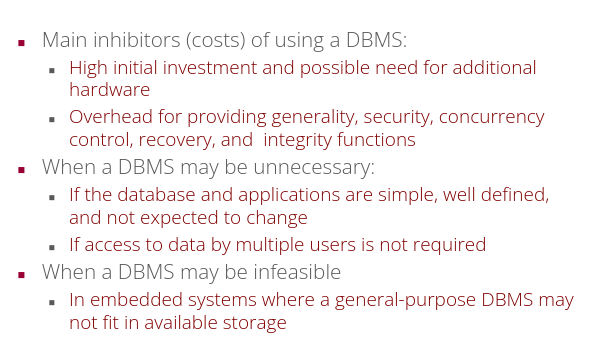
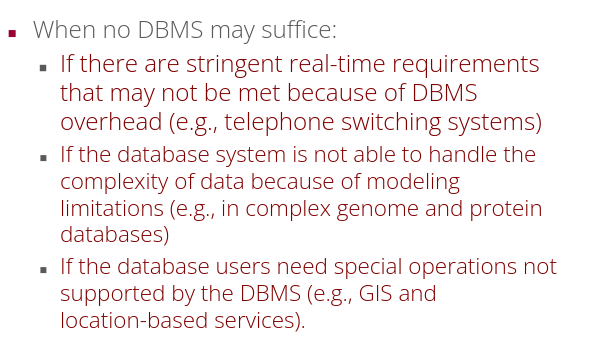
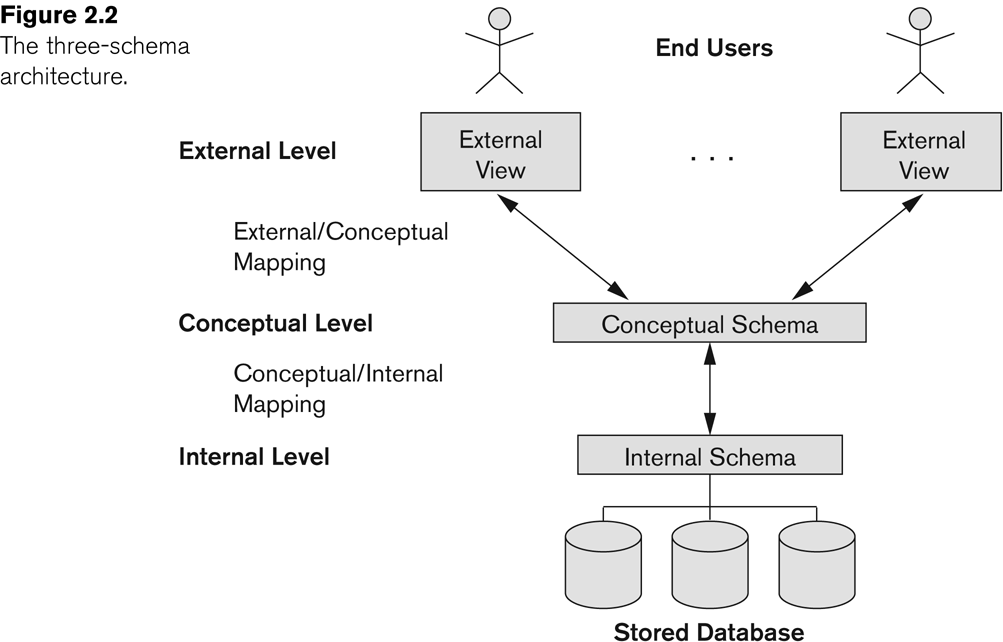
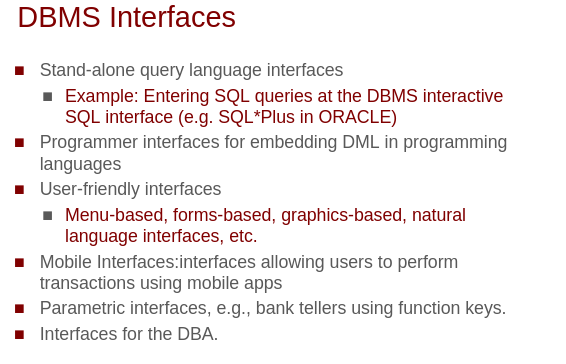
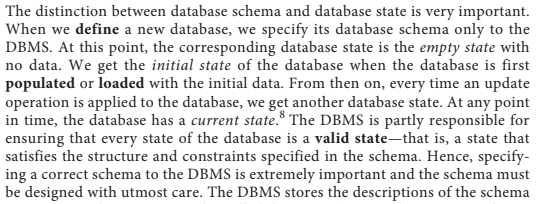
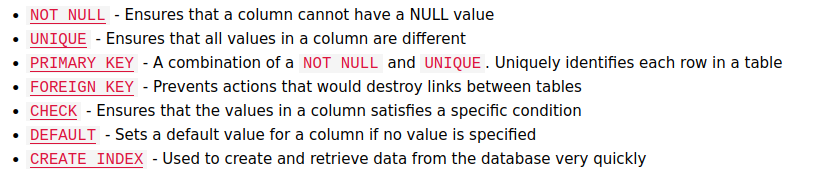
1. DBMS → A software that provides services like creating and managing data.
2. Database System → DBMS software with database.
3. Traditional → Only Numerical and textual databases. Nowadays data became more than numerics and text.
4. FIle based system and database system both uses files to store data. The way they handle data makes them different.
5. File Based System
   1. **Redundent data will be there.** Mone kor ekta school er exam dept student grades rakhtese file e and Finance dept student fee rakhtese arek file e. Both are using students name, id, phone number but as they both are in separate file, they can not access each others data. Because of that we will need more space as they are writing same file.
   2. **Cant update all file in a single click.** Ekhn mone kor student bollo tar name bhul ashce grade sheet e. So exam dept ekta click kore shob jaigai name update kore dite parbe na cause everyone has different files, so oi student ke shob dept e giye personally update korte hobe. Which is time consuming.
   3. Basically the data will not be up to data, need redundant efforts to update data and more space for a common data.
   4. Here use has to write all the procedures for managing the database. And because of that the user will be able to see everything in detail.
   5. No crash mechanism. If something happens data will be gone,
   6. No permisson mechanism. Like you can alter permission and give access to anyone.
6. Database system
   1. A common database will be there and everyone will be able to access the data and use it as they need.
   2. Here it will save time and space. And it will be easy to update.
   3. User will not require to write any type of code. Administrator will write it for you. Because of that user will not see everything.
   4. Has a crash recovery machanism. If system fails data can be recoverd.
   5. Has a permission mechanism. Using this we can control how a certain user will be able to access the file.
7. Features of DBMS
   1. Reduce redundancy
   2. Define a particular database in terms of data types, structures and constrains.
   3. Can manipulate the data - retrieval. Modification,
   4. Can process and share data to many users, yet keeping all data valid and consistent.
   5. Can provide presentation and visualization of data in multiple ways
   6. Provides protection of data
8. Meta data → Data about data. It returns the information about what types of data is in DBMS. this helps DBMS software to work with different DBMS application. The DBMS stores the descriptions of the schema constructs and constraints also called the meta-data.
9. Data Abstraction → Data abstraction generally refers to the **suppression of details** of data organization and storage, and the **highlighting of the essential features** for an improved understanding of data.
10. OLTP → online transaction process, it allows hundreds of concurrent transactions to execute per second.
11. Concurrency control → it guarantees that each transaction is correctly executed or not.
12. Main characteristics of Database approach →
    1. Self describing nature → DBMS has metadata
    2. Insulation between program and data → program data independence
    3. Data abstraction → will only show the data that users need.
    4. Support of multiple views of the data → admin can control which user see what
    5. Sharing data and multi-user transaction processing → multiple user can retrieve and update simultaneously. Also it can process multiple transaction at a time.
13. Advantage of using database →
    1. Sharing and update data among multiple users. Which will reduce redundancy.
    2. Controlling what a user will see. Multiple interfaces for different user class.
    3. Providing storage structure for efficient queries.
    4. Backup and recovery service.
    5. Representing complex relationships among data.
14. Database user
    1. Actors on the scen → who actually use DBMS
       1. Database administrators →Those who wil control and monitor everything of a database like where should data get stored or which software to use.
       2. Database Designer → those who will define the content, structure and build the blueprint of a DBMS. They are responsible to communicate with end user and develop what
       3. Software Engneers → Responsible for implementing the database and actually build it using programming language and tools.
       4. End user → Those who will use the content of database.
          1. Casual → access database when needed. Ex - middle or higher level managers
          2. Naive → Do not now how DBMS works but they use everyday in their life. Most of the end users are here. Ex - facebook user
          3. Sophisticated → familiar with database and has capibity to build their own database. They do not write program code but do use SQL queries directly.
          4. Stand alone → they make their own database application because their data do not fit on traditional formate.
    2. Workers behind scen → who designs and develops the DBMS software and tools
15. 
16. 
17. Data Model →Data models in DBMS help to understand the design at the conceptual, physical, and logical levels as it provides a clear picture of the data making. So it becomes easier for developers to create a physical database and also non technical people to understand if their requirements getting fulfilled or not. Data models are used to describe how the data is going to be stored, accessed, and updated in a DBMS. basically **A set of concepts to describe the structure of a database.**
18. Categories of data models →
    1. Conceptual → high level, provides the design of database without any technical and details term. **It provides overview ot concept of the database.**
    2. Represtational/ Logical → **provides the description** of tables, object oriented classes, xml tags, document structure and many more. document structures are defined in this model. **Representational data models hide many details of data storage on disk but can be implemented on a computer system directly.**
    3. Physical → low level, **provides the actual detail about physical structure to store the data.** Like partitions, CPU spaces etc
    4. Self describing data models → **The data storage in systems based on these models combines the description of the data with the data values themselves.**
19. Database schema → **description of database** which includes database structure, data types and constraints on the database. It changes very frequently and also known as intension.
20. Schema Diagram → illustrative display of database schema.
21. Database state/occurence/instance→ **The data in the database at a particular moment in time** is called a database state or snapshot. It shows actual data stored in a particular moment. It is also known as database instance. It changes whenever database is updated.
22. Three schema architecture → it does not explicitly used in commercial DBMS products. it is made for two reasons
    1. Program data independence
    2. Support multiple views of the data
23. DBMS schema three level
    1. External schema→ uses conceptual data model.
    2. Conceptual schema → describe structure and constrains for the whole database for a community of user. Uses implementation data model
    3. Internal schema → describes physical storage structures and access path. Uses physical data model.
    4. 
24. Data independence → Data independence can be achived if you can change lower level scheme without affecting higher level scheme. For example, in three level scheme if i change anything in physical level and if it does not affect logical level then physical data independence achieved. On the other hand,if we change logical data and if it does not affect external scheme then logical data achived. Data is separated from the programs, so that the changes made to the data will not affect the program execution and the application.
    1. Logical data independence → capacity to change conceptual schema without having to change external schema. Ex- adding new attribute without changing existing application, merging two records into one, or dividing two records, Add/Modify or Delete a new attribute.
    2. Physical data independence → capacity to change internal schema without having to change conceptual schema. Ex - changing location of database, modifying index, switching to new ds, using new storage.Change in compression techniques, Hashing algorithms and storage devices etc.
25. DDL VS DML → <https://www.geeksforgeeks.org/difference-between-ddl-and-dml-in-dbms/>
26. 
27. Difference between database schema and database state
28. Question →
    1. Difference between data level structure and three schema architecture?

Lecture -5

1. **Key Constraint** : Values should be Unique. Can take NULL values unless it's a Primary Key and NOT NULL constraint has not been set.
2. **Entity Integrity Constraint** : Values should be Unique and NOT NULL.
3. 
4. Unique e NULL ashte pare. And multiple NULL ashte pare