 9.435614481…

```
my_int = 1
my_float = 2.2
```

Text Data Types: A data type that contains readable and non-readable **letters**.

- **Character** – a single letter, alphanumeric (A-Z), digit (0-9), blank space, punction, special characters (%$&@...)
- **String** – a sequence of characters e.g: Words, sentences and paragraphs.

```
my_char = 'a'
my_string = "We prefer to help ourselves"
```

Composite: A data type that contains **cells of data** that can be **accessed via an index or a key**.

- **Array** – a group of elements that contain the same data type, can be accessed via their index (position)
- **Dictionary (Hash)** – a group of elements where a key can be used retrieve a value
(*Composites can be both data-types and data structures*)

```
my_arr = ['live','long','and','prosper']
my_dict = { "Speed": 1, "Accuracy": 2 }
```

Binary Data Type – represented by a **bit or a series of bits (a byte)**, Which is either 0 (off) or 1 (on)

```
one_byte = int('11110000', 2)
```

Boolean Data Type – A datatype that is either **True or False**

- Some languages represent a Boolean as:
  - A bit as a Boolean eg. 0 (false) or 1 (true)
  - The first letter e.g: f (false) or t (true)
  - The whole word (python): True or False

```
my_bool = True
```

Enumeration (Enum) Data Type – a group of constant (unchangeable) variables e.g: DIAMOND, SPADE, HEART, CLUBS

- Can be data type and/or data structure, varies on the language

```
class Shake(Enum):
  VANILLA = 7
  CHOCOLATE = 4
  COOKIES = 9
  MINT = 3

Shake.VANILLA
Shake.CHOCOLATE
Shake.COOKIES
Shake.MINT
```

12. Schema vs Schemaless

What is a Schema?

A schema (in terms of databases) is **a formal language which describes the structure of data** (blueprint) of a database. A schema can **define many different data structures** that serve different purposes for a database.

Different data structures (relational databases):

- Tables - Fields - Relationships - Views - Indexes - Packages - Procedures - Functions - XML schemas	- Queues - Triggers - Types - Sequences - Materialized views - Synonyms - Database links - Directories

What is Schemaless?

Schemaless is when the **primary "cell" of database can accept many types**. This allow developers to **forgo upfront data modelling**.

Common Schemaless databases are:

- Key/Value
- Document
- Columns
    - Wide Column
- Graph

13. Query and Querying

What is a query? – A query is a request for data results (reads) or to perform operations such as inserting, updating, deleting data (writes). A query can perform maintenance operations on the data and is not always restricted to just working with the data that resides within the database.

```
SELECT * FROM
CrewMembers
```

What are data Results? – Results are the data returned from a query
What is querying? – That is the act of performing a query

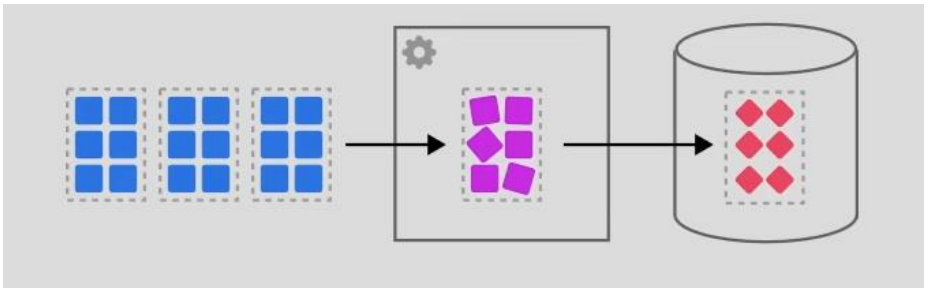
What is query language? – A scripting or programming language designed as the format to submit a request or actions to the database.
Notable query languages:

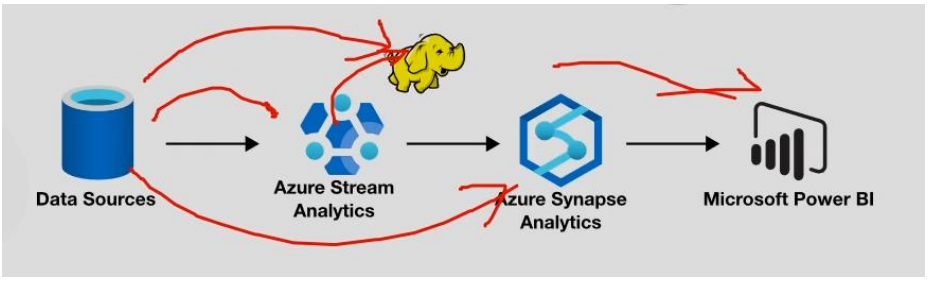
- **SQL**
- GraphSQL
- Kusto
- Xpath
- Gremlin

Id	Name	Title	StarDate
10	Picard	Captain	41133.7
11	Riker	Commander	41154.2
12	Data	Lt Commander	41114.3
13	Troi	Lt Commander	41412.1
14	Crusher	Command	41520.4

14. Batch vs Stream processing

Batch Processing – When you send batches (collection) of data to be processed. Batches are generally scheduled: e.g. Every day at 1PM. Batches are not real-time. Batches processing is ideal for very large processing workloads. Batch processing is more cost-effective.

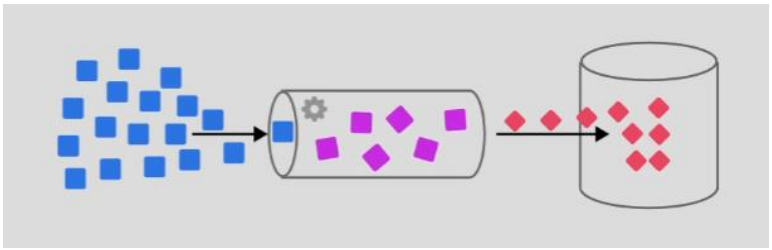


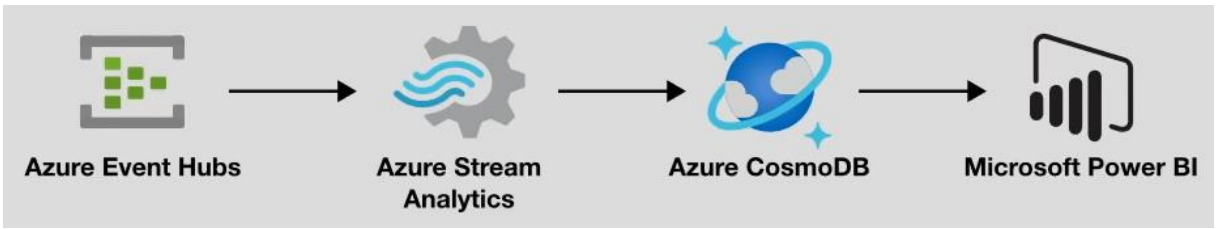


Stream Processing – When you process data as soon as it arrives:

- **Produces** will send data to a stream and
- **Consumers** will pull from the stream

Stream processing is good for real-time analytics or real-time processing (streaming video). Much more expensive than batch processing.





15. Relational data

Tables

A logical grouping of rows and columns. Think like an Excel spreadsheet.

- Tabular data – data that makes use of table data structures

Views

Views is a result set of a stored query on data **stored in memory** (a temporary or virtual table)

Materialized Views

Material Views is a result set of stored queries on data **stored on disk**.

Indexes

A copy of your data sorted by one or multiple columns for faster reads at the cost of storage

Constraints

Rules applied to writes, that can ensure data integrity: e.g. don't allow duplicate records

Triggers

A function that is trigger on specific database events

Primary Key

One or multiple columns that uniquely identify a table in a row e.g. Id

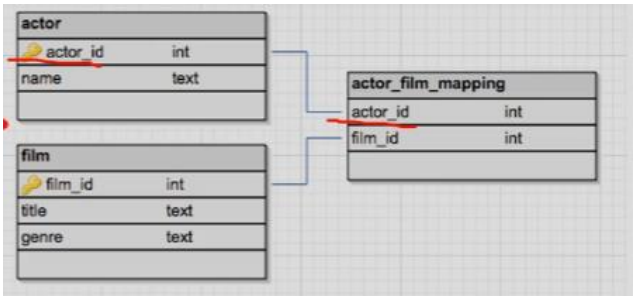
Foreign key

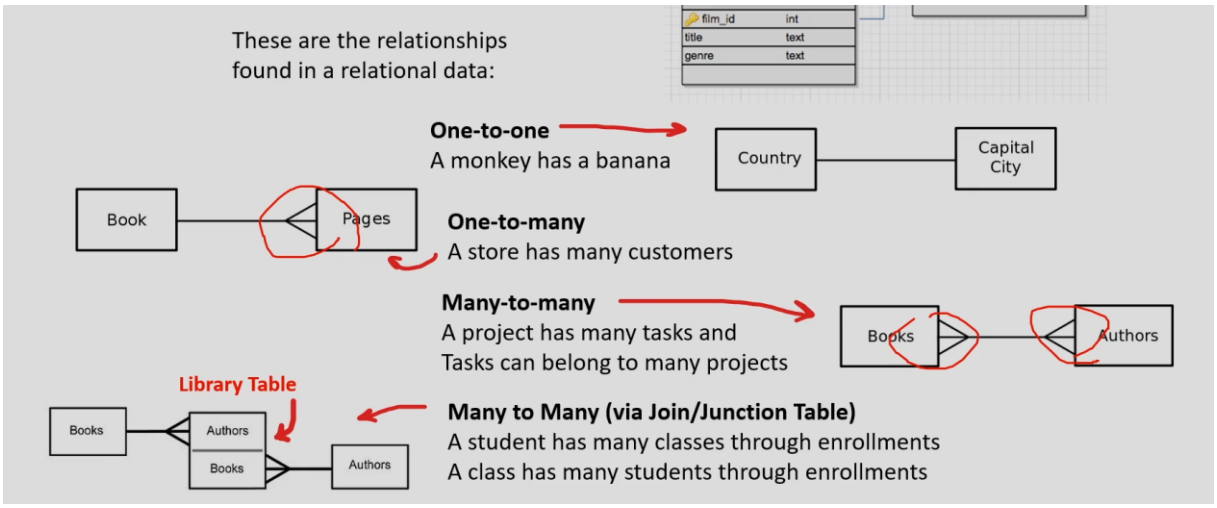
A column which holds the value of primary key from another key to establish a relationship

- A relationship is when two tables have a reference to one another to join data together

16. Relational data – Relationships

Relational databases **establish relationships to other tables** via foreign keys. Referencing another table's primary key.

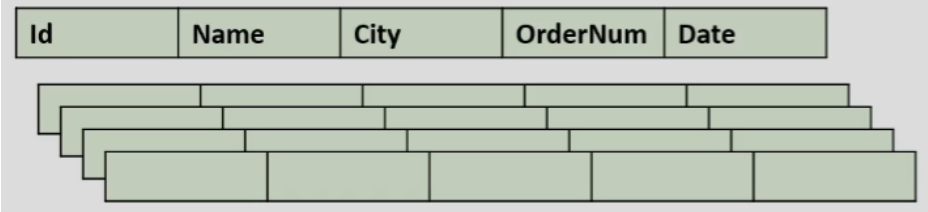




17. Row store vs Column store

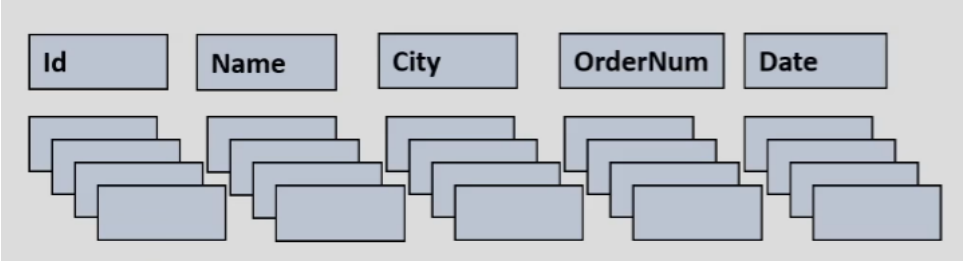
Row-Store

- Data is organized in rows
- Traditional relational databases are row-stores
- Good for general purpose databases
- Suited for online transaction processing (OLTP)
- Great when needing all possible columns in a row is important
- Not the best at analytics or massive amounts of data



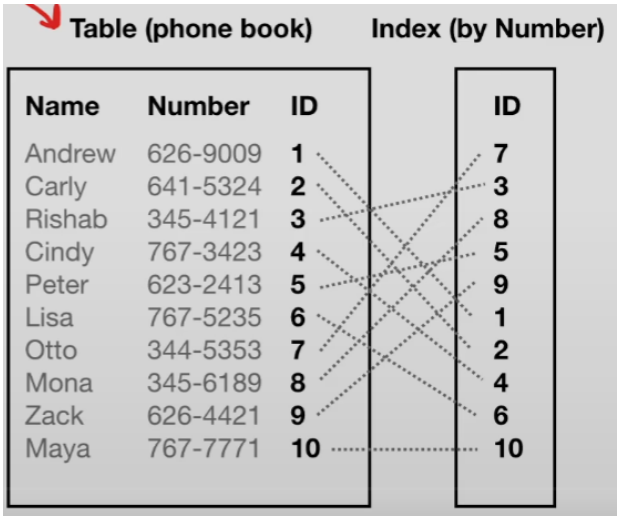
Column-store

- Data is organized into columns
  - Faster at aggerating values for analytics
- NoSQL store or SQL-Like databases
- Great for vast amount of data
- Suited for Online analytical processing (OLTP)
- Great when you only need a few columns



18. Database Indexes

A database index **is a data structure that improves the speed of reads from the database table** by **storing the same or partial redundant data** organized in a more efficient logical order.



The logical ordering is commonly determined by one or more columns: sort key(s)

A common data structure of an index is a **Balanced Tree (B-Tree)**

19. Data Integrity vs Data Corruption

Data integrity is the **maintenance and assurance of, data accuracy & consistency** over its entire life-cycle. Used at as **proxy term for data quality**, data validation is a pre-requisite for data integrity.

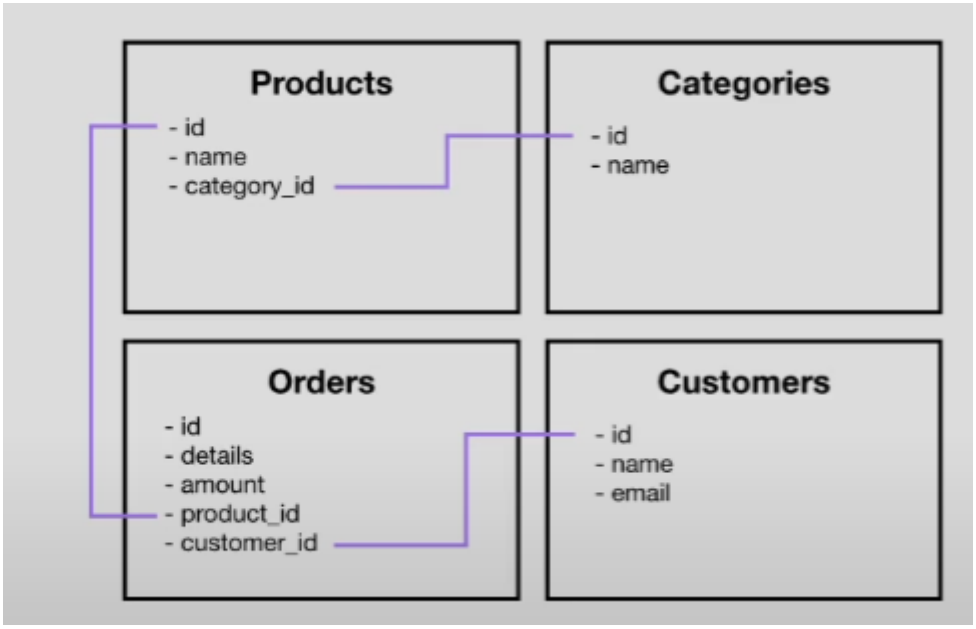
The goal of data integrity:
Ensure data is recorded exactly as intended
Data integrity is the ***opposite*** of Data corruption

Data corruption is the **act or state of data not being in the intended state** and will result in data loss or misinformation. **Data corruption** occurs when untended changes results when reading & writing: • Unexpected hardware failure • Human error w/ inputting/modifying data • Malicious actors with intent of corrupting ur data • Unforeseen side affects for automated operations via computer code	**How do we ensure data integrity?** • Have a well defined and documented data modelling • Logical constraints on your databases items • Redundant and versions of ur data to compare & restore • Human analysis of the data • Hash function to determine if changes have been tampered • Principle of least-privilege, (limiting access to specific actions for specific user roles)

20. Normalized vs Demoralized data

Normalized

A schema design to store **non-redundant** and **consistent data**



Data Integrity is maintained

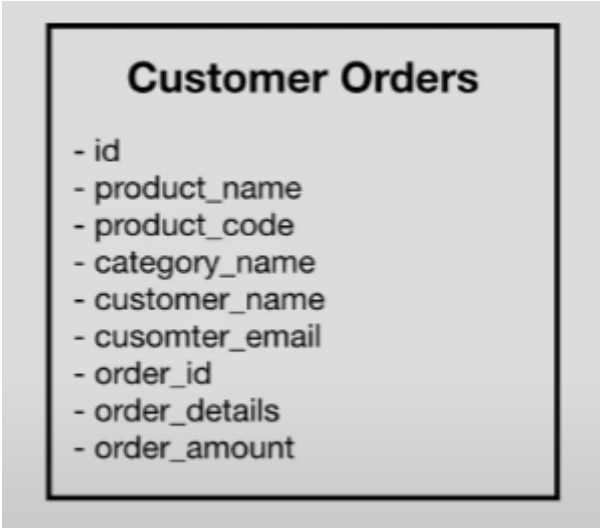
Little to no redundant data

Many tables

Optimizes for storage of data

Denormalized

A schema that **combines data** so that **accessing data (querying) is fast**



- Data Integrity is not maintained
- Redundant data is common
- Fewer tables
- Excessive data, storage is less optimal

21. Pivot Tables

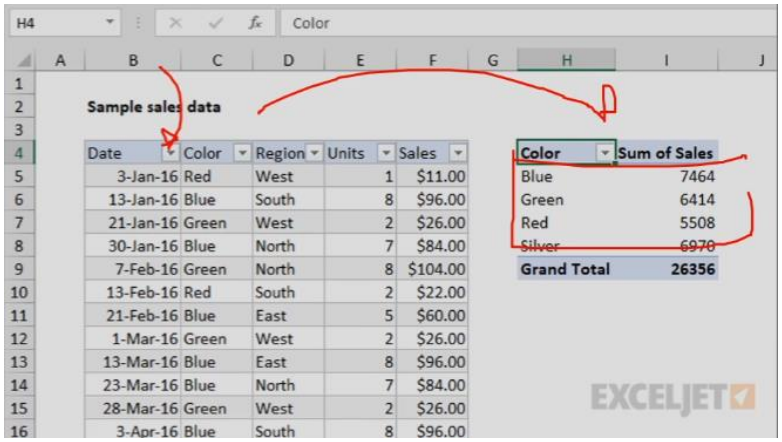
A pivot table is **a table of statistics that summarizes** the data of a more extensive table from a: Database, Spreadsheet or Business intelligence (BI) tool

- Pivot tables are a technique in **data processing**
- They arrange and rearrange (or "pivot") statistics in order to **draw attention to useful information**
- This leads to finding **figures and facts quickly** making them integral to data analysis

In **Microsoft Excel** its very easy to create Pivot Tables. Think of a pivot tables as an interactive report where you can quickly aggregate (group) your data based on various factors e.g:

- By Year/Month/Week or Day
- Sum, Average, Mix or Max

"PivotTable" used to be a trademarked word owned by Microsoft



22. Strongly Consistent vs Eventually Consistent

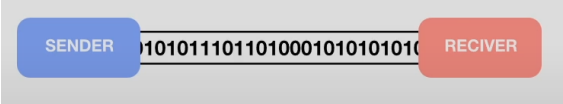
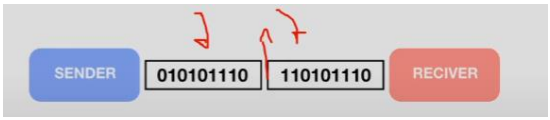
What is data consistency?

When data being kept in two different place and **whether the data exactly match** or do not match. When you have to have duplicates of your data in many places and need to keep them up-to-date to be exact matching. Based on how data is transmitted and service levels cloud service providers will use these two terms:

Strongly consistent	**Eventually Consistent**
• Every time you request data (query) you can expect consistent data to be returned with x time (1sec). • We will never return to your old data. But you will have to wait at least 2 sec for the query return	• When you request data, you may get back inconsistent data within 2 secs • We are giving you whatever data is currently in the database, you may get new data or old data, but if you wait a little bit longer it will generally be up to date

23. Synchronous vs Asynchronous

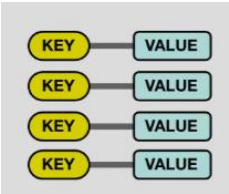
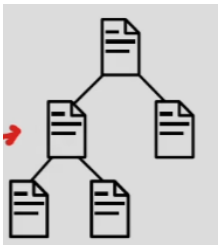
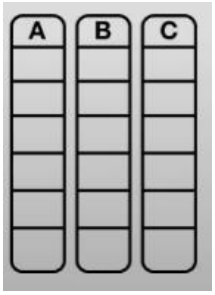
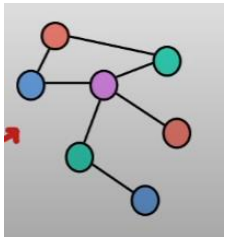
Synchronous & Asynchronous can refer to **mechanism for data transmission or data replication**.

Synchronous **Continuous stream of data that is synchronized by a timer or clock** (*guarantee of time*) Can only access data once transfer is complete • Guaranteed consistency of data return at time of access • Slower access times  A company has a primary db, but they need to have a backup db in case their primary db fails. The company cannot lose any data, so everything must be in-sync. The db is not going to be accessed while it is standing by to act as replacement.	**Asynchronous** **Continuous stream of data separated by start and stop bits** (*no guarantee of time*). Can access data anytime but may return older version or empty placeholder • Faster access times, not guarantee of consistency  A company has a primary db, but they want read-replica (copy of db) so their data analytics person can create computationally intensive reports that do not impact the primary db. It does not matter if the data is exactly 1-to-1 at time of access.

24. Non-Relational Data

A non-relational database **stores data in a non-tabular form** and will be optimized for different kinds of data-structures.

Types of non-relational databases

Key/Value • Each value has a key • Designed to scale • Only simple lookups	
Document • Primary entity is a JSON-like data-structure called a document	
Columnar • Has a table-like structure but data is stored around columns instead of rows	
Graph • Data is represented with nodes and structures. Where relationships matter	

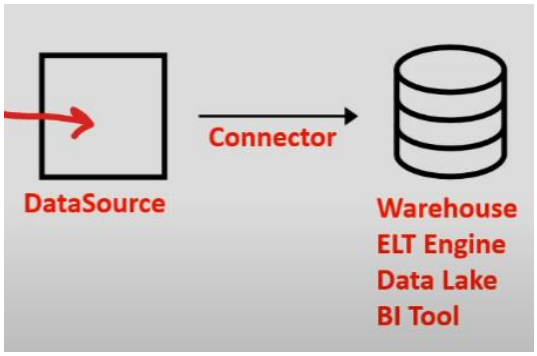
Sometimes non-relational database can be both Key/value & Document. E.G: Azure CosmoDB or Amazon DynamoDB

25. Data Sources

What is a data source?

A data source is **where data originates from**.

An analytics tool may be connected to various data sources to create a visualization or report

A **data source** could be a: • Data lake • Data warehouse • Datastore • Database • Data requested on demand from an API endpoint from a web-app • Flat files (e.g: excel spreadsheet)	

Extracting data from data sources

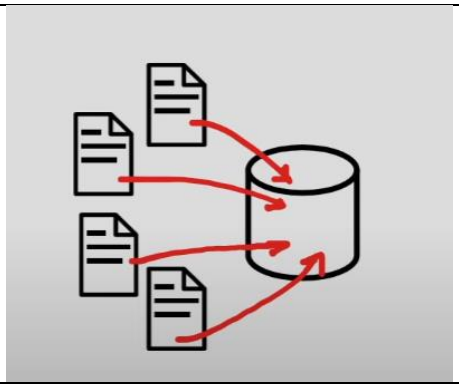
A data tool like Business Intelligence (BI) software would establish a connection to multiple data sources. A BI will *extract* data which could pull data at the time report, or could pull data on a schedule, or data could be streamed. The mechanism for extracting data will vary per data source.

26. Datastore

A Datastore a **repository for persistently storing and managing collections** of **unstructured or semi-structured data**.

Data store is a very broad term, and interchangeable used with databases. But generally, a data store indicates working unstructured or semi-structured data.

A datastore can be specialized in storing

• Flat files (documents, spreadsheets)
• Email (from an email server)
• Databases (complex data stores developed using formal design and modeling technics)
  ○ Relational Databases
  ○ NoSQL Databases
  ○ Object-Oriented Databases
• Data stores designed to be distributed across many machines
• Directory services
	

27. Database

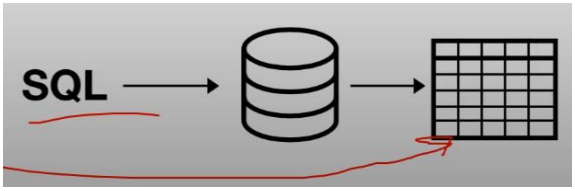
A database is a **data-store that stores semi-structured and structured data**.

A database (db) is more **complex data stores** because it **requires using formal design and modelling techniques**

Databases can be generally categorized as either:

- **Relational databases**
  - Structured data that strongly represents tabular data (tables, rows & columns)
    - Row-oriented or Columnar-oriented
- **Non-relational databases**
  - Semi-structured that may or may not distantly resemble tabular data.

Databases have a rich set of functionalities

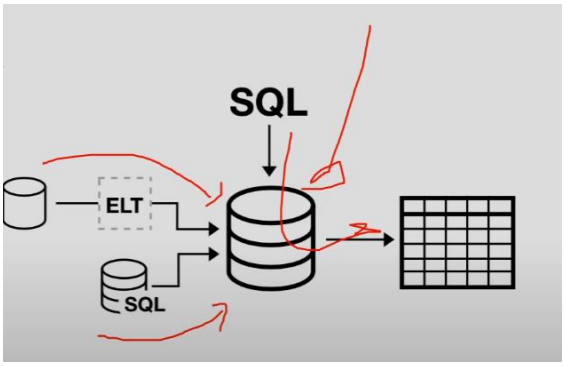
• Specialized language to query (retrieve data) • Specialized modeling strategies to optimize retrieval for different use cases • Finer tune control over the transformation of the data into useful data structures or reports	

Normally a database infers someone is using a **relational row-oriented data store**.

28. Data warehouse

A **relational** data-store designed for **analytic workloads**, which is generally **column-oriented data-store**.

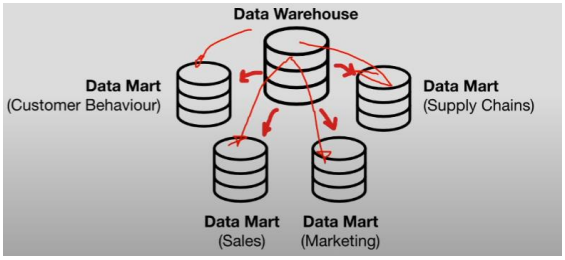
Companies will have **terabytes and millions of rows of data**, and they need a fast way to be able to produce analytics reports.

Data warehouses generally perform **aggregation** • Aggregation is grouping data (eg. Find a total or average) • Data warehouses are optimized around columns since they need to quickly aggregate column data Data warehouse are generally designed by **HOT** • Hot means they can return queries very fast even though they have vast amount of data Data warehouses (DW) are infrequently accessed meaning they aren't intended for real-time reporting but maybe once or twice a day or once a week to generate business and user reports. A DW needs to consume from a relational db on a regular basis.	

29. Data Mart

A data mart is a **subset of a data warehouse**
A data mart will store under **100 GB and has a single business focus**

- Data mart allows different teams or departments to have control over their own dataset for their specific use case
- Data marts are generally designed to be ***read-only***
- Data marts also increase the frequency at which data can be accessed
- The cost to query the data is much lower and so queries can be performed multiple times a day or even hourly



30. Data Lakes

A **data lake** is **a centralized storage repository that holds a vast amount of raw data (big data)** in either a semi-structured or unstructured format.

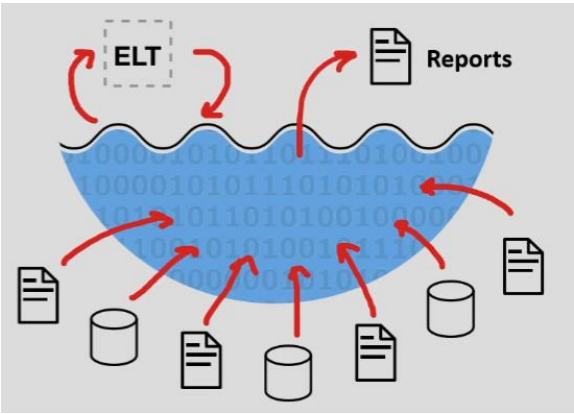
A data lake lets you store all your data without careful design or having to answer **questions** on the future use of the data. ***(Hording for data scientist)***

A data lake is commonly accessed for data workloads such as:

- Visualizations (**Business Intelligence**)
- Real-time analytics
- Machine Learning
- On-premise data

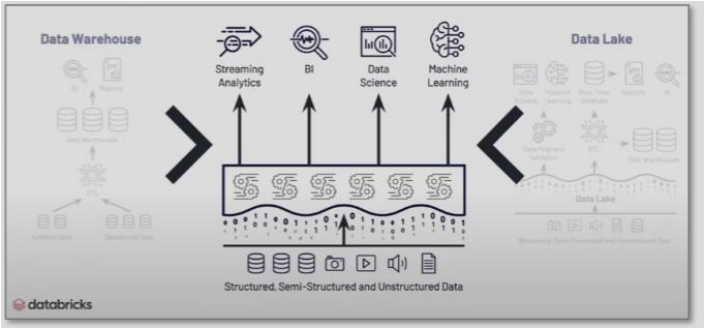
Data lakes are great for data-scientists but its very hard to use data lake for BI Reporting.

If data lakes are not well-maintained they can become **data-swamps** (a mess of data – a data corruption)



31. Data Lakehouse

A Data Lakehouse combines the best elements of a **data lake** and a **data warehouse**

Data Lakehouses compared to a Data Warehouse can:	Data Lakehouses compared to a Data lake can:
• Support video, audio and text files • Support data science and ML workloads • Have support for both Streaming & ELT • Work with many open-source formats • Data will generally reside in a data lake or blob stores	• Perform BI tasks very well • Much easier to setup and maintain • Has management features to avoid a data lake becoming a data swamp • More performant than a data lake
	

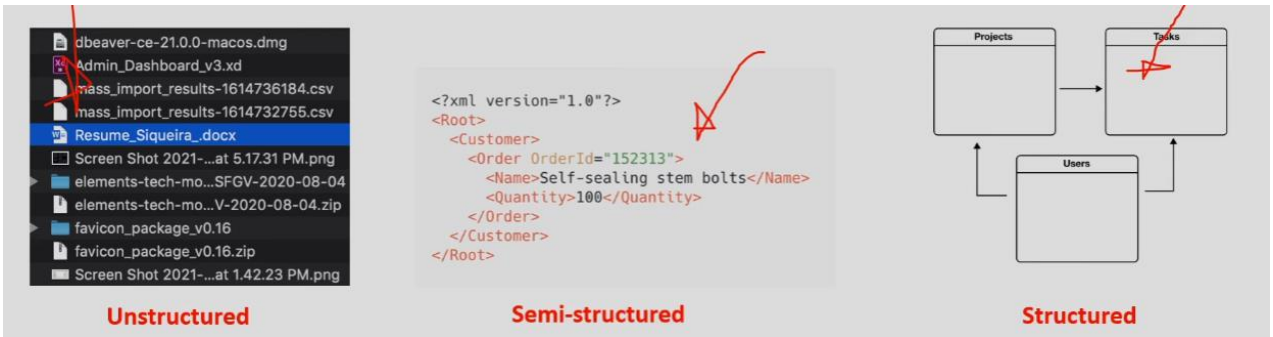
32. Data Structures

What is a Data Structure?

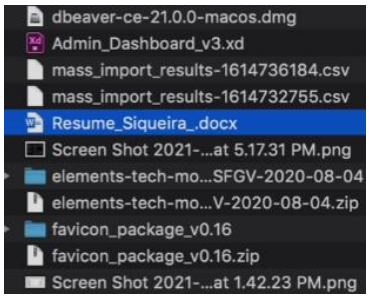
Data that is organized in a **specific storage format**, that enable easy access and modification.
A data structure can store various data types.

Data can be abstractly described to have a degree of structure:

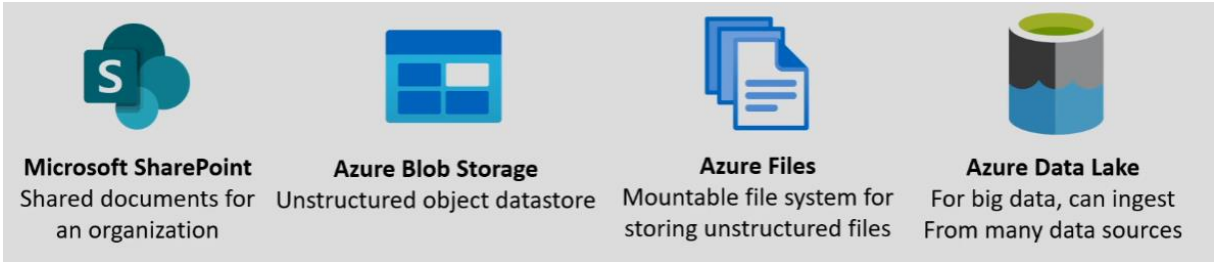
- **Unstructured** – a bunch of lose data that has no organization or possibly relation
- **Semi-structured** – data that can be browsed or searched (with limitations)
- **Structured** – data that can be easily browsed or searched



33. Unstructured data

Unstructured data is just a **bunch of loose data**, think of a junk folder on your computer with a bunch of random files, not optimized for search or analysis, or even simply no relation between various data.	

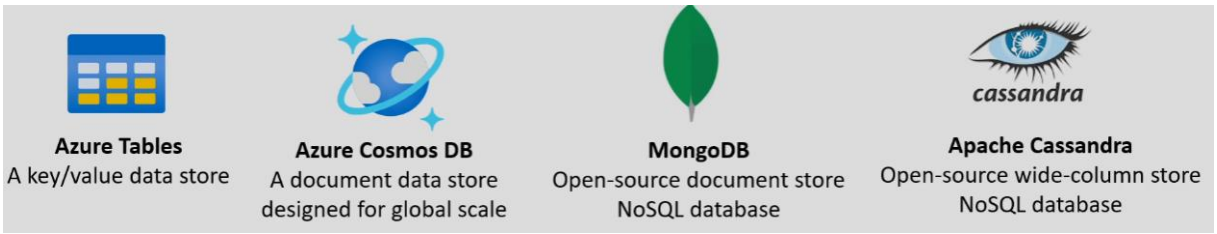
Microsoft and Azure services that store unstructured data



34. Semi-structured data

Semi-structured data is (***no schema**) has some **form of relationship**, its easy to browse data to find related data, you can search data but there are limitations or when you search you will pay at a competitive or operation cost. Co-create semi-structure data structures: • XML, JSON, AVRO, PARQUET	

Azure and other services that store semi-structured data



What is a semi-structured data?

Semi-structured data is data **that contains fields**. The fields don't have to be the same in every entity. You only define the fields that you need on a per-entity basis.

Common semi-structured data structures:

JavaScript Object Notation (JSON)

Format used in JavaScript notation; Store data in memory, read & write from files.

Apache Optimized Row Columnar format (ORC)

Organizes data into columns rather than rows (columnar store data structure)

Apache Parquet

Another columnar data structures. A parquet file contains row group

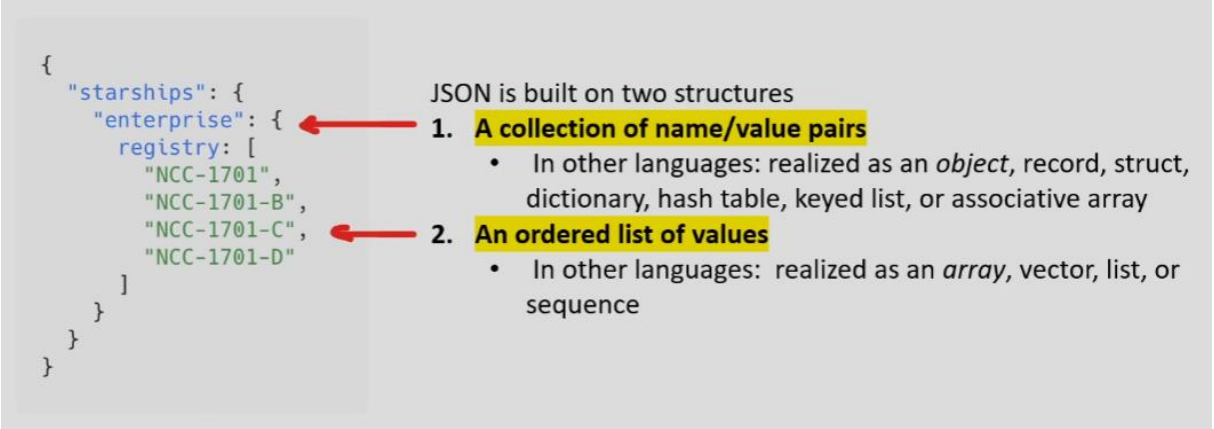
Apache AVRO

Row-based format. Each record contains a header that describes the structure of the data in the record.

35. Semi-Structured – JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format

- It is easy for humans to read & write
- It is easy for machines to parse and generate
- It is based on a subset of the JavaScript



JSON is a **text format** that is completely language independent.

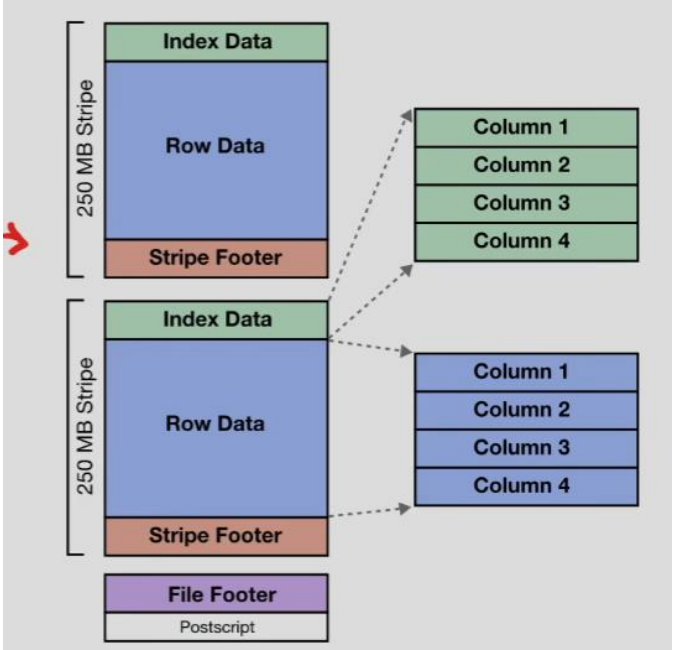
36. Semi-Structured – ORC

Apache ORC (Optimized Row Columnar) a storage format of the Apache Hadoop ecosystem

- It is similar to Parquet files & is the successor of RCFiles
- It was developed by Facebook to supports columnar reads, predictive pushdown and lazy reads.
- It is more storage efficient than RCFiles (taking up 75% less space)
- ORC only support Hadoop's Hive and PIG
- ORC performs better with Hive than Parquet files
- ORC files are organized into a **stripe of data**

The Anatomy on an ORC file

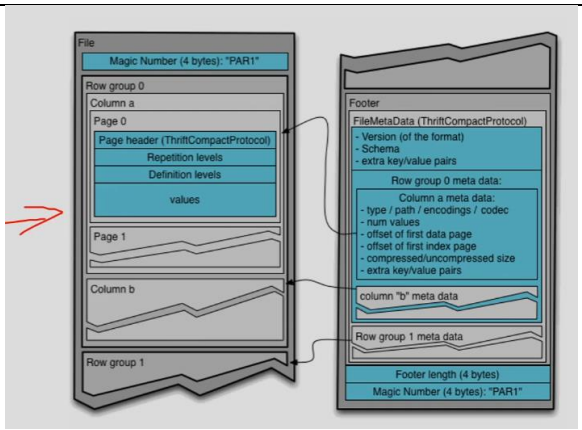
- **File footer** stores auxiliary information
  - List of stripes in the files
  - Number of rows per stripe
  - Each column's data type
  - Column-level aggregates count, min, max & sum
- **Stripe footer** contains directory of stream locations
- **Row data** is used in table scans
- **Index data** includes min & max values for each column and the row positions within each column



37. Semi-Structured – Parquet

Apache Parquet is a **columnar storage file format** available to **any project in the Hadoop ecosystem** (Hive, Hbase, MapReduce, Pig, Spark).

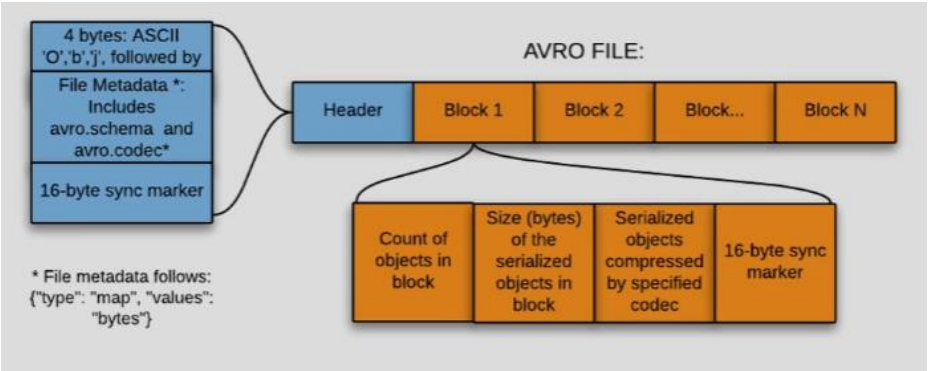
- Parquet is built to support very efficient compression and encoding scheme
- Uses the record shredding and assembly algorithm



38. Semi-Structured – AVRO

	Apache AVRO is a row-based format that provides: • Rich data structures • A compact, fast, binary data format • A container file, to store persistent data • Remote procedure call (RPC) • Simple integration with dynamic languages

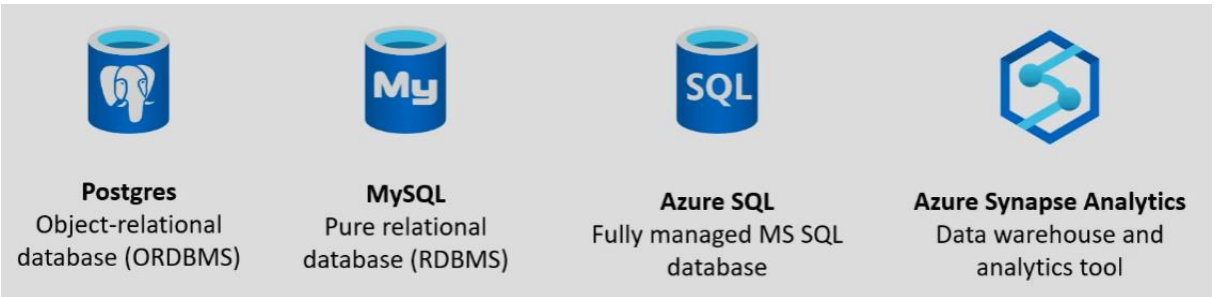
Avro provides functionality similar to systems such as Thrift, Protocol Buffers



39. Structured Data

Structured data is (schema) data that has a relationship, its easy to browse to find related data, it's easy to search data. The most common structured data is tabular data (representating row and columns).

The most common data structure is **tabular data**

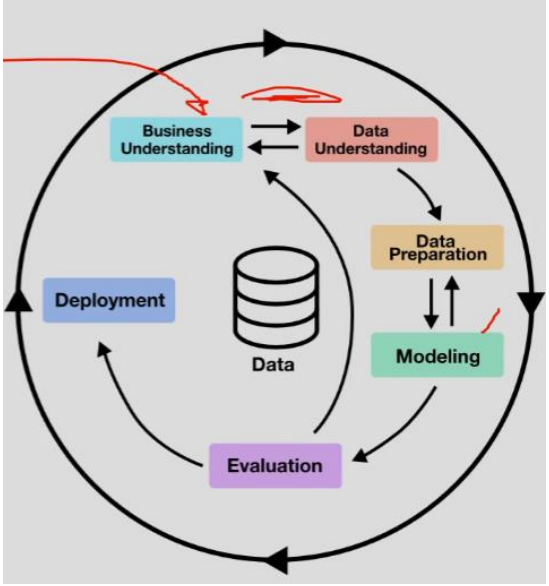


40. Data Mining

What is Data Mining?
The **extraction of patterns and knowledge** from large data **(not the extraction of data itself)**.

Cross-industry standard process for data mining (CRISP-DM) defines Data Mining into 6 phases:

1. **Business understanding**
   What does the business need?
2. **Data understanding**
   What data do we have, & what data we need?
3. **Data preparation**
   How do we organize the data for modeling?
4. **Modeling**
   What modeling techniques should we apply?
5. **Evaluation**
   Which data model best meets the business objective?
6. **Deployment**
   How do people access the data?



41. Data Mining Methods

Data mining methods or techniques is a way to **find valid patterns and relationships** in huge data sets.

Classification: Classify data in different classes
Clustering: A division of information into groups of connected objects
Regression: Identify and analyze the relationship between variables because of the presence of other factors
Sequential: Evaluating sequential data to discover sequential patterns
Association Rules: Discover a link between two or more items, finds a hidden pattern in the data set.
These common ***constraints*** (math formulas) are used to determine significant & interesting links:

- Support – Indication of how frequently the itemset appears in the dataset
- Confidence – Indication of how often the rule has been found to be true
- Lift – Indication of importance compared to other items
- Conviction – Indication of the strength of the rule from statistical independence

Outer Detection: Observation of data items in the data set, which do not match an expected pattern or expected behavior
Prediction: Used a combination of other data mining techniques such as trends, clustering, classification to predict future data

42. Data Wrangling

What is Data Wrangling?

The process of transforming and mapping data from one "raw" data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. Also known as data munging.

There are the **6 core steps** behind data wrangling:

1) Discovery Understand what your data is about and keep in mind domain specific details about your data as you move through the other steps	**2) Structuring** You need to organize your content into a structure that will be easier to work for our end results.
3) Cleaning Remove outliers, change null values, remove duplicates, remove special characters, standardize formatting	**4) Enriching** Appending or enhancing collected data with relevant context obtained from additional sources
5) Validating Authenticate the reliability, quality, and safety of the data	**6) Publishing** Place your data in a datastore so you can use it downstream

43. Data Modeling

What is a Data Model?

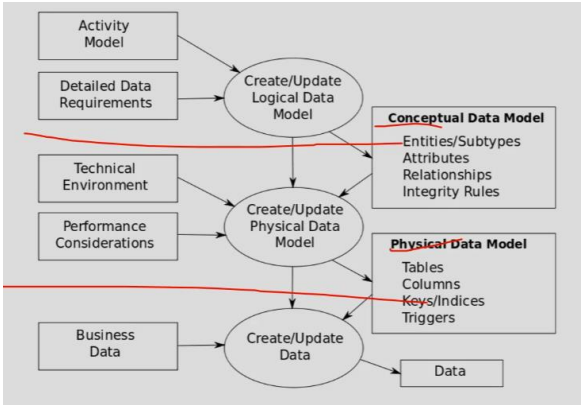
An abstract model that **organizes elements of data and standardizes how they relate to one another** and to the properties of real-world entities. E.g: A data model could be a **relational database** that contains many tables.

A data model could be:

- **Conceptual**
  - How data is represented at the organization level abstractly without concretely defining how it work within software
    - E.g. tables & columns, object-oriented classes
- **Logical**
  - How data is presented in software
    - E.g: tables & columns, object-oriented classes
- **Physical**
  - How data is physically stored
    - E.g: partitions, CPUs, tablespaces…

What is data modeling?

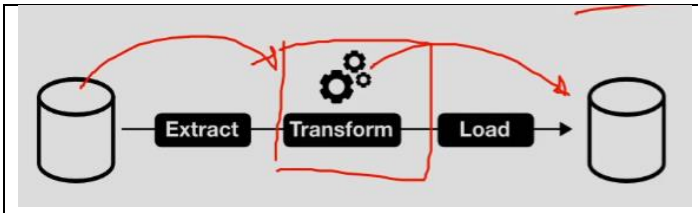
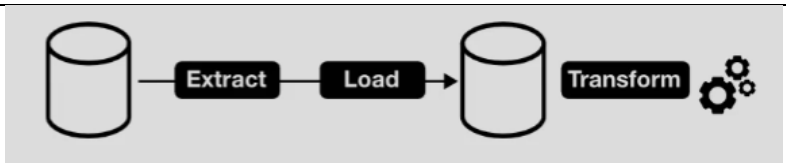
A process used to **define and analyze data requirements needed to support the business processes** within the scope of corresponding information systems in organizations.



44. ETL vs ELT

ETL & ELT is used when you want to **move data from one location to another**, where the datastore/databases have different data structures so you need to **transform the data for the target system**.

E.g: Moving *SQL Server database* into *CosmoDB Tables*

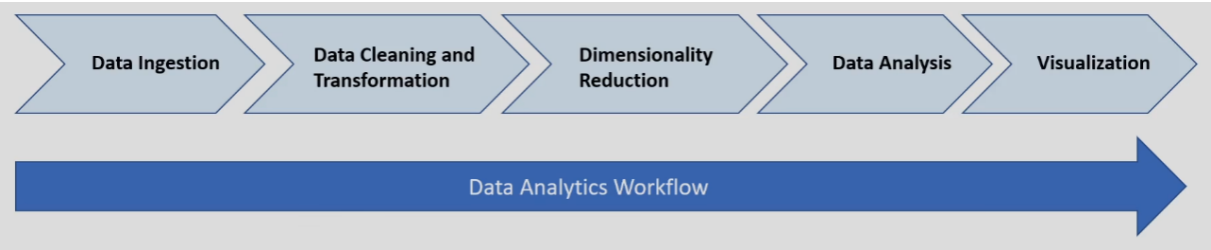
	
Extract, Transform and Load (ETL) • Loads data first into a staging server & then into target system • Used for **on-premises**, relational and structured data • Used for a small amount of data • Doesn't provide data lake support • Easy to implement • Mostly supports relational data	**Extract, Load and Transform (ELT)** • Loads data directly into the target system • Used for scalable cloud structured and unstructured data sources • Used for large amounts of data • Provides data lake support • Requires specialized skills to implement and maintain • Support for unstructured data readily available

45. Data Analytics

What is Data Analytics?

Data analytics is concerned with **examining, transforming, and arranging data** so that you can **extract and study useful information**.

A data analyst commonly uses SQL, Business Intelligence (BI) tools and Spreadsheets



46. Key Performance Indicators (KPIs)

Key Performance Indicators (KPIs) are type of performance measurement that a company or organization use to determine **performance over time**	

KPIs can **evaluate the success** of an organization or of a specific organization activity

There are 2 categories of measurements for KPIs:

1. **Quantitative**
- Properties can be measure with a numerical result
- Facts presented with a specific value
    - E.g: monthly revenue, number of signups, number of reported defects

2. **Qualitative**
- Properties that are observed and can generally not be measured with a numerical result
- Numeric or textual value that represent personal feelings, tastes, or opinions
    - E.g: Customer sentiment

47. Data Analytic Techniques

Descriptive Analytics – What happened?

- Specialized metrics
  - Key Performance Indicators (KPI)
  - Return on Investment (ROI)
- Generating sales & financial report
- Accurate, comprehensive, live-data & effective viz

Diagnostic Analytics – Why did it happen?

- Supplemental to descriptive analysis
- Drill down, or investigate descriptive metrics to determine root cause
- Find & Isolate anomalies into its own datasets & apply statistical techniques

Predictive Analytics – What will happen?

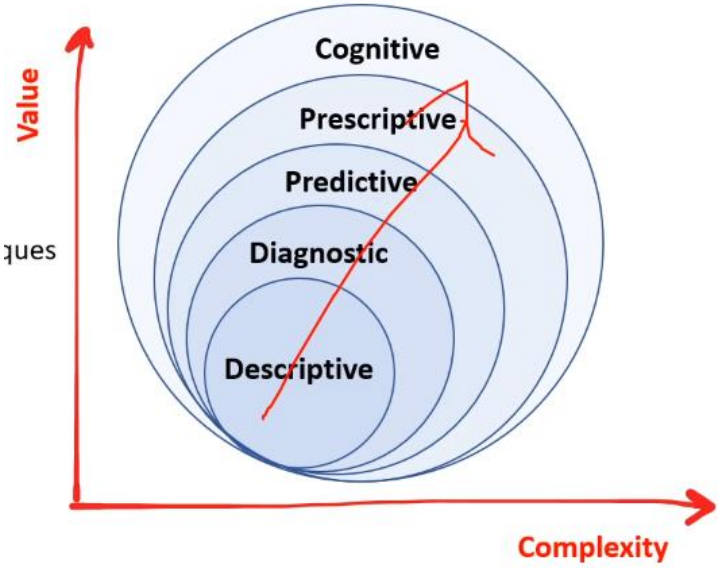
- Use historical data to predict trends / reoccurrence
- Statistical & ML techniques applied
  - E.g: Neural Networks, Decision Trees, Regression, Classification

Prescriptive Analytics – How can we make it happen?

- Goes a step further than predictive and uses ML by ingesting hybrid data to predict future scenarios that are exploitable

Cognitive Analytics – What-if this happens?

- Using analytics to draw patterns to create what-if scenarios and what actions can be taken if those scenarios become reality



48. Microsoft OneDrive

Microsoft OneDrive is **a storage & storage synchronization service** for files which resides in the cloud. Similar products: DropBox, GoogleDrive, Box & *Microsoft SharePoint.

One drive is intended for personal storage for a single individual. You pay for different size of storage (5GB ***free***, 100GB, 1TB, 6TB…). You do not worry about the underlying hardware's the durability, resilience, fault tolerance, or availability.

Files can be shared easily to another user via:

- A shareable link
- Or to a specific email that also has a OneDrive account

Files are accessed via:

- A web-application (web interface)
- Via shared folders that hold a reference to the files stored in the cloud

File can be synchronized

- A copy resides on a local computer HDD is copied to the cloud
- A file residing in the cloud can be copied to a local computer hard drive
- Copying occurs automatically when files are changed
    - Difference in files could result in conflicts a user must choose to which file to keep

Files can be versioned (you can recover older versions of a files)

- Older files may retain for 30 days and be automatically deleted

49. Microsoft 365 - SharePoint

Microsoft 365 SharePoint is **a web-based collaborative platform** that integrates into Microsoft Office. Intended **for document management and shared storage**.

SharePoint Sites

Data within SharePoint organized around Sites. A site is a collaborative space for teams with the following components:

- **Document Library**
- Pages
- Web Parts
- And more…

SharePoint Document Library

A document library is a file storage & synchronization but ***designed for teams***. It is very similar to OneDrive, but files are owned by the company and not an individual. You can apply robust permissions to access files within or outside your organization. A Site always has a default Document Library called "documents".

50. Data Core Concepts Cheat Sheet

Data – Units of information
Data Documents – types of abstract groupings of data
Data Sets – unstructured logical grouping of data
Data Structures – structured data

- **Unstructured – a bunch of lose data that has no organization or possibly relation**
  - Flat files – various files that can reside in a file system
- **Semi Structured – data that can be browsed or searched (with limitation) e.g: CSV, XML, JSON…**
  - XML – markup language that looks like html (e.g: <hello><world>earth</world></hello>)
  - JSON – a text file that is composed of dictionaries & arrays (e.g: {"hello": ["earth», «mars"]}
  - RCFiles – a storage format designed for MapReduce framework
  - ORC – a columnar data-structure, 75% more efficient than RCFiles, limited computability (works great well with HIVE)
  - AVRO – a row-wise data-structure for Hadoop systems
  - Parquet – a columnar data-structure that has more support for Hadoop systems than ORC
- **Structured – data that can be easily browsed or searched (e.g: tabular data)**
  - Tabular data – data that is arranged as tables, think excel spreadsheets

Data Types – How single units of data are intended to be used
Database Administrator – **configures and maintains a database** (e.g: Azure Data Services or SQL server)
Data Engineer – **Design & implement data tasks** related to the transfer and storage of **big data**
Data Analyst – **Analyzes business data** to reveal important information
Software as a Service (SaaS) – **A product that is run & managed by the service provider**
Platform as a Service (PaaS) – **Focus on the deployment & management** of your apps
Infrastructure as a Service (IaaS) – **Basic building blocks** for cloud IT. Provides access to networking, computers & data storage space.
Datastores – Unstructured or semi-structured data to housing data, a board term than can encompass anything that stores data
Databases – Structured data that can be quickly accessed & searched (generally relational, row-based, tabular data for OLTP)
Data warehouses – Structured or semi-structured data for creating reports & analytics (column based, tabular data for OLAP)
Data marts – a subset of data warehouse for a specific business data task
Data lakes – combines the best of data warehouses & data lakes
Notebooks – data that is arranged in pages, designed for easy consumption
Batching – When you send batches (a collection) of data to be processed. Not real-time
Streaming – When data is processed as soon as it arrives. It's real-time

Relational data – Data that uses structured tabular data and has relationships between tables

- One-to-One – a monkey has a banana
- One-to-Many – a store has many customers
- Many-to-Many – a project has many tasks, and tasks can belong to many projects
- Join/Junction Table – A student has many classes through enrollments and A class has many students through enrollments
- Row-store – data organized in rows, optimized for OLTP (general computing & transactions)
- Column-store – data organized in columns, optimized for OLAP (analytics)
- **Indexes** – a data-structure that improves the read of databases

Pivot Table – it's a table of stats that sum the data of a more extensive table from a Database, Spreadsheet or BI tool.

Non-relational data – Data that is semi-structured associated with Schemaless & NoSQL db

- Key Value – Each value has a key, designed to scale, only simple lookups
- Document – Primary entity is an XML or JSON-like data structure called a document
- Columnar – Has a table-like structure but data is stored around columns instead of rows
- Graph – Data is represented with nodes and structures: Where relationships matter

Data Modeling – an abstract model that organizes elements of data & standardizes how they relate to one another and to real-world entities

Schema – a formal language to describe the structure of data used by db and data stores during the data modeling phases

Schemaless – generally used when upfront data modelling can be forgone because the schema is flexible, normally used with NoSQL db.

Data Integrity – the maintenance and assurance of data accuracy & consistency over its entire life-cycle.

Data Corruption – the act or sate of data not being in the intended state & will result in data loss or misinformation

Normalized – A schema designed to store non-redundant and consistent data

Denormalized – A schema that combines data so that accessing data (querying) is fast

Extract, Transform and Load (ETL) – transform data from one data store to another, loads data in an intermediate stage, doesn't work with data lakes

Extract, Load and Transform (ELT) – transformation done at the target data store, works with data lakes, more common in cloud services.

Query – when a user requests data from a data store by using a query language to return a data result

Data Source – A data source is **where data originates from**. Analytics and data warehouses tools may be connected to various data sources

Data consistency – When data is being kept in two different place and **whether the data exactly match** or do not match

- **Strongly Consistent** – Every time you request data (query) you can expect consistent data to be returned within a time
- **Eventually Consistent** – When you request data you may get back inconsistent data (stale data)

Synchronization – continuous stream of data that is synchronized by a timer or clock (guarantee of time)
Asynchronization – continuous stream of data separated by start & stop bits (no guarantee of time)
Data Mining – The **extraction of patterns and knowledge** from large amounts of data **(not the extraction of data itself)**
Data Wrangling – The process of transforming and mapping data from one "raw" data form into another format
Data Analytics – Data analytics is **examining, transforming, and arranging data** so that you can **extract & study useful information**.
Key Performance Indicators (KPI) – type of performance measurement that a company or an organization use to determine **performance over time**
Descriptive Analytics (What happened?) – Accurate, comprehensive, live-data and effective visualizations e.g: dashboards, reports, KPI, ROI…
Diagnostic Analytics (Why did it happen?) – Drill down to investigate root cause, focused on subset of descriptive analytics dataset
Predictive Analytics (What will happen?) – Use historical data with **statistics and ML to generate trends or predictions**
Prescriptive Analytics (What will happen?) – Using hybrid data with ML to predict future scenarios that are exploitable
Cognitive Analytics (What-if this happens?) – Using ML and NLP to determine what-if scenarios to create plans if they happen
OneDrive – storage & storage synchronization service for a single user
SharePoint – storage & storage synchronization service for an organization

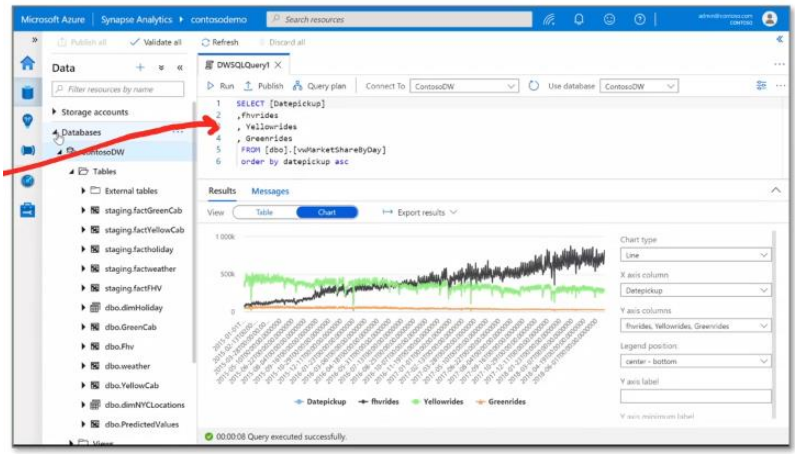
II. Azure Synapse and Data Lake

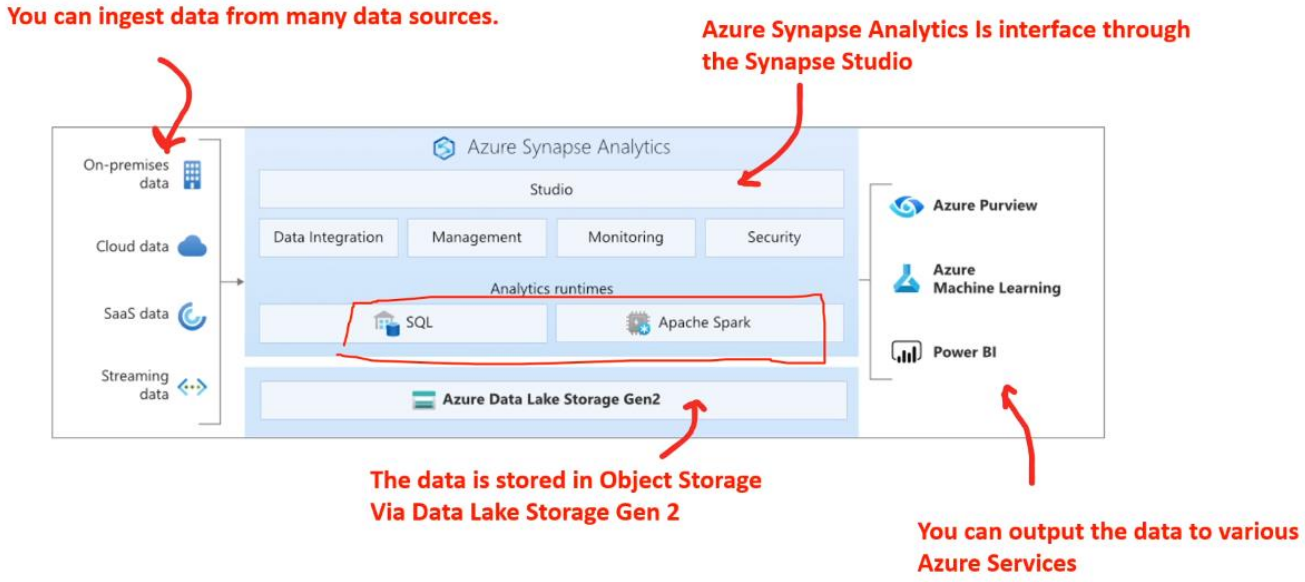
1. Azure Synapse Analytics

Azure Synapse Analytics is a **Datawarehouse** and **unified analytics platform**

Build ETL/ELT processes:

- In a code-free visual environment
- Easily ingest data from more than 95 native connector
- Deeply integrated Apache Spark
- Use **T-SQL** queries on both your data ware house and Spark engines
- Support multiple languages : T-SQL, Python, Scala, Spark SQL, and .Net
- Integrated with Artificial Intelligence (AI) and BI tools
    - Azure Machine Learning
    - Azure Cognito Services





2. Synapse SQL

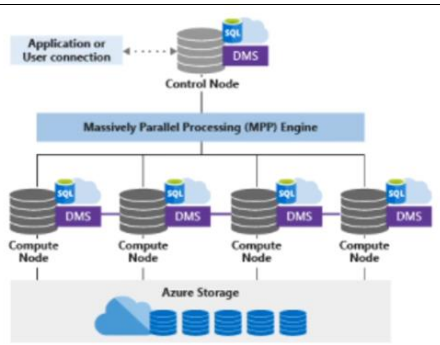
Synapse SQL is a distributed version of T-SQL designed for data warehouse workloads

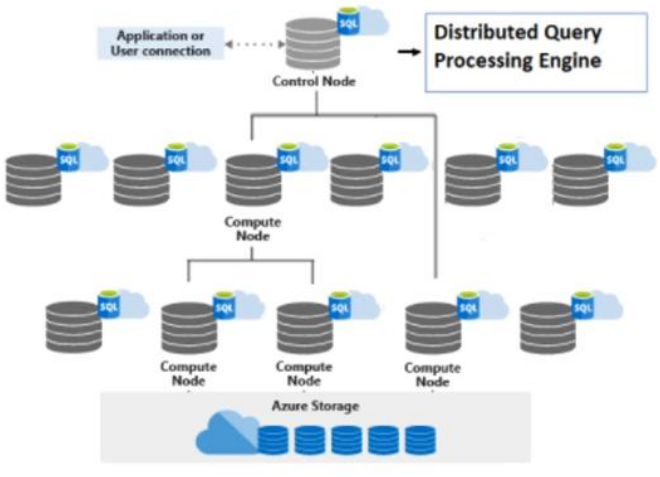
- Extends T-SQL to address streaming & machine learning scenario
- Use built-in **streaming** capabilities to land data from cloud data sources into SQL tables
- Integrate AI with SQL by using ML models to score data using the T-SQL PREDICT function
- And offers both **serverless** and **dedicated** resource models

For **unpredictable** workloads (unplanned or bursty) use **the always-available, serverless SQL endpoint**. (**serverless**)

For **predictable** workloads (**dedicated**)

- Create dedicated SQL pools to reserve processing power for data stored in SQL tables

Dedicated SQL pool is a query service over the data in your **data warehouse** The unit of scale is an abstraction of compute power that is known as a **data warehouse unit (DWU)**. Once your dedicated SQL pool is created, you can import big data with simple PolyBase T-SQL queries, and then use the power of the distributed query engine to run high-performance analytics.	

Serverless SQL pool is a query service over the data in your **data lake** Scaling done automatically to accommodate query resource requirements As topology changes over time by adding, removing nodes or failovers It adapts to changes and makes sure your query has enough resources and finishes successfully	

3. Apache Spark for Synapse

Azure Synapse can **deeply and seamlessly integrate** with Apache Spark

- ML models with SparkML algorithms & AzureML integration for Apache Spark 2.4
  - With built-in support for Linux Foundation Delta Lake
- Simplified resource model that frees you from having to worry about managing clusters
- Fast Spark start-up & aggressive autoscaling
- Built-in support for .NET for Spark allowing you to reuse your C# expertise and existing .NET code within a Spark application

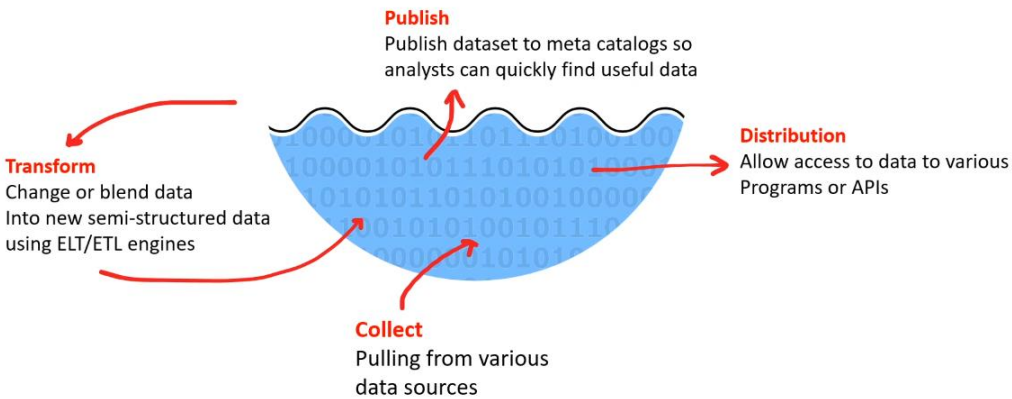
4. Apache Spark with Data Lake

Azure Synapse removes the traditional technology barriers between using SQL & Spark together. You can seamlessly mix and match based on your needs and expertise.

- Tables defined on files in the data lake are seamlessly consumed by either Spark or Hive
- SQL & Spark can directly explore and analyze Parquet, CSV, TSV, and JSON files stored in the data lake
- Fast, scalable data loading between SQL and Spark db

5. Introduction to Data Lakes

A data lake is a **centralized data repository for unstructured & semi-structured data**. A data lake is intended to store vast amount of data. Data lakes generally **use objects (blobs) or files** as its storage medium.



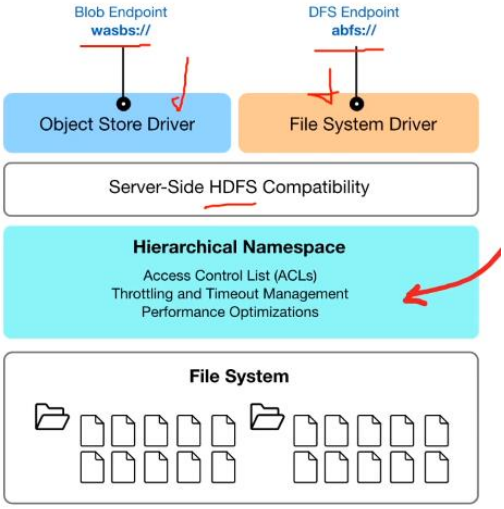
Azure Data Lake Storage Gen1 *(depracted)*

The first version of Data Lake Storage and will be retired in 2024. New users should use Gen2

Azure Data Lake Storage Gen2

Data Lake Storage is Azure Blog storage which has been extended to support big data analytics workloads:

- Designed to handle **petabytes of data** and **hundreds of gigabits of throughput**
- In order to efficiently access data, Data Lake Storage adds a **hierarchical namespace** to Azure Blob Storage



6. PolyBase (**IMPORTANT**)

PolyBase is a **data virtualization feature** for **SQL Server**

PolyBase enables your SQL Server instance to query data with T-SQL directly from:

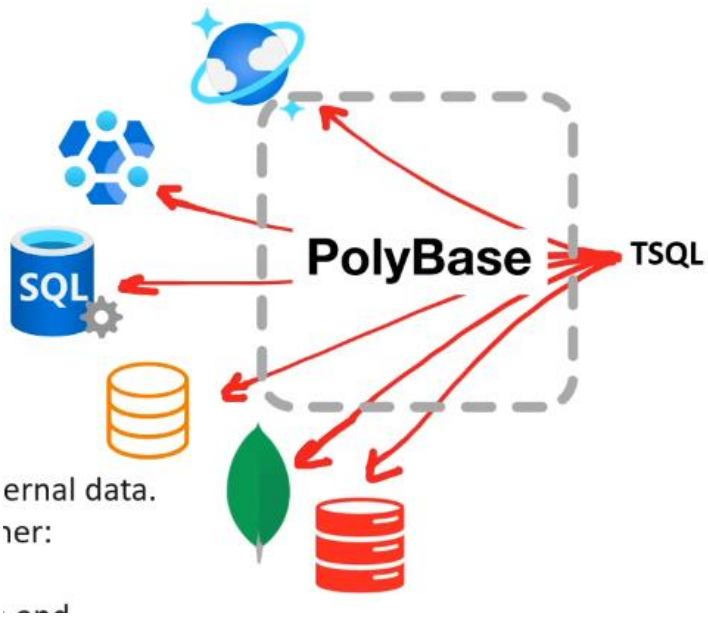
- SQL Server
- Oracle
- Teradata
- MongoDB
- Hadoop clusters
- Cosmos DB

Without separately installing client connection software

PolyBase allows you to join data from a SQL Server instance with external data.

Prior to PolyBase to join data external data sources you could either:

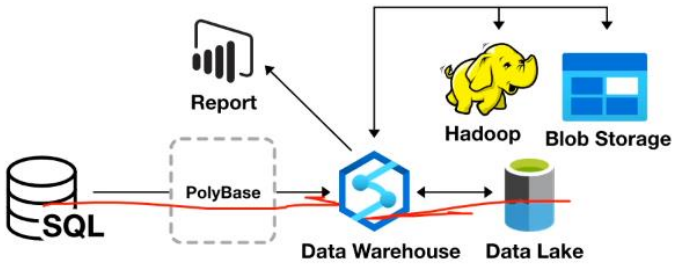
- Transfer half your data so that all the data was in one location
- Query both sources of data, then write custom query logic to join and integrate the data at the client level



7. Azure Synapse Analytics – ELT

You can **perform ELT using Synapse SQL** in Azure Synapse Analytics

This the fastest & most scalable way to load data is through PolyBase external tables and the COPY statement

With PolyBase and the **COPY statement**, you can access external data stored in Azure Blob storage or Azure Data Lake Store via the T-SQL language. The basic steps for implementing ELT are: 1. Extract the source data into text files 2. Load the data into Azure Blob storage or Azure Data Lake Store 3. Prepare the data for loading 4. Load the data into staging tables with PolyBase or the COPY command 5. Transform the data 6. Insert the data into production tables	

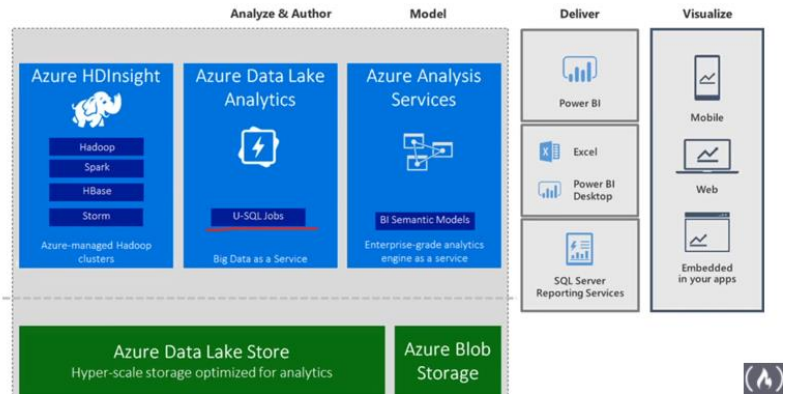
7. Azure Data Lake Analytics

Azure Data Lake Analytics is an **on-demand analytics job service** that simplifies big data

Instead of deploying, configuring, and tuning hardware…

You **write queries** (via U-SQL) to **transform your data and extract valuable insights**.

Exporting approximately 2.8 billion rows of TPC-DS store sales data (~500 GB) into a CSV file took less than 7 minutes and importing the full 1 TB set of source data into Azure Analysis Services by using the Azure Data Lake connector took less than 6 hour



U-SQL

U-SQL is a structured query language included within Data Lake Analytics to **perform queries** on your data lake.

U-SQL can query and combine data from a variety of data sources, including:

- Azure Data Lake Storage
- Azure Blob Storage
- Azure SQL DB
- Azure SQL Data Warehouse,
- SQL Server instances running in Azure VMs

You can install **Azure Data Lake Tools** for Visual Studio to perform U-SQL jobs on your Azure Data Lake

```
DECLARE @in  string = "/Samples/Data/SearchLog.tsv";
DECLARE @out string = "/output/result.tsv";

@searchlog =
    EXTRACT UserId          int,
            Start           DateTime,
            Region          string,
            Query           string,
            Duration        int?,
            Urls            string,
            ClickedUrls     string
    FROM @in
    USING Extractors.Tsv();

@rs1 =
    SELECT Start, Region, Duration
    FROM @searchlog
    WHERE Region == "en-gb";

@rs1 =
    SELECT Start, Region, Duration
    FROM @rs1
    WHERE Start >= DateTime.Parse("2012/02/16")
          AND Start <= DateTime.Parse("2012/02/17");

OUTPUT @rs1
TO @out
```

8. Azure Synapse and Data Lake Cheat Sheet

A data lake is a **centralized data repository for unstructured & semi-structured data**

- A data lake is intended to store vast amounts of data
- Data lakes generally use objects (blobs) or files as its storage medium

Azure Data Lake Store (Gen 2)

- Azure Blob storage which has been extended to support big data analytics workloads
- In order to efficiently access data, Data Lake Storage adds a **hierarchical namespace** to Azure Blob Storage
    - ACLs, Throttling Management, Performance Optimizers
- You can access the data lake via (Blob) wasbs:// or (File system) abfs://

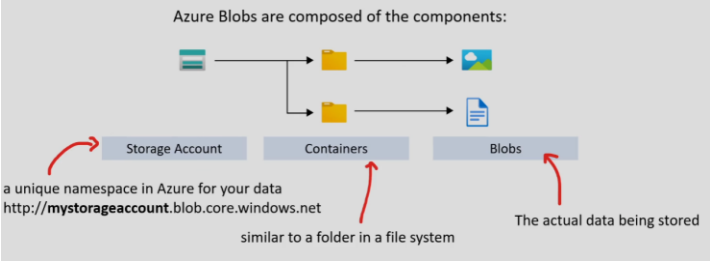
Azure Synapse Analytics – a **data warehouse** and **unified analytics platform**

- Has two underlying transformations engines: SQL Pools & Spark Pools
- Synapse SQL is T-SQL but designed to be distributed
    - SQL Dedicated Pools – reserves compute for processing
    - Serverless Endpoints – on-demand, no guarantee of performance
- Data is stored on Azure Data Lake Store (Gen2)
- Operations are performed within the Azure Synapse Studio
- **PolyBase – enables your SQL Server instance to query data with T-SQL (used to connect many relational data sources)**

III. ACCOUNT STORAGE

1. Azure Blob Storage

Blob storage is an **object-store** that is optimized for **storing massive amounts of unstructured data**. Unstructured data is data that doesn't adhere to a particular model or definition, such as text or binary data.



Azure Storage supports **3 types** of blobs:

1. Block blobs

- Store text & binary data
- Made up of blocks of data that can be managed individually
- Store up to about 4.75 TiB of data

2. Append blobs

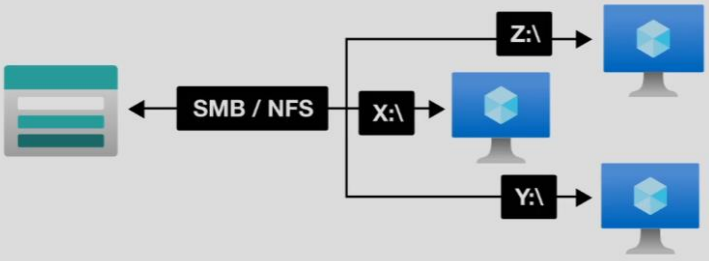
- Optimized for append operations
- ideal for scenarios such as logging data from virtual machine

3. Page blobs

- store random access files up to 8 TB in size.
- store virtual hard drive (VHD) files and serve as disks for Azure virtual machine

2. Azure Files

Azure Files is a fully managed **file share** in the cloud. A file share is a **centralized server for storage** that allows **multiple connections**. *It is like having one big shared drive that everyone (VMs) can work on at the same time.*



To connect to the file share a **network protocol** is used:

- Server Message Block (SMB)
- Network File System (NFS)

When a connection is established the file share's filesystem will be accessible in the specific directory within your own directory tree. This is knowing as **mounting**

Use Cases

- Completely **replace or supplement** on-premises files servers Network Attach Storage (NAS) devices
- **Lift-and-Shift** your on-premise storage to the cloud via Classic Lift or Hybrid Lift
    - "Lift-and-Shift" when you move workloads without rearchitecting, Importing local VMs to the cloud
    - Classic Lift – where both the app & its data are moved to Azure
    - Hybrid Lift – where the app data is moved to Azure Files, and the app continues to run on-premises
- **Simplify cloud development**
    - Shared app settings – Multiple VMs & dev workstations need to access the same config file
    - Diagnostic share – All VMs log to the file share, dev can mount & debug all logs in a centralized place
    - Dev/Test/Debug – Quickly share tools for dev needed for local environments
- **Containerization**
    - You can use Azure Files to persist volumes for stateful containers

Why use Azure files instead of setting up your own File Share server?

- **Shared Access** – Already setup to work with standard networking protocols SMB & NFS
- **Fully managed** – it's kept up to date with security patches, designed to scale
- **Resiliency** – Build to be durable & always working
- **Scripting & Tooling** – You can automate the management & creation of file shared with Azure API & PowerShell

3. Account Storage Cheat Sheet

Azure Storage Accounts – an umbrella service for various forms of managed storage:

- Azure Tables
- Azure Blob Storage
- Azure Files

Azure Blob Storage – Object storage that is distributed across many machines

- Support 3 types
    - Blob blobs – store text & binary data, blocks of data that can be managed individually, up to 4.7TiB
    - Append blobs – Optimized for append operations, ideal for logging
    - Page blobs – store random access files up to 8TB in size

Azure Files is a fully managed **file share** in the cloud

- To connect to the file share, a network protocol is used
    - Server Message Block (SMB)
    - Network File System (NFS)

Azure Storage Explorer – a standalone cross-platform app to access various storage formats within Azure Storage accounts

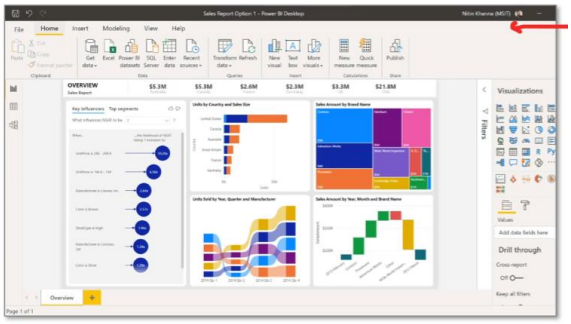
IV. POWER BI

1. Business Intelligence (BI)

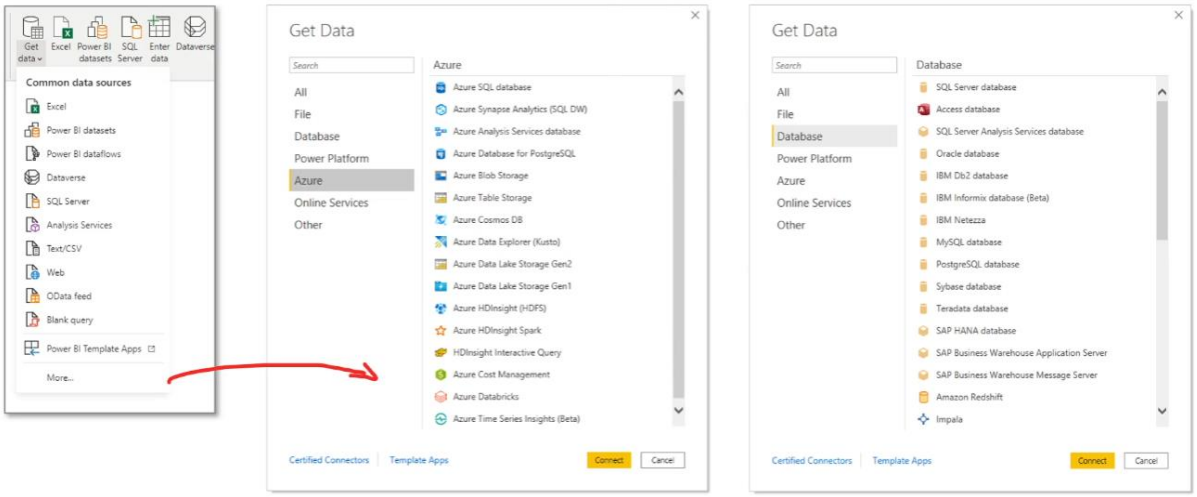
Business Intelligence (BI) is both a data-analysis strategy & **technology** for business information's. The most popular BI tools are: • Tableau • Microsoft Power BI • Amazon QuickSight	Business Intelligence (BI) helps organizations to make data-driven decisions by (BI) combining: • Business analytics • Data mining • Data visualization • Data tools • Infrastructure • Best practices

2. Microsoft Power BI

Power BI is a Business Intelligence tool for **visualization business data**

	**Power BI Desktop** A way to design and ingest reports **Power BI Mobile** View reports on the go **Power BI Service** Access & modify reports in the cloud **Power BI embedded** A way to embed Power BI components into your apps

Microsoft Power BI can ingest data from **many data sources**.

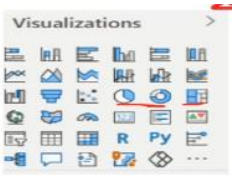


Power BI can directly integrate with Azure Services:

Power BI Desktop	**Power BI Service**
Download as a free *Windows* app And installed on a local *Windows* computer **Report designers** use the Desktop app to publish Power BI reports to the Power BI service	Cloud-based service where users view & interact with the reports **Users** in the Power BI Service can edit the reports and create visuals based on the existing data model and they can share & collaborate with co-workers

PowerBI Desktop	**Both**	**Power BI Service**
• Many data source • Transforming • Shaping & modeling • Measures • Calculated columns • Python • Themes • RLS creation	• Reports • Visualizations • Security • Filters • Bookmarks • Q&A • R Visuals	• Some data sources • **Dashboards** • Apps & workspaces • Sharing • Dataflow creation • Paginated reports • RLS management • Gateway connections

3. PowerBI Visualizations



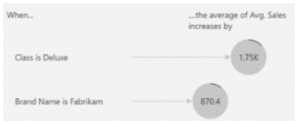
Power BI has **many kinds of visualizations**. We'll cover the most common ones.

Bar and column charts
See how a set of variables changes
Across different categories

Line charts
Overall shape of an entire series of values

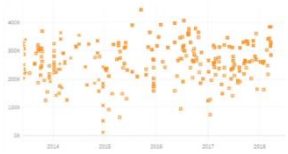
Quarter	Q1		Q2	
Year	Revenue	YTD Revenue	Revenue	YTD Revenue
2015	$45,186	$45,186	$70,609	$115,795
2016	$52,154	$52154	$73,542	$125,696
2017	$51,388	$51,388	$68,149	$118,537
2018	$48,281	$48,281	$66,853	$115,134
2019	$53,145	$53,145	$49,135	$102,280

Matrix
Tabular structure that summarizes data

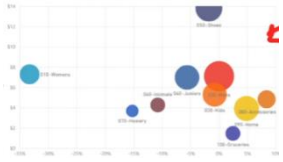


Key Influencers
The major contributors to a selected result or value

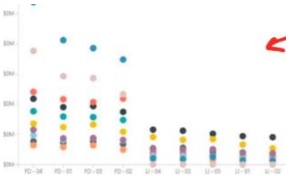
Treemap
Charts of colored rectangles, with size representing the relative value of each item



Scatter
Relationship between two numerical values (X and Y Axis)
A bunch of dots on a graph



Bubble Chart
A scatter chart that replaces data points with bubbles, larger bubbles Representing a third dimension



Dot plot Chart
Similar to a bubble chart and scatter chart, but can plot categorical data along the X-Axis

Filled map
A geographic map where different area can be filled in eg. States different colors or range of colors

4. Power BI Embedded

Azure Power BI Embedded is a platform-as-a-service (PaaS) analytics embedding solution That allows you to quickly embed **visuals**, **reports** and **dashboards** into an app.

For independent software vendors (ISV)	**For developers**
Enables you to visualize app data, rather Than building that service yourself	Embed reports and dashboards into an App for their customer

To use Azure Power BI Embedded

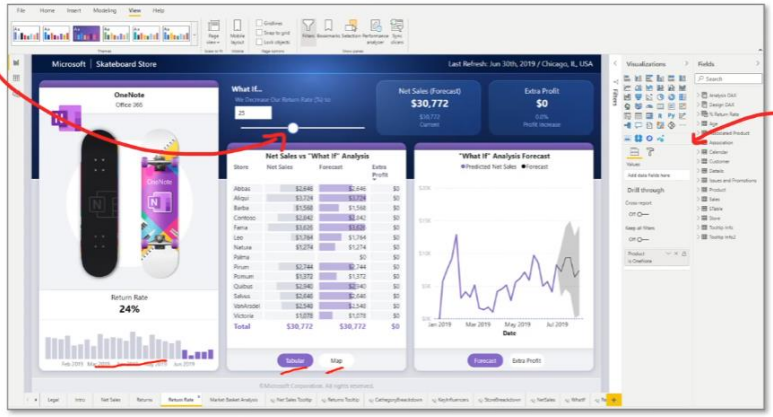
- You need a Power BI Pro user
- You need to create an App Workspace
- You need to choose a capacity
    - Billing works via a capacity-based, hourly metered mode

5. Power BI Interactive Reports

Power BI allows you to generate **reports which are interactive**

One way you can reports interactive is by **having knob and controls** directly in the report

Reports can be highly **stylized**

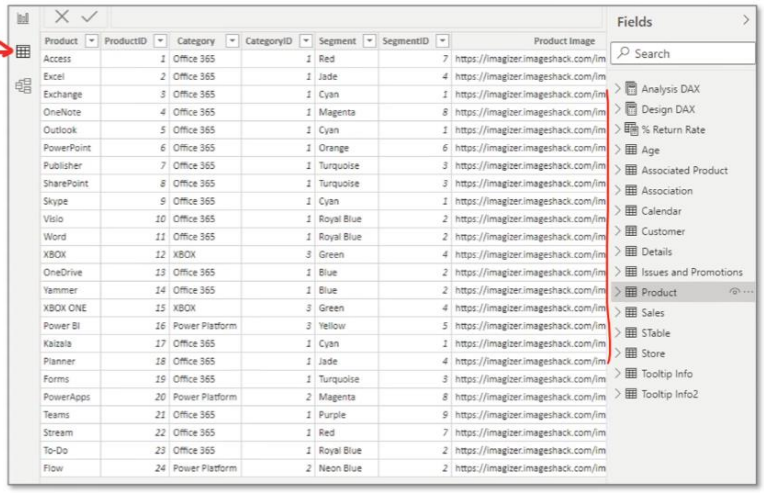


Assembling reports is as easy as choosing a **visualization** and dragging in expected fields

A report can contain may **pages**

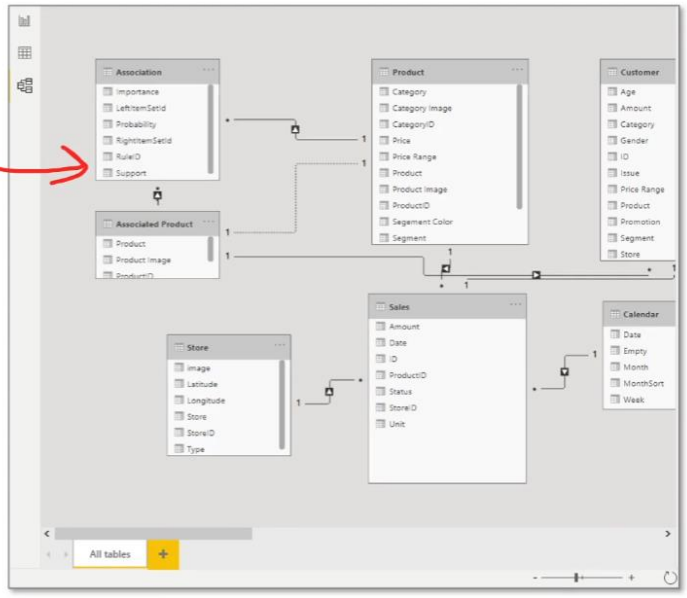
You can view the **underlying data** for an Interactive Report

(you can't do this with Dashboards)



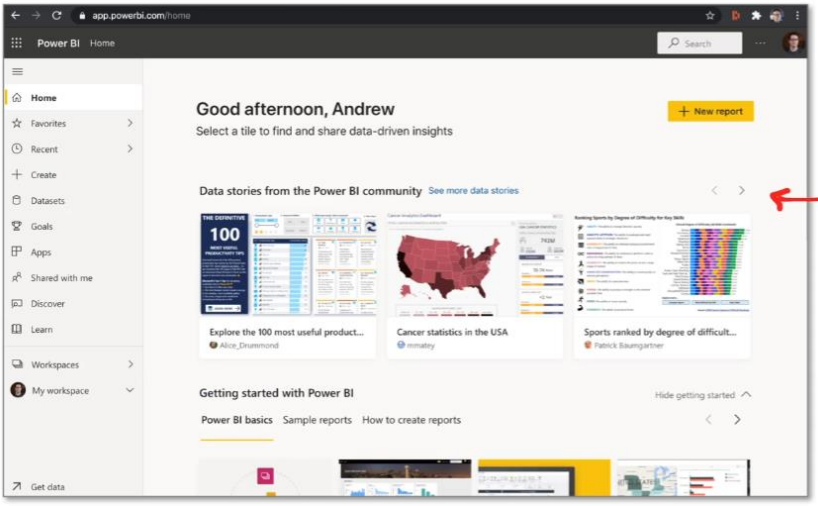
You can view **and modify the underlying data models** for a report

(you can't do this with Dashboards)



6. Power BI Service

Power BI is **cloud-based service** where the users view & interact with the reports
And where they can create **Dashboards**



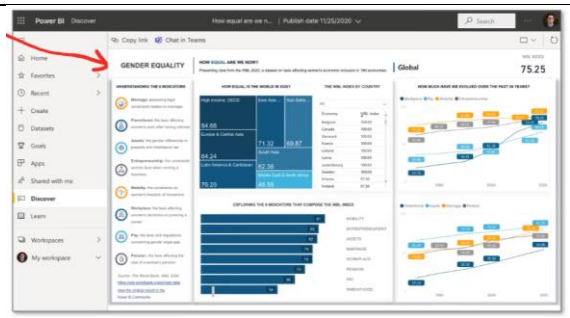
You access Power BI Service by visiting app.powerbi.com

Dashboard Tiles

A **tile** is a **snapshot of your data**, pinned to the dashboard

A tile can be created from a

- **Report**
- **Dataset**
- Dashboard
- Q&A box
- Excel
- SQL Server Report Service (SSRS) reports
- And more

Power BI dashboard is a **single page**, often called a canvas, that tells a story through visualizations The visualizations you see on the dashboard are called **tiles**. You can **pin** tiles to a dashboard from reports	

7. Reports vs Dashboards

Capability	Dashboards	Reports
Pages	One page	One or more pages
Data sources	One or more reports and one or more datasets per dashboard	A single dataset per report
Filtering	Can't filter or slice	Many different ways to filter, highlight, and slice
Set alerts	Can create alerts to email you when the dashboard meets certain conditions	No
Feature	Can set one dashboard as your featured dashboard	Can't create a featured report
Can see underlying dataset tables and fields	No. Can export data but can't see the dataset tables and fields in the dashboard itself	Yes. Can see dataset tables and fields and values that you have permissions to see
Customization	No	Can filter, export, view related content, add bookmarks, generate QR codes, analyze in Excel, and more

8. Paginated Reporting (RDL)

Paginated Reports

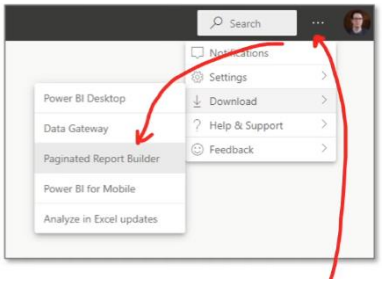
A report designed to fit into page format so they can be printed or shared. The data display of all data are tables which can span multiple pages.

Report Definition Language (RDL)

An XML representation of a SQL Server Reporting Services report definition. A report definition contains data retrieval and layout information for a report. Paginated Reports are just a visualization of an .rdl file.

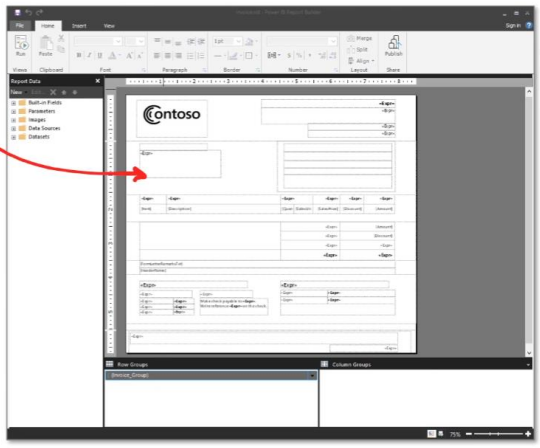
Power BI Report Builder

To design **pixel-perfect** paginated reports, you use Power BI Report Builder. It is a tool specifically for the creation of Paginated Reports.



You can download **Power BI Builder** from Power BI Service

This is what a **Paginated Report**
Looks like in Power BI Report Builder



9. Power BI Cheat Sheet

Business Intelligence (BI) – both a data-analysis strategy and **technology** for business info. Helps organizations make data-driven decisions

Power BI Desktop – A desktop app to design interactive reports from various data sources and can be published to Power BI Service

Power BI Service – A web-app to view reports, and create interactive shareable dashboards by pinning various dataset and report visualizations

Power BI Mobile – A mobile web-app to view BI reports on the go

Power BI Report Builder – Windows app build pixel-perfect printable reports (used to build paginated reports)

Power BI Embedded – embed Power BI visualizations into web-apps

Interactive Reports – Reports in Power BI, drag visualizations, load data from many data sources (Both in Desktop & Service)

Dashboards – Build sharable dashboards by pinning various Power BI visualizations (a single page report designed for a screen) *Only Service*

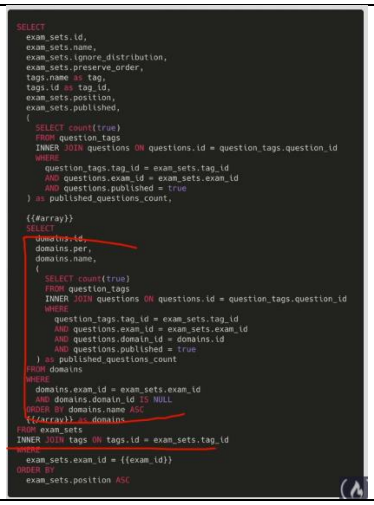
Visualizations – A visualization is a chart or graph that **is backed by a dataset**.

V. RELATIONAL DATABASES

1. Structured Query Language

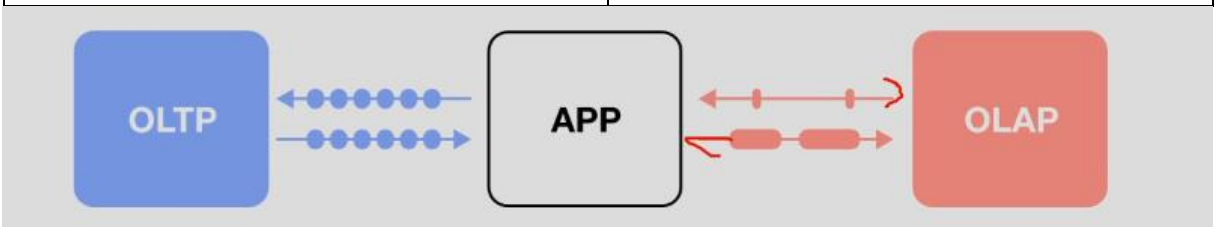
Structured Query Language (SQL) designed to **access and maintain data for a relational database management system (RDBMS)**.

We use SQL to get to **insert, update, delete, view** data from our database's tables.

SQL can join many tables and include many functions to transform the final outputted result. The SQL Syntax was standardized (ISO 9075). Relational databases will mostly adhere to this standard while adding in their **own database specific features.** SQL is a highly transferable skill & we see SQL being used in Non-Relational db to provide a popular & familiar querying tool.	

2. OLAP vs OLTP

Online **Analytical** Processing (OLAP)	Online **Transaction** Processing (OLTP)
Data Warehouse A data warehouse is built to store large quantities of historical data and **enable fast, complex queries across all the data**.	**Database** A database is built to store current transactions and enable **fast access to specific transactions for ongoing business processes**.

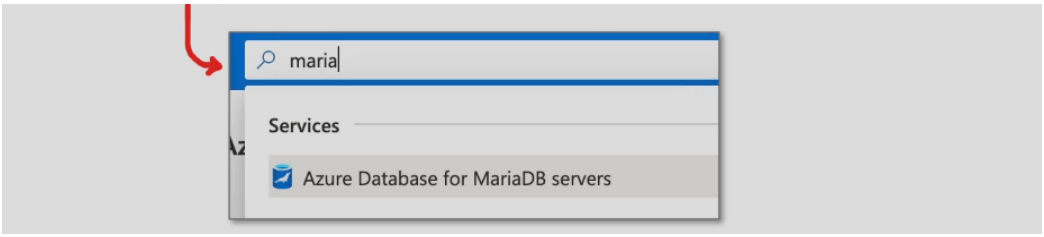


• Multiple Data Sources
• **Long transactions** (long & complex queries)
  ○ With an emphasis on reads.
• Few transactions
• Throughput sensitive
• Large payloads
Use case: (Analytics)
Generating Reports
	• Single Data Source
	• **Short transactions** (small & simple queries)
	  ○ With an emphasis on reads.
	• Many transactions
	• Latency sensitive
	• Small payloads
	Use case: (General Purpose)
	Adding items to your shopping cart

3. Open Source Relational Databases (OSRD)

MySQL®	• Created by MySQL AB then acquired by Sun Microsystems and then acquired by Oracle. • MySQL is a **pure relational database (RDBMS)** • It is a simpler db which makes it **easy to setup, use and maintain**. • Has multiple storage engines: InnoDB & Mylsam • The **most popular** relational database
MariaDB	• **MariaDB is a fork of MySQL** by the **original creators of MySQL AB**. • After Oracle acquired MySQL there was concern that Oracle may change the open-source licensing or stop future MySQL being free to use.
PostgreSQL	• PostgreSQL evolved from the **Ingres project** at the University of California • Postgres is an **object-relational db (ORDBMS)** • Just has a single store engine • The **most advanced** relational database ○ Full text search, table inheritance, triggers, rows & data types require less thought

If you want to **deploy** an open-source database on Azure you need to search by its name



4. Read Replicas

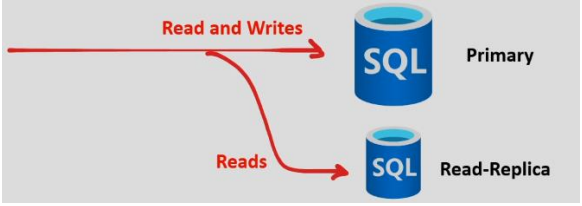
What is a read replica?

A read replica is a copy of your database that is kept synced to your primary database. This additional database is u sed to improve **read contention** by offloading reads to a database dedicated to perform read operations.

Read Replicas are available for:

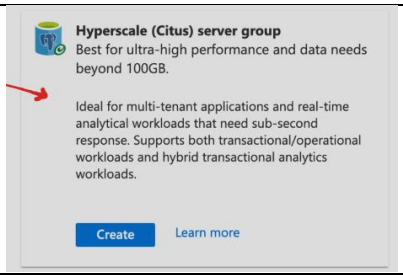
- Azure SQL Database
- Azure SQL Managed Instance

You can have multiple read-replicas for a database



In simple use cases, a Read Replica can act as an OLAP for a relational database that is very small.

5. Citus on Azure

	**Citus** is an **open-source Postgres extension** that transforms Postgres into a **distributed database**. Citus extends postgres to provide better support for: • Database sharding (easy horizontal scaling) • Realtime queries (great for real-time analytics dashboards) • Multi-tenancy (great for SaaS company's) • Time series workloads
Azure Postgres for **HyperScale deploy option** is just a managed postgres cluster that use the Citus Extension	

6. Azure SQL Family

Azure has **multiple solutions** for relational databases

	**SQL Server on Azure Virtual Machines** • When you need OS-level control access • When you need to lift-and-shift your workloads to the cloud • When you have existing SQL licenses and you want to save money via Azure Hybrid Benefit
	**SQL Managed Instance** • When you have an existing db that u want to modernize • Broadest SQL server engine computability • Highly available, disaster recovery, automated backups • Ideal for most migrations to the cloud
	**Azure SQL Database** • Fully managed SQL databases • Designed to be fault-tolerant • Built-in disaster recovery • Highly available • Designed to scale
	**SQL Servers** The underlying servers for Azure SQL Database

7. Azure Elastic Pools

Azure SQL Database elastic pools are a simple, cost-effective solution for **managing and scaling multiple databases** that have **varying and unpredictable usage demands**.

Databases in an elastic pool are on:

- A single server
- Share a set number of resources at a set price

Elastic pools in Azure SQL Database enable SaaS developers to optimize the price performance for a group of databases within a prescribed budget while delivering performance elasticity for each database.

8. Relational Databases Cheat Sheet

Structured Query Language (SQL) – designed to **access and maintain data for a relational database management system (RDBMS)**

Online Transaction Processing (OLTP) – frequent & short queries for transactional information (eg. Databases)

Online Analytical Processing (OLAP) – complex queries for large databases to produces reports & analytics (eg. Data Warehouses)

MySQL – it's a **pure relational database (RDBMS)** it is easy to setup & use, most popular open-source relational db.

MariaDB – it's a fork of MySQL

Postgres – it's an object-relational db (ORDBMS), it is more advanced & well liked among developers

Read Replicas – a duplicate of your database kept in-sync with the main to help to reduce reads on your primary databases

Azure SQL – An umbrella service for different offerings of MS SQL databases hosting services

- **SQL VMs** – for lift-and-shift when you want OS access & control, or you need to bring-your-own-license (BYOL) for Azure Hybrid Benefit
- **Managed SQL** – for lift-and-shift when you the broadest amount of compatibility with SQL versions
    - You can use Azure Arc to run this service on-premise
    - Gives you many of the benefit of a fully-managed databases
- **SQL Databases** – Fully managed SQL databases
    - Run a single server
    - Run as a database (collection of servers)
    - Run in an Elastic Pool (databases of different sizes residing on one server to save costs)

Connection Policy

- Three modes:
    1. Default – choose Proxy or Redirect initially depending on if the server is within or outside the Azure Network
    2. Proxy – outside the Azure network, proxied through a gateway
        a. Listen on **port 1443 when connecting via Proxy** mode through a gateway outside the Azure Network
    3. Redirect – redirected within the Azure Network

VI. T-SQL

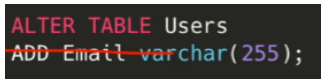
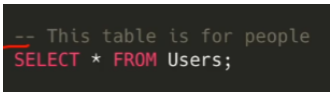
1. T-SQL

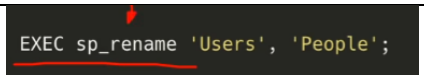
Transact-SQL (T-SQL) is a set of programming extensions from Sybase & Microsoft that add several features to the Structured Query Language (SQL).

T-SQL expands on the SQL standard to include:
• Procedural programming
• Local variables
• Various support functions for string processing
• Data processing
• Mathematics
• Changes to the DELETE & UPDATE statements
	For Microsoft SQL Server there are five groups of SQL Commands:
	• Data **Definition** Language (DDL)
	  ○ Used to define the database schema
	• Data **Query** Language (DQL)
	  ○ Used for performing queries on the data
	• Data **Manipulation** Language (DML)
	  ○ Manipulation of data in the db
	• Data **Control** Language (DCL)
	  ○ Rights, permissions and other controls of the db
	• **Transaction Control** Language (TCL)
	  ○ Transaction within the db

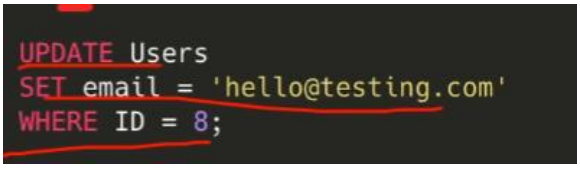
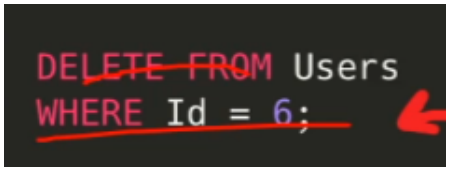
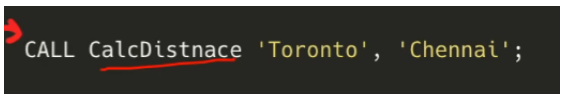
2. Data Definition Language

A Data Definition Language (DDL) is SQL syntax commands for **creating & modifying the database or database objects** (eg: table, index, views, store procedure, function & triggers).

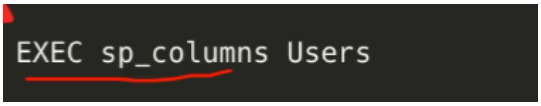
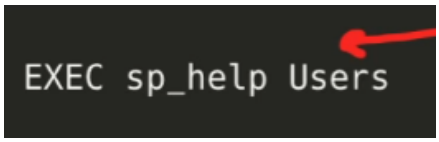
CREATE Create database or db objects to create db and its objects	
ALTER Alters the structure of an existing database	
DROP Delete objects from the database	
TRUNCATE Remove all records from a table	
COMMENT Add comments	
RENAME	

Rename a database object	

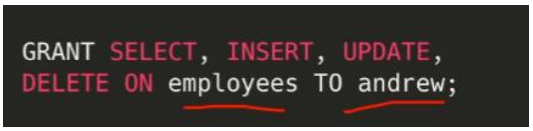
3. Data Manipulation Language (DML)

INSERT Insert data into a table	
UPDATE Updates existing data within a table	
DELETE Delete all records from a table	
MERGE – UPSERT To insert and update records at the same time	
CALL Call a PL/SQL or Java subprogram	
LOCK TABLE Concurrency control to ensure two people are not writing to the program at the same time	

4. Data Query Language (DQL)

SELECT Select data from a table, and be able to specific exactly which columns to return	
SHOW Select data from a table, and be able to specific exactly which columns to return	
EXPLAIN PLAN Returns the query plan for a Microsoft Azure Synapse Analytics SQL statement **without running** the statement	
HELP Reports information about a database object	

5. Data Control Language (DCL)

GRANT Allow users access privileges to database	
REVOKE Withdraw users access privileges given by using the GRANT command	

6. Transaction Control Language (TCL)

Transaction Control Language commands are used to manage **transactions** in the database.

COMMIT

Set to permanently save any transaction into the database

ROLLBACK

Restores the database to last committed state

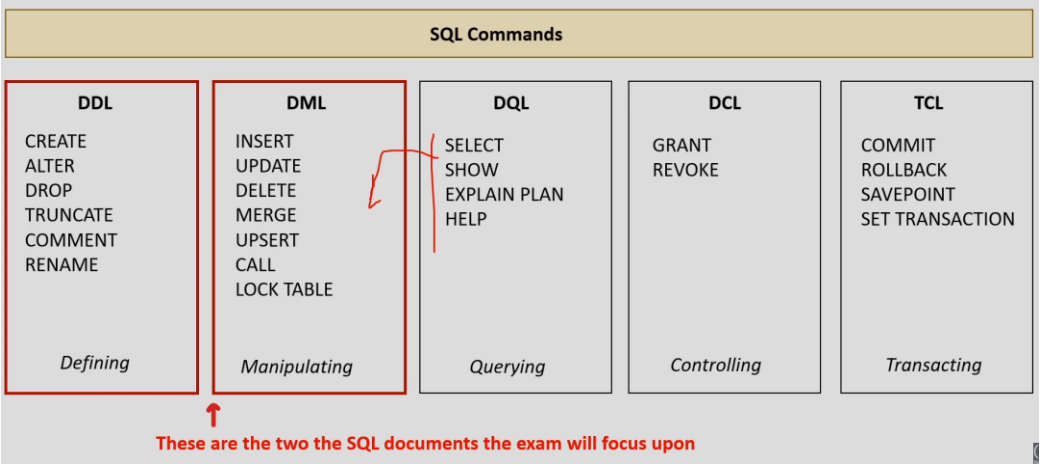
SAVEPOINT

Used to temporarily save a transaction so that you can rollback to that point whenever necessary

SET TRANSACTION

Specify characteristics for the transaction

7. MS SQL Commands



8. T-SQL Cheat Sheet

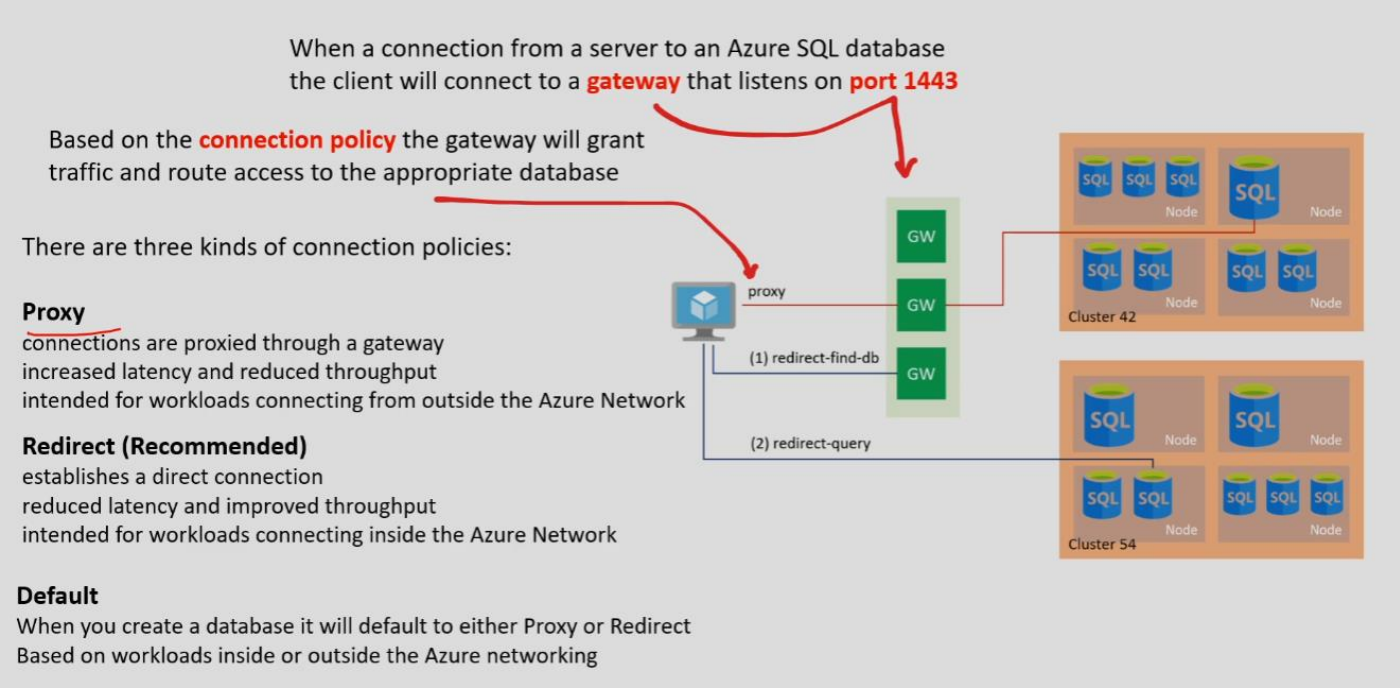
Transact-SQL (T-SQL) is a set of programming extensions from Sybase & Microsoft that add several features to the Structured Query Language (SQL).

For Microsoft SQL Server there are five groups of SQL Commands:

- Data **Definition** Language (DDL)
  - Used to define the database schema
- Data **Query** Language (DQL)
  - Used for performing queries on the data
- Data **Manipulation** Language (DML)
  - Manipulation of data in the database
- Data **Control** Language (DCL)
  - Rights, permissions and other controls of the database
- **Transaction Control** Language (TCL)
  - Transactions within the database

VII. DATABASE SECURITY

1. Connectivity Architecture



2. Database Authentication

During setup of your MS SQL database, you must select an authentication mode

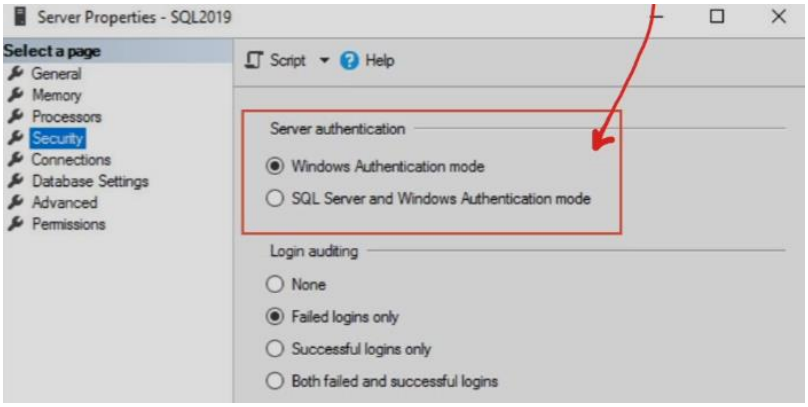
- **Windows Authentication mode**
  - Enables Windows Authentication and disables SQL Server Authentication
- Mixed mode
  - Enables both Windows Authentication and SQL Server Authentication

Windows Authentication (recommended)

- Specific Windows user and group accounts are trusted to log into SQL Server
- Very secure, and very easy to modify or revoke access

SQL Server Authentication

- A username and password are set & stored on the primary database
- Cannot use Kerberos security protocol
- Login password must be passed over the network at the time of the connection (additional attack points)
- Easier to connect to database from outside a domain or from a web-based interface



3. Network Connectivity

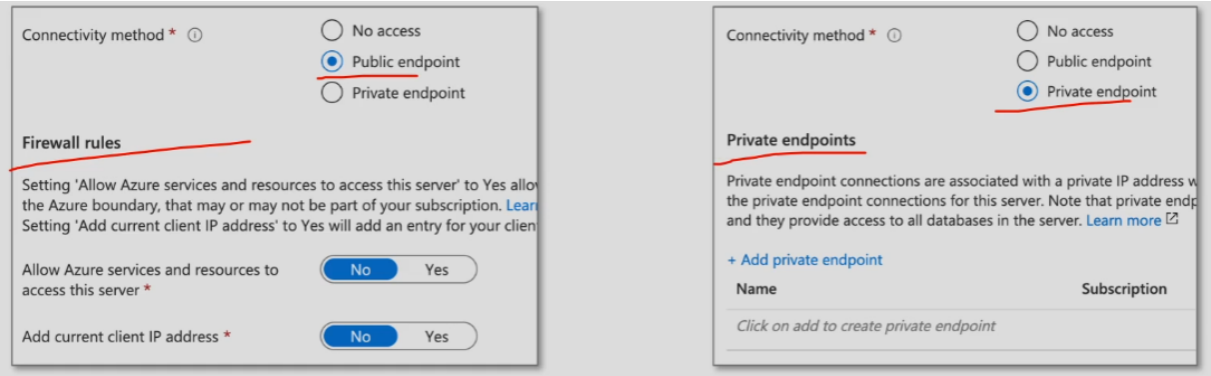
For SQL Database you can choose network connectivity to have a **Public** or **Private** Endpoint.

Public Endpoints are reachable outside the Azure Network over the internet

- You would use **Firewall rules** to project the database

Private Endpoints are only reachable within the Azure Network (or a connection originating from inside the network)

- You would use **Azure Private Link** to keep traffic within the Azure Network

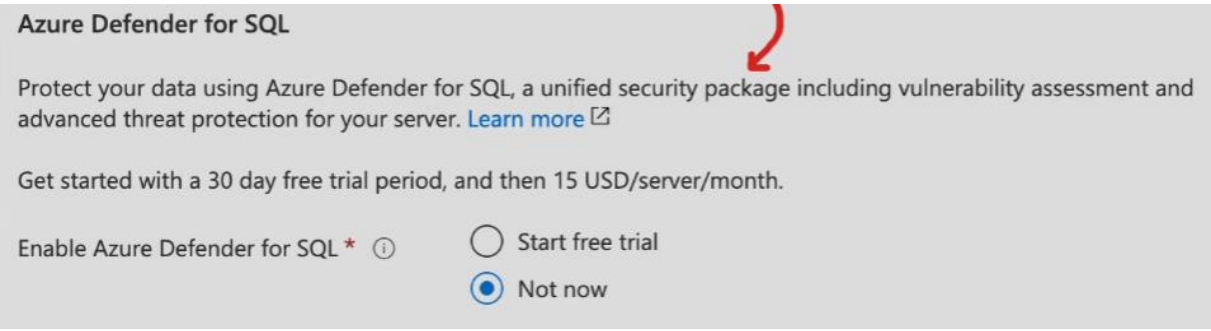


4. Azure Defender for SQL

Azure Defender for SQL is **a unified package for advanced SQL security capabilities** for **Vulnerability Assessment** and **Advanced Threat Protection**

Azure Defender is available for:	What it does:
• Azure SQL Database • Azure SQL Managed Instance • Azure Synapse Analytics	• Discovering and classifying sensitive data • Surfacing and mitigating potential db vulnerabilities • Detecting anomalous activities

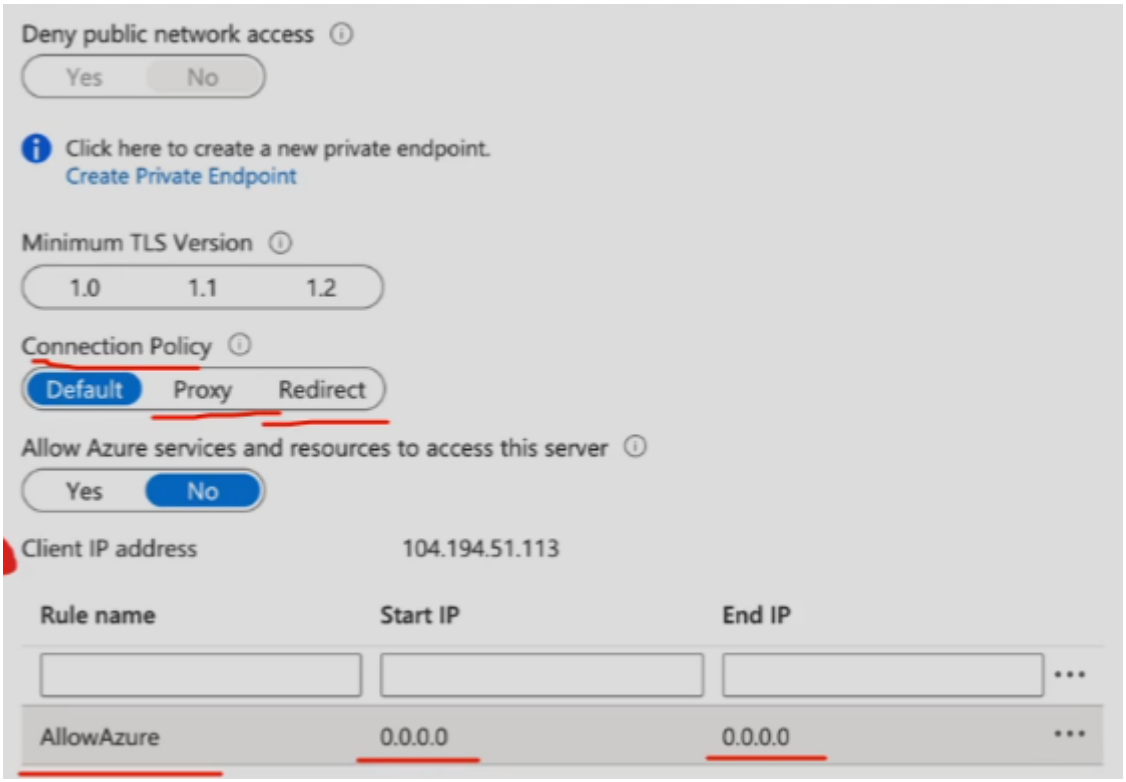
You can turn it on at anytime and you pay a monthly cost



5. Azure Database Server Firewalls

Azure databases are protected by **server firewalls**
A server firewall is an internal firewall that resides on the database server
All connections are rejected by default to database

You can set server firewall rules within the Azure Portal



You can set server firewall rules via T-SQL

```
EXECUTE sp_set_database_firewall_rule N'OnlyAllowServer','0.0.0.4','0.0.0.4';
```

6. Always Encrypted

Always Encrypted is a feature that encrypts columns in an Azure SQL Database or SQL Server

If you had a column for credit cards you would want to have it Always Encrypted

Always Encrypted uses two types of keys:

- **Column encryption keys** – used to encrypt data in an encrypted column
- **Column master keys** – a key-protecting key that encrypts one or more column encryption keys

You apply Always Encrypted using T-SQL

7. Role-Based-Access-Controls (RBAC)

Role-Based-Access-Controls (RBAC) is when you can apply roles to users to grant the fine-grade actions for specific Azure services

SQL DB Contributor

- Manage SQL databases, but not access to them
- Can't manage their security-related policies or their parent SQL servers

SQL Managed Instance Contributor

- Manage SQL Managed Instances and required network configuration
- Can't give access to others

SQL Security Manager

- Manage the security-related policies of SQL servers and databases
- But not access to SQL servers

SQL Server Contributor

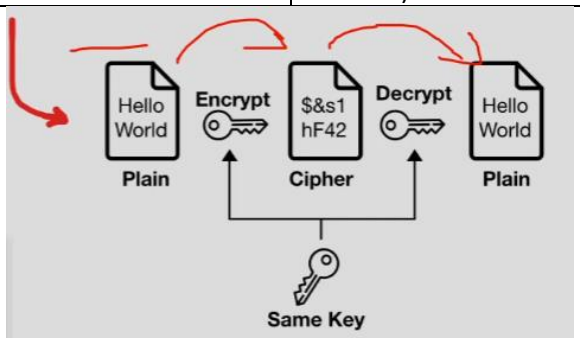
- Manage SQL servers and databases
- But not access to them SQL servers

8. Transparent Data Encryption

Transparent Data Encryption (TDE) **encrypts data-at-rest** for Microsoft Databases

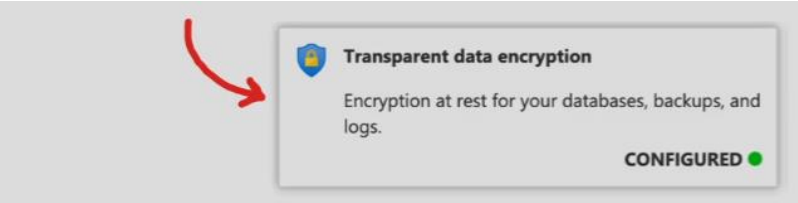
TDE can be applied to:

• SQL Server • Azure SQL Database • Azure Synapse Analytics	• TDE does real time I/O encryption & decryption of data & log files • Encryption uses a **database encryption key (DEK)** • Database boot record stores the key for availability during recovery • The DEK is a **symmetric key** (same cryptographic keys for both the encryption of plaintext and the decryption of cipher text)



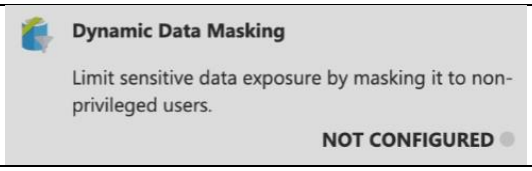
Steps to apply TDE to a database:

- Create Database Master Key
- Create a Certificate to support TDE
- Create Database Encryption Key
- **Enable TDE on Database**



9. Dynamic Data Masking (DDM)

Data Masking is when a **request for data is transformed to mask sensitive data**.

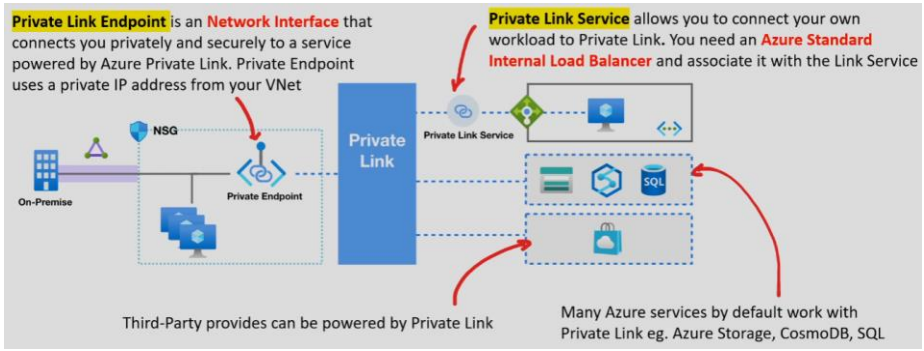
Dynamic Data Masking can be applied to: • Azure SQL Database • Azure SQL Managed Instance • Azure Synapse Analytic	

You create Dynamic Data **Masking Policy**:

- SQL users excluded from masking – **users who can get data unmasked**
- Masking rules – what fields should be masked
- Masking functions – how to apply masking to fields

10. Private Links

Azure Private Links allows you to **establish secure connections** between Azure resource so traffic **remains within the Azure network**



11. Database Security Cheat Sheet

MS SQL Data Authentication

Two modes when setting up MS SQL Server (remoted into Windows Machine):

- Windows Authentication (WA) mode – enables (WA) and disables SQL Server Authentication (SSA)
- Mixed mode – enables both WA & SSA

- Windows Authentication (recommended) – authenticate via windows users
- SQL Server Authentication – username and password, connect from anywhere

Network Connectivity

- Public Endpoint – reachable outside the Azure Network over the internet (use server firewall from protection)
- Private Endpoint – only reachable within the Azure Network (AN) (use Azure PrivateLinks to keep traffic within AN)

Azure Defender SQL – **a unified package for advanced SQL security capabilities** for **Vulnerability Assessment** and **Advanced Threat Protection**

Server Firewall Rules – an internal firewall that resides on the db server, all connections are **rejected by default** to db

Always Encrypted – a feature that encrypts columns in an Azure SQL Database or SQL Server

Role-Based-Access-Control (RBA) for databases:

SQL DB Contributor – Manage SQL db, but not access to them, can't manage their security related policies or their parent SQL servers

SQL Managed Instance Contributor – Manage SQL Managed Instances and required network configuration, can't give access to others

SQL Security Manager – Manage the security-related policies of SQL servers and db, but not access to them SQL servers.

SQL Server Contributor – Manage SQL servers and databases, but not access to them SQL servers

Transparent Data Encryption (TDE) – **encrypts data-at-rest** for Microsoft Databases

Dynamic Data Masking – you can choose your db columns to that will be masked (obscured) for specific users

Azure Private Links – allows you to **establish secure connection**s between Azure resources so traffic **remains within the Azure Network**

VIII. Azure Tables & CosmoDB

1. Key Value Store

A Key/Value stores a **unique key** alongside a value

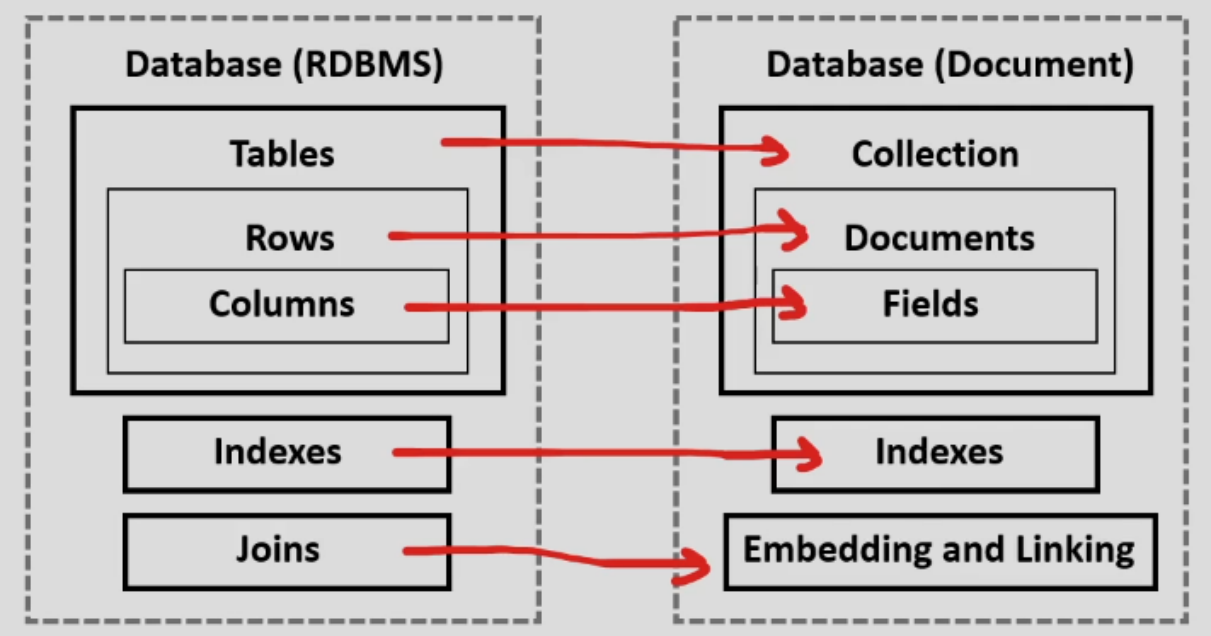
Key values stores are **dumb & fast**, they generally lack features like: • Relationships • Indexes • Aggeration	<table><tr><th>Key</th><th>Value</th></tr><tr><td>Data</td><td>10101010001010110010100101010001</td></tr><tr><td>Worf</td><td>01101011000101010101011100010</td></tr><tr><td>Ro Laren</td><td>00101010010101100101010101010</td></tr></table>
A simple Key/Value store will interpret this data resembling a dictionary (aka Associate arrays or Hash)	<table><tr><th>Key</th><th>Value</th></tr><tr><td>Data</td><td>{species: android, rank: 'lt commander' }</td></tr><tr><td>Worf</td><td>{species: klingon, rank: 'lt commander' }</td></tr><tr><td>Ro Laren</td><td>{species: bajoran, affiliation: 'maquis'}</td></tr></table>
A Key/Value store can resemble tabular data, it does not have to have the consistent columns per row (hence its Schemaless)	<table><tr><th>Key (Name)</th><th>Species</th><th>Rank</th><th>Affiliation</th></tr><tr><td>Data</td><td>andriod</td><td>Lt commander</td><td></td></tr><tr><td>Worf</td><td>klingon</td><td>Lt commander</td><td></td></tr><tr><td>Ro Laren</td><td>bajoran</td><td></td><td>maquis</td></tr></table>

Due to their simple design they can scale well beyond a relational database

2. Document Store

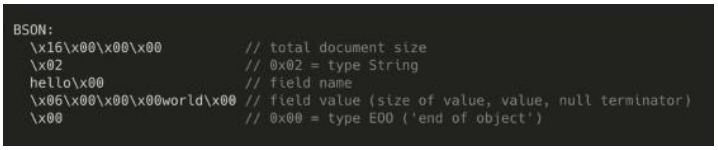
A document store is a NoSQL database that stores **documents** as its primary data structure
A document could be an XML but more commonly is JSON or JSON-Like
Document stores are sub-class of Key/Value stores

The components of a document store compared to Relational database



3. MongoDB

MongoDB is an open-source document database
Which stores JSON-like documents
The primary data structure for MongoDB is called **BSON**

Binary JSON (BSON)	
• BSON is a subset of JSON and so its data structure is very similar • BSON is designed to be **efficient both in storage and scan-speed** compared to JSON • BSON has more data-types than JSON: ○ Eg. Datetime, byte arrays, regular expressions, MD5 binary data, JavaScript code	

What it looks like to perform an operation on a MongoDB databases →

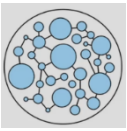
```
db.inventory.insertMany([
   { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
   { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
   { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
   { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

- MongoDB supports searches against:
  - Fields
  - Ranged queries
  - Regular-expressions

- MongoDB supports **primary** and **secondary** indexes
- High availability can be obtained via replica sets (replica to offload reads or acts a stand-by in case of failover)
- MongoDB scales horizontally using sharding
- MongoDB can run over multiple servers via load balancing
- MongoDB can be used as a file system, called **GridFS**
  - With load balancing and data replication features over multiple machines for storing files
- MongoDB provides three ways to perform aggregation (grouping data during a query)
  - Aggregation pipeline
  - Map-reduce
  - Single-purpose aggregation

- MongoDB supports fixed-size collections called capped collections
- MongoDB claims to support multi-document ACID transactions

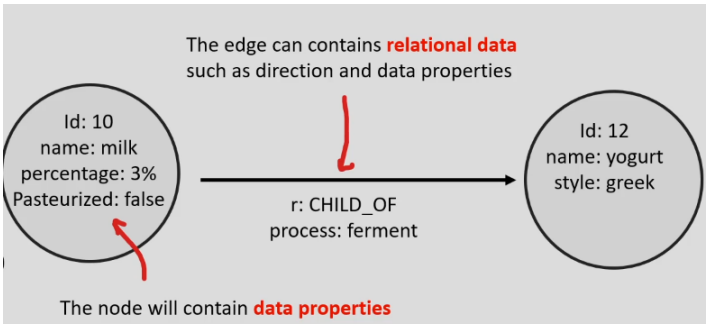
4. Graph Database



A graph database is a database composed of a data structure that uses vertices (nodes, dots) which form **relationship** to other vertices via edges (arcs, lines)

Use cases for Graph Database:

- Fraud detection
- Real-time recommendation engines
- Master data management (MDM)
- Network and IT operations
- Identity and access management (IAM)
- Traceability in Manufacturing
- Contact Tracing
- Data Lineage for GDPR
- Customer 360-degree analysis (marketing)
- Product recommendations
- Social Media graphing
- Feature Engineering (ML)



5. Apache TinkerPop and Gremlin

Apache TinkerPop is a **graph computing framework** for both Graph databases (OLTP) and graph analytic systems (OLAP)

TinkerPop enables developers to use **a vendor-agnostic distrusted framework** to traverse (query) many different graph systems:

Alibaba Graph Database	**Hadoop (Spark)**	**OrientDB**
Amazon Neptune	HGraphDB	OverflowDB
ArangoDB	Huawei Graph Engine Service	Apache S2Graph
Bitsy	HugeGraph	Sqlg
Blazegraph	IBM Graph	Stardog
CosmosDB	JanusGraph	TinkerGraph
ChronoGraph	JanusGraph (Amazon)	**Titan**
DSEGraph	**Neo4j**	Titan (Amazon)
GRAKN.AI	neo4j-gremlin-bolt	Titan (Tupl)
		Unipop

TinkerPop includes a graph traversal language called **Gremlin**
Which is the single language that can be used for all these graph systems

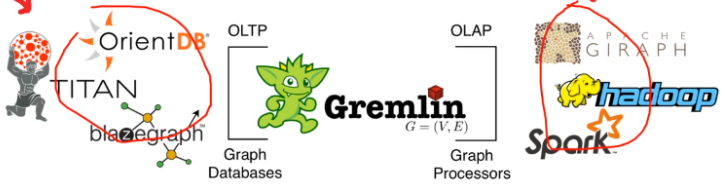
6. Gremlin

Gremlin is the **graph traversal language** for **Apache TinkerPop**

```
g.V().has("name","gremlin").as("a").
  out("created").in("created").
    where(neq("a")).
  groupCount().by("title")
```

Gremlin is design to the "Write once, run anywhere" (**WORA**)
Gremlin traversal can be evaluated as

- **real-time database query (OLTP)**
- or as a **batch analytics query. (OLAP)**



Gremlin Host Language Embedding means You can use your favourite programming language to write Gremlin

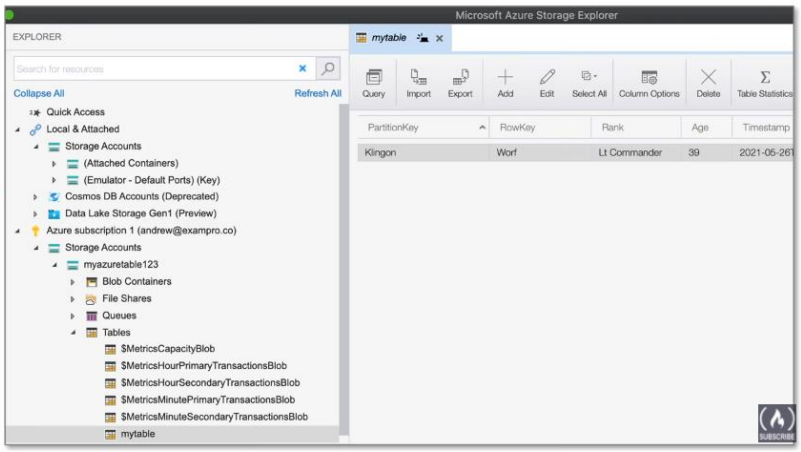
7. Azure Tables

Azure Table storage is a a **NoSQL key/value datastore** within Azure Storage Accounts

Azure Table stores **non-relational** structured data with a schemaless design

There are two ways to interact with Azure Tables:

- Azure Table Storage API
- **Microsoft Azure Storage Explorer**



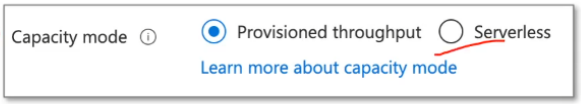
8. Azure CosmoDB

Azure CosmoDB is a service for fully-managed NoSQL databases
that are designed to scale and high performance

CosmoDB supports **different kinds** of NoSQL database engine which you interact via an API:

- **Core SQL** (**document** datastore)
- Azure Cosmos DB API for **MongoDB** (**document** datastore)
- **Azure Table** (**key/value** datastore)
- **Gremlin** (**graph** datastore)
  - Based on **Apache TinkerPop**

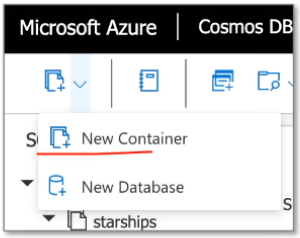
All of these NoSQL engines specific **capacity**:
- Provisioned throughput (pay for guarantee of capacity)
- Serverless (pay for what you use)

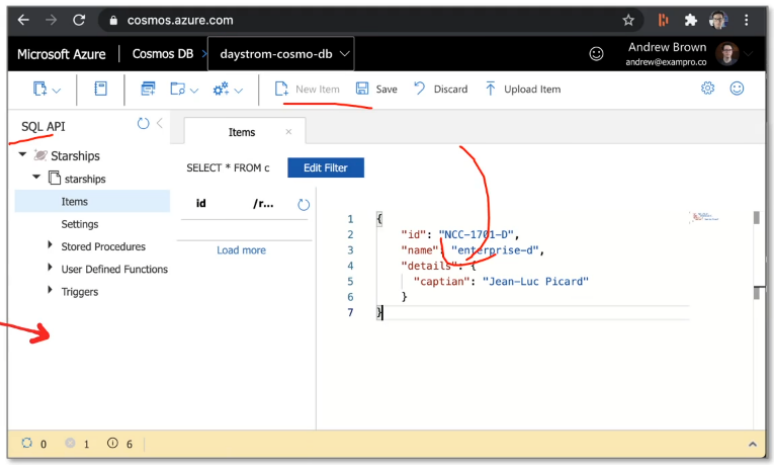


9. CosmoDB Explorer

CosmoDB Explorer is **a web interface** to explore and interact with your CosmoDB accounts

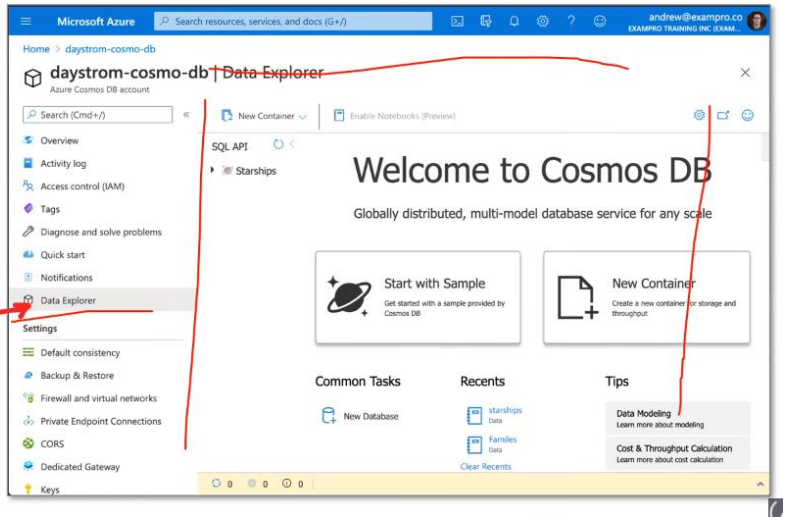
cosmos.azure.com



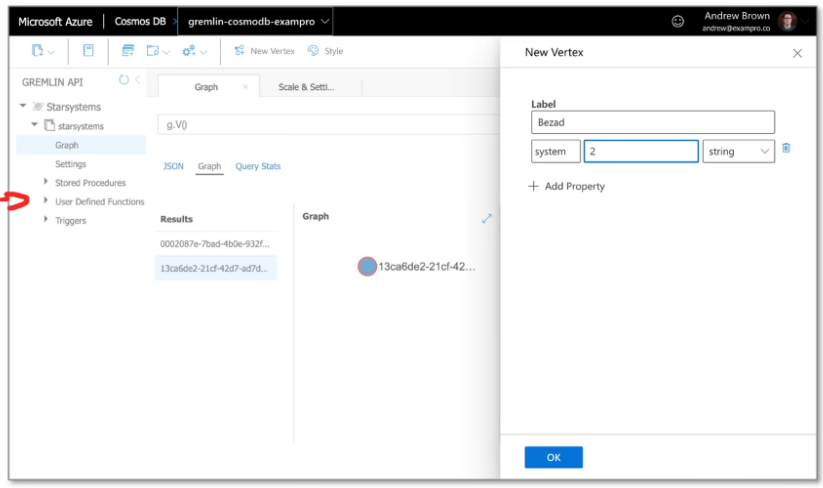


Here is a **CosmoDB Core SQL**
In CosmoDB Explorer adding a
document to the database

When navigating within Azure to a
ComsoDB Account under **Data Explorer**
is the same as CosmoDB Explorer



Using CosmoDB Explorer
With a **Graph Database**
(Gremlin)



10. Azure Tables Account Storage vs CosmoDB

When comparing Account Storage Azure Table vs CosmoDB Table API

Feature	**Azure Table Storage**	**Azure Cosmos DB Table API**
Latency	Fast, but no upper bounds on latency.	Single-digit millisecond latency for reads and writes
Throughput	Variable throughput model limit of 20,000 operations/s.	Guaranteed backed by SLAs. No upper limit on throughput
Global distribution	Single region with one optional readable secondary read region	30+ regions
Indexing	Only primary index on PartitionKey and RowKey. No secondary indexes.	Automatic and complete indexing on all properties, no index management.
Query	Query execution uses index for primary key, and scans otherwise.	Queries can take advantage of automatic indexing on properties for fast query times.
Consistency	Strong within primary region Eventual within secondary region.	Five well-defined consistency levels
Pricing	Consumption-based	consumption-based or provisioned capacity
SLAs	99.99% availability	99.99% availability SLA (some conditions does not)

11. Azure Tables & CosmoDB Cheat Sheet

Azure Tables – a key/value data store

- Can be hosted on Account Storage, its designed for a single region & single table
- Can be hosted on CosmoDB, its designed for scale across multiple regions

CosmoDB – a fully-managed NoSQL service that **supports multiple NoSQL engines** called APIs

- Core SQL API (default) – a document database, you can use SQL to query documents
- Graph API – a graph db that you can use in Gremlin to traverse the nodes and edges
- MongoDB API – a MongoDB database (document db)
- Tables API – Azure Tables Key/Value

Apache TinkerPop – an open-source framework to have an agnostic way to talk to many graph db

- Gremlin – Graph traversal language to traverse nodes & edges

MongoDB – an open-source document db

- Binary JSON (BSON) – A storage & compute optimized version of JSON, introduces new data types

CosmoDB Explorer – a web-UI to view cosmos db.

IX. Azure Tables & CosmoDB

1. Apache Hadoop

Hadoop is an open-source framework for **distributed processing of large data sets**

Hadoop allows you to distribute:
- Large dataset across many servers (eg. HDFS)
- Computing queries across many servers (eg. MapReduce)

These computer servers do not need to be specialized hardware and can run on common hardware

The Apache Hadoop framework has the following:
- Hadoop Common – collection of common utilities & libraries that support other Hadoop modules
- Hadoop Distributed File System (HDFS) – a resilient and redundant file storage distributed on clusters of common hardware
- Hadoop MapReduce – writes apps that can process multi-terabyte data in-parallel on large clusters of common hardware
- Hbase – a distributed, scalable, big data store
- YARN – manages resources, nodes, containers & performs scheduling
- HIVE – used for generating reports using an **SQL** language
- PIG – a high-level **scripting** language to write complex data transformations

Hadoop can integrate with many other open-source projects via **Hadoop components**

2. Apache Kafka

Apache Kafka is an **open-source streaming platform** to create **high-performance data pipelines,** streaming analytics, data integration, and mission-critical applications. *Kafka was originally developed by LinkedIn, and open-sourced in 2011.*

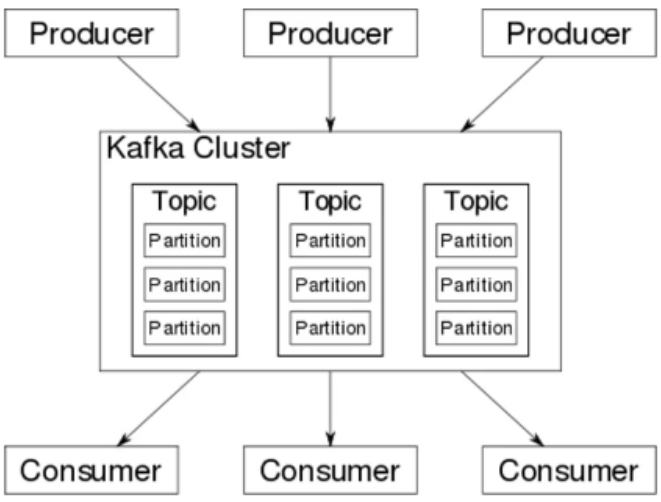
Kafka was written in Scale and Java.
To use Kafka, you need to write **Java** code.

In Kafka data is stored in partitions on a Kafka Cluster which can span multiple machines (distributed computing)

Producers publish message in a key/value format using the Kafka Producer API

Consumers can listen for messages and consume them using the Kafka Consumer API

Messages are organized into **Topics**. Producers will push message to topics and consumers will listen on topics.



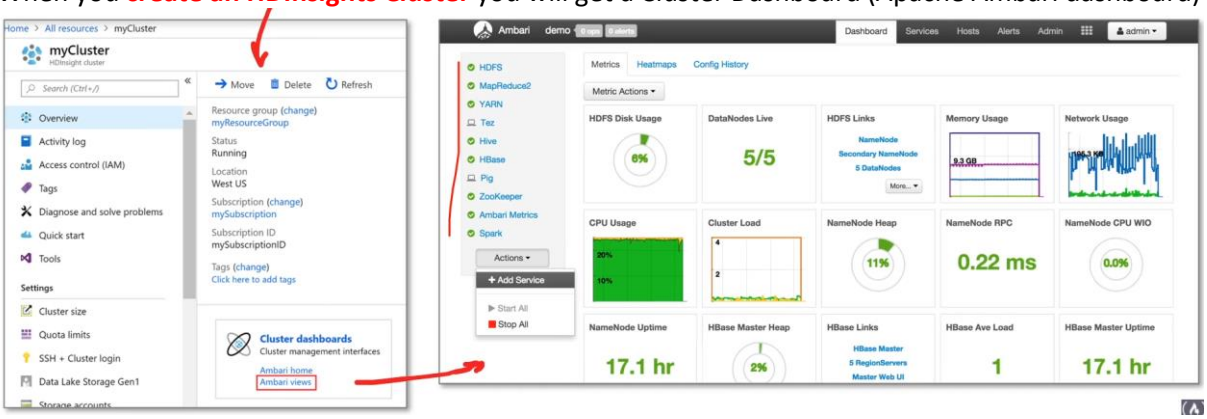
3. Azure HDInsights

Azure HDInsight is a **managed service to run popular open-source analytics service**

HDInsights supports the following frameworks:	HDInsights has broad range of scenarios such as:
• **Apache Hadoop** • Apache Spark • Apache Storm • Apache Hive • Apache Hbase • Low Latency Analytical Processing (LLAP) • R	• Extract, Transform and Load (ETL) • Data Warehousing • Machine Learning • Internet of Things (IoT)

Apache Ambari is an **open-source Hadoop management web-portal** for **provisioning**, **managing** and **monitoring** Apache Hadoop clusters

When you **create an HDInsights Cluster** you will get a Cluster Dashboard (Apache Ambari dashboard)



4. Hadoop Cheat Sheet

Apache Hadoop – open-source framework for distributed processing of large data sets

- **Hadoop Distributed File System (HDFS)** – a resilient and redundant file storage distributed on clusters of common hardware
- **Hadoop MapReduce** – writes apps that can process multi-terabyte data in-parallel on large clusters of common hardware
- **Hbase** – a distributed, scalable, big data store
- **YARN** – manages resources, nodes, containers and performs scheduling
- **HIVE** – used for generating reports using an **SQL** language
- **PIG** – a high-level **scripting** language to write complex data transformations
- **Apache Spark** – can perform 100x faster in memory and 10x faster than disk than Hadoop, supports ETLs, Streaming and ML flows
- **Apache Kafka** – a streaming pipeline and analytics service
- **HDInsights** – is a managed service to run popular open-source analytics service. It is fully-managed Hadoop system

X. Azure and Databricks

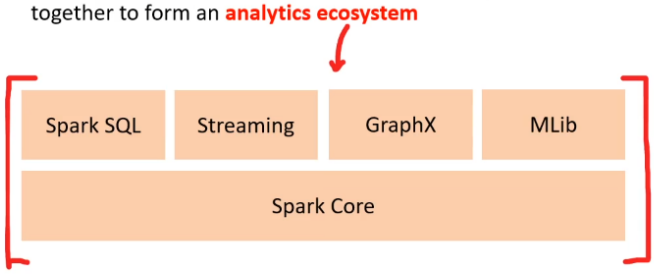
1. Apache Spark

Apache Spark is an open-source **unified analytics engine** for **big data and machine learning**

Spark lets you run workloads much faster than Hadoop:
- 100x faster in memory
- 10x faster than disk

Which is why Spark is described as ***lightning fast***

Apache Spark is a collection of libraries that work well together to form an **analytics ecosystem**



Spark Core
The underlying engine and API. The API supports the following programming languages:
R, SQL, Python, Scala, Java

Spark SQL
Introduces a data structure called a DataFrame which can be used with DSL to work with structure and semi-structured data

Spark Streaming
Allows Spark to ingest data from many streaming services:
HDFS, Flume, Kafka, Twitter, Kinesis

GraphX
distributed graph-processing framework

Machine Learning Library (MLib)
a distributed machine-learning framework which common machine learning and statistical algorithms

2. Apache Spark – RDD API

Resilient Distributed Dataset (RDD) is a domain specific language (DSL) to execute various parallel operations on an Apache Spark cluster

Common RDD API functions

map
flatMap
mapPartitions
filter
distinct
reduce
count
first
take
countByValue
sortBy
groupBy
fold

union
add
substract
intersection
saveAsTextFile
saveAsHadoopFile
saveAsPicketFile
min
max
mean
status
parallelize

Example of RDD API

```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```

3. Databricks platform

Databricks is a software company specializing **in providing fully managed Apache Spark clusters**. The company founders were the creators of Apache Spark, Data Lake and MLFlow

Databricks has two offerings:

Databricks Platform — Databricks cloud-based Spark platform with an ease-to-use web UI

- Launch fully managed Spark clusters
- Launch notebooks to write code and interact with Spark
- Create workspaces to collaborate with team members
- Role Base Access Controls
- Create jobs for ELT or data analysis tasks that run immediately or on a schedule
- Create MLFlow Workflows
- **Available on all main cloud service providers eg. AWS, Azure, GCP**

Databricks Community Edition – free version of Databricks Platform for educational user.

- Create a free micro-cluster that terminates after 2 hours when idle
- No workspace, jobs or RBAC

Azure Databricks is a **partnership between Microsoft and Databricks** to offer the **Databricks Platform within the Azure Portal** running on Azure compute services

Azure Databricks offers two environments:

Azure Databricks Workspace

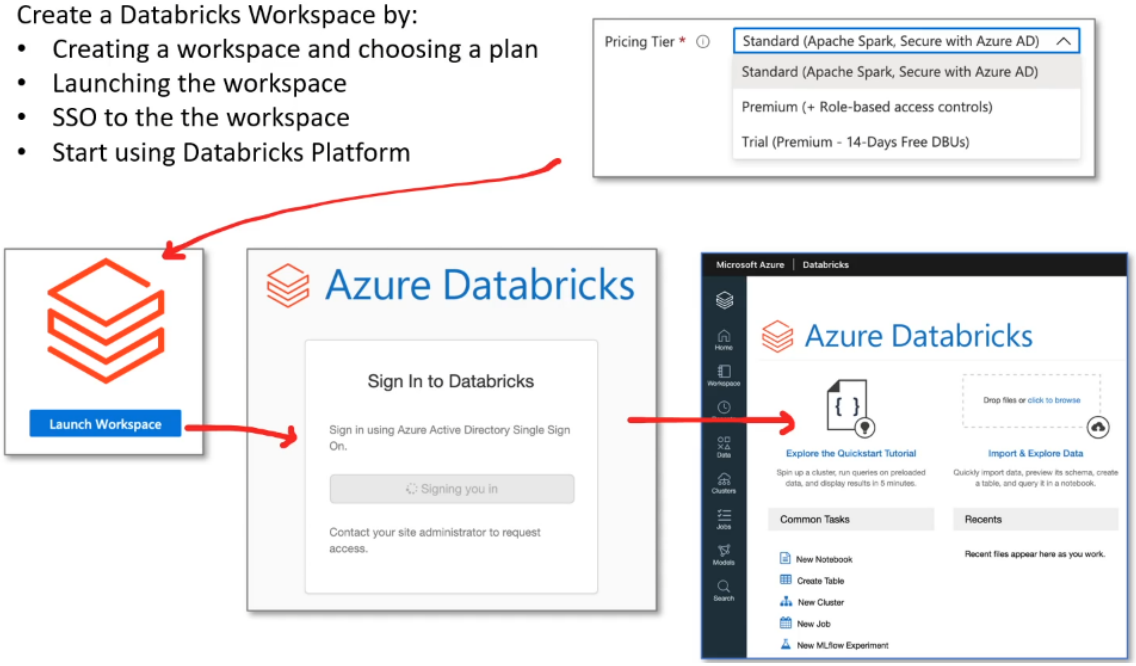
- The DataBrick Platform with integrations to **Azure data-related services** for building big data pipelines.
  - Batching: Azure Data Factory
  - Streaming: Apache Kafka Event Hub, or IoT Hub
  - Storage: Azure Blob Storage or Azure Data Lake Storage

Azure Databricks SQL Analytics

- run SQL queries on your data lake
- create multiple visualization types to explore query results
- build and share dashboards

Create a Databricks Workspace by:

- Creating a workspace and choosing a plan
- Launching the workspace
- SSO to the the workspace
- Start using Databricks Platform



4. Azure and Databricks Cheat Sheet

Apache Spark – an open-source **unified analytics** engine for **big data and machine learning**

- 100x faster in memory than Hadoop
- 10x faster in disk than Hadoop
- Perform ETL (batch), streaming and ML workloads
- The Apache ecosystem is composed of:
    - **Spark Core** – The underlying engine and API
    - **Spark SQL** – Use SQL and also new data structure called DataFrame to work with data
    - **Spark Streaming** – ingest data from many streaming services
    - **GraphX** – distributed graph-processing framework
    - **Machine Learning Library (MLib)** – a distributed machine-learning framework
    - **Resilient Distributed Dataset (RDD)** is a domain specific language (DSL) to execute various parallel operations on an Apache Spark cluster.

Databricks is a software company specializing in **providing fully managed Apache Spark clusters**. **Azure Databricks** is **a partnership between Microsoft and Databricks** to offer **the Databricks platform within the Azure Portal** running on Azure computer services

Azure Databricks offers two environments:

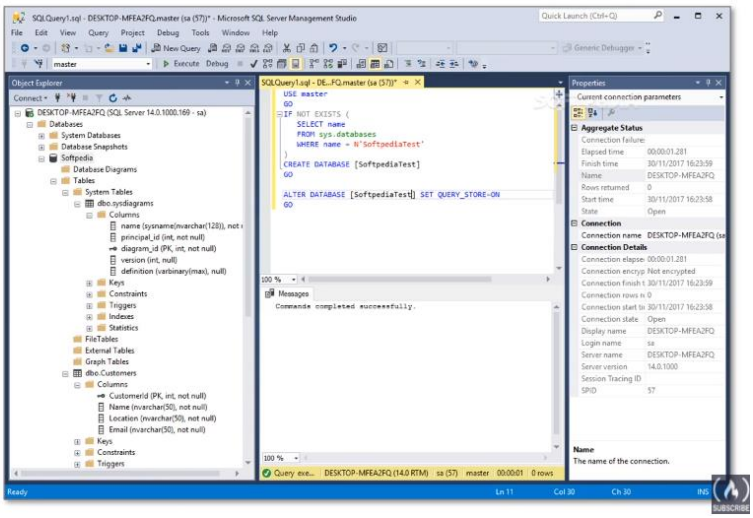
- **Azure Databricks Workspace** – Databricks Platform with integrations to **Azure data-related services** for building big data pipelines.
- **Azure Databricks SQL Analytics** – Run query in your data lake

XI. ELT and SQL tools

1. SQL Server Management Studio (SSMS)

SSMS is an IDE for **managing any SQL infrastructure**

Access, configure, manage, administer, and develop all components of
- SQL Server
- Azure SQL Database
- Azure Synapse Analytics



Object Explorer
view and manage all of the objects in one or more instances of SQL Server

TemplateExplorer
build and manage files of boilerplate text that you use to speed the development of queries and scripts

Solution Explore *(deprecated)*
build projects used to manage administration items such as scripts and queries

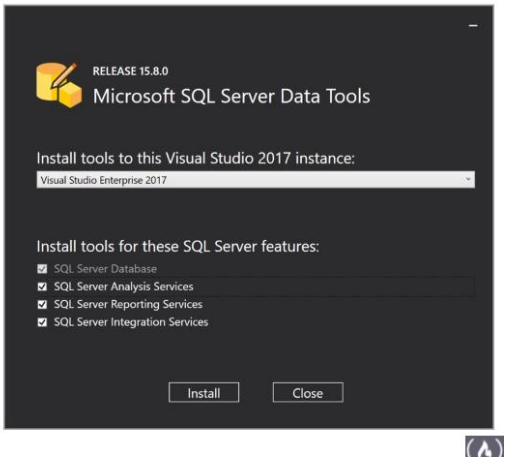
2. SQL Data Tools (SSDT)

SSDT transforms database development by introducing a ubiquitous, declarative model that spans all the phases of database development inside Visual Studio

use SSDT Transact-SQL to build, debug, maintain, and refactor databases.

SSDT also provides a visual Table Designer for creating and editing tables in either database projects or connected database instances
Be able to view control data related files
Easy to publish to SQL Database or SQL Server

SQL Server Object Explorer in Visual Studio offers a view of your database objects similar to SQL Server Management Studio (SSMS)
- allows you to do light-duty database administration and design work
- easily create, edit, rename and delete tables, stored procedures, types, and function
- edit table data, compare schemas, or execute queries by using contextual menus right



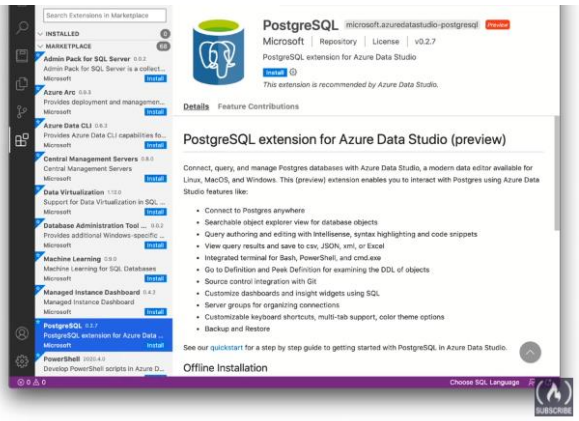
3. Azure Data Studio

Azure Data Studio is a **cross-platform database tool** for data professionals using **on-premises** and **any cloud data platforms** for **Windows, macOS and Linux**.

Query, design and manage your databases and data ware houses

Azure Data Studio offers:

- a modern editor experience with IntelliSense
  - **Very similar experience to Visual Studio Code.**
- code snippets
- source control integration
- integrated terminal
- built-in charting of query result sets
- customizable dashboards
- Jupyter Notebooks connected to your datasets
- A marketplace of free extensions
  - SQL Database Inspector (inspect data with just a few clicks)
  - Kusto (KQL) extension for Azure Data Studio
  - PostgreSQL extension for Azure Data Studio
  - And many many more!

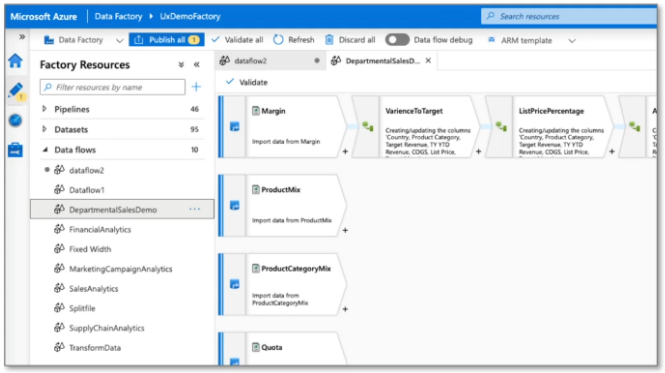


4. Azure Data Factory

Azure Data Factory is a managed service for **ETL, ELT and data integration**

Create **data-driven workflows** for orchestrating **data movement** and **transforming data** at scale

- Create Pipelines to schedule data-driven workflows
- build complex ETL processes that transform data visually with data flows
- using compute services such as Azure HDInsight Hadoop, Azure Databricks, and Azure SQL Database
- publish your transformed data to data stores such as Azure Synapse Analytics
- raw data can be organized into meaningful data stores and data lakes



Pipelines
a logical grouping of activities that performs a unit of work

Activities
A processing step in a pipeline

Datasets
data structures within the data store

Linked services
define the connection information for data sources to connect to Data Factory

Data Flows
logic to determine how data moves through a pipeline or is transformed

Integration Runtimes (RI)
compute infrastructure used by Azure Data Factory

Control flow
orchestration of pipeline activities that includes chaining activities in a sequence, branching

5. Microsoft SQL Server Integration Services (SSIS)

SSIS is a platform for building **enterprise-level data integration and data transformations** solutions

You can perform the following tasks with SSIS
- Copy files
- Download files
- Loading data into data warehouses
- Cleansing data
- Mining data
- managing SQL Server objects
- managing SQL Server data

Perform ELT with variety of sources:
- XML
- Flat files
- Relational data sources

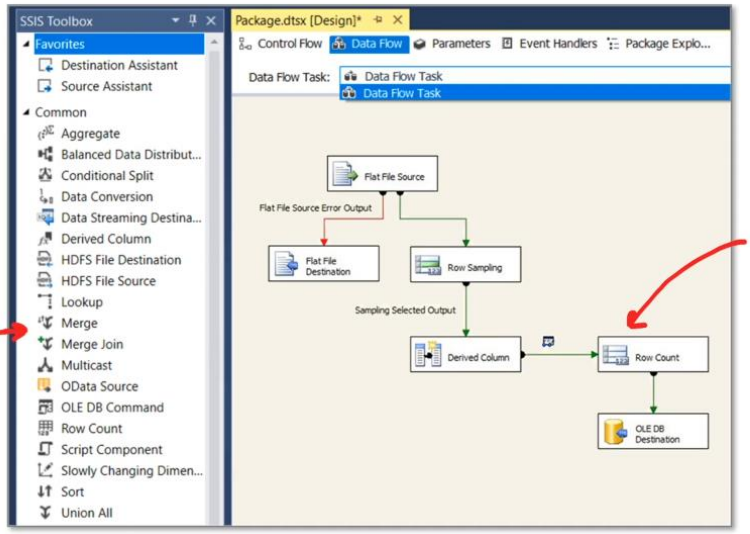
SSIS can be used to automate SQL Server databases
SSIS can be used as an integration runtime in Azure Data Factory

SSIS has…
- built-in tasks and transformations
- graphical tools for building packages
- Integration Services Catalog database
  - where you store, run, and manage packages

Use **Graphical Integration Services tools** for to integrate and transform data without having to write code

SSIS Designer is a graphical tool that you can use to create and maintain Integration Services packages.

SSIS allows you to drag out data transformations



Design different kinds of control or data flows

6. ETL and SQL Tools Cheat Sheet

Azure Data Factory is a managed service for **ETL, ELT and data integration**

- Create **data-driven workflows** for orchestrating **data movement** and **transforming data** at scale
- Build ELT pipelines visually without writing any code via a web-interface

SQL Server Integration Services (SSIS) – a platform for building **enterprise-level data integration and data transformations** solutions

- A low-code tool for building ELT pipelines, very similar to Azure Data Factory but existed 15 years prior
- Integrates with Azure Data Factory

Azure Data Studio – an IDE similar to Visual Studio Code, that is cross-platform and works with SQL and non-relational data, has many extensions.

SQL Server Management Studio (SMSS) – an IDE for **managing any SQL infrastructure** that only works for Windows. More mature than Data Studio.

SQL Server Data Tools (SSDT) – Visual studio extension to work and design visually SQL databases