

DBMS

SQL Tutorial For Beginners In Hindi | DBMS Tutorial | SQL Full Course In Hindi | Great Learn...  greatlearning Learning for Life
Info Watch later Share

What is Database?

Database is a collection of information organized for easy access, management and maintenance.

- Examples:
 - Telephone directory
 - Customer data
 - Product inventory
 - Visitors' register
 - Weather records



3:36 / 2:58:15 • Intr...    YouTube 

logical structure of a database is called data models.

Types of Data Models

- **Record based logical model**
 - Hierarchical data model
 - Network data model
 - Relational data model
- **Object based logical model**
 - Entity relationship model

object-based logical model is the blueprint of your database



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



★ **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...

CC ⚙ YouTube []

❖ **Database Management System**

★ **Definition:** Collection of programs that enables users to create and maintain the database.



❖ 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



Introduction to Database Management Systems

Properties of Database



1. A database **represents some aspects of the real world** (miniworld).
2. A database is a **logically coherent collection of data** with some **inherent meaning**.
3. A database is **designed, built and populated** with data for a **specific purpose**.

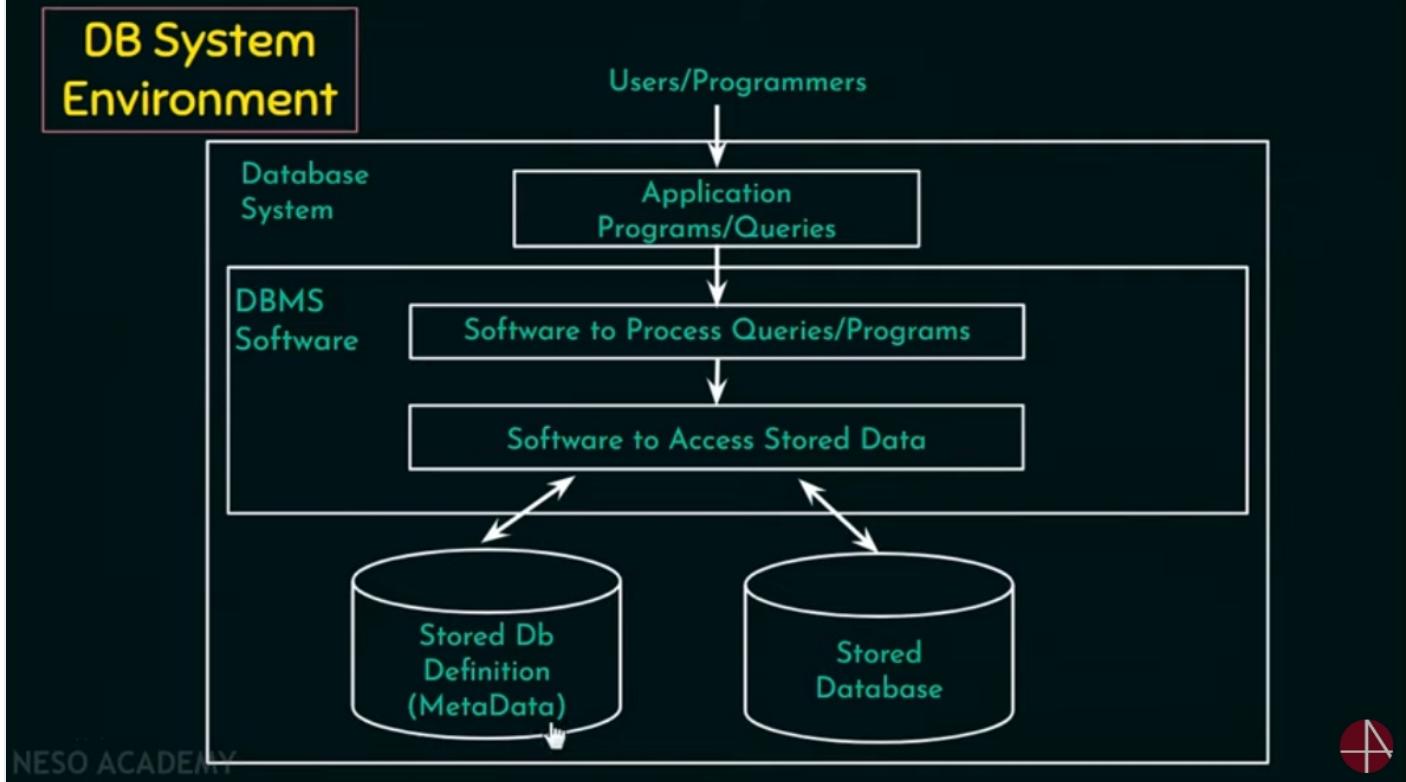
MORE VIDEOS



▶ 🔍 7:14 / 11:03 • Pro...

CC ⚙️ YouTube

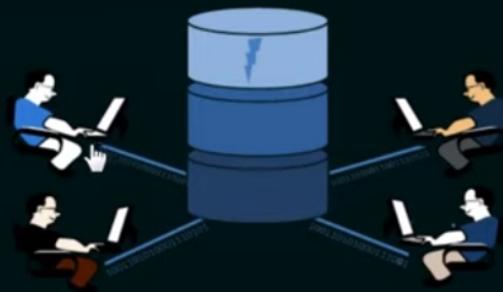
DB System Environment



❖ FILE SYSTEM APPROACH



❖ DBMS APPROACH



NESO ACADEMY

all the data can be stored in a single repository and can use it accordingly.

❖ 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.





Data Abstraction

- ★ In traditional file processing → structure of data files is embedded in the application programs.
- ★ In database approach → structure of data files is stored in the DBMS catalog → separate from access programs

(program-data independence).

[MORE VIDEOS](#)

▶️ 🔊 6:54 / 12:46 • Dat...

YouTube

❖ 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).



❖ Sharing of Data and Multiuser Transaction Processing

- ★ A multiuser DBMS allows multiple users to access the database at the same time.
- ★ DBMS must include concurrency control. OLTP processes needs concurrency control
- ★ OLTP (Online Transaction Processing) → major part of database application. Transaction means program under execution
- ★ DBMS must enforce several transaction properties:
 - i. Isolation
 - ii. Atomicity



Isolation : if two or more users are using shared resources, and if one user changes something from a resource then it should be isolated 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)



❖ 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**. 



❖ 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.



canned transaction : standarrd type of queries.



★ 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.

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???

- ★ 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



▶ soaudio 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.



❖ 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.

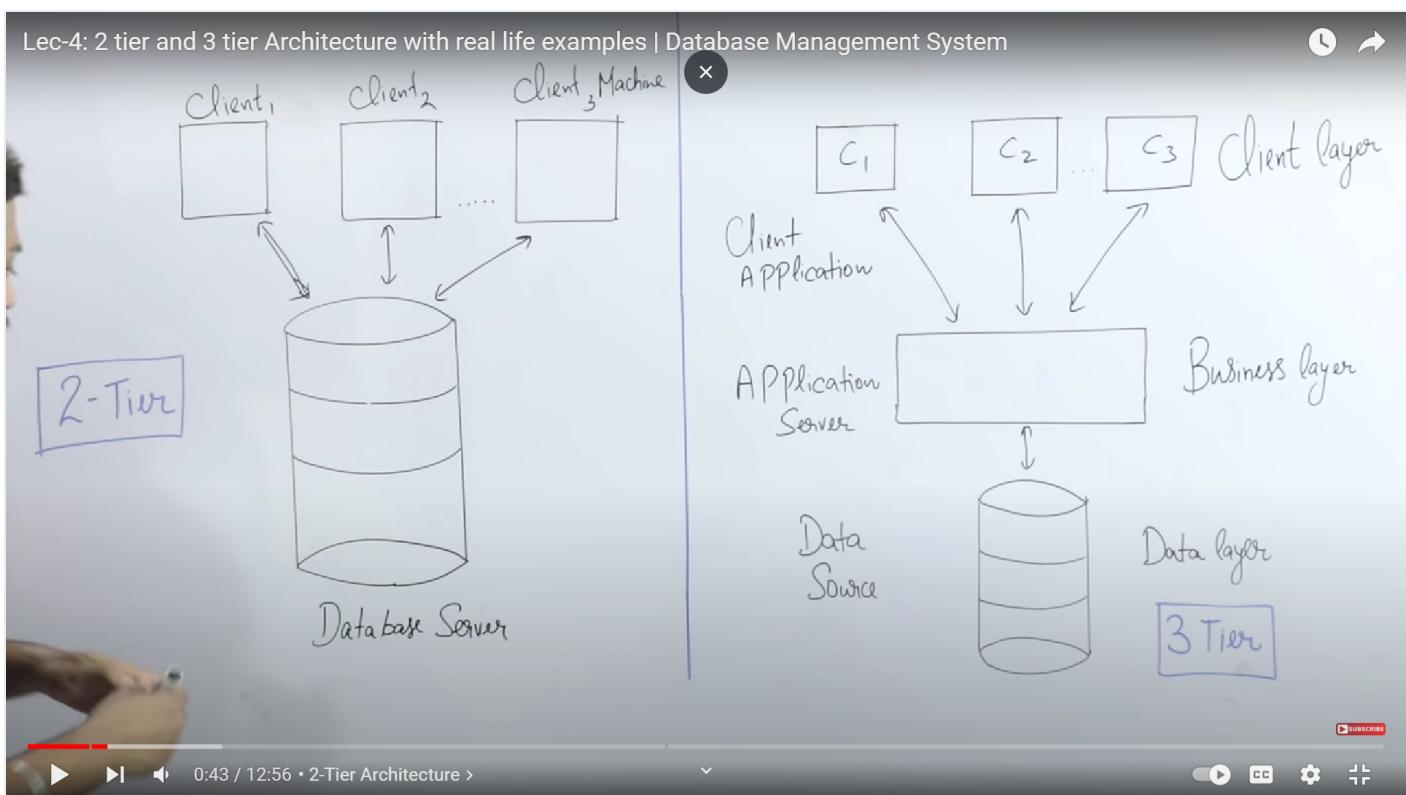


3 tier architecture

→ user

→ business

→ database



❖ Three-Schema Architecture

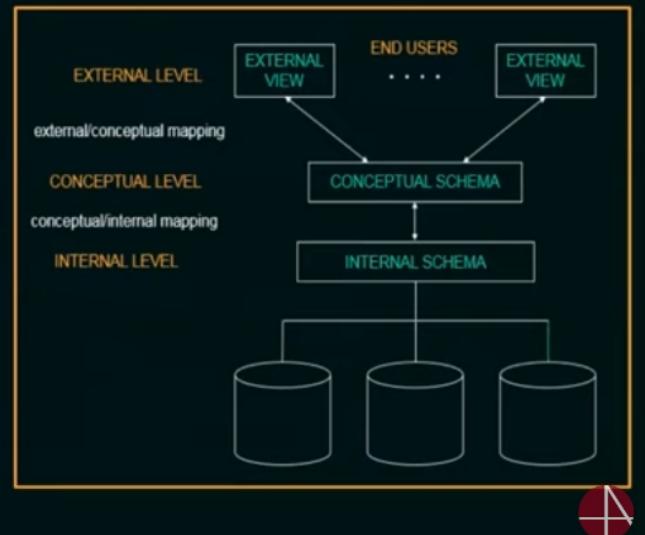
★ **Goal:** to separate the user applications and the physical database.

★ **3 Levels:**

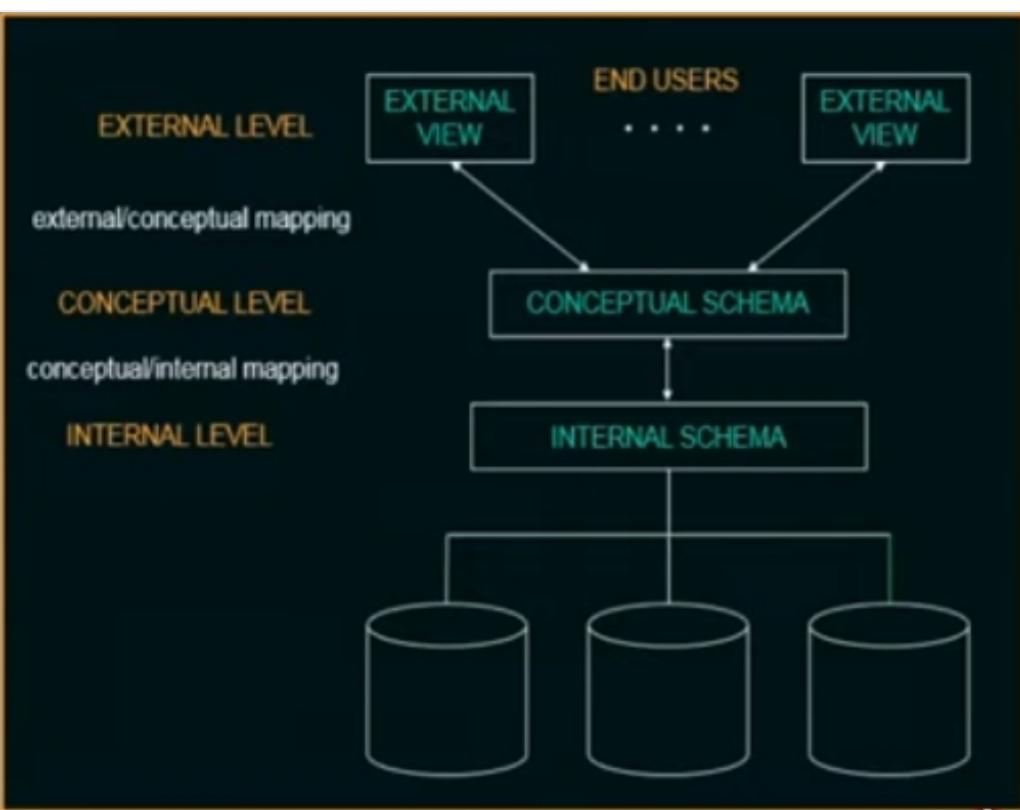
● **Internal Level:**

→ Describes the physical storage structure of the database.

→ Describes complete details of data storage and access paths.



access path - efficient path for searching a record.



Three-Schema Architecture & Data Independence

Watch later Share

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

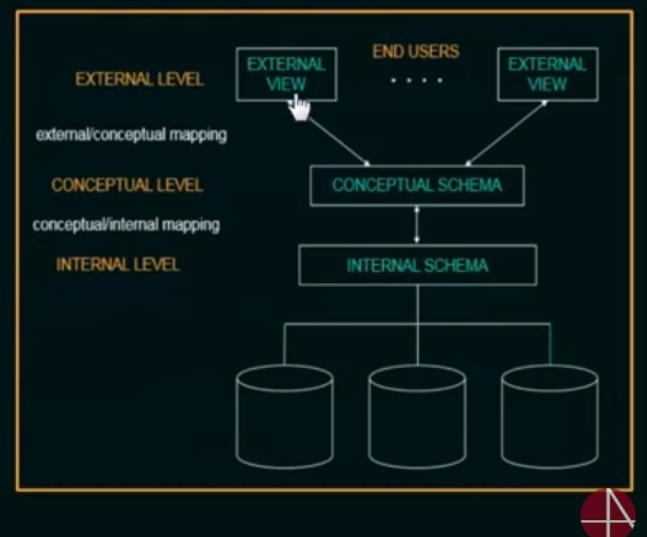
MORE VIDEOS

The diagram is identical to the one at the top of the slide, showing the Three-Schema Architecture with External, Conceptual, and Internal levels, and their respective views and databases.

▶ 2:01 / 6:49 Co... CC 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.



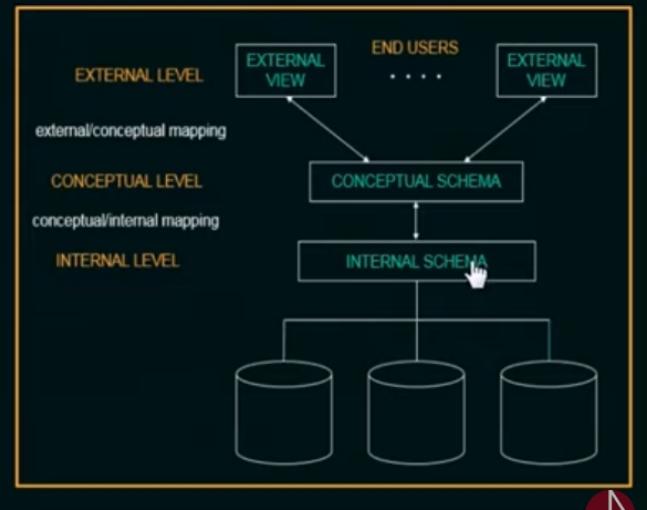
nesoacademy.org



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

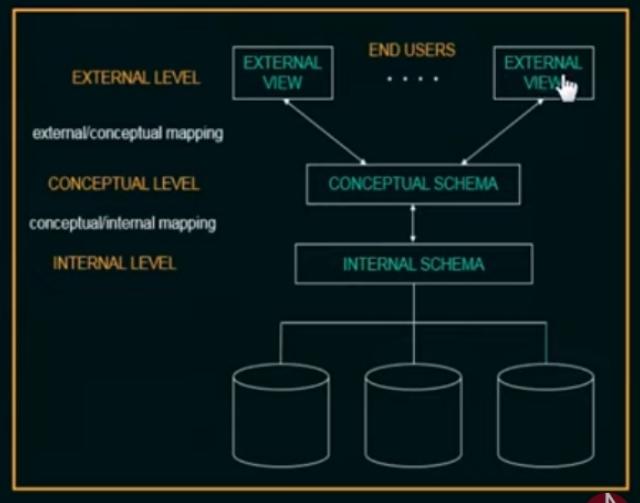


types:

❖ Data Independence

★ Logical Data Independence:

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



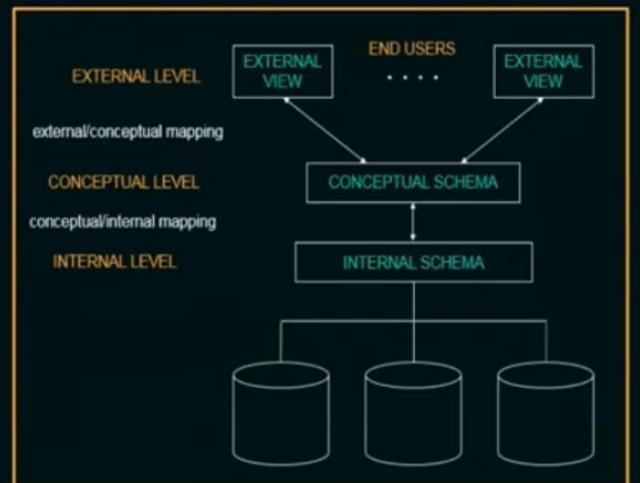
nesoacademy.org



❖ Data Independence

★ Physical Data Independence:

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



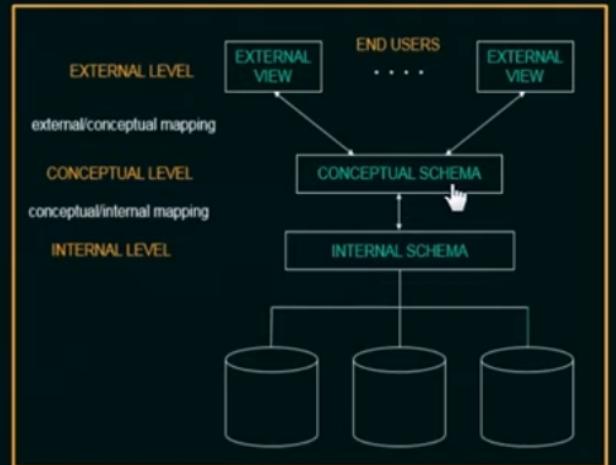
nesoacademy.org



like - changing the path of a record.

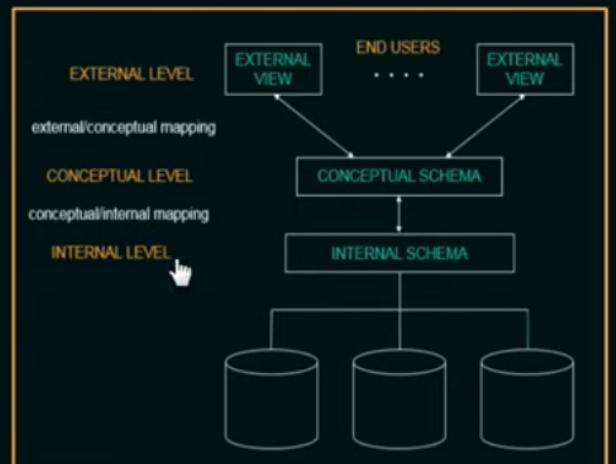
❖ DBMS Languages

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



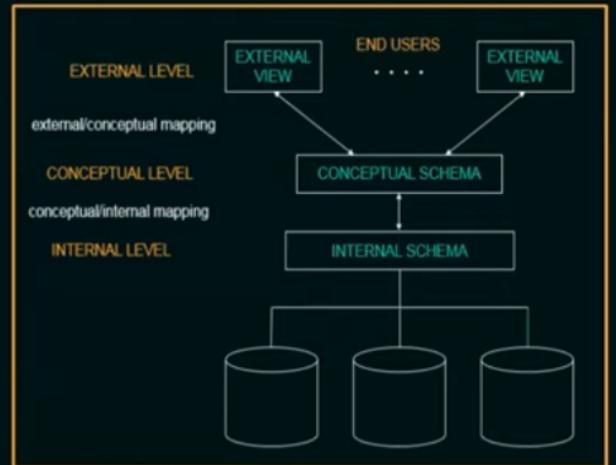
❖ DBMS Languages

- ★ 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

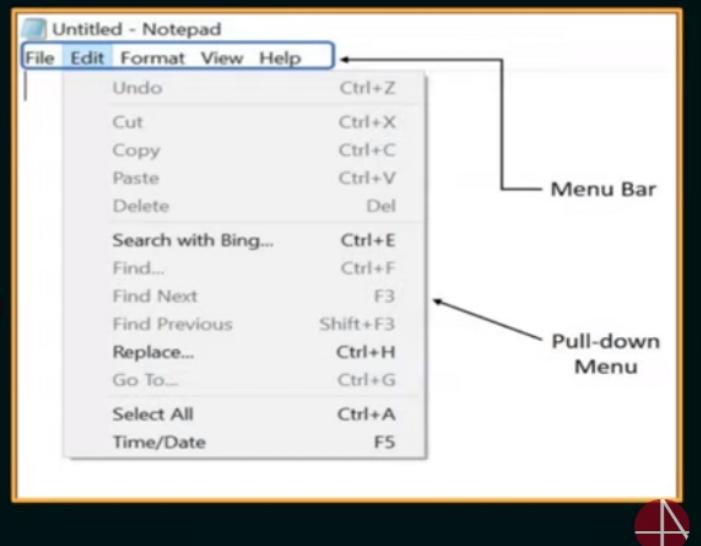
- ★ Types of DMLs:
 - High-level (or non-procedural DML):
 - Used to specify complex database operations in a concise manner.
 - Low-level (or procedural DML):
 - Embedded in a general purpose programming language.



❖ 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.

A screenshot of a web-based form for hotel booking. The form fields include:

- Check-in Date: * [Text Input]
- Check-Out Date: * [Text Input]
- No of People : Adult [Select Box] Children [Select Box]
- Special Requirement [Text Input]
- Personal Information**
- Date of Birth [Text Input]
- Anniversary [Text Input]
- First Name*: [Text Input]
- Email Address*: [Text Input]
- Your Country [Text Input] [Select Box] (Currently selected: India)
- City*: [Text Input]
- State [Text Input]
- Phone*: (Include Country/Area Code) [Text Input]

*Represents compulsory fields.

Submit Reset





★ 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



nsoacademy 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.

nesoacademy.org



DBMS Languages, Interfaces & Classification

Classification of DBMS



★ Several criteria used to classify DBMS:

- Data model
- Number of users
- Cost of DBMS
- Number of sites
- Types of access path
- Generality

MORE VIDEOS



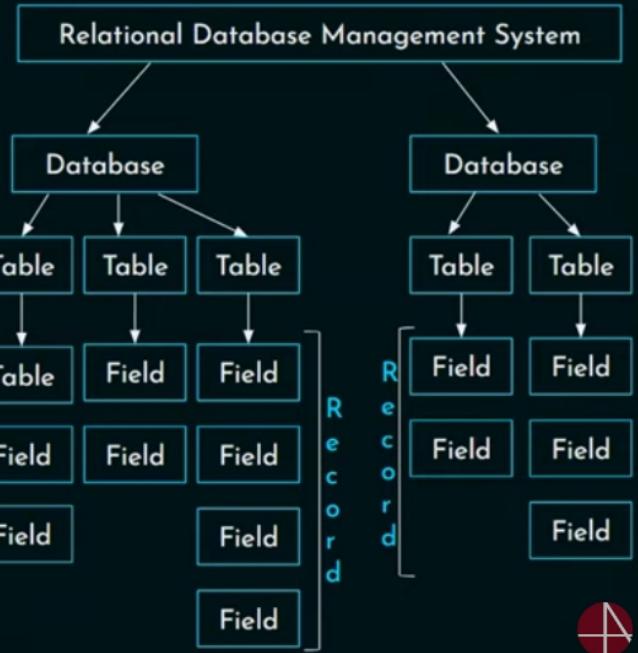
▶ sound 8:42 / 12:23 • Cla...

CC YouTube

❖ Classification of DBMS

★ Data Models:

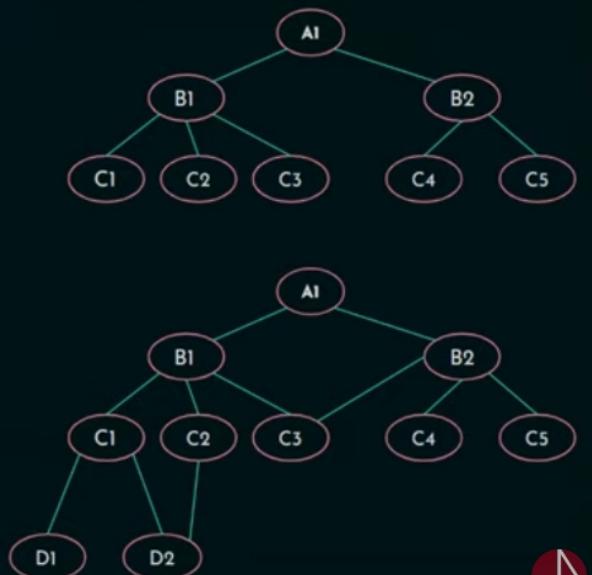
- **Relational data model:** represents database as a collection of tables.
- **Object-oriented database model:** defines database in terms of objects, their properties & their operations.



❖ Classification of DBMS

★ Data Models:

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



❖ Classification of DBMS

★ Number of Users:

- Single-user systems: supports only one user at a time.
- Multiuser systems: supports multiple users concurrently.

★ Cost of DBMS

- Low cost: Cost of these systems → between \$100 & \$3000.
- Medium cost: Cost varies between \$10,000 & \$100,000.



nesoacademy.org



high cost also there.



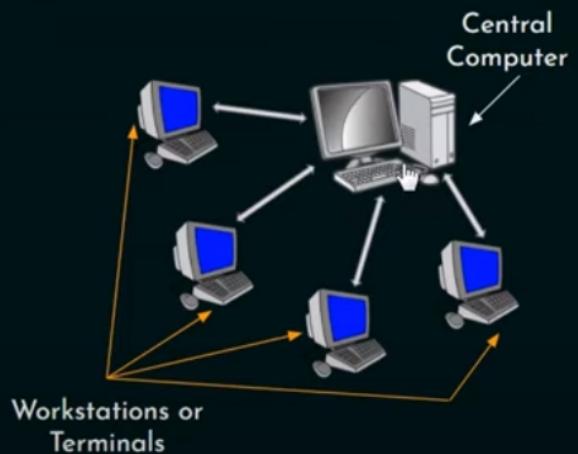
DBMS Languages, Interfaces & Classification

Classification of DBMS

Watch later Share

★ Number of Sites:

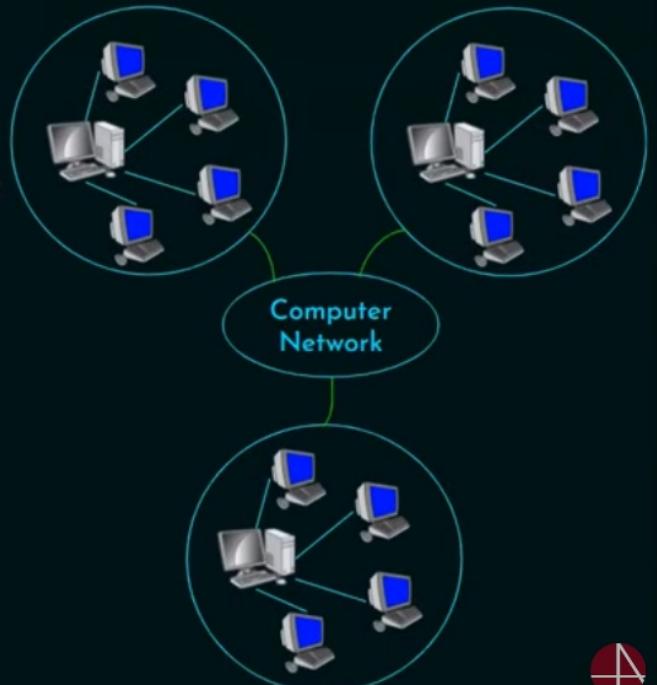
- Centralized database system: DBMS & the database → reside at a single computer site.



❖ 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

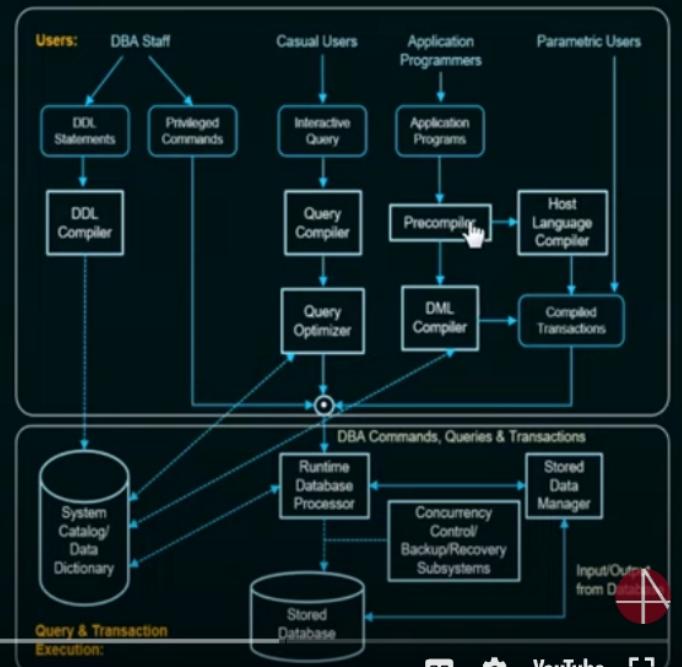
DBMS Component Modules

Watch later Share

- **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...

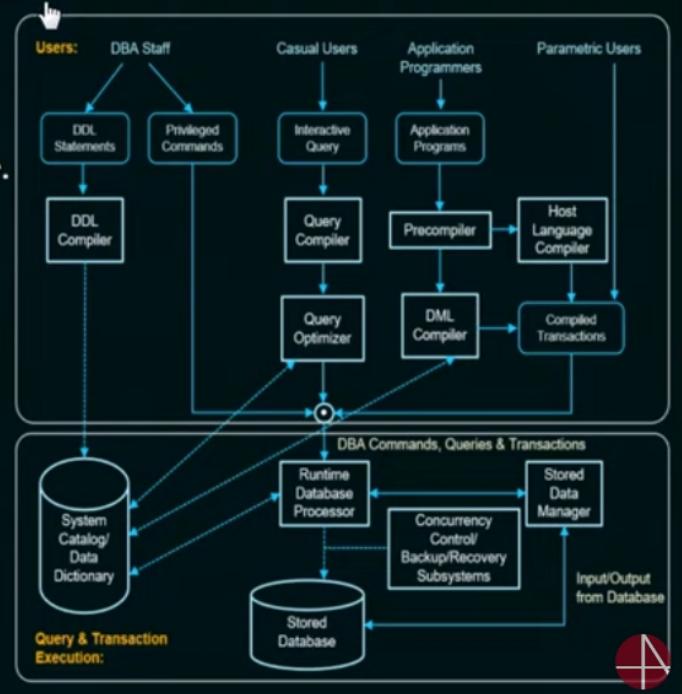


CC YouTube

❖ DBMS Component Modules

- **DML compiler** → compilation of DML commands into object code.
- **Runtime database processor** → handles database accesses at runtime.
- **Stored data manager** → data transfer between disk & main memory.

nesoacademy.org



Database System Environment

Database System Utilities

Watch later Share

- ★ Common utilities have the following types of functions:

- **Loading:**

- i. Used to **load existing data files** into the database.
- ii. The source file format and the target data file structure are mentioned to the utility



Automatically reformats the data



MORE VIDEOS

▶ sound 4:03 / 7:28 • Dat...

CC YouTube

other utilities:

backup, file reorganization, performance monitoring.

❖ Other Tools

- ★ CASE tools → used during the design of the database.
- ★ Data Dictionary (or data repository) system → storing catalog information.
- ★ Application Development Environments → provides an environment for developing database applications. Ex: JBuilder, PowerBuilder system.
- ★ Communications software → remote access to the database.

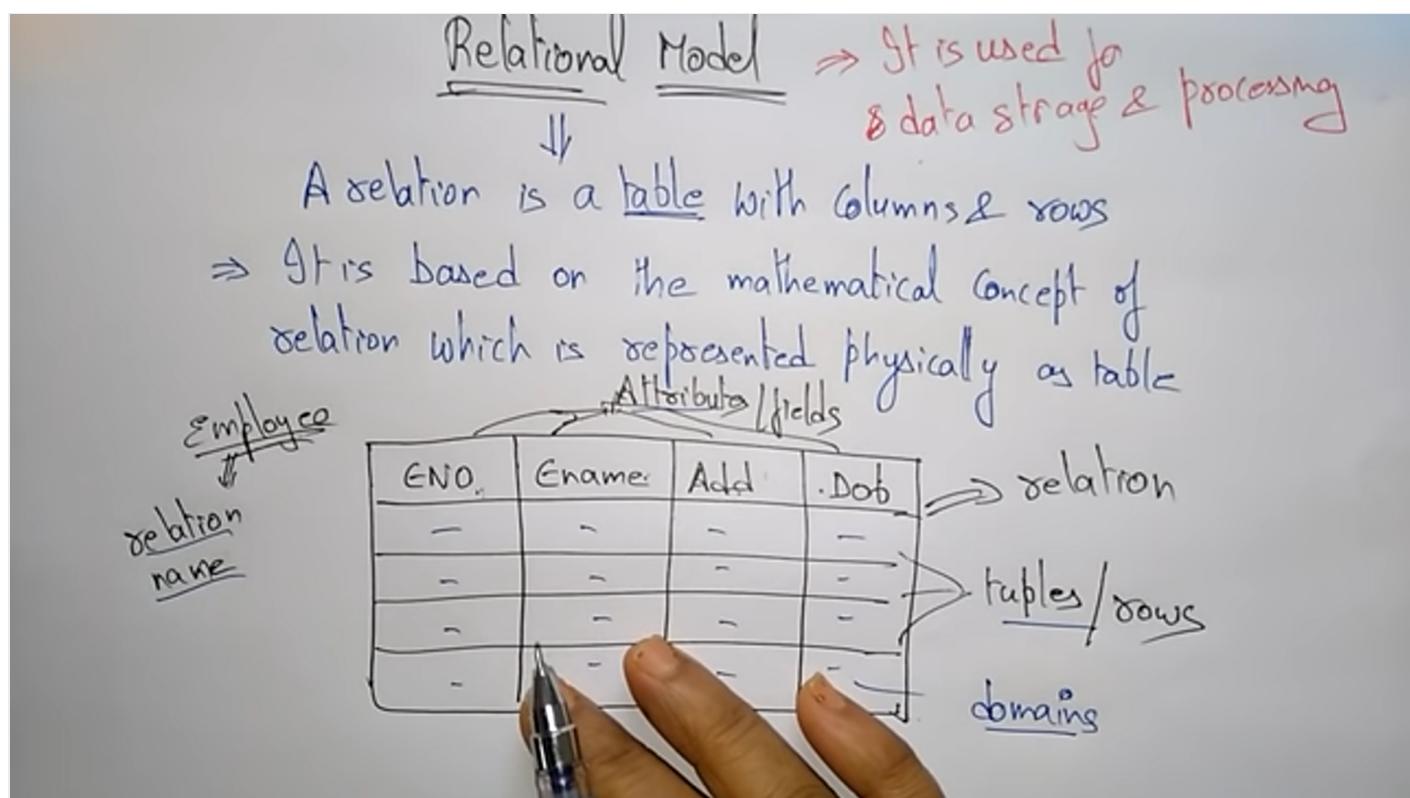


Relational model

relation ↔ table

a relation is a table with rows and column.

it is used for data storage and processing



To find the degree of relation :- = 4
no: of attributes in a relation is called degree.

Find the cardinality of relation!

no: of tuples in a relation is called cardinality

❖ Relational Model Constraints

★ Constraints on databases:

- Inherent model-based: Inherent in the data model.
- Schema based: Defined directly in the schemas of the data model.
- Application based: Must be expressed and enforced by the application programs.

other constraints are:

data dependencies

→ functional

→ multivalued

Schema-based constraints are mainly focused.

❖ Schema-based Constraints

★ Domain Constraints:

- Must be an atomic value.
- Performs data type check.

| STUDENT | RollNo | Name | Phone | Age |
|---------|---------|------------|-------|-----|
| 1 | Jeremy | 2563251425 | 14 | |
| 2 | Charles | 1234568978 | 14 | |
| 3 | Weston | 3625368914 | A | |

Violates Domain Constraints

★ Key Constraints:

- An attribute that can uniquely identify each tuple in a relation is called a key.

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

❖ 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**.

nesoacademy.org

| STUDENT | <u>RollNo</u> | 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

$SK = \{ \text{RollNo} \}, \{ \text{Email} \},$
 $\{ \text{RollNo, Name} \}, \{ \text{RollNo, Age} \},$
 $\{ \text{RollNo, Email} \}, \{ \text{Name, Email} \},$
 $\{ \text{Age, Email} \},$
 $\{ \text{RollNo, Name, Age, Email} \}$



❖ Schema-based Constraints

★ Key Constraints

• Candidate Keys:

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

| STUDENT | <u>RollNo</u> | 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 |



★ Referential Integrity Constraint:

- Foreign key must satisfy the following:
 - (i) same domain.
 - (ii) Value of FK in a tuple either occurs as a value of PK i.e.,
 $t_1[\text{FK}] = t_2[\text{PK}]$ or is null.

