Pre-requisites

09 September 2022 08:34

Lab Setup Requirement	Hardware CPU - Intel Core i3/i5/i7 processor, RAM - at least 8 GB HDD-512 GB / 1 TB, OS - Windows 10 /8.1/11, MS Word/excel, PowerBI desktop (optional)

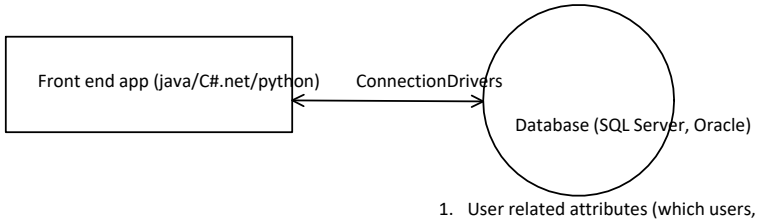
Pre-requisites
Software - SQL Server 2016, 2017 or 2019 Enterprise/Developer edition, Visual Studio 2019/2022/VS code, A Valid Azure Subscription, Azure CLI, Storage explorer, SQL Server Management Studio/Azure Data Studio, Microsoft Azure Subscription, Git tools and GitHub account, Azure PowerShell, PowerShell ISE, Note: Linux environment VMs (Apache hadoop/big data) (Linux OS) Tool: Putty.exe Azure based Linux servers, Vmware player/ virtual box AWS Tools for VS code, GCP Tools for VS code.

Azure Subscription (trial)
-- 12 months of free service + 30 days of 200 USD credit
-- enterprise subscription

Database Fundamental & SQL Server BI 2016

08 September 2022 18:45

Application metadata



1. User related attributes (which users, port, MSSQL - 1433, encryption, protocol (TCP)
2. SQLConnection
3. ADOConnection (ADO.Net)

Data Dictionary

1. Names of all of the database tables and their schemas (Sales, Customers, Orders, Employees…)
2. Details of all the tables in the database like owners of the tables, the security constraints, when the tables were created etc.
3. Physical information of the tables in the database - where the tables in the db itself have been stored and how
4. Table constraints includes primary key information, foreign key information etc.
5. Information related to database views which are visible

Employee Table - Active Data Dictionary -- self updating

Passive Data Dictionary -- manually updated to match the database

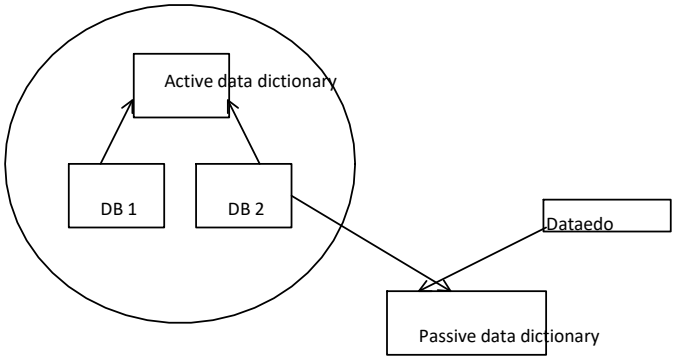
Field Name	Data Type	Field size for display	Description	Example	Dept_id
Employee No	INT	10	Unique ID for the employee	444007	1
Employee Name	VARCHAR(50)	20	Name of the Employee	James Hobb	23

Example of Passive Data Dictionary -

Dataedo -- tools

Column	Data Type	Description
Field Name	10	
Data Type	20	

Dataedo



Different Types of Database

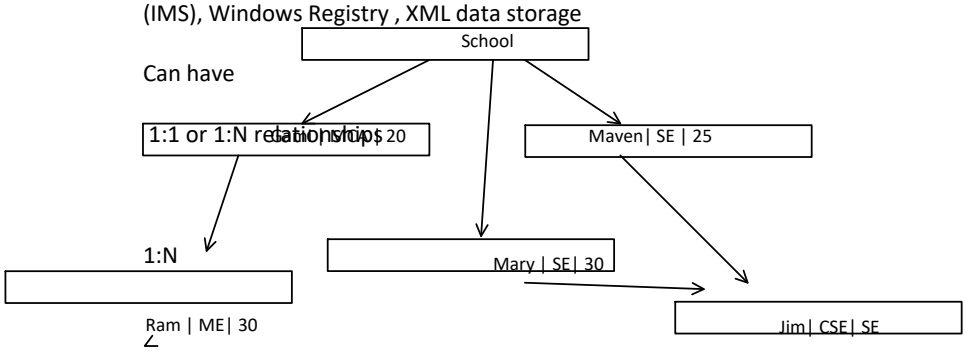
1. Relational Database - consists of set of tables with columns and rows
2. Object-oriented database - information can be presented in the form of objects as in object-oriented programming. Inclined towards into objects e.g. multimedia record in a relational database can be defined as definable data object, MongoDB has offering of Object oriented database
3. Distributed database - consists of two or more files located in different sites, e.g. SQL server mirror databases, distributed dbs

4. Data Warehouses - central repository for data storage, includes a type of database designed for faster query and analysis(SSAS, SSIS)
5. NoSQL databases - non-relational db - support for unstructured, semi-structured data, dynamic schema, flexible and faster data retrieval (Cassandra, Mongo DB, Couch DB, Azure Cosmos DB, AWS Document DB, AWS Dynamo db)
6. Graph Databases - nodes - entity , attribute - relationship

e.g. Apache Tinkerpop , Azure Cosmos db Graph API , Neo4j

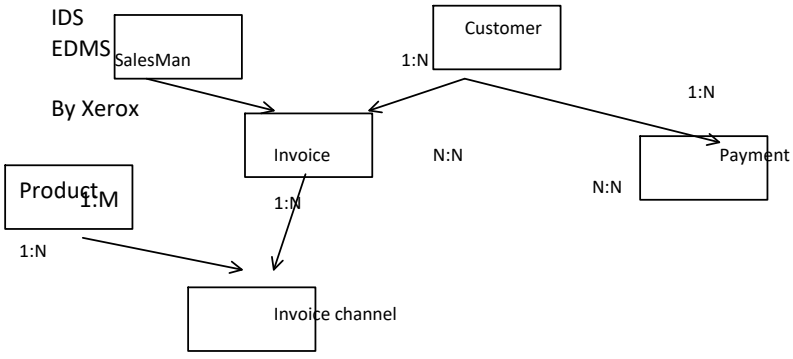
7. Cloud databases - Databases as a service (DBaaS) e.g. Azure SQL database, Azure SQL managed instance
8. Document / JSON database - designed storing, retriving and managing document oriented information. (Azure Cosmos db SQL API, document db, AWS document db, data being stored in key-value pairs,

Hierarchical data model - COBOL (DB2) - IBM Information Management System (IMS), Windows Registry , XML data storage



Network data model

1. An owner record which is the same as of the parent in the hiearchical model
2. A member record which is same of child in the hiearchical mode



Employee Table 1

Fields	Columns(attribute 1) Emp ID	Columns(attribute 2)	Attribute 3
Row 1 (records/tuples)	1001		
Row 2 (records/tuples)	1002		

Employee ID - foreign

Table 2
EmployeeAddress

Fields	Columns(attribute	Columns(attrib	Attribute

Hiearchy Data models

Mainframes DBMS for IBM

IBM IMS and RDM Mobile - embedded db

Employee table

Emp No	First Name	Last Name	Dept
1001	Alan	Turing	Finance

Device table

Serial no	Type	User emp no
001	Monitor	1001

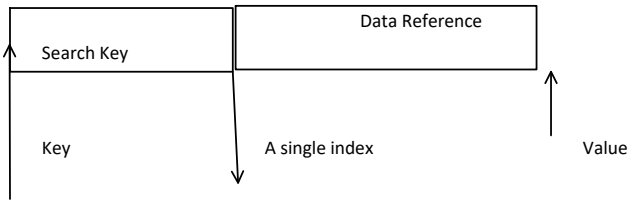
	1) Emp ID	ute 2)	3
Row 1 (records/tuples)	1001		
Row 2 (records/tuples)	1002		

Entity Integrity - ensures the primary key in a table is unique and the value is not set to null

Referential Integrity - requires every value in a specific foreign key column should be found in the primary key of the table from which it is originated.

Index

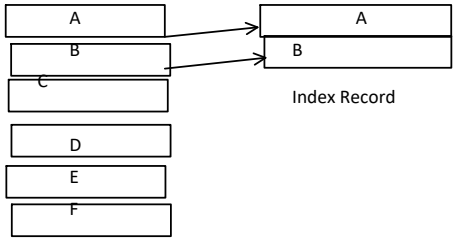
- First column for a table is the Search key which can contain a copy of the primary key of the table. These values are stored in sorted order so that the corresponding data access can be faster.
- The second column is the data reference or Pointer which can contain a set of pointers holding the addresses of the underlying disk blocks where the specific key values are stored.



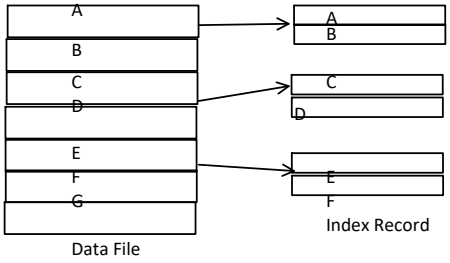
1. Access Types - value based search, range of access over data records
2. Access Time - time required to find the data element
3. Insertion time - time taken to find the specific space and to insert the new data
4. Deletion time - time taken to find an element & to delete it
5. Space overhead - additional space required by an index

File storage mechanism to follow for indexing

1. Sequential file organization -



Dense Index



Sparse Index

Foreign Key

1. The foreign key constraint is used tp prevent actions what would destroy links between tables.
2. A foreign key is a field (collection of fields) on a table refers to primary key in another table.
3. A table with the foreign key is called as child table, and the table with the primary key is called parent/referenced table.

s			

2. Hash file organization - indices are choosen based on values being distributed uniformly. Hash buckets where the value is assigned determined by hash function.

a) Clustered index - you can define an index with upto 16 columns , The max size of this index should be 900 bytes

The columns defining for clustered index is termed as clustering key.

SQL server to order the data in the table in accordance to the clustering key.

a) Non clustered index

- Do not impose a sort order on the table
- Restriction wise max size supported as 900 bytes & can be promoted max 16 columns of a table, max 249 non-clustered indexes can be created on a table.

WHERE clause used to specify the condition while fetching the data from single table or joining from multiple tables. When, a given condition is satisfied, use the WHERE clause to filter the specific records and fetching the necessary records.

Surrogate Key

School A

Reg No	Name	% obtained
201010	Brian	66

Reg No	Name	% obtained
201010	Brian	66
202012	Max	50

School B

Reg No	Name	% Obtained
CS300	Ava	50
DS500	Maria	60

Merging these two tables in a single sql table

1. Automatically generated by the system
2. It hold anonymous integer
3. It contains unique value for all records in the table
4. Value cannot get modified
5. Easier identification purposes

Surr_id	Registration no	Name	% obtained
1	201010	Brian	66
2	202012	Max	50
3	CS300	Ava	50

from single table or joining from multiple tables. When, a given condition is satisfied, use the WHERE clause to filter the specific records and fetching the necessary records.

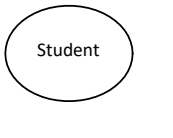
ALTER ID, NAME, SALARY FROM CUSTOMER WHERE NAME = 'XYZ'
DELETE ID, NAME, SALARY from CUSTOMER WHERE NAME = 'ABC'

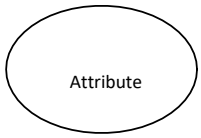
a) Multilevel index

ER Modelling

Entity - A Entity is an object with a physical existence - a particular person, car, house,

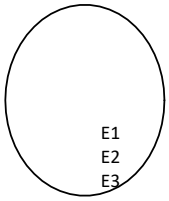
A entity is an object of entity type & set of all entities is called as entity set.

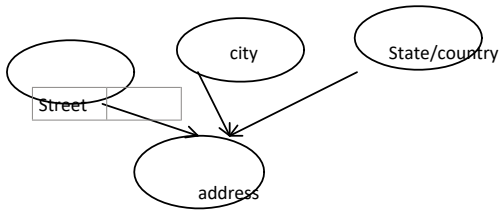


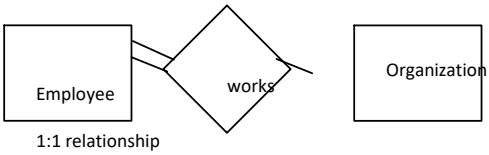


Student_roll_no
Student_name
Student_DOB
Student_Address

Key Attribute -
student_roll_no







1:1 relationship
Students - enrolled to - Courses

M:N M:1

Unary relationship

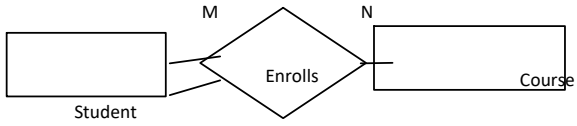
Multivalued attribute

Derived attributes
Student_age,
Employee_bonus

Participation constraint

1. Total Participation - each entity must be participated in the relationship.
2. Partial Participation - entity in an entity relationship may or may not participate in the relationship.

Employees --- Taking leaves -- During vacation (M:N)



PersonId (Primary Key)	LastName	FirstName	Age
1	Bill	Johns	30
2`	Maria	Sophia	21

SQL Command

1. DDL - CREATE, DROP, ALTER, TRUNCATE
2. DML - INSERT, UPDATE, DELETE
3. DCL - GRANT , REVOKE
4. TCL - Commit, Rollback, Savepoint
5. DQL - SELECT

Orders table

OrderID	OrderNumber	PersonID (Foreign Key)
1	77445	2
2	88445	1

The foreign key constraint prevents invalid data from being inserted into the foreign key column, since it has to be one of the contained value in the parent table.

Normalization

09 September 2022 18:33

- Normalization is the process of organizing the data into the database
- Normalization is used to minimize the redundancy from a relation or set of relations. It's also defined to eliminate the undesirable features like insertion, update and deletion anomalies.
- Normalization divides the large unnormalized tables into smaller and links them using relationships.
- The normal form is used to reduce the redundancy from the database level.

Purpose of normalization -

Data modification anomalies can be differentiated into the types:

1. Insertion Anomaly - a new row/tuple cant be inserted into a relationship due to lack of data
2. Deletion Anomaly - The delete anomaly refers to the scenario, where the deletion of data from the row results in loss of some other important data due to lack of proper key based attribute level relationships.
3. Updatation Anomaly - The update anomaly can exist when an update operation of a single data value requires multiple rows/tuples of data to be amended/updated.

	1NF	2NF	3NF	BCNF	4NF	5NF
Conditions	Elimination of repeating of groups	Eliminate the partial functional dependency.	Reduce the transitive dependency	More advanced level than 3NF. More stricter enforced for 3NF.	Eliminates the concepts of multi-values dependency .	Eliminates the concepts of joining dependency
Feature	A relation in first normal form when it consists of only an atomic value. One attribute contains only one value for a specific row.	Tables should be in 1NF + non-key attributes which are fully functional type they must dependent on the primary key	The relation fulfills the criteria for 2NF + no transitive relationship exists.	A table in BCNF, if there is a functional dependency exists x-> y (x is assumed to be the super key). Table should be in 3NF. For every functional dependency, the left side of relationship of the table fulfills the criteria for super key.	The table should in BCNF, it should have no multi-value dependency .	The table should be in 4NF + there cant be any join level dependency exists in the table, joining of the table should not incur any data loss or any joining should not have any particular loss.

Benefits of normalization -

1. Reduce the data redundancy
2. Greater data organization & consistencies.
3. Flexible level of database design
4. Enforce the concepts the referential integrity

First Normal Form (1NF)

- The table should be in the form where the one attribute should contains only one value based on specific row / tuple
- A table should have atomic values (no duplication of values on attributes) , enforce non-repetitive groups/attributes

- The column should have one single valued attribute.

Unnormalized table - Employee table

Employee_id	Employee_name	Employee_Phone	Employee_Email	Employee_HireDate	Employee_Salary
001	Mark	1988233222 111222333	mark@contoso.io Mark.b@fabrikum.com	01/01/2021	4000
002	John	222444111 444333777	john@contoso.com jo@adven.com	03/02/2017	3500

1NF is fulfilled for Employee table

Employee_Id	Employee_Name	Employee_Phone	Employee_Email	Employee_HireDate	Employee_Salary
001	Mark	1988233222	mark@contoso.io	01/01/2021	4000
001	Mark	111222333	Mark.b@fabrikum.com	01/01/2021	4000
002	John	222444111	john@contoso.com	03/02/2017	3500
002	John	444333777	jo@adven.com	03/02/2017	3500

2NF - Second Normal Form

- To be in 2NF, the tables should be in 1st Normal Form.
- All non-key attributes should be fully functional dependent on the primary key of the table

In this table, non-prime attribute Employee_Name is dependent on the Employee_ID which is proper subset of a candidate key.

Employee_detail table

Employee_ID	Employee_Name	Employee_HireDate
001	Mark	01/01/2021
002	John	01/01/2021
003	..	..

Employee_Salary table

Employee_ID	Employee_Salary	
001	4000	
002	3500	

Employee_Contacts table

Employee_ID	Employee_Phone	Employee_Email
001	1988233222	mark@contoso.io
002	111222333	john@contoso.com

Professor table (1NF)

ID	Course	Univ Name	Name
10	CSE	Stanford	
15	IT	Harvard	

15	ME	Harvard	
30	DB	Princeton	
30	CA	Princeton	

2NF

Professor_detail table

ID	Univ Name	Name
10	Stanford	Mark
15	Harvard	Mark
30	Princeton	John

Fulfills the partial dependency.

Professor_subjects table

ID	Courses
10	CSE
15	IT
15	ME
30	DB
30	CA

Third Normal Form (3NF)

- A table can be in 3NF, if it is in 2NF, should not have any partial functional dependency
- It should reduce the data duplication
- It can achive the integrity
- No transitive dependency between non-prime attributes/columns.

  A -> B -> C => Column A is dependent on B , B dependent on C, if A is also dependent on C, then it's called as transitive dependency.

  Employee_details (2NF)

Emp_id	Emp_Name	Emp_Zip	Emp_State	Emp_City
222	Mark	70045	Arizona	Phoenix
333	Harry	34404	Utah	Lake
555	Jerry	40032	Arizona	Maveric
335	Hannah	33406	New Mexico	Titan

  Super Key relation -> (Emp_id), (Emp_id) (Emp_Name), (Emp_id)(Emp_Name)(Emp_Zip)…

  Candidate key -> Emp_id

  Delhi - 11….
  Mumbai - 4….

  Emp_State and Emp_City is dependent on the Emp_Zip & Emp_Zip is dependent on the Emp_id. The non-primary key attribute (Emp_State) and (Emp_City) transitively dependent on primary key (Emp_ID). It violates the criteria of 3NF.

  Employee tbl

Emp_ID	Emp_Name	Emp_Zip

222	Mark	70045
333	Harry	34404
555	Jerry	40032
335	Hannah	33406

Employee_zipcode table

Emp_Zip	Emp_State	Emp_City
70045	Arizona	Phoenix
34404	Utah	Lake
40032	Arizona	Maveric
33406	New Mexico	Titan

BCNF - (Boyce Codd Normal Form)

- It is stricter than 3NF, more advanced than 3NF.
- A table will be in BCNF if every dependency exists like with a super key to no the attribute, x -> y. (x is the super key of the table).

Employee_details table (3NF)

Emp_id	Emp_country	Emp_dept	Depart_Name	Emp_depart_no
333	US	Manufacturing	Design	001
444	UK	Software	Engineering	002
555	US	Architecture	Development	003
666	US	Machinery	Development	004

Functional Dependency ->

Emp_id -> Emp_Country
Emp_Dept -> Department_name, Emp_depart_no

Candidate key -> (emp_id, emp_dept)

This table is not in BCNF, because neither the Emp_Dept and Emp_id alone the keys.

Emp_country table

Emp_id	Emp_country
333	US
444	UK
555	US
666	US

Candidate key - Emp_id

Emp_department table

Emp_dept	Emp_Department_name	Emp_dept_no
Manufacturing	Design	001
Software	Engineering	002
Architecture	Development	003
Machinery	Development	004

Candidate Key - Emp_Dept

Third table should have both of these candidate keys

(Emp_id, Emp_dept)

Emp_dept_mapping table

Emp_Id	Emp_Dept
333	Manufacturing
444	Software
555	Architecture
666	Machinery

The left side of both the functional dependencies is a key. --> This table is in BCNF.

Forth Normal Form (4NF)

- A table is in fourth normal form (4NF), if it's in already in BCNF & has no multivalue dependency.

A -> B, if for single value of A, multiple values of B exists, then it's called as multivalue dependency.

Student table (3NF)

ID	Course_enrolled	Programming_skills
10	Computer Science	C
10	Engineering Math	java
30	IT	Networking
40	CS	Compiler Design
55	Bioinformatics	C

There's multivalue dependency exists for the student with ID=10

Course_enrollment table

ID	Enrolled_courses
10	Computer Science

10	Engineering Math
30	IT
40	CS
55	Bioinformatics

skills

ID	Programming
10	C
10	java
30	Networking
40	Compiler design
55	C

Fifth Normal Form - 5NF

- A table is in 5NF, if it's already in 4NF & should not contain the join dependency. The joining should not incur any data loss.
- 5NF is satisfied when all the table are being broken into as many as tables as possible to avoid redundancy.
- 5NF is called project level join normal form.

Employee table

Employee_Name	Employee_HireDate	Employee_Designation
John	01/01/2020	Sr. Software Engineer
Mark	01/03/2021	Software Arch
Mark	01/03/2021	Software Engineer
Celine	04/02/2014	Sr. Software Engineer
Alan	01/03/2021	Network Engineer

P1 level

Employee_Desig	Emp_Name
Sr. Software Engineer	John
Software Arch	Mark
Software Engineer	Mark
Sr. Software Engineer	Celine
Network Engineer	Alan

P2 level

Emp_Name	Emp_HireDate
John	01/01/2020
Mark	01/03/2021
Mark	01/03/2021
Celine	04/02/2014

Alan	01/03/2021

P3 Level

Emp_Design	Emp_HireDate
Sr. Software Engg	01/01/2020
Software Arch	01/03/2021
Software Engineer	01/03/2021
Sr. Software Engineer	04/02/2014
Network Engineer	01/03/2021

TSQL concepts

08 September 2022 18:45

SQL Table Design

1. Data types : A data type is fundamental constraining element of a database which restricts the range of possible values that are allowed to be stored in a column

a) Numeric data type :
- tinyint (0-255)
- Smallint(-32768 to 32767)
- Int (storage space - 4 bytes)
- Bigint(8 bytes)
- Decimal(p,s) fixed precision and s - scale numbers , precision - max total no of decimal digits can be stored , scale - number of decimal digits which are stored to the right of precision point.
- Numeric(p,s) - a constant data value can be automatically converted to a numeric data value. SQL server uses the default rounding options when converting a number to decimal or numeric one with smaller precision and scale.
- Smallmoney (-214748.00… 214748.00) - 4 bytes
- Money - 8 bytes accuracy of 10000 of monetary units.
- Real -3.4 , -1.18 to positive values (4 bytes)
- Float - 4 bytes or 8 bytes

b) Character data type
- Char(n) - 1 byte / character upto 8k bytes
- Varchar(n) - max 8k bytes
- Text - stores upto 2 gb
- Nchar(n) - 2 bytes per character max 4k bytes
- Nvarchar(n) - 2 bytes per character max of 4k bytes
- Ntext - 2 bytes per character stored upto 2 gb

  The (n) characters defined sets the max no of characters allowed to be stored in the column
  Nvarchar and varchar - the amount of storage consumed is equal to the number of characters being stored.
  Varchar(max), nvarchar(max) - 2 gb of data

c) Binary data type
  Binary data type can be fixed length or variable length

- Binary (sizes of column data entries are consistent) upto 8k bytes
- Varbinary - variable length binary data
- Varbinary(max) - storage exceeds beyond 8k bytes
- Image - variable length binary data upto 2 gb
- Alternative varbinary(max)

d) Spatial data type:
- Geography - implemented as .net CLR (latitude, longitude)
- Geometry - store points, lines, curves

e) FileStream data type:

BLOB data stored , not restricted to 2 gb of limit of file system

f) HiearchyID data type:

Storing of nodes & edges/vertices of graphs, flowcharts

SQL Server Column properties

g) Sparse Columns

The attribute of a specific row if requires very small values & need small storage space, then can use the Sparse property.

-- Temporal tables

It is a database feature which brings built-in support for providing the information about the data stored in the table in time, rather than only the data which is correct at the current moment of time.

System-versioned temporal table is kind of user table designed to keep a full history of data changes, allowing for easy point-in time analysis.

Temporal tables have two explicit defined columns with datetime2 data type, these columns are called as period columns.

Benefits

- Auditing all data changes and performing data forensics
- Calculate the data column change trends over the time
- Maintain a slowly changing dimension for the decision support apps
- Recover from accidental data damages and errors

string connectionString = "Data Source:MSSQL1;"+"Initial Catalog=sampleDB;Integrated Security=SSPI;" +
"MultipleActiveResultSets=True"

Session cache -> logical session
SqlClient driver (c#) caches the MARS session within a connection. 10 MARS session.

SQL Server Clauses:

1. SQL Order By Clause:
- Order the result set of a specific query by the specified column list and optionally it also limits the rows returned to a specified range. The order in which the rows are returned in a result set are not been guaranteed unless an ORDER BY clause is defined
- Determine the order om which Ranking function values are applied to the result set. --> Ranking function helps to return a ranking value for each row in a partition.

ORDER BY clause is not supported for CREATE TABLE AS SELECT(CTAS) statements in Azure Synapse & AAS.

ORDER BY expressions
[collate collation_name]
[ASC | DESC]

2. HAVING Clause -
- Specifies the search condition for a group or an aggregate. HAVING clause can be used only with the SELECT statement. HAVING is typically used with the GROUP BY clause. When the GROUP BY clause is not used, there's an implicit single, aggregated group.

[HAVING <search_criteria>] - one or more predicates for groups/aggregates to meet.

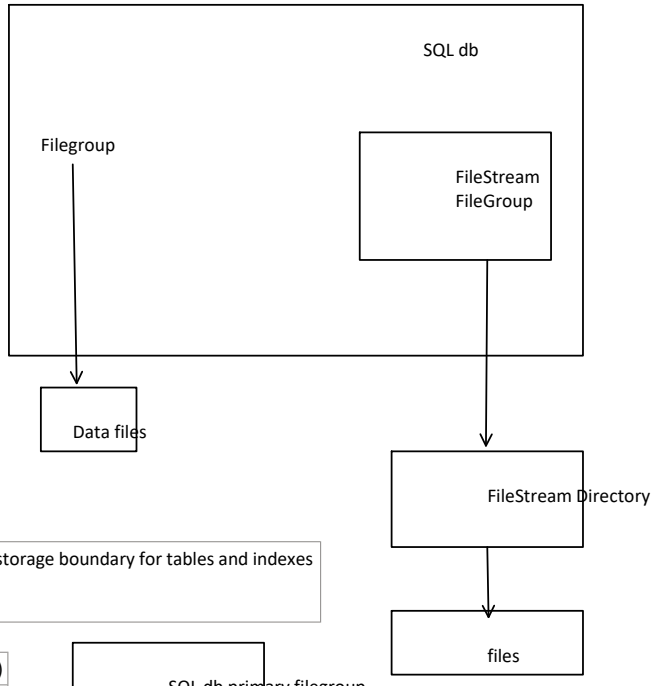
The text, image and ntext data cant work with HAVING Clause.

FileStream in SQL Server

Database structure

- .mdf file - primary database
- .ndf file - secondary db
- .ldf file - transaction log file
-
- Data and log file for SQL server

a) Physical file name Fi
b) Initial file size
c) File growth factor
d) Maximum size
e)
f)



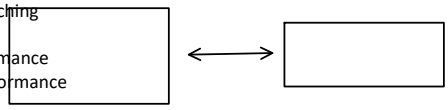
data filestream full-text	Filegroup is responsible to create storage boundary for tables and indexes

Emp id	Emp_name	Emp_phone (varbinary(max))
1	Alan	0*E98886AX998CC
2	Matt	0*F588886AX998CC

Emp id	Emp_name	Emp_phone (varbinary(max)) FILESTREAM
1	Alan	0*E98886AX998CC
2	Matt	0*F588886AX998CC

Documents are stored in the file system and the db has a particular FILESTREAM

- Does not have to use high memory and memory buffer pool for caching Large objects.
- FILESTREAM enables caching at the system cache providing performance Benefits for large media files without affecting core sql server performance



SQL server db
Primary group

File system documents

Temp tables advantages

- Store data temporarily, large datasets needed to perform data transformation and modification

-- in-memory based optimized tables, schemas and data required to store until the db restarts.
-- required to store these tables in memory pools
-- retriction in terms of memory usage but storage for disk

Index Operators

OFFSET FETCH

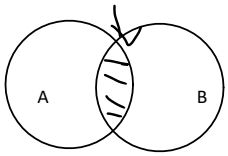
ID	Name
1	…
2	…
3	…
4	…
5	…
6	…

OFFSET clause - specifies the number of rows to skip before starting to return rows from the query.

FETCH clause - defines the number of rows to return after the OFFSET clause has been processed.

Skip first two rows and fetch next 4 rows only, we can use OFFSET and FETCH clauses with ORDER BY clause.

INNER JOIN -- helps to create a new table by combining rows which has matching values in two or more tables.

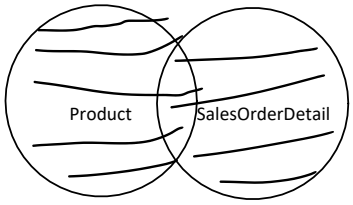


.

Outer Join - to join or match the rows between tables, want to get the matched rows along with unmatched rows from one or both tables.

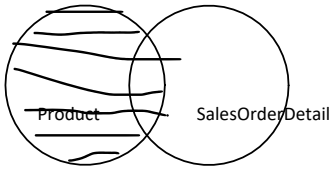
- SQL full outer join
- SQL left outer join
- Sql right outer join

Full Outer Join - In full outer join, all of the rows from both of the tables are included, if there's any unmatched rows, it will show NULL values from them.



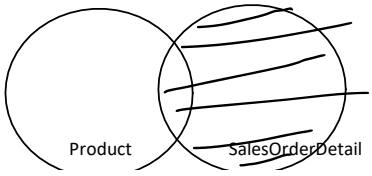
Left Outer Join - in left outer join, we can get the specific rows from the output.

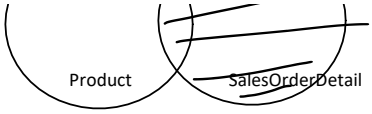
- It gives the output of the matching row/rows between both of the tables.
- If no records are found to have matching, it will show such records with null values.
- Based on the joining clause on the two tables are specified, all data is returned from the left table.
- On the right table, the matching data is returned in addition to the NULL values where a record exists in the left table, but not in the right table.



Right Outer Join - Based on two tables, specified in the JOIN clause, all data is going to return from the right table. On the left table, the matching data is returned in addition to NULL values where a record exists in the right table but not in the left table.

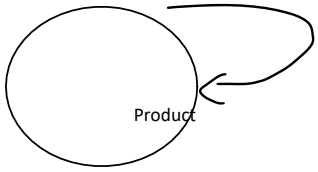
- It gives the output of matching row between two tables.
- If no records are matching from the right table, it will show these records with the NULL value.





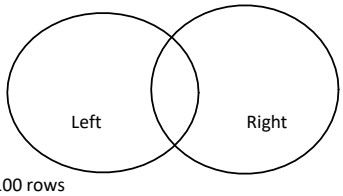
Self join - in any practical circumstances, the same table is specified twice with two different aliases in order to match the data within the same table.

Self join when it's a requirement to create a result set joining the records in the table with some other records in the same table.



Cross join - Based on two tables specified in the Join clause, a Cartesian product is created, if a WHERE clause does the filtering for the rows. The size for the cartesian product is based on the multiplication of the number of rows from the left table by the number of rows in the right table.

- Cross join returns all rows for all of the possible combinations for two tables.
- It generates all the rows from the left table which is then combined with all of the rows from the right table.
- This kind of joining is called Cartesian product (A*B)



Left = 100 rows Right = 100 rows

100 * 100 = 10,000 rows

Employee table

Emp_Name	EmpSalary	Rank_id
Alan	500	1
Alan	800	1
alan	400	1
Rachel	600	4
rachel	400	4
Tony	300	6

Row No
1
2
3
4
5
6
7
8
9
10

Pivot table:

Student table

Student	Subject	Marks
Jacob	Maths	100
Jerry	CS	70
Mark	Science	80

Student	Maths	CS	Science
Jacob	100		
Jerry		70	
Mark			80

VendorID	Year	
	2001	
	2002	

VendorID	2001	2002	2003

	2003	
	2004	

SQL Server View

A View in SQL Server is simply a SELECT statement which has been given a name and stored in a database.

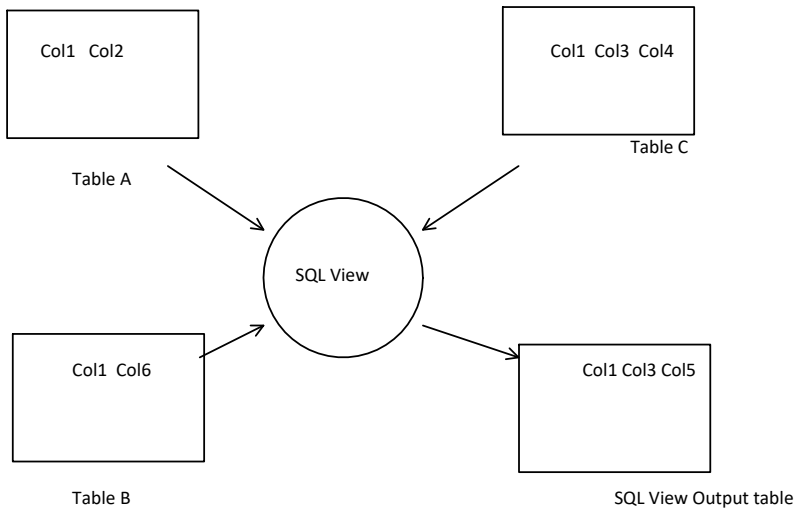
Data is stored in RDBMS in the form of tables, stored procedures, views etc.

Drawbacks:

- Normalization is a database process which is used for organizing the data in the database by splitting the large tables into smaller tables.
- These multiple tables in SQL server are linked using the relationships.
- Developers who are writing queries to retrieve those data from the multiple tables and columns, they need to perform multiple joining and complex queries.

To overcome all of these challenges, SQL server has the concept of Views.

- A view in SQL server is a virtual table which contains the data from one or multiple tables.
- Similar to like SQL table, the view name should be unique in the database.
- View contains a Set of predefined SQL queries to fetch the data from the database itself.
- So, a view contains database tables from the single or multiple databases as well.
-



- SQL Server View can retrieve the data from multiple tables
- It can show the View output in the output table

1. Create a SQL View

```
Create view view_name
As
Select column1, column2, column3…. columnN from tables
Where conditions;
```

Features of SQL Server View:

- Since View is a stored name for a SELECT statement, the SELECT statement which is defined for the View , can reference tables, views, and functions.

Core Features

- **The select statement contain the COMPUTE and COMPUTE BY clause**
- **USE the INTO keyword**
- **Use an Option clause**
- **Reference a temp table or variable of any type**
- **Contain an ORDER BY clause unless a TOP operator is specified.**
- **The View can contain multiple SELECT statements as long as can define the UNION and UNION ALL operators.**

SQL Server Stored Procedures

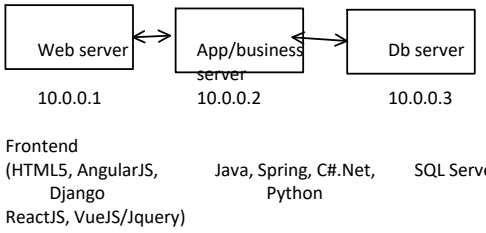
SQL Server stored procedure is a batch of statements grouped together as a logical unit and stored in the database.

The stored procedure accepts the parameters and executes the T-SQL statements in the procedure in SQL Server.

Benefits for Stored Procedure

-- It can be easily modified : we can easily modify the code inside the stored procedure without the need to restart or deploying any application. For e.g. Whenever the logic needs to change, we just to execute the procedure with a simple ALTER PROCEDURE command.

-- Reduced network traffic - procedures can passed over the network instead of the Whole tsql code.



-- Reusability - stored procedures can be executed by multiple users or multiple client Apps without the need to write code again.

-- Security -- Stored procedures can reduce the threat and vulnerabilities by eliminating Direct access to the tables. Applies the encryption by encrypting the stored procedure.

-- Performance efficiency - The SQL Server stored procedure while executed for the first time, it creates A plan and stores it in the memory buffer pool so that the plan can be used in the next time when the same query / procedure is executed.

Drawbacks of Stored Procedure
-- Testing & Debugging : testing of logic encapsulated in Stored procedure is difficult.

-- Debugging -- not possible in stored procedure

-- Version Control --- not supported in SP

-- Cost --
-- Portability - in terms of Versioning and Vendor product aspect Oracle -> SQL Server

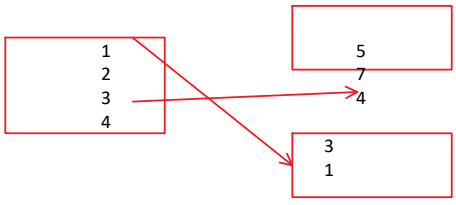
A sql index is a quick lookup table for finding records for users as required to search without going for the entire table scanning.
SQL indexes are kind of performance tool which helps to optimize the searching of records from the tables without row by row search operations.

Clustered index

a) The data we can move in memory has to be in sequential or sorted order
b) There should be a key value means it cant have repeated values.
c) It will perform sorting on the tables only with clustered indexes
d) Only one clustered index in a table for general cases
e) Same like dictionary where data is arranged in alphabetical orders.
f) Index contains the pointer to the block but not direct data

Non-clustered index

a) The data is stored in one place and index is stored in another place.
b) Since , in non-clustered index, data and the pointer is stored seperately.
c) Hence, it's very common to have multiple non clustered indexes in a table



SQL Functions

11 September 2022 22:06

Functions
In-built functions for SQL Server

Special Functions in t-SQL
- Row Number Function
- Rank and Dense Rank Function
- Calculate Running Total in t-SQL
- NTILE Function
- Lead and Lag Functions
- FIRST VALUE Function
- Window Functions
- LAST VALUE Function
- PIVOT and UNPIVOT
- CHOOSE Function
- IIF Function
- EOMONTH Function
- DATEFROMPATS Function

SQL Server STRING Function

CHAR	Convert an ASCII value to character
CONCAT	JOIN two or more strings into one string
DIFFERENCE	The difference (values) of two strings
FORMAT	Return a value formatted with the specified format and optional values
LEN	Returns a number of characters of a character string
Lower	Returns a string on lowercase format
REPLACE	Replaces all occurrences of a substring within a string with another substring
REPLICATE	Returns a string repeated a specified number of times
RIGHT	Extract a given a no of characters from a string starting from Right

REPLACE(input_string, substring, new_substring) -- any string expression to be search
Substring -- is the string which has to be replaced
New_substring -- is the replace string

REPLICATE('MANGO', 20) results;

RIGHT(input_string, no_of_characters)
Input can be literal string, variable, column

Input string - can be string, variable & data type can be anything except TEXT, NTEXT / VARCHAR()
-- no_of_characters - is a positive integer which defines the actual no of characters of the input_string to be returned.

SELECT RIGHT('SQL Server', 6) RESULTS

Results
Server

SQL Sequence is available for SQL Server , Azure SQL db.

(3,5,8,9,19)….
(2,3,4,5)… (2,4,6…)

In SQL Server, A Sequence refers to a user-defined schema bound object which generates a sequence of numbers according to the specified specification. A Sequence in SQL Server contains numeric values which can be ascending or descending order at the defined interval & may cycle as if requested.

Create sequence [schema_name] .sequenceName
[AS integer_type]
[START WITH start_value]
[INCREMENT BY increment_value]
[{Minvalue[min_value]} | {No_Minvalue}]
[{Maxvalue[min_value]} | {No_MaxValue}]
[Cycle | {No_Cycle}]

-- A SQL server sequence is also should be unique in the current db

-- use any valid integer type for creating sequence e.g. tinyint, smallint, int, bigint or decimal and numeric with scale of 0

Start_value should be in the ranges of min_value and max_value

EOMONTH function - return the last day of the month of a specified date with optional offset.

EMONTH(start_date [offset]);

User-defined Functions

1. User-defined functions are routines function-body can accept parameters, they can perform actions, complex calculations, can return the results as value. These return values could be either be a single scalar set or result set

Benefits of user-defined functions

1. Customized functions with modular programming -- create the function only once, store it in the database, then use it any times. You can modify independently of the program source code.
2. Faster execution - tsql based user-defined functions (udfs) can reduce the compilation cost by caching the results in the plans and reuse them in repeated execution.
3. reduce network traffic - the function can be invoked in the where clause to reduce the number of rows sent to the client.

SQL Server UDF are of three types

- Scalar function - this function returns a single data value of the type defined for the RETURN clause.

- Table-valued function - returns the table data type.
- System function - String, System - CAST, CONVERT, ISNULL, Ranking (Row_number, Rank, Dense_Rank, Ntile) , Dynamic Management view (DMVs)

Features of Function Category

a) Deterministic function - returns the same result every time when they are called with a set of input values and same state of the db.
   Example: AVG() returns the same result for a specific input dataset

b) Non-deterministic function -- may return different values each time when they are called with a specific set of input values,
   Example:
   GETDATE() - returns a different value every time

   -- Limitations of SQL Server UDFs.

   -- SCHEMABINDING -- SQL Server UDFs can also be created with SCHEMABINDING, we wont be able to delete the function.
- Cant delete the function if there're computed columns are available in the function and indexing.

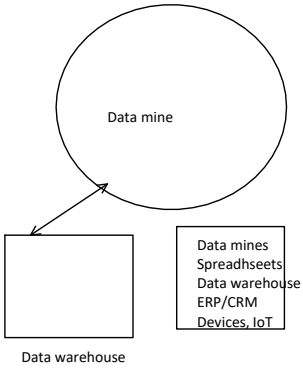
Data Warehouse Concepts

08 September 2022 18:46

1. Datawarehouse definition
2. What is Data Marts
3. What are the Data Lakes? Why should we design a Data Lake?
4. Examples of Data Warehouse..
5. Examples of Data Lake & Data Mart.
6. Data Warehouse Architecture
7. Tables design in datawarehouse
   - Star Schema design in SQL Server
   - Fact table
   - Dimension table

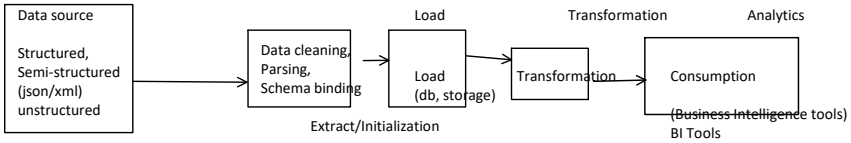
   Benefits of Data ware house
8. Defining the concepts on data modeling
   - Star Schema (demo)
   - Snowflake schema
9. Definition of data integration
10. OLAP (Online Analytical processing) , benefits, use case
11. Difference between OLAP (Data warehouse) and OLTP (sql database)
12. Introduction to SQL Server Analysis service (SSAS)
13. Data mining concepts , cube, dimension.

Tools :
SSMS (SQL Server engine)
SSAS - SSAS - SQL Server Analysis Service
Visual Studio (2008, 2012, 2015, 2017, 2019/2022)



1. Each section of data mines consists all sorts of product, product information, schemas, store information, product numbers
2. Banking/BFSI, Healthcare, Retail, Manufacturing
3. Data warehouse acts as central repository to get information from different sources and consolidates data through loading, processing and transforming.

Data Pipeline Architecture



1. PowerBI tool
2. SSRS (SQL Server Reporting Services)
3. Tableau
4. MSTR (Microstrategy)
5. QLIK

ERP (Enterprise Resource Planning)
SAP
CRM (Customer Relationship Management) - Dynamics 365, Salesforce CRM
IoT (internet of Things) weather forcasting, climate changing, sensor information in manufacturing

<weather>
<temp>31</temp>

Extract - Load - Transform (ELT) --
Extract - Transform - Load (ETL) -- PowerBI

Query
Customer+ Product

Requirement of Data Warehouse

1. Data ware house is built to overcome the limitation of database
   Databases are in (GB , MB) size max of a TB.
2. 500 - 600 TB, 1024 TB (1 PB)
3. Reporting, Analysis purpose we need Data warehouse
4. Business decision, analytical trends information has to be stored in Data warehouse

Examples of applications of Data Warehousing :

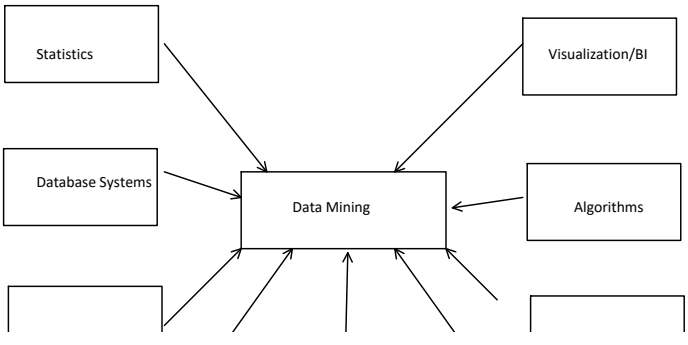
1. Social Media websites : analysing the large datasets, stored in a central repository. Data warehouse
2. BFSI - large datasets are stored in the central repo. That is Data warehouse
3. Retail - product recommendation based on large datasets stored in data ware houses. The analysis is performed on this data warehouse datasets to provide this kind of real time recommendation.

ETL and ELT are different. Depends on from scenario to scenario of use cases,

1. After data extraction and cleaning the data can be loaded into database and storage before transformation (ELT)
2. After data extraction and cleaning, when the data is transformed (querying, aggregation, stored procedures) then finally the data is loaded and moved into data warehouse (central repository of the company) (ETL) , BI dashboards can developed from taking data from the warehouse or analytics later.

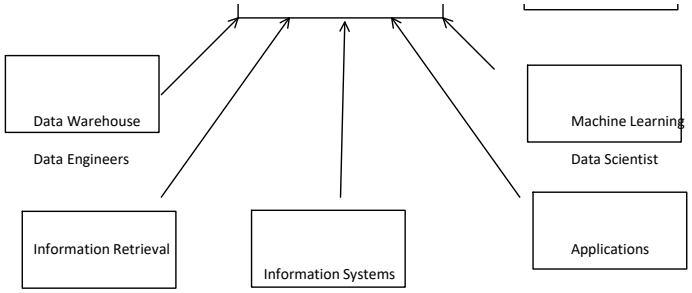
ELT + ETL

ETL = Extract, Transform and Load (data is loaded and stored in data warehouse) then moved to BI.
ELT = Extract, Load, Transform (data is transformed and stored in the database, data warehouse or data storage) then move to BI



1. Decision making systems
2. Recommendation systems
3. Predict models on hidden patterns
4. Predict the future trends
5. Informed Business Decisions

1. Data Warehouse
2. Transactional database
3. Time series db
4. Social media websites

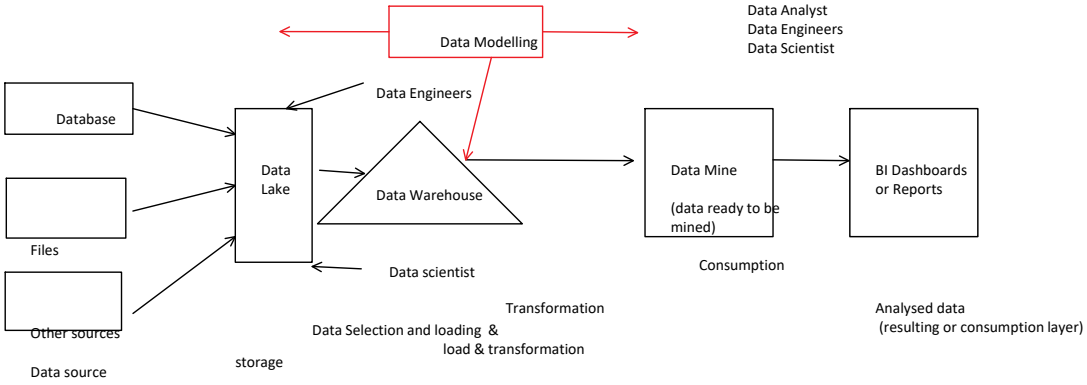


1. Data Warehouse
2. Transactional database
3. Time series db
4. Social media websites

Age	Region	Previous Purchased Product	Score	Recommendation
		Medcines	8	Cloths
		Book reader	6	Books, Digital app

50- 60 Rural
40-50 Urban

Data Analyst
Data Engineers
Data Scientist



1. Data - preprocessing - cleaning, integration, selection
2. Transformation
3. Data Mining
4. Data evaluation, visualization & presentation

1. IBM DB2
2. AWS Redshift
3. Azure SQL Data warehouse
4. Snowflake
5. Oracle Exadata

a) Volume (TB -> PB)
b) Velocity
c) Variety (structured, semi-structured, unstructured)
d) Veracity (formats, csv/tsv, .tar.gz)

Data Marts:

Data Warehouse is divided into smaller subsections as per business function, purpose and usage.

Customer -> Data Warehouse

1. Customer Inventory ---> Data Marts
2. Customer Demographics --> Data Marts
3. Customer Order -- > Data Marts.

a) Data Marts are easy to create
b) Less complex
c) Accelerate the business process

Data Lake:

Repository which stores all type of data whether structured, unstructured or semi-structured with any volume.

Principles

a) Volume (TB -> PB)
b) Velocity
c) Variety (structured, semi-structured, unstructured)
d) Veracity (formats, csv/tsv, .tar.gz)

1. Data Warehouse is for Analytics purpose
2. Data Lake is for data storage purpose

a) Batch -- SQL db
b) Real-time -- IoT devices, weather data

Examples:

1. Azure Data Lake
2. AWS Data Lake with S3
3. Informatica
4. Snowflake
5. Teradata
6. SAS

Data Modelling

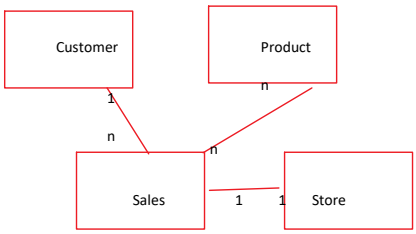
It's a process of creating a visual representation of either a while information system or it could involve collection and creating data visuals from different data sources/segments.

--- Data Models are built around the business needs.
-- Data can be modeled at various levels of abstraction
-- Data Modelling encompasses the standarized schema and formal techniques to manage the data resources in accordance to consistent and, predictive manner.

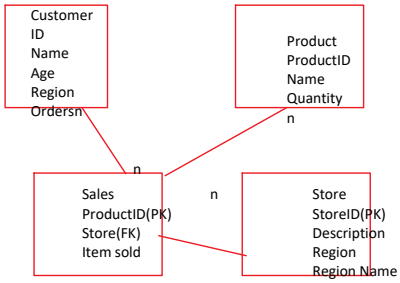
Types of Data Models

a) Conceptual Data Model
b) Physical Data Model
c) Logical Data Model

a) Conceptual Data Model - As per the domains identified - Banking, Retail.



b) Logical Data Model - More clear visuals and based on attributes the relationships has to be derived.



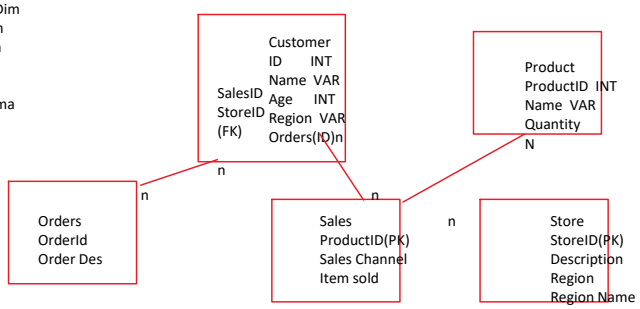
-- Physical Data Modelling -- They provide a schema for how the data will be physically stored within a database. They can offer a finalized design which can be implemented on relational db.

Kinds of tables

Key table is the Fact table
Smaller relationships maintained based on business contexts defined in Dimension tables.

Customer_Fact
Product_Dim
Store_Dim
Sales_Dim

Star Schema



1. Identify the entities
2. Identify the key properties or attributes of the entities
3. Get relationships between the entities and attributes

-- Erwin
-- ER studio

-- Reduce the errors in analytics pipeline
-- improve the app and data pipeline performance

-- ease of data mapping throughout the org
-- improve the communication between BI and developers.

Demo

Building a Star Schema in SQL Server db through SSMS

No	Dimension
1	**Product Dimension**
2	**Store Dimension**
3	**Order Dimension**
4	**Date Dimension**
5	**Territory Dimension**

1.	Fact table	Granularity: by each sales order	
	Fact_Sales	1) Sale subtotal 2) Tax amount 3) Shipping cost	
		Granularity: by each product on very order 1) Quantity ordered / product 2) Product subtotal 3) Discount	

Benefits of Star Schema

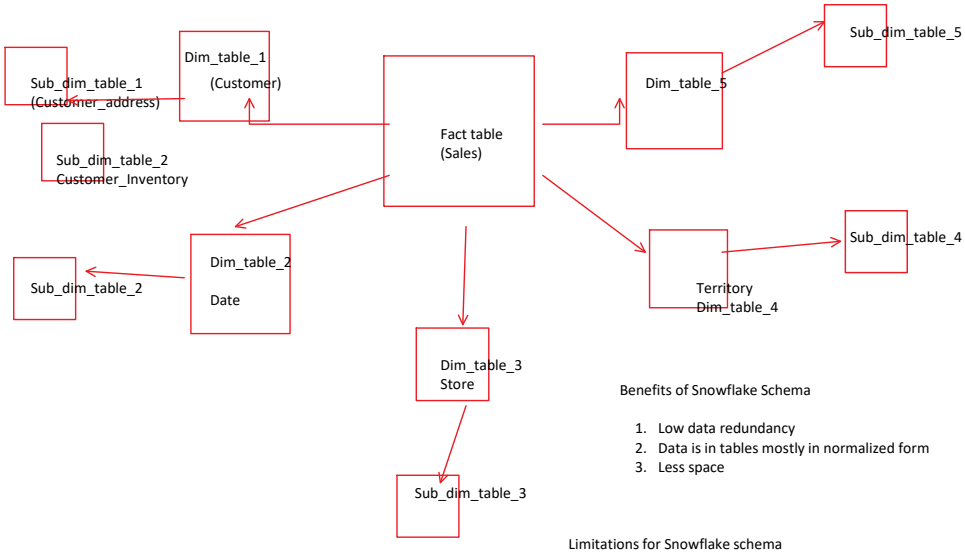
1. Takes less time for the query execution
2. Design is very simple
3. Query complexity is low.

Constraints of Star Schema

1. Takes more space in terms of storage
2. Data is highly redundant since no normalization is done

Snowflake Schema

1. It consists of fact and dimension tables
2. The dimension tables as well as sub-dimension tables are contained.



Benefits of Snowflake Schema

1. Low data redundancy
2. Data is in tables mostly in normalized form
3. Less space

Limitations for Snowflake schema

1. Design is very complex
2. Query complexity is higher than star schema
3. More number of foreign keys
4. Foreign keys are more

Facts Table

-- Features:

1. The Fact table contains the measuring of the attributes of a dimension table.
2. In the fact table, there're more number of records than dimension table.
3. Fact table forms a vertical table.
4. The attribute format of fact table is in numeric and text format.
5. Comes after the dimension table
6. The number of fact table is less than the dimension table in a schema.
7. It is used for analysis purpose and decision making.

Dimension table

-- Features

1. While in Dimension table, there're more number of attributes compared to fact table.
2. Dimension table forms the horizontal table.
3. The attribute format for dimension table is mostly text/varchar format
4. It comes before the fact table.
5. The number of dimension tables are more than fact table in a schema.

Measures:

Measures are the set of aggregates which we want to calculate such as sum of orders or the amount of total sales in a region etc.

A Dimension table stores the data required to define dimensions, dimension attributes and fact table is used to define the measures.

Types of OLAP

1. Relational OLAP (ROLAP) - Star Schema based - data is stored in relational database. Data can be stored multidimensionally on order to view multidimensionally.
   Adding WHERE clause in SQL statement

Star schema

2. Multidimentional OLAP - stores the data on disk in a specialized array structure. OLAP is performed on relying to the multidimensional capacity of arrays.

Here the multidimensional array is stored in a linear collection according to the nested traversal if the axis in pre-determined order.

SQL Server Cube. -- boundary to the measures

3. Hybrid OLAP - mixes the features of both relational OLAP and multidimensional OLAP. Allows to store huge volume of data with greater scalability than Relational OLAP.

Benefits: has the faster performance due to facilities of SQL server Analysis service Cube & stored the detailed data, Performance wise much better. Databases are stored in most functional way.

4. Web based OLAP - used for web applications
5. Desktop based OLAP - used for desktop analytical processing
6. Mobile OLAP - used for Mobile apps & analytics

SELECT Case expression
When expression1 then result1
When expression2 then result2
When expression3 then result3
Else result

A	B	C
20	20	23
20	20	20
20	21	22
13	14	30

End

1. Isoscales -

Azure Cloud Fundamentals

08 September 2022 18:45

1. Benefits of Cloud Computing

a) Cloud is easy to access for everyone. All the resources like Servers, disks, network, IP address, storage, databases, data warehouses all are accessible to everyone globally through internet.
b) Developing new application and services, storage, databases on cloud
c) It is useful to migrate the existing applications, databases, storage to the cloud as per supportability of the cloud vendor
d) Software on demand
e) Analysis of data
f) Streaming of audio, video, media

Cloud migration --
Capex -> Opex (Operational expenditure)
(capital expenditure model)

a) Scalability - increase the size or number of instances for the servers, databases or storage for the resources on cloud.

Use Case 1

-- horizontal scalability (scale out) -- recommended to go for horizontal scalability for long term business.
-- vertical scalability (scale in) --

b) SLA -- Service Level Agreement -- 99.9% of uptime (managed services) a downtime of 0.001 sec / week/month.
Azure VMs a SLA of 99.99% of uptime -- (a downtime of nanosec/less than millisecond per month)

Associated with each and every cloud resource for any cloud provider.

Cloud provider can compensate for any downtime for missed SLA/ guaranteed uptime.

This is part of cloud based business model for the cloud providers to their customers.

Private Cloud

Features

a) Private cloud is also known the internal cloud or corporate cloud.
b) Private cloud provides a high level of security data is accessible over the internal network only to limited set of users based on accessibility.

e.g. HP data centers, Ubuntu cloud, Azure Stack (Microsoft private cloud), Anthos,
Elastra -private cloud, canonical private cloud, Vmware private cloud

Advantages of private cloud -

a) More control - more control over the resources and hardware rather than public cloud, because it's only available To the selected users.

b) Security & privacy - private cloud has more improved security as compared to public cloud.
c) Improved performance - gives better performance with improved speed and capacity.

Cons:

a) High cost - the cost is higher since the resource set up and hardware resources, software, apps, network, storage/db are all managed internally for a particular org level
b) Restricted area of operations - private cloud is accessible only to a particular org, within the org boundary, Hence the operations are limited.
c) Limited scalability - private cloud can be scaled only with the capacity of the internal hosted resourced within the Org boundary.

Horizontal scalability -

Increase the number of server instances based on requirement -- scale out
Decrease the number of server instances based on the requirement -- scale in

Hybrid Cloud

Advantages

1. Flexible and secure - it provides flexibility of mix of public cloud and secure resources because of private cloud
2. Cost effective - hybrid cloud costs less than the private cloud. It helps the organization to save costs for both of the infra and application support
3. Adaptable - A hybrid cloud is capable of adopting to the increasing demands as per company requirement for disk storage, memory, and application infrastructure.

Constraints:

1. Networking - on-prem network has to connect to the public cloud network, lot of complex network config and settings required.
2. Reliability - depends on the cloud service provider
3. Infra compatibility - with the dual level of infra , a private cloud + public cloud model there is a chance that they are running in different data centers.

	Cloud Model	Benefits	Example
IaaS	Rent for respective infra required for business - Compute-	Less complex Less development cycle	AWS EC2 AWS VPC AWS Subnet AWS EBS

	- Storage/disk - Network/ip - Memory is only managed by the cloud provider & their security Not managed - -- OS patching -- OS image -- application security		Azure VM Azure Network Azure Disk
PaaS	Managed services (compute, storage, database, data warehouse, analytics services) - Compute - Storage/disk - Memory - Network - OS - OS patching - Security for infra managed by cloud providers	Less burden on infra provisioning Developers can focus more into their app/business logic Language agnostic (.net, java/jsp, springboot, react/angular, golang, ruby, python/django)	AWS EBS (Elastic beanstalk), Azure App service (Web apps, Mobile apps, API apps), Azure SQL db, Azure Hadoop services, Azure SQL DW, Azure Analysis services, Azure Data lake services
SaaS	Managed services (compute, storage, database, data warehouse, analytics services) - Compute - Storage/disk - Memory - Network - OS - OS patching - Security for infra managed by cloud providers - Application - Application security All of these will be provided by the cloud provider	Almost zero burden on the end-user in terms of app dev and deployment No burden on app scalability, reliability, resiliency , app backup, DR	Office 365 Salesforce
Serverless	The apps are hosted into this serverless resources where you cant access those servers	Zero downtime Almost 100% scalability Programming Language agnostic	Azure Function AWS Lambda

Use Case 2

Web Server 1
data base server

Memory - 32 gb -> 128 GB
Disk - 5 tb -> 32 TB

30-40 raised request for loan through the app,
Increasing the compute capacity for one specific resource, it's called vertical scalability

Azure Fundamentals

08 September 2022 18:46

Snowflake

1. Create SnowFlake db and Schema

```
create or replace database Retail;

2.
select current_database(),
current_schema();
```

3. Create SnowFlake Datawarehouse

```
create or replace warehouse retail
with
  warehouse_size='X-SMALL'
  auto_suspend = 180
  auto_resume = true
  initially_suspended=true;

4.Create table

create or replace table emp_basic (
  first_name string ,
  last_name string ,
  email string ,
  streetaddress string ,
  city string ,
  start_date date
  );
```

1. Azure Fundamentals
2. Azure Global Infrastructure -> Azure Regions, Data center
3. Azure Resource Manager
4. Azure Resource Group
5. Azure Web App
6. Azure Storage
7. Azure Virtual Network
8. Azure Virtual Machine
9. Azure SQL db

Scenario 1: IaaS (Infra as a Service)

Web/App Scenario

SharePoint App + includes the data from
3rd-party API (Mulesoft)

Database Sc enario

SQL Server db + Includes depend ency
like .net objects, CLR stored procedures
SQL Server on Azure VM

Azure Virtual Machine
Network (IP CIDR)
Public IP address

- Service deployment **Shared Responsibility Model**
-
- Taken care by the Cloud Provider

a) CPU
b) Memory
c) Disk storage
d) Network
e) Security

- Not Taken care by Cloud provider / have to take care by Customer
- Underlying OS
- OS patching and update
- Application deployment
- Security of the Apps and data

Scenario 2: PaaS (Platform as a Service)

Web / App Scenario +

A sample .net/java application includes 3-tier architecture, it includes lot of
.js, html, css files in front-end and lot of worker process(windows batch) process in its business logic

The app deployment on Azure should be cost-effective.

Azure Web app

Shared Responsibility

Managed by cloud vendor

- CPU
- Memory
- Disk
- Network
- Security of Infra

Database

SQL Server db

200-400 GB db size
Scalability -
High availability
Zero downtime

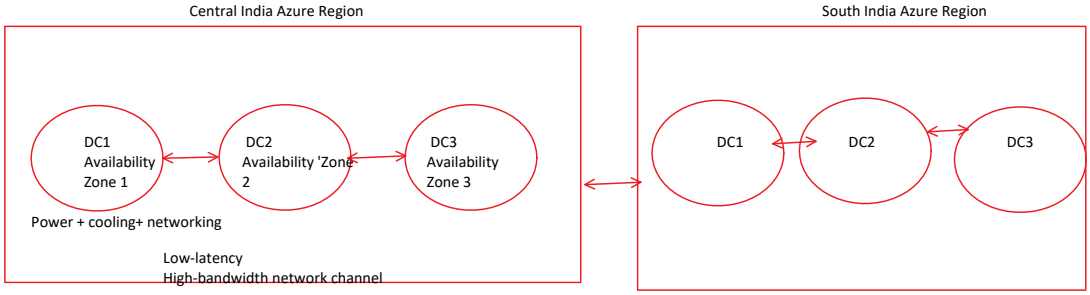
Read Replicas are also required
Azure SQL server + SQL db (Region 1) (primary)
Azure SQL Server + SQL db (Region 2) (Secondary)
(read replica)

Azure SQL database

- Guest OS
- Guest OS update/Patching
- Deployment tools

Not managed by cloud vendor

- Application code
- App security
- Database coding (SQL queries, Stored procedure, Views.. Functions)
- Data security



LRS - 3 copies of storage account in the region (dev/demo)

GRS - Geo-redundant storage (3 + 3) = 6 copies of storage accounts are created across primary and secondary region

India based Azure Regions

Azure Geography - (Market basis it includes one or more regions)

-- Allows the customer the specific data residency
-- manage compliance as per app and data needs to close
-- Geographies are fault tolerant to withstand/prevent the complete regional failure through Dedicated high bandwidth networks.

-- GDPR (compliance)

Azure Availability Zones

-- Availability Zones (AZ) are unique physical buildings within Azure region to protect apps and data from facility level issues.
-- it offers high availability to protect your application + data from the datacenter failure
-- Each Zone is comprised of one or more data centers containing independent power, cooling and networking.

Azure Global Network

Refers to all the components in networking & is part of Microsoft global WAN (wide area network), fibers, cables, routers, switches, PoPs, etc.

Azure Resource Manager

-- The request for creation, updating or deletion of resources On Azure is handled through Azure Resource Manager.
-- Then, Azure resource manager handles the request by authentication and authorizing the request before forwarding It to the specific service

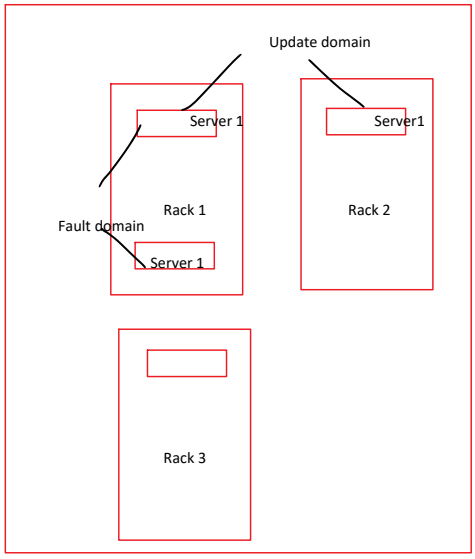
It consists of the following components

a) Resources - A manageable item in Azure
   - Azure VM
   - Azure Database
   - Azure Storage
   - Azure Network
   - Azure App services
   - Subscriptions

b) Resource group - A logical container which holds all of the resources together.
   - Resources to be part of a Resource group
     - VM
     - Storage
     - Web App

Resource Group

- You can decide which resource is to be part of which resource group
- The azure resource to be deployed in the same region of the resource group

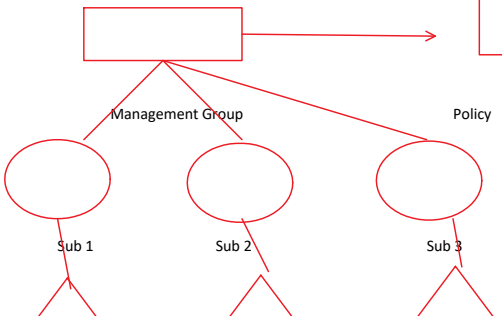


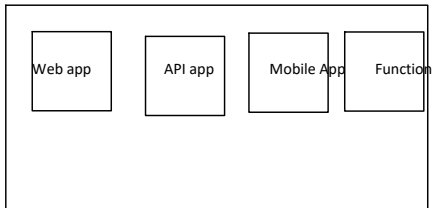
c) Resource Provider - A service which supplies Azure resources. As End user/customer, we don't have to create/configure anything.

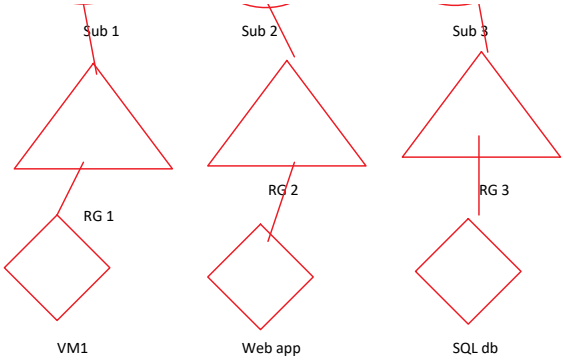
Microsoft.Compute - VM related resources
Microsoft.Web - web app related resources
Microsoft.Storage - includes all storage related resources (blobs, tables, queues, files)

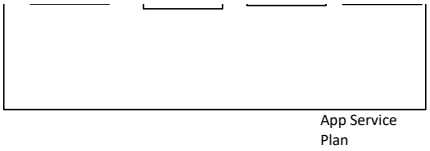
Benefits of Resource Manager

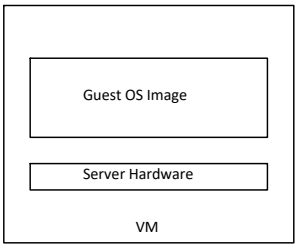
-- Manage your infrastructure as a unit as a template rather than a script
-- deploy, manage and monitor the resource as a group
-- redeploy the solution throughout the development lifecycle (SDLC)
-- Apply tags to your resources and resource groups based on organization requirements (uniquely identify the resources in the container and also it's used for billing purpose as a org unit)

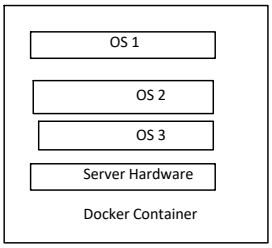












Virtual Network

Fundamental building block for the private network in Azure.

It helps to create many resources accessible over specified network

a) Azure VM
b) Azure Web App
c) Azure Storage
d) Azure SQL db

Vnet is also similar to a traditional network that you can operate in your own datacenter.

Benefits of Azure Vnet

a) Enables secure connectivity to Azure resources from on-prem to this resource through internet
b) Vnet can also help traffic filtering, routing network traffic and integration with other Azure services.

.dockerimage file

Run webserver
Port 8080
Config
Container url:

Pre-requisites	Docker Desktop (Windows/Linux) VS code / Visual Studio
Profile	Create a profile in Docker hub
publish	Publish your local customized images to docker

Basic PowerShell Scripting

08 September 2022 18:46

PowerShell Desired State Extension

1. Configuration management

Example: deployment requires some properties for the server (size, network, storage) and the config of the OS.

2. Compliance

Example: you want to audit or deploy settings to all machines in scope either reactively to existing machines or proactively to new machines as they are deployed.

"metadata": {
"category": "Guest Configuration"
"guestConfiguration:{
"name:"AzureWindowsbaseline"
"version" "1.*"

Difference between Cmdlet and Command

1. Cmdlets are .net framework class objects & not just standalone executables.
2. Cmdlets can be easily constructed from a few lines of code
3. In Cmdlets, the parsing, error representation, output formatting are not handled by cmdlets. It's done by Windows PowerShell Runtime.
4. Cmdlets are the process which works on objects not on text stream. Objects can be passed as output for pipelining.
5. Cmdlets are record based as they process a single object at a time.

For(I= initialization; i<= length -1 ; i++)

A regular expression is special sequence of characters that can help to match or find other strings or set of strings using a specialized pattern.

They can help to search, edit, manipulate text and data.

Subexpression	Matches
^	Matches the beginning of the line
$	Matches the end of line
\A	Beginning of entire string
\z	Ending of entire string
\Z	End of the entire string except the allowable final line terminator
[…]	Matches any single character in brackets
[^..]	Matches any single character not in brackets
\w	Matches the word characters
\W	Matches the nonword characters
\s	Matches the whitespace
\S	Matches the nonwhitespace
\d	Matches the digits
\D	Matches the nondigits
\G	Matches the point where the last match finished
\n, \t	Matches newlines, carriage returns, tabs etc.

Providers in PowerShell can provide access to data and the components
That couldn't be accessible easily from the command line.

The data is represented in a consistent format through providers.

Providers are .NET programs which provides access to specialized data stores
For easy viewing and management.

- Alias provider
- Drive : Alias

Certificate provider
- Drive: cert
- Objects: Microsoft.PowerShell.Commands.X509CertificateStore

Environment provider
- Drive: env
- System.Collections.DictionaryEntry

FileSystem provider
- Drive: C:

Objects: System.IO.FileInfo, System.IO.DirectoryInfo

- Variable provider
- Drive: variable
- Object: System.Management.Automation.PSVariable

Registry Provider
Drive: HKLM, HKCU
Objects: System.Win32.RegistryKey

Apache Hadoop Overview

20 September 2022 17:56

Constraints and Reasons for evolving Big Data in Software Development

1. Volume aspects - KB, GB, TB, PB, ZB (10to the power9-11) , traditional RDBMS to Big Data pillar

e.g. Twitter feed data, (social sentiment analytics) -> 10 TB , weblog, clickstream

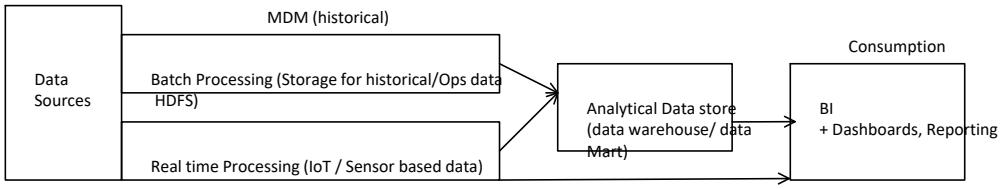
2. To get insights from data - to solve the real time analytical queries and answers as per the business requirements.
3. Data variety - RDBMS + weblog + clickstream + BI -> dashboards (semi-structured data, quasi-structured and unstructured data) , media (audio + video)
4. Velocity - IoT, sensors (latitude, longitude, temperature) , fast rate the data should be processed and transformed. It is defined as the rate in which data is received and acted on. All of data in traditional RDBMS is stored on disk where as the real time data streams can be directly processed in memory instead of disk.

Volume

Velocity

Variety

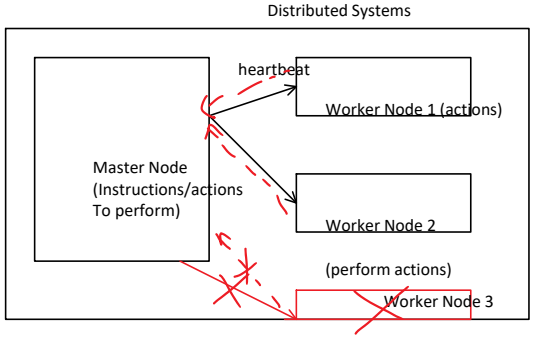
Velacity -- measurement of different aspect of data, twitter feed data, clickstream , weblog



Azure - Hadoop(HDFS), Event Hub (real time streaming engine for processing)

(Social Media,
Website(Clickstream, weblog)
Weather data from Sensors

Big Data Architecture - Pipeline Ver 1

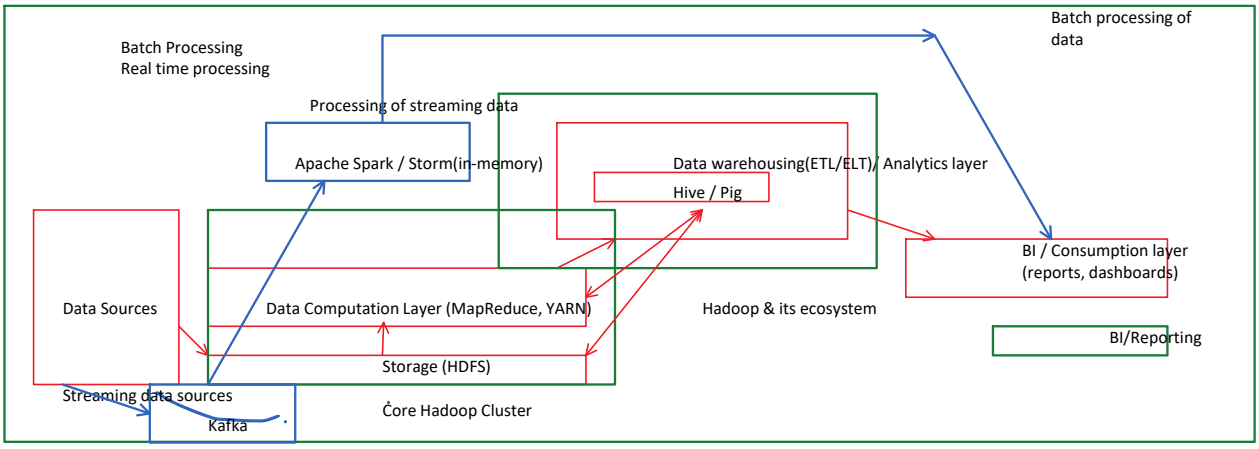


Apache Hadoop

a) Hadoop Distributed File System (HDFS) -> Storage layer
b) MapReduce -> (Computation layer)
c) YARN -> (Yet another Resource negotiator) (MapReduce V2)

Rest of Apache Hadoop ecosystem components

a) Apache Hive - Data warehouse and analytics tool in Hadoop
b) Apache Pig - data querying tool with scripting language (Load, Transform & DUMP)
c) Kafka - Real time stream processing engine



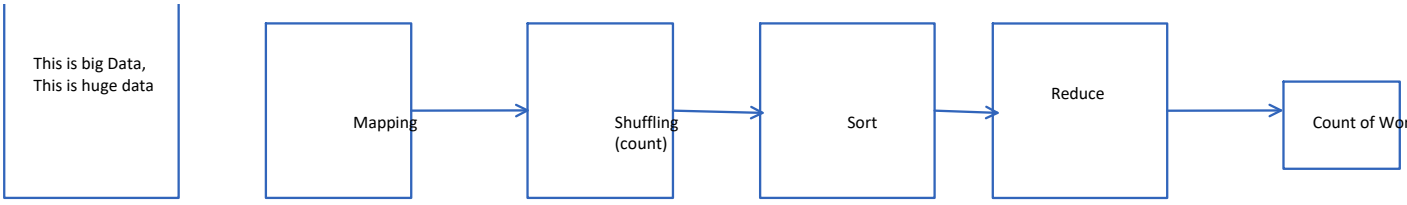
Data ingestion, extraction, initial processing

Batch Processing = Hadoop & its ecosystem (HDFS, Mapreduce, Pig and Hive)
Real time processing = Kafka + Spark

Big Data Pipeline - Ver 2

Apache Spark
Apache tez

Count the number of words



500 - 1 PB

Mapper jobs

Job_1 = big, bigger, not big
Job_2 = data
Job_3 = is

Mapper class

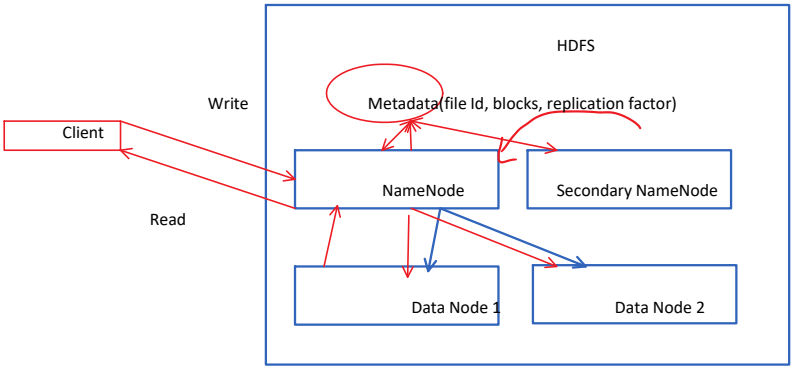
Big = 100
Not big = 300
Bigger = 500
Is = 300
This = 500

(bubble sort, quick sort)

Big & not big, Bigger = 900
Is = 1000
This = 2000

Big Data Use Cases

1. Product Development - Social media
2. Predictive maintenance - Aerodynamics,
3. Customer experience - Telcom, Banking
4. Fraud and Compliance - Banking, Financial
5. Drive innovation - BFSI, Healthcare, marketing
6. Operational efficiency -



Easy to debug and readable
HDFS is the default storage for Hadoop ecosystem components

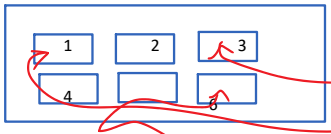
Namenode - (Master node)
a)check the heartbeats from the datanodes
b) Sends the instructions to the datanode

Secondary Namenode- (Master node)
a) Works as a replication for primary namenode
b) In case of primary namenode failure, the secondary namenode Becomes the primary node
c) Once recovered from failure, the old primary namenode becomes the new New Secondary namenode.

Data storage principles

a) In HDFS, data is stored in blocks which are called as data blocks.
b) Data blocks are small as much as individual granular block with max size of 126 MB (e.g. Hadoop v2)

Datanode - (Worker/slave node)
a) Performing all kinds of data storage in the storage blocks
b) Performing actions for data storage as per namenode



Data is appended as per blocks

This is big data
This is good data 10*1024*1024 / 128 = 81920 blocks in HDFS datanode

Computation should take place in the location of data

Neither data should travel to the location of compute.

Apache Pig

a) It can handle structured, semi-structured or unstructured data & stores the data into the HDFS.
b) Every task or query running on pig is executed through MapReduce.

Benefits:

a) Ease of programming:
b) Optimization of complex data processing (Load, transform, DUMP)

c) Flexible and with in-built operators (sort, filter, join, foreach)
d) Supported for Avro based file format (row wise)

MapReduce	Apache Pig
It is a low level data processing	It is a high level data flow tool
Complex programs through java/python	Simple pig latin which are simpler also less in no of lines
Data operations are difficult compared to pig	Pig latin these built-in operators are available
It does not allow nested data type	It provides the nested data types like tuple, bag, and map

Apache Hive

Built on top of Apache hadoop, it provides the following features

a) Tools to enable easy access to data via SQL. SQL like query interface which is called HQL.
b) Enables us to perform data warehouse tasks such (extract, transform, load) (ETL) operations, analysis & reporting.
c) Query execution is done through MapReduce, Apache Tez (framework for faster data processing in Pig and hive), Apache Spark.
d) Procedural language with HPL-SQL
e) Sub-second based query retrieval support when using with Hive LLAP, Apache YARN.
f) Hive supports different file formats with lot of built-in connectors like (CSV/TSV), Apache Parquet, Apache ORC (columnar file)
g) Traditional data warehouse workloads.

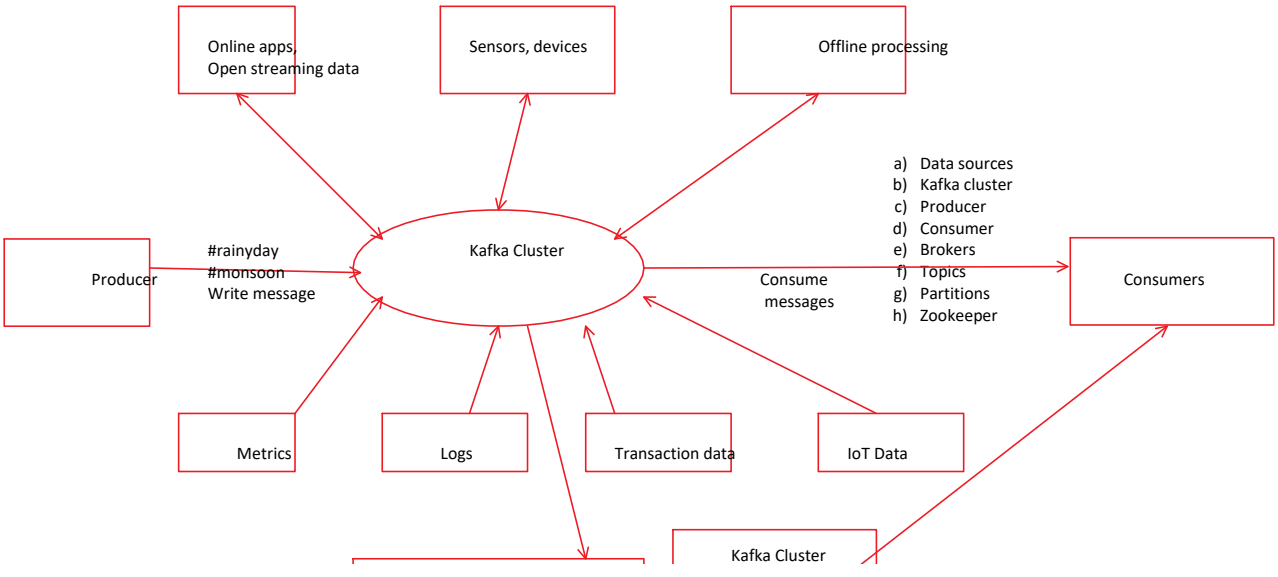
Apache Pig	Apache Hive
Pig operates on the client side	Hive operates on the server side of the cluster
Pig uses the pig-latin script for analysis	Hive uses hiveql (hql) language
Pig latin is purely procedural language	Hive is declarative SQL support
Pig has support for Avro file format	Hive has support for Parquet & ORC
Pig is suitable for complex and nested data structure	Hive is suitable for batch processing OLAP systems
Pig does not support schema to store data	Hive supports schema which can used for data insertion into table
Pig does not have support for JDBC/ODBC	Hive has the support for JDBC & ODBC
Pig does not have any metastore (metadata store)	Hive has metastore with support for various DDL language (SQL, mySQL, postgreSQL)
Pig operates quickly & data loads quickly	Hive loads data slowly
.pig is extension for pig-latin scripts	Any file formats (.hql) file format

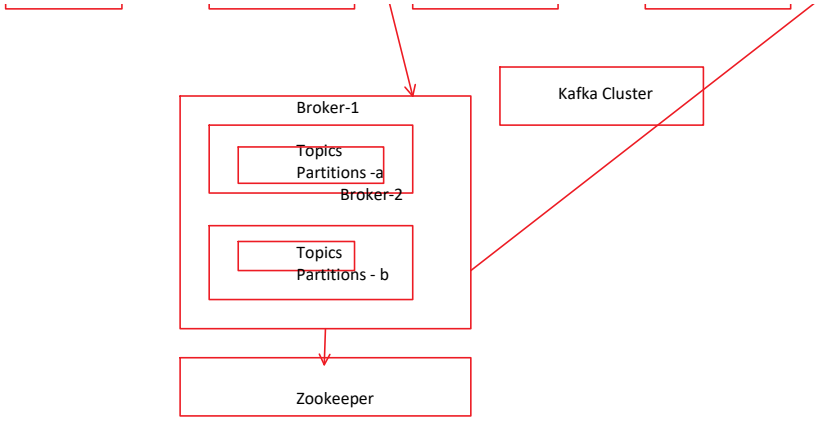
Azure HDInsight is 100% core Apache Hadoop platform

Hadoop ecosystem Tools Supports

- HDFS
- MapReduce
- YARN
- Pig
- Hive
- Kafka
- Spark
- Storm
- Sqoop

Apache Kafka works as a real time streaming engine which helps in the ingestion of data. It provides a fault-tolerance, distributed real time streaming platform where the producers can produce the data from the various data sources, then get processed to get it consumed by the consumers.




Kafka Cluster
Broker-1
Topics
Partitions -a
Broker-2
Topics
Partitions - b
Zookeeper

Apache Hadoop (Deep Dive)

08 September 2022 18:46

Reason behind choosing different file formats support in HDFS

1. Faster read time
2. Faster write time
3. Splittable files
4. Schema evolution support (modify the data fields)
5. Advanced compression support (gzip, LDAP, snappy)
6. Snappy data compression can lead to high speed and reasonable less data latency travelling over the network
7. File formats can help to manage the diverse data set.

Why Data serialization for storage formats?

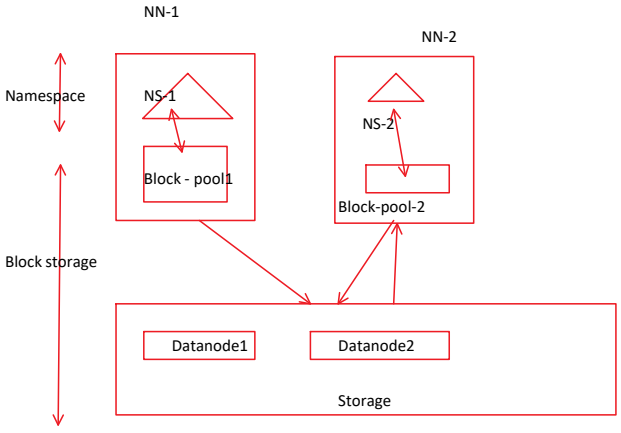
1. To process the dataset faster
2. Whenever proper data formats are required to maintain and transmit the data over the network without schema support on another end.
3. Data without structure or format whenever requires to process, complex errors can occur, data deserialization can help through metadata.
4. Serialization can help data validation over transmission.
   Benefits

1. Compact
2. Fast
3. Extensible and interoperable
4.

File Formats	Benefits (pros/cons)
CSV file format	Requires compression support, using CSV in HDFS can increase the reading performance cost, schema support is also within the limit.
JSON	Better performance than CSV, records are retrieved even faster
AVRO	Row oriented file format and data serialization framework
Sequence	Complex reading operations
RC	(Row-Columnar) file format where write operations are comparatively slower, optimized RC (ORC) they can have better support , read operation is average
Parquet	As a columnar file format, optimized for storing data, processing and analysis

HDFS Namenode block management

1. Provides the datanode level cluster relationship by handling individual registrations, periodic heart beats.
2. Processes various data blocks and maintains the location of these data blocks within the datanode
3. Supports block related operations such as CRUD ops, of the blocks
4. Manages the data blocks replication, block level replication for under replicated blocks and allowing core read \write operations.



Block-pools are a set of blocks responsible to belong to a single namespace, whereas the Datanodes store blocks for all the block pools in the cluster. Each and every block pool is Managed independently, allows the ns to generate the blockids for new blocks to co - ordinate with other ns.

A ns and block pool together known as namespace volume.

ClusterID - uniquely identify the nodes in the cluster. When a namenode gets formatted, the identifier is either provided or it can be auto generated. This cluster ID is used for formatting other namenodes in the cluster.

Key benefits for HDFS federation

1. Namespace and namenode scalability - federation adds the namespace horizontal scalability. Large deployments or deployments using a lot of small files can be benefitted by namespace scalling allowing more namenodes.
2. Performance - file system throughput is not limited by a single namenode, adding more nn can scales the cluster for better throughput and read/write ops
3. Isolation - multiple namenodes can help to manage different categories of apps and users can be isolated to different namespaces.

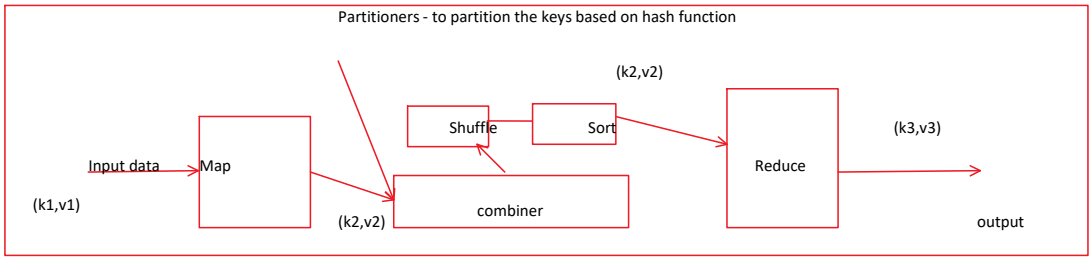
ALT + CNTL = coming to primary
CNTL + G = insert to VM

1. ResourceManager -- Master node
2. NodeManager -- Worker node
3. MRAppMaster - per application

Mapreduce Applications need to specify input/output locations and map and reduce functions via implementations of appropriate interfaces and/or abstract classes. There're other job parameters, defined as job configuration.

Mapreduce as a framework consists of <key, value> pair model. The framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job.

- Key, value pairs has to be serializable by the framework and need to implement Writable interface. The Key classes have to implement the WritableComparable interface to facilitate the sorting by the framework.
- (input) <k1, v1> -> map -> <k2, v2) -> combine -> <k2,v2) -> reduce -> (k3, v3) -> output
- Combiner to save the bandwidth as much as possible by minimizing the number of key/value pairs which will be shuffled across the network and to be provided as an input to the reducer.



Partitioner - paritioning of keys coming from the intermediate map output and is being controlled by Partitioner. The partitioner is used to derive the partition.
On the basis of key-value pair, each map output gets partitioned.
Default partitioner (Hash partitioner) computes a hash value for the key and assigns the partition based on the result.

In a MR job, the mapper executes first , then combiner then partitioner

Mapreduce has its own data type called Writable.

This Writable data type implements the WritableComparable interface.
The WritableComparable interface is a combination of Writable and ComparableInterface.
The Writable data types works with the following data types:

Integer -> IntWritable = it is the hadoop variant of integer. It is used to pass the integer nos and key or values.
Float -> floatWritable = used to pass the floating point numbers as key or values
Long-> LongWritable = used to pass hadoop variant of long data type as key/values
Double->DoubleWritable = pass the double to store double values
String -> Text = hadoop variant of string to pass characters as key/value
Byte-> ByteWritable= hadoop variant of byte to store sequence of bytes
Null-> NullWritable-> hadoop variant of null to pass null as key/value.

The Comparable interface is used for comparing when the reducer sorts the keys, and Writable can write the results back to local disk.

Hive Internal Table	Hive External Table
Managed table where the data gets loaded directly from your local drive	Hive external table is more efficient where it follows the loose coupling and the data gets loaded from HDFS
Entire lifecycle of hive table is managed by hive itself, DDL, DML operations and underlying datasets	Data gets loaded into the table from HDFS
Once the internal table is dropped, the underlying table's data gets deleted	Once the external table is dropped, the underlying table's data doesn't get deleted, because the data is stored in HDFS
Create table 'tablename'	Create external table 'tablename'
Data Path has to be specified from local drive	Data path has to be specified by 'HDFS path'

Partitioning in Hive - Apache Hive organizes the tables into partitions for grouping same type of data together based on a column or partition key.
- Each table in hive can have one or more partition keys to identify a particular partition.
- Using partition, it makes faster to the querying of the data or slicing of the data.

E.g. using the stu_dept column is a partition column / partition key.

Pros & Cons

Pros -
1. It helps to distribute the execution load horizontally
2. In partition, faster execution of queries can happen with low volumes of data.

Cons -
1. There's a possibility that, too many small partitions can be created, -- too many small directories
2. Partitioning can be effective low volume data (GB-TB), for huge volume of data (PB) level, it takes time to process the data through query

Bucketing - Once the partition is done, then the hive tables or partition can be further subdivided based on hash function of a column in the table to give an extra structure to the data for efficient querying purpose which can be called as bucketing

Problem: Datanode is not running

pre-requisite

stop all running hadoop services
stop-all.sh

Steps:

1. Remove the contents from datanode & directory

sudo rm -r /home/ani/hadoop/hdfs/namenode/datanode

2. format the namenode

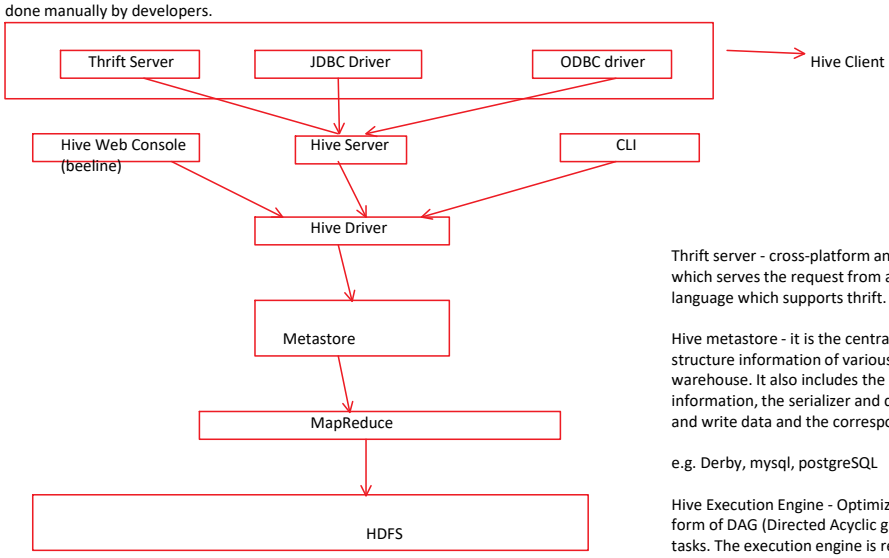
hdfs namenode -format

3. start-dfs.sh
start-yarn.sh

Pros -

- It provides faster query response like partitioning
- In bucketing due to equal volume of data in each partition, joins at the Map side will be quicker.

Cons -
We can define a number of buckets during the table creation. But loading of data into the buckets has to done manually by developers.



Thrift server - cross-platform and cross-language service provider which serves the request from all the supported programming language which supports thrift.

Hive metastore - it is the central repository which stores all the structure information of various tables and partitions in the warehouse. It also includes the metadata of the column and its type information, the serializer and deserailizers which are used to read and write data and the corresponding HDFS files where data is stored.

e.g. Derby, mysql, postgreSQL

Hive Execution Engine - Optimizer generates the logical plan in the form of DAG (Directed Acyclic graph) of map-reduce tasks and HDFS tasks. The execution engine is responsible for executing the incoming hiveql queries in the order being executed.

Datatypes in Pig

Type	Description	Example
Int	Signed 32 bit integer	4, 40, 300
Long	Signed 64 bit integer	15L
Float	32 bit floating point	2.5f, 5.5F
Double	32 bit floating point	1.5, 1.5e2
charArray	Character Array	Hello World
Tuple	Ordered set of fields	(12,43)
Bag	Collection of tuples	{(12,43), (55,100)}
Map	Collection of tuples using characters	[hello#world]

Operators in Pig

LOAD - LOAD is a relational operator, used to load the data from the file system

LOAD from 'hdfs://localhost:9000/pig_data' as (id:int,name:chararray) USING PigStorage;

DISTINCT Operator

Eliminate the duplicate tuples.

FILTER Operator

Is used to remove duplicate tuples in a relation, also sorts the given data and then eliminates the duplicates.

FOREACH Operator
Generate the data transformations based on columns of data. It is recommended to use foreach operator to work with tuples of data.

GROUP Operator

GROUP operator is used to group the data into one or more relations. It groups the tuples which contains a similar group key. If the group key has more than one field, it treats as tuple otherwise it will be the same type as that of the group key. Hence, GROUP operator provides a relation that contains one tuple per group.

ORDER BY Operator

Sorts a relation based on one or more fields. It also maintains the order of tuples.

LIMIT operator

Is used to limit the number of output tuples.

JOIN - to join two or more relations
COGROUP - to group the data into two or more relations
CROSS - to create cross product of two or more relations
UNION - to combine two or more relations into a single relation
SPLIT - to split a single relation into two or more relations.
DUMP- to print the contents of a relation in a console

Relation_name = LOAD '<input_file_path>' USING Function as schema;
DUMP relation_name;

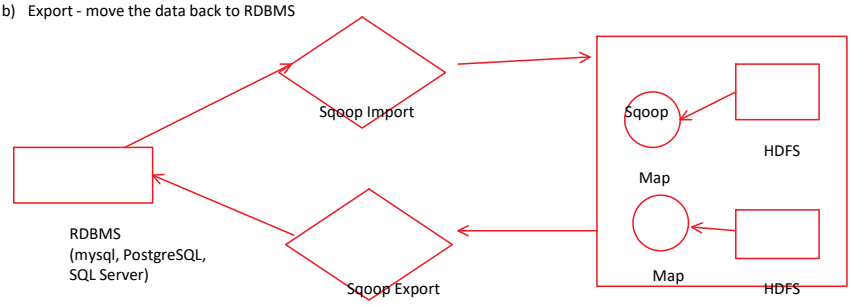
Relation_name = the relation in which we want to store the data
<input_file_path> = defines the hdfs data directory

Function - choose the set of functions from Apache Pig framework (PigStorage, TextLoader, JsonLoader, BinStorage)
Schema - define the schema of the data, (col1:data_type1,col2:data_type2….);

Apache Sqoop

1. Sqoop helps to connect the result from the SQL queries into the HDFS
2. Sqoop also helps us to load the processed data directly into the hive / Hbase
3. It performs the security operations of data with the help of Kerberos
4. With the help of Sqoop, we can perform compression of the processed data
5. Sqoop is also highly powerful and efficient in nature
6. Sqoop helps to perform ETL operations in a very fast and cost-effective manner
7. With the help of Sqoop, we can perform the parallel processing of the data which leads us to fasten the overall process.
8. Sqoop uses the MapReduce mechanism for its operations which also supports the fault tolerance

a) Import - importing data from RDBMS to hadoop cluster
b) Export - move the data back to RDBMS



File formats

a) Delimited texts (default format) (--as-textfile), non - binary data types
b) Sequence Files - binary format used for individual records in custom record-specific data types. (VARBINARY)
c) Avro data file format are compact, binary format
d) BLOB, CLOB data type (--mysql-delimiter) , --lines-terminated-by-<char>, --escaped-by<char>, --inline-job--limit

Conditional Import

1. Append
2. Lastmodified

--incremental argument can be specified to define the incremental import to perform

Append mode can be defined when importing a table where new rows are continously being added with increasing row_no values. We can check the column containing the row_no value with --check-column argument. Sqoop imports the rows where the check column has a value > one defined in the --last-value option

Flume Event

A flume event is a basic unit of data which needs to be transferred from source to destination

Flume Agent
Flume agent is an independent JVM in Apache Flume. Agent receives events from clients or other Flume agents and passes it to the next destination which can be a sink, or other agents.

Agent -> source, channel, sink

Source - Avro source, thrift source, twitter 1% source , Exec source

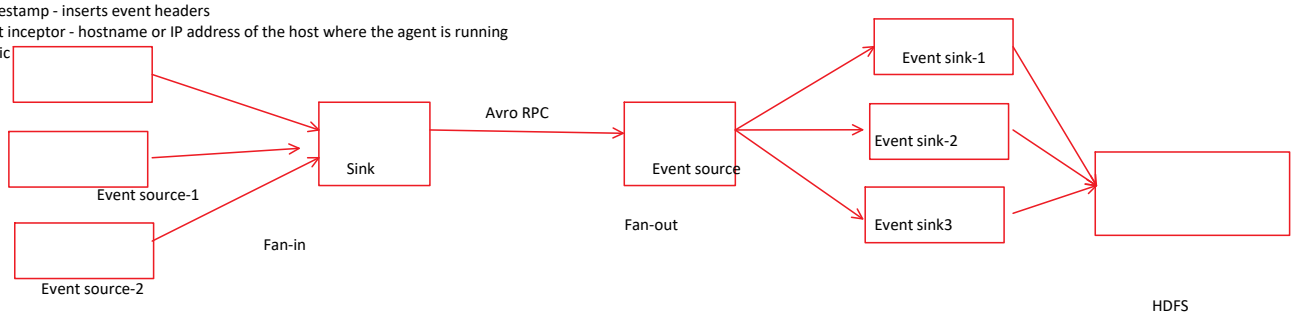
Channel - File system channel, JDBC channel or Memory channel

Sink - HDFS sink

Flume Inceptors

a) Channel selectors - which channel to be selected for transferring the data when multiple channels are existing.
   -> default
   ->Multiplexing
b) Sink Processors - invoke a particular sink from a group of sink

a) Timestamp - inserts event headers
b) Host inceptor - hostname or IP address of the host where the agent is running
c) Static

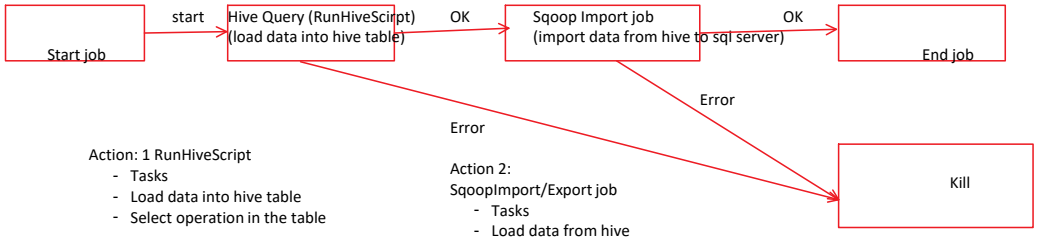


c) Static interceptor - append a static header with static value to all events, it's possible with the static flume interceptors.

d)

A1.sources.r1.interceptor.type = static
A1.sources.r1.interceptor.i1.key= datacenter

Oozie workflow



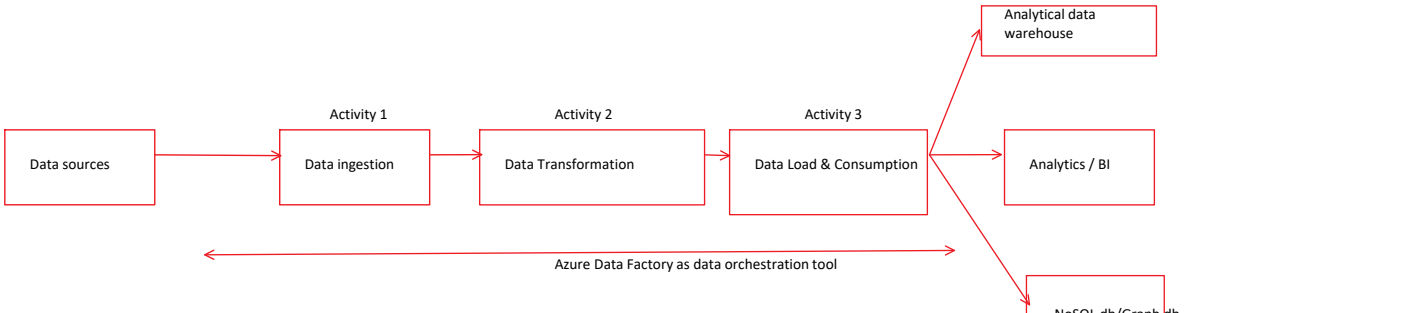
Action: 1 RunHiveScript
- Tasks
- Load data into hive table
- Select operation in the table

Oozie bundlers (bundling the multiple Jobs/tasks altogether in an action)

Action 2:
SqoopImport/Export job
- Tasks
- Load data from hive to SQL server
- Select operation in the sql server db table

Azure Data Factory

08 September 2022 18:47



-- It helps for data movement from source to destination

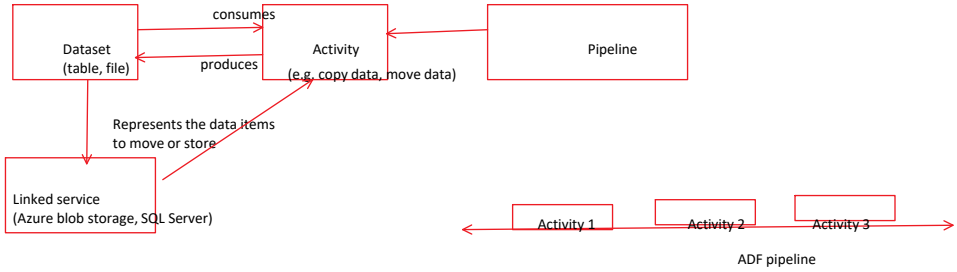
- It helps to perform the core ETL/ELT operations on data pipeline
- Data integration with various data consumers, data marts and data warehouses
- Azure data factory allows us to create data driven workflow
- In Azure data factory, we can create data pipeline from data ingestion, transformation to consumption
- We can use any Azure based services, 3rd - party services are also supported for importing and exporting data.
- Code free data analytics service which can help data ingestion, data pipeline end to end from data ingestion, consumption through its data flows, control flows we can monitor and schedule.

1. Linked Service - creates a linking connection between the data source and the Azure Data Factory pipeline.
2. We must have to create the linked service to link up the data source to the Data Factory

e.g. database connection strings which define the connection information required for the service to connect to the external resource

- Azure storage linked service connecting the storage account to the ADF service. (connection string of the storage account)

Is a logical consolidation of multiple activities



Activity - It is the resource defines us which action/operation to perform on the data.

(copy data, move data,
Aggregation of data,
Schema enrichment,
Looping , joining, combining,
Complex data analysis,
User-defined customized operations)

Microsoft.Sql
Microsoft.Storage
Microsoft.Synapse
Microsoft.DataLakeStorage
Microsoft.DataLakeAnalytics
Microsoft.Databricks
Microsoft.AnalysisServices

Copy Data Activity - A hybrid data connectors

- The copy activity is executed on an Integration Runtime

a) Accessible over the IP address, integration runtime is responsible to connect to the underlying data sources based on the private cloud/network securely, reliably along with scalability.
b) When we are copying the data to and from the data sources located on-premise on in network access control (Azure vnet, private cloud network), in case of on-premise or private cloud network, we have take help of self hosted integration runtime of data factory

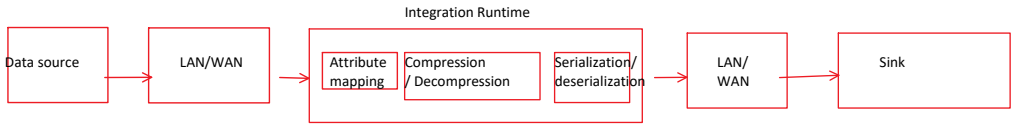
Data copy
a) (self-hosted integration runtime)

- On-premise SQL server/mysql/openshift based db --> azure database

---> private cloud based data sinks

b) (Azure Integration runtime) (default)

- Azure , AWS, any cloud based resources data transfer/data copy operation

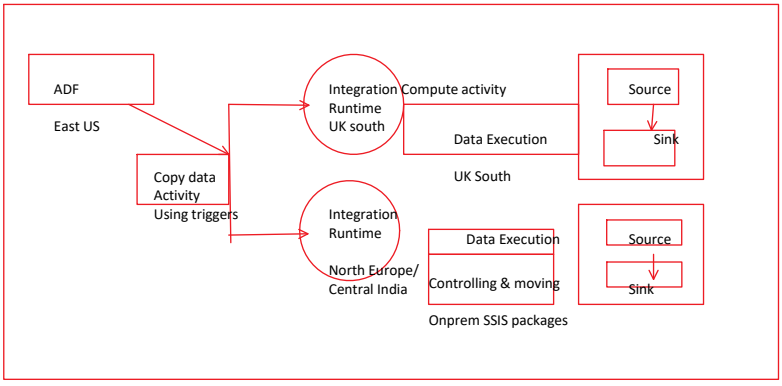


- Read the data from the data source
- Perform the serialization / deseralization of the data transformation, compression/decompression, attribute mapping, schema enrichment, schema drift checking , performing the configuration of input dataset, output dataset and manages the end to end copy activity in the ADF pipeline
- Writes the data back to the sink (based out same public cloud/private cloud or network)

Integration Runtime is the compute infrastructure used by ADF to provide the facility to connect to various data sources and sinks across different regions around the globe for data integration purpose.

It provides the benefit of connecting across the different network environment through the ADF integration runtime.

a) Azure Integration Runtime -- data copy, move, transfer, transformation activity
b) Self-hosted integration runtime -- data copy/move/transform activity through on-prem/private cloud network
c) Azure-SSIS integration runtime -- data copy/move/transform activity for SSIS data packages to Azure



Azure Integration Runtime (IR)

Azure	Public network	Private link
	Data Flow	Data Flow
	Data Copy	Data Copy
	Data Movement	Data Movement

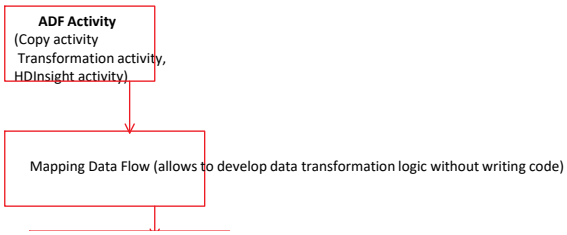
- Able to connect to multiple data sources and sinks across public and private network (private endpoint on Azure over private link)
- Run copy activity (from Azure storage account deployed over private endpoint of private network to the another cloud resource (AWS S3 bucket deployed over AWS private VPC/ private network)
- Various transformation activity - copy activity, pig activity, MapReduce activity, spark activity, Azure Data Lake activity, web activity, copy activity
- Fully managed, serverless compute resource in Azure
- Elastic scalability

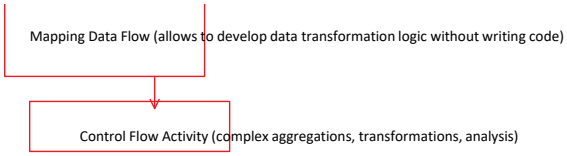
Lab-2:

Copy Data Pipeline (Goal for this lab)

a) Create Azure SQL server and Database (sample)
b) Keep the data in .csv format within ADF pipeline
c) Create the sink and ADLS Gen2 (Azure data lake storage account) (created in previous lab)
d) Copy the data from SQL db to ADLS gen2 in csv format

Pre-requisites (ADF v1)
Visual Studio 2013 / 2015
Azure Data Factory tool for VS 2015





Mapping Data Flow in ADF

- Mapping data flows are core part of data transformation in ADF.
- The resulting data flows can be executed as activities within the Data Factory pipeline that use the scaled -out apache spark clusters.
- Data flow activities can be operationalized using the existing ADF schedule, control flow and monitoring capacity.

Authoring Data Flow Components :
Mapping data flow has a unique authoring canvas designed to build transformation logic easy. The data flow canvas us separated into three parts -

a) The top bar - searching of options like data flow debug mode
b) Graph bar - displays the transformation stream while apply data processing logic. It also shows about the lineage of the source data as it flows to one or more sinks.

We can select "Add source" to add a new data source,
We can select "add new transformation" to add a new transformation logic

c) Configuration panel - shows the settings specific to the selected transformation. All the parameters can be defined onto it.

Schema Drift

It is the scenario for the data sources when often there's a change in metadata, fields, columns and types which can be added, removed or modified.

Without handling the schema drift, the data flow becomes vulnerable to upstream data changes, as part of transformation logic

Typical ETL patterns fail when the incoming columns and the fields get changed because they tend to tied to the typical data source names.

-- define sources which can provide mutable data types, field names and values/sizes
-- define transformation parameters which can work with the data patterns instead of hard-coded values and fields.
-- define the expression which can understand the patterns to match incoming fields instead of using named fields.

Flowlets

A reusable container of ADF activities which can be created from an existing mapping data flow or started from scratch.

With the flowlets we can create logic for doing data operations like address changing/string trimmer, mapping the input and output to the columns based on the calling data flow for a dynamic streaming (used for code reusability purpose).

Pre-requisites for Data Flow Lab

1. Azure Data Factory (create using Azure portal)
2. Azure Data Lake Storage (ADLS gen2) created (lab 1) (managed identity permission to be given)
3. Azure SQL db with sample adventure works database (lab2)
4. Create the Linked Services for ADLS and Azure SQL db (lab2) in ADF
5. Create the dataset for ADLS and Azure SQL db (lab2) in ADF
6. Create the pipeline with Copy data Activity

Create the ADF Mapping Data Flow - Lab 4

a) Created few datasets with ADLS
b) Created the actual Data flow in ADF
c) Connected with the ADLS dataset sources & did the basic transformation(select -> column mapping, remove duplicates, Lookup -> joining with product stream with productcategory and productmodel stream)
d) Created the Sink on the Dataflow using ADLS gen2 dataset within the same container which contains the output from the dataflow using parquet format, highly performance efficient, compression enabled, columnar based file format which is faster processing
e) To run the dataflow, an ADF pipeline has created with the data flow activity there used the existing built data flow, then we debug the pipeline to run the data flow.

Control Flow Activity

The Control Flow activity allows to build complex , iterative processing logic within the ADF pipeline

Activities	Description
Append variable	Append variable activity could be used to add a value to an existing array variable defined in a ADF pipeline
Set Variable	Set Variable activity can be used to set the value of an existing variable of type String, Bool or array defined in the ADF pipeline
Execute Pipeline	The Execute pipeline activity allows a ADF pipeline to invoke another pipeline
If condition	If condition activity allows directing pipeline execution based on the evaluation of certain expressions
Get Metadata	Get Metadata activity can be used to retrieve metadata of any data in ADF
ForEach	The Foreach activity can be defined a repeated controlling process in the ADF pipeline

Lookup Activity	Lookup activity can be used to retrieve the dataset from any of the Azure supported data sources applying lookup stream for retrieval of data mainly from left side table. Similar to left outer join in SQL. It is used to reference the another stream
Filter Activity	Can be applied on the pipeline to define a filtering expression to an input array
Until Activity	Until Activity allows to execute the set of activities to put in a loop until the condition associated with the activity evaluates to True.
Wait Activity	Allows to pause the pipeline execution for the specified time period
Web Activity	Web activity can be used to call a custom REST endpoint from the ADF pipeline
Azure Function	Allows to run the Azure Function in the ADF pipeline. Azure Function is a serverless core Azure PaaS service part of Azure App service. It allows to build cross-platform apps using any standard OOPS language, it helps to trigger the app/data pipeline to real time based on certain conditions e.g. Complex data processing scenario, an alert message can be triggered from the ADF pipeline based on new conditions, workflow processing as defined.

ADF Pipeline Variables

The process of creating ADF pipeline variables is similar to create the parameters. Unlike the parameters, the ADF variables can only be three data types.

- String
- Boolean
- Array

Add Dynamic Content - This window allows to build dynamic expressions interactively using the system variables and functions.

We'll use two kind system variables - a) Pipeline name
b) Pipeline Run Id
Function - Collection functions, Conversion functions, Date Functions, Logical functions, Math functions, String functions - Concat(), append(),

Lab 5: Build a reusable Pipeline

Pre-requisites

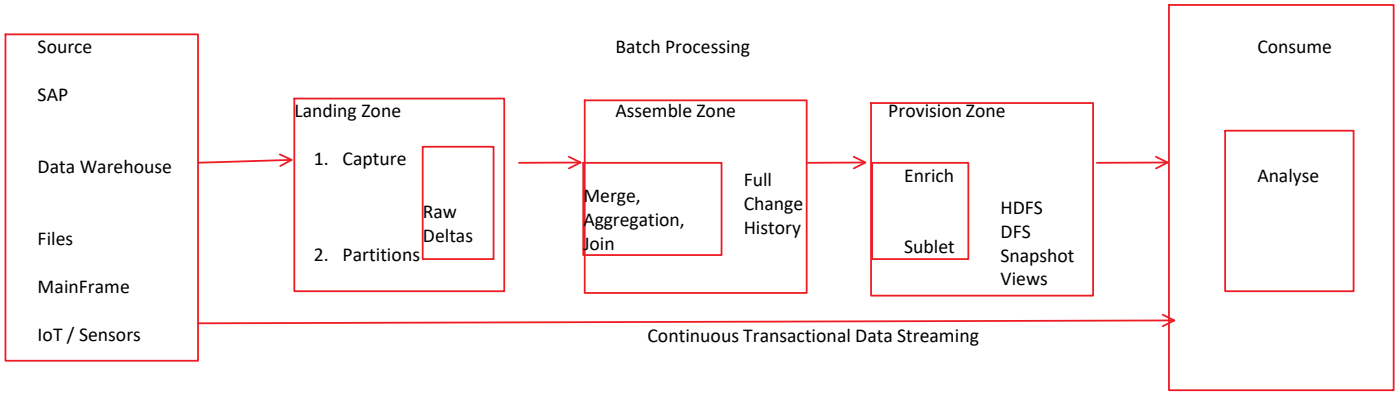
1. Azure Data Factory (create using Azure portal)
2. Azure Data Lake Storage (ADLS gen2) created (lab 1) (managed identity permission to be given)
3. Azure SQL db with sample adventure works database (lab2)
4. Create the Linked Services for ADLS and Azure SQL db (lab2) in ADF

Goal:

a) Create a reusable dataset (using dynamic content)
b) Create a reusable pipeline to retrieve the data from SQL dataset dynamically (10 different tables) & sink to ADLS storage.

Azure Data Lake Gen2

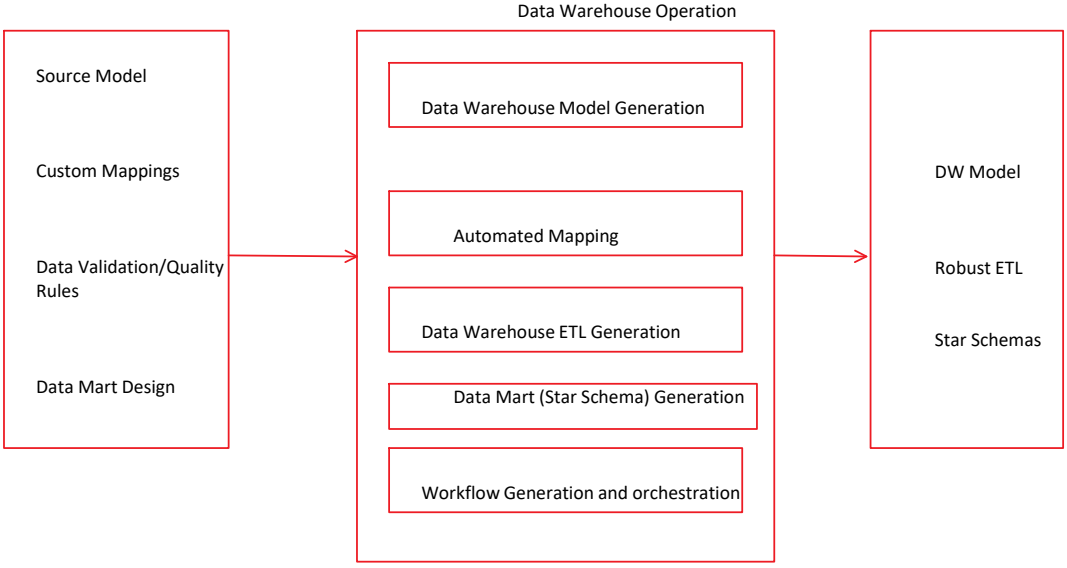
08 September 2022 18:47



Benefits:

a) Massive volumes of structured and unstructured data like ERP transactions and weblogs can be stored in 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 Warehouse



Benefits :

a) Little or no data prep is required, making it easier for us to do analysis and business users to access and analyse the data
b) Accurate, complete data which can be available more quickly, so that business can turn the information into insight faster
c) Unified, harmonized data offers a single source of truth, building out the data insights and decision making across the business lines.

	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 time Even accessible 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	Data from a data lake, with its large volume of unstructured data can be used by data engineers and data scientists who prefer to study data in its	Data from a data warehouse is typically accessed by managers, business stakeholders looking for the quick insights from the business KPI, and as the data has been structured to

	raw form to gain new and unique business insights	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 its structured only while required	ETL (Extract, Transform, Load), in this process, data is extracted from its sources, scrubbed, parsed, then make it structured for business - end analysis
Cost	Storage costs are fairly cheaper in a data lake, also less time consuming to manage which reduces the operational costs	Data warehouses cost more than the data lakes, and required more time to manage, resulting in additional operational costs.

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 with 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 improves the directory management operations, which overall makes efficient the job performance
b) Management wise earlier we can organize, manipulate the files through directories and subdirectories
- c) Security is enforceable we can define POSIX permissions 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 HDI, Azure DataBricks and Azure Synapse Analytics.

Azure Blobs	Container	Virtual Directory	Blob	
ADLS Gen2	Container	Directory	File	

.Root
/ Child
/folders
/file1, file2… fileN

Difference between ADLS Gen1 And ADLS Gen2

Difference	ADLS Gen1	ADLS Gen2
Account Root Permissions	Permission required on 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 container root can view the account root content
RBAC roles and access control	All users in RBAC owner role are superusers. All other users (non-superusers) need to have permission 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 is set, on the parent directory/items before the child directory/items are being created. The file store permission can be inherited from parent to child directory/folder..
Nested file or nested directory creation for non-owner user	Check whether write-execute (wx) permission 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 , file/directory is created, the final permission will be same as the user provided permission	File/directory is created , the final permissions will be calculated based on the [user provided permission + umask) value.
UMask	Clients can apply umask on the permission on new file/directory before 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.

Access Control Permissions set up for ADLS gen2
RWX -- read + write + execute
R-X -- read + execute
R-W - read + write
R-- Read
--- no permissions

ADLS umask

For a ADLS Gen2 container, the mask which is applied for ACL level of the root directory, ("/") defaults to the value of 750 for directories, the value of 640 for files.

	Directories	Files
Owning user	rwx	Rw--
Owning group	r-x	r--
Other	---	…

Lambda Architecture

1. Batch processing Layer - ADLS Gen2, Azure Blob storage, HDInsight Hive activity, pig activity
2. Speed data processing layer - Azure event hub, stream analytics, Spark streaming
3. Serving layer - Analytics data store as well as includes the Azure Analysis service, BI and analytics layer.

1. Model --> on data access layer is built
2. View --> (.cshtml, .vbhtml)
3. Controller ---> takes all the data models fetched from the asp.net mvc model & defines the logic (Itemcontroller, HomeController)

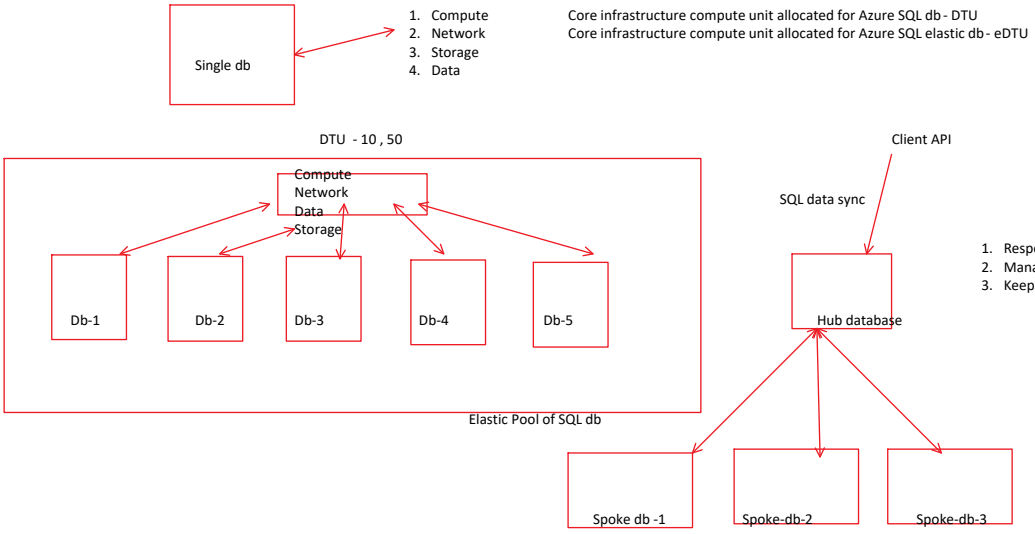
Azure SQL database

08 September 2022 18:47

TCO = Total cost of ownership

Total cost of ownership is referred where the cost of cloud/ Azure resources can be optimized according to its underlying infra, network, storage, backup, high availability features. Also, it's defined the options of how the cost of the specified cloud resources can be reduced without bringing down the core functionality/ business requirements.

SQL Server on Azure VM	Azure SQL Managed Instance	Azure SQL database
Simple migration of application where the core SQL Server on-prem features are required. (e.g. .net framework runtime, CLR, SQL Server Agent, SQL Server Broker, Log shipping) etc.	Lift - shift of database migration From on-premise to Azure where We can get all of the core SQL db benefits Over cloud , without managing the underlying compute, storage, network, disk	Purely managed SQL database offering where no licensing required, no underlying administration of server, SQL database, network is required. End to end database migration is feasible with just a click focusing into the core sql development features.
License is required, either through BYOL, license has to be procedure	No license required, entire SQL db on-prem 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 are there (max 5 TB -> 100 TB)
SSAS, SSRS, SSIS, SQL Server broker, .net framework runtime, CLR integration, distribution 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, ACID, CLR related functions, stored procedures based on Windows runtime etc. are not supported.
SLA - 99.9% Automated backup, Azure site recovery for disk level backup of database	SLA - 99.99% , 99.95% Automated backup, point-in time restore, geo-replication, high availability, automated patching	SLA - 99.99% Automated backups, active-geo replication, high availability & DR, replication on both AZ and regional level.



Azure Storage usage scenarios

Blob storage - store all kinds of object based stores, like media files, documents, large files
File share - store all kinds of filesystems, on-premise file share can be migrated to azure file share
Queue storage - messages coming from messaging platform - ActiveMQ, Kafka,
Table Storage - non-relational dataset that not compatible with Codd's rules
Disk storage - disks in form of virtual hard disks can be stored in Azure Storage and can be re-utilized for usage in other VMS

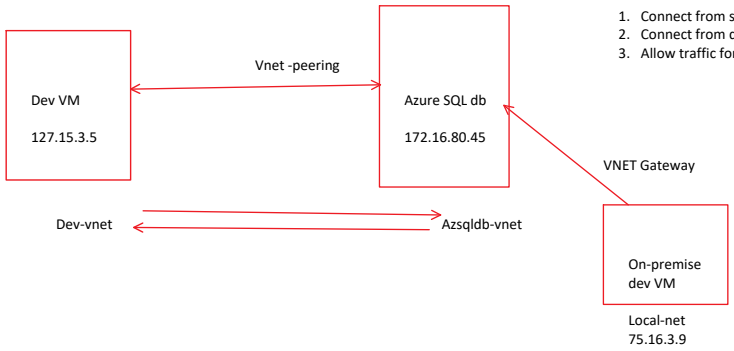
Lab 1 and Lab 2 :

1. Provision an Azure SQL elastic pool db
2. Connect to the database from SSMS / ADS
3. Create Azure SQL db using CLI
4. Create Azure SQL db on VM
5. Migrate on-premise SQL db to Azure SQL db using Azure Database migration assistant tool (Azure Data Studio)

Data Migration from on-premise SQL server db to Azure SQL db

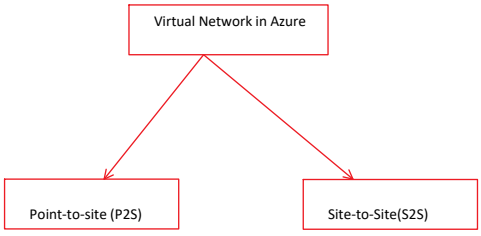
On-prem	Azure SQL db	Azure SQL Managed Instance	SQL Server on Azure VM
adventureworks	Data Migration Tool	Azure SQL extension of Azure Data Studio	Azure SQL extension of Azure Data Studio
adventureworks	SQL data import-export wizard (SSMS)	Data Migration Tool	Data Migration Tool

Key - AES 256 bit (Customer-managed key)
SHA 128/256 bit



1. Connect from source vnet to destination vnet
2. Connect from destination vnet to source vnet
3. Allow traffic forwarding capability and allow remote gateway

Express Route - a Managed service for connecting your on-premise network to Azure Virtual Network (in AWS Direct Connect)



1. Multiple users part of same domain of the corpnet tries to access the bunch of Azure resources on Azure Vnet.

1. A single user is connected over
   Versatile resources on Azure Vnet

1. Principle of least privilege - to provide the minimal permission to the user based on their accessibility level and to do respective work on Azure

Encryption at Rest	Azure SQL db	Azure SQL managed instance
Transparent data encryption (TDE)	Real-time encryption and decryption of the db, backups, transaction log files	Real-time encryption and decryption of the db, backups, transaction log files
	Enabled by default for encryption of data at rest	Enabled by default for encryption of data at rest
	During encryption , the TDE encrypts the storage of the entire database by using a symmetric key called the db encryption key (DEK). During the db startup time, the encrypted DES is decrypted and used for decryption and re-encryption of the db files in the sql db engine.	During encryption , the TDE encrypts the storage of the entire database by using a symmetric key called the db encryption key (DEK). During the db startup time, the encrypted DES is decrypted and used for decryption and re-encryption of the db files in the sql db engine.
	During the usage of AKV, for customer managed key - TDE protector is used to protect the data encryption key. Is stored as asymmetric key in key vault	During the usage of AKV, for customer managed key - TDE protector is used to protect the data encryption key. Is stored as asymmetric key in key vault
Encryption in Transit	Enabled through TLS 1.2 protocol	Enabled through TLS 1.2 protocol

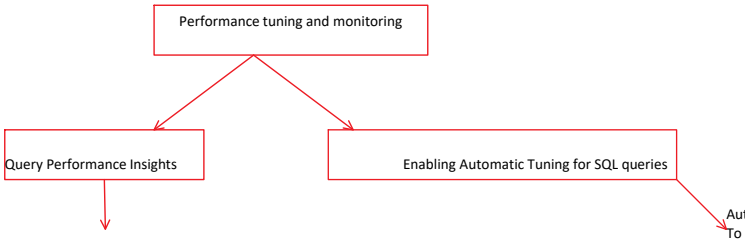
1. Create and configure Azure SQL db over private link

a) Existed VNET/ Create a new VNET and allow bastion host
b) Create a VM
c) Azure SQL server to be enabled over the private endpoint
d) Test the connectivity to the SQL server private endpoint

Performance tuning and recommendations

1. Create index over Azure SQL database tables (clustered or non-clustered)
2. Apply Schema
3. Automatic tuning for Azure SQL db tables

Performance tuning and monitoring



Automatically manage the data service
To continuously monitor the queries and identifies the action which we can Use to optimize the Performance of workload.
We can review recommendations and manually could apply them which is known as automatic tuning mode.

It provides query analysis for single or pooled databases which helps to identify the Top resource consuming and long-running queries in the workload.

- Deeper insight s into the database (DTU) consumption
- Details on top database queries by CPU, duration, and execution count
- The ability to drill down into the details of a query, to view the query text and the history of resource utilization.
- Azure advisors for database recommendation

MAXDOP - the maximum degree of parallelism (MAXDOP) is a server configuration option for running SQL Server on multiple CPU. It controls the number of processors used to run a single statement in parallel plan execution.

The default value for MAXDOP is 0 which enables the SQL server to use all possible processors.

In Azure SQL db, we can check and configure the dmv named as sys.database_scoped_configurations system catalog view.

```
Select [value] as currentMaxDop from sys.database_scoped_configurations
WHERE [name] = 'MAXDOP';
```

Azure Blob Storage

08 September 2022 18:48

Azure Analysis Service

08 September 2022 18:48

Azure Synapse Analytics

08 September 2022 18:48

Apache Spark

08 September 2022 18:48

Python Programming

08 September 2022 18:48

Azure Databricks

08 September 2022 18:49

Overview of AWS

08 September 2022 18:49

Overview of GCP

08 September 2022 18:49