**Data warehouse concepts**

A data warehouse is a **centralized repository that stores data from various sources in a single location**, making it easier to analyze and report data (OLAP). And it is maintained separately from the organization’s operational database**.**

- Consolidates data from multiple sources (e.g., databases, files, APIs)

- Stores data in a structured and transformed way (e.g., tables, schemas)

- Optimizes data for querying and analysis (e.g., indexing, partitioning)

- Enables data governance and quality control

**What is database schema-** A database schema is the overall structure or organization of a database, including: attributes & records/ **rows and columns**

**It shows how the data is organized and related to each other.**

- What tables are in the database

- What columns are in each table

- How the tables are connected

- What rules apply to the data

**Types of data**

**1. Structured data**: Highly organized and formatted data (it represents tabular format, combination of rows and columns such as:

- Relational databases (e.g., SQL Server, Oracle)

- Tables (e.g., Azure Table Storage)

- Files (e.g., CSV, JSON)

**2. Semi-structured data:** Partially organized data with some level of structure, such as- XML files, JSON files

**3. Unstructured data:** Unorganized and raw data, such as:

- Images, Audio files, Video files, Text files, logs, social media posts

**Difference between OLAP and OLTP?**

OLAP & OLTP are two different types of data processing:

**OLAP (Online Analytical Processing):** OLAP for data warehousing and analytics

**OLTP (Online Transactional Processing):** OLTP for data integration and transactional processing.

**Key differences:**

- Purpose: OLAP is for analysis, while OLTP is for transactions

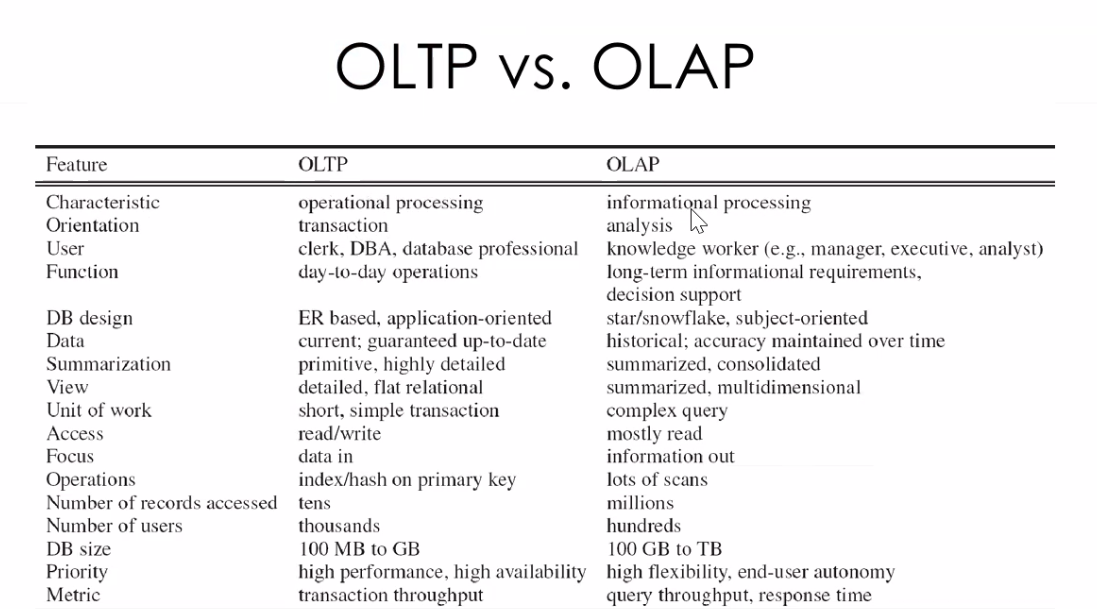
- Data format: OLAP uses multidimensional data, while OLTP uses relational data

- Workload: OLAP is read-heavy, while OLTP is write-heavy

- Performance: OLAP optimizes for query performance, while OLTP optimizes for transactional performance

**OLAP: Eg- Data warehousing, data mining, and data analytics**

**OLTP: Eg- Customer relationship management, order processing, and financial transactions**



**Types of tables and what is it ?**

**Destination Table is where the processed data is stored, and the Fact Table is a type of table that stores specific types of data (measurable data) in a structured format.**

1. **Destination Table:** Destination Table is where your transformed data is stored after processing. It as the "target" or "output" table.

Eg: You have a pipeline that processes sales data and stores the results in a table called "SalesSummary". In this case, "SalesSummary" is the Destination Table.

1. **Fact table:** is a table that stores measurable data, like numbers or quantities. It's like a table that contains the "what" and "how many" details.

Eg: A Fact Table called "Sales" might contain columns like:

+ Date+ Region+ Product+ Quantity Sold+Revenue

**Destination Table:**

- **"Where the data is stored"**

- Is a table that stores the transformed and processed data

- Is the target table where the data is loaded after processing

- Can be a relational database table, a cloud storage location, or a data warehouse table

-Has a **primary key that** is referenced by fact tables

**Fact Table:**

- **"What the data is about" (measurable data)**

- Is a table that stores measurable data, such as numbers or quantities

- Contains the "what" and "how many" details about a specific business process or event

- Typically has a date column and measurable / numerical data types (e.g., quantity, revenue)

- Has **foreign keys** that link to dimension tables

**what is azure cloud**

Azure Cloud or Microsoft Azure, is a cloud computing platform and set of services offered by Microsoft. It allows users to build, deploy, and manage applications and services through Microsoft-managed data centers across the globe.

**Azure Cloud - features and services**

1. Compute Services (Virtual Machines, Functions, etc.)

2. Storage Services (Blobs, Files, Disks, etc.)

3. Database Services (Azure SQL, Cosmos DB, etc.)

4. Security and Identity Services (Azure Active Directory, etc.)

5. Networking Services (Virtual Networks, Load Balancers, etc.)

**Difference between on-Prem and On Cloud**

On-Premises (On-Prem) and On-Cloud are two different deployment models for software, applications, and infrastructure.

**On-Premises (On-Prem):**

- Infrastructure and data are stored on-site, within an organization's own facilities.

- Hardware and software are managed and maintained by the organization.

- Data is stored locally, data center (1000TB)

- High investment

- Examples: Local data centres, servers, and applications installed on-premises.

**On-Cloud:**

- Infrastructure and data are stored remotely, in a cloud provider's data centers.

- Hardware and software are managed and maintained by the cloud provider.

- Data is stored in a remote location, accessible via the internet.

- Less investment, data center (30TB)

- Examples: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and Software-as-a-Service (SaaS) applications.

**Key differences:**

- Location: On-Prem is on-site, while On-Cloud is remote.

- Management: On-Prem is managed by the organization, while On-Cloud is managed by the cloud provider.

- Cost: On-Prem typically requires upfront CapEx, while On-Cloud requires ongoing OpEx.

**Storage account**

**a storage account is a container that stores data in Azure.**

A clod-based storage service that stores and manage data and it provide a centralized repository for storing and accessing data and it stroes data permanently, allowing and retrieval as needed.

**Types of storage:**

1. **Blob**: its kind a place holder or container where all the data can be stored. we cannot create directories/folder within the folders (we can store the data up to 500TB). Flat name spaces/ it is cheaper. It good to keep all kind of data (structure and semi structure data)
2. **ADLsgen2:** it’s a upgraded version of blob storge and ADLS1. we can create directories within directories/folders. (we can store the data above 1000 TB). Hierarchical name spaces.

**ETL= Extraction Transformation and Loading**

1. **Extract**: Retrieve data from various sources, such as databases, files, or applications.

2. **Transform**: Clean, aggregate, and transform the data into a standardized format using ADF's data transformation capabilities, such as Azure Functions, or Stored Procedures.

3. **Load:** Write the transformed data to a target data store, like a data warehouse, or data lake.

In ETL, the transformation step occurs before loading the data into the target system.

**ELT= Extraction Loading and Transformation**

1. **Extract**: Similar to ETL, retrieve data from various sources.

2. **Load:** Load the raw, untransformed data directly into a target data store

3. **Transform**: Transform the data after it's been loaded, using ADF's data transformation.

In ELT, the transformation step occurs after loading the data into the target system.

**Key differences:**

- ETL transforms data before loading, while ELT loads data before transforming.

- ETL typically requires more processing power and storage before loading, whereas ELT stores raw data and transforms it on demand.

- ELT allows for more flexibility and scalability, especially when dealing with large datasets or real-time data.

**What is ADF?**

**--SSIS on prem tool to do ETL operations**

ADF = Azure Data Factory, which is a cloud-based data integration service provided by Microsoft Azure. It allows users to create, schedule, and orchestrate ETL (Extract, Transform, Load) workflows to move and transform data from various sources to different destinations.

ADF is used for following use cases.

* Data migration from one data source to another source
* On Prime to cloud data migration and cloud to cloud
* ETL purpose
* Automated data flow

**Concepts/components of Azure Data Factory**

These components work together to enable data integration, transformation, and orchestration in ADF.

**Pipeline**: A logical grouping of activities that perform a unit of work.

**Activity:** A single task that performs a specific operation, such as data movement or transformation.

**Dataset:** Reference or points to the data where you want to use/perform the activities. It represents data structure within the data stores, which simply point to or refence the data you want to use in your activities.

**Linked Service:** A connection to a data store or compute service.

**Trigger:** A mechanism that initiates the execution of a pipeline.

**Execution:** The process of running a pipeline or activity

**Integration Runtime (IR):** The compute infrastructure that ADF uses to run activities.

**Data Flow:** A visual representation of the data transformation process.

A Data Flow consists of:

Sources: Where data comes from (e.g., databases, files, APIs).

Transformations: Operations applied to data (e.g., filter, sort, aggregate, join).

Sinks: Where transformed data is written (e.g., databases, files, data lakes).

**Source:** The data store or service that provides the input data.

**Sink**: The data store or service that receives the transformed data.

**INTEGRATION RUNTIME (IR**) is a compute infrastructure in Azure Data Factory (ADF) that allows you to integrate data from various sources and destinations. It's a cloud-based engine that enables data integration, data transformation, and data movement.

It will create bridge between two differences sources

Integration Runtime provides:

1. Data Movement: Move data between different sources and destinations, such as databases, files, and cloud services.

2. Data Transformation: Transform and process data using various activities, such as mapping, filtering, and aggregating.

3. Data Integration: Integrate data from multiple sources into a single, unified view.

**1. Azure Integration Runtime**: Cloud-based IR, suitable for most data integration scenarios.

**Public network or cloud to cloud.**

**2. Self-Hosted Integration Runtime:** On-premises IR, used for data integration with **on-premises data sources**. It allows you to lift and shift your existing data integration workflows to the cloud.

**It helps move data from On-Prem/private to Cloud.**

**3. Azure-SSIS Integration Runtime:** A specialized IR for SQL Server Integration Services (SSIS) packages. It enables you to run SSIS packages in ADF.

**This is kind of ETL tool**

**what is linked services? And what are the components of linked service?**

Connection string for the storage.

**a Linked Service is a connection to a data store or a compute resource**

The components of a Linked Service in Azure Data Factory (ADF) are:

**1. Connection Information**: Specifies the details for connecting to the data store or compute resource, such as:

- Server name

- Database name

- Username

- Password

- File path

**2. Authentication Type:** Determines the authentication method to use, such as:

**3. Credential:** Stores sensitive information, like:

- Passwords

- Access keys

- Secrets

**4. Encryption**

**5. Type:** Specifies the type of linked service, such as:

- Azure Blob Storage

- Azure SQL Database

- Oracle Database

- Hadoop Distributed File System (HDFS)

**6. Description**: Provides a description of the linked service.