Custom training (Day 1 - 29/08/2023)

Database - Collection of data
Data - Piece of information
Information - Facts

DBMS(Database Management System) - is a software system that enables users to efficiently create, manage, and access databases.

Purpose of DBMS:

- Efficiently manage and organize large volumes of data
- Provide mechanisms for data retrieval, storage, and manipulation
- Ensure data integrity,security, and consistency
- Support concurrent access by multiple users

Key Components of DBMS:

- Data: information that is stored, organized and managed
- Database: a collection of related data that is stored in a structured way
- Software: The DBMS software that allows users to interact with databases
- Hardware: The physical equipment on which the DBMS runs

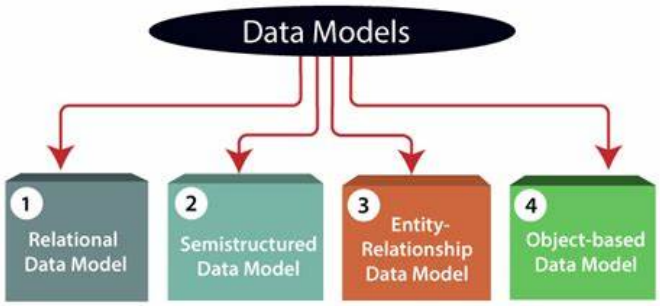
Advantages of using DBMS:

- Centralized data storage and easy access
- Data security and access control
- Data integrity and consistency
- Data independence and abstraction
- Concurrent data and transaction management

Different types of database - Relational, Object oriented, No SQL, Hierarchical, Graph, Network
MySQL, Oracle, SQL Server

Basics of DBMS:
Data Models

- Data models are frameworks used to represent data, relationships, constraints, and rules within a DBMS
- Common data models include hierarchical, network, relational and object-oriented.



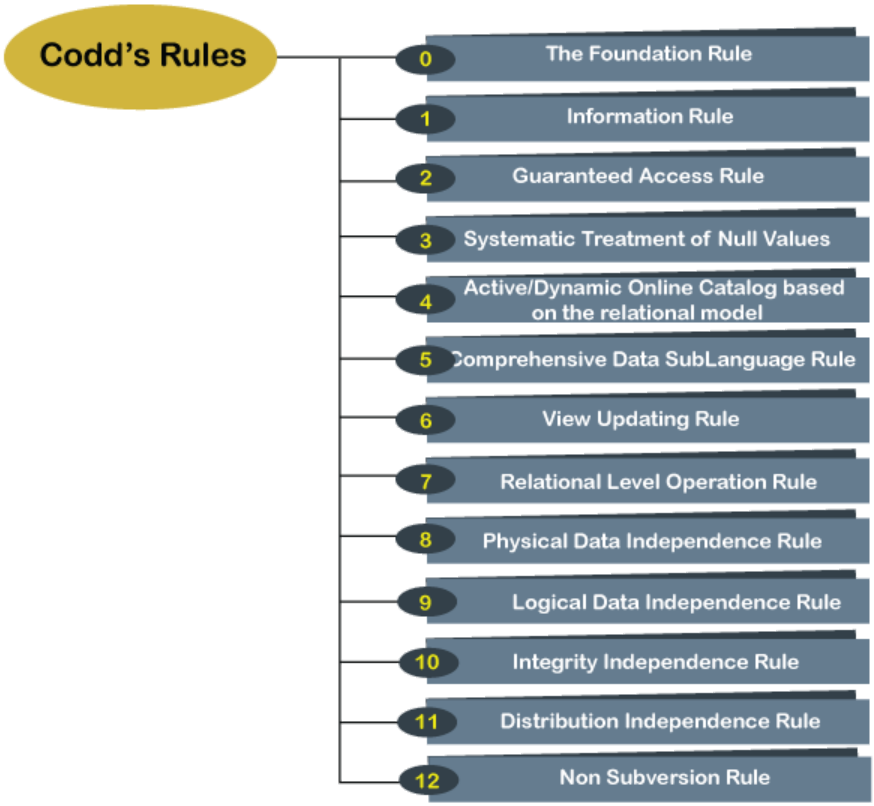
Relational Data Model:

- Grid like mathematical structures consisting of rows and columns
- All data is logically structured in form of tables

Structured DB, Unstructured DB

DBMS V/S RDBMS (Database v/s Relational Database)

Codd's rule



Rule 0: The Foundation Rule

The database must be in relational form. So that the system can handle the database through its relational capabilities.

Rule 1: Information Rule

A database contains various information, and this information must be stored in each cell of a table in the form of rows and columns.

Rule 2: Guaranteed Access Rule

Every single or precise data (atomic value) may be accessed logically from a relational database using the combination of primary key value, table name, and column name.

Rule 3: Systematic Treatment of Null Values

This rule defines the systematic treatment of Null values in database records. The null value has various meanings in the database, like missing the data, no value in a cell, inappropriate information, unknown data and the primary key should not be null.

Rule 4: Active/Dynamic Online Catalog based on the relational model

It represents the entire logical structure of the descriptive database that must be stored online and is known as a database dictionary. It authorizes users to access the database and implement a similar query language to access the database.

Rule 5: Comprehensive Data SubLanguage Rule

The relational database supports various languages, and if we want to access the database, the language must be the explicit, linear or well-defined syntax, character strings and supports the comprehensive: data definition, view definition, data manipulation, integrity constraints, and limit transaction management operations. If the database allows access to the data without any language, it is considered a violation of the database.

SQL : DDL, DML, DQL, TCL, DCL

Primary key, Foreign key

Rule 6: View Updating Rule

All views table can be theoretically updated and must be practically updated by the database systems.

Rule 7: Relational Level Operation (High-Level Insert, Update and delete) Rule

A database system should follow high-level relational operations such as insert, update, and delete in each level or a single row. It also supports union, intersection and minus operation in the database system.

Rule 8: Physical Data Independence Rule

All stored data in a database or an application must be physically independent to access the database. Each data should not depend on other data or an application. If data is updated or the physical structure of the database is changed, it will not show any effect on external applications that are accessing the data from the database.

Rule 9: Logical Data Independence Rule

It is similar to physical data independence. It means, if any changes occurred to the logical level (table structures), it should not affect the user's view (application). For example, suppose a table either split into two tables, or two table joins to create a single table, these changes should not be impacted on the user view application.

Rule 10: Integrity Independence Rule

A database must maintain integrity independence when inserting data into table's cells using the SQL query language. All entered values should not be changed or rely on any external factor or application to maintain integrity. It is also helpful in making the database-independent for each front-end application.

Rule 11: Distribution Independence Rule

The distribution independence rule represents a database that must work properly, even if it is stored in different locations and used by different end-users. Suppose a user accesses the database through an application; in that case, they should not be aware that another user uses particular data, and the data they always get is only located on one site. The end users can access the database, and these access data should be independent for every user to perform the SQL queries.

Rule 12: Non Subversion Rule

The non-submersion rule defines RDBMS as a SQL language to store and manipulate the data in the database. If a system has a low-level or separate language other than SQL to access the database system, it should not subvert or bypass integrity to transform data.

SQL

Entity: is a distinct and real-world object, concept, or thing that can be identified and described

In DBMS, entities represent main objects about which data is stored

ER diagram activity

ER diagram represents :

1. Entities
    - weak entity : An entity that relies on other entities. Does not have any primary key

      Room no→ classroom → relation ← school → school number

      The classroom is a weak entity because room no. can be common for various schools but school number can never be common between two schools

2. Attributes (Key Attributes, composite attributes, multivalued attributes, derived attributes)
3. Relationship (**one to one** : when a single element of an entity is associated with another a single element of another entity eg, one student can have only one laptop, one student can have only one id card, student → has -- id card, etc., **one to many**, **many to one**, **many to many**)

Uses of ER Diagram in

1. Er diagram helps you conceptualize the database
2. Er diagram gives you better understanding of the information to be stored in the database

3. It reduces complexity
4. It helps to describe elements using entity relationship model
5. It allows users to get a preview of the logical structure of the database.

Er diagram symbols:

- Rectangles: This er diagram symbol represents entity types
- Ellipses: This symbol represents attributes
- Diamonds: This symbol represents relationship types
- Lines: It links attributes to entity types and entity types with other relationship types
- Primary keys: here, it underlines the attributes
- Double ellipse: represents multi-valued attributes.

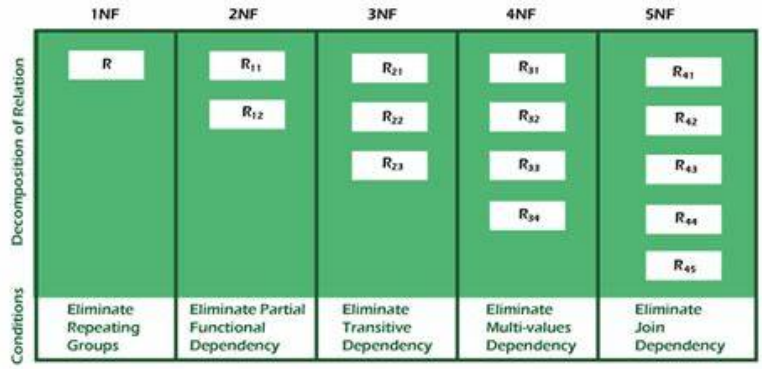
ER Modelling: vital tool for designing and conceptualizing complex database structures.

Need for ER modeling:

- Data complexity
- Data redundancy
- Data integrity
- Effective communication
- Query optimization

Normalization (splitting/dividing of larger tables into smaller tables and linking them using relationships)

- Process of organizing the data in database
- Used to minimize the redundancy from a relation or set of relations
- 1nf, 2nf, 3nf, 4nf, 5nf



1NF :

- Most basic form of normalization, which ensures there are no two same entries in a group.
- The most important rule is
    1. Each cell should contain a single value.
    2. Each record should be unique.

2NF:

- All the subsets of data that can be placed in multiple rows are placed in separate tables.
- Rules:
    1. It should be in 1nf already.
    2. The primary key should not be functionally dependent on any candidate key.

3NF:

1. Rules:
    a. It should be in 2NF already.
    b. It should not have any transitive functional dependencies.

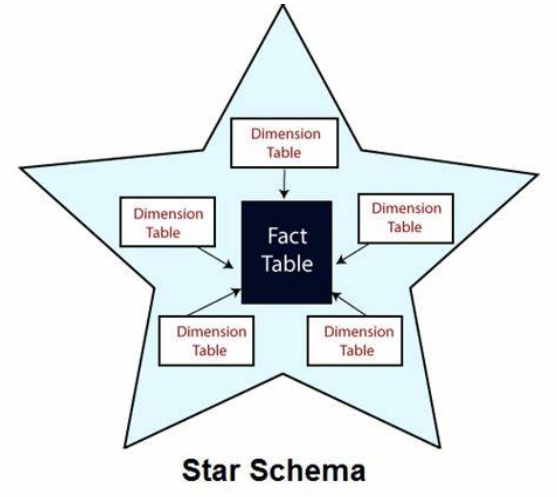
LUNCH BREAK

Case study

BCNF

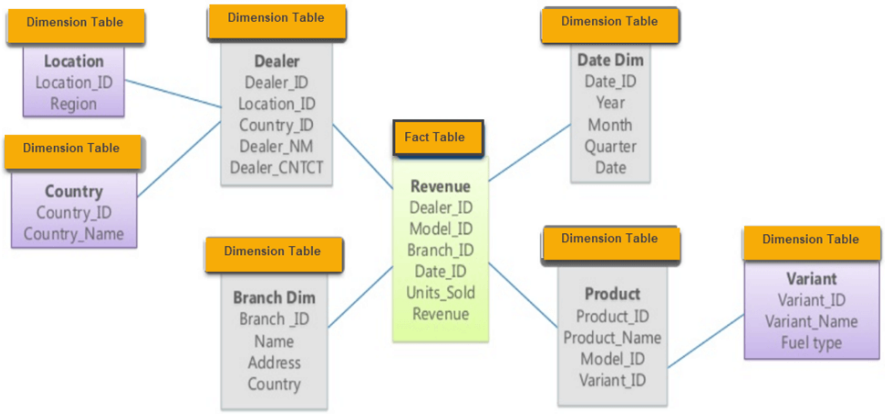
Star Schema

- a type of database schema commonly used in data warehousing and business intelligence systems
- Designed to optimize querying and reporting for analytical purposes
- Characterized by a central fact table connected to dimension tables in a star-like structure, hence the name “star schema”



Snowflake:

- Snowflake schema is a type of database schema that extends the star schema by further normalizing dimension table to eliminate redundancy
- The result is a structure that resembles a snowflake with dimension tables linked through multiple levels of relationships.



Day 2 (30/08/2023)

EDW - Enterprise Data Warehouse

RCD - Rapidly Changing Dimension is the dimension which has attributes where values will be changing often.

SCD -

- Slowly Changing Dimension is a concept in data warehousing and database management that deals with managing changes to dimension data over time.
- Dimension data includes descriptive attributes like customer names, product categories or geographic locations, which are often used for analysis,reporting and data visualization.
- They are categorized into different types (Type 1, Type 2, Type 3 and more) based on how historical changes to dimension data are handled.

Type 1 (Overwrite) -

- The old data is simply overwritten with new data when changes occur.
- Historical data is lost and the dimension table reflects only the latest state of data.

UCD - Unchanging

Denormalization -

- It is a database design technique that involves intentionally introducing redundancy into a relational database schema to improve query performance.
- While normalization is the process of organizing data to minimize redundancy and data anomalies, denormalization is used when the priority Is optimizing read performance for specific types of queries.

Day 2 (technically)

BIG Data:

- Refers to vast and complex volume of data that exceeds the processing of traditional database systems and requires specialized tools and techniques to store, process, and analyze effectively.
- It encompasses large datasets that are difficult to manage, process and analyze using traditional data processing methods.
- 4 V's of big data: volume, velocity, variety, veracity

Big data characteristics, challenges of big data (photo in phone)

Characteristics of big data:

1. Volume: big data involves massive volumes of data that can range from terabytes to petabytes and beyond.
2. Velocity: data is generated and collected at high speeds, often in real-time, from various sources like sensors, social media and devices.
3. Variety: data comes in diverse formats: structured(relational databases), semi-structured(XML, JSON) and unstructured(text, images, videos).
4. Veracity: refers to the quality and accuracy of the data, as big data sources may produce noisy, inconsistent or unreliable data.
5. Value: extracting valuable insights from big data can lead to improved decision-making, new revenue opportunities, and better customer experiences.
6. Variability: data flows can be unpredictable and vary over time. Handling data patterns is a challenge.

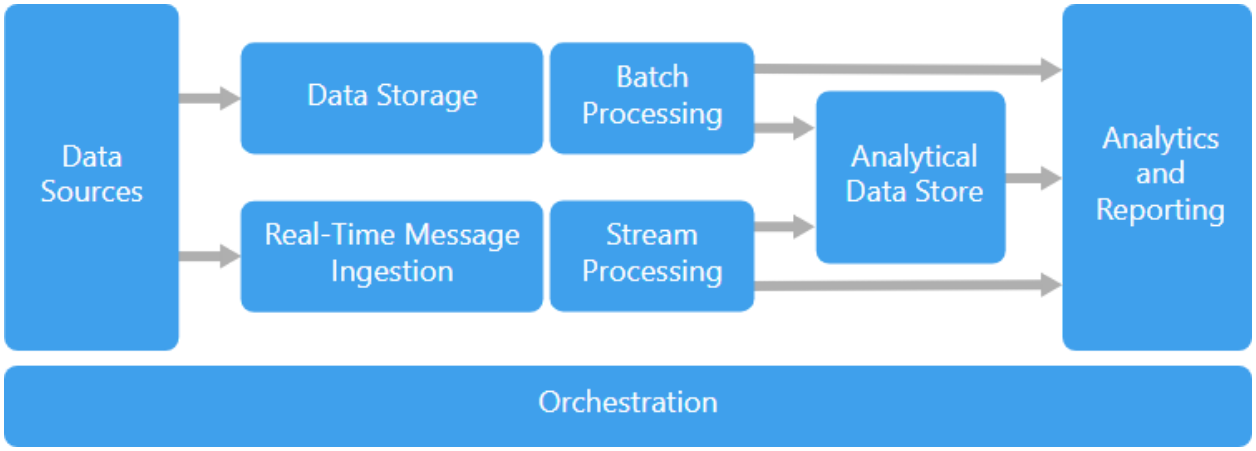
Challenges of big data:

1. Storage and Management: storing and managing massive volumes of data requires distributed and scalable storage systems.

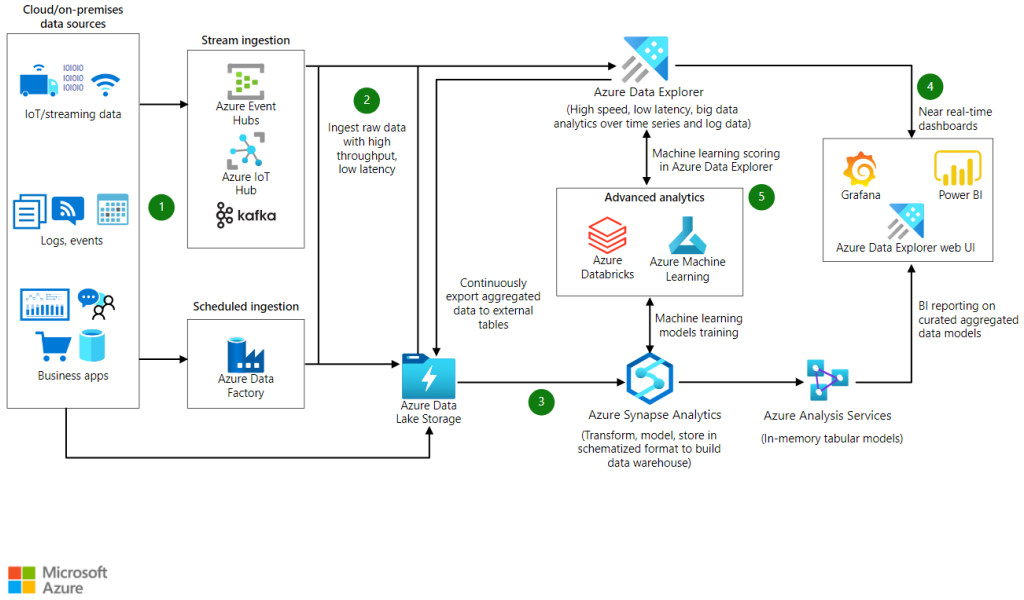
2. Processing power: traditional processing tools may struggle to handle the processing demands of big data. Distributed computing and parallel processing are needed.
3. Data integration: integrating and combining data from different sources with varying formats is complex.
4. Data quality: ensuring data quality and accuracy when dealing with diverse and rapidly generated data is a challenge.
5. Privacy and security: protecting sensitive data becomes more challenging with the increased volume and distribution of data.
6. Analysis and insights: extracting meaningful insights from big data requires advanced analytics techniques and tools.

Big data architecture style (Big data architecture style - Azure Architecture Center | Microsoft Learn)

Skeleton architecture



Azure data



Importance in big data:

1. Structured data: traditional relational databases have been managing structured data for decades. The analysis is relatively straightforward, making it useful for reporting and decision-making.
2. Unstructured data: it forms a significant portion of big data. Extracting insights from it requires advanced technologies like text mining, sentiment analysis, and image recognition.
3. semi -