Pre-requisites

13 February 2023 10:04

CPU i5/i6,
RAM - at least 12 GB,
Disk Storage - 500GB

Windows 10, 11 Professional / Enterprise
SQL Server 2017 / 2019 Developer / Enterprise
MS Office
PowerBI
Alteryx

username	email	Password	invitation_link
db01@mml.local	db01@mml.local	uavdxoa5yg	https://laas.makemylabs.in/wsm/KPMG-Databrick-skip20230213170942?invitation_id=35acb6a3-341b-4330-b87a-8a810472703c&context=True

Database Fundamentals and SQL Server

13 February 2023 10:07

Database types & Models

1. **Flat file database** - kind of text database where each line of the plain text file holds only a single record (e.g. MS access)
2. **Hierarchical database** - based on hierarchical data model, it's viewed as a collection of tables, data is designed into a tree like structure where each record consists of one parent record and many child record. (e.g. IBM DB2 - IBM information Management system (IMS), Windows Registry, XML data storage.
3. **Network Model database** - it can consists of parent segments and this segment can be grouped together as levels but there always exists a logical association between the segments belonging to any level.
4. **Relational database** - consists of tables and columns, rows.
5. **Object-oriented database** - information can be represented in the form of object - oriented programming inclined towards the objects like e.g. multimedia records in a relational database can be definable data object.
6. **Distributed database** - consists of two or more files located in different sites / location.
7. **NoSQL database** - non-relational db which has support for unstructured, semi-structured data as well as it can include dynamic schema, flexible data model for faster data retrieval
   e.g. Mongo db, Cassandra, Azure Cosmos db, Couch db etc.

8. **Graph database** - node-entity (rows in the table), attributes (relationship/columns) . e.g. Neo4j, Azure Cosmos db Graph API etc.

ACID properties in RDBMS

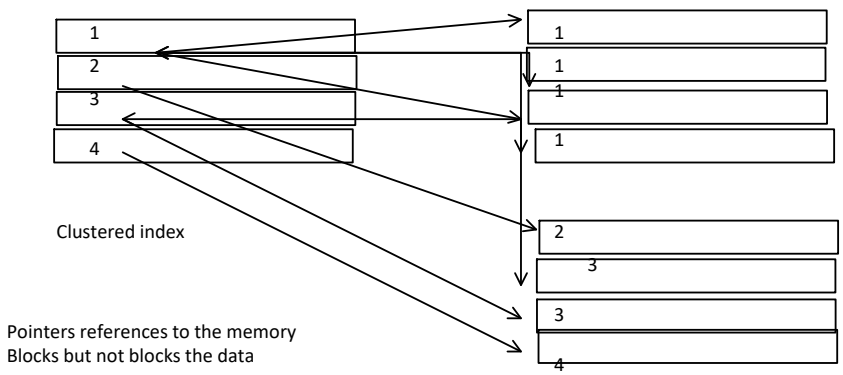
A = Atomicity	The entire transaction should take place at once or doesn't happen at all
C = Consistency	The database must be consistent before and after the transaction
I = Isolation	Multiple transactions can occur independently without interference
D = Durability	The changes of a successful transaction occurs even if a system failure occurs.

Clustered & Non-clustered Index

1. In a table, there can be one clustered index, can be multiple non-clustered index.
2. Clustered index is much more faster, non-clustered index is slower.
3. Clustered index requires less memory for operations, non-clustered indexes requires more memory for operations.
4. In clustered index, index is the main data. In case of non-clustered index, index is the copy of the data.
5. Clustered index store pointers to blocks, not the data. Non-clustered index store both the value and a pointer to actual row which holds the data.
6. Primary Keys of the table by default is considered as a clustered index. Composite key used with unique keys of the table defines the non-clustered index.
7. A clustered index is a type of index in which table records are physically recorded to match the index. A non-clustered index is a special type of index in which the logical order of the index doesn't match physical stored order of the rows on disk.
8. Clustered index size is larger, non-clustered index size is smaller.

Features of Indexes

1. A index can speed up the data retrieval and query execution very quickly by optimization
2. Indexes can be created or dropped with no effect on the data
3. When an index is created, it includes a column containing a wide range of values.



Primary Key	Unique Key
A table can have only one primary key	A table can have more than one unique key unlike the primary key
A primary key cant accept null values	Unique key constraint can accept null values for a column
	Unique key constraints are also referenced by the foreign key of another table, it can be used when developer wants to enforce a unique constraints on a column or group of columns which is not a primary key
Primary key has the support of auto-increment values.	A unique key does not support auto-increment value
We cant change or delete values stored in primary key	We can change the unique key values

Surrogate Keys

Surrogate keys are called synthetic primary keys which are generated when a new record is inserted into the table automatically by the database which can be declared as the primary key of the table.

Features of surrogate key

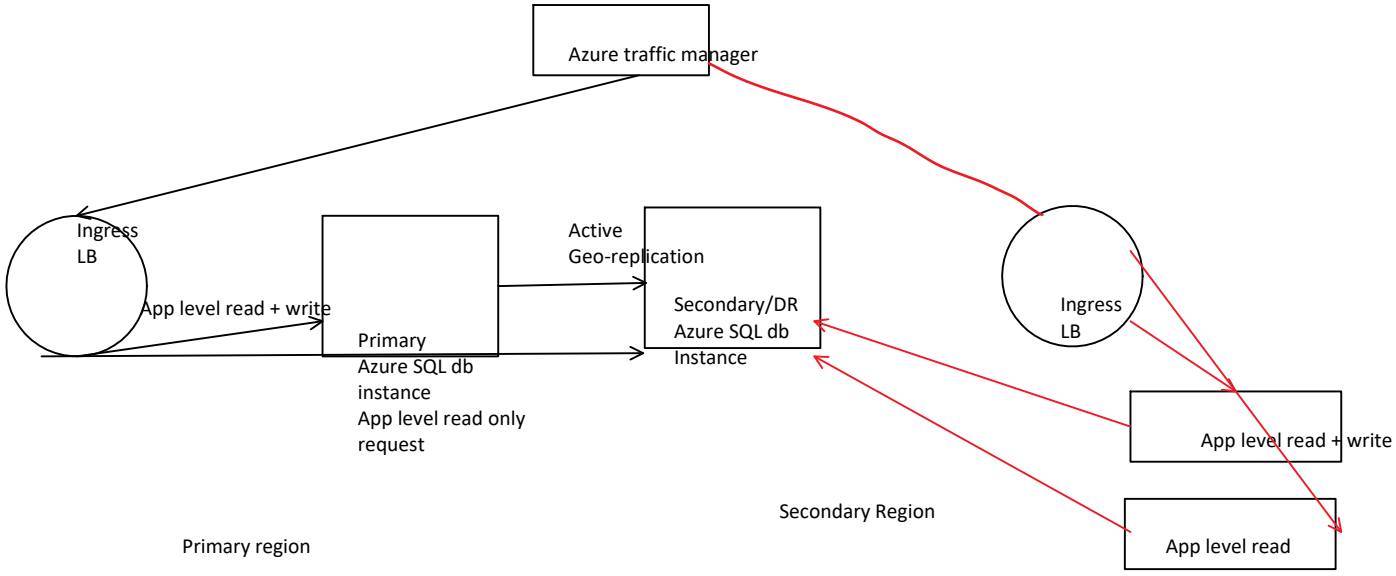
1. It's a sequential number outside the database which's made available to the user and application or it just acts as an object which's present in the database but not visible to the user/application.
2. It's automatically generated by the system
3. It holds an anonymous integer
4. It contains unique values for all records of the table
5. The value can never be modified by the user or application
6. Surrogate keys are called the factless key which is added just for the case of identifying unique values and contains no relevant fact which is useful for the table.

Surr_no	Reg_no	Name	Marks
1	21101	Mark	50
2	32281	Henry	70
3	43353	Alan	60
4	CS101	Maria	80
5	CS201	John	67

Azure SQL

14 February 2023 09:31

SQL Server on Azure VM (IaaS)	Azure SQL Managed Instance (PaaS)	Azure SQL database (PaaS)
Azure SQL VM is preferred for the scenario of simple lift-shift or rehost of existing full SQL server database engine with core administration services like e.g. service broker, SQL Server Agent, SQL mirrors, DTC, .NET and CLR data types & functions, SQL server BI support.	Lift - shift of database migration including core SQL db features like service broker, SQL server agent, distributed transactions and .net/CLR data types etc. without any management overhead.	Purely managed SQL database offering with no licensing requirement, no underlying administration of server, SQL database or network is required. End to end database migration is feasible with a just few clicks into the core sql development features.
License is required. Either through BYOL with AHB or pay-as-you-go model, license has to be procured.	No license is required, entire SQL on-premise feature can be availed with the managed platform service.	No license is required. Only SQL db dev specific features are available. There's a limitation of db sizes (~100TB)
SQL Server BI (SSRS, SSAS, SSIS), SQL Server broker, .net framework runtime, CLR integration, distributed transactions, database mail etc. all features are supported	.net framework, functions, distributed transactions, ACID , automated backups, HA are also supported along with no administration required.	No support for .net framework runtime, distributed transactions, CLR related functions, stored procedures, windows runtime etc. are not supported
SLA - 99.95% Automated backups and business continuity through FCI and availability groups. Migration can be implemented through Azure migrate (Azure site recovery).	SLA - 99.99% automated backups, point-in time restore, geo-replication, high availability, automated patching etc. features are available.	SLA - 99.99% , automated backups, active geo-replication, high availability, and disaster recovery, replication on both availability zone and regional level.



Migration to Azure SQL database

We can migrate SQL Server database running on-premises or to :

a) SQL Server on Azure VM
b) SQL Server Managed instance
c) Azure SQL db

We can migrate the following db types -

1. SQL Server running on-prem or SQL server running on VM (vmware/hyper-v etc.)
2. SQL server running on AWS EC2, Google compute engine
3. SQL server on AWS RDS
4. Cloud SQL for SQL server - GCP

1. Azure Migrate - Discovery and assess single database or at scale from different env
2. Azure SQL migration extension for Azure Data Studio - migrate to Single databases at scale, it can run in both online and offline modes.

3. Import-export service / BACPAC - migrate individual LOB apps databases , suited for smaller databases and doesn't require a separate migration service or tool.
4. Bulk Copy - migrate / transform data from source SQL server db to Azure SQL db. There's a downtime for exporting data at source and importing at the target.
5. Azure Data Factory (ADF) - source SQL server db to target Azure SQL db. Cost is important consideration and is based on factors like pipeline triggers, activity runs and duration of data movements.
6. SQL data sync - synchronize data between source and target databases. Suitable to run continuous sync between Azure SQL db and on-premise / Azure SQL db. It can have higher performance impact depending on the workload.

Migration Steps

1. Discover -
2. Assessment
3. Migrate
4. Cutover
5. Optimize

A) Data Migration Assistant (DMA)

Scenario to choose for Azure SQL Managed Instance

Pools are well suited for a large number of databases with specific utilization patterns, for a particular database, the pattern is characterized by low average utilization with infrequent utilization spikes.

Conversely, multiple databases with persistent medium-high utilization shouldn't be placed in the same elastic pool.

The more databases, we can add into a pool, the greater the savings become. Depending on the app utilization pattern, it's possible, to see savings as few as two AWS S3 db.

Overutilization and underutilization of DTU usage can be overcome through Azure SQL elastic db.
A elastic pool allows these unused DTUs to be shared across multiple databases. A pool reduces the DTUs needed and the overall cost.

The best size for a pool depends on the aggregate resources required for all databases in the pool

1. Maximum compute resources utilized by all databases in the pool. Compute resources are indexed by either eDTUs or vCores depending on the purchasing model.
2. Maximum storage bytes utilized by all databases in the pool.

Business Continuity for Azure SQL Elastic db

1. Point-in time restore - point-in-time restore uses automatic database backups to recover a database in a pool to a specific point in time.
2. Geo-restore - Geo-restore provides the default recovery option when a database is unavailable because of a incident in the region where the db is hosted.
3. Active geo-replication - For applications, which has more aggressive recovery requirements, than geo-restore can offer, we can configure active geo-replication or auto-failover group.

Azure SQL Managed Instance

vCore based purchasing model

A vCore represents a logical CPU and provides the option to choose the physical characteristics of the hardware (the no of cores, the memory, the storage sizes). The vCore based purchasing model gives us the flexibility, control, transparency of individual resource consumption and a straight forward way to translate on-prem workload requirements to the Azure platform.

vCore based purchasing model depends on -

- Service tier
- Hardware configuration
- Compute resources (the no of vcores and amount of memory)
- Reserved database storage
- Actual backup storage

Benefits of vCore based purchasing model used by Azure SQL managed instance

- Control over hardware configuration to better match the compute and memory requirements of the workload
- Pricing discounts for AHB and reserved instance
- Greater transparency in the hardware details which empowers compute, helping facilitate planning for migrations from on-prem deployments
- Higher scaling granularity with multiple compute sizes available.

Backup Storage

- Point in time restore - The storage consumption depends on the rate of the change of database and retention period configured for backups. We can configure a separate retention period of each database between 0 to 35 days for SQL managed instance. A backup storage amount is equal to the configured max data size is provided at no extra charge.

- Long term retention (LTR) - customers have the option to configure the long term retention of full backups for upto 10 years,

Feature	General Purpose	Business Critical
Scenario	Most standard business workloads. Offers budget-oriented, balanced and scalable compute and storage options	Offers business applications with highest resilience to failures by using several isolated replicas, and provides the highest I/O performance.
Read-only Replicas	0	1
HA replica	One replica is available	Three HA replica is available & one read-scale replica
Read-only replicas with failover groups enabled	One additional read-only replica and two total readable replicas, which includes the primary replica	Two additional read-only replicas, three total read-only replicas, four total readable replicas which includes the primary replica.

SQL Server on Azure VM (HADR configurations)

A Windows Server Failover Cluster is used for high availability and disaster recovery (HADR) with SQL Server on Azure VM.

Best Practices

- Deploy the SQL Server VM to multiple subnets to avoid the dependency on the Azure LB or a distributed network to route traffic to HADR solution.
- Change the cluster to less aggressive parameters to avoid unexpected outages from transient network failure to Azure platform maintenance.
- Place the SQL Server VM in a AS or different Azs.
- Use a single NIC per cluster node
- Configure the cluster quorum voting to use 3 or more odd numbers of votes.

a) Cloud witness - it's ideal for deployments in multiple sites, multiple zones and multiple regions. Use a cloud witness for disk quorum whenever possible unless using a shared-storage cluster solution.

b) Disk witness - it's the most resilient quorum option and is preferred for any cluster which uses Azure shared disks (like shared SCSI, iSCSI of fiber SAN). A clustered shared volume can be used for disk witness.
c) Fileshare witness - is suitable when the disk witness and cloud witness are unavailable

SQL database auditing

- Retain - an audit trail of selected events. We can define categories of database actions to be audited.
- Report - on database activities. We can use pre-configured reports and a dashboard to get started quickly with activity and event reporting.
- Analyse - reports, we can find suspicious events, unusual activity and trends.

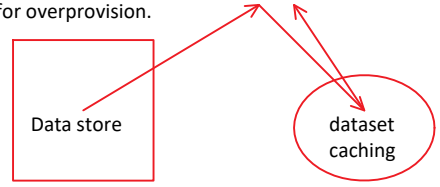
Server level vs database level audit policy

- An audit policy can be defined for a specific database or as a default server policy in Azure SQL db.

- A server policy applies to all existing and newly created databases on the server.
- If server auditing is enabled, it always applies to the database. The database will be audited, regardless of the database auditing settings.
- When auditing policy is defined at the database level to a log analytics workspace or an event hub subscription destination, these operations will not keep the source database level auditing policy like
- Database copy
- Point -in time restore
- Geo - replication

Performance improvement measures for Azure SQL db

1. Caching can improve app performance - database queries can be stored over the cache and return the queries much faster than the db when the app requests them. Not only this approach, helps for significant reduction for latency but also reduces the load on the database, lowering the need for overprovision.



- Determine whether the item is held in cache
- If the item is not available, currently in cache, read them from the data store
- Store a copy of the item in the cache

2. Caching are better than db at handling high throughput of requests - enabling the app to handle more simultaneous users.
3. Caches are typically most popular beneficial for read-heavy workloads where the same data is being accessed again and again. Caching pricing, inventory, session state or financial data are some of the examples.

4. Resolve the server index fragmentation with automatic tuning - keep the fast query performance is paramount for app with rdbms, one of the common cause for degraded performance is index fragmentation. With the indices, The SQL server can quickly locate the row with the data which user is requesting for, the first step is to identify the degree of fragmentation.

5. Resolve the fragmentation - there're three primary options for automatic tuning with Azure SQL db

a) CREATE INDEX - create new indices which can improve the performance
b) DROP INDEX - Drops redundant and unused indices (>90 days)
c) FORCE LAST GOOD PLAN - identifies queries using the last known good execution plan

MAXDOP configuration

a) In Azure SQL database (both for single and elastic pool db), the default MAXDOP setting for each new single database and elastic pool database is 8.

b) For Azure SQL managed instance, the max degree of parallelism instance option will be set to 8 by default.

This default prevents unnecessary resource utilization, while still allowing the database engine to execute queries faster using parallel threads.

- Azure SQL db level, MAXDOP can be controlled at the db level using MAXDOP database-scoped configuration.
- For Azure SQL managed instance, customers can also set the server 'max degree of parallelism' configuration option & can control MAXDOP at the resource governor workload group level.
- For all of Azure SQL deployment options, MAXDOP can additionally be controlled at the individual query level by using OPTION (MAXDOP) query hint where it actually overrides MAXDOP configurations set in the database or instance scope.

Azure Data Factory

14 February 2023 09:31

ETL is the process for integrating and loading of the data for computation and analysis. It's also the primary method to process data for traditional data warehousing and BI applications.

Benefits

1. Extract the data from the legacy systems
2. Cleanse the data to improve data quality and establish consistency
3. Load the data into the target database.

Azure Data Factory is a managed data orchestration and integration platform which helps to build complex Extract, Transform and Load (ETL) or ELT projects with data integration features.

Data Orchestration is the practice of acquiring, cleaning and matching, enriching and making data accessible across the technology teams. The effective data orchestration captures data from several sources and unifies it within a centralized system, making it organized and ready to use for getting insights from data.

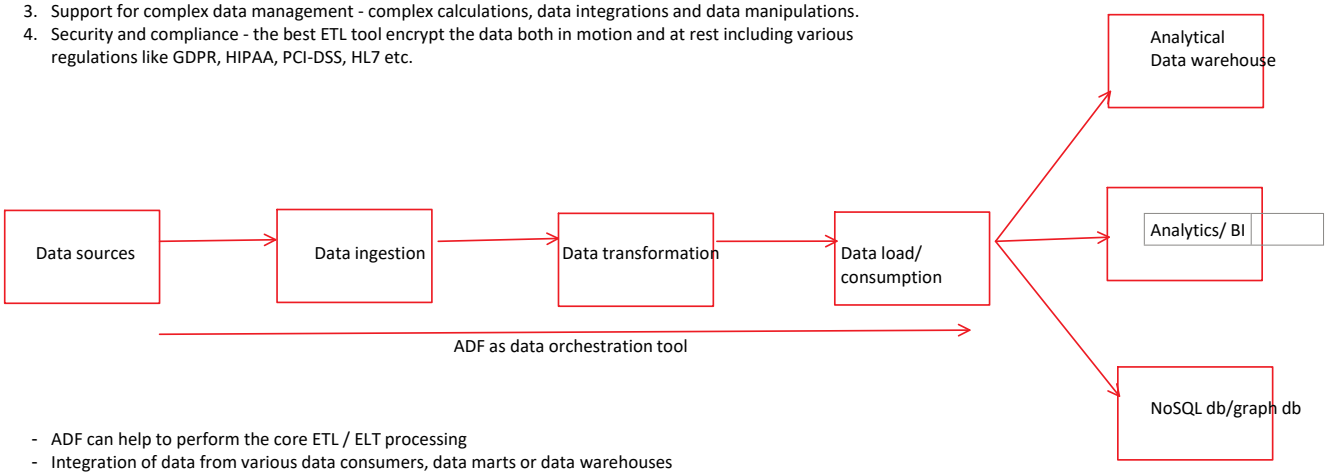
In ETL perspective, orchestration means automated management, co-ordination and management of complex data pipelines. ETL tools run on a schedule.

ETL (Extract, Transform & Load)	ELT (Extract, Load and Transform)
ETL can perform the end to end data transformation after loading the raw data into Transformation layer and captures the insights from the data.	ELT copies or exports the data from the data source locations, but instead of loading it to a staging area for transformation, it loads the raw data to the target data store to be transformed as required.
The ETL process involves more structured datasets and data has to be rational in nature, should have proper keys (PK, FK) to integrate the relationships between multiple tables	ELT is useful for high-volume, unstructured datasets as loading can occur directly from the data source. ELT is more ideal for Big Data Analytics pipeline because it can clean, parse the unstructured, semi-structured data and can load the data directly into the native format. It's ideal for big data use cases.
In on-premise, ETL data pipeline is more common	On Cloud, Azure ELT pipeline is more popular

Azure Data Factory (ETL/ ELT tool)

1. Comprehensive automation support - leading ETL tool (ADF) can automate the entire data flow, from data sources to the target data warehouse. Many tools recommend rules for extracting, transforming and loading of the data.

2. Visual drag-drop support - the functionality can be used for specifying rules and data flows.
3. Support for complex data management - complex calculations, data integrations and data manipulations.
4. Security and compliance - the best ETL tool encrypt the data both in motion and at rest including various regulations like GDPR, HIPAA, PCI-DSS, HL7 etc.



- ADF can help to perform the core ETL / ELT processing
- Integration of data from various data consumers, data marts or data warehouses
- Allows us to create the data driven workflows through ingestion, transformation and consumption
- Code free ETL tool, it helps to integration of data through ingestion, transformation and consumption through the data flows, control flows and scheduling of the ADF pipeline.

ETL process

1. Extract

Raw data gets copied into / exported from source location to a staging area. Data management can extract the data from the variety of data sources that can be structured or unstructured.

- SQL / noSQL
- CRM/ERP/MDM
- Flat files

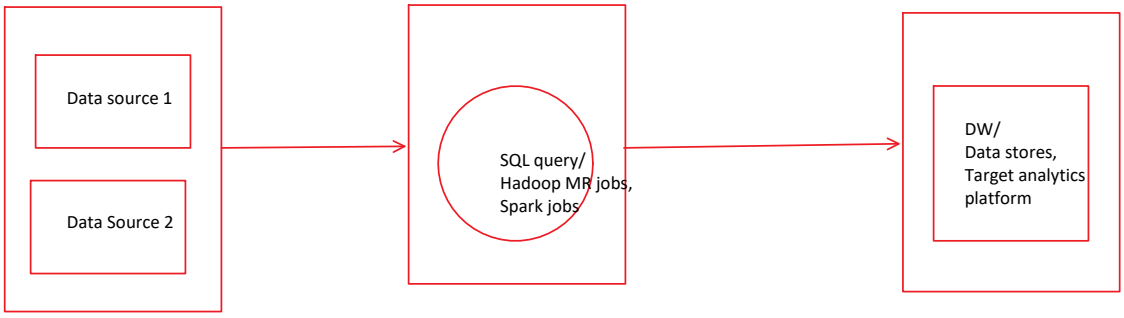
- Web pages

2. Transformation

In the staging area, the raw data undergoes the data processing, data has to be transformed and consolidated for its intended analytical use cases -

- Filtering, cleansing and de-duplicating, validating and authenticating the data
- Performing calculations, translations, summarizations of raw data
- Conducting audits to ensure data quality and compliance
- Removing, encrypting or protecting data governed by regulators
- Formatting the data into tables, joined to match the schema of the target data warehouse.

3. Load
   In the last phase, the transformed data is moved from the staging area to a target data warehouse. Which involves loading of the data , followed by the periodic loading of incremental data changes and full refreshes to erase and replace the data in the data warehouse.



- The data transformation can take place usually involves various operations, such as filtering, sorting, aggregating, joining of the data, cleaning, deduplicating and validating the data.

- As part of offering, these three ETL processes can execute in parallel to save time. e.g. while the data is being extracted, and a loading processing can begin working on the prepared data. Rather than just waiting for the entire extraction process to complete.

Core Components of Azure Data Factory

1. **Linked Service** - Creates a linking connection between the data source and the ADF pipeline.

   We must create the linked service to link up the data source with the data factory.

   e.g. db connection string which defines the connection information required for the service to connect to the external resource.

2. **Dataset** - It's the named view of the data which simply points to or references the data which required to be used in the ADF activities as inputs and outputs.

e.g. SQL server db, files

3. **Activities** - The activities refers to the actions, jobs / tasks can be performed on the data. Activities also can produce a dataset & the datasets consume the activities.

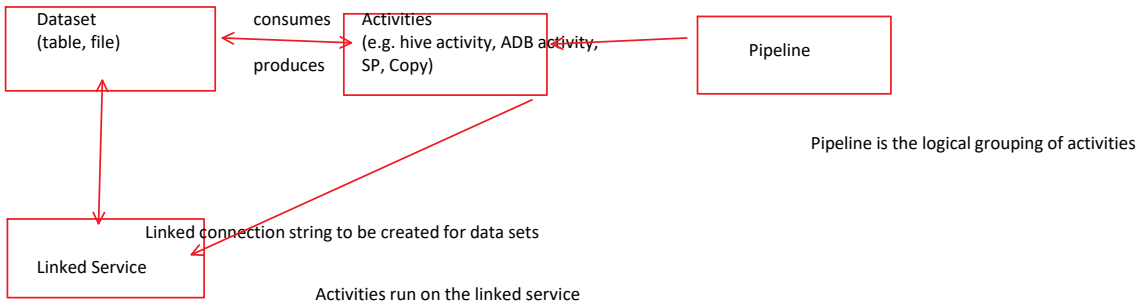
while copying/moving the data from Azure Blob storage to Azure SQL db, the storage and the linked service connection string need to be created.

- Activity can help us to copy the data
- Deduplicating the data
- Formatting the data removing all nulls, NANs
- Applying normalization, remove redundancies
- Adding column headers and formatting

4. **Pipeline** - It's logical grouping of activities, which can be monitored, managed and scheduled. The activities in a pipeline define actions to perform on the data.

e.g. an ADF pipeline could contain the set of activities which can ingest and clean log data, then transform the mapping data flow to analyse the log data.

- The pipeline allows to manage the activities as a set instead of each one individually. We can deploy and schedule the pipeline instead of activities independently.
- The activities in a pipeline defines the action to perform on the data.



Hands-on Lab 01

1. Create Azure Data factory
2. Explore the Data Factory Studio
3. Explore the ADF activities, pipelines and datasets, linked services
4. Monitoring and management of ADF

Hands-on Lab 02

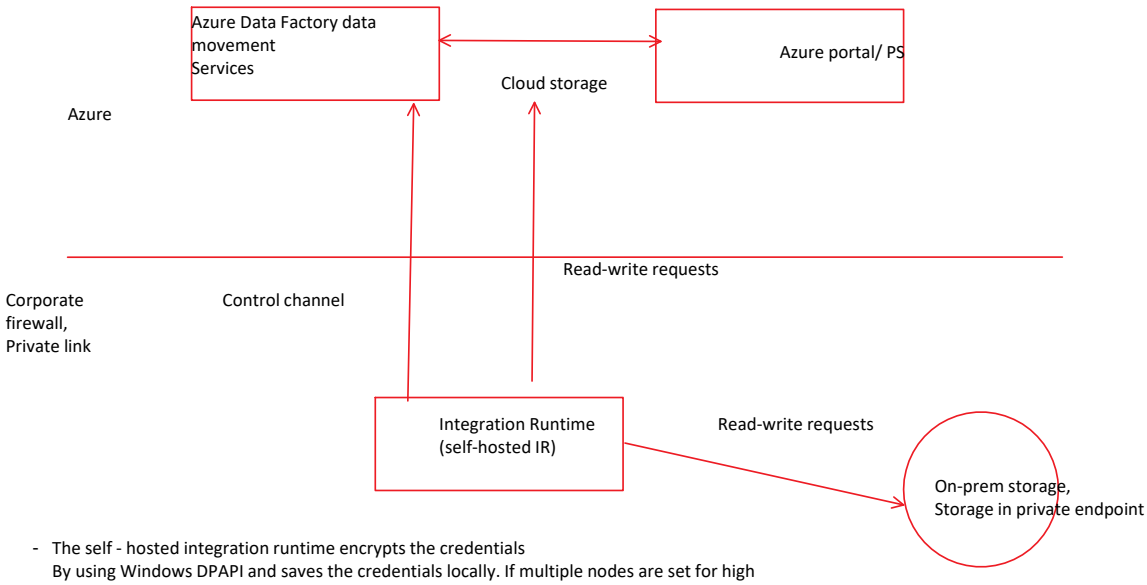
1. Transferring data from Azure SQL db to Azure Blob storage using Copy activity and Copy data tool.

a) Provision Azure SQL db with sample db
b) Provision Azure blob storage
c) Use the copy data tool to copy data from Azure SQL db to Azure blob storage/ADLS gen2
d) Create the ADF pipeline
e) Monitor the pipeline

The Integration Runtime (IR) is the compute infrastructure used by ADF to provide the data integration capabilities across the different environments.

- Data Flow - Execute a data flow in a managed Azure compute environment.
- Data movement - Copy data across the data stores in a public or private networks (for both on-prem or Vnets). This service provides support for built-in connectors, format conversion, column mapping and performant / scalable data transfer.
- Activity dispatch - Dispatch and monitor transformation activities running on a variety of compute services like ADB, Azure HDI, ML Studio, Azure SQL db and SQL server.
- SSIS package extension - natively execute SSIS packages in a managed Azure compute environment.
- So, the integration runtime provides the bridge between the activities and linked services. It specifies the compute environment where the activity is to be performed in the closest region to the target data store or compute service to maximize the performance while allowing flexibility to meet security and compliance requirements.
-
-

a) Azure - Data Flow, Data movement, Activity Dispatch (public and private endpoints)
b) Self-hosted - Data movement, Activity dispatch (public and private endpoints)
c) Azure-SSIS - SSIS package extension (public and private endpoint)



- The self - hosted integration runtime encrypts the credentials
  By using Windows DPAPI and saves the credentials locally. If multiple nodes are set for high

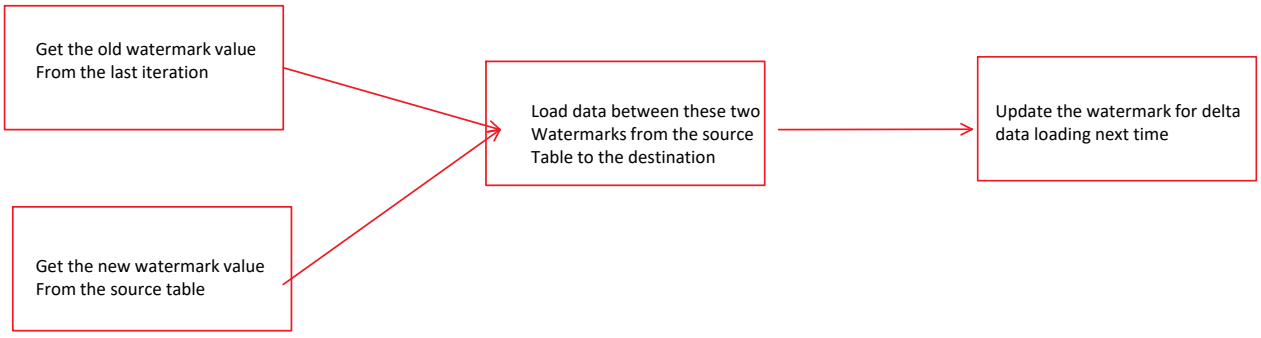
availability, the credentials are further synchronized across other nodes.
- The self hosted integration runtime can directly communicate over cloud based storage service like Azure blob storage over a secure HTTPS channel.

Hands-on Lab 03

1. Create SQL Server db (on-premise)
2. Create a Blob storage
3. Provision over self - hosted integration runtime on ADF
4. Create the pipeline

Incrementally (or delta) loading of data after an initial full data load is a widely used context.

1. **Delta data loading from database/data store by using watermark**

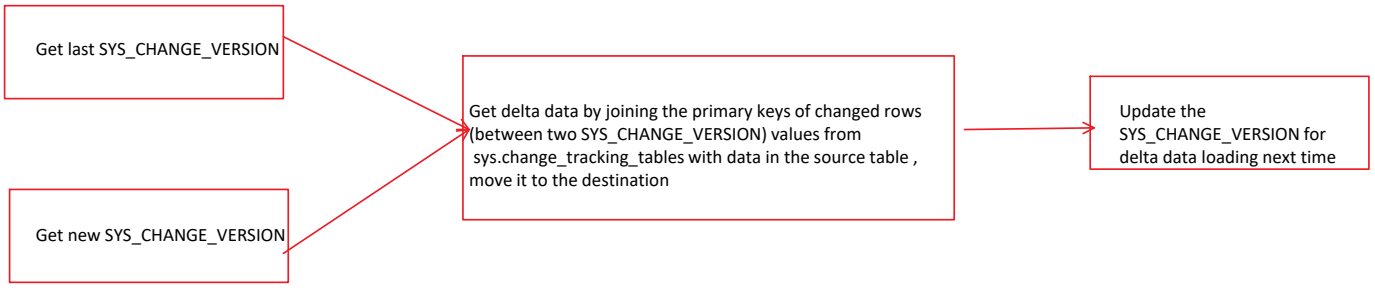


We can define a watermark in the source database. A watermark is a column which has the last updated timestamp or an incrementing key. The delta loading solution loads the changed data between the old watermark and a new watermark.

- Create two lookup activities - first lookup activity is to retrieve the last watermark value. Use the second Lookup activity to retrieve the new watermark value. These watermark values are passed to the Copy activity.
- Create a Copy activity which copies the rows from the source data store with the value of the watermark column > old watermark value and < the new watermark value. It also copies the the delta data from the source data store to Blob storage as a new file.
- Create a StoredProcedure activity which updates the watermark value for the pipeline which runs the next time.

2. **Delta table data loading from SQL db using The Change Tracking Technology**

Change tracking technology is the lightweight solution in SQL/ Azure SQL db which provides efficient change tracking mechanism for apps. It enables an application to easily identify the data which was inserted, updated or deleted.



1. Initial loading of historical data (run once)
2. Incremental loading of delta data on a schedule
   - Get old and new SYS_CHANGE_VERSION values
   - Load the delta data by joining the primary keys of the changed rows (between two SYS_CHANGE_VERSION values) from sys.change_tracking_tables with data in the source table

then move the delta table to destination.
- Update the sys_change_version for the delta loading next time.

Incremental Load

- Create two lookup activities to get old and new SYS_CHANGE_VERSION from Azure SQL db/ SQL db to pass it to copy activity.
- Create one copy activity to copy the inserted / updated/ deleted data between the two SYS_CHANGE_VERSION values from Azure SQL db to Blob storage
- Create one Stored Procedure activity to update the value of SYS_CHANGE_VERSION for next pipeline run.

3. **Loading of new & changed files by using LastModifiedDate & by using time partitioned folder or file name**

   ADF will scan all of the files from the data source store, can apply the filters over by their last modified data and to copy only the new and updated file since the last time to the destination store.

4. **Loading of new & changed files by using time partitioned folder or file name**

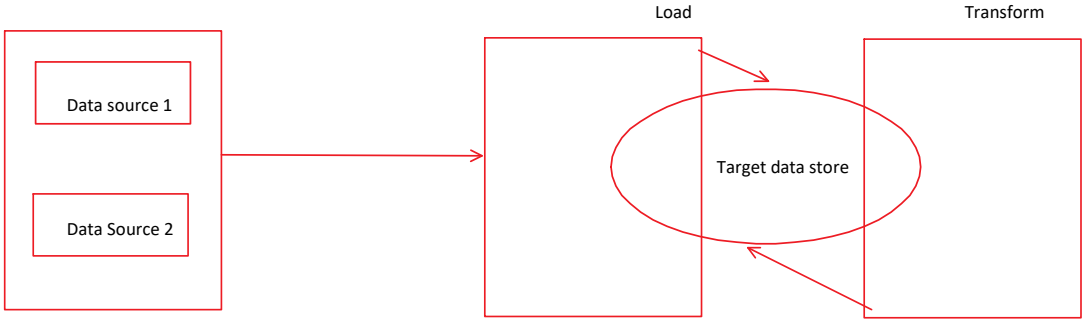
Copy of new files/ folders can be possible through ADF, when they're partitioned with timeslice information as part of file o r folder name

ELT in ADF Pipeline (Extract, Load and Transform)

Extract, Load and transform differs from the ETL mainly in where the transformation takes place. In the ELT pipeline, the transformation can occur in the target data store instead of using a separate transformation engine.

In the ELT pipeline, the transformation takes place in the target data store, the processing capabilities of the target data store are used to transform the data. This simplifies the ELT architecture by removing the transformation engine from the pipeline.

Extract



1. In Azure, the source flat files in scalable storage, such as HDFS, Azure Blob storage and ADLS gen2 are all can be used as data storage layer for extraction.
2. For in-memory based data processing platform like Spark, data warehouse tool hive, polybase, SQL query, pig/latin scripts , MR jobs etc can be used to query the source data.
3. In the ELT pipeline, the key difference is that the data store used to perform the transformation & is the same data store where the data is ultimately getting consumed. The data store reads directly from the scalable storage , instead of loading the data into its own proprietary storage. This approach skips the data copy present in ETL which often just can be time consuming operation for large data sets.
4. The data store only manages the schema of the data and applies the schema on read.
5. The final phase of ELT pipeline, is to transform the source data into a final format which is more efficient for the types of queries which need to be supported.

ADF (is a code - free ETL/ ELT tool)

1. **Ingest**

- **The data can ingested on multi-cloud and on-premise based hybrid storage using Copy data**
- **There're 100+ native connectors**
- **Serverless autoscale**
- **Use wizard for quick copy of data**

2. **Control Flow**
- **Design code free data pipeline**
- **Generate pipelines via SDK**
- **Utilize the workflow constructs - loops, branches, conditional execution, variables and**

parameters.

3. **Data Flow**
   - Code free data transformations which execute on Spark cluster
   - Scaled out (add more cluster instances) in Azure integration runtime
   - Generate the data flows via SDK
   - Designers UI is provided for data engineers and analysts

4. **Schedule**
   - **Build and maintain operational schedules for all of the data pipelines**
   - **We can execute the ADF pipeline manually, programmatically, tumbling window and schedule basis.**

5. **Monitor**
   - **View active execution and pipeline history**
   - **Detail activity and data flow execution status**
   - **Establish the alerts and notifications**

Mapping data flows are visually designed data transformations in ADF. Data flows allow data engineers to develop data transformation logic without writing code. The resulting data flows are executed as activities within the ADF pipeline which uses scaled-out Apache spark clusters. Data Flow activities can be operationalized using existing Azure data factory scheduling, control flow and monitoring capabilities.

ADF Data flow data types

- Array
- Binary
- Boolean
- Complex
- Decimal
- Date
- Float
- Integer
- Long
- Map
- Short
- String
- Timestamp

In ADF, the mapping data flow are operationalized within the ADF pipelines using the data flow activity. All a user has to do is specify which integration runtime to use and pass in parameter values.

Schema Drift in ADF data flow activities

Schema drift is the case where the data sources often change metadata, fields, columns and types which can be added, removed, or changed on the fly.

Without handling for schema drift, the data flow becomes vulnerable to the upstream data source changes. Typical ETL patterns fail when the incoming columns and fields change because they tend to be tied with the source names.

To protect against the schema drift

- Define the sources which has mutable field names, data types , values and sizes.
- Define the transformation parameters which can work with data patterns instead of hard-coded fields and values.
- Define expressions which can understand patterns to match the incoming fields, instead of using the named fields.
- Schema drift can be applied for both data source and sinks.

Transformation of the drifted columns

When the data flow in ADF has drifted columns, we can access the transformations with the methods like

- Use the 'byPosition' and 'byName' expressions to explicitly reference a column by name or position number.
- Add a column pattern in a derived column or aggregate transformation to match any combination of name, stream, position and origin or type.
- Add rule-based mapping in a SELECT or Sink transformation to match drifted columns or column aliases via the pattern.

Control Flow Activity

In ADF, Control Flow activity involves the orchestration of pipeline activities including the chaining of activities in a sequence , branching, defining the parameters at the pipeline level.

- The Control Flow activity also involves passing arguments while invoking the pipeline.
- It also includes custom-state passing and looping containers.
- The Control Flow activity defines the how control / sequence activities can pass through from one task to another task.

Features

1. Control Flow activity is an orchestration of pipeline activities in ADF.
2. The orchestration includes the chaining of activities in a sequence, branching and defining the parameters at the pipeline level.
3. It also helps to pass the arguments while invoking the pipeline on demand or from a trigger.

Example - Lookup activity in ADF.

1. Lookup activity can return upto 5000 rows, if the result set contains more records so then the first 5000 rows will be returned.
2. The lookup activity output supports up to 4 mb in size. Activity will fail if the size exceeds the limit.
3. The longest duration for lookup activity before timeout is 24 hours.

Foreach Activity

It's a control flow activity in ADF. The foreach activity can define the repeating control flow in ADF pipeline. This activity can be used to iterate over a collection and executes specified pipeline activities in a loop.

Control Flow Activities in ADF

Control Flow Activity Name	Purpose / Definition
Append Variable	Append variable activity could be used to add a value to an existing array variable defined on the ADF pipeline.
Set Variable	Set variable activity can be used to set the value of an existing variable of type string, boolean or array defined on a ADF pipeline
Execute variable	The execute pipeline activity allows ADF pipeline to invoke another pipeline
If condition	If condition activity allows directing pipeline execution, based on evaluation of certain expressions. The If condition activity provides the same functionality that an If statement provides in programming. It executes a set of activities when the condition evaluates to true and another set of activities when the condition evaluates to false.
Get Metadata	Get Metadata activity can be used to retrieve the metadata of any data in ADF
The Foreach activity	This activity defines the repeating control flow in ADF pipeline. This activity is used to iterate over a collection and executes the specified activities in a loop. The loop implementation of this activity is similar to Foreach looping structure in programming.
Lookup	Lookup activity can retrieve a dataset from any ADF supported data sources.
Filter	Filter activity can be used in a pipeline to apply a filter expression to an input array
Until	Executes the set of activities in a loop until the conditions associated with activity set to True.
Wait	Wait activity allows pausing pipeline execution for the specified time period.
Web Activity	Web Activity can be used to call custom REST endpoint from an ADF pipeline
Azure Function	Allows to run Azure Function in the ADF pipeline
Validation activity	We can use the validation activity in a pipeline to ensure the pipeline only continues execution once it has validated the attached dataset reference exists, that it meets the specified criteria or timeout has reached.
Webhook activity	It can control the execution of the pipelines through custom code, with the webhook activity, code can call an endpoint and pass it to a callback URL. The pipeline run waits for the callback invocation before it proceeds to the next activity.
Switch Activity	This switch activity provides the same functionality which is a switch statement in programming. It evaluates a set of activities corresponding to a case which matches the condition evaluation.

Webhooks are automated messages sent from the apps when an event occurs. Webhooks have a message, or payload or sent to the unique URI.

There're two kinds of Custom activities are available in ADF.

- Data movement activities to move data between the supported data source and sink data source.
- Data transformation activities to transform data using compute services (Azure HDInsight, Databricks)
- To move the data to/from the data store which the service doesn’t support or to transform /

process data in a way which's not supported by the service. We can create a custom activity with the data movement or transformation logic and use the activity in the pipeline.

Azure Data Lake

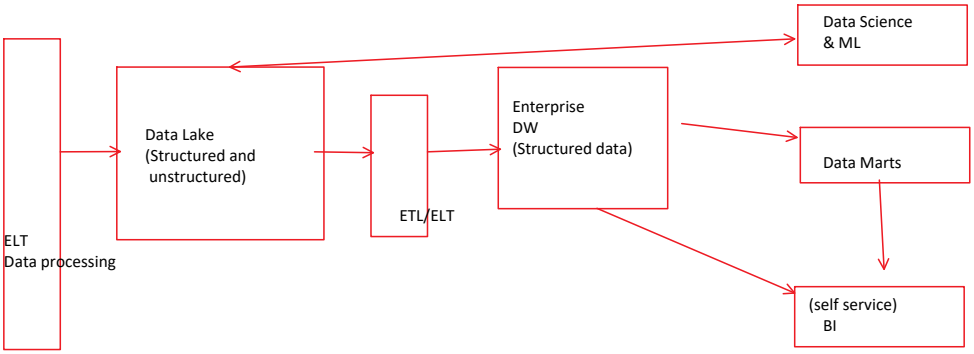
14 February 2023 09:31

A data lake provides the ability to store the data irrespective of volume, variety, veracity of the data. This data is able to be extra flexible means it provides a lot more latitude to do different kinds of analysis.

Benefits of Data lake

- Data lake supports the growing thrust of data analysis and data science models as well as critical requirement of data governance.
- It facilitates the modern data management platform built on enterprise scale platform which provides easy access for business users.
- Provides granular level access control based on the role based permission level.

Data Swamp - A data swamp happens when a data lake consists of miscellaneous data which no longer has any sort of structure. A data swamp can occur when adequate data quality and data governance measures are not implemented. Sometimes, a data swamp can also arise from a data warehouse due to hybrid models.



- If a data lake holds too much of data in a poorly organized manner without suitable metadata management and reliable data governance, relevant data becomes increasingly difficult to find.
- The information content of the data lake decreases, even though new data is constantly being added. A lack of lifecycle management of the data also leads to the silting up of a data lake.
- Data loses its relevance, if the data still remains in the data lake, more and more data with a lack of relevance accumulates over long periods of time.
- Incorrect time stamps of data set also leads to information which cant be found or evaluated.

Features of Data Swamp

1. Big Data without any organization and documentation through a data catalog or role concept.
2. Missing metadata information of the structured and unstructured data
3. Outdated and faulty data
4. No CDO or product management platform users
5. Missing or broken relationships between the datasets.

Prevention aspects

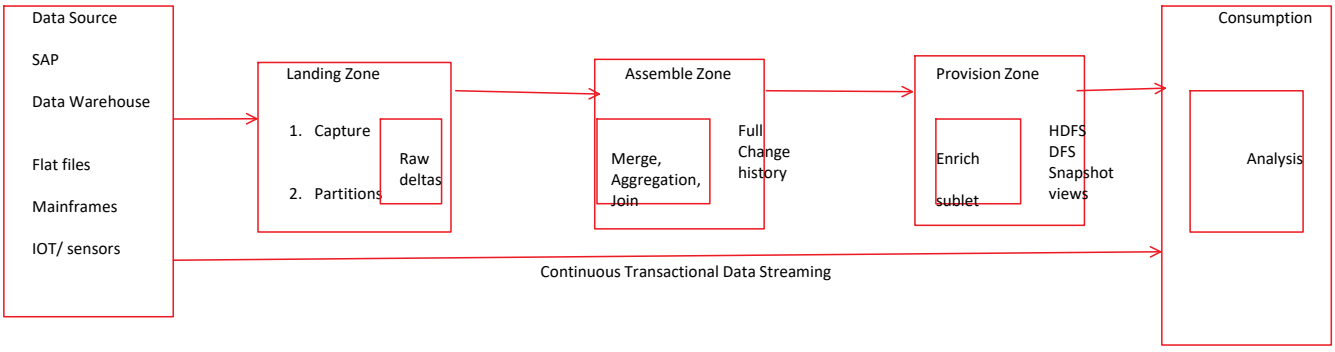
1. Create data catalog which builds the actual clarity of the data. It ensures that data reaches to the right personas based on RBAC.
2. Faulty/old data should be archived
3. Labelling of data origin, metadata labelling and meaningful nomenclature.

Data Marts

A data mart is curated subset of data which's generated for analytics and BI users. Data Marts are often created as repository of pertinent information for a subgroup of workers or a particular use case.

Difference between Data Mart and DW

- Slow and overloaded data warehouses are often the underlying reason for the creation of data marts and frequently serve as their underlying data source.
- Often, the data volumes and analytics use cases increase, enterprises cant serve every analytics use case without degrading the performance of their data warehouse, so they export a subset of data to the mart for analytics.



a) Massive volumes of structured and unstructured data like ERP transactions, and weblogs can be stored in a cost effective manner.
b) Data is available for use for faster transactions by keeping it in a raw state
c) A broader range of data can be analysed in new ways to gain unexpected and previously unavailable insights.

	Data Lake	Data Warehouse
Data Storage	A data lake contains all of an organization's data In a raw, unstructured format and can store the data for indefinite period of time. Even can be accessible for immediate or future use	A data warehouse contains the structured data such that it can be cleaned and processed, ready for strategic analysis based on predefined business needs.
Users / personas	Data from data lake with its huge volume of unstructured data can be used by data engineers, data scientists who prefer to study data in its raw form to gain new business insights	Data from DW is typically being accessed by managers, business stakeholders looking for the quick insights from the business KPI and as the data has been structured to provide answers to the pre-determined queries for analysis
Schema	Schema is defined after the data is stored in the data lake, unlike data warehouse making the process of capturing and storing the data faster.	In DW, the schema is defined before the data is stored, the lengthens the time it takes to process the data, but once it's completed, the data which made in DW, is ready as consistent, confident to use across the org.
Processing	ELT (Extract, load, transform), in this process the data is extracted from the source for storage in the data lake, and it's structured only while required	ETL (Extract, Transform and Load). In this process, data is extracted from its sources, scrubbed, parsed and then make it structured for business end analysis
Cost	Storage costs for fairly cheaper in a data lake, also less time consuming to manage which reduces the operational costs	Data warehouse cost much more than the data lakes, and required more time to manage, resulting in additional operational cost.

Scenario	Azure Storage Solution
Massively scalable and secure objects for storing into cloud and utilize for cloud-native workloads, archives , data lakes , high performance computing and to store massive volumes of data in the form of objects	Azure Blob storage
Massively scalable and secure data lake solution for the high performance analytical workloads which requires the support for both blob and file storage together and can provide the granular level access control and policy	Azure Data Lake storage
Simple, secure and serverless enterprise-grade cloud based file share, migrate on-premise file system or connect on-premise file share with Azure	Azure File Share
Store high performance real time streaming messages in pub-sub mechanism with fault tolerance and scalability support	Azure Queue storage
Store the random dataset non-compliant to RDBMS standard (no primary key/ unique key, no consistency and redundancy), non-compliant to Codd's rule. The storage required proper data partitioning for consistency	Azure Table Storage
High performance, durable block storage in the scenario of business critical apps	Azure Disk storage
Synchronization of on-premise file share with Azure file share including caching support for on-prem data as a hybrid cloud file share	Azure file sync
Appliances and solutions for transferring data into and out of Azure quickly and effectively	Azure Data Box (batch processing and real time stream processing)

ADLS Gen2

1. Hierarchical namespace - It organizes the objects/ files into the hierarchy of directories for efficient data access. There's a common object store naming convention used which slashes in the name to define the hierarchy directory structure.

- This hierarchy structure becomes real with ADLS gen2, operations such as renaming or deleting of a directory, there's no need to enumerate and process all objects which share the name prefix of the directory.
-

a) Performance wise it improves the directory management operations, which overall makes efficient job performance.

b) Management wise earlier, we can organize, manipulate the files through directories and subdirectories.
c) Security wise, it's enforceable to apply POSIX permission on the directories and individual files.
d) ABFS driver is used to access the data in the ADLS gen2, it's available within all Apache hadoop environment. The env includes Azure HDInsight, Azure Databricks and Azure Synapse analytics.

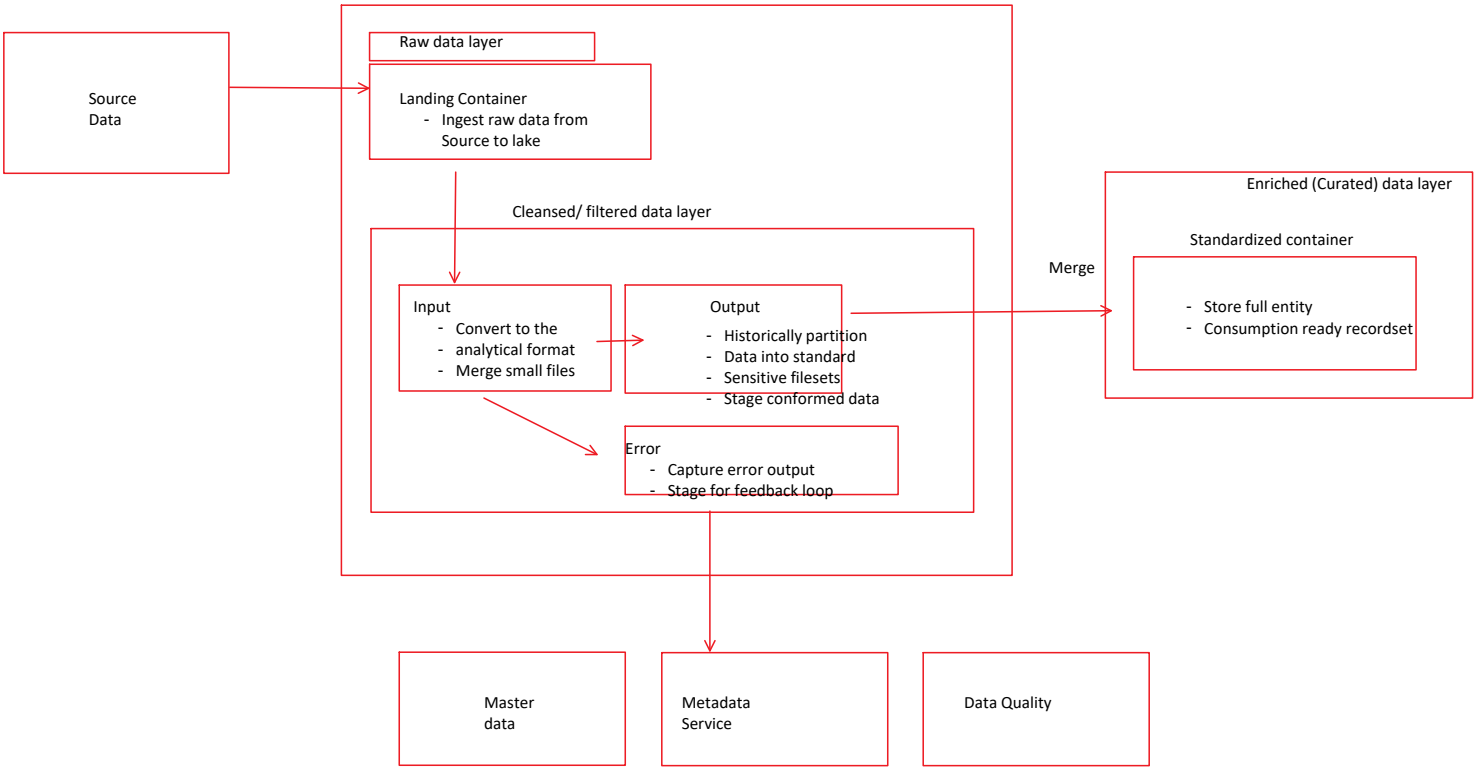
1. User delegation SAS - A user delegation SAS is secured with Azure AD credentials, also by the permission specified for the SAS. A user delegation SAS applies to Blob storage only.

2. Service SAS - A service SAS is secured with the storage account key. A service SAS delegates access to a resource in only one of the Azure storage services. Blob storage, Queue storage, Table storage or Azure files.

3. Account SAS - An account SAS is secured with the Storage account key. An Account SAS delegates access to resources in one more of the storage services. All of the operations available via a service or user delegation SAS are available via an account SAS.

From an user perspective

- Delegate access to service level operations
- Read, write and delete operations which aren't permitted with a service SAS.



Difference between ADLS Gen1 and ADLS Gen2

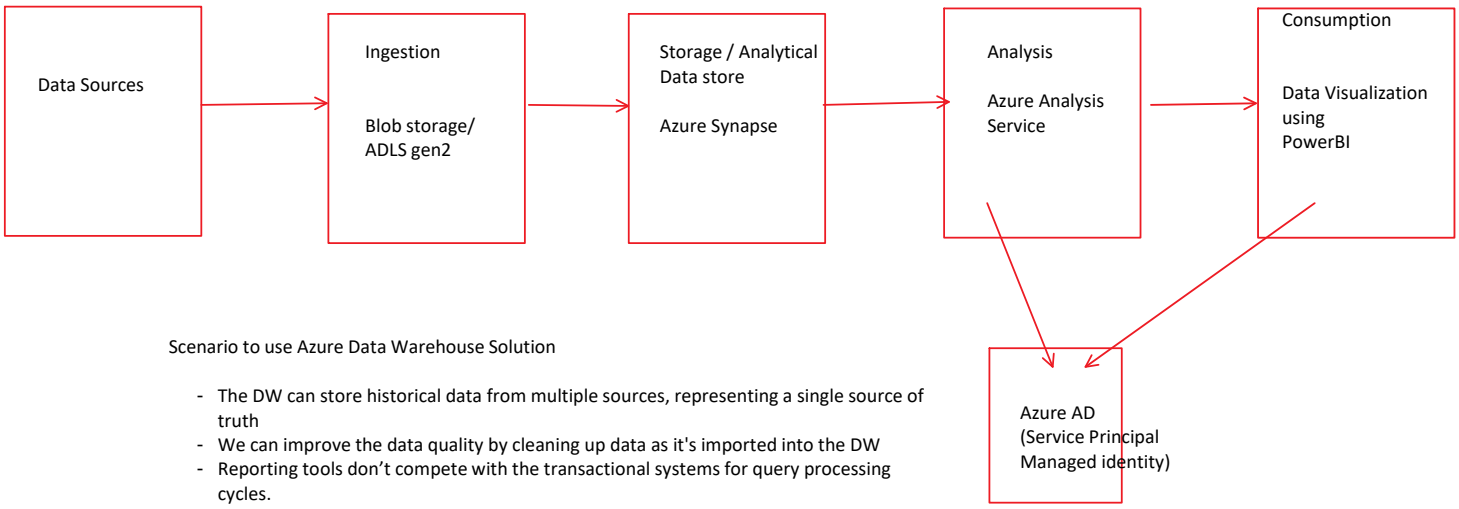
Difference	ADLS Gen1	ADLS gen2
Account Root Permissions	Permissions required for account - root - RX (minimum) , read only / read-execute or RWX (read-write-execute) to get an account root content view	An user with or without permission on root container can view the account content
RBAC roles and ACLs	All users in RBAC owner role are superusers. All other users (non-superusers) need to have permissions which should abide by the file/folder level ACL.	All users in RBAC storage blob data owner role are superusers. Rest of other users can be provided with different roles (contributor, reader) etc. which can govern their read, write and delete permission.
Store default permission	Permission for an item (file/directory) cant be inherited from the parent directory to child	File store permissions can be inherited if the default permission

	directory.	is set, on the parent directory / items before the child directory/items are being created. The file store permissions can be inherited from parent to child directory/folder.
Nested file / directory	Check whether the write-execute (wx) permissions for owner is imposed in the sub directory	Does not add wx permissions in the subdirectory
User provided permission	When a file/directory is created, the final permission will be same as the user provided permission.	File/directory is created, the final permission will be calculated based on the [user provided permission + umask] value.
Umask	Clients can apply umask on the permission for the new file/directory before the request sent	Clients can provide umask as the requested query params during the file and directory creation. The default umask will be applied 027 on the file/directory level.

Umask is used to modify the default ACLs are set on the child item when creating a file/folder. Umask is a 9-bit value on parent folders which contains an RWX value for owning user, owning group and other.

Azure SQL Data warehouse

14 February 2023 09:31



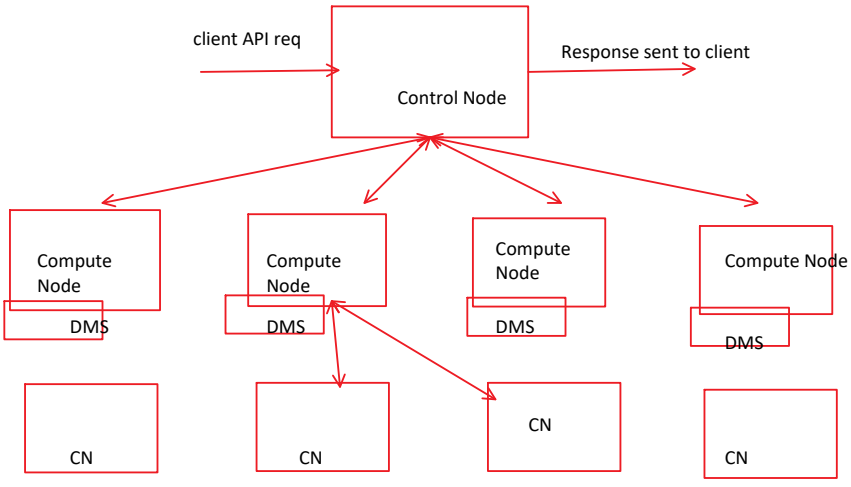
Scenario to use Azure Data Warehouse Solution

- The DW can store historical data from multiple sources, representing a single source of truth
- We can improve the data quality by cleaning up data as it's imported into the DW
- Reporting tools don't compete with the transactional systems for query processing cycles.
- A DW allows transactional system to focus on handling writes, while the DW also satisfies the majority of read requests.
- Data Warehouse can also consolidate data from different sources post transformation
- Data mining tools can find the hidden patterns in the data using automatic methodologies
- Data warehouse make it easier to provide secure access to authorized users, while restricting access to others. Business users don't need to access to the source data.
- DW makes it easier to create BI solution like OLAP cubes.

Scenario to choose for MPP architecture (Azure Synapse Analytics (formerly Azure SQL DW)

- If the data size exceed over 1 TB and are expected to continuously grow .
- If the data sizes are smaller, but the workloads are exceeding the available resources of the SMP multiprocessing (i.e. Azure SQL db etc.) then consider SQL DW (MPP architecture)
- MPP multiprocessing architecture (SQL DW) can be scaled out by adding more compute nodes (which have their own CPU, memory and I/O subsystems). There're physical limitations to scaling up a server which point scaling out is more flexible.
- In terms of querying, modelling and data partitioning, MPP solution as a complete decoupled compute and storage layer provides efficient data warehouse model.

Dedicated SQL Pool Architecture



- In the dedicated SQL pool, the control node works as the brain & orchestrator of the MPP engine
- As client API request, sent to dedicated sQL pool, the control node processes the query and converts the code based on distributed SQL plan. It's being executed on cost based optimization engine.
- After the DSQL plan has been generated, for each subsequent step, the control node sends the command to execute in each of the compute resources.
- The compute nodes are the worker nodes. They run commands as provided to them from the Control node.
- Compute node is measured using the SQL data warehouse units (DWU).
- The smallest compute unit (100 DWU) consists of control node and a compute node.
- Within the control & compute nodes, the data movement service (DMS) component can handle the movement of data between the Compute nodes themselves or from Compute nodes to the control node.
- DMS also includes the Polybase stack. An HDFS bridge is implemented within DMS to communicate with the HDFS file system.
- Polybase for SQL DW supports both Azure blob & ADLS gen2 storage.

Polybase in Azure SQL DW

Polybase is the fastest and most scalable SQL DW loading method which uses tSQL to combine and bridge the data across the RDBMS, Azure blob storage, ADLS gen2, hadoop distributed file system

- Polybase is one of the recommended option to load data into SQL DW.
- Polybase can load data from gzip, bzip2, snappy compressed files.
- Data loading onto SQL DW via polybase is not limited by the control node and as to scale out the DWU. The data transfer throughput also increases.
- We can use INSERT INTO clause in order to load incremental data into SQL DW. For full logging operation, when inserting into a populated partition which will impact on the load performance. The roll back operation on a large transaction can be expensive. Recommended to split over a big transaction into smaller batches.

Best Practices for Polybase for data loading into SQL DW

- A single polybase load operation provides the best performance.
- The load performance scales as we increase DWUs.
- Polybase automatically parallelizes the data load process, so we don't need to explicitly break the input data into multiple files and issue concurrent loads, unlike some traditional data loading methods. Each data reader automatically can read 512 MB data for each file of Azure blob storage and 256 MB for ADLS gen2.
- Multiple readers will not work against the compressed files - gzip files. Only a single reader can be used per gzip compressed file since uncompressing the file in the buffer is single threaded.

Partitioning as effective data loading methodology

1. Partitioning can also be used to improve the query performance.
2. A query which applies a filter to partitioned data can limit the scan to only qualified partitions. This method of filtering can avoid a full table scan and can only scan a smaller subset of data.
3. With the introduction of clustered columnstore indexes, the predicate elimination performance benefits are less beneficial, for e.g. if the sales fact table is partitioned into 36 months using the sales date field, then querying over the filter on the sale date can skip searching for partitions which don't match the filter.

Distributed table

A distributed table appears a single table but the rows are actually stored across 60 distributions. The rows are distributed with a hash or round-robin algorithm.

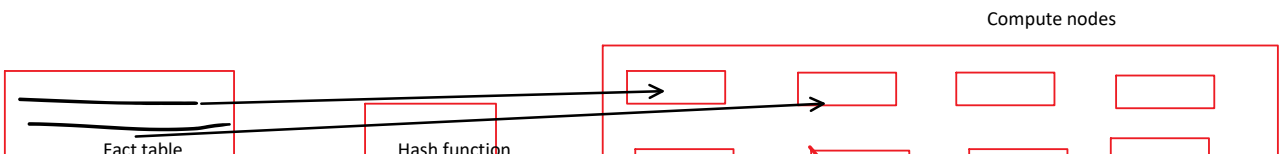
Hash-distribution improves the query performance on large fact tables.

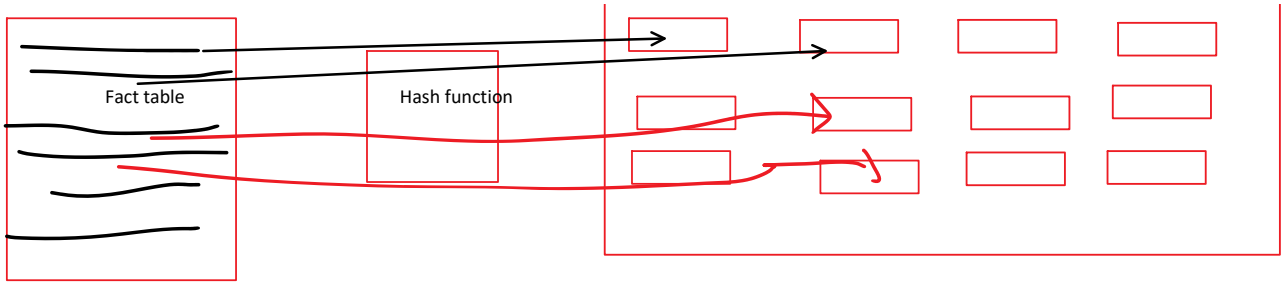
Round-robin distribution is useful for improving the data loading speed.

Another table storage option is to replicate small tables across all the compute nodes.

1. Hash distributed table (for Fact table)

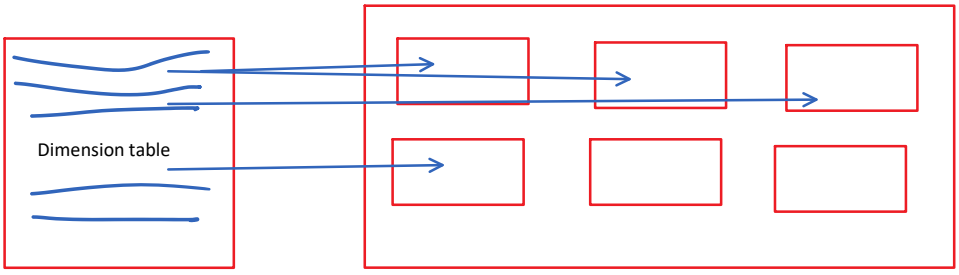
A hash-distributed table distributes table rows across the compute nodes by using a deterministic hash function to assign each row to one distribution.





- A hash-distribution table rows maps across the Compute nodes by using a deterministic hash function to assign each row to one direction.
- Identical values always hash to the same distribution unit.
- SQL DW analytics, has a built-in knowledge of the row location of tables.
- In dedicated SQL pool, the knowledge is utilized to minimize the data movement between the queries, which also improves the query performance.
- For star schema, large fact tables are designed with the hash distributed table format.
- When the table size on disk 2-5 GB, table has large no of rows and frequent insert, update and delete operations.

2. Round-robin distribution (for dimension tables)



- Distributes the table rows evenly across all distributions
- The assignment of rows is used for distribution is random
- Rows with equal values are not guaranteed to be assigned in the same direction.

Scenarios

1. Design the dimension table with large no of columns
2. There's no specific joining key
3. When the table is temporarily stored staging table
4. There's no common joining key available with other key tables.

Table Distribution	Hash-distributed	Round-robin	Replicated
Scenario	Large fact tables consisting of large no of Rows	Dimension tables with large no of columns	Dimension tables with full copy option from round-robin value
Size	Size of the fact table is at least 5 GB or more	Dimension table size is at least of 2 GB	Replicated tables 2 GB or less
Performance efficiency	Most performance efficient since hash distribution key is imposed using the distribution column	Less performance efficient since the table rows are distributed randomly over the compute nodes of the cluster, not matching same order the compute nodes while distribution of rows.	Full table copy is performed over the same compute node, less data movement (DMS) , maximize the query throughput and performance

Azure Databricks

14 February 2023 09:31

Data Analytics

14 February 2023 09:32

Alteryx

14 February 2023 09:32