Database Management Systems
SYLLABUS

- **UNIT-I:**

  Introduction To Database And System Architecture: Database Systems and their Applications, Database Vs File System, View of Data, Data Models, Database Languages- DDL and DML, Transaction Management, Database users and Administrators, Database System Structure.

  Introduction to Database Design: ER Diagrams, Entities, Attributes and Entity sets, Relationships and Relationship set, Extended ER Features, Conceptual Design with the ER Model, Logical database Design.

- **UNIT-II**

  SQL: Queries and Constraints: Form of Basic SQL Query, SQL Operators, Set Operators, Nested Queries, Aggregate Operators, NULL values, Integrity Constraints Over Relations, Joins, Introduction to View, Destroying / Altering Tables and Views, Cursors, Triggers and Active Databases.

- **UNIT-III**

  RELATIONAL MODEL: Introduction to Relational Model, Basic Structure, Database Schema, Keys, Relational Algebra and Relational Calculus.

  Storage and Indexing: File Organizations and Indexing-Overview of Indexes, Types of Indexes, Index Data Structures, Tree structured Indexing, Hash based Indexing.

- **UNIT-IV**

SCHEMA REFINEMENT AND NORMAL FORMS : Introduction to Schema Refinement, Functional Dependencies, Reasoning about FD, Normal Forms, Properties of Decomposition.

- **UNIT-V**

TRANSACTION MANAGEMENT TRANSACTIONS: Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability.

Concurrency Control: Lock based Protocols, Timestamp based protocols

Recovery System: Recovery and Atomicity, Log based recovery, Shadow Paging, Recovery with concurrent Transactions, Buffer Management.

- **TEXT/REFERENCE BOOKS**

1. "Data base Management Systems", Raghurama Krishnan, Johannes Gehrke,TATA McGraw Hill 3rd Edition
2. "Data base System Concepts", Silberschatz, Korth, McGraw hill, Vedition.
3. "Introduction to Database Systems", C.J.Date PearsonEducation.
4. "Database Systems design, Implementation, and Management", Rob & Coronel5th Edition.
5. "Database Management Systems", P. Radha Krishna HI-TECH Publications2005.
6. "Database Management System", ElmasriNavate PearsonEducation.
7. "Database Management System", Mathew Leon,Leo.

Course Objectives

1. To understand the different issues involved in the design and implementation of a database system.
2. To understand Structured Query Language for manipulating the Data.
3. To study the physical, conceptual and logical database designs
4. To provide concepts of Transaction, Concurrency and Recovery Management Strategies of a DBMS
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Outcomes:

1. Identify the role of Database System Applications and the design issues related.
2. Design the logical model for the applications and apply indexing techniques.
3. Construct a Database Schema, manipulate data using a SQL.
4. Can apply the Schema Refinement techniques for a database design for optimized access.
5. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.

UNIT-I

DBMS

INTRODUCTION

A.PAVITHRA
ASSISTANT PROFESSOR
DEPT OF IT
GRIET, HYDERABAD

Introduction

A database is a collection of related data

Eg: telephone numbers.

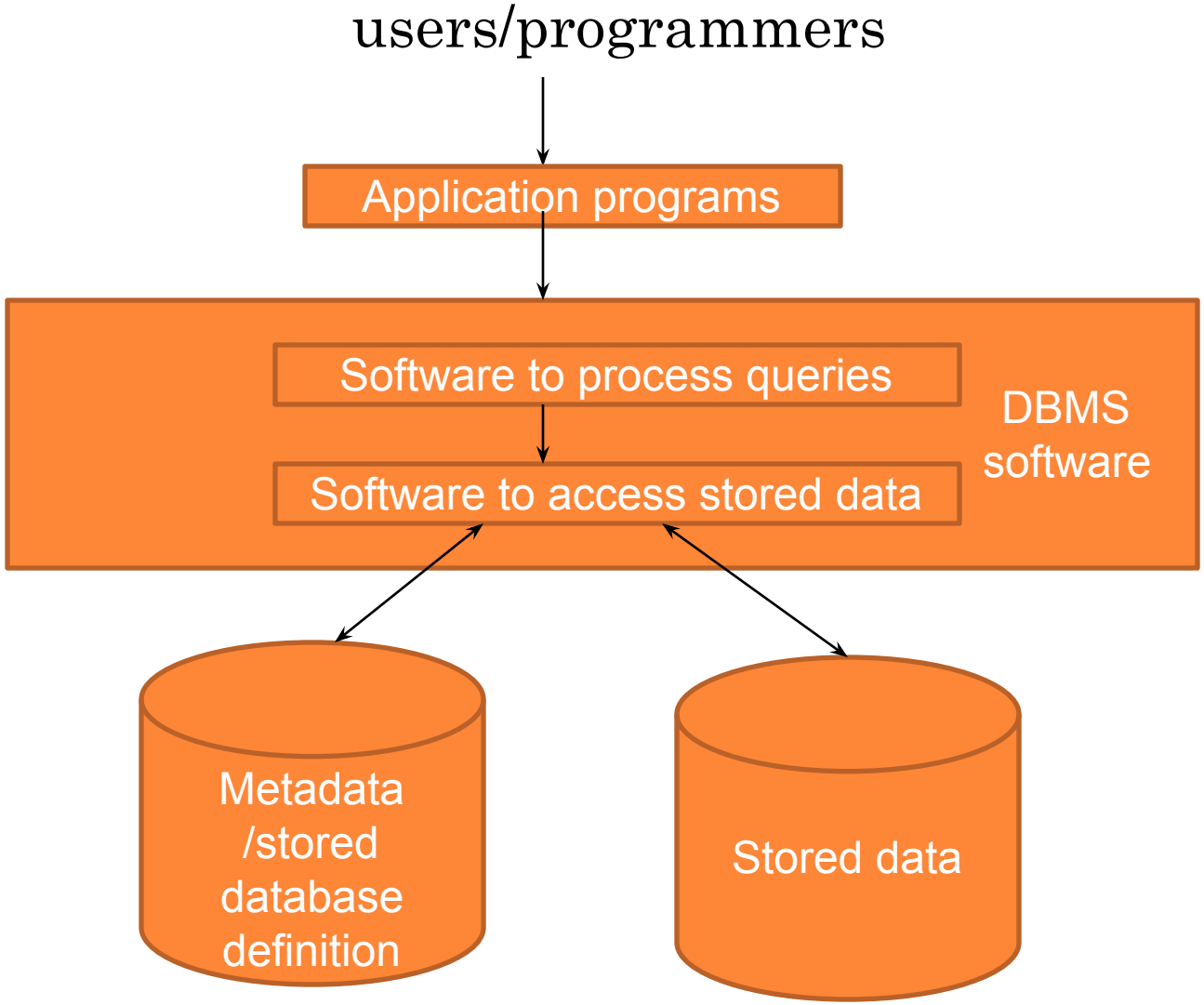
Properties

- A database represents some aspects of the real world called as mini world, changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning.
- A database is designed, built, and populated with data for a specific purpose.

- A database can be of any size and varying complexity.

  E:g: library, income tax department
- A database may be generated and maintained manually or it may be computerized
- The DBMS is a general-purpose software systems that facilitates the process of defining, constructing, manipulating, retrieving and sharing database among various users and applications.
- DBMS include protecting the database and maintaining it over long period of time.
- Protection includes both system protection against hardware or software malfunctions, and security protection unauthorized access.

A simplified database system environment



- A database is a well-organized collection of data that are related in a meaningful way which can be accessed in different logical orders but are stored only once.
- A database system is nothing more than a computer based record-keeping system.
- It is a collection of interrelated data stored together without harmful or unnecessary redundancy.

Data base system applications

Data base is widely used

- Banking
- Airlines
- Universities
- Credit card transactions
- Telecommunication
- Finance
- Sales
- Library
- Manufacturing
- Human resources

Database system Vs file systems

File system

- One way to keep the information on the computer and store it in operating system files.
- This system in order to allow the users to manipulate the information, the system must have number of application programs to manipulate the files.

File processing systems has number of major disadvantages:

Data redundancy

- Same information may be duplicated in several places
- Eg: address and telephone number of a particular customer may appear in a file that consists saving account records and in a file that consists of checking- account records

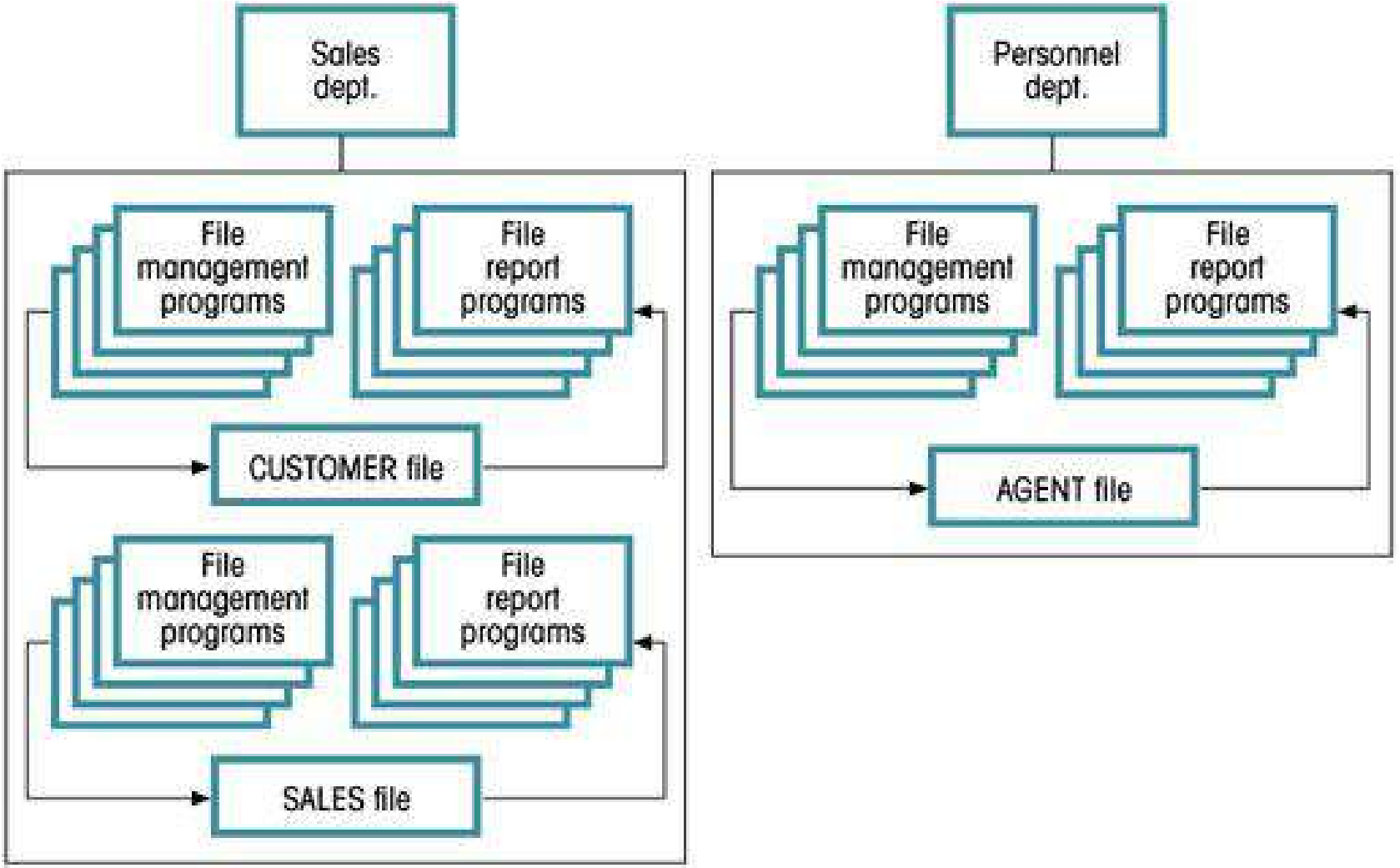


FIGURE 1.5 A SIMPLE FILE SYSTEM

Data inconsistency

- Various copies of same data may no longer agree
- Eg: change of customer address may be reflected in savings account records but not in other records such as checking- account record .

Difficulty in accessing data

- File processing environment do not allow wanted data to de retrieved in convenient and efficient way.
- Eg: bank officer want the information of customers who lives in Hyderabad area, then the data processing department takes few days time to generate the information either manually or to write application program.
- If he want to trim the query again few days are required.

Data isolation

- As the data is scattered in different files , and the files may be in different formats, writing a new application programs to retrieve the appropriate data is difficult.

Integrity problem

- Data values stored in the data base must satisfy certain consistency constraints .
- E:g: (acc-balance>2500) this become a part of application program
- The constraints has changed or added it is difficult to change the program code.

Atomicity problems

- Failures of computer may leave database in an inconsistent state with partial updates carried out in file systems
- The fund transfer must be atomic
- Eg : transfer of funds from one account to another should either complete or not happen at all before the system fails.

Concurrent -access anomalies

- To provide supervision is difficult because data may be accessed by many different application program that have not been coordinated previously
- E:g : account 500 –A withdraw 50 and B withdraw 100 the balance should be 350 but it will shows as 450 for A and 400 for B which are not correct.

Security problems

- Every user of the database system should not access all the data
- As application programs are added to the file-processing system in an ad hoc manner, maintaining security is difficult.
- E:g: payroll personnel – bank employees, not customer accounts.

Advantages of DBMS

- Control of data redundancy
- Data consistency
- More information from the same amount of data.
- Sharing of data
- Easily Maintenance
- Improved data integrity
- Improved security
- Increased productivity
- Reduce time
- Backup and recovery services
- multiple user interface

Disadvantages of DBMS

- Complexity
- Size
- Cost of DBMS(Hardware and Software)
- Performance
- Higher impact of a failure

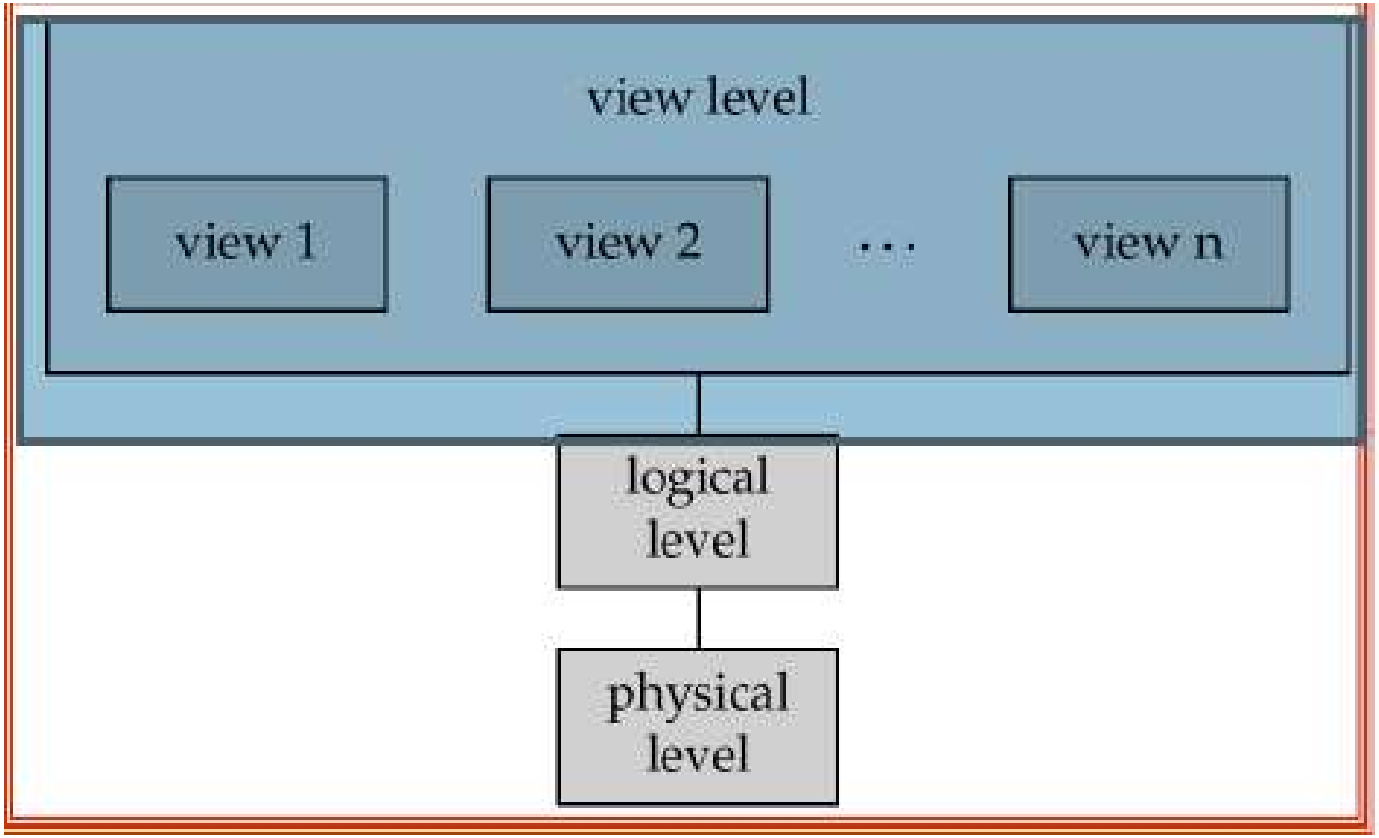
View of data

- A major purpose of a database system is to provide users with an abstract view of data where the system hides certain details of how the data are stored and maintained

Data abstraction

- Complexity is hidden by the developers from the users through several levels of abstractions.

- Physical level
- Logical level
- View level


view level
view 1
view 2
. . .
view n
logical level
physical level

Physical level

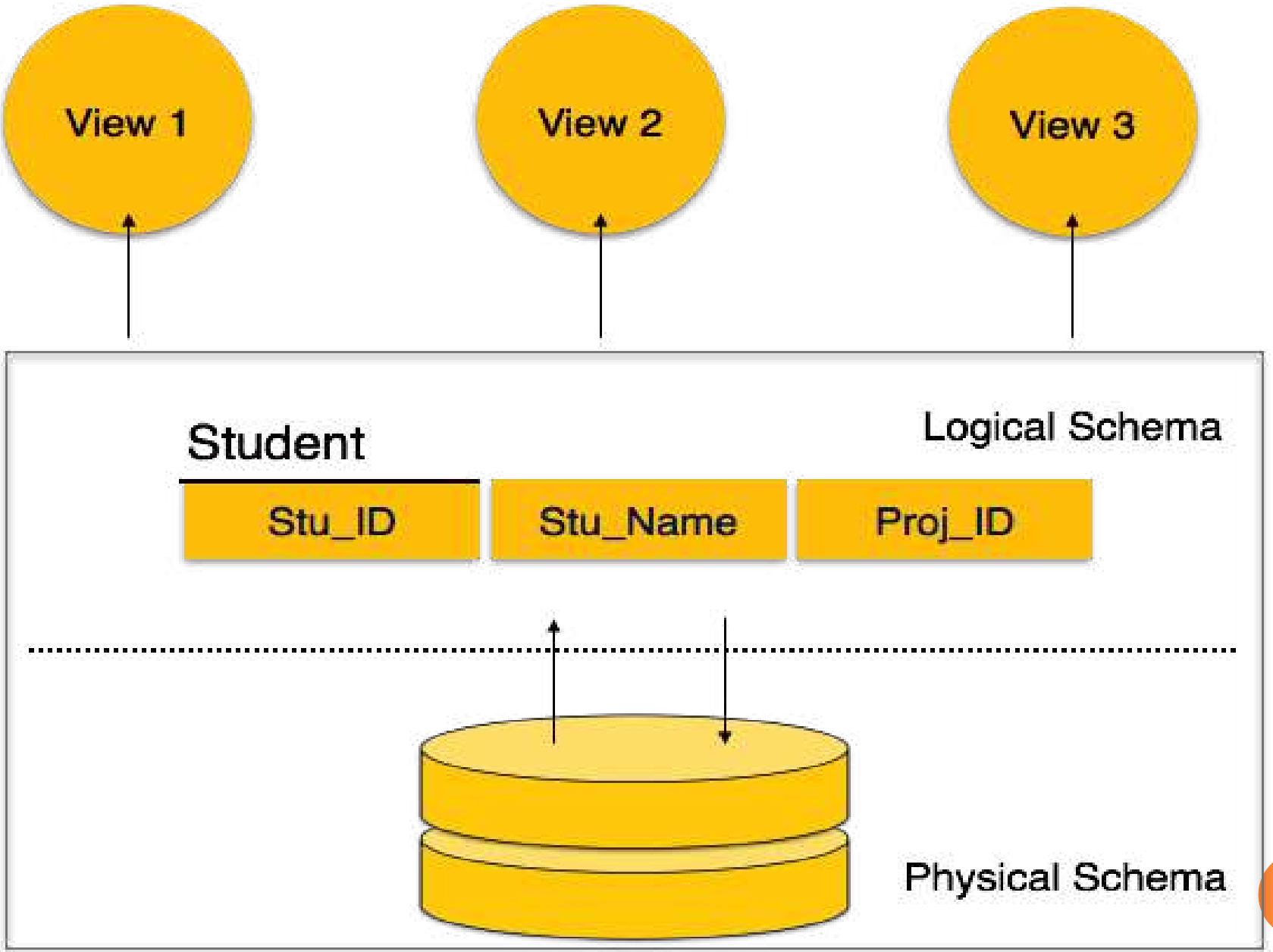
- How the data are stored.
- Data base administrators will be aware of certain details collected from the data base programmers.
- Lowest level of abstraction.

Logical level

- What data is stored in the database and what relationship exist among those data.
- Here each record is describe by the type definition and the interrelationship among all the records.
- Programmers & data administrators works at this level.
- Second level of abstraction.

View level

- This is the highest level of abstraction which describes only a part of the entire program.
- The system may provide many views from the same database.
- Application programs hide details of data types.
- Views can also hide information for security purposes.
- Database users will see these views


View 1
View 2
View 3
Student
Logical Schema
Stu_ID
Stu_Name
Proj_ID
Physical Schema

Data models

- It is the structure of a data base
- It is the collection of describing data, data relationships , and consistency constraints

These are classified as 4 categories

1. **Relational models**
2. **Entity-relationship model (E-R models)**
3. **Object-based data model**
4. **Semi structured data model**

1.Relational models

- Relational Model is the most widely used model.
- In this model, the data is maintained in the form of a two-dimensional table.
- All the information is stored in the form of row and columns.
- The basic structure of a relational model is tables. So, the tables are also called *relations* in the relational model.
- ***Example:*** In this example, we have an Employee table.

Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

EMPLOYEE TABLE

Deffinitions

- ***Entities:*** Entity is a real-world thing. It can be a person, place, or even a concept.

  Example: Teachers, Students, Course, Building, Department, etc are some of the entities of a School Management System.

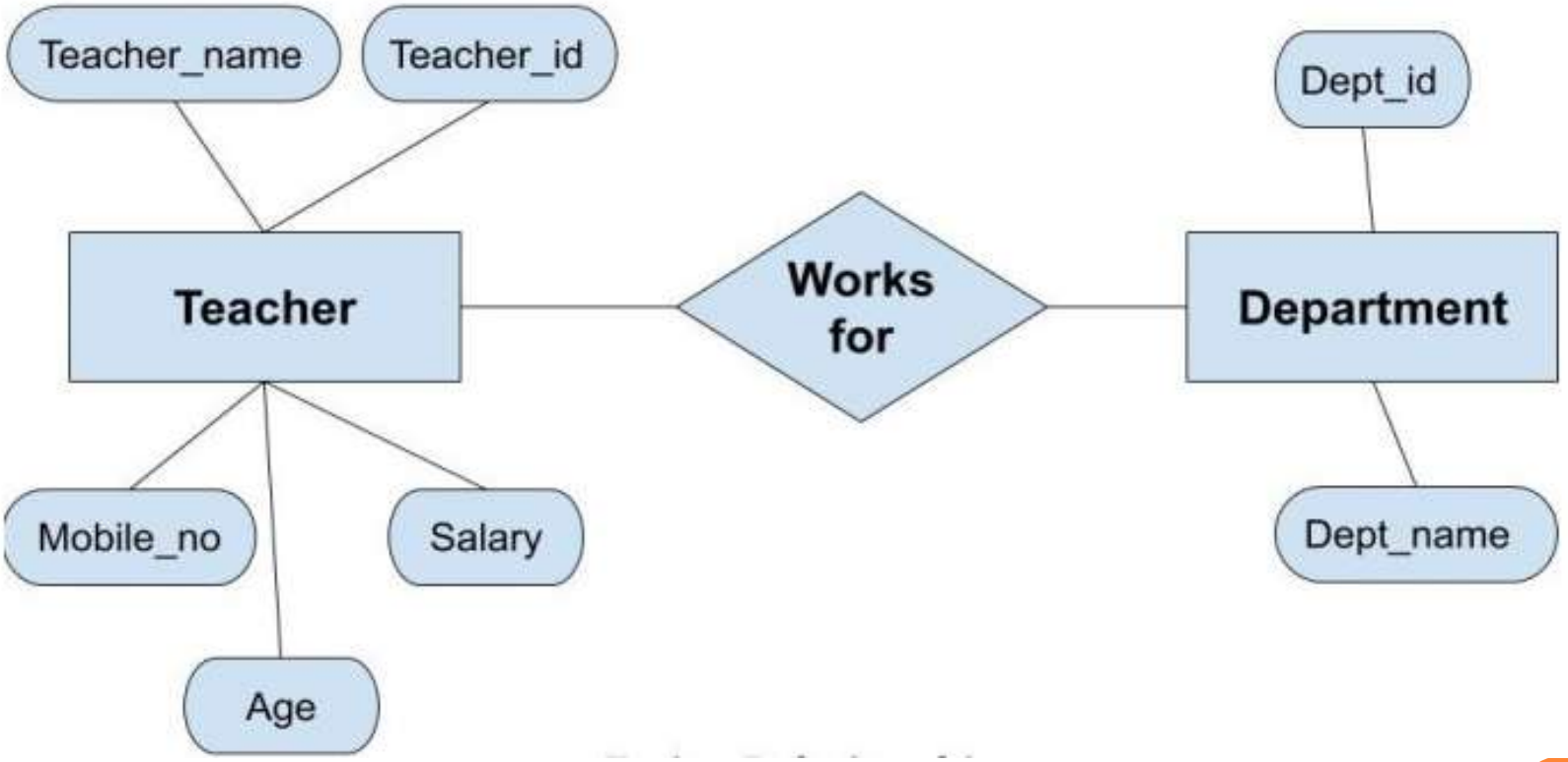
- ***Attributes:*** An entity contains a real-world property called attribute. This is the characteristics of that attribute.

  Example: The entity teacher has the property like teacher id, salary, age, etc.

- ***Relationship:*** Relationship tells how two attributes are related.

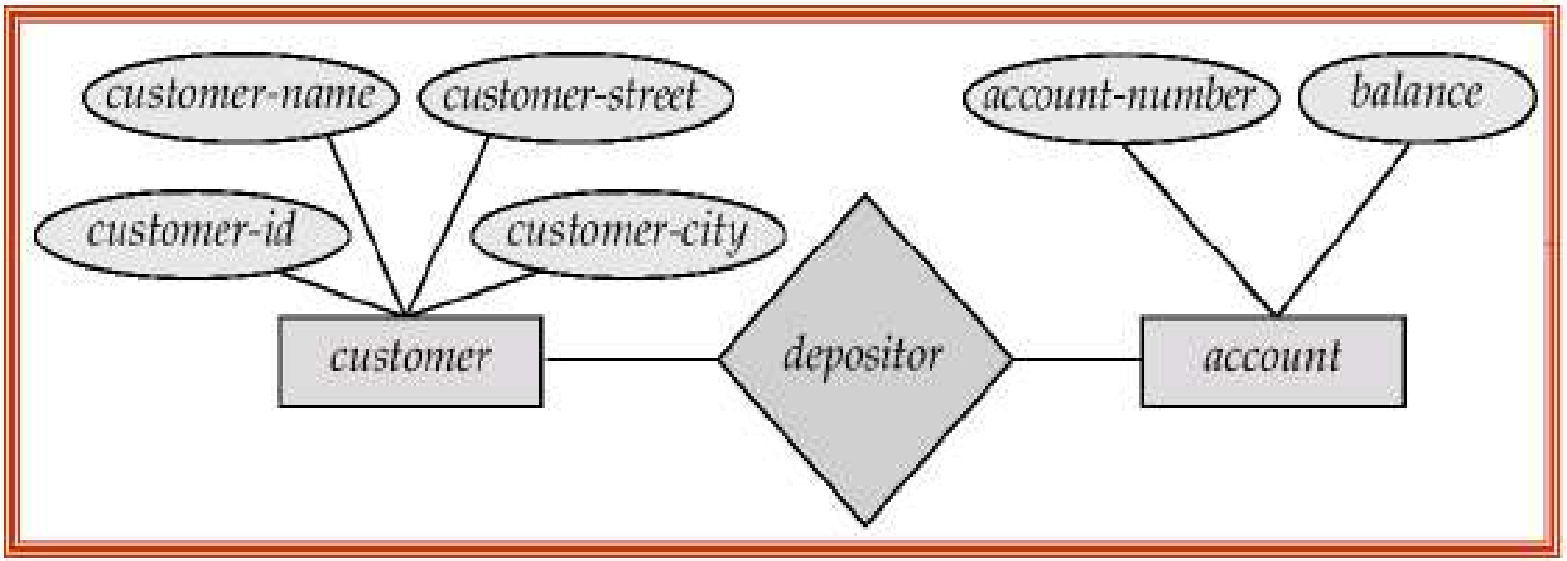
  Example: Teacher works for a department.

Example

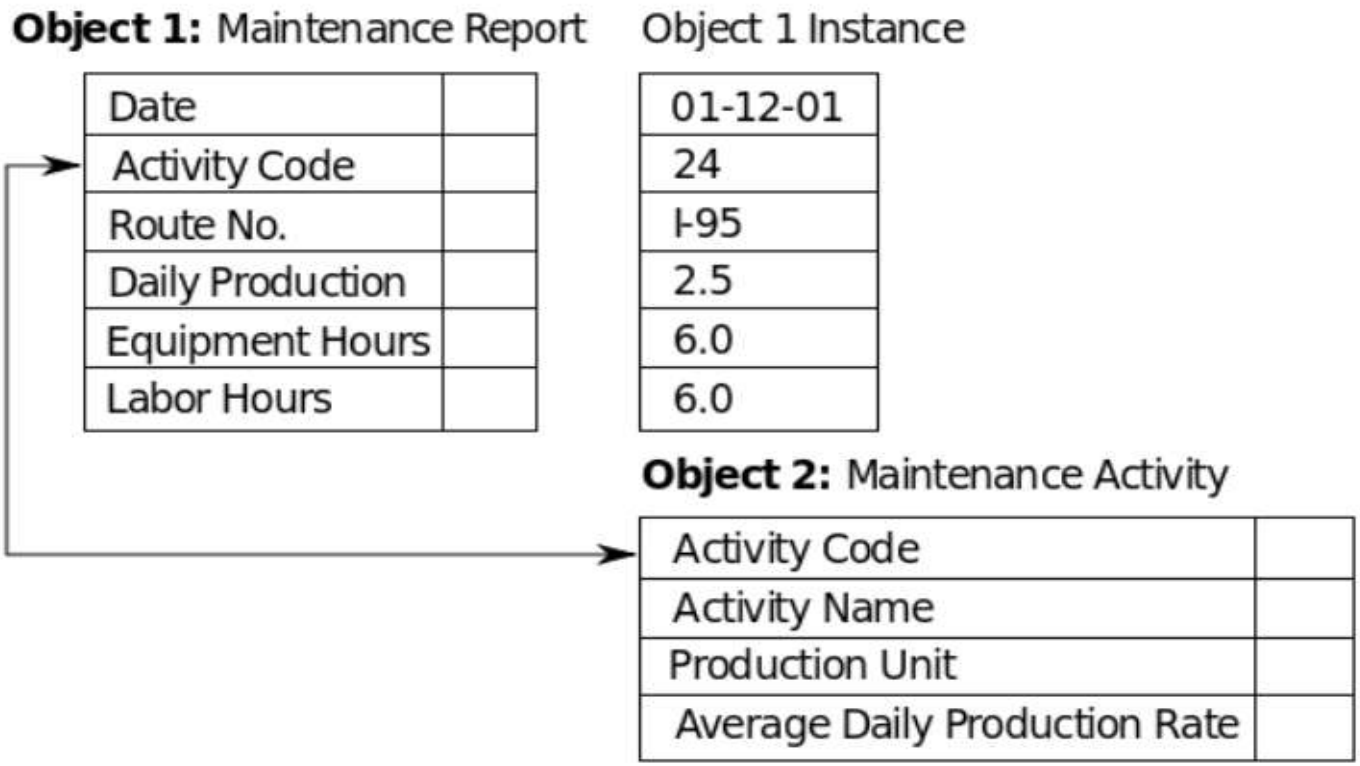

Teacher_name
Teacher_id
Dept_id
Teacher
Works for
Department
Mobile_no
Salary
Dept_name
Age

2. Entity-Relationship Model (E-R Models)

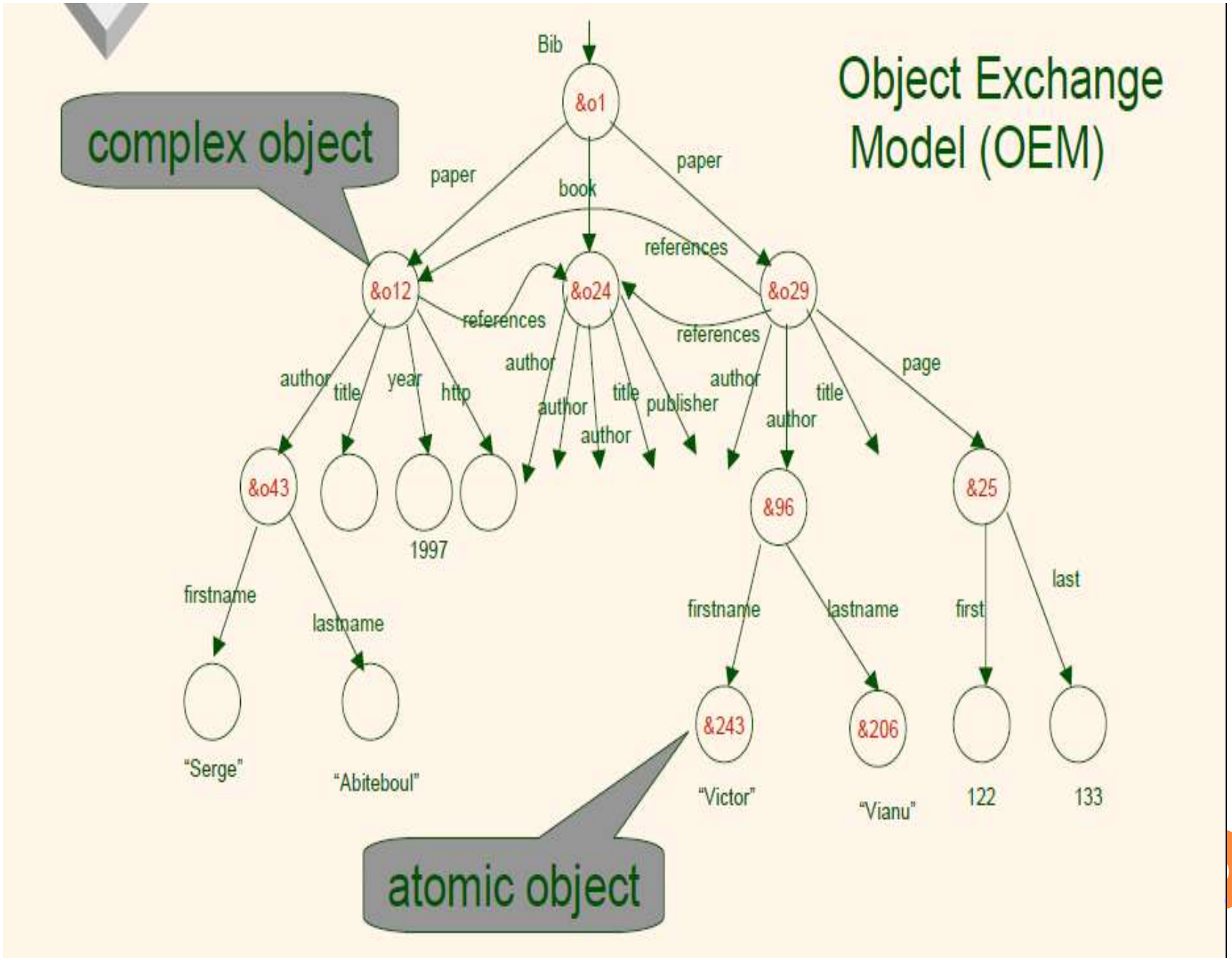
- These are the collection of objects called entities and relationship among these objects.



3.Object-Based Data Model



Object Exchange Model (OEM)

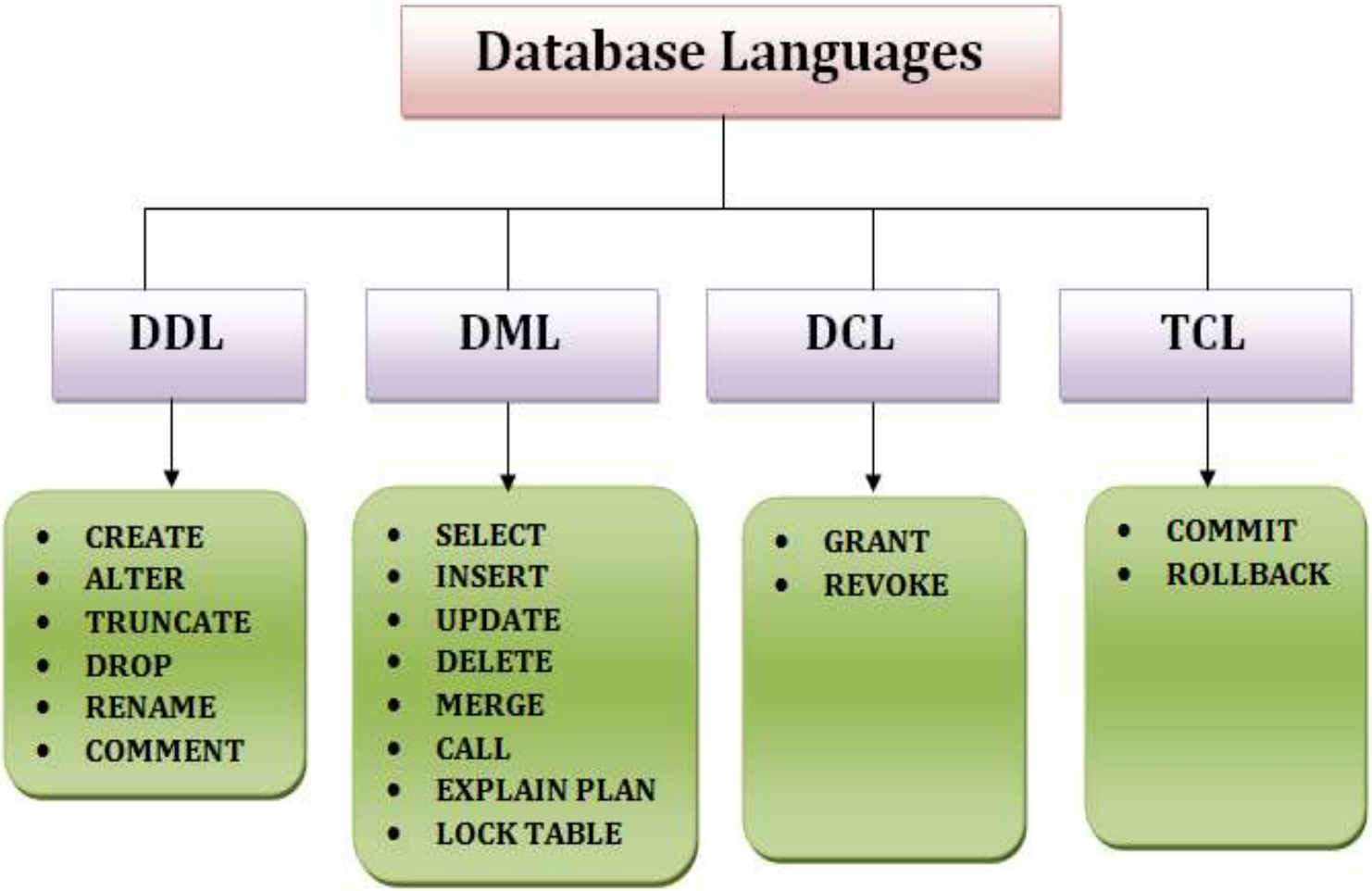


Database Languages

- A DBMS has appropriate languages and interfaces to express database queries and updates.

- Database languages can be used to read, store and update the data in the database.

Types of Database Language



1. Data Definition Language

- **DDL** stands for **D**ata **D**efinition **L**anguage. It is used to define database structure or pattern.
- It is used to create schema, tables, indexes, constraints, etc. in the database.
- Using the DDL statements, you can create the skeleton of the database.
- Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

- Here are some tasks that come under DDL:
  - **Create:** It is used to create objects in the database.
  - **Alter:** It is used to alter the structure of the database.
  - **Drop:** It is used to delete objects from the database.
  - **Truncate:** It is used to remove all records from a table.
  - **Rename:** It is used to rename an object.
  - **Comment:** It is used to comment on the data dictionary.

2. Data Manipulation Language

- **DML** stands for **D**ata **M**anipulation **L**anguage. It is used for accessing and manipulating data in a database. It handles user requests.
- Here are some tasks that come under DML:
  - **Select:** It is used to retrieve data from a database.
  - **Insert:** It is used to insert data into a table.
  - **Update:** It is used to update existing data within a table.
  - **Delete:** It is used to delete all records from a table.
  - **Merge:** It performs UPSERT operation, i.e., insert or update operations.
  - **Call:** It is used to call a structured query language or a Java subprogram.
  - **Explain Plan:** It has the parameter of explaining data.
  - **Lock Table:** It controls concurrency.

3. Data Control Language

- **DCL** stands for **D**ata **C**ontrol **L**anguage. It is used to retrieve the stored or saved data.
- The DCL execution is transactional. It also has rollback parameters.
- Here are some tasks that come under DCL:
  - **Grant:** It is used to give user access privileges to a database.
  - **Revoke:** It is used to take back permissions from the user.

4. Transaction Control Language

- TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.
- Here are some tasks that come under TCL:
  - **Commit:** It is used to save the transaction on the database.
  - **Rollback:** It is used to restore the database to original since the last Commit.

Database Users

- Users are differentiated by the way they expect to interact with the system
- Application programmers – Application programmers are computer professionals they interact with system through DML calls embedded in a program written in a host language (C,PASCAL,PL/1 ..etc..)
- Sophisticated users –Interact with the system without writing programs .They form their requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework like CADD Systems, Expert systems, complex data systems (audio, video)etc..
- Naïve users – invoke one of the permanent application programs that have been written previously
  - E.g. Bank tellers, clerical staff, Rly reservation counter clerks, ATMS

Transaction Management

- A *transaction* is a collection of operations that performs a single logical function in a database application
- Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Storage Management

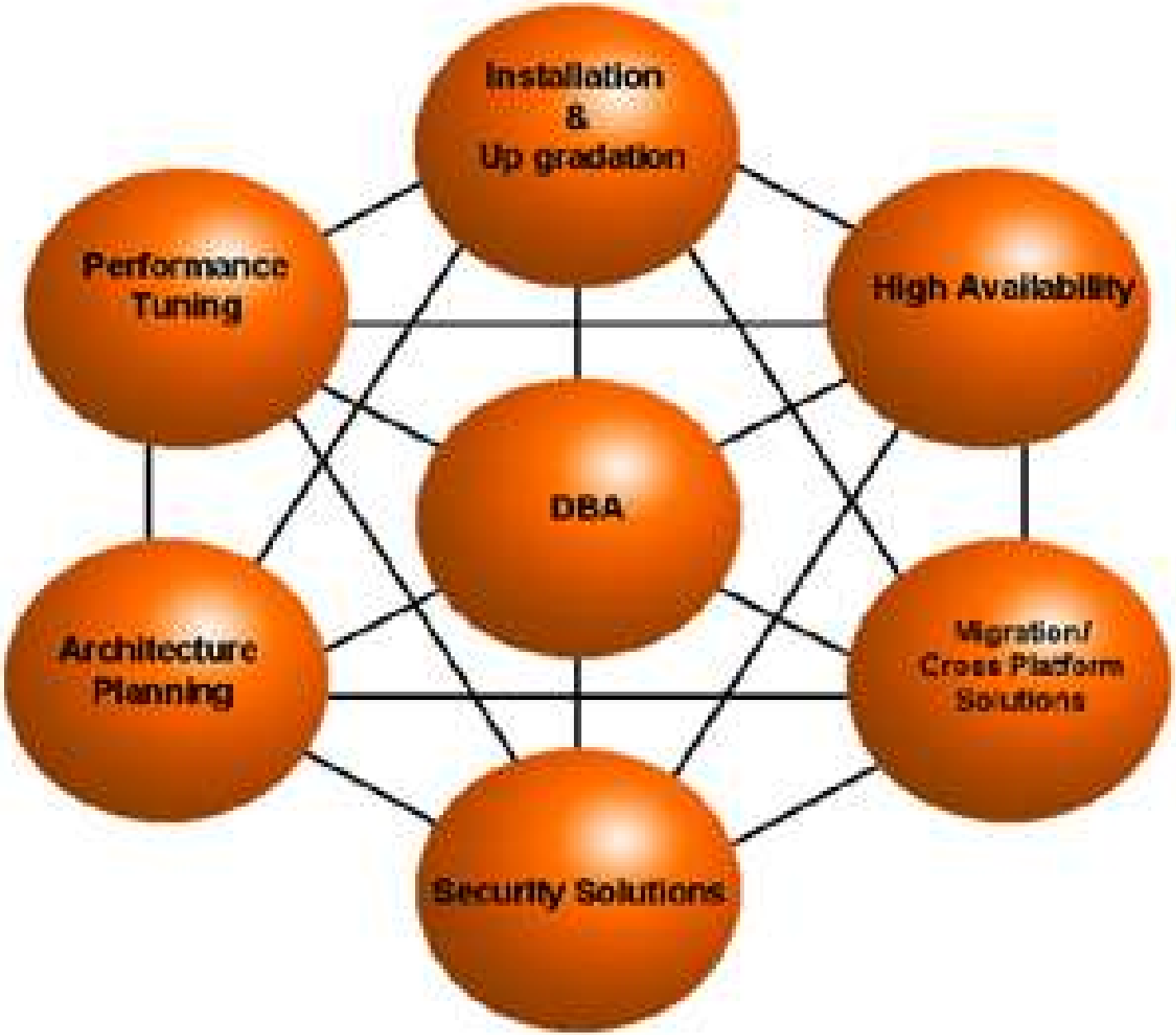
- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application .
- programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
  - interaction with the file manager
  - efficient storing, retrieving and updating of data

Database Administrator

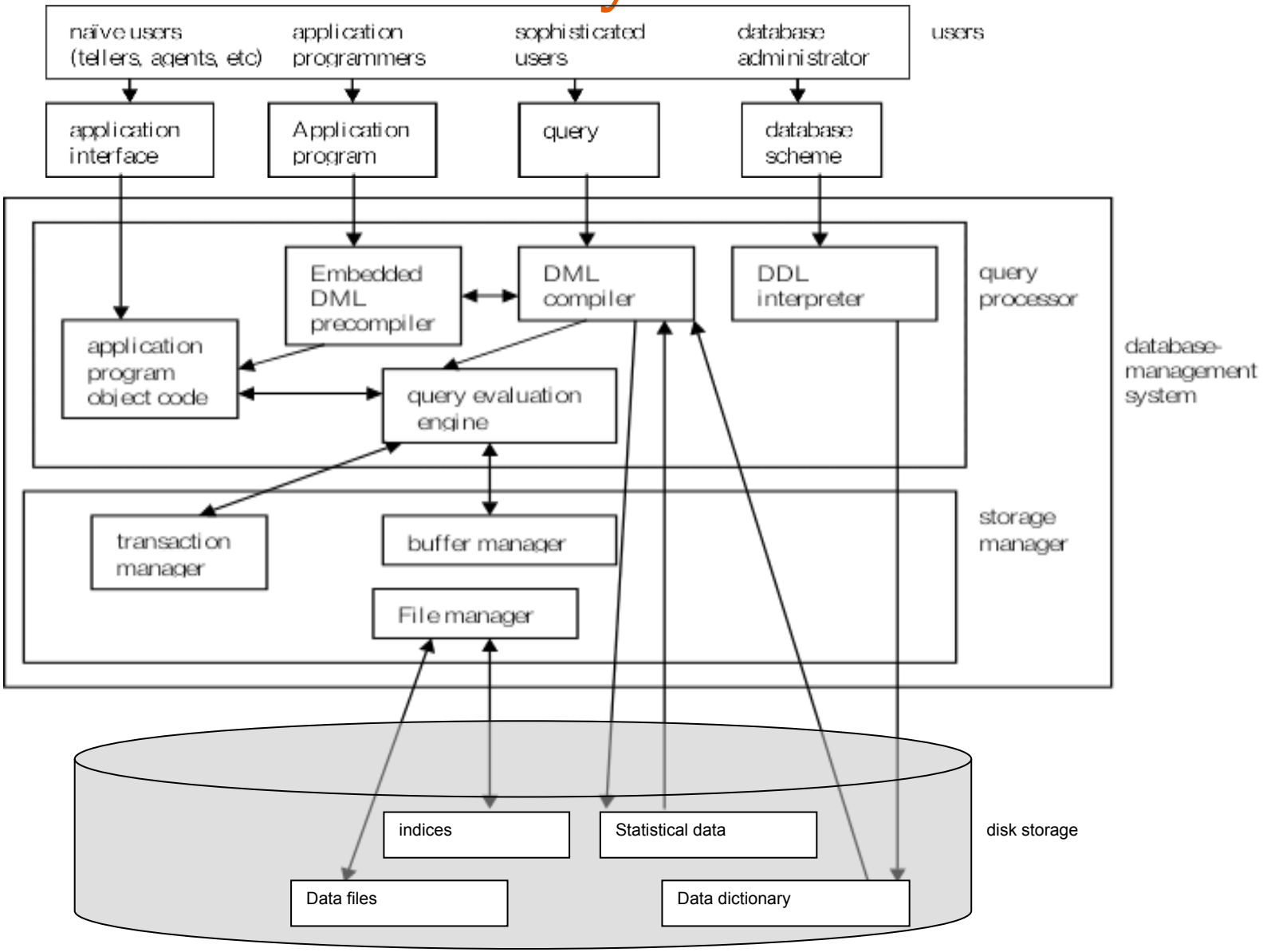
- A **Database** Administrator (**DBA**) is individual or person responsible for controlling, maintenance, coordinating, and operation of **database** management system.

- Their role also varies from configuration, **database** design, migration, security, troubleshooting, backup, and data recovery.

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.


Installation
&
Up gradation
Performance
Tuning
High Availability
DBA
Architecture
Planning
Migration/
Cross Platform
Solutions
Security Solutions

Overall Database System Structure



Thank you