UNIT-I

CONCEPTUAL MODELING

1. INTRODUCTION

- **Data:** data is a collection of raw facts and figures.
- **Database:** database is a collection of interrelated data.
- **DBMS:** database is a collection of interrelated data and set of programs can access that system.

1.1. DATABASE SYSTEM APPLICATIONS:

1. Enter Price Information:
   - *Sales:* customers, products, purchases
   - *Accounting:* payments, receipts, account balance, assets.
   - *Human Resources:* employee records, salaries, tax deductions
   - *Manufacturing:* production, inventory, orders, supply chain
   - *Online Retails:* order tracking, customized recommendations
2. *Banking and Finance:* all transactions
   - *Credit card Transaction:* generation of monthly statements.
   - *Finance:* storing information about holdings and sales,
3. *Universities:* registration, grades
4. *Airlines:* reservations, schedules
5. *Telecommunications*: keeping records of calls made, generating monthly bills

1.2. PURPOSE OF DATABASE SYSTEMS:

In the early days, database applications were built directly on top of file systems

1.2.1. Drawbacks of using file systems to store data:

- Data redundancy and inconsistency: Multiple file formats, duplication of information in different files
- Difficulty in accessing data: Need to write a new program to carry out each new task
- Data isolation: multiple files and formats
- Integrity problems: Hard to add new constraints or change existing ones
- Atomicity of updates: Failures may leave database in an inconsistent state with partial updates carried out. Example: Transfer of funds from one account to another should either complete or not happen at all
- Concurrent access anomalies: Example: Two people reading a balance and updating it at the same time

1.3. VIEW OF DATA:

A database is a collection of interrelated data and set of programs that allow users to access and modify these data. A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data stored and maintained.

1.3.1. Data Abstraction:

Major purpose of DBMS is to provide users with abstract view of data i.e. the system hides certain details of how the data are stored and maintained. Since database system users are not computer trained, developers hide the complexity from users through 3 levels of abstraction, to simplify user's interaction with the system.

Levels of Abstraction

- Physical level of data abstraction: How the data are actually stored. This s the lowest level of abstraction which describes how data are actually stored.

- Logical level of data abstraction: This level hides what data are actually stored in the database and what relations hip exists among them. Describes data stored in database, and the relationships among the data.

- View Level of data abstraction: View provides security mechanism to prevent user from accessing certain parts of database. application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

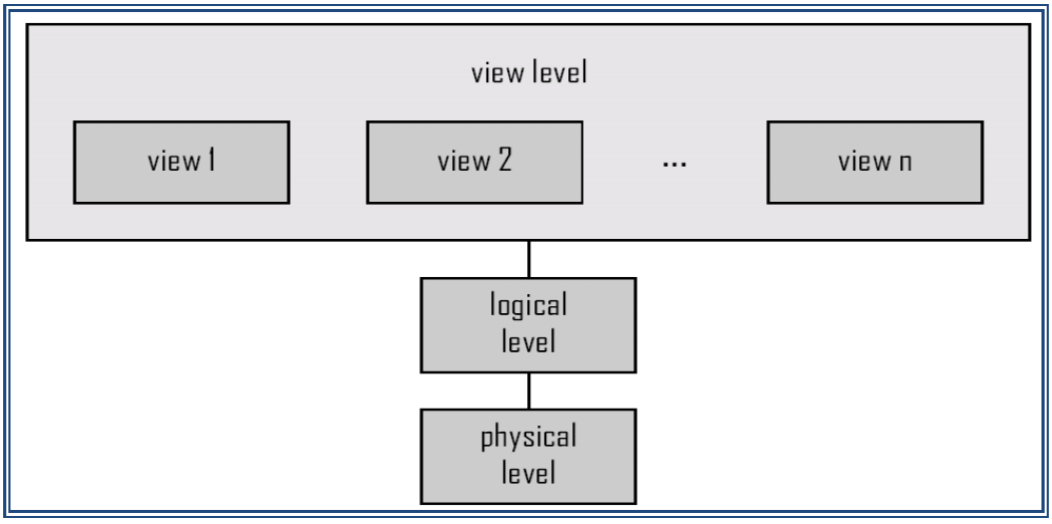


Figure 1.TheThree level of abstraction.

1.3.2. Instances and schemas:

Instance: The collection of information stored in the database at a particular movement is called an instance of the database.

Similar to types and variables in programming languages

Schema: the overall design of the database is called the database schema.

Example: The database consists of information about a set of customers and accounts and the relationship between them .Analogous to type information of a variable in a program

- **Physical schema**: database design at the physical level.
- **Logical schema**: database design at the logical level.

1.3.3. Data models:

Data Model: Underlying the structure of a database is the data model,
A collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints.

- Relational model: The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name, Tables are also called known as relations.

- Entity-Relationship Model: The Entity –Relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects.

  An entity is a "thing" or "object" in the real world that is distinguishable from other object

- Object-Based Data Models: Object-oriented Programming (especially in Java, C++, or C#).

- Semi structured Data Model: The semi structured data model permits the specification of data where individual data items of the same type may have different sets of attributes.

- Other older models:
  - Network Model
  - Hierarchical Model

1.4. DATA BASE LANGUAGES:

A Database provides a DDL to specify the database schema and a DML to express database queries and updates.

1.4.1 Data-Manipulation Language

A data-manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model.

The types of access are:

- Retrieval of information stored in the database
- Insertion of new information into the database
- Deletion of information from the database
- Medication of information stored in the database

There are basically two types:

- Procedural DMLs require a user to specify what data are needed and how to get those data.
- Declarative DMLs (also referred to as non procedural DMLs) require user to specify what data are needed without specifying how to get those data.

1.4.2. Data- Definition Language (DDL):

We specify a database schema by a set of definitions expressed by a special language called a data-definition language (DDL).The DDL is also used to specify additional properties of the data.

SQL provides a rich DDL that allows one to define tables, integrity constraints, assertions, etc…

Example:
```
create table account (
    account_number    char(10),
    branch_name       char(10),
    balance           integer)
```

In addition, the DDL statement updates the data dictionary, which contains metadata; the schema of a table is an example of metadata.

1.5. DATA BASE ACCESS FROM APPLICATION PROGRAMS:

Application programs are programs that are used to interact with the database.

To access the database, DML Statements need to be executed form the host language. ‘

There are two ways o do this.

- By Providing an Application Program interface (set of procedures) that can be used to send DML and DDL statement to the database and retrieve the results.(ODBC and JDBC).

- By extending the host language syntax to embed DML calls within the host language program. A special character prefaces DML calls and preprocessor called the DML pre complier ,converts the DML statements to normal procedure calls in the host language

SQL: widely used non-procedural language

Example 1: Find the name of the customer with customer-id 192-83-7465

select *customer.customer_name*
from *customer*
where *customer.customer_id* = ‘192-83-7465’

Example 2: Find the balances of all accounts held by the customer with customer-Id 192-83-7465.

select *account.balance*
from *depositor, account*
where *depositor.customer_id* = ‘192-83-7465’ and
depositor.account_number = account.account_number

Example 3: Find the name of the customer with customer-id 192-83-7465

select *customer.customer_name*
from *customer*
where *customer.customer_id* = ‘192-83-7465’

1.6. 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.

1.7. THE QUERY PROCESSOR

The query processor components include:

- DDL interpreter, which interprets DDL statements and records the definitions in the data dictionary.
- DML compiler, which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs query optimization; that is, it picks the lowest cost evaluation plan from among the alternatives.

Query evaluation engine, which executes low-level instructions generated by the DML compiler.

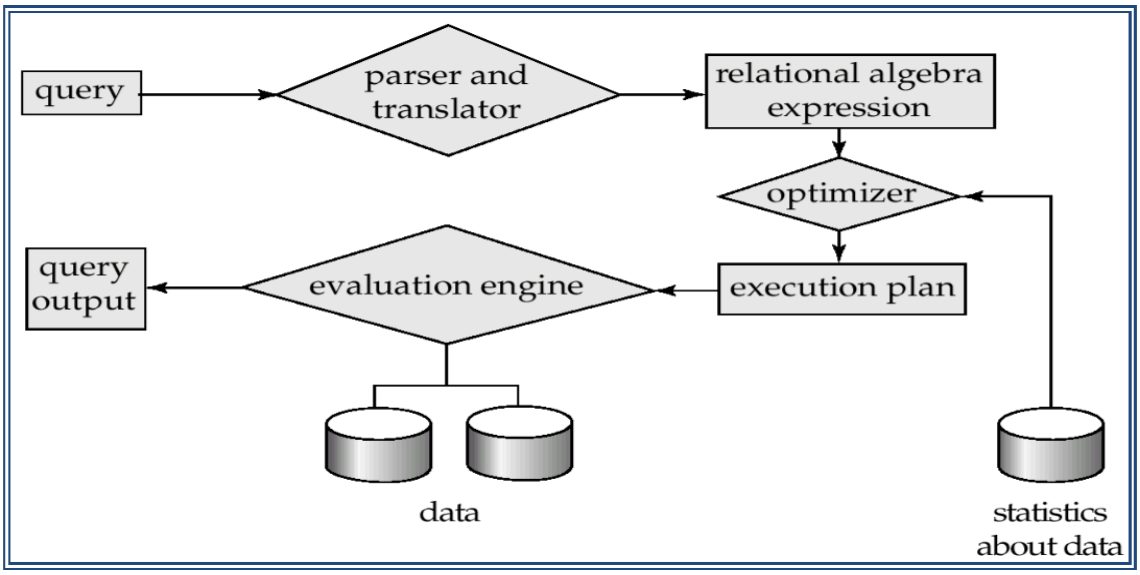


Figure 1.7.The Query Processor

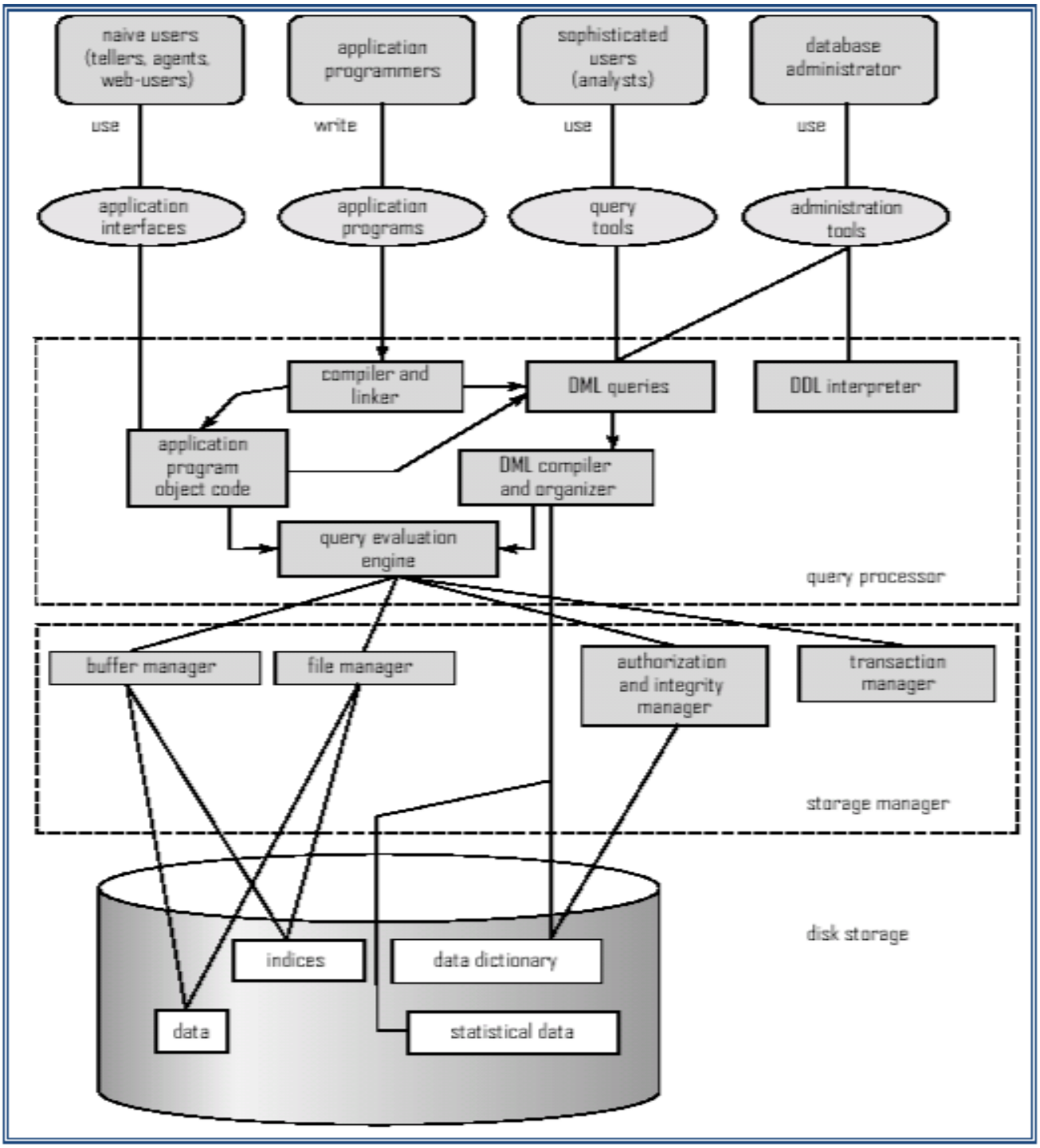
1.8. DATABASE ARCHITECTURE:

The architecture of database systems is greatly influenced by the underlying computer system on which the database is runs:

Database system can be.

- Client-server
- Parallel (multiple processors and disks)
- Distributed

Overall System Structure



1.8.1. Database Application Architectures:

- Database applications are usually partitioned into two or three parts, as in Figure1.8.1..In a two-tier architecture, the application resides at the client machine, where it invokes database system functionality at the server machine through query language statements.

– Application program interface standards like ODBC and JDBC are used for interaction between the client and the server. In contrast,

– In a three-tier architecture, the client machine acts as merely a front end and does not contain any direct database calls.

– Instead, the client end communicates with an application server, usually through a forms interface.

– The application server in turn communicates with a database system to access data.

– The business logic of the application, which says what actions to carry out under what conditions, is embedded in the application server, instead of being distributed across multiple clients.

– Three-tier applications are more appropriate for large applications, and for applications that run on the World Wide Web.

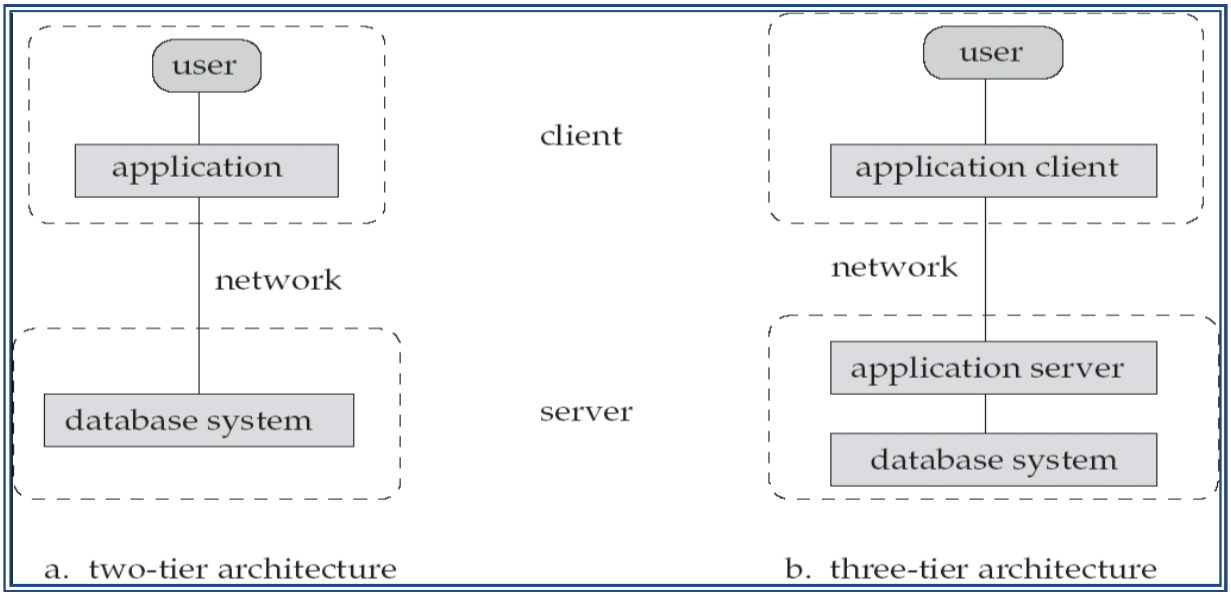


Figure1.8.1.Two-tier and Three –tier architecture.

1.9. DATABASE USERS AND ADMINISTRATORS:

A primary goal of a database system is to retrieve information from and store new information into the database. People who work with a database can be categorized as database users or database administrators.

1.9.1.Data base Users and User Interfaces

There are four different types of database system users, differentiated by the way they expect to interact with the system.

Different types of user interfaces have been designed for the different types of users.

- Naïve users **are** unsophisticated users who interact with the system by invoking one of the application programs that have been written previously. For example, a clerk in the university who needs to add a new instructor to Users are differentiated by the way they expect to interact with the system department A invokes a program called New - hire. This program asks the clerk for the name of the new instructor, her new ID, the name of the department (that is, A), and the salary

- Application programmers: are computer professionals who write application programs. Application programmers can choose from many tools to develop user interfaces. Rapid application development (RAD) tools are tools that en- able an application programmer to construct forms and reports with minimal programming effort.

- Sophisticated users: interact with the system with out writing programs. In- stead, they form their requests either using a database query language or by using tools such as data analysis software. Analysts who submit queries to explore data in the database fall in this category.

- Specialized users: are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework. Among these applications are computer aided design systems, knowledge- base and expert systems, systems that store data with complex data types (for example, graphics data and audio data),and environment-modeling systems.

1.9.2. Database Administrator

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a data base administrator (DBA).

The functions of a DBA include:

- Schema definition. The DBA creates the original database schema by executing a set of data definition statements in the DDL.

- Storage structure and access-method definition.

- Schema and physical-organization modification.
  - Routine maintenance.
  - Periodically backing up the database.
  - Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.
  - Monitoring jobs running on the Data base.