Syllabus

ITC303 : Database Management System

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lectures: 03 Hours/Week	03	**End-Sem (TH) :** 80 Marks

Course Objectives : The course aims:

1. To introduce various programming paradigms and the basic constructs that underline any programming language.
2. To understand data abstraction and object orientation.
3. To introduce the basic concepts of declarative programming paradigms through functional and logic programming.
4. To design solutions using declarative programming paradigms through functional and logic programming.
5. To introduce the concepts of concurrent program execution.
6. To understand use of scripting language for different problem domains.

Course Outcomes : On successful completion, of course, learner/student will be able to:

1. Understand and Compare different programming paradigms.
2. Understand the Object Oriented Constructs and use them in program design.
3. Understand the concepts of declarative programming paradigms through functional and logic programming.
4. Design and Develop programs based on declarative programming paradigm using functional and/or logic programming.
5. Understand role of concurrency in parallel and distributed programming.
6. Understand different application domains for use of scripting languages.

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	**Comment** Basic knowledge of operating systems and file systems, Any programming	**(02)**	--
I	**Database System Concepts and Architecture**	Introduction, Characteristics of Databases, File system v/s Database system, Data abstraction and Data Independence, DBMS system architecture, Database Administrator (DBA), Role of DBA **Self-learning Topics :** Identify the types of Databases. **(Refer Chapter 1)**	**(05)**	CO1
II	**The Entity-Relationship Model**	Conceptual Modeling of a database, The Entity-Relationship (ER) Model, Entity Type, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Weak entity Types	**(05)**	CO2

Sr. No.	Module	Detailed Content	Hours	CO Mapping
		Generalization, Specialization and Aggregation, Extended Entity-Relationship (EER) Model. **Self-learning Topics :** Design an ER model for any real time case study. **(Refer Chapter 2)**		
III	**Relational Model & Relational Algebra**	Introduction to Relational Model, Relational Model Constraints and Relational Database Schemas, Concept of Keys: Primary Kay, Secondary key, Foreign Key, Mapping the ER and EER Model to the Relational Model, Introduction to Relational Algebra, Relational Algebra expressions for Unary Relational Operations, • Set Theory operations, • Binary Relational operation Relational Algebra Queries **Self-learning Topics :** Map the ER model designed in module II to relational schema.. **(Refer Chapters 3 & 4)**	**(05)**	CO3
IV	**Structured Query Language (SQL) & Indexing**	Overview of SQL, Data Definition Commands, Set operations, aggregate function, null values, Data Manipulation commands, Data Control commands, Complex Retrieval Queries using Group By, Recursive Queries, nested Queries ; Integrity constraints in SQL. Database Programming with JDBC, Security and authorization: Grant & Revoke in SQL Functions and Procedures in SQL and cursors. Indexing : Basic Concepts, Ordered Indices, Index Definition in SQL **Self-learning Topics :** Physical design of database for the relational model designed in module III and fire various queries. **(Refer Chapters 5, 6, 7, 8, 9 & 10)**	**(08)**	CO4
V	**Relational Database Design**	Design guidelines for relational Schema, Functional Dependencies, Database tables and normalization, The need for normalization, The normalization process, Improving the design, Definition of Normal Forms- 1NF, 2NF, 3NF & The Boyce-Codd Normal Form (BCNF). **Self-learning Topics :** Consider any real time application and normalization upto 3NF/BCNF **(Refer Chapter 11)**	**(07)**	CO5

Sr. No.	Module	Detailed Content	Hours	CO Mapping
VI	**Transactions Management and Concurrency and Recovery**	Transaction: Transaction concept, State Diagram, ACID Properties, Transaction Control Commands, Concurrent Executions, Serializability – Conflict and View, Concurrency Control: Lock-based-protocols, Deadlock handling Timestamp-based protocols, Recovery System : Recovery Concepts, Log based recovery. **Self-learning Topics :** Study the various deadlock situation which may occur for a database designed in module V. **(Refer Chapters 12, 13 & 14)**	**(07)**	CO6

❑❑❑

Module I

Syllabus : Introduction, Characteristics of Databases, File system v/s Database system, Data abstraction and Data Independence, DBMS system architecture, Database Administrator (DBA), Role of DBA, Self-learning Topics: Identify the types of Databases.

Module II

Syllabus : Conceptual Modeling of a database, The Entity-Relationship (ER) Model, Entity Type, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Weak entity Types, Generalization, Specialization and Aggregation, Extended Entity-Relationship (EER) Model. Self-learning Topics: Design an ER model for any real time case study.

Module III

Syllabus : Introduction to Relational Model, Relational Model Constraints and Relational Database Schemas, Concept of Keys : Primary Key, Secondary key, Foreign Key, Mapping the ER and EER Model to the Relational Model.

Module V

Module VI

❑❑❑

CHAPTER 1

Module I

Introduction To Database Concepts

Syllabus

Introduction, Characteristics of Databases, File system v/s Database system, Data abstraction and Data Independence, DBMS system architecture, Database Administrator (DBA), Role of DBA, Self-learning Topics: Identify the types of Databases.

1.1 Introduction to DBMS

Q. Explain the detailed concept of DBMS.

- All of us are very much familiar with the term called as data. We come across term data regularly in our day to day life. The name of a person, the price of a book, a number of students in a college, pin code of a city, etc. are some examples of data.
- In our daily life, we have to remember the bulk amount of data. But it is easier for us to remember only some amount of data.

Example

- You may be in a position to tell accurately the age, height, income, educational qualification, residential address, etc. of your close friends.
- But it is very difficult for you to memorize all these information for a large number of individuals in your company.

1. Data

Q. Write a short note on : Data

- The facts and figures that can be recorded in system and that have some special meaning assigned to it.
- The system can be a manual system (register) or it can be a computerized system.
- **Example :** Data of a customer like name, telephone number, address and product purchased date etc.
- As need of data increased, there was need to develop a computer based system for storing and managing data as a file system or information system.

2. Database

Q. Write a short note on : Database

- A database is a collection of data items stored in one place and having something common between them Like a college database contains teachers, students, books, canteen etc. college is common (Base) between all above data items.
- So Data with a common base (Background) is called as Database (Data + Base).
- **Example :** College database stores information about students, teachers, classes, subjects (data with college as base).
- The database acts as a logical collection of relevant data. It is designed to offer an organized mechanism for storing, managing and retrieving stored information.

Student table

Sid	Name	Class	Major

Course Table

Cid	Name	Hours

Department Table

Did	Name

Marks Table

Sid	Cid	Marks	Grade

Fig. 1.1.1 : Sample Student Database

3. Database Management System (DBMS)

Q. Discuss Database system. **MU - May 17, 3 Marks**

- A **Database Management System (DBMS)** is a collection of software or programs which help user in creation and maintenance of a database (set of information). Hence it is also known as a computerized record-keeping system.

- DBMS is the software system that helps in the process of defining, constructing, manipulating the database.
- Database management system has become an integral part of the information systems of many organizations as it is used to handle a huge amount of data.
- Computer-based **Information Systems (IS)** is capable of serving to many complex tasks in a coordinated manner. Such systems handle large volumes of data, multiple users and several applications in a centralized database environment.
- The heart of an Information System (IS) is database management system. This is because most **Information Systems (IS)** have to handle huge amounts of data. This core module of an IS is also called as Database Management System (DBMS).
- Examples
  - MS Access, Fox Pro by Microsoft.
  - Oracle by Oracle corp.
  - SQL Server By Microsoft.
  - Ingres, DB2 by IBM.

Ex. 1.1.1 : Draw EER diagram for Library Management System showing aggregation. **MU - Dec.17,10 Marks**

Soln. :

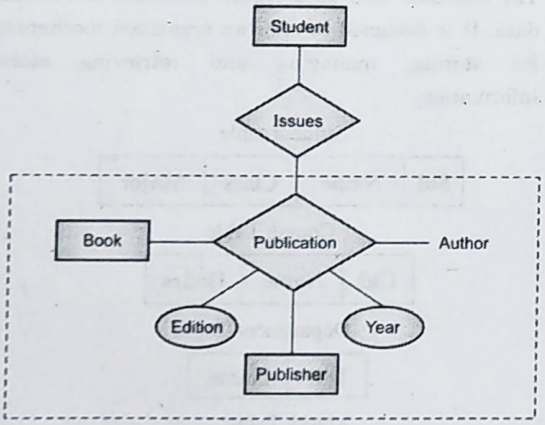

Fig. Ex. 1.1.1 : Library Management System
(Refer Chapter 2 for Details)

1.2 Characteristics of DBMS

Q. Explain the features of DBMS.
Q. Explain various advantages of Databases.

- The database approach has many important characteristics due to which database has become an integral part of the software industry.
- The various characteristics of the databases are as mentioned below :

1. Data integrity
2. Data security
3. Data independence
4. Transaction control – Commit and Rollback
5. Concurrency control
6. Data recovery – Backup and Restore

1. Data integrity

- Integrity constraints provide a way of ensuring that changes made to the database by authorized users that do not result in a loss of data consistency and correctness.
- Database integrity concern with the correctness and completeness of data in the database.
- This objective can never be guaranteed, one cannot ensure that every entry made in database is accurate.
- Some examples of incorrect data are as below :
  1. Student taking admission to branch which is not available in college.
  2. Employee assigned with non existing department.
  3. Sometime inconsistency introduced due to system failures.

2. Data security

- A DBMS system always has a separate system for security which is responsible for protecting database against accidental or intentional loss, destruction or misuse.
- Data in database should be given to only authorized users.
- Only authorized users should be allowed to modify data.
- Authorized users are able to access data any time he wants.

3. Data independence

Data Independence can be defined as the capacity to change data kept at one place without changing data kept at other locations.

4. Transaction control – rollback

- The changes made to database can be reverted back with help of rollback command.
- The changes can be saved successful with help of commit data command.

5. Concurrency control

- The data in database can be accessed by multiple users at same point of time.
- Such operations allowed by sharing same data between multiple users.

6. Data recovery - backup and restore

- Database recovery is the process of restoring the database to original (correct) state after database failure.
- The main element of database recovery is the most recent database backup.
- If you maintain database backup efficiently, then database recovery is very straight forward process.

1.3 File System v/s Database System

Q. Discuss the advantages of Database system over Files system. **MU - May 12, 5 Marks**

Q. Explain advantages of DBMS over file system. **MU - Dec. 14, 10 Marks**

Q. List four significant differences between file processing system and database management system. **MU - May 15, Dec. 15, 5 Marks**

Q. Compare the traditional file system with database. **MU - May 19, Dec.19, 5 Marks**

1. Redundancy can be reduced

- As we are using relational approach for data organization, data is not stored in more than one location.
- Repetition of information can be avoided which in turn saves storage space.

2. Inconsistency can be avoided

With the usage of database, it is assured that all the users access actual or true data present in the database.

3. Data can be shared

- Multiple users can login at a time into the database to access information.
- They can manipulate the database in a controlled environment.
- **Example :** In yahoo portal, many users are accessing data in database in a controlled manner.

4. With a **centralized control of data**, the database system may be designed for an overall optimal performance for entire organization.

5. Standards can be enforced

- Standards (rules and regulations for coding and designing) can be enforced on the database to regulate the access to the database.
- Primary Key constraint or foreign key constraint can be enforced on database which will be helpful for accessing data from database.

6. Security restrictions can be applied

- Security is the process of limiting access to the database server itself for some users.
- It is the most important for security and needs to be carefully planned.

7. Integrity can be maintained

Through integrity, one can ensure only accurate data is stored within the database.

8. Data independence can be provided

- None of the users need to know the technical aspects of the database to access it.
- They are physically as well as logically independent to access the database.

9. New applications may be developed using the existing database.

1.4 Database Users

Q. Discuss different Users. **MU - May 17, 3 Marks**

Q. Explain in detail different database users. **MU - Dec. 17, 3 Marks**

1. Naive users

- Naive users are users who interact with the system using application programs that have been developed previously.
- For example, Student wants to pay fees Rs.50 then accountant will invokes a program called fees_payment(). This program asks the accountant for the amount of fees to be paid.
- The typical graphical user interface for naive users is a kind of form interface, where the user can fill in appropriate fields of the form.
- A given end user can access the database via one of the applications or can use an interface provided as an integral part of the database system software (such interfaces are also supported by means of applications, of course, but those applications are built-in, not user-written, e.g., query language processor)
- Naive users can read reports generated from the database.

2. Application programmers

- Application programmers responsible for writing application programs that use the database.
- Application programmers are developers or computer professionals who write application programs.
- Application programmers develop user interfaces using any preferred language.
- **Rapid Application Development (RAD)** tools are available nowadays that enable an application programmer to construct application without writing code.
- Some programming languages combine control structures with database language statements. Such languages, sometimes called fourth-generation languages.

3. Sophisticated users

- Sophisticated users interact with application without writing programs by using a database query language.
- This query will be solved by query processor.
- **Online Analytical Processing (OLAP)** tools is used to view summaries of data in different ways which helps analysts (e.g. sales of region, city etc.) with OLAP analysts can use data mining tools, which help them find certain kinds of patterns in data.

4. Specialized users

- Creates the actual database and implements technical controls needed to enforce various policy decisions.
- Specialized users are sophisticated users who develop database applications.
- The DBA is also responsible for ensuring that the system operates with adequate performance and for providing a variety of other related technical services.

1.5 Data Model

Q. Explain different data models with its advantages and disadvantages.

MU - May 14, May 16, 10 Marks

- The data model will give you the idea how your final system or software will look like after development is completed.
- This concept is exactly like real world modeling in which before constructing any project (Bridges, Buildings, Towers etc.) engineers create a model for it, this model gives you the idea about how your project will look like after construction.
- A data model is an overview of a software system which describes how data can be represented and accessed from software system after its complete implementation.
- Data models define data elements and relationships among various data elements for a specified system.

Definition

According to Hoberman (2009)

- "A data model is a way finding tool for both business and IT professionals, which uses a set of symbols and text to precisely explain a subset of real information to improve communication within the organization and thereby lead to a more flexible and stable application environment."
- Data model is a simple Abstraction of complex real world data gathering environment.

1.6 Benefits of Data Modeling

- A data model is a set of concepts that can be used to **describe the structure of data** in a database.

  In Fig. 1.6.1 (Simple Logical Data model) we have described structure of student data and class data.
- Data models are used to **support the development** of information systems by providing the definition and format of data to be involved in future system.

  Data model is acting like a guideline for development also gives idea about possible alternatives to achieve targeted solution.
- A data model can be sometimes referred to as a **data structure**, especially in the context of programming languages.

  In Fig. 1.6.1 (Simple Logical Data model) student and class are data structure of type class.

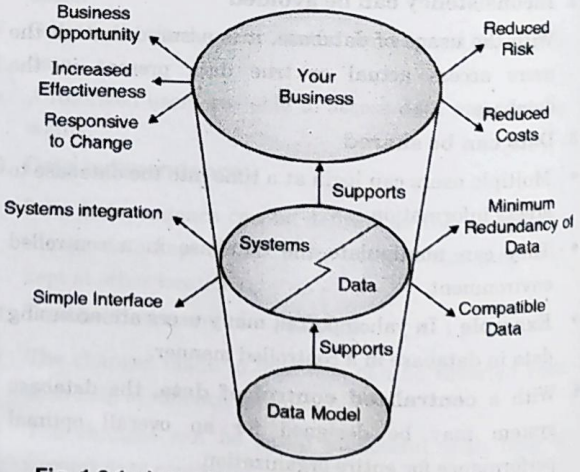

Fig. 1.6.1 : Important roles of Data Model in DBMS

a. Reduced risk

- Data model prevents system from **future risk and failure** by defining structure of data in advance.
- As we got idea of final system in the beginning of development itself so if need to have any revision or improvement we can do it in system, as actual system is not yet developed.

b. Reduced cost

As we got an idea of final system at the beginning of development itself, so we can reduce cost of project by proper planning and cost estimations as actual system is not yet developed.

c. Minimizes redundancy and data compatibility

Data repetition and data type compatibility can be checked and removed with help of data model.

d. Improves effectiveness of system

We can improve Graphical User Interface (GUI) of system by making its model and get it approved by its future user (user of that system) so it will be simple for them to operate system and make entire system effective.

1.7 DBMS System architecture

1. Introduction

- The basic building block for any of model is Entities, Attributes, relationships and constraints.
- This Model uses collection of tables to represent relationships amongst the data.

2. Entity

- A fundamental component of a model. An entity is having its own independent existence in real world.

  Example, A Student, Faculty, Subject having independent existence.
- An entity may be an object with a physical existence or it may have logical existence.

  Example, Entities like Department, Section, adult (age>18) may have physical existence or it may have only logical existence.

3. Attributes

- Each entity has its own properties which describes that entity such properties are known as **attributes.** In relational model, the column in relation (Table) or field of data is also called as Attribute.
- The single attribute will contains the similar type of data of all entities in relation.

Name
Mahesh
Suhas
Jay
Sachin

Example, The Name attribute in above student relation will contains the name of all student entities in student relation.

4. Relationships

- A relationship is an association among several entities. E.g. Employee works for Department.
- **Degree :** The degree of relationship type is number of participating entity types in a particular relation. Data model uses three types of relationships as below:

  a. One is to one
     1. One entity is associated with at most one other entity.
     2. **E.g.** One department can have only one manager.

  b. One is to Many
     1. One entity is associated with any number of entities in other entity.
     2. **E.g.** One teacher may teach to many students.

  c. Many is to Many
     1. One entity is associated with any number of entities in other entity.
     2. **E.g.** Books in library issued by students.

1.8 Object Based Logical Models

Q. Explain object based logical model.

- The data is stored in the form of objects, which are structures called *classes* that display the data within it.
- The fields are instances of these classes called as objects. This model is used in file management systems.
- The DBMS (Database Management System) developed with help of such model is called as OODBMS (Object Oriented Database Management System).
- Object oriented databases evolved to handle more complex applications such as databases for scientific experiments, geographic information system, engineering design and manufacturing.

- This model represents DB in terms of objects, their attributes and their behaviours.

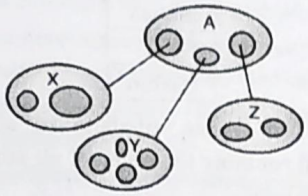

Fig. 1.8.1 : Object model

Advantages

1. OO (Object Oriented) features provide a clear modular structure which is good for defining abstract data types where internal implementation details are hidden.
2. This model is easy to maintain and modify existing code as we can create new model with small change in existing.

Disadvantages

1. This model is often provided through object oriented languages such as C++ and Java.
2. Practically very complex and inapplicable many a times.

1.9 Record Based Logical Models

Q. Explain Record based logical model.

a. Introduction

- The relational model first proposed by **E. F. Codd** hence he is known as father of Relational Model.
- A relational database is a collection of 2-dimensional tables which consists of rows and columns.

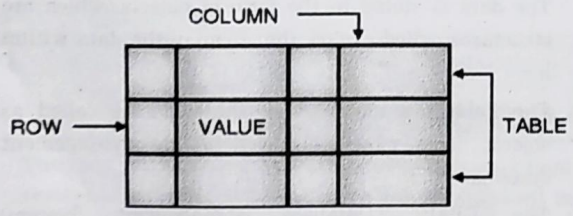

Fig. 1.9.1 : Relational model

b. Example

Most of the popular commercial DBMS products like Oracle, Sybase, MySQL etc. are based on relational model.

c. Advantages

1. **Relational algebra :** A relational database supports relational algebra and various operations of the set theory (like union, intersection etc.)
2. **Dynamic views :** In a RDBMS, a view is not a part of the physical schema, it is always dynamic. Hence changing the data in a table also changes the data present in view.
3. **SQL (Structured Query Language) :** For data access in RDBMS we have English like query language called as structured Query language (SQL) which can be used for accessing data from RDBMS. Most of the database vendors support the SQL standard.
4. **Excellent data security :** Relational databases support the concept of user rights (every user is assigned with some database permission called as user rights), thus meeting the security needs of databases.

1.10 Hierarchical Model / Tree Model / XML Based Data Model

Q. Explain Hierarchical model.

a. Introduction

- This was developed by joined efforts of IBM and North American Rockwell known as Information management system.
- It was the first DBMS model.
- The data is sorted hierarchically, either in top down or bottom up approach of designing.
- This model uses pointers to navigate between stored data.
- This model represents data as a hierarchical tree.

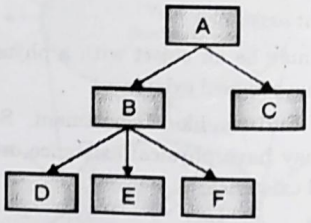

Fig. 1.10.1 : Hierarchical model

b. Example

One of the popular DBMS based on hierarchical model is Information Management System (IMS) from IBM.

c. Advantages

1. **Conceptual simplicity**

   Relationship between various levels is logically very simple. Hence database structure becomes easier to view.

2. **Database security**

   Security is given by DBMS system itself it does not depends on whether programmer has given security or not.

3. **Simple creation, updation and access**

   Hierarchical model is simple to construct with help of pointers or similar concepts and very simple to understand also adding and deleting records is easy in tree structure using pointers. This file system is faster and easy data retrieval through higher level records in tree structure.

4. **Database integrity**

   There is always Parent Child Association between different levels of records in files. Hence child record is attached with the parent record which maintains the integrity.

d. Disadvantages

1 **Complex implementation**

Only data independence is not enough for designer and programmers to build database system they need to have knowledge of physical data storage which may be complex.

2. **Difficult to manage**

Any change in a location of data needs change in all application programs that accesses changed data. Data access is restricted by Pointer path.

3. **Lack of structural independence**

- Change in database structure does not affects data access is called as structural independence. Advantage of data independence may be restricted by structural independence.
- Complex application programming
- Programmers must know how physical data is stored in order to access data. Even programmer knows path of data storage.

1.11 Network Model

Q. Explain network database model.

a. Introduction

- Like the hierarchical model, this model also uses pointers toward data but there is no need of parent to child association so it does not necessarily use a downward tree structure.
- This model used in network databases.

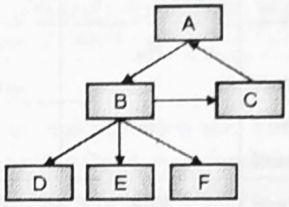

Fig. 1.11.1 : Network model

b. Example

IDS (Integrated Data Store) is one of the DBMS product based on network models. This was developed by joined efforts of IBM and North American Rockwell known as Information Management System.

c. Advantages

1. **Simple design :** The network model is simple and easy to design and understand.
2. **Ability to handle many types of relationship**
   a. The network model can handle the one-to-many or many-to-many or other relationships.
   b. Hence network model manages multi user environment.
3. **Ease of data access**
   a. In a network model an application can access a root (parent) record and all the member records within a set (child).
   b. Provide very efficient and high speed retrieval.

d Disadvantages

1. **System complexity**
   a. In a network model, data are accessed one record at a time.
   b. This can increase the complexity of system for accessing multiple records at a time.
2. **Lack of structural independence**

Any changes made to the database structure (or data) require the application programs to be modified before it can access data.

1.12 Comparison of All Data Models

Q. Compare various data models.

Sr. No.	Parameter	File system Model	Hierarchical Model	Network Model	Relational Model	ER Model	Object oriented Model
1.	Data Independence	No	Yes	Yes	Yes	Yes	Yes
2.	Structural Independence	No	No	No	Yes	Yes	Yes
3.	Storage Type	File	Segment	Record	Relation (Table)	Entity	Class
4.	Single row storage	Record	Segment occurrence	Current Record	Row(tuple)	Entity Occurrence	Object – instance of class
5.	Basic storage	Field	Segment field	Record field	Relation Attribute	Entity attribute	Object attribute
6.	Storage Identifier	Index	Sequence Field	Record key	Key	Entity key attribute	Object identifier
7.	Advantages	*Simple to implement, *Low cost implementation	*Promotes Data sharing, *conceptual simplicity, *Handle simple relationships(1:N) *Flexible data access	*conceptual simplicity, *Handle complex relationships (M:N)	*Tabular view, *Adhoc query capability, *improves management and implementation simplicity	*Very good *conceptual simplicity, *Effective *communication tool, *Integrated with dominant relational tools	*Semantic contents, *Promotes data integrity
8.	Disadvantages	*Limited implementations *Lack of standards	*Complex implementation, *Lack of standards *Limited implementations (No DML)	*Simplicity limits efficiency *Complex navigational system	*Hardware and software required	*Limited constraint representation *Limited relationship representation * No DML	* Slow development Of standards * complex navigation * Slow transactions if overload on system
9.	Examples	Operating system, Notepad, CSF(Comma Separated Files)	IMS (Information Management System)	IDS (Integrated Data Store)	Oracle, DB2, SQL SERVER etc.	ER diagrams	Database using C, java etc.

1.13 Schema and Instances

Q. Write a note on : Database Schema and Instances

1. Database Schema

- Database schema is a structure denotes the logical view of complete database.
- Schema consists of entities and relationship among these entities.
- It is similar to programming data types and variable to store data.
- The database schema is the information about database or table structure.

Student table	Sid	Name	Class	Subject

Fig. 1.13.1: Student Database Schema

2. Types of Database Schemas

a. Physical Database Schema

- This schema represents physical structure of data or actual storage of data.
- It is similar to actual variable in programming.
- It defines how the data will be stored in memory.

Example,

- The name is stored as a character data in storage.

b. Logical Database Schema

- This schema shows the logical structure need to be applied on the data stored.
- Similar to as data type of variable.
- It can be defined as tables, views, and integrity constraints.

Example,

The database consists of information about a set of students and departments in a college and the relationship between them

2. Database Instance

- The data content of the database at a particular point in time is also referred as a database instance.
- It is similar to the value of a variable,

Student table	A001	Dinesh	FYJC	DBMS

Fig. 1.13.2 : Student Database Instance

- It is always possible that this data will change with time. So it is very dynamic in nature.

1.14 Three-Levels Schema Architecture\ Data Abstraction

Q. Explain three-level architecture of DBMS.

Q. State and explain various levels of database abstraction.

Q. Explain physical, conceptual and view level abstraction of DBMS.

1. Introduction

- The goal of the three-schema architecture is to separate the front end (user applications interface) and the back end (physical database).
- The three-schema architecture is a tool with which the user can visualize the schema levels in a DBMS. Many DBMS systems do not separate the three levels completely, but support the three schema architecture to some extent.
- A description of data in terms of a data model is called a schema.
- The description of a database is called **database schema**, which is specified during database design and it does not expected to change frequently.

2. Database architecture

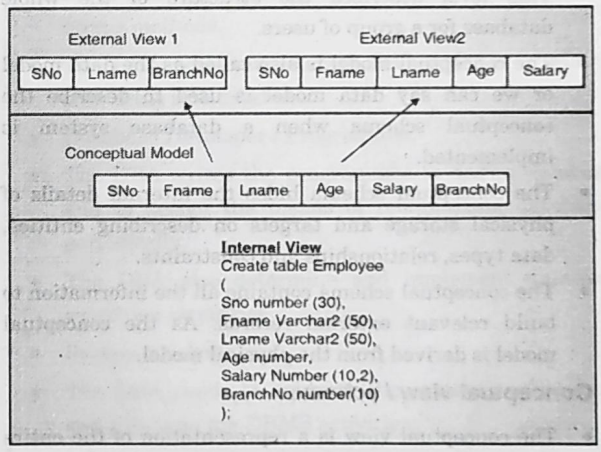

Fig. 1.14.1 : Database schema levels

a. Internal level (physical level)

- The internal level is very close to physical storage of data.
- This level describes the physical storage structure of the data in database.
- The internal (or physical) database is stored on secondary storage devices, mainly the magnetic disk.

- Describes the complete details of data storage and various available access methods for the database.
- At its ground level, it is stored in the form of bits with the physical addresses on the secondary storage device.
- At its highest level, it can be viewed in the form of files and simple data structures.

Internal view/ schema :

- The internal view defines the various stored data types and specified what indexes exist, how that stored fields are represented and so on.
- The internal schema uses a physical data model.

Example,

```
Create table Employee
( Sno        number (30),
  Fname      varchar2 (50),
  Lname      varchar2 (50),
  Age        number,
  Salary     number (10,2),
  BranchNo   number (10)        );
```

b. Conceptual level

- This level describes the structure of the whole database for a group of users.
- The conceptual model is also called as the data model or we can say data model is used to describe the conceptual schema when a database system is implemented.
- The conceptual schema hides the internal details of physical storage and targets on describing entities, data types, relationships and constraints.
- The conceptual schema contains all the information to build relevant external records. As the conceptual model is derived from the physical model.

Conceptual view / schema

- The conceptual view is a representation of the entire content of the database.
- The conceptual view includes definitions of each of the various conceptual data types.

c. External level (view level)

- The external level is the one closest to the user, i.e., it is the related with the way data is viewed by individual end users.
- The external level includes a number of user views or external schemas.
- Each external schema describes the segment of the database that is required for a particular user group and hides the rest of the database from that user group.
- External views are the proper interface between the user and the database, as an individual user can hardly be expected to be interested in the entire database.
- The external model is derived from the conceptual model.

External view / schema

External schema consists of definitions of each of the various external data types in that external view.

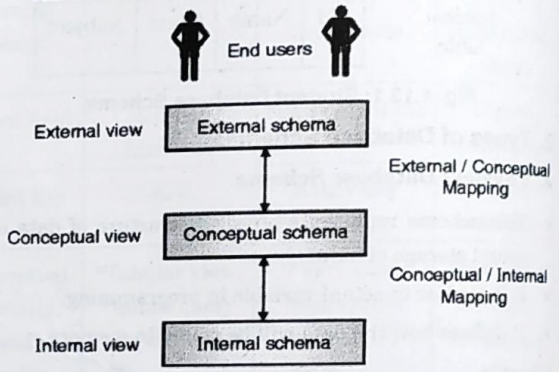

Fig. 1.14.2 : Three level schema architecture

d. Mapping

- The processes of transforming requests and results between various levels of architecture are called mappings.
- These mappings may be time-consuming, so small databases do not support external views.
- **External / conceptual mapping :** The DBMS must transform a request on an external schema into a request against the conceptual schema.
- **Conceptual / internal mapping :** A certain amount of mapping is necessary to transform requests between the conceptual and internal levels.

1.15 Data Independence

Q. Define data Independence and explain types of data Independence. **MU - Dec. 13, 5 Marks**

Q. Define data independence. **MU - Dec. 14, 2 Marks**

Q. Explain the term : Data Independence. **MU - May 16, Dec. 17, Dec.18, 2 Marks**

Q. Write short notes on : Data Independence. **MU - Dec. 16, Dec.19, 5 Marks**

Concept of data independence can be explained with help of 3 schema architecture. The three-schema architecture can make it easier to achieve true data independence.

1. Definition

Data Independence can be defined as the capacity to change one level of schema without changing the schema at the next higher level.

2. Types

a. Logical data independence
b. Physical data independence

a. Logical data independence

- Logical data independence is a capacity to change the conceptual schema without having any changes to external schemas. (or application programs)
- Separating the external views from the conceptual view enables us to change the conceptual view without affecting the external views. This separation is sometimes called logical data independence.
- **Example :** We may change the conceptual schema by removing a data item. In this case the external schemas that refer only to the remaining data should not be affected.

b. Physical data independence

- Physical data independence is a capacity to change the internal schema without having any changes to conceptual schema.
- The separation of the conceptual view from the internal view enables us to provide a logical description of the database without the need to specify physical structures. This is often called physical data independence.
- **Example :** By creating additional access paths to improve the performance of retrieval. If the same data as before remains in the database, we should not have to change the conceptual schema.

1.16 Database Administrator (DBA)

Q. Define following terms : DBA. **MU - May 19, 5 Marks**

- The database administrator is responsible for the overall planning of the company's data resources, for the design of data, and for the day-to-day operational aspects of data management.
- A database administrator is a person responsible for the installation, configuration, up gradation, maintenance and monitoring databases in an organization.
- The overall planning of corporate data is the strategic aspect of the database administration function and involves company-wide planning of existing data and assessment of organization-wise data standards.

1.17 Role of Database Administrator (DBA)

Q. Discuss the role of Database Administrator. **MU - May 16, 5 Marks**

1.17.1 Roles of DBA

Q. Explain Role of DBA? **MU - May 18, Dec.19, 5 Marks**

- The DBA needs to performs many roles to keep the database up and running,
- System Administrator / Designer
- The database administrator need to manage DBMS software and server.
- He is also responsible for deciding on the storage and access methods.
- The DBA performs all data field updates or adding new fields into database.
- Database Developer / Programmer
- The DBA writes the programmes to design database and to design the means of reorganizing databases periodically.
- The DBA also determine and implement database searching strategies.
- System Analyst
- The DBA needs to analyses the system performance and fine tune the DBMS activities.
- DBA needs to take care of system crashes by planning proper recovery procedures.
- He will also specify techniques for monitoring database performance

1.17.2 Responsibilities of DBA

Q. Write short notes on : Responsibilities of Database Administrator. **MU - May 12, 5 Marks**

- The various responsibilities of DBA are as follows,

- Designing overall Database schema
- The DBA is responsible for designing overall database schema (tables and fields). Also responsible for deciding on the data storage and access methods.
- Selecting and installing database software and hardware
- The DBA selects the suitable DBMS software like Oracle, SQL Server or MySQL. The Designing the means of reorganizing databases periodically.
- Designing Authorization/Access Control
- The DBA will decide the user access levels and security checks for access and data manipulations.
- Designing Recovery Procedures
- In order to take care of system crashes DBA needs to design the system recovery procedures and also specifying techniques for monitoring database performance.
- Operations Management
- The operations management of database administration deals with data problems arising on a day-to-day basis. Specifically, the responsibilities include
  - Investigation of errors found in the data.
  - Supervision of restart and recovery procedures in the event of a failure.
  - Supervision of reorganization of databases.
  - Initiation and control of all periodic dumps of data.

1.17.3 Skills required for DBA

Q. List the important skills required to a Database Administrator (DBA).

- The various programming and soft skills are required to DBA are as follows,
  - Good communication skills
  - Excellent knowledge of databases architecture and design and RDBMS
  - Knowledge of Structured Query Language (SQL).
- In addition, this aspect of database administration includes maintenance of data **security,** which involves maintaining security authorization tables, conducting periodic security audits, investigating all known security breaches.
- To carry out all these functions, it is crucial that the DBA has all the accurate information about the company's data readily on hand. For this purpose he maintains a *data dictionary.*
- The data dictionary contains definitions of all data items and structures, the various schemes, the relevant authorization and validation checks and the different mapping definitions.
- It should also have information about the source and destination of a data item and the flow of a data item as it is used by a system. This type of information is a great help to the DBA in maintaining centralized control of data.

1.18 Detailed DBMS Architecture

1.18.1 DBMS Architecture

Q. Draw and explain database system structure.
MU - Dec. 13, May 15, May 17, 10 Marks

Q. Describe overall architecture of DBMS with diagram.
MU - May 15, May 18, 10 Marks

- A database system can be separated into two different modules that deal with all operations of the overall system.
- Components of a database system,
  1. Query Processor Components
  2. Storage Manager/ Storage Management
  3. Transaction Management
- The storage manager is important because databases typically require a huge amount of storage space.

1.18.2 Query Processor Components

Q. Write a short notes on : Query processor.

1. Introduction

The query processor will accept query from user and solves it by accessing the database.

2. Parts of query processor

a. DDL interpreter
b. DML compiler
c. Query evaluation engine

a. DDL interpreter

This will interprets DDL statements and fetch the definitions in the data dictionary.

b. DML compiler

- This will translates DML statements in a query language into low level instructions that the query evaluation engine understands.
- A query can usually be translated into any of a number of alternative evaluation plans for same query result DML compiler will select best plan for **query optimization**.

c. Query evaluation engine

This engine will execute low-level instructions generated by the DML compiler on DBMS.

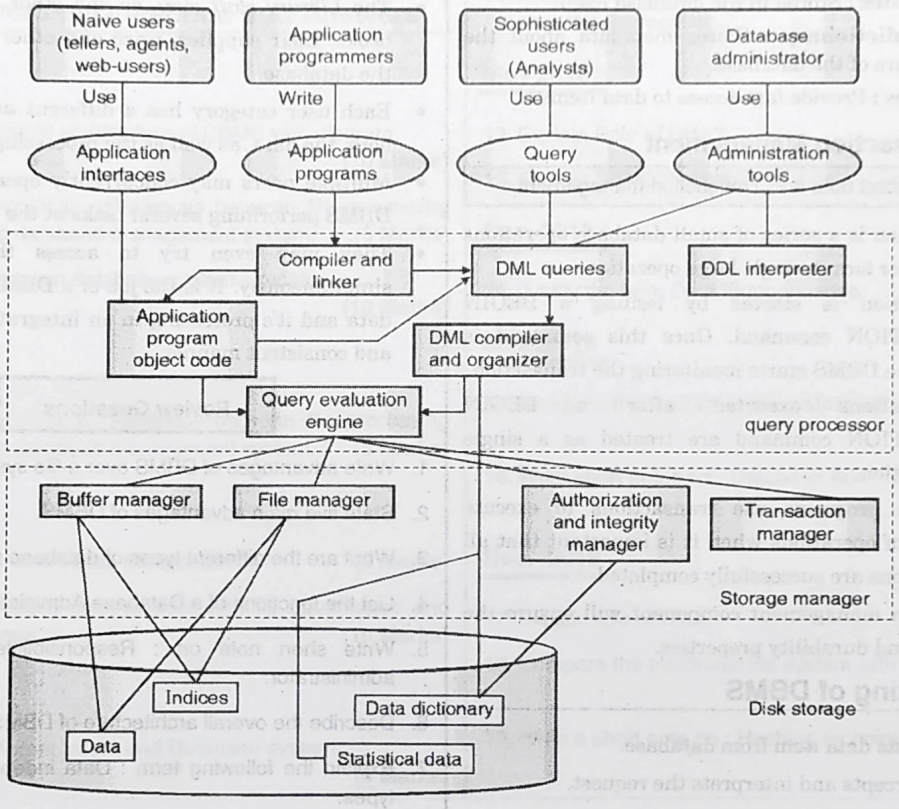

Fig. 1.18.1 : Components of DBMS

1.18.3 Storage Manager / Storage Management

Q. Write a short notes on : Storage management.

- A **storage manager** is a program module which acts like interface between the data stored in the database and the application programs and queries submitted to the system.
- The data is stored on the disk using the file system.
- The storage manager is programme which is responsible for the interaction with the file manager.
- The storage manager translates the various databases language statements into low level file system commands.
- Thus, the storage manager is responsible for storing, retrieving and updating data in the database.
- The storage manager components include :
  - **Authorization and integrity manager :** Checks for integrity constraints and authority of users to access data.
  - **Transaction manager**, which ensures that the database remains in a consistent (correct) state although there is system failures.
  - **File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

- **Buffer manager**, which is responsible for retrieving data from disk storage into main memory. The buffer manager is an important part of the database system, as it enables the database to handle data sizes that are much larger than the size of main memory.

- Data structures implemented by storage manager,
  - **Data files :** Stored in the database itself.
  - **Data dictionary :** Stores metadata about the structure of the database.
  - **Indices :** Provide fast access to data items.

1.18.4 Transaction Management

Q. Write a short note on : Transaction management.

- A transaction is a series of small database operations that together form a single large operation.
- A transaction is started by issuing a BEGIN TRANSACTION command. Once this command is executed the DBMS starts monitoring the transaction.
- All operations executed after a BEGIN TRANSACTION command are treated as a single large operation.
- Application programs use transactions to execute sequences of operations when it is important that all the operations are successfully completed.
- Transaction management component will ensure the atomicity and durability properties.

1.19 Working of DBMS

a. User requests data item from database.

b. DBMS intercepts and interprets the request.

c. Retrieves the data from the physical database.

d. Constructs the record using physical/conceptual mapping.

e. Records constructed using relevant conceptual / external mapping.

f Derives the required external record from conceptual record.

Example

- Consider the situation in a library. Here, we have **data** corresponding to *books, authors, suppliers, borrowers*, etc. The total volume of data stored and handled in a library may be quite large.
- The Library DBMS may require several **operations,** such as *issue, return* or *purchase* of books; handle **queries** relating to *book information, borrowing information*, etc.
- Moreover, there are different types of **users** who operate various stages or activities. For instance, a *borrower* may merely view certain information, whereas an *issuer* may be allowed to update the status of a book during issue or return.
- The *Library staff* may, on the other hand, add new books, their supplier, price and other information to the database.
- Each user category has a different **access right** on both, the data, as well as the processing capabilities.
- Multiple users may concurrently operate the library DBMS performing several tasks at the same time.
- They may even try to access the same data simultaneously. It is the job of a DBMS to handle the data and it's processing in an integrated, coordinated and consistent manner.

Review Questions

1. Write advantages of DBMS over a file system.
2. State five main advantages of DBMS.
3. What are the different types of database system users?
4. List the functions of a Database Administrator (DBA).
5. Write short note on : Responsibilities of database administrator.
6. Describe the overall architecture of DBMS with diagram.
7. Explain the following term : Data independence and its types.
8. Explain the features of DBMS.
9. Explain three-level architecture of DBMS.
10. Explain need of data model in DBMS.
11. Explain the difference between external, internal and conceptual schemas. How these different layers are related to the concepts of logical and physical and physical data independence ?
12. Describe advantages of database management system over file-processing system.
13. What are the advantages of file processing system which were removed by DBMS ?
14. Why would you choose a database system instead of simply storing the data in operating system files ?

15. What is Data model ? Explain various types of data models.

16. Explain Hierarchical and network database model.

17. Compare various data models available.

18. Explain various data model with their advantages and disadvantages.

1.20 University Questions and Answers

May 2015

1. Describe overall architecture of DBMS with diagram. **(10 Marks)**

2. List four significant differences between file processing system and database management system. **(5 Marks)**

3. Draw and explain database system structure. **(10 Marks)**

Dec. 2015

4. List four significant differences between file processing system and database management system. **(5 Marks)**

May 2016

5. Explain the term : Data independence **(2 Marks)**

6. Explain different data models with its advantages and disadvantages. **(10 Marks)**

May 2017

7. Discuss different Users and Database system. **(5 Marks)**

8. Draw and explain database system structure. **(10 Marks)**

Dec. 2017

9. Explain Data Independence. **(5 Marks)**

10. Explain in detail different database users. **(10 Marks)**

11. Draw EER diagram for Library Management System showing aggregation. **(10 Marks)**

May 2018

12. Explain Role of DBA? **(5 Marks)**

13. Explain DBMS architecture. **(10 Marks)**

Dec. 2018

14. Justify the term Data Independence. **(5 Marks)**

May 2019

15. Compare the traditional file system with database. **(5 Marks)**

16. Write short notes on : Database Administrator. **(5 Marks)**

Dec. 2019

17. Explain role of DBA. **(5 Marks)**

18. Compare the traditional file system with database. **(5 Marks)**

19. Write a short note on : Hashing techniques **(5 Marks)**

❑❑❑