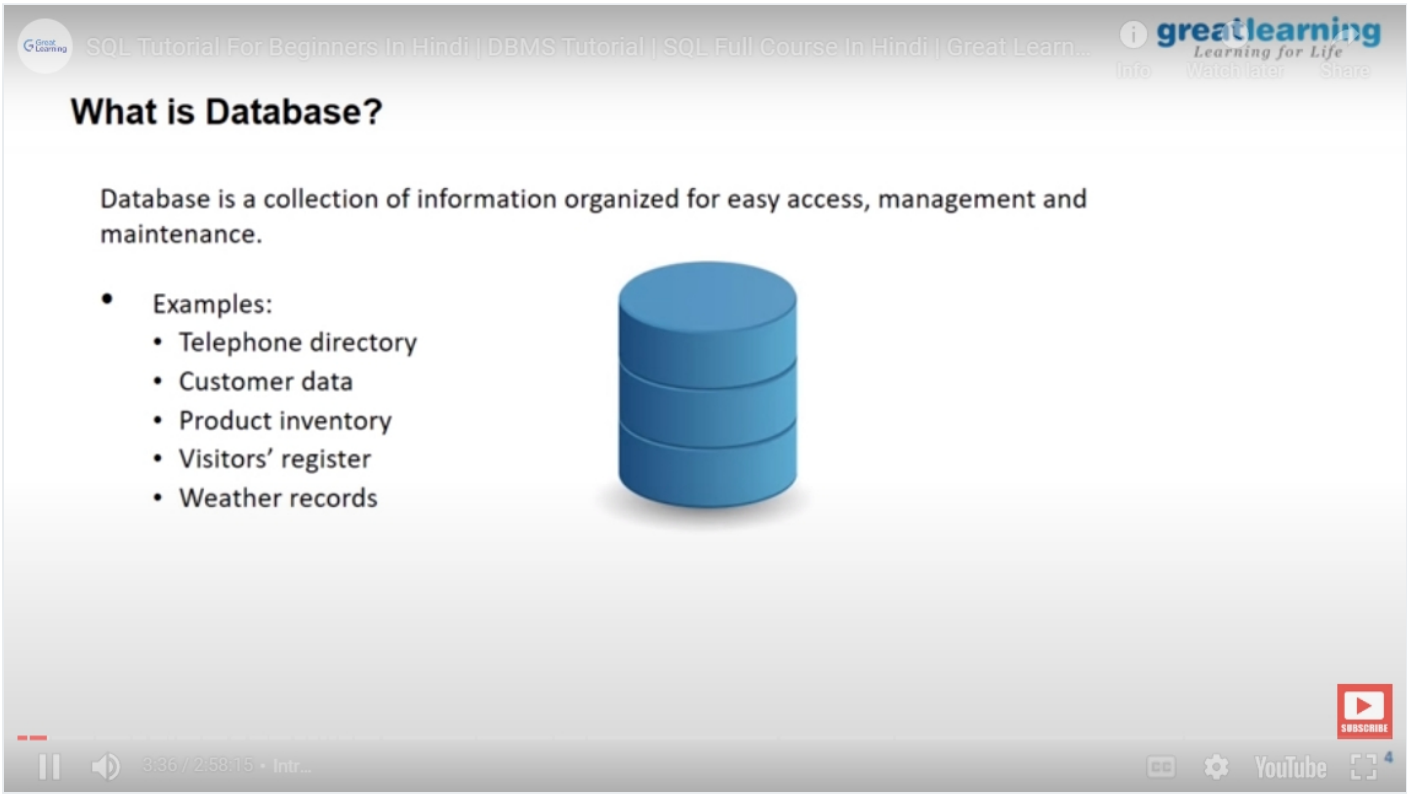
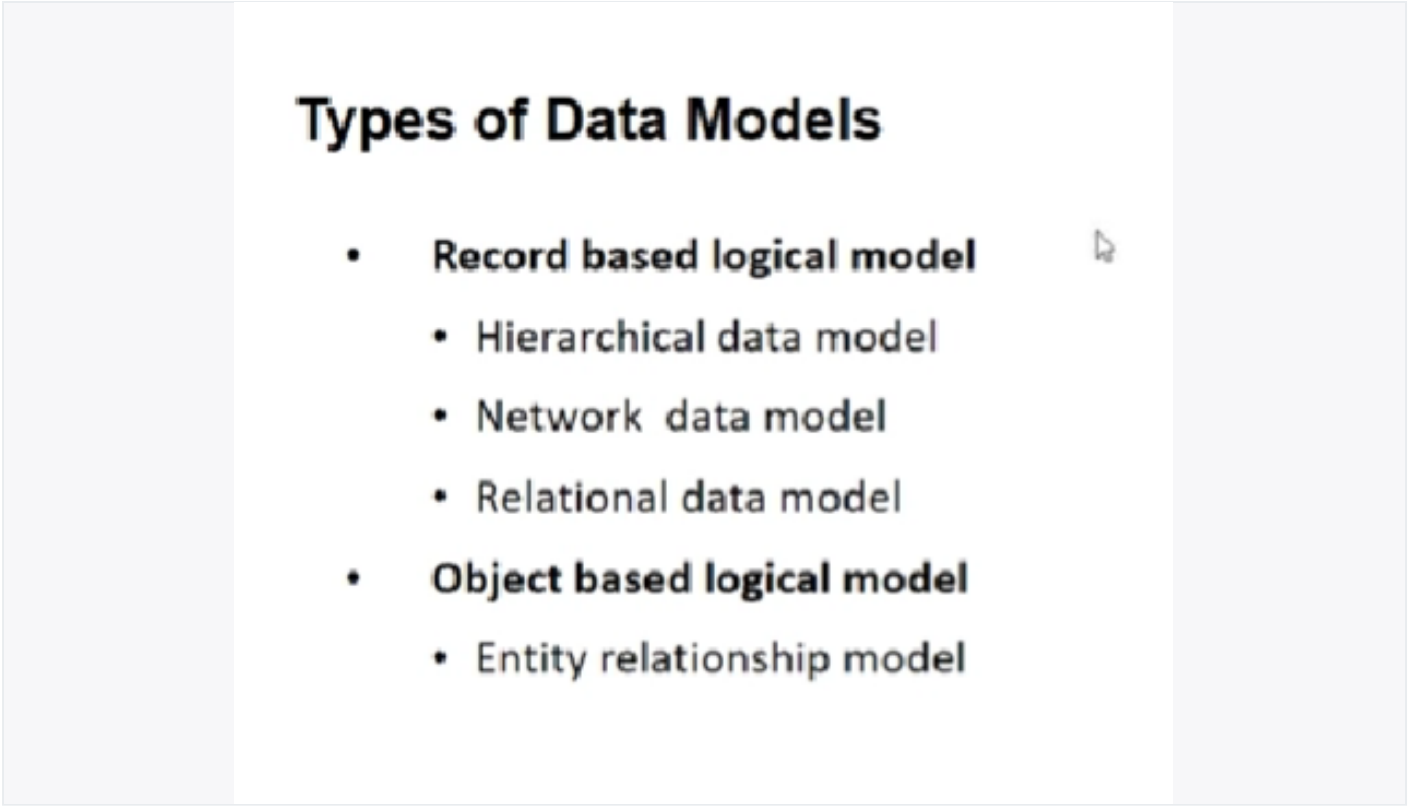
DBMS



logical structure of a database is called data models.



object-based logical model is the blueprint of your database

Search for: What is the difference between composite key and foreign key?

What do you mean by composite key?

A composite key is made by the combination of two or more columns in a table that can be used to uniquely identify each row in the table when the columns are combined uniqueness of a row is guaranteed, but when it is taken individually it does not guarantee uniqueness, or it can also be understood as a primary key made ... 26-Apr-2021

A compound key is similar to a composite key in that two or more fields are needed to create a unique value. However, a compound key is created when two or more primary keys from different tables are present as foreign keys within an entity. The foreign keys are used together to uniquely identify each record.

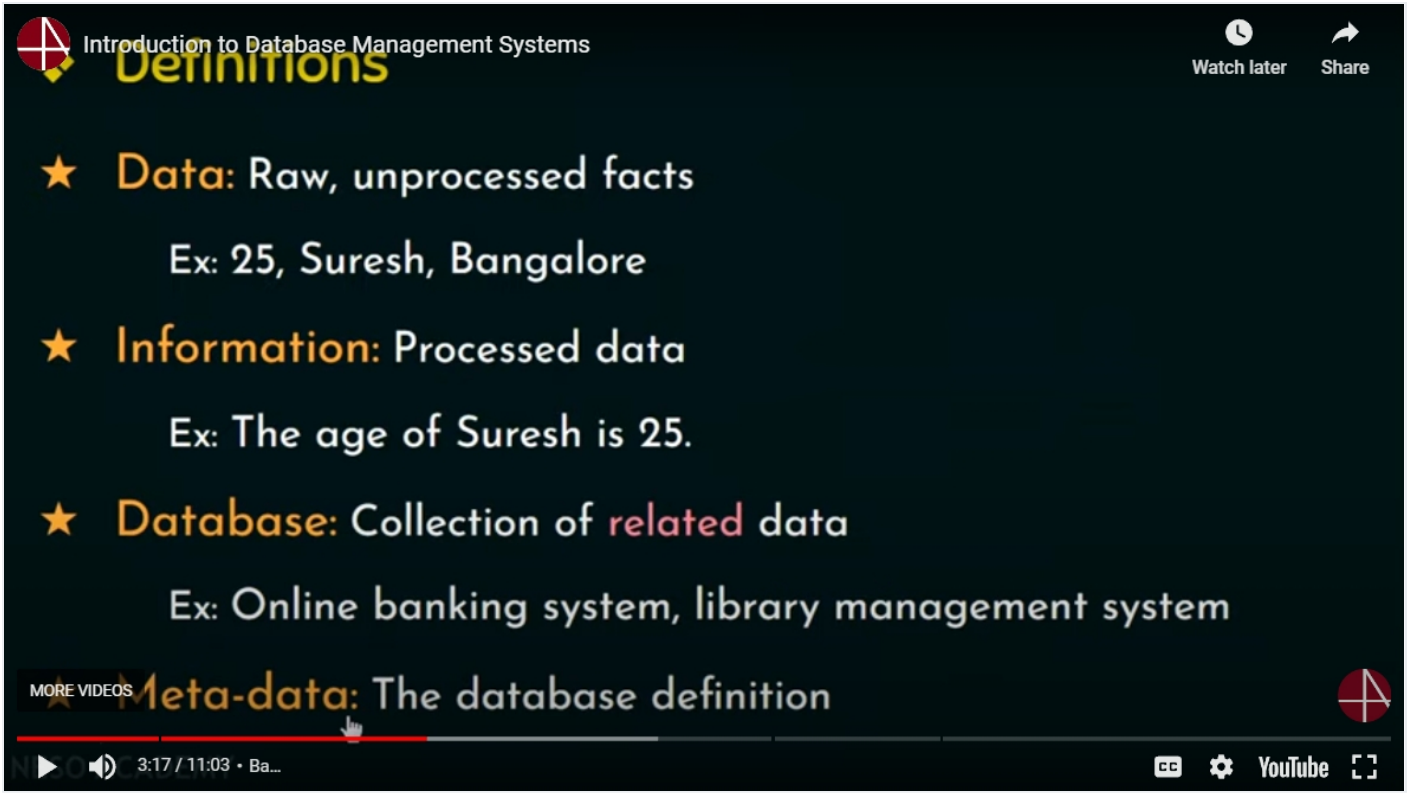
Basis	File System	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.

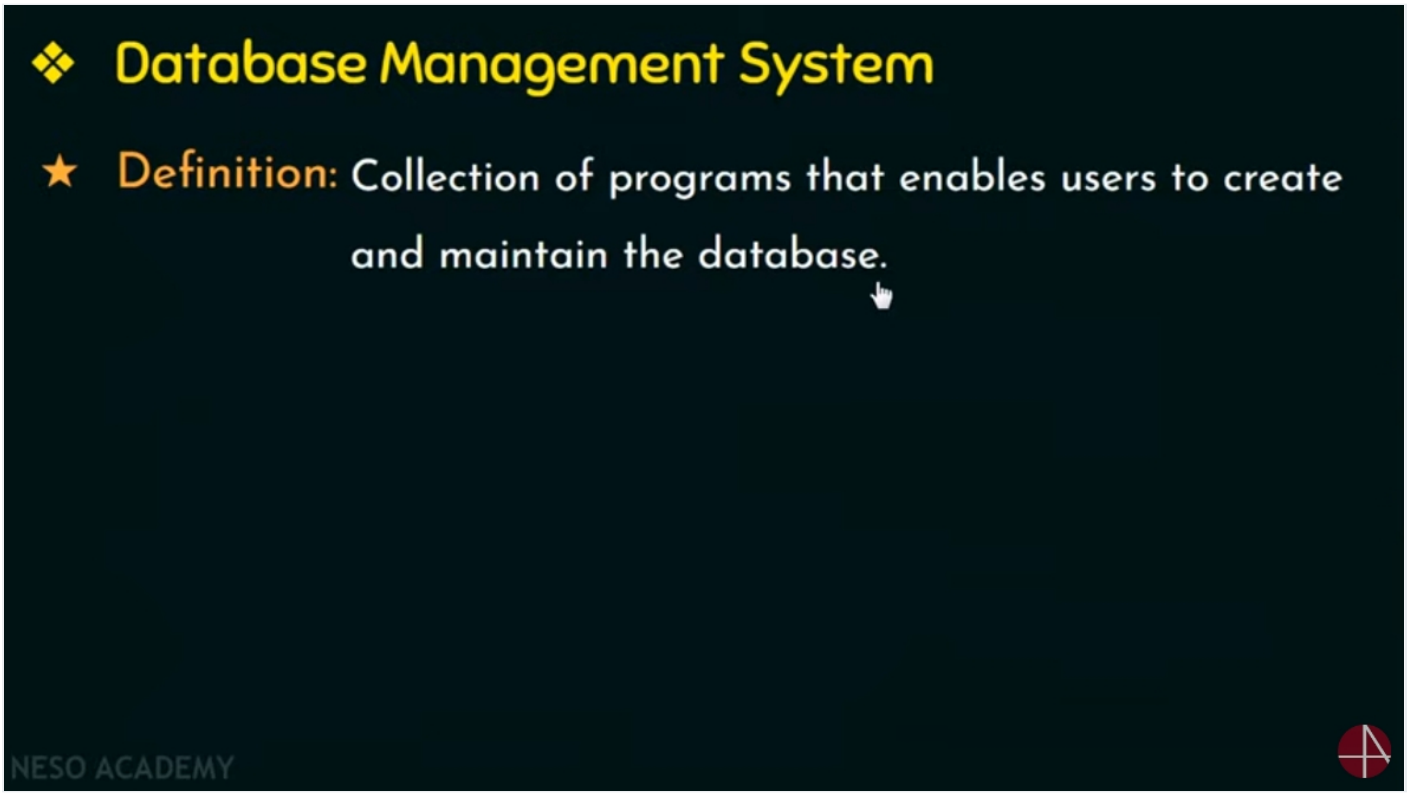
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
Data Independence	There is no data independence.	In DBMS data independence exists.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The user has to write procedures for managing databases	The user not required to write procedures.

RDBMS	DBMS
Data stored is in table format	Data stored is in the file format
Multiple data elements are accessible together	Individual access of data elements
Data in the form of a table are linked together	No connection between data
Normalisation is not achievable	There is normalisation
Support distributed database	No support for distributed database
Data is stored in a large amount	Data stored is a small quantity
Here, redundancy of data is reduced with the help of key and indexes in RDBMS	Data redundancy is common
RDBMS supports multiple users	DBMS supports a single user
It features multiple layers of security while handling data	There is only low security while handling data
The software and hardware requirements are higher	The software and hardware requirements are low
Oracle, SQL Server.	XML, Microsoft Access.

.....

DATABASE MANAGEMENT SYSTEM


Introduction to Database Management Systems
Watch later
Share
Definitions
★ Data: Raw, unprocessed facts
Ex: 25, Suresh, Bangalore
★ Information: Processed data
Ex: The age of Suresh is 25.
★ Database: Collection of related data
Ex: Online banking system, library management system
MORE VIDEOS
★ Meta-data: The database definition
3:17 / 11:03 • Ba...
YouTube

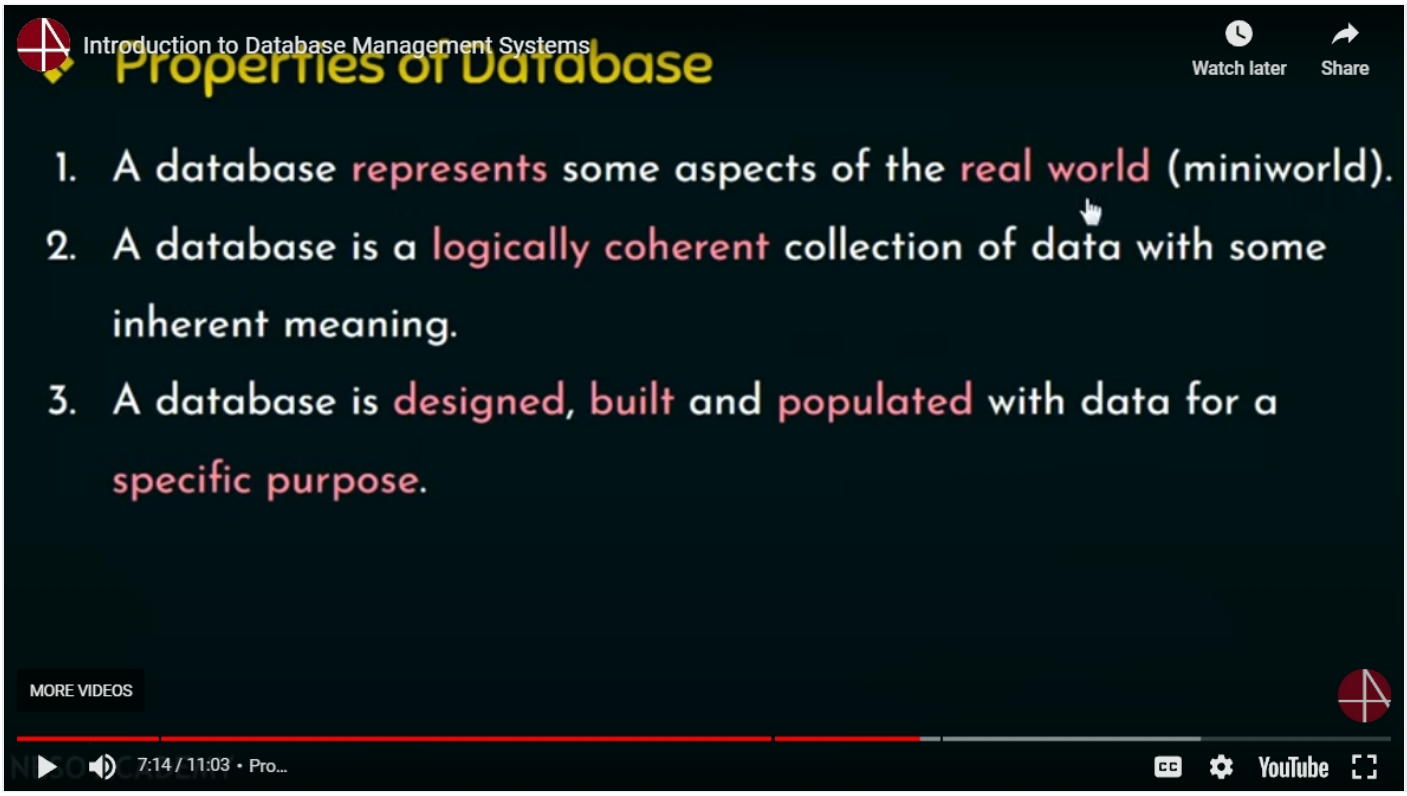

Database Management System
★ Definition: Collection of programs that enables users to create and maintain the database.
NESO ACADEMY

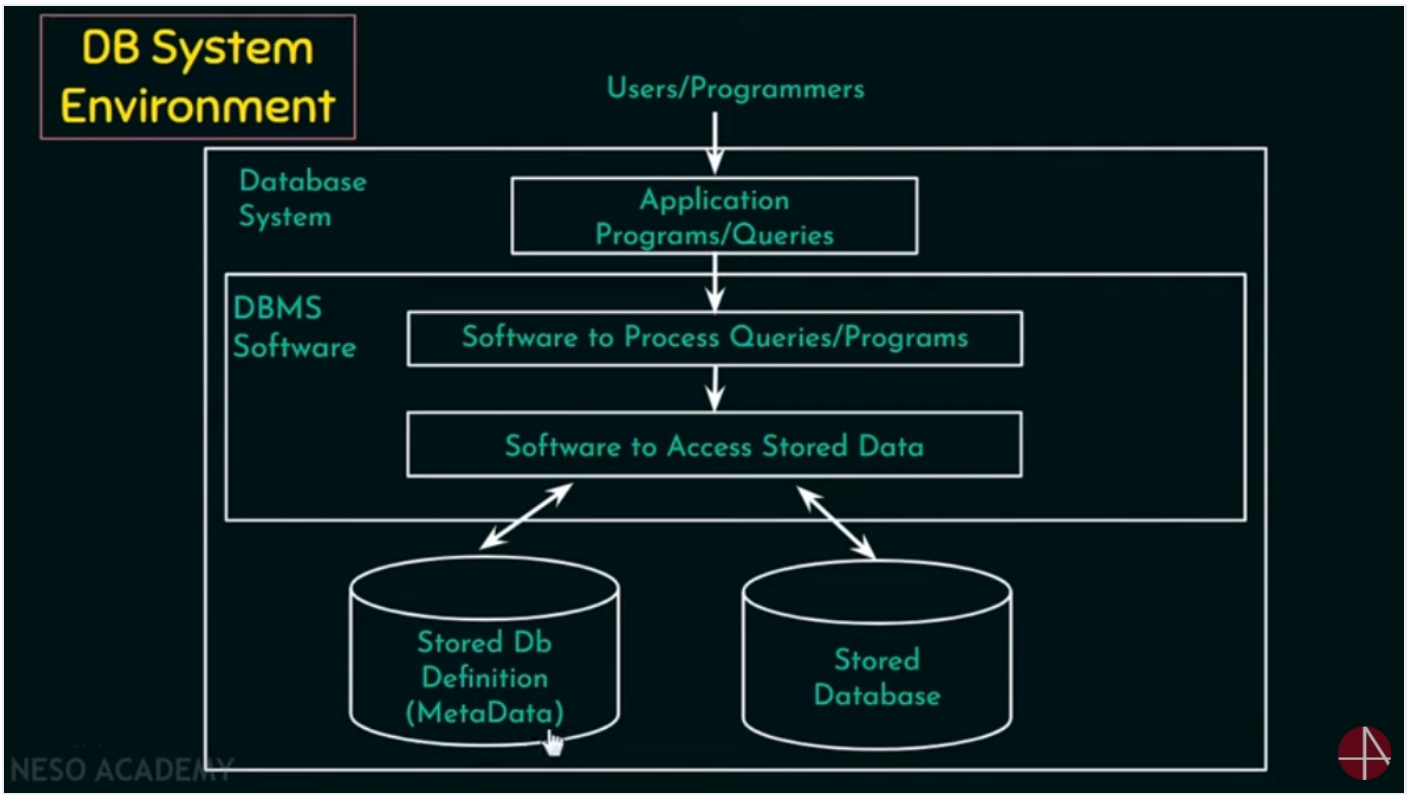
Database Management System

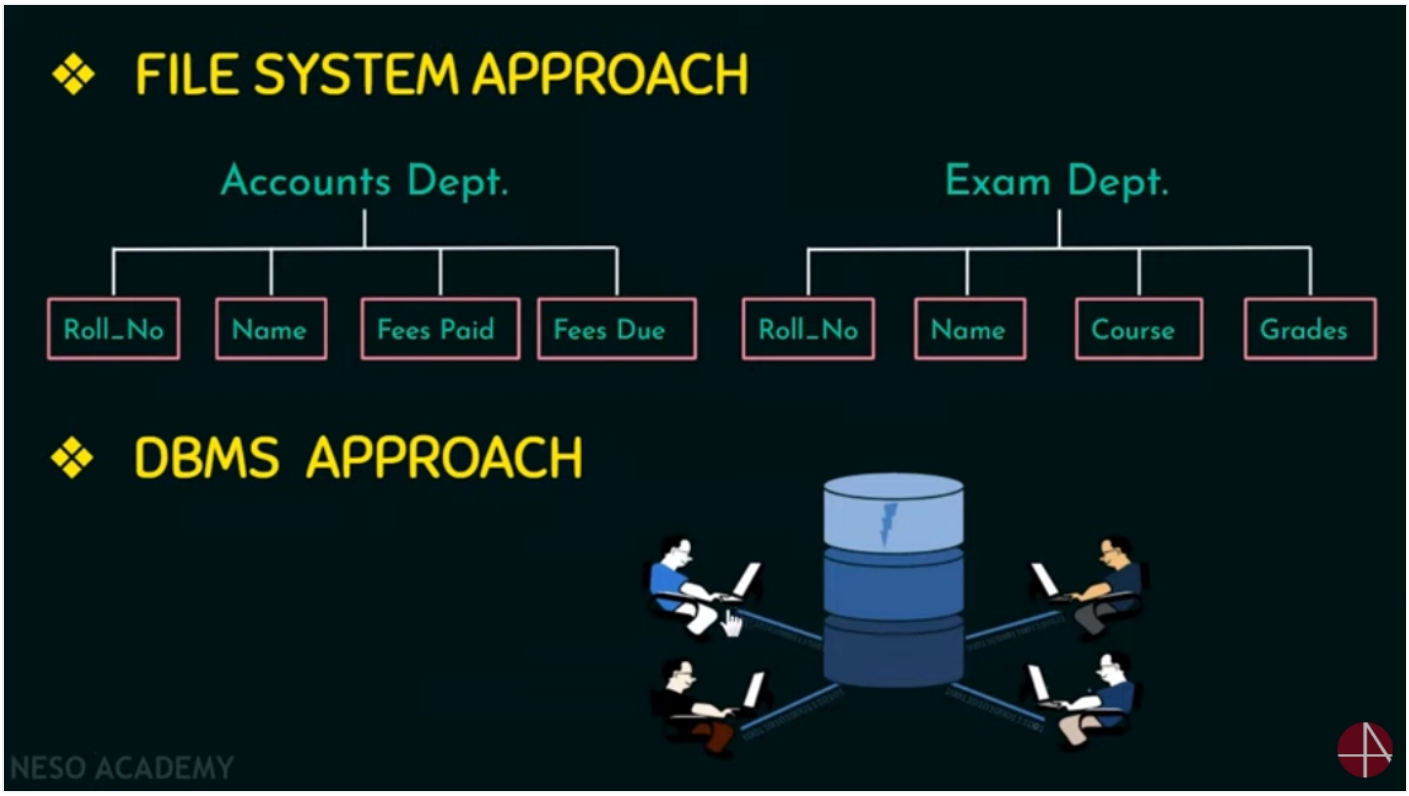
Functionalities:

- **Define:** Specifying the data type, structures and constraints for the data to be stored.
- **Construct:** Process of storing data on some storage medium.
- **Manipulate:** Querying the database to retrieve specific data, updating database and generating reports.
- **Share:** Allows multiple users and programs to access the database concurrently.

NESO ACADEMY







all the data can be store in a single repository and can use it accrdingly.

❖ Characteristics of DBMS Approach

1. Self-Describing Nature of a Database System
2. Insulation between Programs and Data, and Data Abstraction
3. Support of Multiple Views of the Data
4. Sharing of Data and Multiuser Transaction Processing

❖ Self-Describing Nature of a Database System

★ Database system → Database + Meta-data (DB Definition)

↓

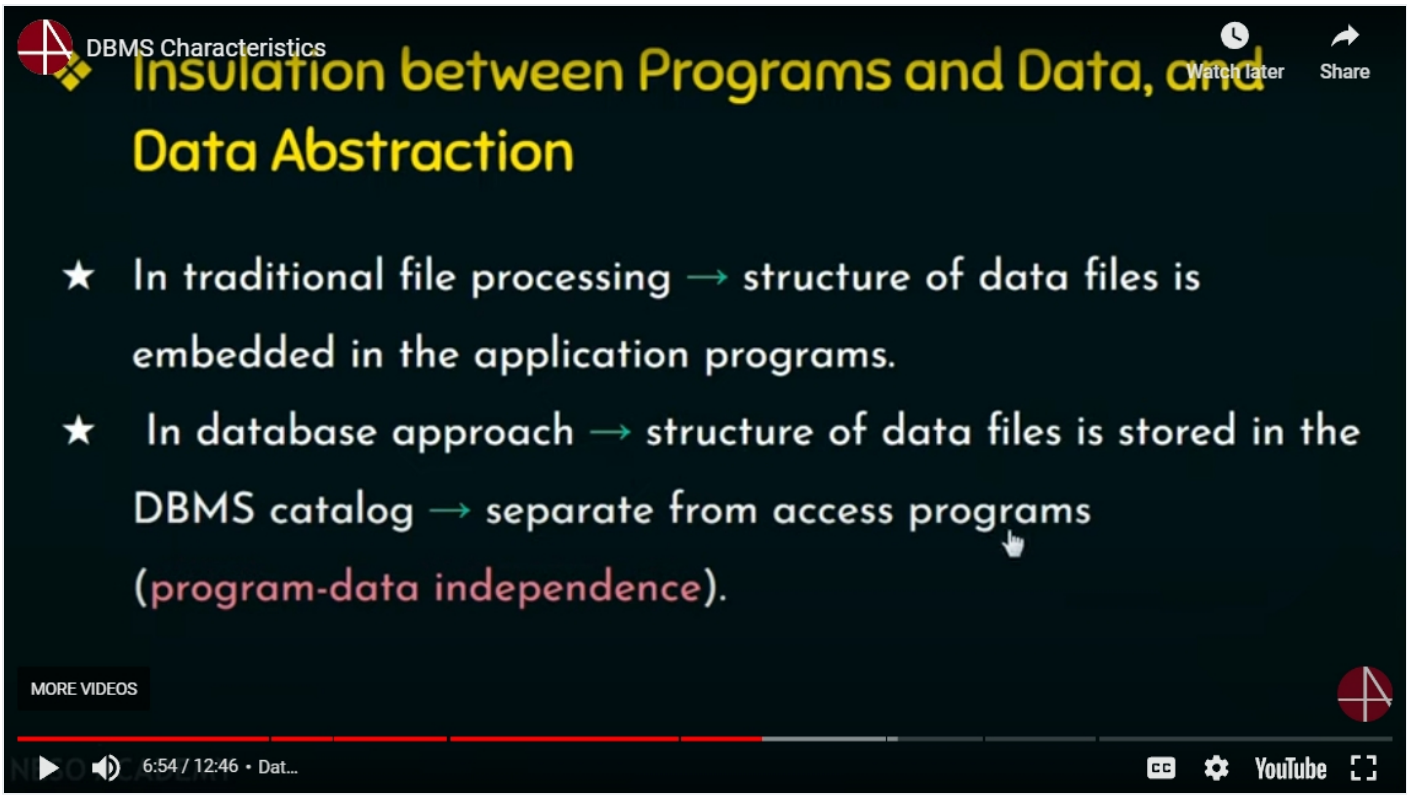
Stored in: DBMS catalog

↓

Used by: DBMS Software & Database Users

★ DBMS software must work equally well with any number of database applications.

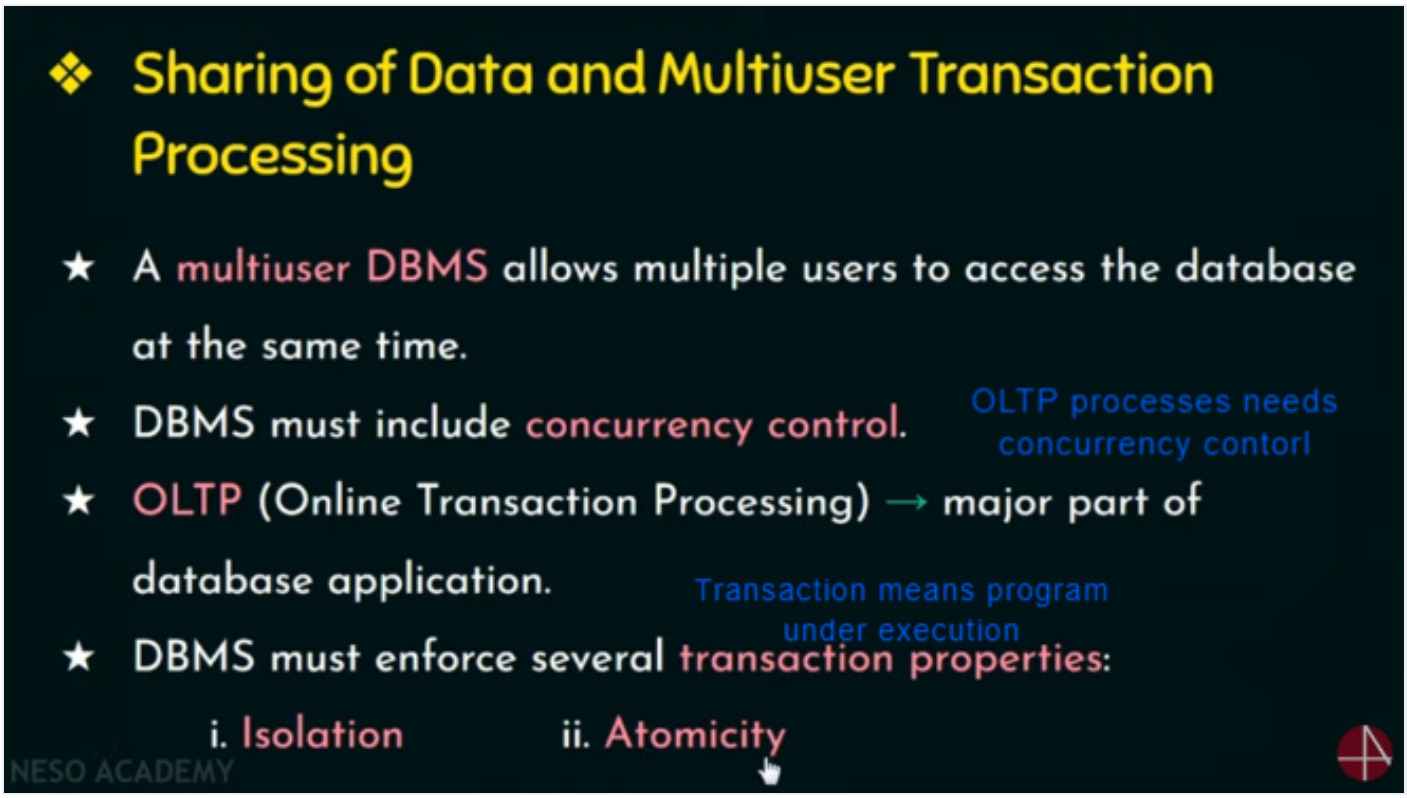
★ In traditional file processing, data definition → part of application programs → work with only one specific DB.



❖ Support of Multiple Views of the Data

★ A database has many users, each of whom may require a different view of database.

★ A view → subset of database → contains virtual data derived from database (not explicitly stored).



Isolation : if two or more users are using shared resources, and if one user changes something from a resource then it should be isloated from other user to get affected.

Atomicity: either the transaction is executed completely or none at all.

❖ Database Users

Actors on the Scene

- ★ Database Administrators
- ★ Database Designers
- ★ End Users
- ★ System Analysts & Application Programmers (Software Engineers)

Workers Behind the Scene

- ★ System designers & implementers
- ★ Tool developers
- ★ Operators & maintenance personnel

❖ ACTORS ON THE SCENE

★ Database Administrators:

- ➔ In database environment, primary resource → database, secondary resource → DBMS & related software.
- ➔ Database Administrator (DBA) responsibilities:
  1) Administering primary/secondary resources.
  2) Authorizing access to the database.
  3) Co-ordinating & monitoring use of database.
  4) Acquiring hardware & software resources as needed.

❖ ACTORS ON THE SCENE

★ Database Designers:

➔ Responsible for:

1) Identifying the data to be stored in the database.
2) Choosing appropriate structures to represent and store data.
3) Communicating with database users → understand their requirements → designs database.

❖ ACTORS ON THE SCENE

★ End Users:

➔ End users → people whose jobs require access to the database → for querying, updating, generating reports.

➔ Several categories of end users:

- Casual end users: Accesses database occasionally → typically middle or high-level managers or other occasional browsers.

ACTORS ON THE SCENE

- End Users:
  - Several categories of end users:
    - Naive or parametric end users: constantly querying and updating database using canned transactions.

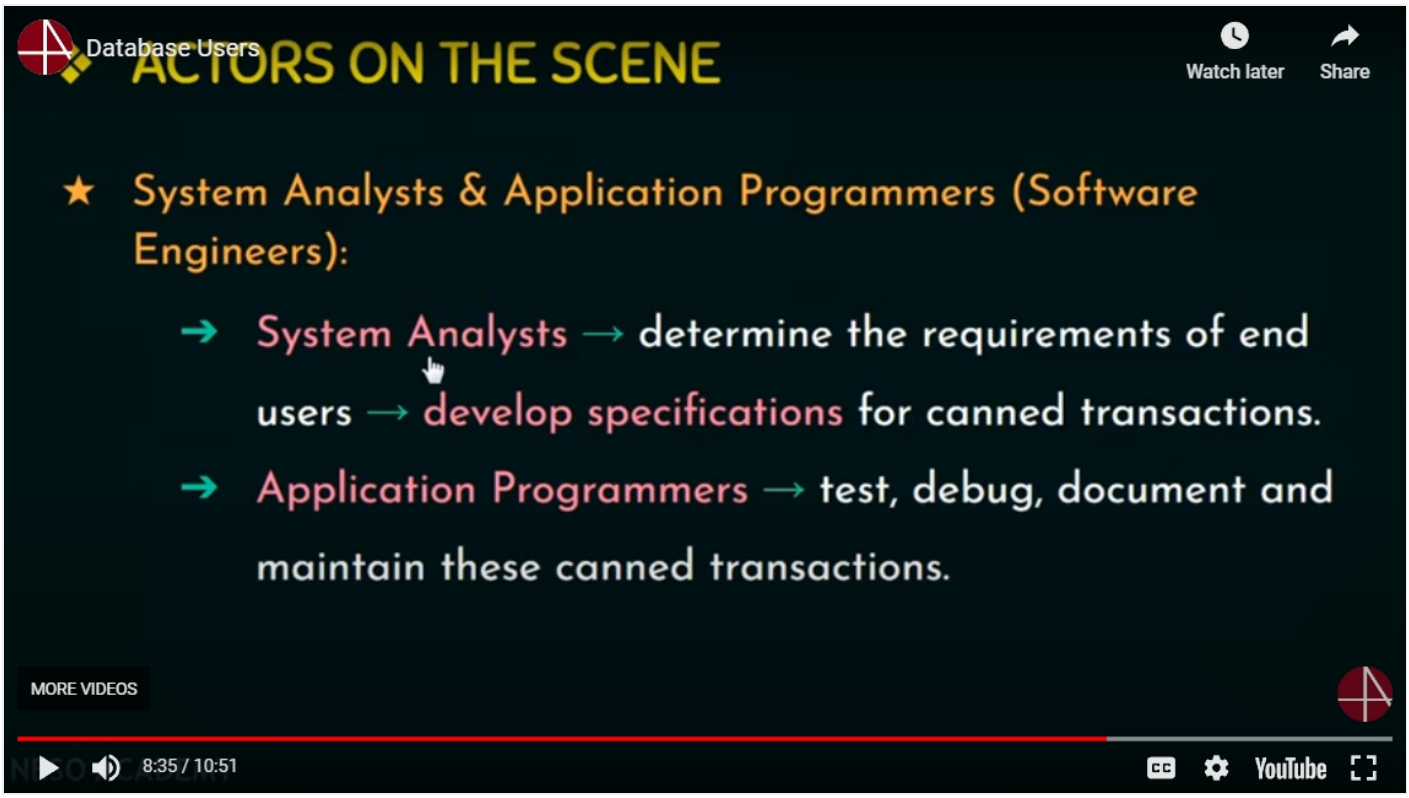
NESO ACADEMY

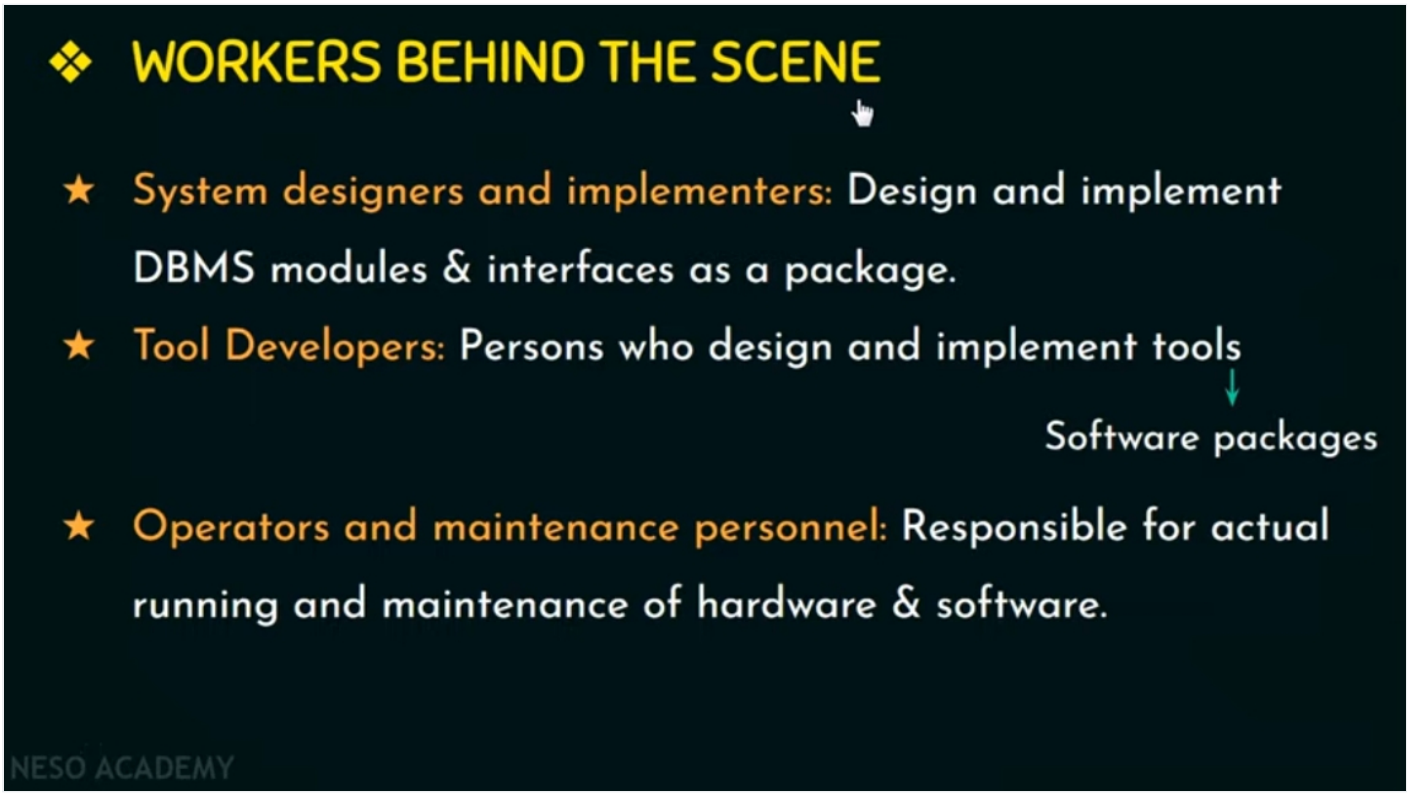
ACTORS ON THE SCENE

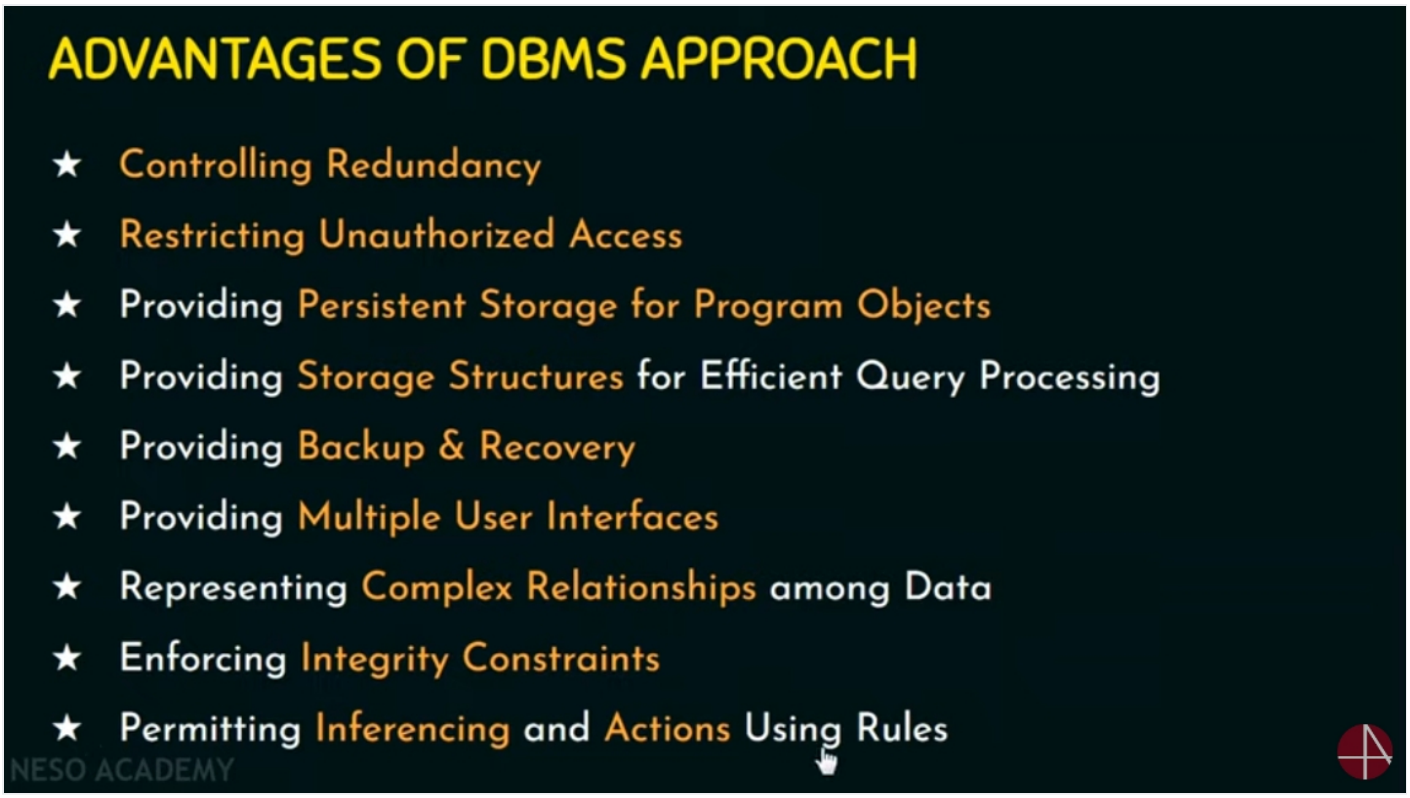
- End Users:
  - Several categories of end users:
    - Naive or parametric end users: constantly querying and updating database using canned transactions.
    - Sophisticated end users: Engineers, scientists, business analysts.
    - Stand-alone users: Maintains personal databases → using ready-made program packages.

NESO ACADEMY

canned transaction : standarrd type of queries.

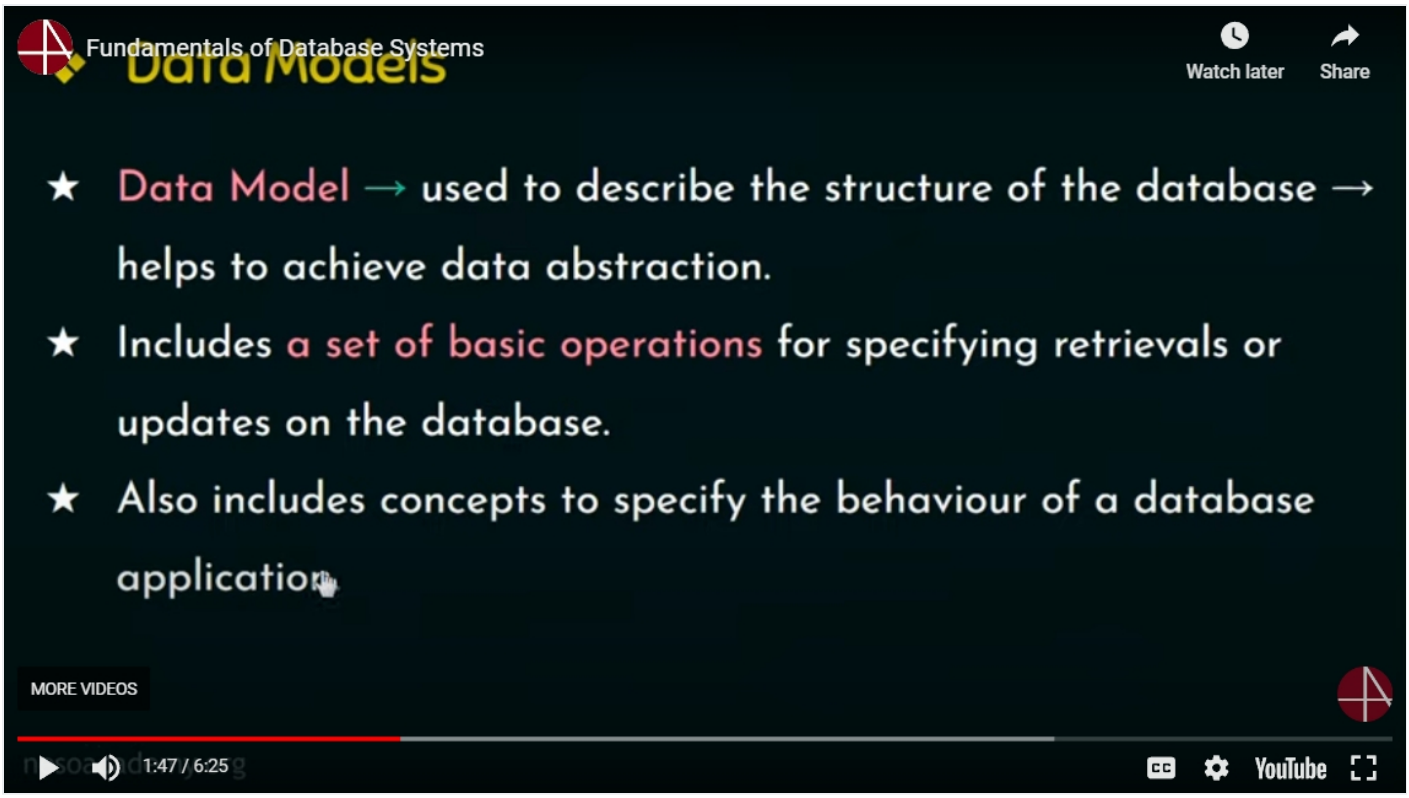

Database Users
ACTORS ON THE SCENE
Watch later
Share
★ System Analysts & Application Programmers (Software Engineers):
➔ System Analysts → determine the requirements of end users → develop specifications for canned transactions.
➔ Application Programmers → test, debug, document and maintain these canned transactions.
MORE VIDEOS
8:35 / 10:51
YouTube

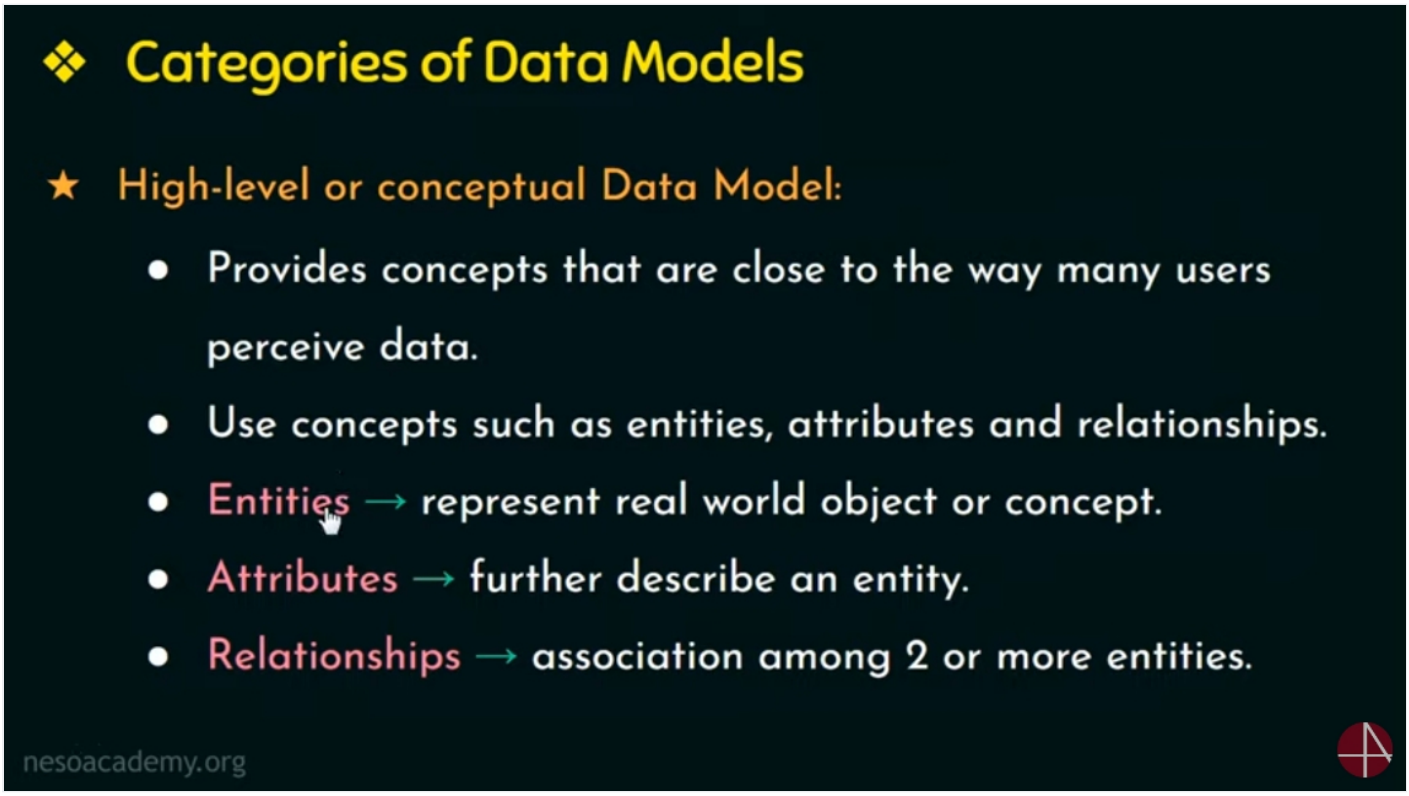

WORKERS BEHIND THE SCENE
★ System designers and implementers: Design and implement DBMS modules & interfaces as a package.
★ Tool Developers: Persons who design and implement tools
Software packages
★ Operators and maintenance personnel: Responsible for actual running and maintenance of hardware & software.
NESO ACADEMY


ADVANTAGES OF DBMS APPROACH
★ Controlling Redundancy
★ Restricting Unauthorized Access
★ Providing Persistent Storage for Program Objects
★ Providing Storage Structures for Efficient Query Processing
★ Providing Backup & Recovery
★ Providing Multiple User Interfaces
★ Representing Complex Relationships among Data
★ Enforcing Integrity Constraints
★ Permitting Inferencing and Actions Using Rules
NESO ACADEMY


❖ WHEN NOT TO USE DBMS
★ Overhead costs of using DBMS:
→ High initial investment
→ Overhead for providing security, concurrency control, recovery
★ Database & applications → simple, well defined and no changes expected.
★ Multiple-user access → not required.
NESO ACADEMY

Data models???


Fundamentals of Database Systems
Watch later
Share
Data Models
★ Data Model → used to describe the structure of the database → helps to achieve data abstraction.
★ Includes a set of basic operations for specifying retrievals or updates on the database.
★ Also includes concepts to specify the behaviour of a database application.
MORE VIDEOS
1:47 / 6:25
YouTube


Categories of Data Models
★ High-level or conceptual Data Model:
• Provides concepts that are close to the way many users perceive data.
• Use concepts such as entities, attributes and relationships.
• Entities → represent real world object or concept.
• Attributes → further describe an entity.
• Relationships → association among 2 or more entities.
nesoacademy.org

❖ Categories of Data Models

★ Low-level or Physical Data Model:

- Describes how data is stored in the computer.
- Access path → structure for efficient searching of database records.

★ Representational (or implementation) Data Model:

- Represent data using record structures → record-based data models.

nesoacademy.org

❖ Terminologies

★ Database Schema: Description of a database.

★ Schema Diagram: Displayed schema.

STUDENT

NAME	ROLL No.	BRANCH	ADDRESS

★ Schema Construct: Each object within the schema.

Ex: STUDENT, COURSE, etc.

★ Database State (or instance or snapshot): The data in the database at a particular moment.

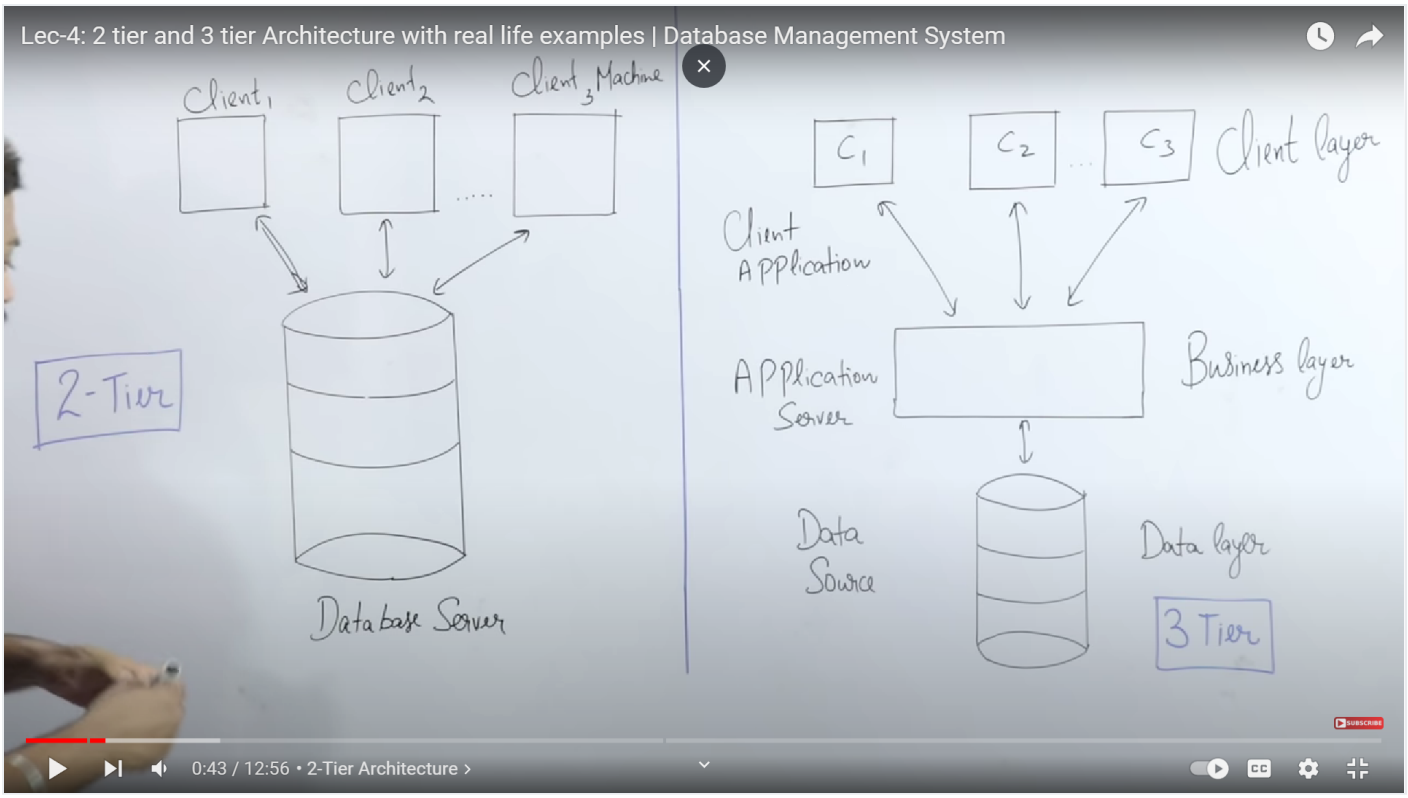
nesoacademy.org

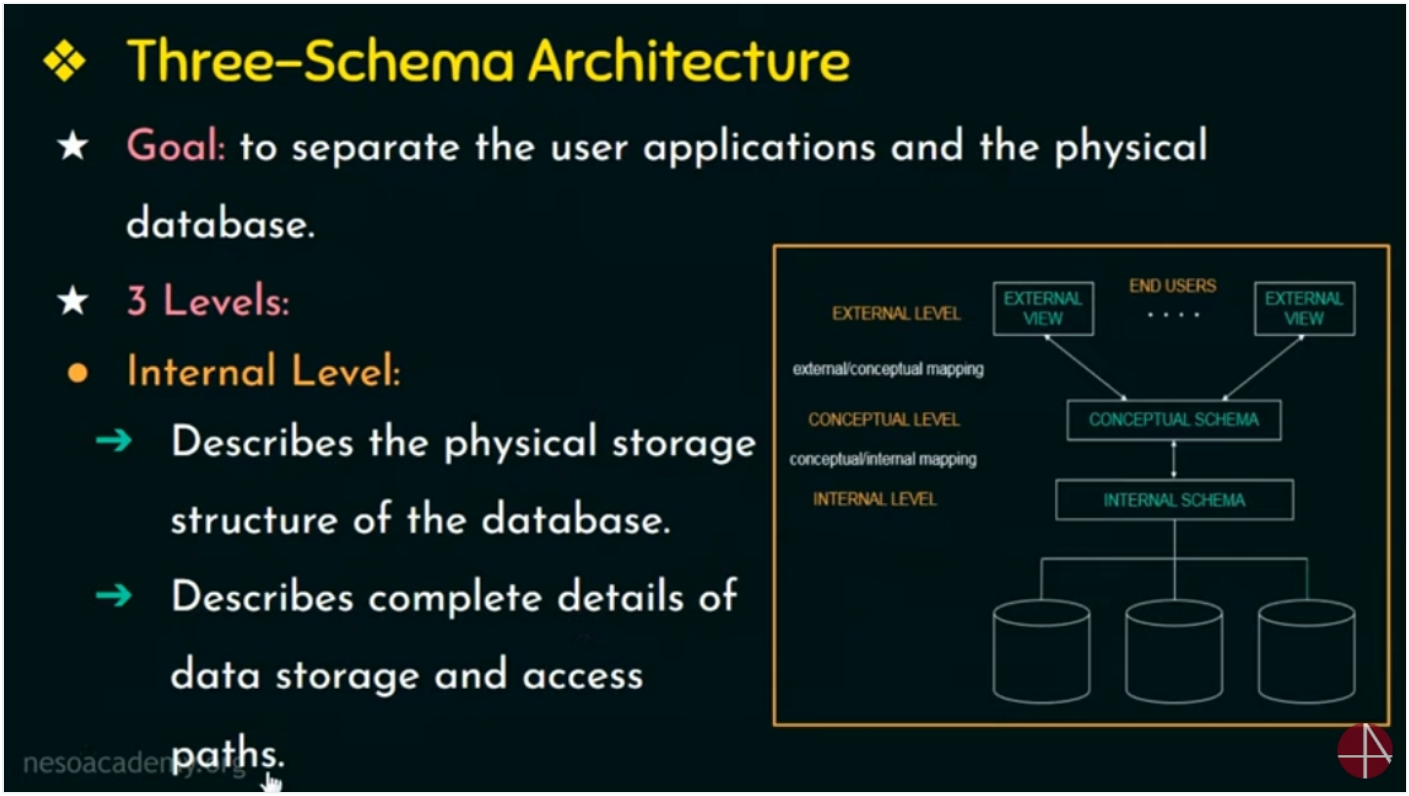
3 tier architecture

→ user

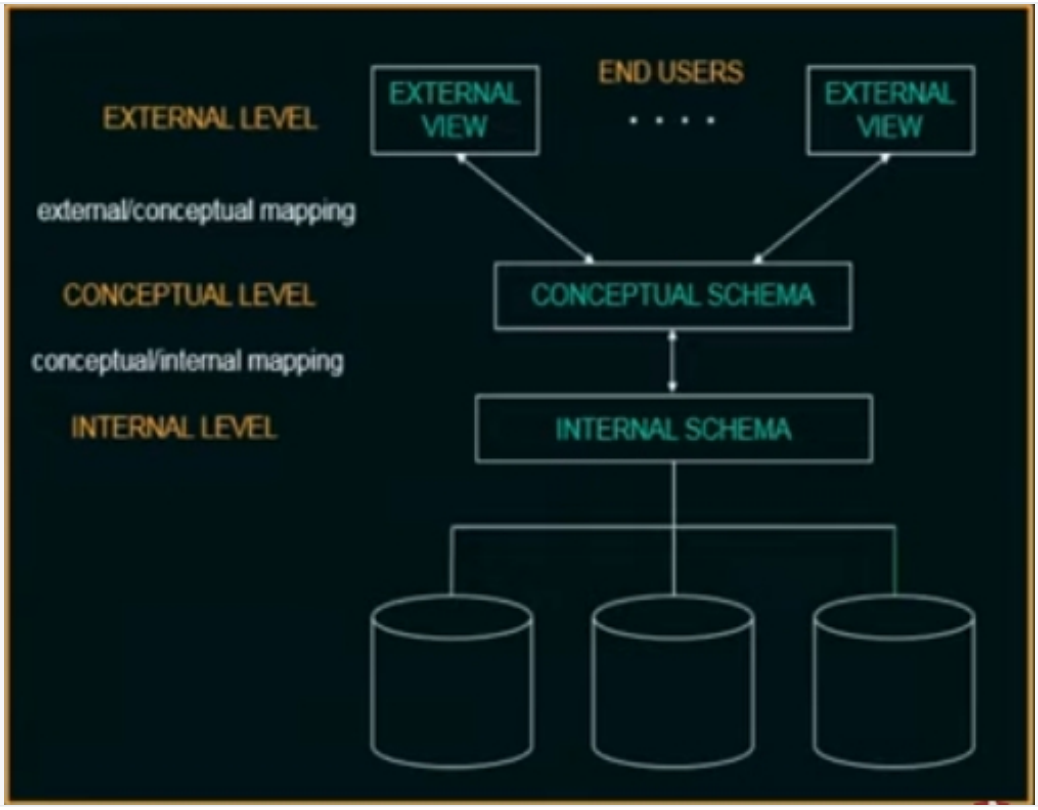
→ bussiness

→ database





access path - effecient path for searching a record.

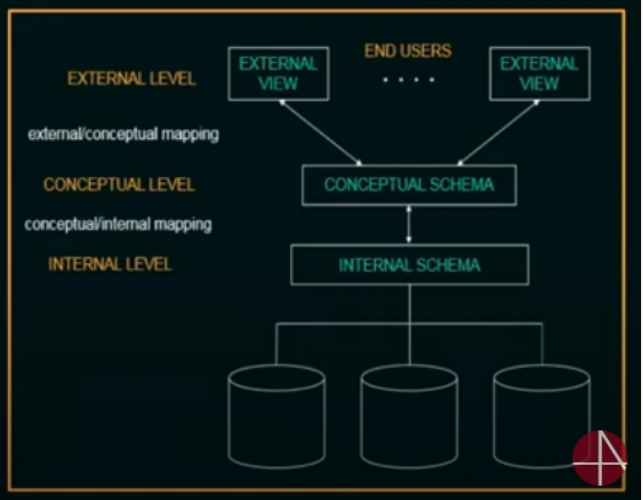


Three-Schema Architecture & Data Independence

Watch later Share

Three-Schema Architecture

- Conceptual Level:
  - → Hides the details of the physical storage structure and concentrates on describing entities, data types, relationships, constraints, etc.



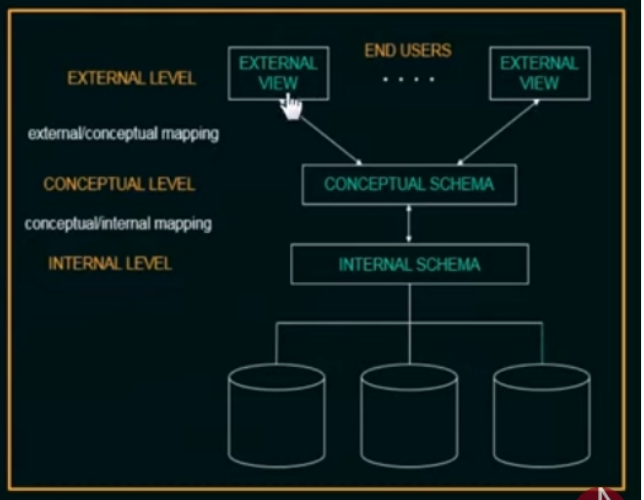
MORE VIDEOS

2:01 / 6:49 • Co...

YouTube

❖ Three-Schema Architecture

- External Level:
  - ➔ Describes the part of the database that a user is interested in and hides the rest of the database from the user group.

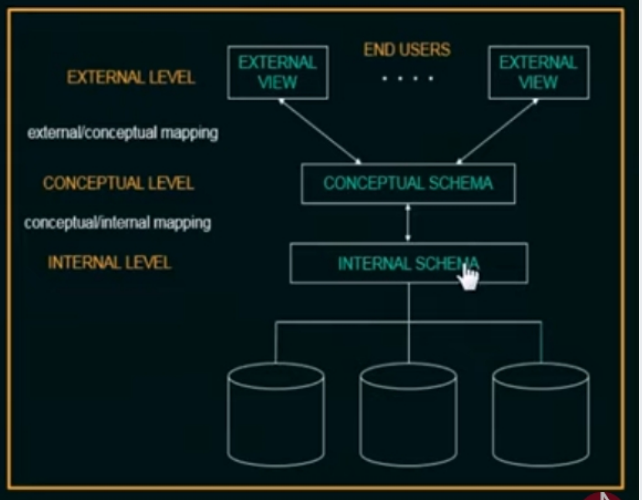


nesoacademy.org

mapping - requesting n getting results.

❖ Data Independence

📌 Definition: Capacity to change the schema at one level of a database system without having to change the schema at the next higher level.



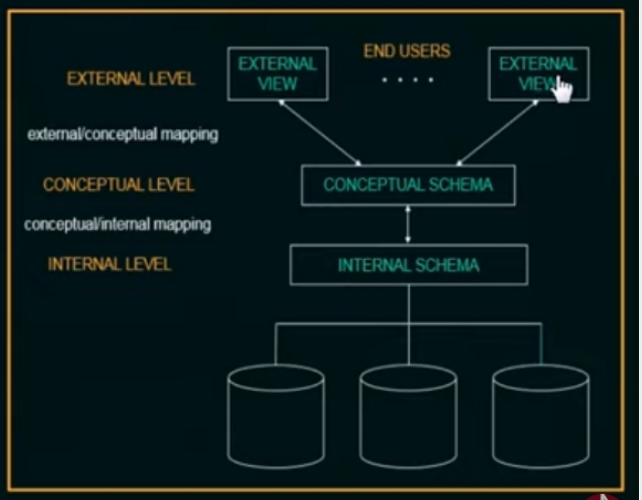
nesoacademy.org

types:

❖ Data Independence

★ Logical Data Independence:

- Ability to modify the conceptual schema without changing the external schemas or application programs.

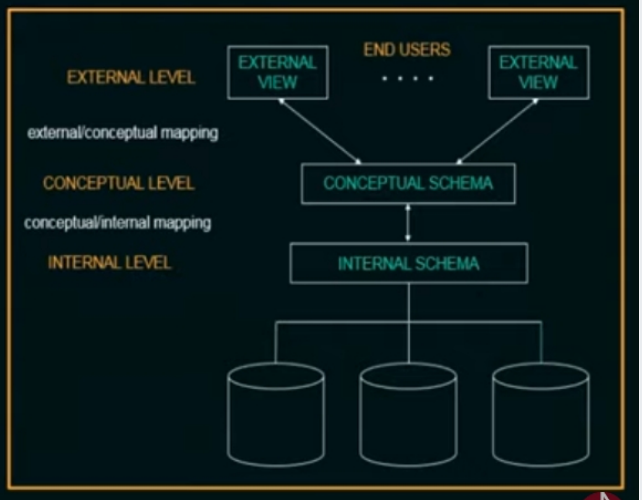


nesoacademy.org

❖ Data Independence

★ Physical Data Independence:

- Ability to modify the internal schema without changing the conceptual schema.
- Changes may be needed → to improve performance.

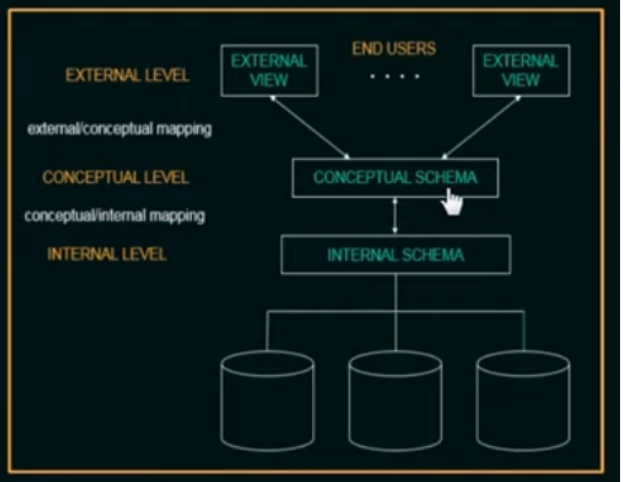


nesoacademy.org

like - chinging the path of a record.

❖ DBMS Languages

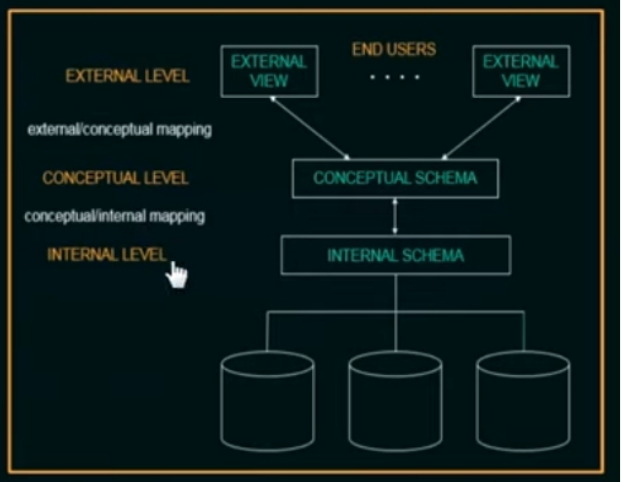
★ In DBMS, where no strict separation of levels is maintained, Data Definition Language (DDL) is used to define the internal and conceptual schemas .



❖ DBMS Languages

★ In DBMS, where a clear separation is maintained between conceptual and internal levels:

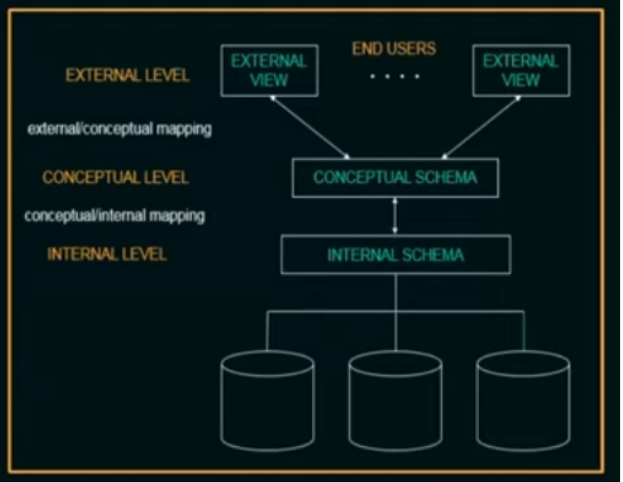
- DDL → used to specify the conceptual schema only.
- Storage Definition Language (SDL) → used to specify the internal schema.



❖ DBMS Languages

★ View Definition Language (VDL) → to specify user views and their mappings to the conceptual schema.

★ Data Manipulation Language (DML) → for manipulation of data in the database.



❖ DBMS Languages

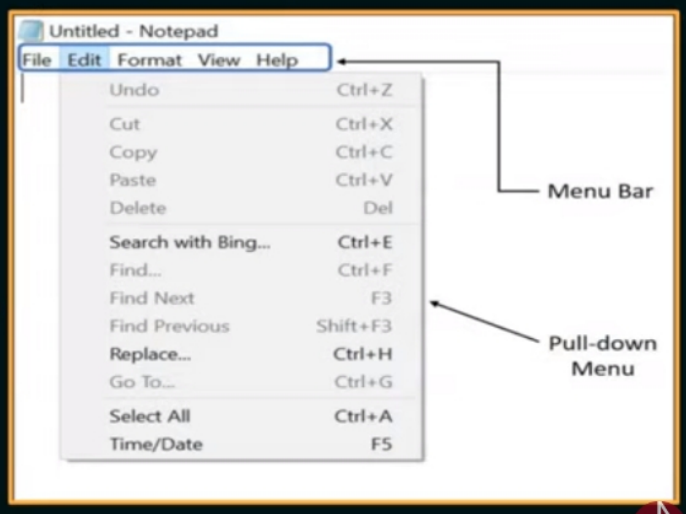
★ Types of DMLs:

- High-level (or non-procedural DML):
  - ➔ Used to specify complex database operations in a concise manner.
- Low-level (or procedural DML):
  - ➔ Embedded in a general purpose programming language.

❖ DBMS Interfaces

★ Menu-Based Interfaces:

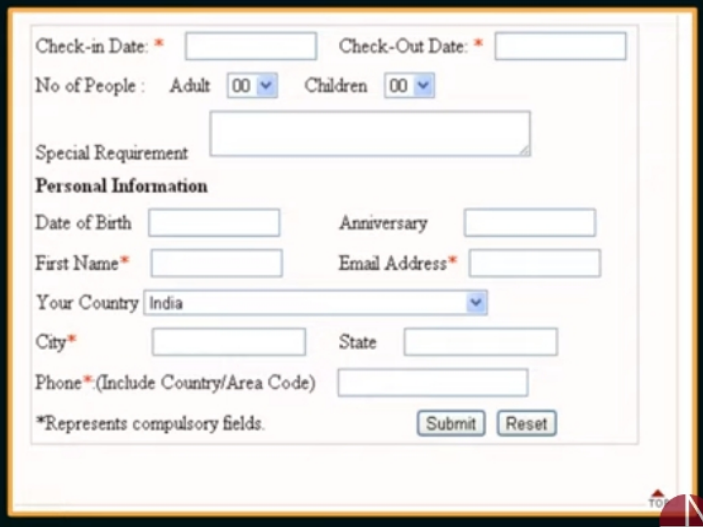
- These interfaces present the users with a list of options (menus).
- Most popular → Pull-down menus.

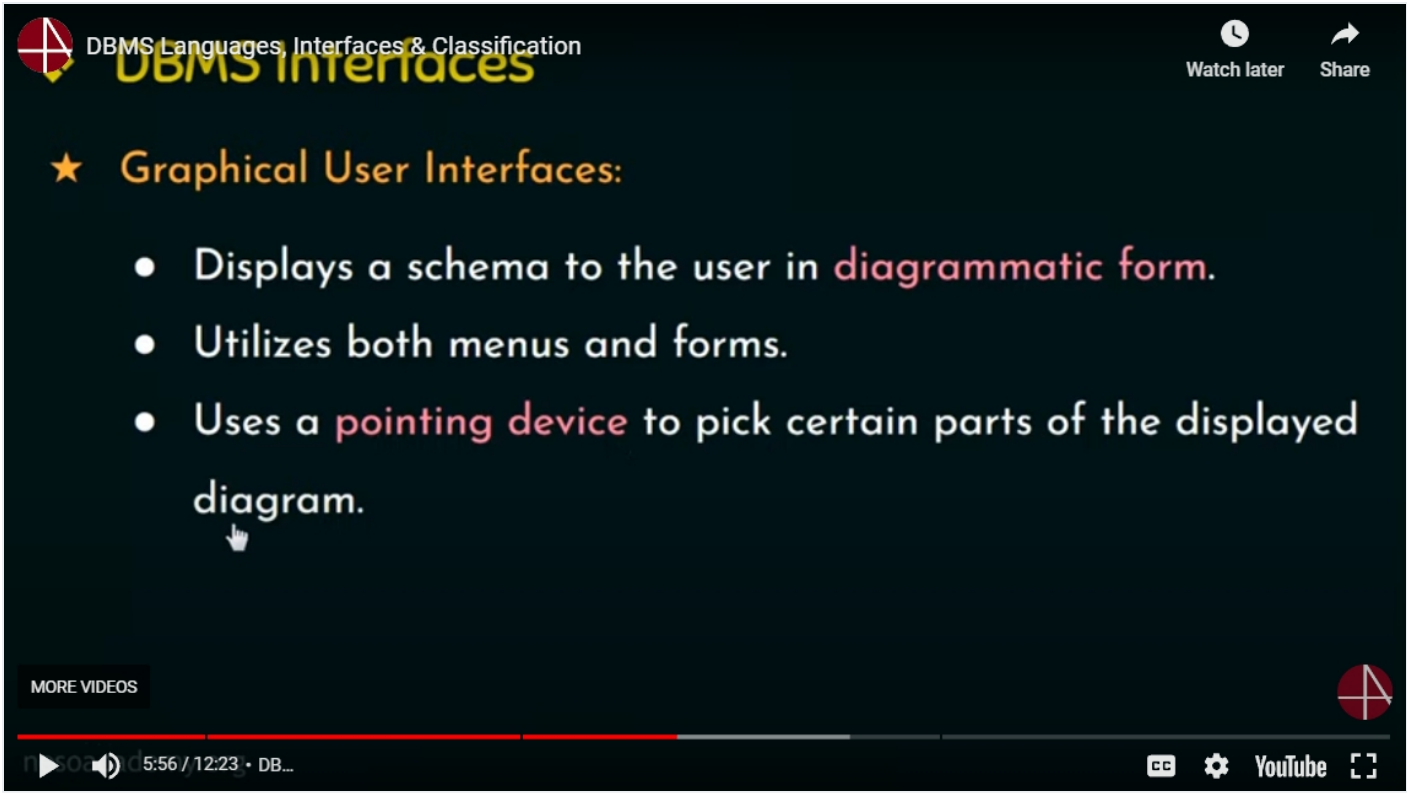


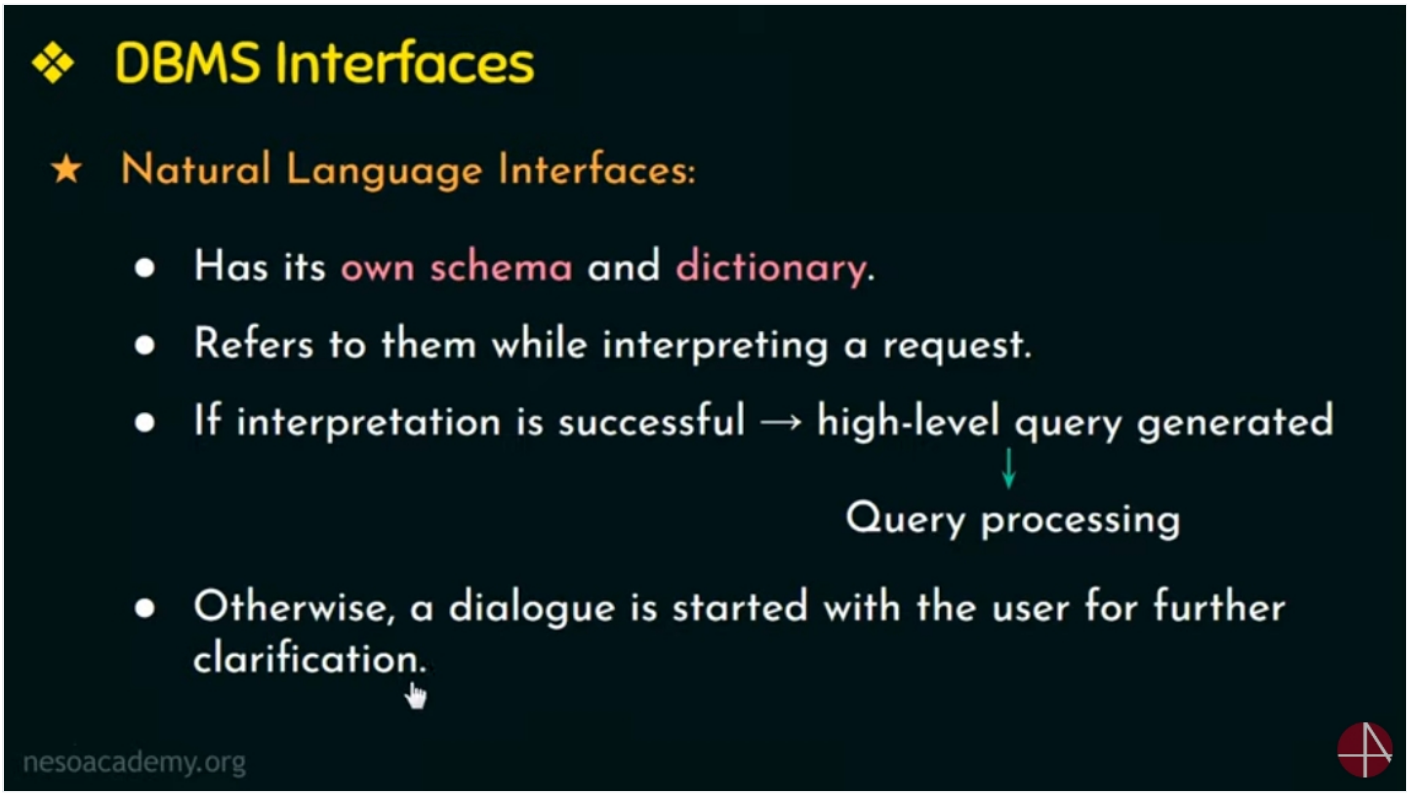
❖ DBMS Interfaces

★ Forms-Based Interfaces:

- Displays a form to each user.
- Designed and programmed for the naive users.




DBMS Languages, Interfaces & Classification
Watch later
Share
DBMS Interfaces
★ Graphical User Interfaces:
• Displays a schema to the user in diagrammatic form.
• Utilizes both menus and forms.
• Uses a pointing device to pick certain parts of the displayed diagram.
MORE VIDEOS
5:56 / 12:23 • DB...
YouTube


DBMS Interfaces
★ Natural Language Interfaces:
• Has its own schema and dictionary.
• Refers to them while interpreting a request.
• If interpretation is successful → high-level query generated
Query processing
• Otherwise, a dialogue is started with the user for further clarification.
nesoacademy.org

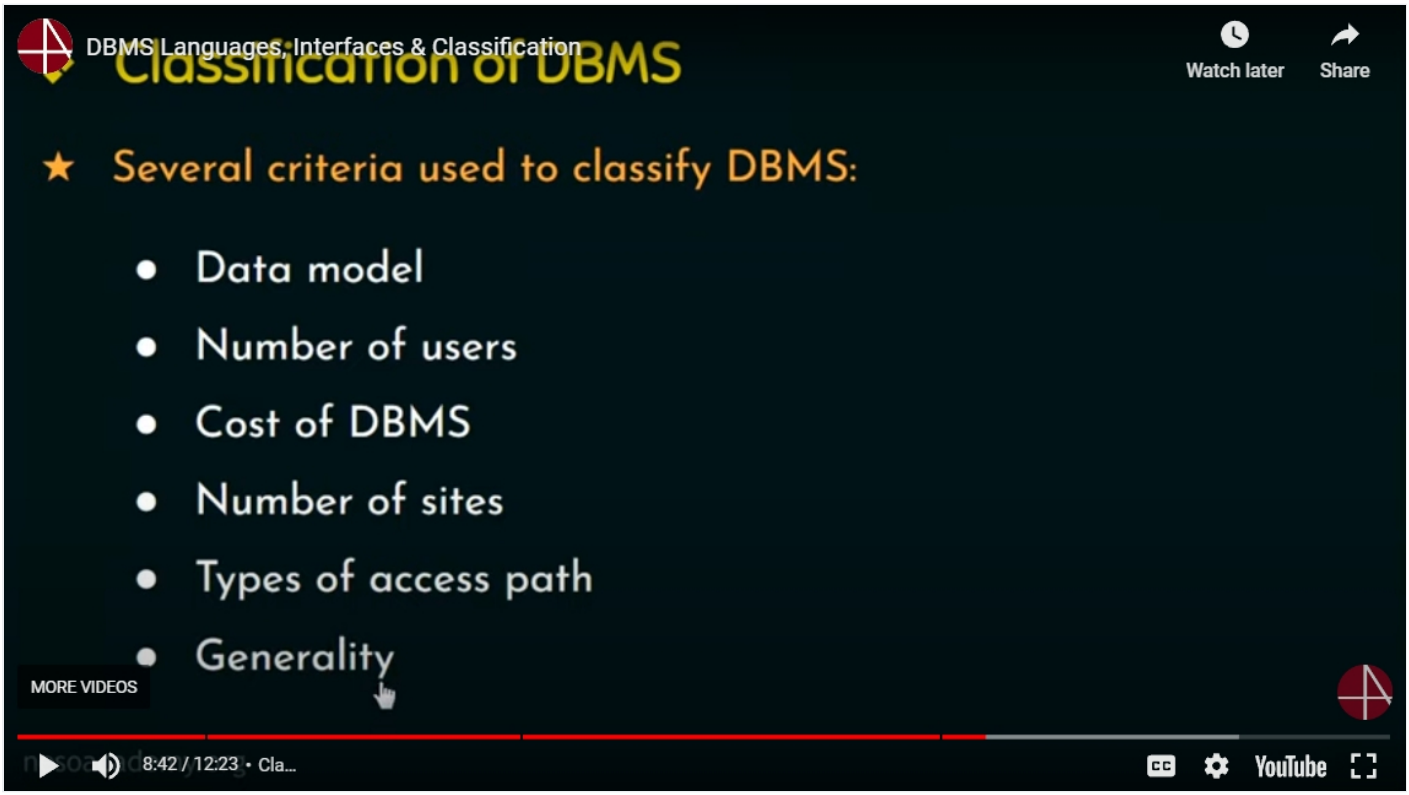
DBMS Interfaces

★ Interfaces for Parametric Users:

- Parametric users → have small set of operations that they must perform repeatedly.
- System analysts and programmers → design & implement a special interface for these users.

★ Interfaces for the DBA:

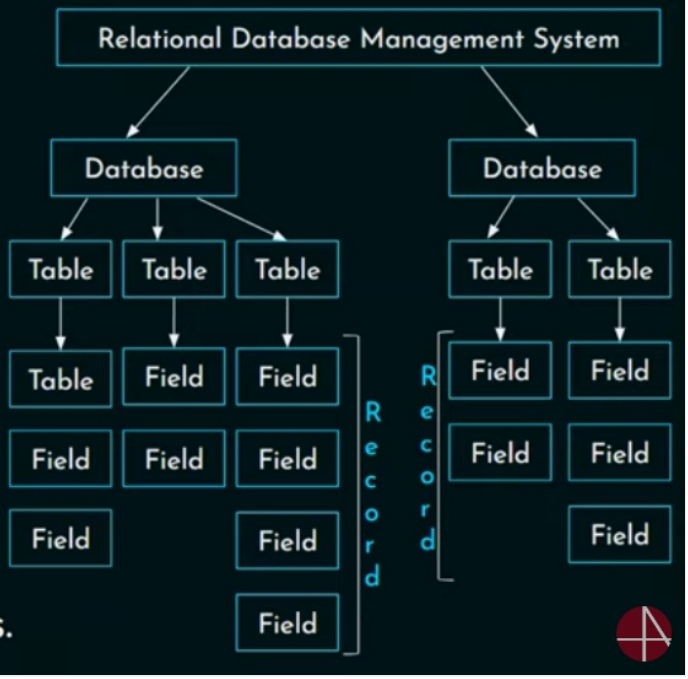
- Privileged commands (like for creating accounts, granting access) that can be used only by the DBA's staff.



❖ Classification of DBMS

★ Data Models:

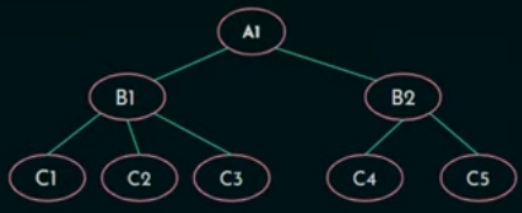
- Relational data model: represents database as a collection of tables.
- Object-oriented database model: defines database in terms of objects, their properties & their operations.

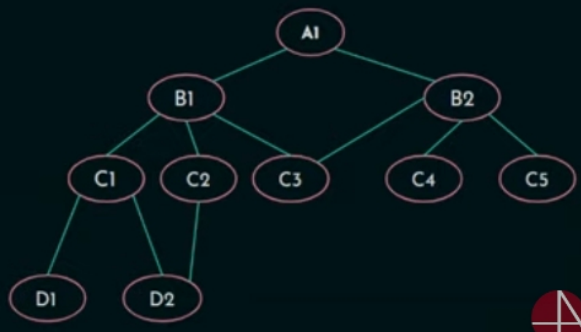


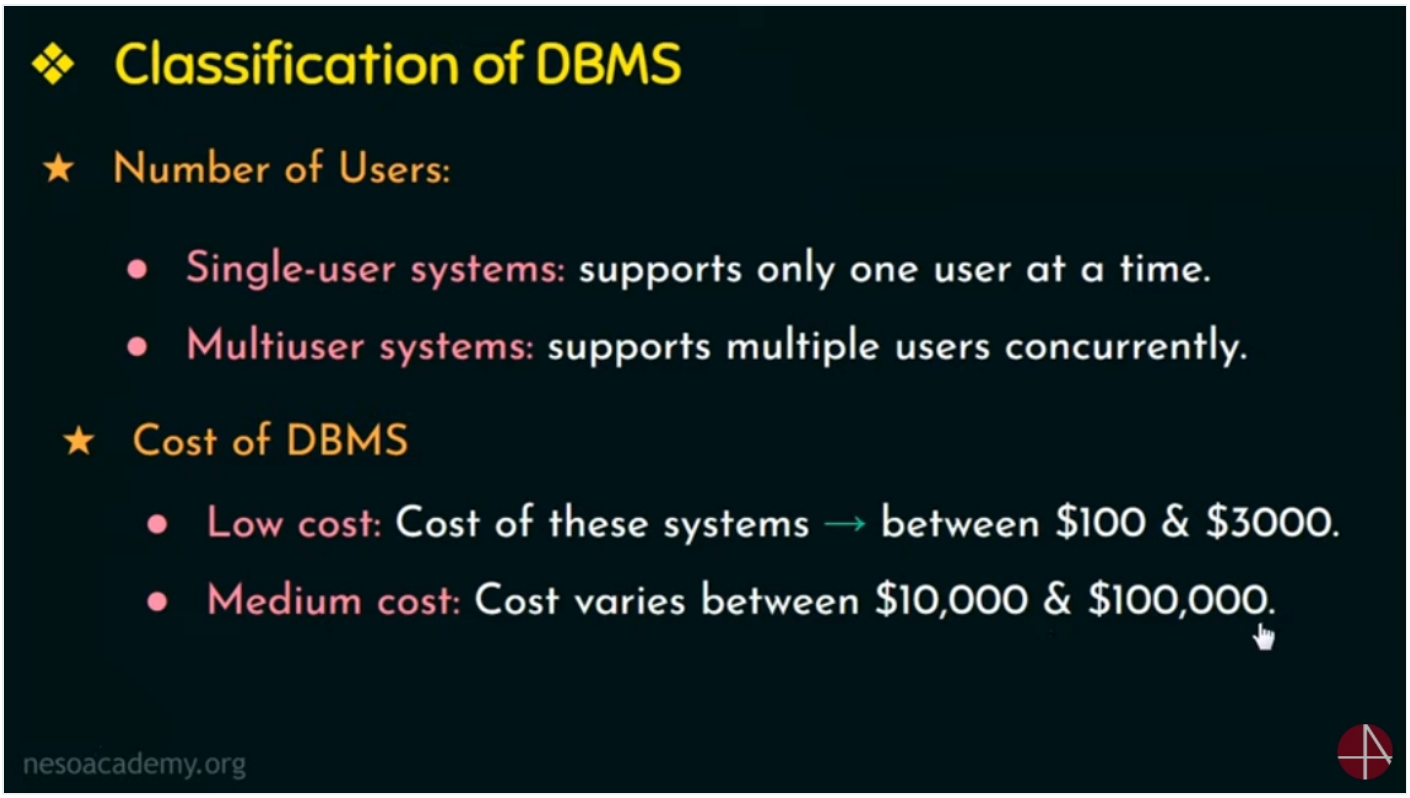
❖ Classification of DBMS

★ Data Models:

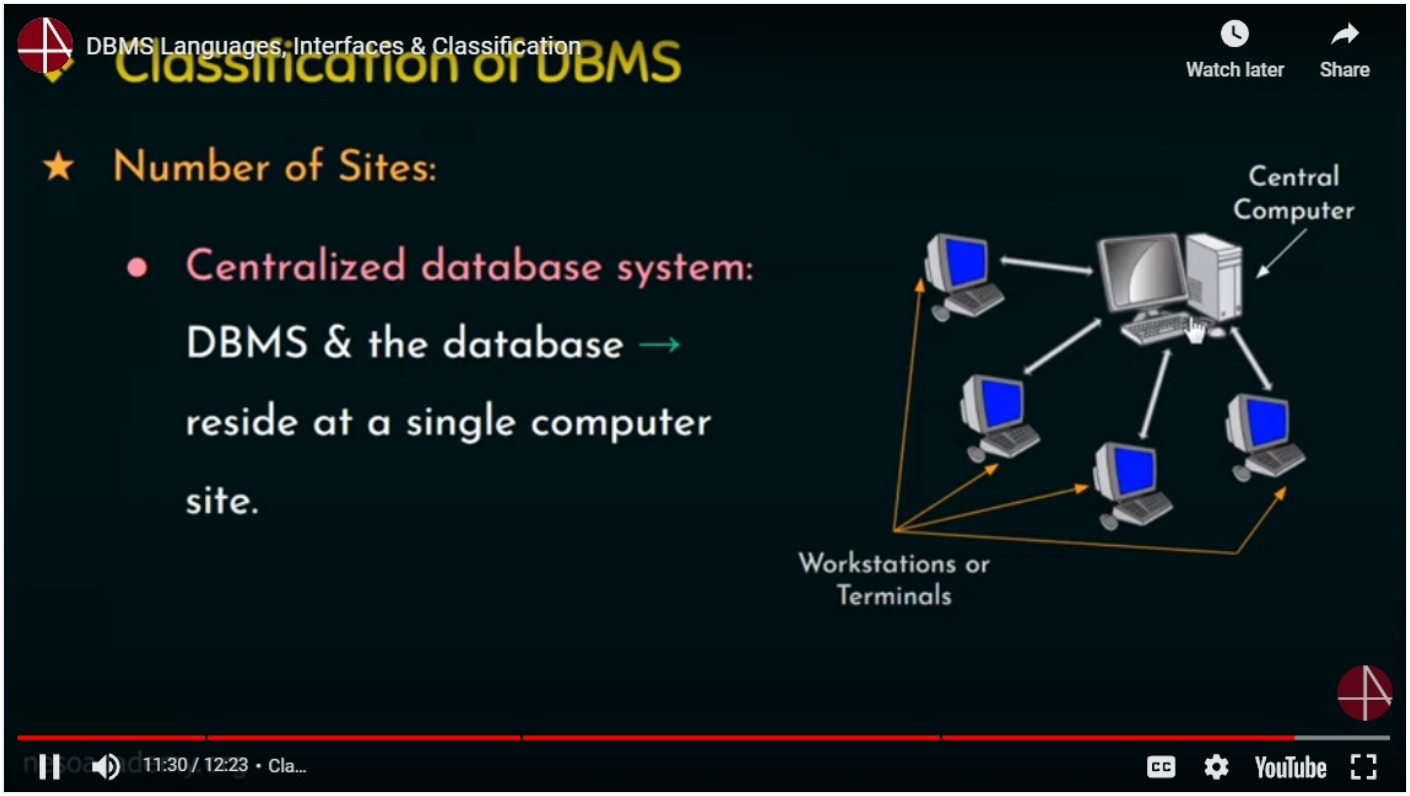
- Hierarchical model: represents data as tree structures.
- Network model: represents objects and their relationships in the form of graph.







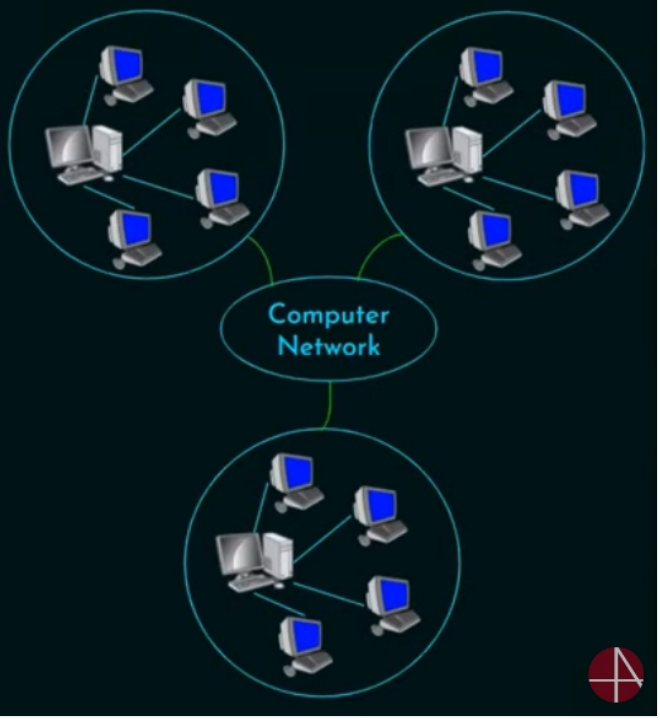
high cost also there.



Classification of DBMS

★ Number of Sites:

- Distributed database system: DBMS & the database → distributed over many sites, connected by a computer network.



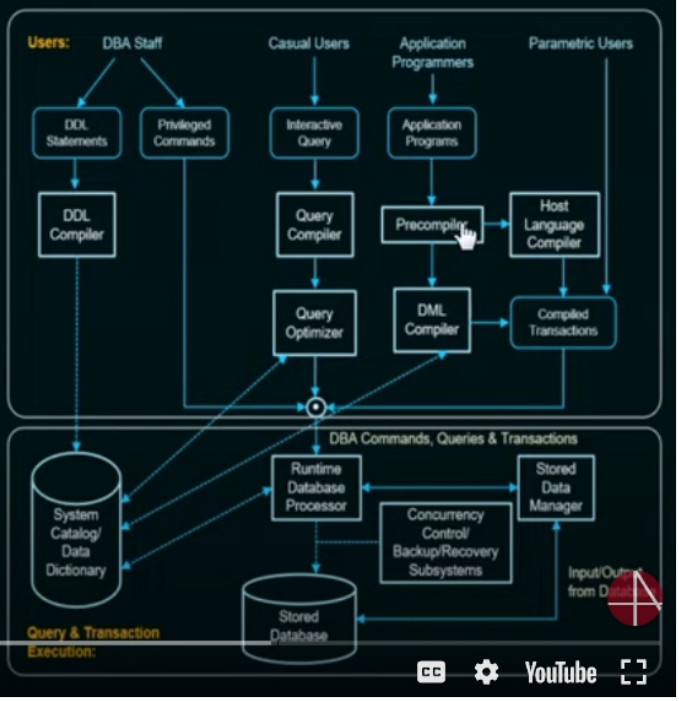
nesoacademy.org

Database System Environment

Watch later Share

DBMS Component Modules

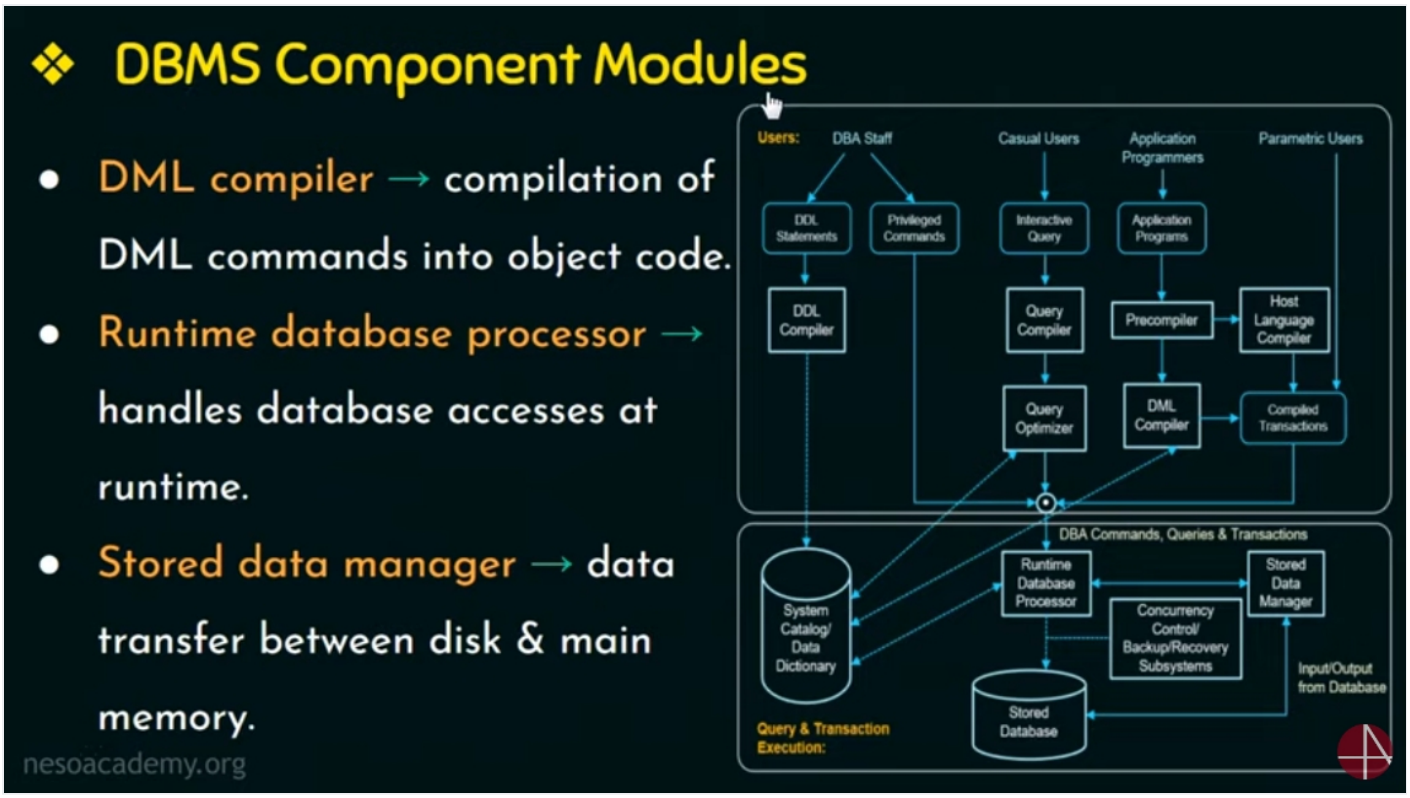
- DDL compiler → processes schema definition → stores it in the catalog.
- Query compiler → handles high-level queries.
- Pre-compiler → extracts DML commands from an application program.

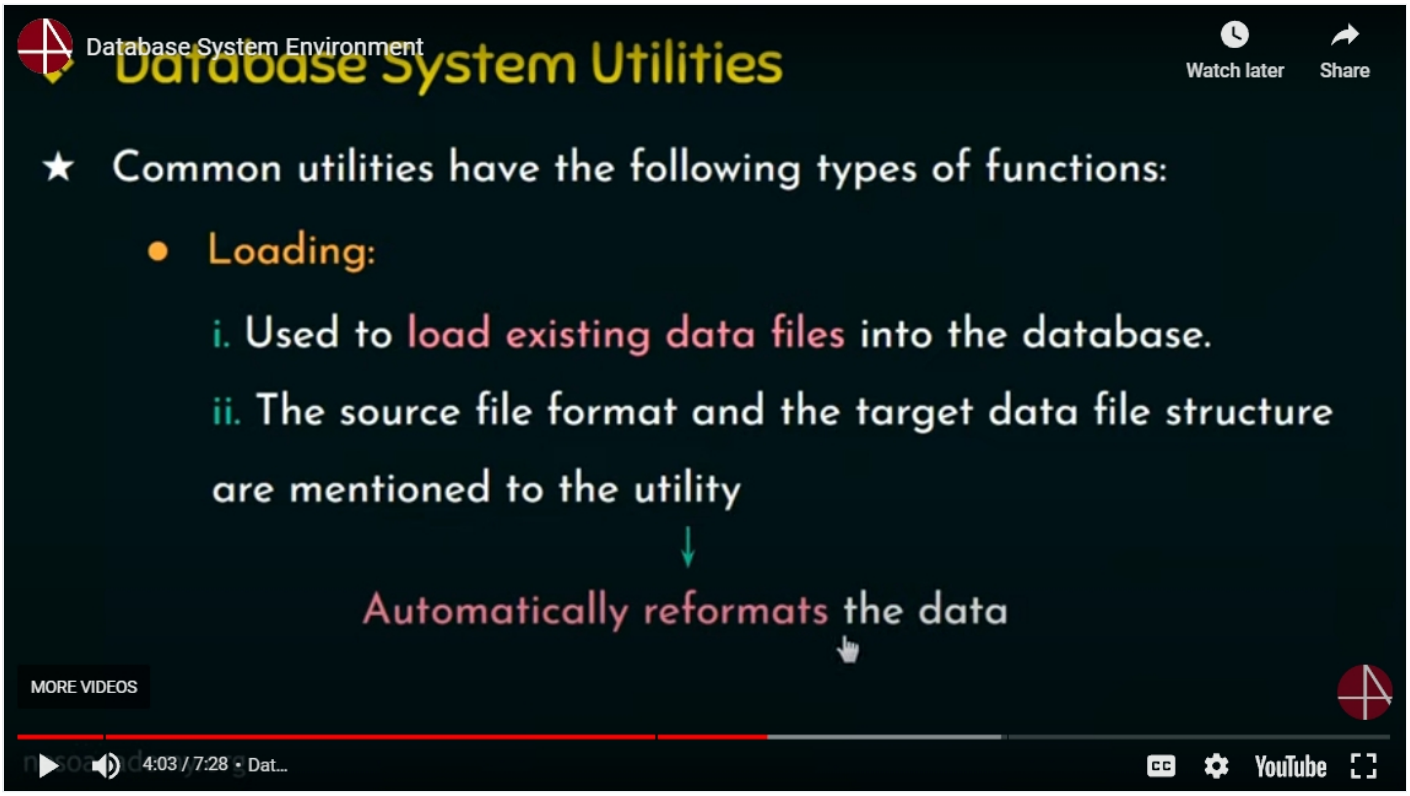


MORE VIDEOS

1:47 / 7:28 • Typ...

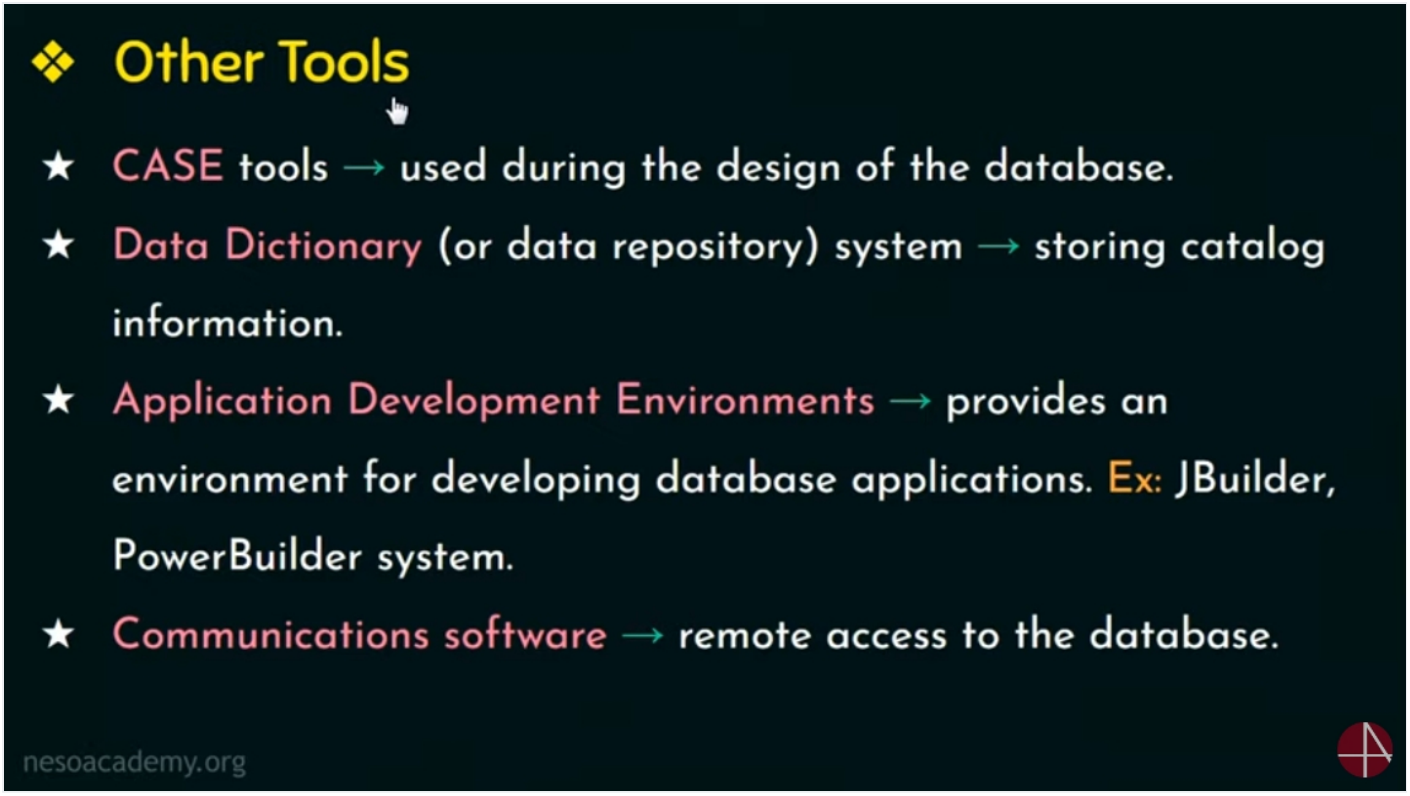
YouTube





other utilities:

backup, file reorganization, performance monitoring.



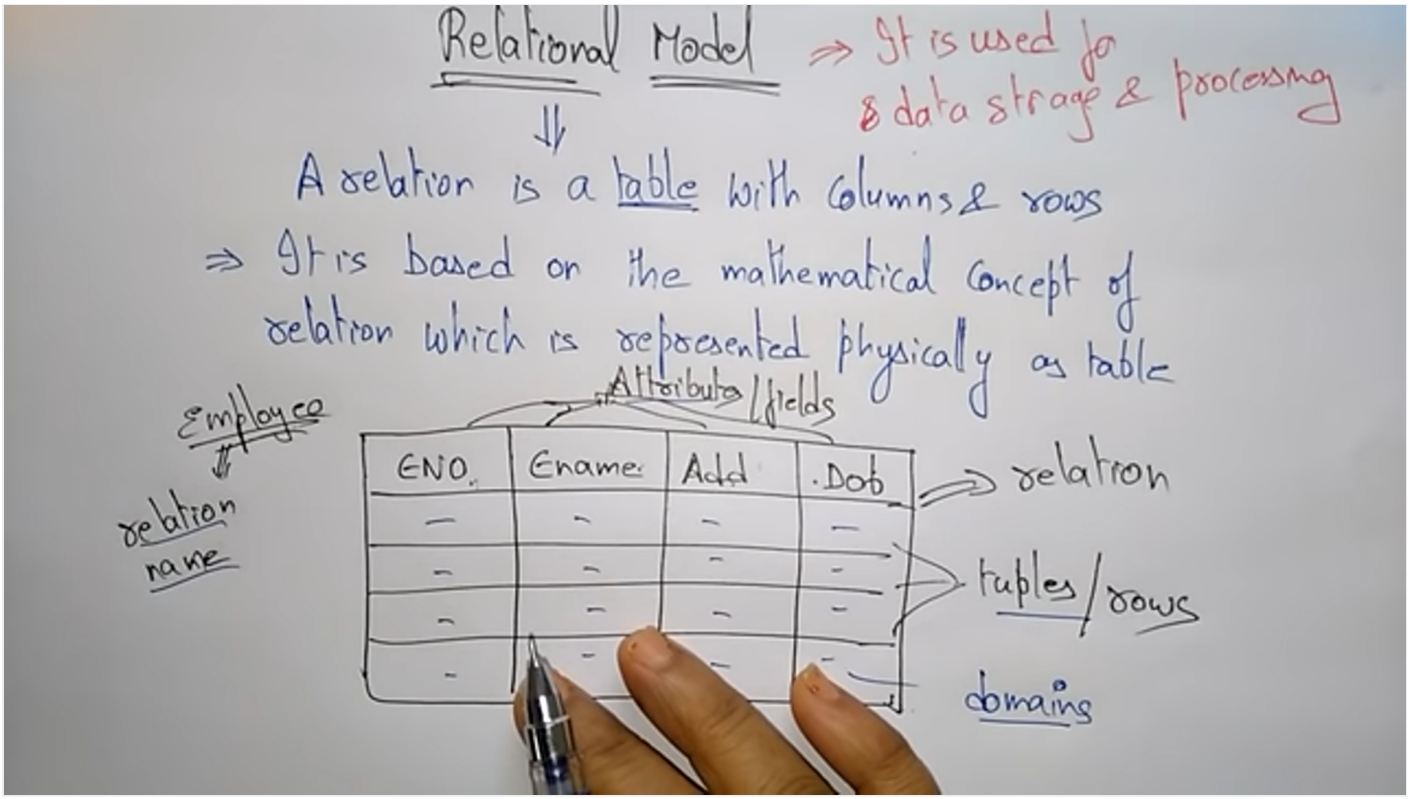
.....

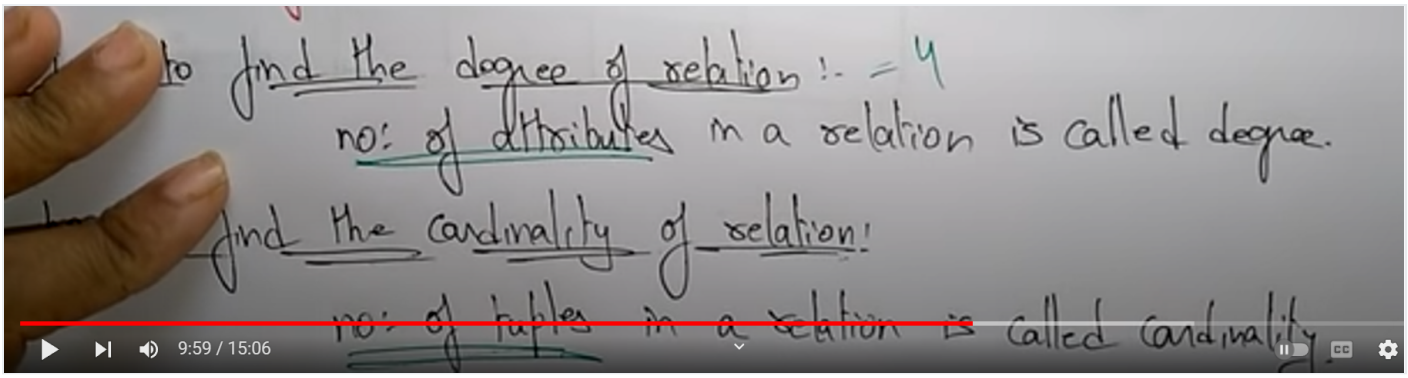
Relational model

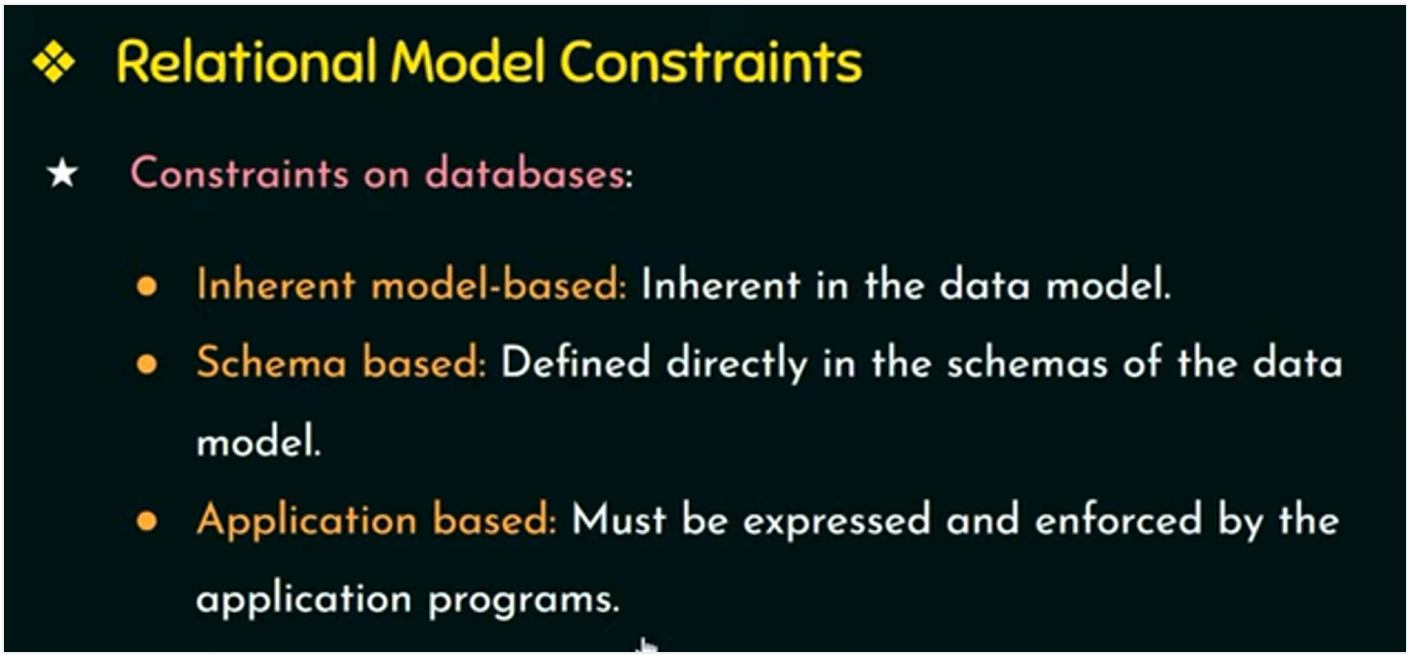
relation ↔ table

a relation is a table with rows and column.

it is used for data storage and processing







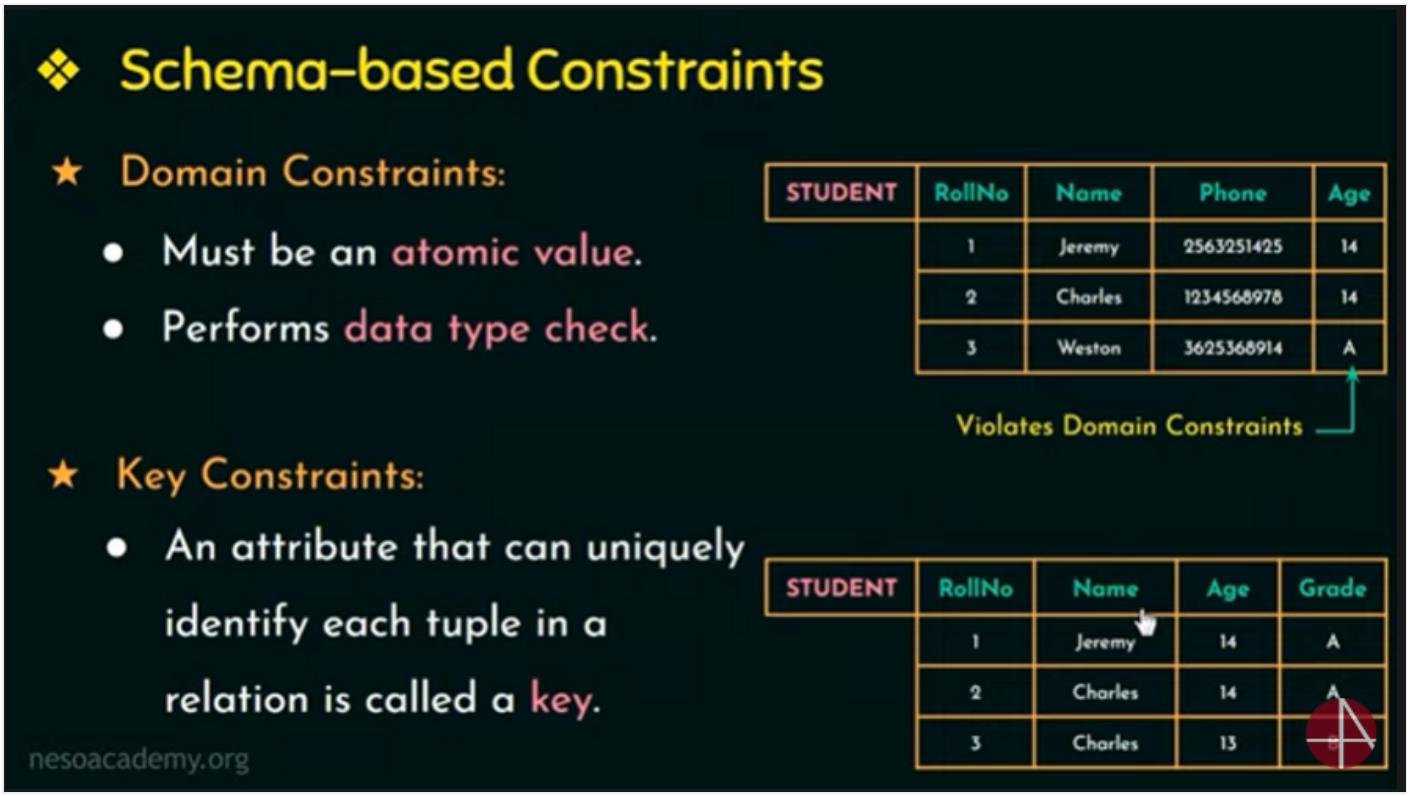
other constraints are:

data dependencies

→ functional

→ multivalued

Schema-based constraints are mainly focused.



❖ Schema-based Constraints

★ Key Constraints:

- A key satisfies 2 constraints:

  (i) Two tuples cannot have identical values for all the attributes in the key.

  (ii) It is a minimal superkey.

STUDENT	RollNo	Name	Age	Email
	1	Jeremy	14	jeremy16@gmail.com
	2	Josh	14	josh25@gmail.com
	3	Charles	15	charly01@gmail.com
	3	Alicia	13	alicia22@gmail.com

Not possible

SK = { RollNo }, { Email },
{ RollNo, Name }, { RollNo, Age },
{ RollNo, Email }, { Name, Email },
{ Age, Email },
{ RollNo, Name, Age, Email }

❖ Schema-based Constraints

★ Key Constraints

- Candidate Keys:

  → Set of attributes that uniquely identify the tuples in a relation.

STUDENT	RollNo	Name	Age	Email
	1	Jeremy	14	jeremy16@gmail.com
	2	Josh	14	josh25@gmail.com
	3	Charles	15	charly01@gmail.com

Schema-based Constraints

★ Constraints on Null Values:

- Specifies whether null values are permitted or not (NOT NULL).

STUDENT	RollNo	Name	Age	Grade
	1	Jeremy	14	A
	2	Charles	14	A
	3	Charles	13	B

★ Entity Integrity Constraint:

- States that no primary key value can be null.

STUDENT	RollNo	Name	Age	Grade
	1	Jeremy	14	A
	2	Charles	14	A
	null	Charles	13	B

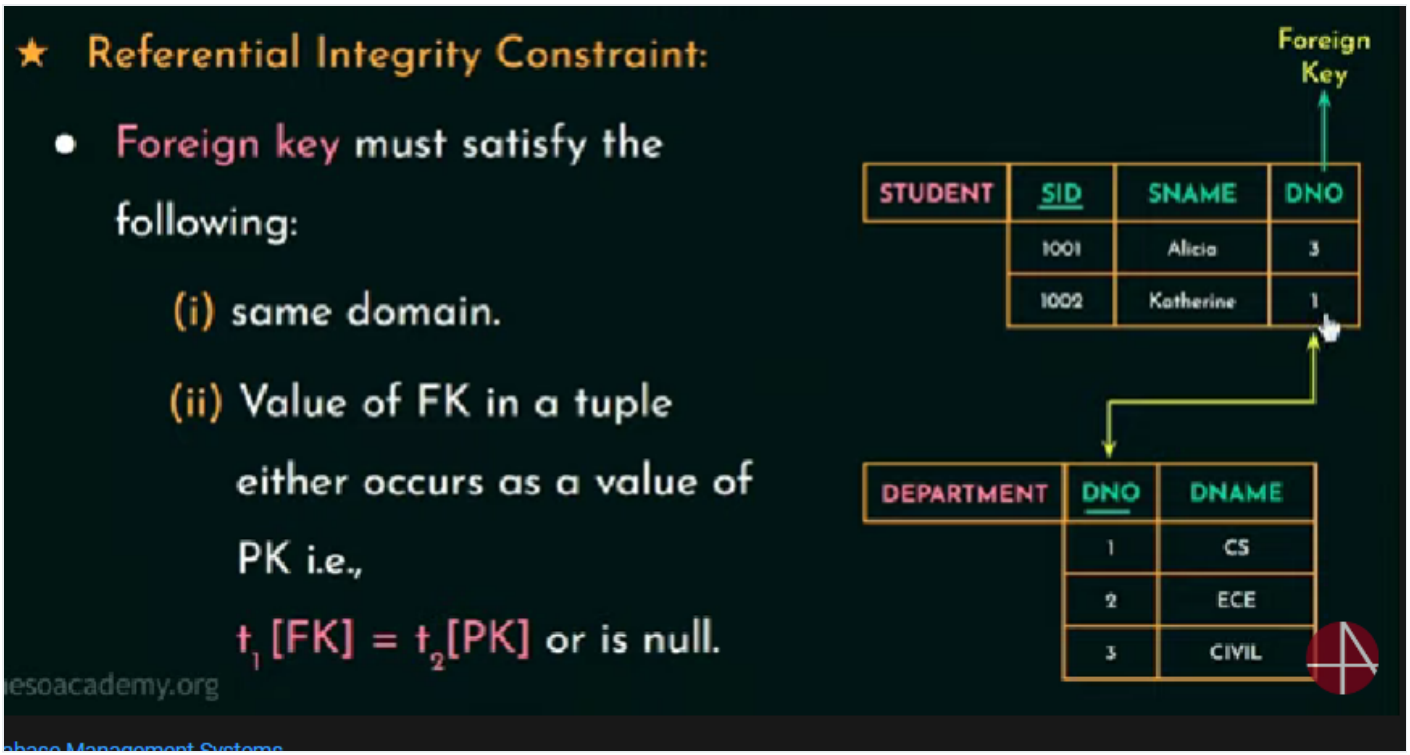
Schema-based Constraints

★ Referential Integrity Constraint:

- Specified between 2 relations.
- States that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation.

STUDENT	SID	SNAME	DNO
	1001	Alicia	3
	1002	Katherine	1

DEPARTMENT	DNO	DNAME
	1	CS
	2	ECE
	3	CIVIL



relation algebra