What is data?

Data is a collection of facts such as numbers, descriptions, and observations used in decision making. You can classify data as structured, semi-structured, or unstructured.

Structured: relational tables stored in SQL Server or Azure SQL Database. The act of setting up the database server is called provisioning.

Semi structure: json and key-value stores and graph databases. Stored in  Azure Cosmos DB.

Unstructured: audio and video files, and binary data files stored in Azure Blob storage (Blob is an acronym for Binary Large Object).

System access

* Read-only access means the users can read data but can't modify any existing data or create new data. Management
* Read/write access gives users the ability to view and modify existing data. Such as salesperson app to record sales
* Owner privilege gives full access to the data including managing the security like adding new users and removing access to existing users. Data analysts and data managers

Data processing

Analytical systems, and transaction processing systems.

The primary function of business computing is transactional system. Online Transactional Processing (OLTP), high volume of data

Splitting tables for separate information is called normalisation, Normalization can enable a transactional system to cache much of the information required to perform transactions in memory, and speed throughput. Querying may become more complex due to several tables

An analytical system is designed to support business users who need to query data and gain a big picture view of the information held in a database. It is used in data ingestion, data transformation, data querying, and data visualization.

**Data Ingestion**: Data ingestion is the process of capturing the raw data from different OLTP system. To process and analyze this data, you must first store the data in a repository of some sort. The repository could be a file store, a document database, or even a relational database.

**Data Transformation/Data Processing**:

**Data Querying**

**Data Visualization**

**Document database stores individually**

A transaction is a sequence of operations that are atomic must all be successful or re run

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

An atomic system must guarantee atomicity in each and every situation, including power failures, errors, and crashes.

Consistency ensures that a transaction can only take the data in the database from one valid state to another

Isolation ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially. A concurrent process can't see the data in an inconsistent state

* *Durability* guarantees that once a transaction has been committed, it will remain committed even if there's a system failure such as a power outage or crash.

Distributed databases(minimise latency but leads to inconsistencies) uses eventual consistency as against isolation ideal where the application doesn't require any ordering guarantees

Data Engineer

As a SQL Server professional, your primary data manipulation tool might be Transact-SQL. As a data engineer you might use additional technologies, such as [Azure Databricks](https://docs.microsoft.com/en-us/azure/azure-databricks/what-is-azure-databricks), and [Azure HDInsight](https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-overview) to generate and test predictive models. If you're working in the non-relational field, you might use [Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/introduction) as your primary data store. To manipulate and query the data, you might use languages such as HiveQL, R, or Python.

The example below uses Transact-SQL commands to create a new database (CREATE DATABASE and ALTER DATABASE commands are part of the Transact-SQL dialect, and aren't part of standard SQL). The script then creates a new table named Customers, and inserts four rows into this table. Again, the version of the INSERT statement, with four VALUES clauses, is part of the Transact-SQL dialect. The -- characters start a comment in Transact-SQL. The [ and ] characters surround identifiers, such as the name of a table, database, column, or data type. The N character in front of a string indicates that the string uses the Unicode character set.

**IF NOT EXISTS (**

**SELECT name**

**FROM sys.databases**

**WHERE name = N'TutorialDB'**

**)**

**CREATE DATABASE [TutorialDB];**

**GO**

**ALTER DATABASE [TutorialDB] SET QUERY\_STORE=ON;**

**GO**

**-- Switch to the TutorialDB database**

**USE [TutorialDB]**

**GO**

**-- Create a new table called 'Customers' in schema 'dbo'**

**-- Drop the table if it already exists**

**IF OBJECT\_ID('dbo.Customers', 'U') IS NOT NULL**

**DROP TABLE dbo.Customers;**

**GO**

**-- Create the table in the specified schema**

**CREATE TABLE dbo.Customers**

**(**

**CustomerId INT NOT NULL PRIMARY KEY, -- primary key column**

**Name [NVARCHAR](50) NOT NULL,**

**Location [NVARCHAR](50) NOT NULL,**

**Email [NVARCHAR](50) NOT NULL**

**);**

**GO**

**-- Insert rows into table 'Customers'**

**INSERT INTO dbo.Customers**

**([CustomerId],[Name],[Location],[Email])**

**VALUES**

**( 1, N'Orlando', N'Australia', N''),**

**( 2, N'Keith', N'India', N'keith0@adventure-works.com'),**

**( 3, N'Donna', N'Germany', N'donna0@adventure-works.com'),**

**( 4, N'Janet', N'United States', N'janet1@adventure-works.com');**

**GO**

Azure Table storage is an example of a key-value store. Cosmos DB also implements a key-value store using the [**Table API**](https://docs.microsoft.com/en-us/azure/cosmos-db/table-introduction).

Azure Cosmos DB implements a document database approach in its Core (SQL) API.

The most widely used column family database management system is Apache Cassandra. Azure Cosmos DB supports the column-family approach through the Cassandra API.